RTB | RWANDA TVET BOARD

RQF LEVEL 5

SWDFA501

SOFTWARE DEVELOPMENT

Frontend Application Development with React.JS

TRAINER'S MANUAL

October, 2024

RTB | RWANDA TVET BOARD

FRONTEND APPLICATION DEVELOPMENT WITH REACT.JS

KOICA Korea International Cooperation Agency

TQUM TVET Quality Management Project

2024

AUTHOR’S NOTE PAGE (COPYRIGHT)

Copies available from:

- *HQs: Rwanda TVET Board-RTB*
- *Web: www.rtb.gov.rw*
- **KIGALI-RWANDA**

Original published version: October 2024

ACKNOWLEDGEMENTS

The publisher would like to thank the following for their assistance in the elaboration of this training manual:

The publisher would like to thank the following for their assistance in the elaboration of this training manual:

Rwanda TVET Board (RTB) extends its appreciation to all parties who contributed to the development of the trainer's and trainee's manuals for the TVET Certificate V in Software Development, specifically for the module "SWDFA501: Frontend Application Development with React.JS."

We extend our gratitude to KOICA Rwanda for its contribution to the development of these training manuals and for its ongoing support of the TVET system in Rwanda

We extend our gratitude to the TQUM Project for its financial and technical support in the development of these training manuals.

We would also like to acknowledge the valuable contributions of all TVET trainers and industry practitioners in the development of this training manual.

The management of Rwanda TVET Board extends its appreciation to both its staff and the staff of the TQUM Project for their efforts in coordinating these activities.

This training manual was developed:

Under Rwanda TVET Board (RTB) Guiding Policies and Directives

RTB | RWANDA TVET BOARD

Under Financial and Technical Support Of

KOICA Korea International Cooperation Agency

TQUM TVET Quality Management Project

COORDINATION TEAM

RWAMASIRABO Aimable

MARIA Bernadette M. Ramos

MUTIJIMA Asher Emmanuel

Production Team

Authoring and Review

IRARORA Jonah

MUKESHIMANA Anastase

MINANI Gervais

Validation

NSENGIYUMVA Emmanuel

MUKARUGWIZA Annonciatha

KWIZERA Emmanuel

Conception, Adaptation and Editorial works

HATEGEKIMANA Olivier

GANZA Jean Francois Regis

HARELIMANA Wilson

NZABIRINDA Aimable

DUKUZIMANA Therese

NIYONKURU Sylvestre

MUKANGWIJE Yvonne

Formatting, Graphics, Illustrations, and infographics

YEONWOO Choe

SUA Lim

SAEM Lee

SOYEON Kim

WONYEONG Jeong

SHYAKA Emmanuel

Financial and Technical support

KOICA through TQUM Project

TABLE OF CONTENT

ACRONYMS

API: Application Programming Interface

CNAME: Canonical Name

CORS: Cross-Origin Resource Sharing

CSP: Content Security Policy

CSR: Client-Side Rendering

CSS: Cascading Style Sheet

DNS: Domain Name System

DOM: Document Object Model

Env: environment

FTP: File Transfer Protocol

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

HTTPS: HyperText Transfer Protocol Secure

IP: Internet Protocol

ISP: Internet Service Provider

ISR: Incremental Static Regeneration

JS: JavaScript

JSON: JavaScript Object Notation

JSX: JavaScript XML

NPM: Node Package Manager

NPX: Node Package Execute

Nslookup: Name server lookup

PWA: Progressive Web application

RTB: Rwanda TVET Board

SSG: Static Site Generation

SSL: Secure Socket Layer

SSR: Server-Side Rendering

TLD: Top Level Domain

TQUM: TVET Quality Management Project

UI/UX: User Interface/User eXperience

URL: Uniform Resource Locator

VS: Visual Studio

WWW: World wide Web

XSS: Cross-site scripting

INTRODUCTION

This trainer's manual includes all the methodologies required to effectively deliver the module titled **" Frontend Application Development with React.JS."** Students enrolled in this module will engage in practical activities designed to develop and enhance their competencies.

The development of this training manual followed the Competency-Based Training and Assessment (CBT/A) approach, offering ample practical opportunities that mirror real-life situations.

The trainer's manual is organized into Learning Outcomes, which is broken down into indicative content that includes both theoretical and practical activities. It provides detailed information on the key competencies required for each learning outcome, along with the objectives to be achieved.

As a trainer, you will begin by asking questions related to the activities to encourage critical thinking and guide trainees toward real-world applications in the labour market. The manual also outlines essential information such as learning hours, didactic materials, and suggested methodologies.

This manual outlines the procedures and methodologies for guiding trainees through various activities as detailed in their respective trainee manuals. The activities included in this training manual are designed to offer students opportunities for both individual and group work. Upon completing all activities, you will assist trainees in conducting a formative assessment known as the end learning outcome assessment. Ensure that students review the key readings and the points to remember section.

MODULE CODE AND TITLE: SWDFA501 FRONTEND APPLICATION DEVELOPMENT WITH REACT.JS

Learning Outcome 1: Develop React.js Application

Learning Outcome 2: Apply Tailwind CSS Framework

Learning Outcome 3: Develop NextJS Application

Learning Outcome 4: Apply Progressive Web Application

Learning Outcome 5: Publish the Application

Learning Outcome 1: Develop React.js application

Indicative contents

1.1 Preparation of React.js Environment
1.2 Applying React Basics
1.3 Applying UI Navigation
1.4 Applying React Hooks
1.5 Implementation of Events Handling
1.6 Implementation of API Integration

Key Competencies for Learning Outcome 1: Develop React.js Application

Knowledge	Skills	Attitudes
● Description of React.JS ● Description of Virtual DOM ● Description of Lifecycle Methods ● Description of UI navigation ● Description of ReactJS hooks ● Description of ReactJS events ● Description of ReactJS API integration, Security and Testing	● Installing NodeJS and Node Package Manager (NPM) ● Installing React tools and libraries ● Creating React Application ● Exploring react project structure ● Applying ReactJS components ● Applying URL Parameters and Nested Routing ● Applying React hooks ● Debouncing and Throttling Events ● Passing Arguments to Event Handlers ● Organizing API Calls ● Performing API Security and testing	● Being creative in developing up-to-date ReactJS apps ● Being Problem-Solving Oriented during development of ReactJS projects ● Paying attention to Details during collection of ReactJS project requirements ● Being updated on latest ReactJS versions ● Being collaborative while developing of robust ReactJS applications

Duration: 40 hrs

Learning outcome 1 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe clearly ReactJS common terminologies based on application requirements
2. Install properly NodeJS and NPM based on application requirements
3. Install properly required packages and libraries according to application requirements
4. Create properly ReactJS application based on user requirements
5. Explore clearly React project structure based on application development techniques
6. Apply correctly React Basics based on application requirements
7. Apply correctly UI Navigation based on application navigation techniques
8. Apply Properly React hooks based on application navigation best practices
9. Implement correctly event handling techniques based on application navigation best practices
10. Implement correctly API integration based on Integration techniques

Resources

Equipment	**Tools**	**Materials**
● Computer	● VS Code ● Browser ● Terminal	● Internet ● Electricity

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Avail a computer lab with internet connectivity
- Avail a VS Code and Browser installed on all computers to be used
- Avail a sample Backend application developed containing APIs to be integrated with frontend

Indicative content 1.1: Preparation of React.js environment

Duration: 10 hrs

Theoretical Activity 1.1.1: Description of ReactJS environment

Notes to the trainer:

- While delivering this content small groups can be used to describe REACTJS environment.
- Avail videos to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Explain the following concepts in ReactJS

a) ReactJS
b) Components
c) JSX
d) Props
e) State
f) Lifecycle methods
g) Hooks
h) Virtual Dom
i) Router
j) Redux

ii. Explain the uses of ReactJS

iii. What are the main Features of ReactJS

Step 2: Ask trainees to write their findings on paper or flipchart
Step 3: Engage trainees to present their findings
Step 4: Provide an expert view and clarify the ideas
Step 5: Address any question or concern raised by the trainees
Step 6: Ask the trainees to read the key readings 1.1.1 in their manual

Points to Remember

- **ReactJS** is a JavaScript library for building user interfaces, utilizing components to encapsulate UI elements.
- **JSX**, a syntax extension, allows HTML-like code within JavaScript.
- **Components** use **props** for data passing and **state** for internal data management. In react JS environment there are two types of components which are: class based components and functional based components.
- **React lifecycle methods** manage component updates.
- **Hooks** like useState and useEffect provide functional components with state and side-effects handling.
- **Virtual DOM** improves performance by efficiently updating UI changes.
- **React Router** enables dynamic navigation.
- **Redux** manages application-wide state.
- ReactJS is widely used for its modular structure, reusability, and performance optimizations, making it ideal for complex UIs.

Practical Activity 1.1.2: Installing NodeJS and Node Package Manager

Notes to the trainer

- This activity has to take place in computer lab where trainees have to Install NodeJS and Node Package manager (NPM)
- Avail videos to be used as didactic material
- Avail NodeJS-msi file for installation
- Avail a computer with internet connection

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer laboratory to install NodeJS and Node Package Manager

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to install NodeJS and Node Package Manager. While demonstrating, explain the steps to be followed

Step 4: Ask trainees to install NodeJS and Node Package Manager and monitor the activity

Step 5: Verify whether NodeJS and Node Package Manager are properly installed

Step 6: Ask trainees to read the key readings 1.1.2 in their manual

Step 7: Ask trainees to perform the task provided in the application of learning 1.1

Points to Remember

Steps to install and verify NodeJS and NPM

- Download Node.js
- Install Node.js
- Check Node.js version by using node -v
- Check NPM version by using npm -v

Practical Activity 1.1.3: Creating ReactJS application

Notes to the trainer

- This activity has to take place in computer lab where trainees have to create a ReactJS application.
- Avail videos to be used as didactic materials

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to create a ReactJS application that will display a user registration form.

Step 2: Provide clear instructions

Step 3: Demonstrate how to create a ReactJS application. While demonstrating explain the steps to be followed

Step 4: Ask trainees to create a ReactJS project as done in step 2

Step 5: Verify whether ReactJS application is created

Step 6: Ask trainees to read key readings 1.1.3 in their manual

Step 7: Ask trainees to perform the task provided in the application of learning 1.1

Points to Remember

Steps to setup a new ReactJS Project

- Open terminal inside a folder you want to create a project
- Type the command *npx create-react-app my-app* in the terminal
- After the project is created, type *cd my-app* to navigate to the project folder
- After navigating to the project folder, type *npm start* to start your application server
- The project will be initialized in a default browser on your computer with a domain name and port number

Practical Activity 1.1.4: Exploring React project structure

Notes to the trainer

- This activity has to take place in computer lab where trainees have to explore a ReactJS project
- Avail videos to be used as didactic material
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to computer lab to explore the ReactJS application files and folders
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to Explore Files and Folders. While demonstrating explain the steps to be followed
Step 4: Ask trainees to Explore Application Files and Folders and monitor the activity
Step 5: Verify whether Files and Folders are explored
Step 6: Ask trainee to read key readings 1.1.4
Step 7: Ask trainee to perform the task provided in the application of learning 1.1

Points to Remember

- Browse to the folder where your project is created
- Open Command prompt
- Type the command to open the project in your text editor
- View and analyse every file and folder from the project explorer in the text editor

Practical Activity 1.1.5: Installing additional React tools and libraries

Notes to the trainer

- This activity has to take place in computer lab where trainees have to install additional React tools and libraries.
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to install additional React tools and libraries

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to install additional React tools and libraries. While demonstrating explain the steps to be followed

Step 4: Ask trainees to install additional React tools and libraries and monitor the activity

Step 5: Verify whether additional React tools and libraries are clearly installed

Step 6: Ask trainee to read key readings 1.1.5

Step 7: Ask trainee to perform the task provided in the application of learning 1.1

Points to Remember

Method 1: Installing a single package

- Install Redux
  Command: npm install redux react-redux
- Install React Router
  Command: npm install react-router-dom
- Install Axios
  Command: npm install axios

Method 2: Install all packages using single line of command

- npm install redux react-redux react-router-dom axios

Application of learning 1.1

ABC company is an Online Car selling company, it needs a front-end application to manage its clients. You are requested to prepare React JS Environment and Create a front end application in ReactJS. Install additional react JS packages like Redux, react router and Axios.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.	NodeJS and Node Package Manager (NPM) are properly installed	NodeJS is installed		
		Node Version is checked		
		NPM Version is checked		
2.	ReactJS application is properly created	Application is created		
		Application is selected		
		Application is initiated		
3.		Project is opened in a text-editor		

	React project structure is well explored	Files and folders are seen in file Manager		
4.	Additional React tools and libraries are properly installed	Redux package is installed		
		React router is installed		
		Axios package is installed		

Indicative content 1.2: Applying React basics

Duration: 5 hrs

Practical Activity 1.2.1: Creating React Class Components

Notes to the trainer

- This activity has to take place in computer lab where trainees have to create React Components
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to:

1. Create a Class Component
2. Include a JSX in class components
3. Include props in class components

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to create class components with JSX and Props. While demonstrating explain the steps to be followed

Step 4: Ask trainees to create class components with JSX and props and monitor the activity

Step 5: Verify whether Class Components are created with props and JSX included

Step 6: Ask trainee to read key readings 1.2.1

Step 7: Ask trainee to perform the task provided in the application of learning 1.2

Points to Remember

Steps to create Class Components

- In an existing ReactJS project, create a file in the src folder and name it appropriately
- Inside the file, create a class that represents the component

- In the body of a class component, create a render () method and return the JSX scripts
- At the end of your file, export the component so it can be imported elsewhere.

Practical Activity 1.2.2: Creating React Functional Components

Notes to the trainer

- This activity has to take place in computer lab where trainees have to create React Functional Component.
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to Create a Function al Component

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to create Functional components. While demonstrating explain th e steps to be followed

Step 4: Ask trainees to create Functional components and monitor the activity

Step 5: Verify whether Functional Components are created with props and JSX included

Step 6: Ask trainee to read key readings 1.2.2

Step 7: Ask trainee to perform the task provided in the application of learning 1.2

Points to Remember

Steps to create a functional based component in ReactJS

- In an existing ReactJS project, create a file in the src folder and name it appropriately
- Inside the file, create a function that represents the component
- In the body of the functional component, return the JSX directly, which replaces the render () method in class components
- At the end of your file, export the component so it can be imported elsewhere.

Practical Activity 1.2.3: Creating React Props

Notes to the trainer

- This activity has to take place in computer lab where trainees have to create Props
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer laboratory to create Props

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to create Props. While demonstrating explain the steps to be followed

Step 4: Ask trainees to create Props and monitor the activity

Step 5: Verify whether Props are used.

Step 6: Ask trainee to read key readings 1.2.3

Step 7: Ask trainee to perform the task provided in the application of learning 1.2

Points to Remember

Steps to create Props in ReactJS

- In an existing ReactJS project, create a two files in the src folder and name them appropriately
- Inside the first file, create a functional based component, import and use a child component
- In the second file, create a child component with a functions with props passed as a function argument
- In the parent component, assign any number of attributes to the child component called in the return method.
- In the child component, access attributes in the parent component from props object
- Start the project and view the output.

Theoretical Activity 1.2.4: Description of Lifecycle methods

Notes to the trainer:

- While delivering this content small groups can be used to describe the lifecycle methods
- Avail videos to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to Describe how the following life cycle metho ds are implemented:

1. componentDidMount
2. componentDidUpdate
3. componentDidUnmount

Step 2: Engage trainees to present their findings

Step 3: Provide an expert view about the task assigned to trainees in step 1.

Step 4: Address any question or concern raised by the trainees

Step 5: Ask the trainees to read key readings 1.2.4 in their manual

Points to Remember

1. componentDidMount: Runs after a component mounts, used for initializing data
2. componentDidUpdate: Invoked after component updates due to state/prop changes, allowing reactions to changes.
3. componentWillUnmount: Triggered before unmounting, used for clean-up like removing subscriptions or timers.

Note: In modern React, these lifecycle methods can be replaced using the useEffect hook in functional component.

Application of learning 1.2.

ABC is a company located in Kigali City, Nyarugenge district. It Specializes in website development. The company wants to develop a web application using react.js. As a ReactJS developer you have been hired by the company and tasked to create a simple React app that displays a list of books. Use a **class component** to manage the book data and a **functional component** to display individual book details. Pass each book's title, author, and description as **props** from the class component to the functional component. Use **JSX** to render the book details in the UI. Ensure that the data is properly passed and displayed.

checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Class components are well created	Class component is created		
		JSX scripts are included in class components		
		Prop are used in class components		
2	Functional components are well created	Functional component well created		
		JSX scripts are included in functional components		
		Prop are used in functional components		

Indicative content 1.3: Application of UI navigation

Duration: 5 hrs

Practical Activity 1.3.1: Applying Basic React navigation

Notes to the trainer

- This activity has to take place in computer lab where trainees have to Install, configure routes and apply basic React Navigation
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to

1. Install React Route
2. Configure Route
3. Apply Basic React Navigation

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to install React Route, Configure Route and Apply Basic React Navigation. While demonstrating explain the steps to be followed

Step 4: Ask trainees to install React Route, Configure Route and Apply Basic React and monitor the activity

Step 5: Verify whether React Route is installed, Configured and applied

Step 6: Ask trainee to read key readings 1.3.1

Step 7: Ask trainee to perform the task provided in the application of learning 1.3

Points to Remember

- **Installation of React Router**
  - ✓ Open your command prompt or terminal in your project directory.
  - ✓ Run the command to install React Router:
- **Configuration of Routes**
  - ✓ First, ensure you have your React components, such as Home and About, created. These will be displayed based on the route.
  - ✓ Configure your routes in the main App.js or another component responsible for routing.
- **Application of Basic React Navigation**
  - ✓ To navigate between different routes, you can create a simple navigation bar using the Link component from react-router-dom
  - ✓ Replace Link component with traditional anchor tag (<a>) in React Router, allowing seamless navigation between pages without reloading the entire application.

Practical Activity 1.3.2: Handling 404 Pages, Redirects and URL Parameters

Notes to the trainer

- This activity has to take place in computer lab where trainees have to perform the following tasks
  1. Handle 404 Pages
  2. Create Redirects
  3. Perform URL parameters
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.3.2

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to:

1. Handle 404 Pages
2. Create Redirects
3. Perform URL parameters

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to Handle 404 Pages, Create Redirects and Perform URL Parameters. While demonstrating explain the steps to be followed

Step 4: Ask trainees to Handle 404 Pages, Create Redirects and Perform URL Parameters and monitor the activity

Step 5: Verify Handle 404 Pages, Create Redirects and Perform URL Parameters are performed

Step 6: Ask trainee to read key readings 1.3.2

Step 7: Ask trainee to perform the task provided in the application of learning 1.3

Points to Remember

- **Handling 404 pages**
  - ✓ Ensure react-router-dom is installed.
  - ✓ Set up the routing in your component
- **Redirects**
  - ✓ In your routing component, add a route that uses <Redirect> to point from the old path to the new path.
- **URL Parameters**
  - ✓ Define a route with a parameter in your App component.
  - ✓ Access the parameter via match.params inside the target component.
  - ✓ In the User component, extract the id parameter from match.params

Practical Activity 1.3.3: Applying Nested Routes

Notes to the trainer

- This activity has to take place in computer lab where trainees have to create Nested Routes
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.3.2

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab create nested route s

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to create nested Routes. While demonstrating explain the steps to be followed

Step 4: Ask trainees to create nested routes and monitor the activity

Step 5: Verify whether nested routes are created

Step 6: Ask trainee to read key readings 1.3.3

Step 7: Ask trainee to perform the task provided in the application of learning 1.3

Points to Remember

Steps to apply Nested Routing

- Create a parent component with an Outlet for child routes.
- Define child components and nested routes inside the parent route.
- Test the navigation and routes

Application of learning 1.3.

ABC is an online business company Located in Kayonza district in Eastern Province-Rwanda. It needs a ReactJS developer to develop an ecommerce website. Assume you have been hired as a ReactJS developer for the company, you are tasked with creating an ecommerce application using React. Begin by installing React Router to manage UI navigation effectively. Configure various routes for product listings, user profiles, and shopping carts. Implement basic navigation to ensure users can easily access different sections. Handle 404 pages for non-existent routes and set up redirects for outdated URLs. Additionally, utilize URL parameters for filtering products and apply nested routing for displaying reviews within individual product pages.

Checklist:

No	Criteria	Indicators	Observation	
			Yes	No
1.	Basic react navigation is well implemented	Routes are installed		
		Routes are configured		
		Basic navigation is applied		
2.	Handling 404 Pages is well implemented URL	react-router-dom is installed		
		Routing is set		
3.	Redirects are well implemented	Redirect is imported		
		Redirect is implemented		
4.	Parameters is well implemented	Route parameter is defined		
		Route parameter is accessed		
		Id parameter is extracted		

Indicative content 1.4: Application of React Hooks

Duration: 5 hrs

Theoretical Activity 1.4.1: Description of Hooks

Notes to the trainer:

- While delivering this content small groups can be used to introduce Hooks
- Avail videos to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to answer the following questions:

i. What do you understand by the term?
   a) Hooks
   b) State Hooks
   c) Effect Hooks
   d) Context Hooks
   e) Ref Hooks
   f) Callback Hooks

ii. Explain different types of Hooks

iii. What are the factors considered while selecting the right Hook to be used?

iv. Why do we combine hooks?

Step 2: Ask trainees to present their findings

Step 3: Provide an expert view about the task assigned to trainees in step 1.

Step 4: Address any question or concern raised by the trainees

Step 5: Ask the trainees to read key readings 1.4.1 in their manual

Points to Remember

- **Overview of Hooks**

  Definition: Special functions in React that allow using state and other features in functional components without classes.
- **Types of Hooks**
  - ✓ State Hooks (useState)
  - ✓ Effect Hooks (useEffect)
  - ✓ Context Hooks (useContext)
  - ✓ Ref Hooks (useRef)
  - ✓ Callback Hooks (useCallback)
- **Additional Hooks:**
  - ✓ useReducer
  - ✓ useMemo
  - ✓ useLayoutEffect
  - ✓ useImperativeHandle
- **Factors for Selecting the Right Hook**
  - ✓ State Complexity
  - ✓ Side Effects
  - ✓ Optimization
  - ✓ Context Needs
  - ✓ DOM Access
- **Reasons for Combining Hooks**
  - ✓ Modularity
  - ✓ Optimization
  - ✓ Code Organization
  - ✓ Reusability

Practical Activity 1.4.2: Implementation of Hooks

Notes to the trainer

- This activity has to take place in computer lab where trainees have to use react Hooks
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to use:

1. State Hooks
2. Effect Hooks
3. Context Hooks
4. Ref Hooks
5. Call-back Hooks

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to use State Hooks, Effect Hooks, Context Hooks, Ref Hooks and Call-back Hooks. While demonstrating explain the steps to be followed

Step 4: Ask trainees to use State Hooks, Effect Hooks, Context Hooks, Ref Hooks and Call-back Hooks and monitor the activity

Step 5: Verify whether State Hooks, Effect Hooks, Context Hooks, Ref Hooks and Call-back Hooks are used.

Step 6: Ask trainee to read key readings 1.4.2

Step 7: Ask trainee to perform the task provided in the application of learning 1.4

Points to Remember

- **Steps to implement State Hooks**
  - ✓ Import the useState hook from React.
  - ✓ Use useState to define a state variable and a function to update it.
  - ✓ Use the state variable in your component's render logic.
  - ✓ Call the state updater function to modify the state based on an event.
- **Steps to implement Effect Hook**
  - ✓ Import the useEffect hook from React.
  - ✓ Define the effect logic inside the useEffect callback.
  - ✓ Optionally return a cleanup function to run when the component unmounts.
  - ✓ Add dependencies in the second argument to control when the effect runs.
- **Steps to implement Ref Hook**
  - ✓ Import the useRef hook from React.
  - ✓ Create a ref object using useRef.
  - ✓ Attach the ref to a DOM element using the ref attribute.
  - ✓ Use the current property of the ref to interact with the DOM element.
- **Steps to implement Callback Hook**
  - ✓ Import the useCallback hook from React.

- ✓ Define a function and wrap it with useCallback to it.
- ✓ Specify dependencies to determine when the function should be re-created

Practical Activity 1.4.3: Performing Performance Optimisation

Notes to the trainer

- This activity has to take place in computer lab where trainees have to perform performance optimization
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to perform performance optimization

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to optimize performance. While demonstrating explain the steps to be followed

Step 4: Ask trainees to optimize performance and monitor the activity

Step 5: Verify whether performance optimization has been implemented

Step 6: Ask trainee to read key readings 1.4.3

Step 7: Ask trainees to perform the task provided in the application of learning 1.4

Points to Remember

To optimize performance, you have to implement the following:

- **Steps to use useMemo**
  - ✓ Identify expensive computations or derived data
  - ✓ Wrap the computation inside a useMemo hook, passing the function that returns the computed value as the first argument.
  - ✓ Provide a dependency array as the second argument, which will trigger re-computation when any of the dependencies change.
  - ✓ Use the memorized value where necessary.
- **Use useCallback Hook**

  - ✓ Identify callback functions or event handlers that are passed as props to child components or that are recreated on every render.
  - ✓ Wrap the function definition inside a useCallback hook, passing the function as the first argument.
  - ✓ Provide a dependency array as the second argument to control when the function gets recreated.
- **Steps to optimize rendering**
  - ✓ Memorize components
  - ✓ Split complex components
  - ✓ Avoid unnecessary state updates
  - ✓ Lazy loading

Practical Activity 1.4.4: Handling Complex State Logic

Notes to the trainer

- This activity has to take place in computer lab where trainees have to handle complex state logic
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to handle complex state logic
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to handle complex state logic. While demonstrating explain the steps to be followed
Step 4: Ask trainees to handle complex state logic and monitor the activity
Step 5: Verify whether Complex state logic has been implemented
Step 6: Ask trainee to read key readings 1.4.4
Step 7: Ask trainee to perform the task provided in the application of learning 1.4

Points to Remember

- **Steps to use useReducer to manage complex state logic**
  - ✓ Import useReducer from React.
  - ✓ Define the Initial State
  - ✓ Create the Reducer Function
  - ✓ Use the useReducer Hook
  - ✓ Handle Button Clicks
  - ✓ Export the Component:

Practical Activity 1.4.5: Managing Global State

Notes to the trainer

- This activity has to take place in computer lab where trainees have to use the following Libraries in management of global state
  1. Context API
  2. Redux
  3. MobX
  4. Zustand
- Avail videos to be used as didactic materials
- Avail a ReactJS project created in Practical activity 1.1.3

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to use the following Libraries in management of global state

1. Context API
2. Redux
3. MobX
4. Zustand

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to manage global state. While demonstrating explain the steps to be followed

Step 4: Ask trainees to manage global state and monitor the activity
Step 5: Verify whether global state has been managed
Step 6: Ask trainee to read key readings 1.4.5
Step 7: Ask trainee to perform the task provided in the application of learning 1.4

Points to Remember

- **Steps to use Context API**
  - ✓ Create a Context
  - ✓ Provide the Context
  - ✓ Consume the Context
- **Steps to use Redux**
  - ✓ Install Redux
  - ✓ Create Reducers
  - ✓ Create a Redux Store
  - ✓ Provide the Store
  - ✓ Connect Components
- **Steps to use MobX**
  - ✓ Install MobX
  - ✓ Create Observable State
  - ✓ Provide Store
  - ✓ Use Observer in Components
- **Steps to use Zustand:**
  - ✓ Install Zustand
  - ✓ Create a Store
  - ✓ Use the Store in Components

Application of learning 1.4.

ABC is a web Application development company located in Kimihurura – Kigali city, it's planning to create a customer's shopping site using React. As a React developer, the company has hired you to build a dashboard application using React. Utilize **State Hooks** to manage local component states and **Effect Hooks** for side effects like data fetching. To optimize performance, you have to strategically combine hooks and implement **Context API** for managing global state, this will allow seamless data access across components. As the app grows, they decide to integrate **Redux** for more complex state management, ensuring that components re-render efficiently and maintain synchronization with the global state.

Throughout the process, employ **Ref Hooks** to access DOM elements directly and improve performance.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.	Hooks are well implemented	Hooks are imported		
		States are defined		
		State variables are used		
		Stated updater are called		
2.	Performance optimization is well performed	useMemo hooks is used		
		use callback Hooks is used		
		memorization of components is performed		
3.	Global state is well managed	Context API is built		
		Context is provided		
		Context is consumed		
		Redux is installed		
		Redux is used		
4.	Ref Hooks are employed	Ref Hooks are imported		
		Ref Hooks are used		

Indicative content 1.5: Implementation of Events handling

Duration: 10 hrs

Theoretical Activity 1.5.1: Description of React Events

Notes to the trainer:

- While delivering this content small groups can be used to describe events
- Avail videos to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to answer the following questions:

i. What do you understand by Events?
ii. Explain different types of Events
iii. Describe Synthetic events
iv. Explain Event Bubbling
v. Explain the terms Debouncing and Throttling

Step 2: Ask trainees to write their findings on paper

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees

Step 6: Ask the trainees to read key readings 1.4.1 in their manual

Points to Remember

- **Events**

These are actions or occurrences that happen in a system, like user interactions.

- **Types of Events include** User Interface Events, Focus and Blur Events, Form Events, Mouse Events, Keyboard Events, Synthetic Events
- **Event Bubbling** is the process where an event starts at the deepest target element and propagates upward through the parent elements, allowing multiple event listeners along the path to handle the event

- **Debouncing and throttling** are techniques used in React (or JavaScript in general) to optimize performance by controlling how frequently a function is executed in response to rapid events, such as typing, scrolling, or resizing. Both are particularly useful when handling events that fire frequently, like onScroll, onResize, or onChange for inputs.

Practical Activity 1.5.2: Using Controlled Components

Notes to the trainer

- This activity has to take place in computer lab where trainees have to use controlled components
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity by asking trainees to go to the computer lab to use controlled components

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to use controlled components. While demonstrating explain the steps to be followed

Step 4: Ask trainees to use controlled components and monitor the activity

Step 5: Verify whether controlled components are clearly used

Step 6: Ask trainee to read key readings 1.5.2

Step 7: Ask trainee to perform the task provided in the application of learning 1.5

Points to Remember

- **Steps to use controlled components**
  - ✓ Create a functional or class component where you will manage your form state.
  - ✓ Use the useState hook (for functional components) or this.state (for class components) to create state variables that will hold the values of your form inputs.

- ✓ Add input fields to your component and set their value prop to the corresponding state variable. This binds the input field's value to the state.
- ✓ Implement an onChange event handler for each input element that updates the state. This ensures the component re-renders with the new value.
- ✓ Create a function to handle form submission. This function can prevent the default form submission behaviour and access the state values.
- ✓ Use the state values in your component logic, such as for form validation, submission, or displaying the values elsewhere.
- ✓ You can repeat these steps for other input types like checkboxes, radio buttons, or selects.

Practical Activity 1.5.3: Passing Arguments to Event Handlers

Notes to the trainer

- This activity has to take place in computer lab where trainees have to Pass Arguments to Event Handlers. The following tasks will be performed individually
    1. Arrow Function (in JSX)
    2. Bind Method
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.5.2

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to Pass Arguments to Event Handlers in the ReactJS application provided using:

1. Arrow Function (in JSX)
2. Bind Method

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to Pass Arguments to Event Handlers. While demonstrating explain the steps to be followed

Step 4: Ask trainees to Pass Arguments to Event Handlers and monitor the activity

Step 5: Verify whether Passing Arguments to Event Handlers is correctly performed

Step 6: Ask trainee to read key readings 1.5.3

Step 7: Ask trainee to perform the task provided in the application of learning 1.5

Points to Remember

- **Steps to pass arguments to event handlers**
  - ✓ Create a functional or class component where you will manage your form state.
  - ✓ Use the useState hook (for functional components) or this.state (for class components) to create state variables that will hold the values of your form inputs.
  - ✓ Add input fields to your component and set their value prop to the corresponding state variable. This binds the input field's value to the state.
  - ✓ Implement an onChange event handler for each input element that updates the state. This ensures the component re-renders with the new value.
  - ✓ Create a function to handle form submission. This function can prevent the default form submission behaviour and access the state values.
  - ✓ Use the state values in your component logic, such as for form validation, submission, or displaying the values elsewhere.
  - ✓ You can repeat these steps for other input types like checkboxes, radio buttons, or selects.

Practical Activity 1.5.4: Using Custom Hooks for Event Listeners

Notes to the trainer

- This activity has to take place in computer lab where trainees have to use custom Hooks for Event Listeners
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.5.2

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to use custom Hook s for Event Listeners

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to use custom Hooks for Event Listeners. While demonstrating exp lain the steps to be followed

Step 4: Ask trainees to use custom Hooks for Event Listeners and monitor the activity

Step 5: Verify whether custom Hooks for Event Listeners are correctly used

Step 6: Ask trainee to read key readings 1.5.4
Step 7: Ask trainee to perform the task provided in the application of learning 1.5

Points to Remember

- **Steps to use Custom Hooks for Event Listeners**
  - ✓ Create a custom hook, useEventListener, to handle event listener logic. This hook will add the event listener when the component mounts and remove it when the component unmounts.
  - ✓ Once the hook is created, you can use it in any component where you need to listen to an event.
  - ✓ You can pass any DOM element to the hook, not just window. For example, you can listen for a click event on a specific button.
  - ✓ If the event listener is attached to dynamic or conditionally rendered elements, make sure the hook handles potential null values by checking if the element exists before adding the event listener.

Practical Activity 1.5.5: Handling Events on Dynamic Lists

Notes to the trainer

- This activity has to take place in computer lab where trainees have to Handle Events on Dynamic Lists
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.5.2

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to use custom Hook s for Event Listeners
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to use custom Hooks for Event Listeners. While demonstrating exp lain the steps to be followed
Step 4: Ask trainees to use custom Hooks for Event Listeners and monitor the activity
Step 5: Verify whether custom Hooks for Event Listeners are correctly used
Step 6: Ask trainee to read key readings 1.5.5

Step 7: Ask trainee to perform the task provided in the application of learning 1.5

Points to Remember

- **Steps to use Custom Hooks for Event Listeners**
  - ✓ Set Up State for the List. To handle dynamic lists, you'll need to store the list's data in a state variable
  - ✓ Attach Event Handlers to List Items
  - ✓ Handle Dynamic Updates (Add/Delete Items)
  - ✓ Pass Data to Event Handlers
  - ✓ Conditional Rendering for Dynamic Behaviour
  - ✓ Update the State on Event Trigger
  - ✓ Ensure you use proper key props in list rendering for optimal rendering performance.
  - ✓ Use useCallback or memo for functions if necessary to prevent unnecessary re-renders.

Application of learning 1.5.

ABC company is a company located in Niboye- Sector, Kicukiro District in Kigali City, it has many employs who are assigned daily task, assume you have been hired as their web developer to develop them a Todo List app in React that will help the employee manager to add, remove, and mark tasks as complete. Implement event handling for user interactions, such as clicking buttons or pressing keys. Use debouncing for search input to improve performance and throttling for scrolling events to manage resource usage. Utilize controlled components to maintain the state of inputs. Pass arguments to event handlers using arrow functions and the bind method. Create custom hooks for managing event listeners, and ensure event handling works seamlessly on a dynamically generated list of tasks

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.	Controlled components are well used	Components are created		
		useState Hooks are used		
		onChange Event Handler is used		

		Function to handle events is created		
2.	Custom Hooks for Event Listeners clearly are used	useEventListener Hook is used		
		Hooks parameters are passed		
3.	Events are well handled on dynamic Lists	State is setup for the list		
		Event Handlers are attached to the list		
		Dynamic updates are handled		
		Data is passed to event handlers		
		Conditional rendering is performed		
		States are updated on event trigger		
		Proper key props are used		
		useCallBack Hooks is used		

Indicative content 1.6: Implementation of API integration

Duration: 5 hrs

Theoretical Activity 1.6.1: Description of API Integration

Notes to the trainer:

- While delivering this content small groups can be used to:
  1. Define API
  2. Give importance of API Integration
  3. List common methods used to fetch data from APIs
- Avail videos to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to answer the following questions:

1. What is an API?
2. What are importance of API Integration?
3. What are the common methods used for fetching data from APIs in React.js?

Step 2: Ask trainees to write their findings on paper

Step 3: Ask trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees

Step 6: Ask the trainees to read key readings 1.6.1 in their manual

Points to Remember

- APIs in React.js enable communication between the front-end and external services, fetching or sending data like user info or weather updates.
- APIs enhance dynamic data handling, scalability, separation of concerns, interactivity, and reusability.
- Common methods for fetching API data include Fetch API, Axios, async/await for asynchronous operations, and React Query for advanced server-side state management with caching and syncing features.

- APIs play a crucial role in making React apps modular, maintainable, and highly interactive without page reloads.

Practical Activity 1.6.2: Installing dependencies (Axios)

Notes to the trainer

- This activity has to take place in computer lab where trainees have to install dependencies, it can be done individually
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to install dependencies

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to install dependencies. While demonstrating explain the steps to be followed

Step 4: Ask trainees to install dependencies and monitor the activity

Step 5: Verify whether dependencies are clearly installed

Step 6: Ask trainee to read key readings 1.6.2

Step 7: Ask trainee to perform the task provided in the application of learning 1.6

Points to Remember

- **Steps to install dependencies**
  - ✓ Ensure Node.js and npm are Installed
  - ✓ Create a React Project
  - ✓ Install Project Dependencies
  - ✓ Add New Dependencies
  - ✓ Run the Project

Practical Activity 1.6.3: Defining and Grouping API Calls

Notes to the trainer

- This activity has to take place in computer lab where trainees have to Define and Group API Calls.
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to define and group API calls in the ReactJS application provided
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to define and group API calls. While demonstrating explain the ste ps to be followed
Step 4: Ask trainees to define and group API calls and monitor the activity
Step 5: Verify whether API calls are clearly grouped and defined
Step 6: Ask trainee to read key readings 1.6.3
Step 7: Ask trainee to perform the task provided in the application of learning 1.6

Points to Remember

- **Steps for Defining and Grouping API Calls**
  - ✓ Install Axios (or Fetch API) for API Calls
  - ✓ Create a Separate API Utility File
  - ✓ Define API Base URL and Common Configuration
  - ✓ Define API Endpoints
  - ✓ Use API Calls in React Components
  - ✓ Perform Error Handling and Response Interception
  - ✓ Handle Caching and Optimizations

Practical Activity 1.6.4: Handling Data Fetching and Responses

Notes to the trainer

- This activity has to take place in computer lab where trainees have to perform Handling Data Fetching and Responses
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to perform Handling Data Fetching and Responses in the ReactJS application provided

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to perform Handling Data Fetching and Responses. While demonst rating explain the steps to be followed

Step 4: Ask trainees to perform Handling Data Fetching and Responses and monitor the activ ity

Step 5: Verify whether Handling Data Fetching and Responses are clearly performed

Step 6: Ask trainee to read key readings 1.6.4

Step 7: Ask trainee to perform the task provided in the application of learning 1.6

Points to Remember

- **Steps for Handling Data Fetching and Responses**
  - ✓ Set Up State to Manage Data and Loading Status
  - ✓ Use useEffect to Fetch Data When Component Loads
  - ✓ Render Based on State (Loading, Error, or Data)
  - ✓ Handle POST or Other Methods

Practical Activity 1.6.5: Error Handling

Notes to the trainer

- This activity has to take place in computer lab where trainees have to perform Error Handling
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to got to the computer lab to perform Error Handling in the ReactJS application provided
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to perform Error Handling. While demonstrating explain the steps to be followed
Step 4: Ask trainees to perform Error Handling and monitor the activity
Step 5: Verify whether Error Handling is clearly performed
Step 6: Ask trainee to read key readings 1.6.5
Step 7: Ask trainee to perform the task provided in the application of learning 1.6

Points to Remember

- **Steps for performing Error Handling**
  - ✓ Choose an API Request Method (Fetch or Axios)
  - ✓ Set Up State for Loading, Error, and Data
  - ✓ Make the API Call with Error Handling
  - ✓ Handle Errors Based on Status Code
  - ✓ Show Loading, Error, and Data States in the UI

Practical Activity 1.6.6: Handling Asynchronous and Concurrency

Notes to the trainer

- This activity has to take place in computer lab where trainees have to perform Asynchronous Handling and Concurrency
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to perform Asynchronous Handling and Concurrency in the ReactJS application provided
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to perform Asynchronous Handling and Concurrency. While demonstrating explain the steps to be followed
Step 4: Ask trainees to perform Asynchronous Handling and Concurrency and monitor the activity
Step 5: Verify whether Asynchronous Handling and Concurrency is clearly performed
Step 6: Ask trainee to read key readings 1.6.6
Step 7: Ask trainee to perform the task provided in the application of learning 1.6

Points to Remember

- **Steps for to performing Asynchronous Handling and Concurrency**
  - ✓ Select the best asynchronous feature (promises) to handle operation
  - ✓ In a ReactJS application, Create or Fetch Asynchronous Data (API Call
  - ✓ Handle Async Operations in useEffect Hook
  - ✓ Understand the Flow
  - ✓ use. then () Instead of async/await
  - ✓ Fetch Data on User Interaction

Practical Activity 1.6.7: Performing API Security and testing

Notes to the trainer

- This activity has to take place in computer lab where trainees have to perform API Security and testing
- Avail videos to be used as didactic material
- Avail sample ReactJS application developed in Practical Activity 1.2.1

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to perform API Secu rity and testing in the ReactJS application provided
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to perform API Security and testing. While demonstrating explain t he steps to be followed
Step 4: Ask trainees to perform API Security and testing and monitor the activity
Step 5: Verify whether API Security and testing is clearly performed
Step 6: Ask trainee to read key readings 1.6.7
Step 7: Ask trainee to perform the task provided in the application of learning 1.6

Points to Remember

- **Steps for performing API Security and testing**
  - ✓ Use HTTPS for Secure Communication
  - ✓ Authentication and Authorization
  - ✓ Input Validation and Sanitization
  - ✓ Rate Limiting and Throttling
  - ✓ CORS (Cross-Origin Resource Sharing)
  - ✓ CSRF Protection
  - ✓ Use Security Headers
  - ✓ Encryption of Sensitive Data
  - ✓ Logging and Monitoring
  - ✓ Security Testing
  - ✓ Conduct regular penetration testing of your API to identify vulnerabilities.

Application of learning 1.6.

Rwanda Metrology Centre is a centre that provides real-time weather forecast. It has hired you as a software developer. The centre needs a software developer to work on integrating a weather API. You are requested to begin by planning the implementation, define the API's purpose and install dependencies like Axios. Organize their API calls by grouping related requests, implementing error handling, and managing asynchronous operations to optimize data fetching. Finally, ensure that the application is secure by testing for vulnerabilities and confirming that the API integration functions correctly, allow users to fetch real-time weather data while maintaining a seamless user experience.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.	Dependences are well installed	NodeJS ids installed		
		React Project is created		
		Dependencies for API Integration are installed		
2.	API calls are well defined and grouped	Axios are installed		
		API utility file is created		
		API base URL is defined		
		Endpoints are defined		
		API calls are used		
		Error handling is performed		
3.	Data fetching and Responses are well handled	State is setup		
		useEffect Hook is used		
		Rendering is performed		
		HTTP Methods are handled		

4.	Asynchronous and Concurrency are well handled	Best Asynchronous Feature is selected		
		Asynchronous data is fetched		
		Async operation is handled		
		Data is fetched on user interaction		
5.	API Security and testing are well performed	HTTPs are used		
		Authentication and Authorization are performed		
		Validation and Sanitization are performed		
		Rate Limiting and Throttling are performed		
		CORS are implemented		
		Security Headers are used		
		Encryption is implemented		
		Logging and monitoring are performed		
		Security testing is performed		
		Penetration testing is implemented		

Learning Outcome 1 End Assessment

Written assessment

Question 1. Choose the correct alternative by circling the letter that corresponds to the answer

1. What is JSX in React?

   a) A JavaScript framework

   b) A JavaScript XML syntax extension

   c) A state management library

   d) A CSS library

2. Which of the following is a lifecycle method in React?

   a) render

   b) componentDidMount

   c) useEffect

   d) setState

3. Which hook is used to manage state in functional components?

   a) useState

   b) useEffect

   c) useContext

   d) useReducer

4. Which of the following is a valid way to pass props in React?

   a) Using state

   b) Through the constructor

   c) As an attribute in JSX

   d) Through the render method

5. What does the Virtual DOM do?

   a) Stores component states

   b) Provides routing in React

   c) Minimizes the number of DOM manipulations

   d) Handles event listeners

6. What is React Router primarily used for?

   a) Managing component state

   b) Navigation between different views

   c) Handling forms

   d) Making API calls

7. Which method unmounts a React component?

   a) componentWillUnmount

   b) componentDidUpdate

c) render

d) shouldComponentUpdate

8. Which React hook would you use to run side effects?

a) useState

b) useEffect

c) useReducer

d) useMemo

9. What does Redux help manage in a React application?

a) UI components

b) Global state

c) Event handling

d) DOM updates

10. Which event type in React is responsible for handling input changes?

a) onClick

b) onChange

c) onSubmit

d) onMouseOver

11. What is a synthetic event in React?

a) A browser native event

b) A wrapper around browsernative events

c) A custom event created by the developer

d) A debugging tool

12. In React, which function is used to handle API calls?

a) useFetch

b) fetch

c) Axios

d) getRequest

13. The main purpose of the Context API in React is:

a) Data fetching

b) State management across components

c) Handling user inputs

d) Event handling

14. What method is used to prevent event bubbling in React?

a) stopBubbling

b) preventPropagation

c) stopPropagation

d) stopImmediatePropagation

15. How is a controlled component defined in React?

a) A component that controls event propagation

b) A component whose state is managed by itself

c) A component whose form data is handled by React state

d) A component that cannot rerender

16. What does the useEffect hook do?

a) Handles side effects in function components

b) Updates component state

c) Allows conditional rendering

d) Enables lifecycle methods in class components

17. What does the Redux store represent?

a) A place where React components are defined

b) The global state for a React application

c) A collection of React hooks

d) A virtual DOM replica

18. What is the main use of MobX in React applications?

a) Manage local state

b) Manage global state

c) API integrations

d) Routing

19. What is the purpose of componentDidUpdate in React?

a) To update the state

b) To perform actions after a component rerenders

c) To initialize a component

d) To unmount a component

20. How can you perform routing in React?

a) Using React hooks

b) Using Redux

c) Using React Router

d) Using event listeners

Question 2. From the questions below, identify the validity of the statement by indicating if the statement is TRUE or FALSE

1. JSX allows embedding HTML within JavaScript code
2. The useState hook is used for side effects in React components
3. The Virtual DOM helps improve the performance of React applications.
4. Props are mutable in React components.
5. The useEffect hook is equivalent to all three lifecycle methods in class components.
6. Redux is mainly used to manage local component state.
7. React Router helps in managing state transitions.
8. The Context API is a way to avoid passing props through multiple components
9. A synthetic event in React is a cross browser wrapper around native events

10. The componentWillUnmount method is used to clean up after a component is removed
11. React components re-render when their props or state changes
12. In JSX, class names are assigned using the class attribute
13. Lifecycle methods are only available in class components
14. The bind () method is used to bind event handlers in functional components
15. MobX is a state management tool similar to Redux
16. A controlled component uses internal state to manage form data
17. Context API and Redux both solve global state management problems
18. Nested routing is supported by React Router
19. Events in React follow the same event delegation as vanilla JavaScript
20. React Developer Tools is used to debug and inspect components in a browser

Question 3. From the questions below, complete the following sentences by inputting the best phrase or group of phrases from a list of phrases provided below.

navigation between different components or pages, JavaScript XML, state, JavaScript function that returns JSX, a component has been rendered to the DOM, side effects, hooks, class components, browser's native events, in memory representation, global state, passing data through the component tree without props, parent components, child components, event.stopPropagation(), state management libraries, clean up resources before a component is removed from the DOM, class components, memorize callback functions, useParams, the useEffect hook or async/await

1. JSX stands for ______________.
2. React Router is used for ______________.
3. The useState hook is used to manage ______________ in functional components.
4. The componentDidMount lifecycle method is called after ______________.
5. A functional component in React is a ______________.
6. React allows developers to manage state using both ______________ and ______________.
7. Synthetic events are a cross-browser wrapper around ______________.
8. A Virtual DOM in React is an ______________ of the real DOM.
9. The useEffect hook is primarily used for handling ______________.
10. Redux helps in managing ______________ in large React applications.
11. The React Context API is used for ______________.
12. Props in React are passed from ______________ to ______________.
13. To prevent an event from bubbling up the DOM, you would use ______________.
14. MobX and Redux are both ______________.
15. The componentWillUnmount lifecycle method is used to ______________.

16. The bind() method in React is used to bind event handlers to ______________.
17. The useCallback hook in React is used to ______________.
18. A controlled component's form data is handled by ______________.
19. URL parameters in React Router are handled using the ______________ method.
20. To handle asynchronous data fetching in React, you can use ______________.

Answers

Question 1

1. b) A JavaScript XML syntax extension
2. b) componentDidMount
3. a) useState
4. c) As an attribute in JSX
5. c) Minimizes the number of DOM manipulations
6. b) Navigation between different views
7. a) componentWillUnmount
8. b) useEffect
9. b) Global state
10. b) onChange
11. b) A wrapper around browser native events
12. b) fetch
13. b) State management across components
14. c) stopPropagation
15. c) A component whose form data is handled by React state
16. a) Handles side effects in function components
17. b) The global state for a React application
18. b) Manage global state
19. b) To perform actions after a component re-renders
20. c) Using React Router

Question 2.

1. TRUE
2. FALSE (useState is used to manage state, not side effects)
3. TRUE
4. FALSE (Props are immutable in React components)
5. TRUE
6. FALSE (Redux manages global state, not local component state)
7. FALSE (React Router is used for navigation, not state transitions)
8. TRUE
9. TRUE

10. TRUE
11. TRUE
12. FALSE (In JSX, class names are assigned using className attribute)
13. TRUE
14. FALSE (bind() is used in class components, not functional components)
15. TRUE
16. FALSE (Controlled components use React state to manage form data)
17. TRUE
18. TRUE
19. TRUE
20. TRUE

Question 3.

1. JSX stands for JavaScript XML.
2. React Router is used for navigation between different components or pages.
3. The useState hook is used to manage state in functional components.
4. The componentDidMount lifecycle method is called after a component has been rendered to the DOM.
5. A functional component in React is a JavaScript function that returns JSX.
6. React allows developers to manage state using both hooks and class components.
7. Synthetic events are a cross-browser wrapper around browser's native events.
8. A Virtual DOM in React is an in-memory representation of the real DOM.
9. The useEffect hook is primarily used for handling side effects.
10. Redux helps in managing global state in large React applications.
11. The React Context API is used for passing data through the component tree without props.
12. Props in React are passed from parent components to child components.
13. To prevent an event from bubbling up the DOM, you would use event.stopPropagation().
14. MobX and Redux are both state management libraries.
15. The componentWillUnmount lifecycle method is used to clean up resources before a component is removed from the DOM.
16. The bind() method in React is used to bind event handlers to class components.
17. The useCallback hook in React is used to memorize callback functions.
18. A controlled component's form data is handled by state.
19. URL parameters in React Router are handled using the useParams method.
20. To handle asynchronous data fetching in React, you can use the useEffect hook or async/await.

Practical assessment

ABC Company is an e-commerce company that is located in Rwamagana District- Eastern province – Rwanda. It has an online platform specializing in selling electronics. The platform is expanding its web application with a user-friendly React.js frontend to improve the shopping experience. They aim to enhance the performance, user interface, and functionality of the website, making it more dynamic, responsive, and interactive. You, as the React.js developer at ABC Company, have been assigned the task of building a Product Listing and Search Page that includes a search bar, filters, product listing, and product details. The page should provide a smooth user experience by incorporating various features using React.js

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.	Preparation of React.js environment	NodeJS and NPM are installed		
		React Application is Created		
		React Project folder is explored Additional react tools are installed		
2.	Applying React basics	Components are created		
		JSX is used		
		Props are implemented		
3.	Applying UI navigation	Route is installed		
		Routes are configured		
		Basic React navigation is implemented		
		404 pages are handled		
4.	Applying React hooks	Hooks are identified		
		Hooks are used		
		Hooks are combined		
		Performance is optimized		
		Complex state logic is handled		

		Global state is managed		
5.	Implementation of Events handling	Events are implemented		
		Controlled components are used		
		Arguments are passed to event handlers		
		Custom Hooks for event Listeners are used		
		Events on dynamic lists are handled		
6.	Implementation of API integration	API are identified		
		Dependencies are installed		
		API calls are organized		
		API security is performed		
		API testing is performed		

END

Further information to the trainer

Banks, A., & Porcello, E. (2020). Learning React: A hands-on guide to building web applications using React and Redux (2nd ed.). O'Reilly Media.

Callahan, B. (2021). React Explained: Your step-by-step guide to React (1st ed.). CodeBrauer.

Cooper, C. (2021). React Design Patterns and Best Practices: Design, build, and deliver production-ready web applications using industry-standard practices (2nd ed.). Packt Publishing.

Grider, S. (2020). The complete React developer course (4th ed.). Udemy.

Haris, B. (2021). React Projects: Build advanced cross-platform projects with React and React Native to become a professional developer (1st ed.). Packt Publishing.

Hinnant, R. (2021). React by example: Building modern web applications using React (1st ed.). Manning Publications.

Lerner, R. (2022). Professional React: State management, performance optimization, and testing (1st ed.). Wiley.

Mackenzie, B., & Dodds, K. (2020). Epic React: Create interactive user interfaces with confidence (1st ed.). Kent C. Dodds Tech LLC.

Moore, S. (2020). React Cookbook: Recipes for mastering modern web development (2nd ed.). Packt Publishing.

Park, S., & Picinich, N. (2021). Fullstack React: The complete guide to React and friends (2nd ed.). Fullstack.io.

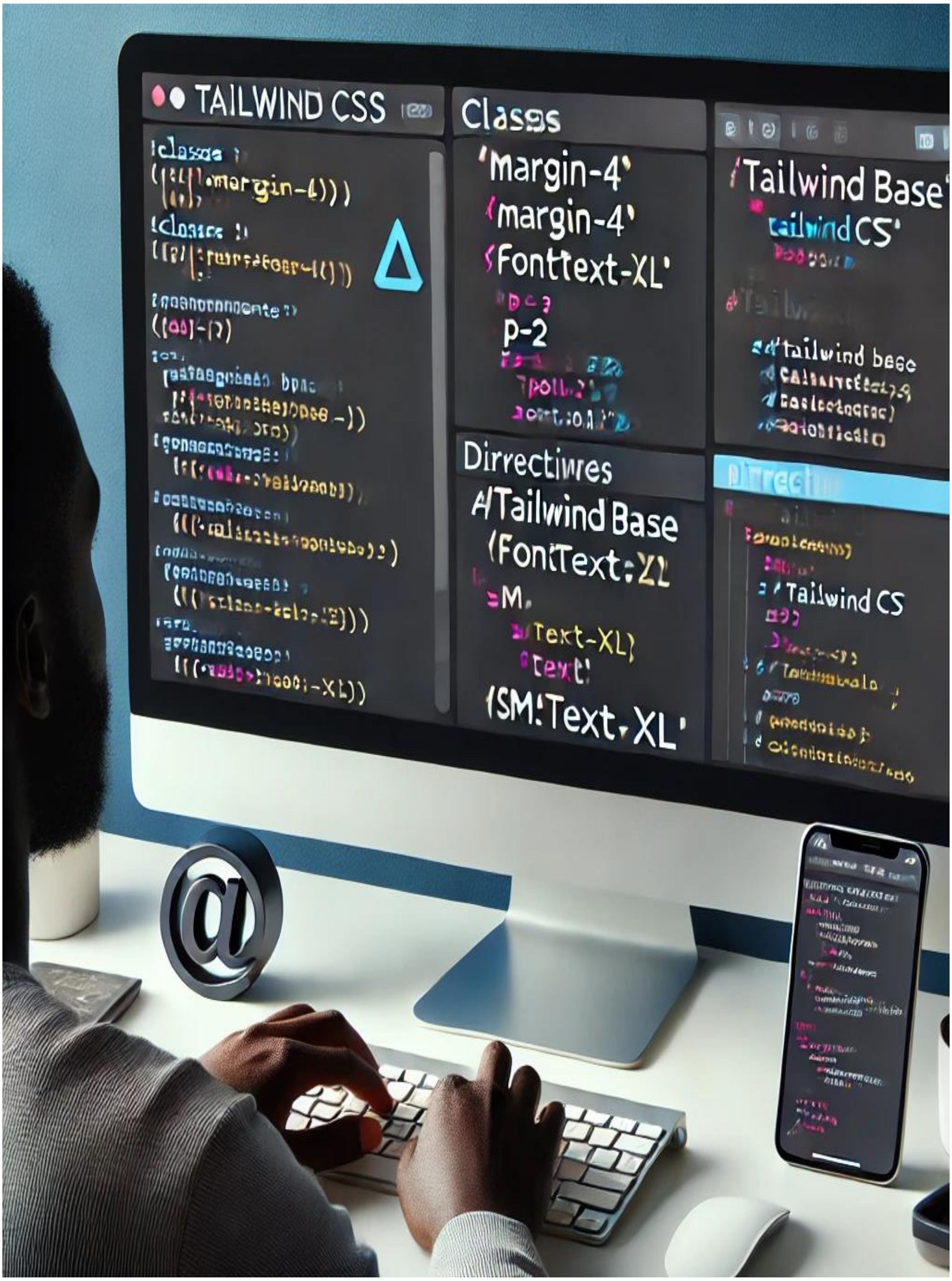
Sharma, A. (2020). React Hooks in Action: With Suspense, Concurrent mode, and more (1st ed.). Manning Publications.

Taube, C. (2021). React Router: Building Single-Page Applications with React (1st ed.). O'Reilly Media.

Wieruch, R. (2021). The Road to React: Your journey to master React.js in JavaScript (4th ed.). Leanpub.

Williams, A. (2021). React and React Native: A complete hands-on guide to modern web and mobile development (3rd ed.). Packt Publishing.

Zolotarev, A. (2020). React Quickly: Painless web apps with React, JSX, Redux, and GraphQL (2nd ed.). Manning Publications.


TAILWIND CSS
Classes
'Margin-4'
'margin-4'
'Fonttext-XL'
p-2
Directiwres
Tailwind Base
FontText
Text-XL
SM:Text-XL'
'Tailwind Base'
tailwind CS'
tailwind base
Tailwind CS

Indicative contents

2.1 Applying Tailwind Utility Classes
2.2 Applying Responsive Design Principles
2.3 Customization of Tailwind Styles

Key Competencies for Learning Outcome 2: Apply Tailwind CSS Framework

Knowledge	**Skills**	**Attitudes**
● Description of animation, transitions, flexbox, grid, states and classes. ● Description of media queries and breakpoints ● Description of typography and readability ● Identification of styles ● Description of responsive design principles	● Applying Tailwind CSS in React.JS ● Applying responsive design principles ● Customizing styles in Tailwind	● Being creative in developing up-to-date Tailwind CSS apps ● Being problem-solving oriented during development with Tailwind CSS ● Being updated on latest Tailwind CSS versions ● Being collaborative while developing of application with Tailwind CSS

Duration: 10 hrs

Learning outcome 2 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe clearly animations and transitions classes used in Tailwind CSS.
2. Describe clearly flexbox and grid classes used in Tailwind CSS
3. Describe properly responsive design principles
4. Identify properly the styles used in Tailwind CSS.
5. Integrate correctly Tailwind CSS in React.JS application.
6. Customize properly Tailwind styles in React.JS application.

Resources

Equipment	**Tools**	**Materials**
● Computer	● VS Code ● Browser	● Internet ● Electricity

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have prepared computer lab with Internet connectivity
- Have visual studio code installed on all computers to be used.
- Have videos to be used as didactic material.
- Have sample React.JS project developed to be used for styling.

Indicative content 2.1: Applying Tailwind Utility Classes

Duration: 4 hrs

Practical Activity 2.1.1: Integrating Tailwind CSS with React.JS

Notes to the trainer

- The trainer asks trainees to integrate the Tailwind CSS in React.JS
- The use of videos as didactic material is required.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to integrate Tailwind CSS with ReactJS.

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to integrate Tailwind CSS with ReactJS. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to integrate Tailwind CSS with ReactJS and monitor the activity.

Step 5: Verify whether Tailwind CSS and ReactJs are properly integrated.

Step 6: Ask trainees to read the key readings 2.1.1 in their manual.

Step 7: Ask trainees to perform the task provided in the application of learning 2.1.

Points to Remember

- Steps to integrate Tailwind CSS in ReactJS application
  - ✓ Install Tailwind CSS via npm.
  - ✓ Initialize Tailwind config files using npx tailwindcss init .
  - ✓ Configure tailwind.config.js to scan your React components.
  - ✓ Add Tailwind directives (@tailwind base; @tailwind components; @tailwind utilities;) to your CSS file.
  - ✓ Import the CSS into src/index.js.
  - ✓ Run your React app, and start using Tailwind classes in your components with npm start

Theoretical Activity 2.1.2: Description of utility-first fundamentals and states

Notes to the trainer:

- While delivering this content small groups can be used to describe the utility-first fundamentals and states.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step1: Introduce the activity and ask trainees to answer the following questions:

i. Define utility-first fundamentals in Tailwind CSS.

ii. Give at least 5 examples of utility-first classes that are used in Tailwind CSS.

iii. What is the role of using the utility-first classes used in Tailwind CSS?

iv. What are the features of utility-first fundamentals in Tailwind CSS?

v. Explain the use of:

  a) text-center
  b) p-4
  c) bg-blue-500

vi. Outline three (3) variant states that are used in Tailwind CSS.

Step 2: Ask trainees to write their findings on paper or flipchart.
Step 3: Engage trainees to present their findings
Step 4: Provide an expert view and clarify the ideas.
Step 5: Address any question or concern raised by the trainees.
Step 6: Ask the trainees to read the key readings 2.1.2 in their manual.

Points to Remember

- **Utility-first fundamentals** in Tailwind CSS is an approach where you build designs using pre-defined utility classes that are designed to be small, single-purpose, and composable.
- **Single-purpose Classes:** Each utility class serves one specific function.
- **Composability classes**: They combine multiple utility classes to achieve complex designs.

- **Responsiveness Built-in classes**: Tailwind makes it easy to create responsive designs by providing utilities for different screen sizes.
- **Hover state** allows you to apply styles to an element when a user hovers over it with their cursor. Tailwind provides a hover: prefix.
- **Focus state** is the style applied to an element when it receives focus with focus: prefix.
- **Active state** is the style applied to an element when it is being interacted with, typically by a user clicking or pressing it with active: prefix.

Practical Activity 2.1.3: Applying Utility-First Fundamentals and states

Notes to the trainer

- The trainer asks trainees to apply the utility-first fundamentals and states as the features of Tailwind CSS in React application.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to apply the utility-first fundamentals and states.

Step 2: Provide the clear work instructions.

Step 3: Demonstrate how to apply the utility-first fundamentals and states. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to apply the utility-first fundamentals and monitor the activity.

Step 5: Verify whether utility-first fundamentals are well applied.

Step 6: Ask trainees to read the key readings 2.1.3 in their manual.

Step 7: Ask trainees to perform the task provided in the application of learning 2.1

Points to Remember

- To apply the utility first fundamentals and states you set the followings:
  - ✓ Browse to the folder where your project is created
  - ✓ Open command prompt

- ✓ Type the command to open the project in your text editor
- ✓ Access the React the file you want to style
- ✓ Use the utility-first fundamentals
- ✓ Apply the states

Theoretical Activity 2.1.4: Description of animation and transitions

Notes to the trainer:

- While delivering this content small groups can be used to describe animation and transitions.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. What do you understand by the term?

a) Animation

b) Transition

ii. Outline four utility classes that are used to animate object with Tailwind CSS.

iii. Enumerate the components of transitions in Tailwind CSS.

iv. Give the main possible classes of:

a) Animation

b) Transition delay

c) Transition timing function

d) Transition duration

e) Transition property

Step 2: Ask trainees to write their findings on paper or flipchart

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees.

Step 6: Ask the trainees to read key readings 2.1.4 in their manual.

Points to Remember

- **Animation** is the ability to apply motion effects to elements, either using pre-defined utility classes or custom animations.

- **Animation classes**: animate-spin, animate-ping, animate-pulse, and animate-bounce.
- **Transition utility**: This class enables transitions on an element. By default, it applies a transition to all properties that change.
- **Transition-property**: You can specify which properties should transition using utilities like transition-colors, transition-opacity, transition-transform, and more.
- **Transition duration:** This utility controls how long the transition lasts. It is defined in milliseconds, and Tailwind provides utilities such as duration-150, duration-300, duration-500, etc.
- **Transition time** function controls the transition's timing function, which dictates how the speed of the transition progresses over time. Tailwind provides utilities like ease-linear, ease-in, ease-out, and ease-in-out.
- **Transition delay** adds a delay before the transition starts.

Practical Activity 2.1.5: Applying animation and transitions

Notes to the trainer

- The trainer asks trainees to apply the animation and transitions.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to apply the animation and transitions.
Step 2: Provide the clear work instructions
Step 3: Demonstrate how to apply the animation and transitions. While demonstrating, explain the steps to be followed.
Step 4: Ask trainees to apply animation and transitions and monitor the activity
Step 5: Verify whether animation and transitions are well applied
Step 6: Ask trainees to read the key readings 2.1.5 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 2.1

Points to Remember

- To apply animation and transitions you set the followings:
  - ✓ Browse to the folder where your project is created
  - ✓ Open command prompt
  - ✓ Type the command to open the project in your text editor
  - ✓ Access the react component
  - ✓ Add animation
  - ✓ Add transitions

Theoretical Activity 2.1.6: Description of flexbox and grid

Notes to the trainer:

- While delivering this content small groups can be used to describe flexbox and grid.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. What do you understand by the term?

a) Flexbox

b) Grid

ii. What are the possible classes of flex direction?

iii. What is flex-wrap-reverse?

iv. Differentiate the following:

a) grow class from shrink class

b) col-span from row-span

v. Give the main possible values of *:

a) grid-flow-*

b) auto-cols-*

c) auto-rows-*

d) justify-*

e) justify-items-*

f) content-*

g) place-content-*

Step 2: Ask trainees to write their findings on paper or flipchart

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees.

Step 6: Ask the trainees to read key readings 2.1.6 in their manual.

Points to Remember

- **Flexbox** simplifies the process of creating flexible and responsive layouts.
- **Flex Container**: A container element that uses Flexbox to lay out its children. You can create a flex container using the **flex** class.
- **Flex Direction**: Controls the direction in which flex items are placed within a flex container. Use **flex-row** for horizontal layout (default) and **flex-col** for vertical layout.
- Flex Wrap: Defines whether flex items should wrap onto multiple lines or stay on a single line. Use **flex-wrap** to allow wrapping and **flex-nowrap** to prevent it.
- **Justify Content**: Aligns flex items along the main axis of the container. Use classes like **justify-start, justify-center, justify-end, justify-between, justify-around, and justify-evenly.**
- **Align Items:** Aligns flex items along the cross axis (perpendicular to the main axis). Use classes like **items-start, items-center, items-end, items-baseline,** and **items-stretch**.
- **Align Self**: Allows individual flex items to override the align-items setting of their flex container. Use classes like **self-auto, self-start, self-center, self-end,** and **self-stretch**.
- Flex Grow and Shrink: Controls how flex items grow and shrink to fit the container. Use classes like **flex-grow** to allow an item to grow and **flex-shrink** to allow it to shrink.
- Flex Basis: Defines the initial size of a flex item before it grows or shrinks. Use classes like **basis-1/4**, **basis-1/2**, and **basis-full**.
- **Grid** is a layout system that allows you to create complex, responsive, and flexible grid-based designs using a set of utility classes.
- **Grid**: Applies CSS Grid display to an element, turning it into a grid container.
- **Grid-cols-***: Defines the number of columns in the grid.
- **Grid-rows-*:** Defines the number of rows in the grid.
- **Col-span-*:** Defines how many columns an item should span.
- **Row-span-*:** Defines how many rows an item should span.

Practical Activity 2.1.7: Utilizing Flexbox and Grid

Notes to the trainer

- The trainer asks trainees to utilize flexbox and grid.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to utilize the flexbox and grid.
Step 2: Provide the clear work instructions.
Step 3: Demonstrate how to utilize the flexbox and grid. While demonstrating, explain the steps to be followed.
Step 4: Ask trainees to utilize the flexbox and grid and monitor the activity.
Step 5: Verify whether flexbox and grid are well utilized.
Step 6: Ask trainees to read the key readings 2.1.7 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 2.1

Points to Remember

- To utilize the flexbox and grid you do the followings:
  - ✓ Browse to the folder where your project is created
  - ✓ Open command prompt
  - ✓ Type the command to open the project in your text editor
  - ✓ Access the react component
  - ✓ Create flexbox
  - ✓ Create the grid

Practical Activity 2.1.8: Applying reusable and custom styles

Notes to the trainer

- The trainer asks trainees to apply the reusable and custom styles.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to apply the reusable and custom styles.
Step 2: Provide the clear work instructions.
Step 3: Demonstrate how to apply the reusable and custom styles. While demonstrating, explain the steps to be followed.
Step 4: Ask trainees to apply reusable and custom styles and monitor the activity.
Step 5: Verify whether reusable and custom styles are properly applied
Step 6: Ask trainees to read the key readings 2.1.8 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 2.1

Points to Remember

- To apply the reusable and custom styles, you do the followings:
  - ✓ Browse to the folder where your project is created
  - ✓ Open command prompt
  - ✓ Type the command to open the project in your text editor
  - ✓ Access the react component
  - ✓ Apply the reusable styles
  - ✓ Apply the custom styles

Practical Activity 2.1.9: Using functions and directives

Notes to the trainer

- The trainer asks trainees to use functions and directives
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to use functions and directives.

Step 2: Provide the clear work instructions.

Step 3: Demonstrate how to use functions and directives. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to use functions and directives and monitor the activity.

Step 5: Verify whether functions and directives have been applied

Step 6: Ask trainees to read the key readings 2.1.9 in their manual

Step 7: Ask trainees to perform the task provided in the application of learning 2.1

Points to Remember

- To use functions and directives, you do the followings:
  - ✓ Navigate to your project folder
  - ✓ Open your project in a text editor
  - ✓ Use functions in Tailwind CSS
  - ✓ Use directives in Tailwind CSS

Application of learning 2.1.

ABC Company is a software development company located in Nyanza district, Southern Province – Rwanda. Its customer has requested it to develop a React.JS application that will be faster for the users. Assume you have been hired as its front-end developer, you are requested to build a responsive dashboard application using ReactJS and Tailwind CSS as part of the web application required**.** The application will feature a header, a sidebar, and a main content area displaying data in both grid and list formats. You are required to implement interactivity with hover and focus states, use animations, transitions, custom styles, functions, directives, and customize Tailwind with utility classes to improve the attractiveness of the dashboard.

Checklist

SN	Criteria	Indicators	Yes	No
1	Tailwind CSS is correctly integrated	Tailwind CSS is installed		
		Tailwind CSS is configured		
		Temperate path is configured		
		Tailwind is imported in the project		
2	Tailwind CSS is correctly applied	Class elements are defined in JSX		
		Focus state is applied		
		Hover state is applied		
		Animations are performed		
		Transitions are performed		
		Flexbox is applied		

Indicative content 2.2: Applying Responsive Design Principles

Duration: 3 hrs

Theoretical Activity 2.2.1: Description of responsive design principles

Notes to the trainer:

- While delivering this content small groups can be used to describe the responsive design principles.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Explain the following principles:

a) Mobile-first approach

b) Flexible grid layouts

c) Responsive images and media

d) Media queries and breakpoints

e) Typography and readability

f) Interactive elements

g) Testing and iteration

ii. What are the five default breakpoints in Tailwind CSS?

Step 2: Ask trainees to write their findings on paper or flipchart

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees.

Step 6: Ask the trainees to read the key readings 2.2.1 in their manual.

Points to Remember

- **Mobile-First Approach** is a strategy where you design your website or application starting with the smallest screen sizes (typically mobile devices) and progressively enhance the layout as the screen size increases.
- **Flexible Grid Layouts** ensure that the layout adapts to different screen sizes and resolutions.
- Responsive images and media by using **w-full, object-cover, object-contain** and **aspect-ratio** utility classes
- Five default breakpoints: **sm,md,lg,xl** and **2xl**.
- **Typography and readability** are crucial components of responsive design, ensuring text remains legible and aesthetically pleasing across various devices and screen sizes.
- **Interactive elements** in responsive design are crucial for providing an engaging and user-friendly experience across different devices and screen sizes.
- **Testing and Iteration** in responsive design, especially when using Tailwind CSS, involves evaluating and refining your web application to ensure it adapts well to various screen sizes and devices.

Practical Activity 2.2.2: Applying responsive design principles

Notes to the trainer

- The trainer asks trainees to apply the responsive design principles with Tailwind CSS in React application.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to apply the responsive design principles.

Step 2: Provide the clear work instructions.

Step 3: Demonstrate how to apply the responsive design principles. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to apply the responsive design principles and monitor the activity.
Step 5: Verify whether the responsive principles are correctly applied.
Step 6: Ask trainees to read the key readings 2.2.2 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 2.2

Points to Remember

- To apply the responsive design principles with Tailwind CSS you do the followings:

✓ Browse to the folder where your project is created
✓ Open command prompt
✓ Type the command to open the project in your text editor
✓ Access the react component
✓ Apply the following responsive design principles with Tailwind CSS:
  - Mobile-First Approach
  - Flexible Grid Layouts
  - Responsive Images and Media
  - Media Queries and Breakpoints
  - Typography and Readability
  - Interactive Elements
  - Testing and Iteration

Application of learning 2.2.

MXC is an online car selling company located in Niboye sector, Kicukiro District in Kigali city. It has an online car selling platform that is not seemless and interactive. It has fixed images that only clearly browse on Laptop devices. As a ReactJS developer, you are requested to upgrade their website pages using ReactJS and Tailwind CSS, focusing on responsive design principles. The new ReactJS application should showcase articles, images, and interactive elements, ensuring a seamless experience across devices.

Checklist:

SN	Criteria	Indicators	Yes	No
1.	React Application is well created	React project is created		
		Components are created		
		Web Contents are included		

2.	Responsive design principles are correctly applied	Mobile –first approach is used		
		Flexible grid layouts are used		
		Media queries and breakpoints are used		
		Typography and readability are used		
		Interactive elements are applied		
		Testing is performed		

Indicative content 2.3: Customization of Tailwind Styles

Duration: 3 hrs

Practical Activity 2.3.1: Customizing Tailwind Styles

Notes to the trainer

- The trainer asks trainees to customize the Tailwind styles.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to customize the Tailwind CSS styles.

Step 2: Provide the clear work instructions.

Step 3: Demonstrate how to customize the Tailwind styles. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to customize the Tailwind styles and monitor the activity.

Step 5: Verify whether styles are well customized.

Step 6: Ask trainees to read the key readings 2.3.1 in their manual

Step 7: Ask trainees to perform the task provided in the application of learning 2.3

Points to Remember

- To customize the tailwind styles with Tailwind CSS you do the followings:
  - ✓ Browse to the folder where your project is created
  - ✓ Open command prompt
  - ✓ Type the command to open the project in your text editor
  - ✓ Access the react component
  - ✓ Customize the default theme
  - ✓ Customize the variants
  - ✓ Customize fonts and typography
  - ✓ Customize colours

- ✓ Customize the functionality with plugins
- ✓ Customize the complex design with directives
- ✓ Use conditional styles with JavaScript

Application of learning 2.3.

Web Tech is a web application development company located in Rwamagana District, Eastern Province – Rwanda. It receives customers with different requests pertaining to websites and other applications. Assume you are a front-end developer for the company. You are working on a customer's ReactJS project where you need to create a custom, responsive landing page for a website. You are using Tailwind CSS for styling. Assume one of the clients with some ReactJS knowledge has specific design requirements that go beyond the default Tailwind configuration indicated below.

1. **Extend the default theme**:
   - o The primary color should be a custom shade of blue (#1A73E8), with secondary color as a custom shade of gray (#333333).
   - o Spacing units should include 72px (18rem).
2. **Add custom variants**:
   - o Add custom hover and focus variants for background color, so that the button background changes to green on hover and red on focus.
3. **Customize fonts and typography**:
   - o Use a custom font Gabriola with a fallback of serif.
4. **Customize colors**:
   - o Create a light and dark mode switch using custom color schemes. In light mode, the background should be white with dark text, and in dark mode, the background should be black with light text.
5. **Customize directives for complex designs**:
   - o Create a custom directive for a responsive 3-column grid layout, adjusting to a 1-column layout on mobile.

Use the above customer specifications to come-up with a required final landing page as required by the customer.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Tailwind styles are correctly customized	Primary color is used		
		Secondary color is used		

		Spacing units are used		
		Hover state is used		
		Focus state is used		
		Gabriola with serif is used		
		Colors are customized		
		Forms plugin is used		
		Custom directive is created		
		Conditional styles with JavaScript are used		

Learning Outcome 2 End Assessment

Written assessment

Q1. Read carefully the following statements about Tailwind CSS framework and answer by TRUE if the statement is correct and by FALSE if the statement is incorrect.

a) Tailwind CSS does not support grid-based layouts for creating flexible, responsive designs. **Answer**: **False**

b) The mobile-first approach means designing for smaller screens first and progressively enhancing the layout for larger screens. **Answer**: **True**

c) Tailwind CSS does not allow developers to customize typography or scale font sizes for responsive design. **Answer**: **False**

d) Tailwind CSS offers utility classes to style interactive elements such as buttons, forms, and hover states, ensuring responsiveness. **Answer**: **True**

e) Testing responsiveness and iterating on design are not necessary when using Tailwind CSS because it automatically makes everything responsive. **Answer**: **False**

f) You can extend Tailwind's default theme by modifying the tailwind.config.js file. **Answer: True**

g) Custom fonts can be added to Tailwind by specifying them in the theme's fontFamily section of the configuration file. **Answer: True**

h) Tailwind CSS plugins can add additional utilities and variants for extended functionality. **Answer: True**

i) You can apply conditional styles in Tailwind by integrating JavaScript logic using libraries like Alpine.js or React's state management. **Answer: True**

Q2. Fill in the blank spaces with appropriate word(s). Select from the given choices in the box.

grid, colors, focus state flex-initial, tailwind.config.js, Hover state, Animation, transition, animate-bounce, delay-*, mobile-first, and media queries

a) .... in Tailwind CSS is a feature that allows you to apply styles to an element when a user hovers over it with their cursor.

b) You can add plugins by installing them and then including them in the plugins array of ....

c) The .... approach is a design strategy where you prioritize mobile users by designing for smaller screens first and then progressively enhancing for larger screens.

d) Add the **..** utility to make an element bounce up and down, useful for things like "scroll down" indicators.

e) ……………………………… is the ability to apply motion effects to elements, either using pre-defined utility classes or custom animations.

f) You can customize the primary color by extending the ……………………………… property in tailwind.config.js

g) Use the ……………………………… utilities to control an element's transition-delay.

h) Tailwind CSS provides pre-configured …………….. through its breakpoint utilities like sm, md, lg, xl, and 2xl.

i) Use ……………………… to allow a flex item to shrink but not grow, taking into account its initial size.

j) In Tailwind CSS, you can implement flexible grid layouts using the ……………………… utility class and specifying the number of columns with grid-cols-{n}.

Answer:

a) Hover state
b) tailwind.config.js
c) mobile-first
d) animate-bounce
e) Animation
f) colors
g) delay-*
h) media queries
i) flex-initial
j) grid

Q3. Match Column A and Column B

Column A		Column B
1.Utilities for controlling gutters between grid and flexbox items		**A.** Auto-cols-min
2. Utility for controlling the size of implicitly-created grid rows.		**B.** shrink-0
3. Utility for controlling the size of implicitly-created grid columns.		**C.** animate-pulse
4. Utility for specifying the rows in a grid layout.		**D.** Gap
5. It is used to prevent a flex item from shrinking.		**E.** grow-0

6. It is used to make an element gently fade in and out, useful for things like skeleton loaders.		**F.** Auto-rows-fr
7.It is used to add the margins to paragraph.		**G**.grid-rows-5
8.It is used to apply the line spacing to text.		**H.**md
9.Example of breakpoint		**I.**m-4
10. It is used to apply the extra-large size to text		**J.** leading-normal
		K. text-xl
		L. text-2xl
		M. p-4

Answers:

1.D

2.F

3.A

4.G

5.B

6.C

7.I

8.J

9.H

10.K

Practical assessment

ABC Company is a software development company located in Nyanza district, Southern Province – Rwanda. The company has developed a ReactJS application with an unpleasant and not attractive design. It wants to improve its website development Techniques to modern ones. Assume it has hired you as a ReactJS expert and tasked you to develop a modern, responsive web application using React.JS and Tailwind CSS with a clean and consistent design across all devices. You are requested to integrate animations, transitions, handle various states like hover and focus, and follow a mobile-first approach. Additionally, the application requires custom styles to match the branding of the company, including custom fonts, colours, and layout directives.

Checklist

SN	Criteria	Indicators	Yes	No
1	Tailwind utility classes are well applied	Tailwind CSS is installed		
		Tailwind CSS is configured		
		Utility-First Fundamentals are used		
		States are used		
		Animation is used		
		Transitions are used		
		Grids are used		
		Reusable styles are used		
		Custom styles are well added		
		Function is used		
2	Responsive design principles are correctly applied	Mobile –first approach is used		
		Flexible grid layouts are used		
		Responsive images are used		
		Media queries and breakpoints are used		
		Typography and readability are used		
		Interactive elements are applied		
		Iteration is used		

		Testing is performed		
3	Tailwind styles are correctly customized	Default theme is used		
		Custom variants are used		
		Custom fonts are used		
		Custom typography is used		
		Colors are customized		
		Plugin is used		
		Custom directive is used		
		Conditional styles with JavaScript are used		

Further information to the trainer

House, C. (2024, September 26). *CSS grid layout guide*. CSS-Tricks. https://css-tricks.com/snippets/css/complete-guide-grid/#aa-css-grid-properties

Refine. (n.d.). *Tailwind CSS flex: A comprehensive guide*. Refine. https://refine.dev/blog/tailwind-flex/#icons

Tailwind CSS. (n.d.). *Create React App*. https://tailwindcss.com/docs/guides/create-react-app

Tailwind CSS. (n.d.). *Functions and directives*. Tailwind CSS Documentation. https://v2.tailwindcss.com/docs/functions-and-directives#screen-1

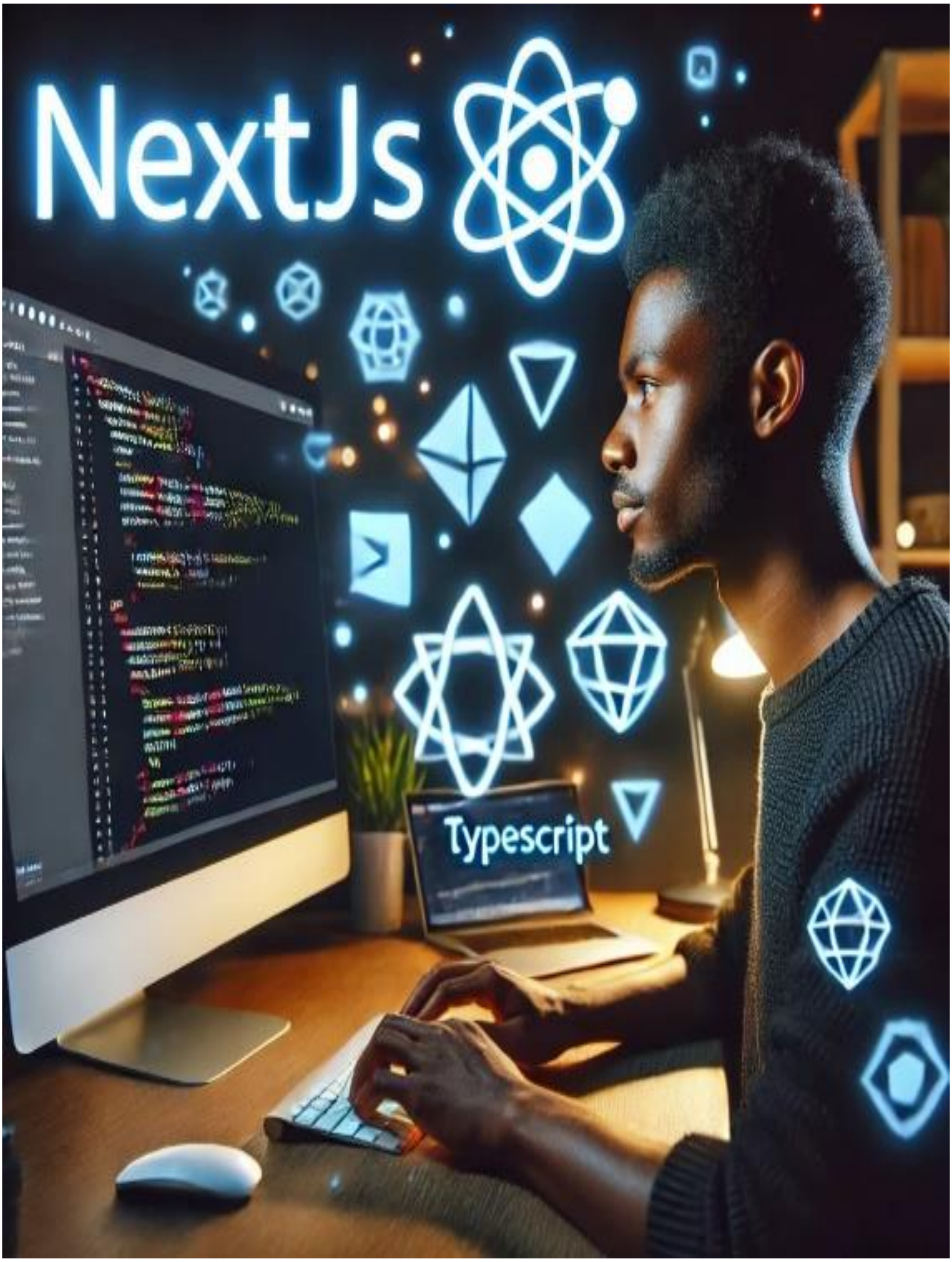
Tailwind CSS. (n.d.). *Gap*. Tailwind CSS. https://tailwindcss.com/docs/gap

Tailwind CSS. (n.d.). *Grid template columns*. Tailwind CSS. https://tailwindcss.com/docs/grid-template-columns

Tailwind CSS. (n.d.). *Installation*. Tailwind CSS. Retrieved October 14, 2024, from https://tailwindcss.com/docs/installation

Tailwind CSS. (n.d.). *Plugins*. https://tailwindcss.com/docs/plugins

Learning Outcome 3: Develop Next.JS Application



Indicative contents
3.1 Applying TypeScript basics **3.2 Setup Next.JS project** **3.3 Implementing Rendering Techniques** **3.4 Implementing routing** **3.5 Creation of API** **3.6 Securing the Application**

Key Competencies for Learning Outcome 3: Develop Next.JS Application

Knowledge	Skills	Attitudes
● Description of Rendering Techniques ● Description of routing concepts ● Definition of API Endpoint.	● Setting up TypeScript environment ● Implementing interface of variables ● Performing functions and data Handling in TypeScript ● Creating NextJS project ● Initializing NextJS project Development ● Implementing Rendering Techniques ● Implementing routing ● Creating API ● Performing Client-Side Security in application ● Performing Server-Side Security in application ● Performing General Security Measures	● Being curios while developing Next.JS project ● Being creative in developing up-to-date next.JS apps ● Being Problem-Solving Oriented during development of Next.JS Application ● Paying attention to Details during implementation of rendering techniques ● Being updated on latest Next.JS versions ● Being Adaptable to the Next.JS environment

Duration: 20 hrs

Learning outcome 3 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe clearly Rendering Techniques and routing concepts based on user requirements
2. Prepare properly Typescript environment based on project requirements
3. Implement correctly interface of variables based on typescript standard
4. Perform correctly Handling functions and Data Handling based on Typescript standard
5. Initialize properly NextJS project Development based on project created
6. Routing is properly implemented based on components.
7. API is properly created based on RESTful design principles.
8. Application is properly secured based on system requirements and security standards

Resources

Equipment	**Tools**	**Materials**
● Computer ●	● VS Code ● Browser ● Terminal ● Postman	● Internet ● Electricity

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Avail a computer lab with internet connectivity
- Avail a VS Code, Node.JS and Browser installed on all computers to be used
- Avail a sample Next.JS application developed to be used in security application

Indicative content 3.1: Applying TypeScript Basics

Duration: 5 hrs

Practical Activity 3.1.1: Applying TypeScript Basics

Notes to the trainer

- While delivering this content, you are required to check if computers have terminal and node.js
- Avail videos to be used as didactic material
- Avail a computer with internet connection

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to go to the computer lab to:

1. Install TypeScript
2. Configuring type Script
3. Implement interface of variables
4. Handling functions
5. Data Handling in TypeScript

Step 2: Provide clear work instructions.

Step 3: Demonstrate how to:

1. Install TypeScript
2. Configuring type Script
3. Implement interface of variables
4. Handling functions
5. Data Handling in TypeScript

While demonstrating, explain the steps to follow.

Step 4: Asks learners to Install TypeScript, configure type Script, Implement interface of variables, Handle functions, Perform Data Handling in TypeScript and monitor the procedures.

Step 5: Verify if TypeScript is installed and configured, interface of variables is implemented, functions and Data are clearly handled in TypeScript

Step 6: Ask trainees to read key readings 3.1.1

Steps7: Ask trainees to perform the task provided in the application of learning 3.1

Points to Remember

- npm install -g typescript command to install Typescripts using terminal
- Steps to implement interfaces for variables in TypeScript
  - ✓ Defining an Interface
  - ✓ Implementing the defined Interface for Variables
  - ✓ Optional Properties in Interfaces
  - ✓ Read-Only Properties in Interfaces
  - ✓ Implementing Methods in Interfaces
  - ✓ Extending Interfaces
  - ✓ Indexable Types in Interfaces
- Data Handling like API data validation, Form validation and, Error handling and exceptions

Application of learning 3.1.

ABC Web Development Company is an IT company in Kigali City – Rwanda. Assume they have hired you as a full stack developer, you are tasked with creating a task management app using ReactJS and TypeScript. Install and configure TypeScript, ensuring the tsconfig.json file is set up. Implement interfaces to define task objects, then develop functions to add, update, and manage tasks. Make sure that API data validation is implemented to ensure data integrity, followed by form validation for user inputs. Finally, integrate additional packages, such as Formik for form handling or Axios for API requests, enhancing functionality and performance.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.		Typescript is well installed		

	TypeScript environment is well prepared	Typescript is well configured		
		tsconfig.json file exist		
2.	Data structures are well classified	Interfaces for task objects are implemented		
		functions to add, update, and manage tasks is defined		
3.	Data is properly handled	API data validation is implemented		
		Form validation is implemented		
		Additional packages are integrated		

Indicative content 3.2: Setup NextJS project

Duration: 2 hrs

Practical Activity 3.2.1: Setup NextJS project

Notes to the trainer

- This activity should take place in a computer lab where trainees should create NextJS Project.
- Avail computers connected to the internet

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to go to the computer lab to create a NextJS project and initialize environment.

Step 2: Provide clear work instructions.

Step 3: Demonstrate how to create NextJS project and Initialize environment. While demonstrating, explain the steps to be followed.

Step 4: Ask learners to create NextJS project and Initialize environment and monitor the activity

Step 5: Verify whether NextJS project is well created and environment is properly initialized

Step 6: Ask trainees to read key readings 3.2.1.

Steps7: Ask trainees to perform the task provided in the application of learning 3.2

Points to Remember

- Steps to prepare a Next.js project environment
  - ✓ Install Node.js and npm
  - ✓ Set Up a New Next.js Project
  - ✓ Navigate to Your Project Directory

  - ✓ Run the Development Server
  - ✓ Explore the Project Structure
  - ✓ Install Additional Packages
  - ✓ Using TypeScript (Optional)
  - ✓ Environment Variables (Optional)
  - ✓ Configure ESLint (Optional)
  - ✓ Build and Production
- Steps to create NextJS Project
  - ✓ Install Node.js (if not already installed)
  - ✓ Create a New Next.js Project
  - ✓ Navigate to the Project Directory
  - ✓ Run the Development Server
  - ✓ Explore Your Next.js App
- Initialize NextJS project environment
  - ✓ Creating Pages and components
  - ✓ Implementing search engine optimization (SEO)
  - ✓ Styling
  - ✓ Caching Strategies

Application of learning 3.2.

ABC Web Development Company is a software development company located in Kacyiru-Kigali City. Assume you have been hired as Next.js developer and tasked with creating a portfolio website for a client by preparing the Next.js environment, installing Node.js and the Next.js framework. You have to also create a new Next.js project using the command line. After project creation, initialize the development environment with essential dependencies and configuration settings. Finally, apply CSS for styling the website, ensuring a responsive design that enhances user experience across various devices.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Next.js environment is well prepared	Node.js is installed		
		Npm exist in your package		
2	Project is well created	Project is created		
		Dependencies are installed		

3	Environment is well initialized	Pages and components are Created		
		Designed pages are user-friendly interfaces.		
		search engine optimization (SEO) is Implemented		
4	CSS is properly applied	styling is handled using CSS Modules		

Indicative content 3.3: Implementing Rendering Techniques

Duration: 2 hrs

Theoretical Activity 3.3.1: Description of Rendering Techniques

Notes to the trainer:

- While delivering this content, a small group can be used for describing Rendering Technique
- Avail videos to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask learners to answer the following questions:

i. What is Static Site Generation (SSG), and how does it work?
ii. How does SSG handle dynamic content updates?
iii. What is Server-Side Rendering (SSR), and how does it differ from SSG?
iv. What are the advantages of SSR for user experience and SEO?
v. What are the performance implications of using ISR?
vi. How does CSR impact initial load time and user experience?

Step 2: Ask learner to write answers provided on flipchart/paper.

Step 3: Engage the trainees to present their findings

Step 4: Provide an expert view and clarify the ideas.

Step 5: Address any questions or concerns raised by the trainees

Step 6: Ask trainees to read the key readings 3.3.1 in the trainee manual.

Points to Remember

- **Static Site Generation (SSG)** is a rendering method where HTML pages are generated at build time, resulting in static files that can be served directly to users.
- **Server-Side Rendering (SSR)** is a rendering technique where HTML pages are generated on the server for each request.
- **Incremental Static Regeneration (ISR**) is an enhancement of SSG that allows static pages to be updated after the initial build.
- **Client-Side Rendering (CSR)** is a rendering method where the browser dynamically generates HTML using JavaScript after the initial load.

Practical Activity 3.3.2: Implementing Rendering Techniques

Notes to the trainer

- While delivering this content, you are required to check if computers have Next.js Framework

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners to go to the computer lab to apply rendering techniques in NextJS Framework

Step 2: Provide clear work instruction.

Step 3: Demonstrate how to apply rendering techniques in NextJS Framework

Step 4: Ask trainees to apply rendering techniques in NextJS Framework.

Step 5: Verify whether rendering techniques are applied

Step 6: Ask trainees to read key readings 3.3.2.

Step 7: Ask trainees to perform the task provided in the application of learning 3.3

Points to Remember

- Steps to implement Static Site Generation (SSG)
  - ✓ Create a Next.js Project:
  - ✓ Use getStaticProps:
- Steps to implement Server-Side Rendering(SSR)
  - ✓ Create a Next.js Project (if not already done)
  - ✓ Use getServerSideProps
- Steps to implement Incremental Static Regeneration (ISR)
  - ✓ Create a Next.js Project (if not already done)
  - ✓ Use getStaticProps with Revalidation:
- Steps to implement Client-Side Rendering (CSR)
  - ✓ Create a Next.js Project (if not already done)
  - ✓ Fetch Data on the Client-Side

Application of learning 3.3.

ABC is a business company located in Kimihurura – Kigali City. It is planning to create a real-time Next.js e-commerce website. Assume you have been hired as a Next.js developer. You are tasked with building a product listing page for an e-commerce site using Next.js. The page should display a dynamic list of products fetched from an API, allowing users to filter by category and sort by price. Additionally, when users scroll down, more products should load seamlessly (infinite scrolling). You are requested to implement efficient rendering techniques, like server-side rendering (SSR) and static site generation (SSG), to handle the dynamic UI changes and optimize performance, ensuring the app renders smoothly without unnecessary re-renders or slow loading times

Checklist

SN	Criteria	Indicators	Yes	No
1	NextJS application is well created	NextJS application is created		
		Pages are created		
		Pages are designed		
2	API Integration is properly implemented	Product Lists are fetched		
		Prices are filtered		

		Seamless loading is enabled		
3	Rendering techniques are properly applied	Static Site Generation is applied		
		Server-Side Rendering is applied		
		Client-Side Rendering is applied		

Indicative content 3.4: Implementing routing

Duration: 4 hrs

Theoretical Activity 3.4.1: Description of Routing

Notes to the trainer:

- While delivering this content, a small group can be used for describing routing

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and request learners to respond to the following questions:

i. Describe the following concepts:
   a. File-system Based Routing
   b. Dynamic Routes
   c. Nested Routes
   d. Link Component
   e. Programmatic Navigation
   f. API Routes
   g. Catch-all Routes

ii. What is the difference between using the <Link> component and a regular <a> tag?

iii. What methods can you use for programmatic navigation in Next.js?

iv. What are some use cases for using dynamic routes in a web application?

Step 2: Ask any learner to write answers provided on flipchart/paper.
Step 3: Engage learners to present their findings
Step 4: Provide an expert view and clarify the ideas
Step 5: Address any questions or concerns raised by the trainees.
Step 6: Ask trainees to read the key readings 3.4.1 in the trainee manual.

Points to Remember

- **File-System Based Routing.** A routing system where the structure of the page's directory directly corresponds to the application's URL paths.
- **Dynamic Routes.** Routes that allow for variable content based on URL parameters, defined using square brackets in the file name (e.g., pages/posts/[id].js).
- **Nested Routes.** Routes organized in a hierarchy using subfolders within the page's directory.
- **Link Component.** A built-in component in Next.js for client-side navigation between pages.
- **Programmatic Navigation.** A method to navigate between routes using code, typically via the useRouter hook.
- **API Routes.** Endpoints created in the pages/api directory that allow you to handle server-side logic and respond to HTTP requests.
- **Catch-all Routes.** Routes that capture all paths under a certain route prefix using double square brackets (e.g., pages/docs/[...slug].js).
- **Linking Components**. The practice of using the <Link> component from Next.js to create client-side navigation between different pages in your application.
- **Programmatic Navigation.** The ability to navigate between routes using JavaScript code instead of static links.
- **Dynamic Routes.** Routes that can handle variable segments in the URL, defined using square brackets in the file name (e.g., pages/posts/[id].js).
- **Query Parameters.** Key-value pairs appended to the URL that can be used to pass additional data to a page (e.g., /search?query=nextjs).

Practical Activity 3.4.2: Implementing of routing

Notes to the trainer

- While delivering this content, you are required to check if computers have Next.js Framework

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to go to the computer lab to implement Next.JS routing.

Step 2: Provide clear work instructions.

Step 3: Demonstrate how to implement Next.JS routing. While demonstrating, explain the steps to be followed

Step 4: Ask learners to implement Next.JS routing and monitor the activity

Step 5: Verify whether Next.JS route is correctly implemented.

Step 6: Ask trainees to read key readings 3.4.2

Step7: Ask trainees to perform the task provided in the application of learning 3.4

Points to Remember

- Steps to implement NextJS Routing
  - ✓ Linking Components
  - ✓ Programmatic Navigation
  - ✓ Dynamic Routes
  - ✓ Query Parameters

Application of learning 3.4.

ABC is a business company located in Kimihurura – Kigali City. It is planning to create a real-time Next.js e-commerce website. It is planning to create an e commerce website with real time loading links enabled by routing. Assume you have been hired to create their application using Next.js to explore key routing features. Start by setting up the project and creating a structured file hierarchy, including a home page, a dynamic route for individual posts, and a posts listing page populated with sample data. Implement a PostLink component to allow navigation to individual posts, and display all posts on the home page. Create a dynamic route that shows the post content and incorporates programmatic navigation for returning to the home page. Finally, enhance the user experience by adding a filtering feature that uses query parameters to dynamically filter posts based on user input, showcasing the core capabilities of Next.js routing.

Checklist

SN	Criteria	Indicators	Yes	No
1	Next.JS project is properly setup	home page is created		
		A dynamic route for individual posts is created		
		A posts listing page populated with sample data is created		
2	Components are clearly linked	PostLink component is Implemented		
		all posts are displayed on the home page		
3	Dynamic Routes are properly implemented	Dynamic route is created		
		Navigation redirect to homepage		
4	Query Parameters are clearly implemented	filtering feature is added		
		filtering feature is applied		

Indicative content 3.5: Creation of API

Duration: 3 hrs

Practical Activity 3.5.1: Creating API

Notes to the trainer

- This activity should take place in a computer lab where trainees should create API.
- Avail computers with internet

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to go to the computer lab to create API, define the API Endpoint, handle Request Types, use Dynamic API Routes, and test the API
Step 2: Provide the clear work instructions.
Step 3: Demonstrate how to create API, define the API Endpoint, handle Request Types, use Dynamic API Routes, and test the API. While demonstrating, explain the steps to be followed.
Step 4: Ask learners to create API, define the API Endpoint, handle Request Types, use Dynamic API Routes, and test the API and monitor the procedures.
Step 5: Verify whether the API is well created, API endpoint is well defined, Request types are well handled, Dynamic API Routes are well used and API is well tested.
Step 6: Ask trainees to read key readings 3.5.1.
Step7: Ask trainees to perform the task provided in the application of learning 3.5

Points to Remember

- **Define the API Endpoint.** This involves creating a specific route in your Next.js application that handles API requests.
- **Handling Request Types.** Next.js API routes can handle different HTTP request methods, such as GET, POST, PUT, and DELETE.
- **Using Dynamic API Routes.** Dynamic API routes allow you to create endpoints that can handle variable parts in the URL.

- **Testing your API.** Testing your API involves verifying that your endpoints work as expected.

Application of learning 3.5.

HTD Company is a software development company located in Kicukiro District – Kigali City. It wants to hire a full stack developer to integrate their front-end application to other external applications in order to share data. Assume you are hired as their developer. You are tasked to develop a simple task management application using Next.js. Create an API that defines a central endpoint at pages/api/tasks.js, which handles various request types such as GET for retrieving tasks, POST for creating new tasks, PUT for updating existing ones, and DELETE for removing tasks. To enhance functionality, implement dynamic API routes like pages/api/tasks/[id].js, allowing users to interact with specific tasks based on their unique identifiers. Finally, test your API using tools like Postman or curl to ensure each operation—retrieving, creating, updating, and deleting tasks—works correctly and returns the expected responses.

Checklist:

SN	Criteria	Indicators	Yes	No
1	API Endpoint clearly defined	API endpoints are selected		
		API endpoints are created		
		API endpoints are implemented		
2	Request Types are well handled	GET is handled for retrieving tasks		
		POST is handled for creating new tasks		
		PUT is handled for updating existing ones		
		DELETE is handled for removing tasks		
3	Dynamic API routes are properly used	Routes are installed		
		Routes are imported		
		Route is used		
4		Testing tool is selected		

	API is properly tested	Testing tool is used		
		Test results are interpreted		

Indicative content 3.6: Securing the Application

Duration: 4 hrs

Practical Activity 3.6.1: Securing the Application

Notes to the trainer

- This activity should take place in a computer lab where trainees should Perform Client-Side Security, Perform Server-Side Security, Perform General Security Measures
- Avail computer connected to the internet.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners to go to the computer lab to Perform Client-Side Security, Server-Side Security, and General Security Measures.

Step 2: Provide clear work instructions.

Step 3: Demonstrate how to perform Client-Side Security, Perform Server-Side Security, and Perform General Security Measures. While demonstrating, explain the steps to be followed.

Step 4: Ask learners to Perform Client-Side Security, Server-Side Security, and General Security Measures and monitor the procedures.

Step 5: Verify whether Client-Side Security is well performed, Server-Side Security well performed, and General Security Measures are properly performed.

Step 6: Ask trainees to read key readings 3.6.1.

Step7: Ask trainees to perform the task provided in the application of learning 3.6

Points to Remember

- Performing Client-Side Security
- Performing Server-Side Security
- Performing General Security Measures

Application of learning 3.6.

XYZ Developers is a web application company located in Kigali city. It has developed a banking application for XMY bank in Next.JS and requires a competent developer to manage security in the application before it is deployed. Assume it has hired you to you to secure their application. You are requested to implement a comprehensive strategy that encompasses both client-side and server-side measures. Start by focusing on validating user inputs, manage sessions with authentication libraries, and configure access controls to limit unauthorized API access. Ensure secure data transmission, protect sensitive information during rendering, and enforce checks for accessing API routes. Establish policies to prevent various types of attacks and incorporate general security measures such as regular audits, secure headers, and keeping third-party libraries updated.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Client-Side Security is well performed	User inputs are validated		
		Sessions are applied		
		Access controls is configured		
2	Server-Side Security is well performed	Data transmission is secured		
		Sensitive information is protected		
		Accessibility of API routes is checked		
3	General Security Measures are properly performed	Regular audits are established		
		Security of headers is implemented		
		Third-party libraries are updated		

Learning Outcome 3 End Assessment

Written assessment

Q1. Read the Following statement and answer by true if correct or false otherwise

TypeScript is a statically typed superset of JavaScript that requires a separate compiler, enhancing type checking and error detection. While it lacks configuration options for module resolution, it offers features like interfaces and enums for better code organization. In frameworks like Next.js, nested routes allow for a structured routing system, but API routes do not support dynamic URL segments. Client-side security measures are essential even with server-side rendering (SSR). The Next.js Link component facilitates navigation within the app rather than to external URLs. Additionally, implementing a Content Security Policy (CSP) is effective in mitigating cross-site scripting (XSS) attacks, enhancing security.

a. TypeScript requires a separate compiler that is different from JavaScript.

b. TypeScript does not support configuration options for module resolution

c. Nested routes allow you to create child routes within a parent route.

d. API routes in Next.js cannot handle dynamic segments in the URL.

e. Performing client-side security measures is unnecessary if you are using server-side rendering (SSR).

f. The Link component in Next.js can only navigate to external URLs.

g. Implementing a Content Security Policy (CSP) can help mitigate cross-site scripting (XSS) attacks.

Answer

1. **True** - TypeScript requires a separate compiler that is different from JavaScript.
2. **False** - TypeScript supports configuration options for module resolution.
3. **True** - Nested routes allow you to create child routes within a parent route.
4. **False** - API routes in Next.js can handle dynamic segments in the URL.
5. **False** - Performing client-side security measures is still necessary even if using server-side rendering (SSR).
6. **False** - The Link component in Next.js can navigate to both internal and external URLs.
7. **True** - Implementing a Content Security Policy (CSP) can help mitigate cross-site scripting (XSS) attacks.

Q2. Read carefully the following questions related to routing in Next.js and circle the most correct alternative alphabetic letter

A) Dynamically generate routes

B) Create pages based on the directory structure

C) Write API routes

D) None of the above

Answer: B) Create pages based on the directory structure

Q3. Which component is used for client-side navigation in Next.js?

A) Router

B) Link

C) Navigate

D) Route

Answer: B) Link

Q4. Dynamic routes in Next.js can be created by using:

A) [] brackets in the filename

B) () brackets in the filename

C) # symbols in the filename

D) None of the above

Answer: A) [] brackets in the filename

Q5. Which of the following is NOT a request type handled by an API?

A) GET

B) POST

C) PATCH

D) EXECUTE

Answer: D) EXECUTE

Q6. To secure your Next.js application against cross-origin attacks, you should implement:

A) CORS

B) HTTPS

C) CSP

D) All of the above

Answer: **D) All of the above**

Q7. Read the following sentences and Fill the gap with the missing word.

The purpose of __________ is to manage user sessions and authenticate requests on the server side.

a. Using __________ in Next.js allows you to pass parameters directly in the URL and retrieve them in your component.
b. Programmatic navigation can be achieved using the __________ method provided by the Next.js router.
c. To create a catch-all route, you would use the syntax __________ in the filename.
d. Testing your API can be done using tools like __________ or Postman to ensure the endpoints are functioning correctly.

Answer:

a. The purpose of **authentication** is to manage user sessions and authenticate requests on the server side.
b. Using **dynamic routing** in Next.js allows you to pass parameters directly in the URL and retrieve them in your component.
c. Programmatic navigation can be achieved using the **push** method provided by the Next.js router.
d. To create a catch-all route, you would use the syntax **[...param]** in the filename.
e. Testing your API can be done using tools like **cURL** or Postman to ensure the endpoints are functioning correctly.

Q8. Match each concept on the left with the correct description or related item on the right.

Concepts	**Descriptions**
1. Preparation of Environment	A. The process of defining the layout and visual appearance of your application using CSS or a CSS-in-JS library.
2. Project Creation	B. The configuration of tools and dependencies, such as Node.js and npm, required to run a Next.js application.
3. Creating Pages and Components	C. A method to enhance web performance by storing and retrieving data from a cache rather than fetching it each time.
4. Implementing Search Engine Optimization (SEO)	D. The use of meta tags and structured data to improve visibility and ranking in search engines.
5. Styling	E. The fundamental building blocks of a Next.js application, typically stored in the pages directory.
6. Caching Strategies	F. The command npx create-next-app used to initiate a new Next.js project.
7. API Integration	

8. Testing and Quality Assurance	

Answers:

1. **Preparation of Environment** - **B.** The configuration of tools and dependencies, such as Node.js and npm, required to run a Next.js application.
2. **Project Creation** - **F.** The command npx create-next-app used to initiate a new Next.js project.
3. **Creating Pages and Components** - **E.** The fundamental building blocks of a Next.js application, typically stored in the pages directory.
4. **Implementing Search Engine Optimization (SEO)** - **D.** The use of meta tags and structured data to improve visibility and ranking in search engines.
5. **Styling** - **A.** The process of defining the layout and visual appearance of your application using CSS or a CSS-in-JS library.
6. **Caching Strategies** - **C.** A method to enhance web performance by storing and retrieving data from a cache rather than fetching it each time.

Practical assessment

XYZ Developers is a web application company located in Kigali city. It needs to hire a Next.JS developer for the company. Assume you are hired as their developer and tasked with preparing a travel booking platform using Next.js and TypeScript. You are requested to start by setting up the Next.js project with TypeScript support. Ensure safety throughout your application. Implement various rendering techniques, such as server-side rendering and static site generation, to optimize performance and improve user experience. Create a robust API to handle user requests for searching and booking accommodations, activities, and transportation. Finally, leverage Next.js routing to establish a seamless navigation system between key pages, including Home, Search Results, and Booking Details, ensuring a smooth user journey throughout the platform.

Checklist

SN	Criteria	Indicators	Yes	No
1	Project environment is properly prepared	Node.js and npm are installed		
		Project is created		
		Typescript is used		
2	Rendering Techniques are correctly implemented	Server-side rendering (SSR) is implemented		
		Static site generation (SSG) is used		

		Loading times is optimized		
3	API is well developed	Robust API is created		
		Endpoints is implemented		
		Errors are handled		
		API responses are validated		
4	Next.js Routing is properly created	Seamless routing system is prepared		
		Key pages are created		
		Dynamic routing is created		
5	User Experience is correctly applied	User-friendly interface is designed		
		Mobile responsiveness is applied		
6	Deployment is well done	The application for deployment is available		
		Hosting platform is chosen		
		Application performance is checked		

END

Further information to the trainer

Morrow, T. (2020). *Understanding Client-Side Rendering vs. Server-Side Rendering*. Smashing Magazine. https://www.smashingmagazine.com/2020/06/client-side-rendering-vs-server-side-rendering/

Tailwind CSS. (n.d.). *Using Next.js with Tailwind CSS*. https://tailwindcss.com/docs/guides/nextjs

ts-node. (n.d.). ts-node Documentation. https://typestrong.org/ts-node/

TypeScript. (n.d.). Interfaces in TypeScript. https://www.typescriptlang.org/docs/handbook/2/objects.html#interfaces

TypeScript. (n.d.). tsconfig.json Reference. https://www.typescriptlang.org/tsconfig

TypeScript. (n.d.). TypeScript Handbook. https://www.typescriptlang.org/docs/handbook/intro.html

Vercel. (n.d.). Configuring TypeScript in Next.js. https://nextjs.org/docs/app/building-your-application/configuring-typescript

Vercel. (n.d.). Create a Next.js app. https://nextjs.org/docs/app/building-your-application/create-next-app

Vercel. (n.d.). Data fetching in Next.js. https://nextjs.org/docs/app/building-your-application/data-fetching

Vercel. (n.d.). File-system routing in Next.js. https://nextjs.org/docs/app/building-your-application/routing/file-system-routing

Vercel. (n.d.). Next.js documentation. https://nextjs.org/docs

Vercel. (n.d.). Next.js project structure. https://nextjs.org/docs/app/building-your-application/project-structure

Vercel. (n.d.). Rendering methods in Next.js. https://nextjs.org/docs/app/building-your-application/rendering

Vercel. (n.d.). Running a development server in Next.js. https://nextjs.org/docs/app/building-your-application/development

Vercel. (n.d.). Setting up ESLint in Next.js. https://nextjs.org/docs/app/building-your-application/configuring-linting

Vercel. (n.d.). Static files in Next.js. https://nextjs.org/docs/app/building-your-application/static-file-serving

Vercel. (n.d.). Using the App Router. https://nextjs.org/docs/app/building-your-application/routing/app-router

Williams, C. (2021). *Static Site Generation with Next.js*. CSS-Tricks. https://css-tricks.com/static-site-generation-with-next-js/

Learning Outcome 4: Apply Progressive Web Application

Indicative contents

4.1 Maintain Responsiveness
4.2 Configuring Web Application Manifest
4.3 Implementation of Service Workers

Key Competencies for Learning Outcome 4: Apply progressive web application

Knowledge	Skills	Attitudes
● Description of mobile first design ● Identification of performance optimization techniques ● Description of manifest file ● Description of service workers	● Maintaining the responsiveness ● Configuring web application manifest ● Implementing the service workers	● Being creative in developing progressive web application ● Being problem-solving oriented during development of progressive web application ● Being updated on latest React.JS versions ● Being collaborative while developing of progressive web application

Duration: 20 hrs

Learning outcome 4 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe clearly mobile first design in web application development.
2. Identify correctly the performance optimization techniques that are used in web application development.
3. Describe clearly the manifest file and service workers in development of web application.
4. Maintain correctly the responsiveness of web application in development.
5. Configure correctly web application manifest in the development.
6. Implement correctly the service workers in development of progressive web application.

Resources

Equipment	**Tools**	**Materials**
● Computer	● VS Code ● Browser ● Terminal	● Internet ● Electricity

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have prepared computer lab with Internet connectivity
- Have visual studio code installed on all computers to be used.
- Have videos to be used as didactic material.
- Have sample React.JS project developed to be used.

Indicative content 4.1: Maintain Responsiveness

Duration: 8 hrs

Theoretical Activity 4.1.1: Description of web application responsiveness

Notes to the trainer:

- While delivering this content small groups can be used to describe the web application responsiveness.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. What do you understand by:

a) Leverage progressive enhancement

b) Prioritizing mobile-first design

ii. What are the performance optimization techniques of PWA?

Step 2: Ask trainees to write findings on paper or flipchart.

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees.

Step 6: Ask the trainees to read key readings 4.1.1 in their manual.

Points to Remember

- **Leverage progressive enhancement** is a design strategy where you build the most basic, core functionality first, ensuring it works across all devices and browsers.
- **Prioritizing mobile-first design** in a Progressive Web Application (PWA) means designing and developing your application primarily for mobile devices before scaling up to larger screens like tablets and desktops.
- **Code splitting** helps break down your application into smaller chunks, which can be loaded on demand.
- **Tailwind CSS Purge** is a feature designed to remove unused CSS from your production build, significantly reducing the size of your final CSS file.

- **Initial load time** is the time it takes for a web page or application to fully load and become usable when a user first accesses it.
- **Minimizing and compressing assets** is to reducing the size of your CSS, JavaScript, HTML, and other assets (like images) to improve load times, performance, and overall user experience.
- **Efficient CSS with Tailwind CSS** refers to using Tailwind CSS to create fast, scalable, and maintainable styles by optimizing how CSS is written and applied in web projects.

Practical Activity 4.1.2: Maintaining web application responsiveness

Notes to the trainer

- The trainer asks trainees to maintain the responsiveness of web application.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to maintain the responsiveness of web application

Step 2: Provide the clear work instructions.

Step 3: Demonstrate how to maintain the web application responsiveness. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to maintain the web application responsiveness and monitor the activity.

Step 5: Verify whether web application responsiveness is correctly maintained.

Step 6: Ask trainees to read the key readings 4.1.2

Step 7: Ask trainees to perform the task provided in application of learning 4.1

Points to Remember

- To maintain the web application responsiveness, you do the followings:
  - ✓ Use Tailwind’s utility classes for styling.
  - ✓ Use Tailwind CSS Purge
  - ✓ Optimizing the image
  - ✓ Make sure your text scales well across devices.

Application of learning 4.1.

ABC is a company located in Kabuga- Kigali city. It sells It equipment online. It has an online website which the customers are not happy about its displays on all gadgets. It requires to make it user friendly to all internet accessing gadget. Assume it has hired you to solve the problem, you are tasked with developing a Progressive Web Application (PWA) for the company that caters to both desktop and mobile users. The company expects the PWA to deliver an optimal and responsive experience across all devices, including those with varying network speeds and hardware capabilities. The PWA should be performant, fast-loading, and fully functional on older browsers as well as the latest ones.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Web application responsiveness is correctly maintained	Basic functionality first used		
		Progressive enhancement with advanced features are used.		
		Design for small screens first used		
		Responsive units and breakpoints are selected		
		Performance and loading times are applied		
		Responsive typography is used		
		Code splitting is used		
		TailwindCSS Purge is used		
		Initial load time is improved		
		Assets are compressed		
		Performance is optimized		

Indicative content 4.2: Configuring Web Application Manifest

Duration: 5 hrs

Theoretical Activity 4.2.1: Description of web application manifest

Notes to the trainer:

- While delivering this content small groups can be used to describe the web application manifest.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. What is manifest file?
ii. What are the key functions of manifest.json?
iii. Outline the common properties of manifest.json?
iv. Explain the steps used for referencing the manifest in your HTML.
v. Explain how can you test React application in development environment with Lig hthouse.

Step 2: Ask trainees to write their findings on paper or flipchart

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas.

Step 5: Address any question or concern raised by the trainees.

Step 6: Ask the trainees to read the key readings 4.2.1 in their manual.

Points to Remember

- **Manifest file** is a JSON file that provides important metadata about the web application, helping to control how the app appears to the user and how it behaves when installed on a device.

- **name**: The full name of the application (used in the install prompt or app listings).
- **short_name**: A short version of the app name, often used when space is limited (e.g., on a mobile home screen).

- **start_url**: The URL that is loaded when the app is launched from the home screen or desktop.
- **display**: Controls how the app is displayed.
- **background_color**: The color shown on the splash screen while the app is loading.
- **theme_color**: The color of the browser's toolbar or the status bar when the app is launched.
- **icons**: Specifies a set of icons for the app in various sizes (used for the app icon on a device's home screen or desktop).
- **orientation**: Defines the default orientation for the app (e.g., portrait or landscape).
- Build your React app for production using: **npm run build**
- Serve the build using a simple server like **serve**: **npx serve -s build**

Practical Activity 4.2.1: Configuring web application manifest

Notes to the trainer

- The trainer asks trainees to configure the web application manifest.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to computer lab and configure manifest file.
Step 2: Provide the clear work instructions.
Step 3: Demonstrate how to configure manifest file. While demonstrating, explain the steps to be followed.
Step 4: Ask trainees to configure the manifest file and monitor the activity.
Step 5: Verify whether manifest file is correctly configured.
Step 6: Ask trainees to read the key readings 4.2.1 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 4.2.

Points to Remember

- To configure web application manifest, you do the followings:
    - ✓ Create a manifest.json file in public folder
    - ✓ Add configuration elements such as name, short_name, description, start_url, display, background_color, theme_color, orientation, icons
    - ✓ Place your app icons in the /public/icons/folder.
    - ✓ Include the manifest in index.html
    - ✓ Configure index.html for web app installation
    - ✓ Testing the Manifest

Application of learning 4.2.

ABC is a University located in Kabuga- Kigali city. It has an eLearning website but students complain of typing its long domain name. Sometimes they forget it due to its length. The university has decided to implement PWA in their website. Assume you have been hired to solve the problem. You are tasked to develop a PWA for the university. The platform should offer various courses that users can access on both desktop and mobile devices. To enhance the user experience, you need to create and configure the web application manifest file, ensuring it meets the PWA requirements.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Web application manifest is correctly configured	Manifest file is created		
		Properties values are used		
		Icon is included		
		Manifest file is linked		
		Test in development is performed		
		Test in production is performed		
		Validation is performed		

Indicative content 4.3: Implementation of Service Workers

Duration: 7 hrs

Practical Activity 4.3.1: Implementing service workers

Notes to the trainer

- The trainer asks trainees to implement the service workers.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to implement the service worker in React.JS application.

Step 2: Provide the clear work instructions

Step 3: Demonstrate how to implement the service worker. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to implement the service worker and monitor the activity.

Step 5: Verify whether service worker is well implemented.

Step 6: Ask trainees to read the key readings 4.3.1 in their manual

Step 7: Ask trainees to perform the task provided in the application of learning 4.3

Points to Remember

- To implement service worker, you do the followings:
  - ✓ Create a service worker file in public directory of React Application.
  - ✓ Register the service worker in the index.js of React Application.
  - ✓ Install service worker by using service worker file created.
  - ✓ Implement cache strategies in service worker file created.
  - ✓ Activate the service worker
  - ✓ Update the service worker.

Application of learning 4.3.

X-buy has a retail e-commerce web application, is experiencing performance challenges. Users often complain about slow loading times, especially in areas with poor network connectivity. Assume you are hired as a front-end developer to solve the above problem. You are tasked to improve user experience by implementing service workers to enable caching of critical assets and provide offline functionality.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Service worker is well implemented	Service worker is created		
		Service worker is registered		
		Caching strategy is implemented		
		Service worker is updated		

Learning Outcome 4 End Assessment

Written assessment

Q1. Read carefully the following statements about Progressive Web Application and answer by TRUE if the statement is correct and by FALSE if the statement is incorrect.

a) Leveraging Progressive Enhancement means starting with a basic version of the application and adding features for more capable browsers. **Answer: True**

b) Prioritizing Mobile-First Design involves designing the application primarily for desktop users and then adapting it for mobile devices. **Answer: False**

c) Utilizing Performance Optimization Techniques can include minimizing image sizes, optimizing code, and reducing the number of HTTP requests to enhance the loading speed of a React application. **Answer: True**

d) Progressive Enhancement ignores the capabilities of modern browsers and focuses solely on older browsers to ensure compatibility. **Answer: False**

e) A Mobile-First Design approach typically results in a better user experience on all devices by ensuring that mobile users have a fully functional and optimized interface. **Answer: True**

f) Performance optimization techniques have no impact on the responsiveness of a React application. **Answer**: **False**

g) The web application manifest is a JSON file that provides metadata about the web application. **Answer: True**

h) The manifest file can specify the application's name, icons, and start URL.
Answer: True

i) The display property in the manifest controls how the app appears on the user's home screen. **Answer: True**

j) Testing and validation of the manifest file can be done using the browser's Developer Tools. **Answer: True**

k) The manifest file should be included in the public directory of a React application to be served correctly. **Answer: True**

l) A service worker must be registered before it can be installed. **Answer: True**

m) A service worker will continue to function even if the user closes the browser. **Answer: True**

n) Service workers can only cache static assets; they cannot handle dynamic content. **Answer: False**

Q2. Fill in the blank spaces with appropriate word(s). Select from the given choices in the box.

<link>, index.js, Lazy loading, index.css, Lighthouse, caches.delete(), display, install and <a>

a) A service worker can be registered in a React application by using the navigator.serviceWorker.register method, typically placed in the …………………………… file of the application.

b) You can reference the manifest file by adding a ………………………. tag in the index.html file inside the public folder.

c) The …………………………… event is triggered when the service worker is being installed. This is where caching of static assets and other initial setup can be performed.

d) The ……………………………………………… method is used to remove specific cache entries.

e) You can use tools like the ……………………………………… tool in Chrome DevTools or online manifest validators to ensure the manifest file follows best practices and is PWA-compliant.

f) The ………………………………… property is set to standalone or full screen to make it appear like a native app.

g) ………………………………………………… is a performance optimization technique used in web development, where certain resources (like images, videos, scripts, or components) are loaded only when they are needed rather than during the initial page load.

Answers:

a) index.js
b) <link>
c) Install
d) caches.delete()
e) Lighthouse
f) Display
g) Lazy loading

Q3. Match the column A with Column B

Description(column A)		**Key term(Column B)**
1.When a new service worker is detected, the existing one is replaced.		**A.** Service workers
2.They act as a proxy between the network and the application.		**B**. Prioritize Mobile-First Design
3.Start design with mobile layouts and scale up to larger screens.		**C**.CSS Media Queries

4.Serve appropriate image sizes based on device resolution.		**D**. Lazy loading
5.Use to apply different styles based on device characteristics.		**E**. Responsive images
6.Utilize tools like Lighthouse or Manifest Validator to check your manifest.		**F**. Name
7.Load resources only when they are needed to improve performance		**G**.JSON
8.Property specifies the name of the application in the manifest		**H**. Validating the manifest file with tools
9.Primary format used to define the manifest file		**I**. Updating Service Worker

Answers

1.I

2.A

3.B

4.E

5.C

6.H

7.D

8.F

9.G

Practical assessment

MXN breads is a Bakery Company found in Rwamagana district- Eastern Province –Rwanda. It needs to have an ecommerce website that will enable its customers to buy online. Assume it has hired you as a full stack developer and tasked you to develop a Progressive Web Application (PWA) for a local bakery using React. The bakery wants to provide users with an engaging and fast experience, accessible on both desktop and mobile devices. Your tasks include ensuring that the app leverages progressive enhancement, prioritizes mobile-first design, and utilizes performance optimization techniques. Additionally, you need to configure the manifest file, implement a caching strategy, and manage the service worker lifecycle effectively.

Checklist

SN	Criteria	Indicators	Yes	No
1	Web application responsiveness is correctly maintained	Basic functionality first used		
		Progressive enhancement with advanced features are used.		
		Design for small screens first used		
		Responsive units and breakpoints are selected		
		Performance and loading times are applied		
		Responsive typography is used		
		Code splitting is used		
		Tailwind CSS Purge is used		
		Initial load time is improved		
		Assets are compressed		
		Performance is optimized		
2	Web application manifest is correctly configured	Manifest file is created		
		Properties values are used		
		Icon is included		
		Manifest file is linked		
		Test in development is performed		
		Test in production is performed		
		Validation is performed		
3	Service worker is well implemented	Service worker is created		
		Service worker is registered		
		Caching strategy is implemented		
		Service worker is updated		

END

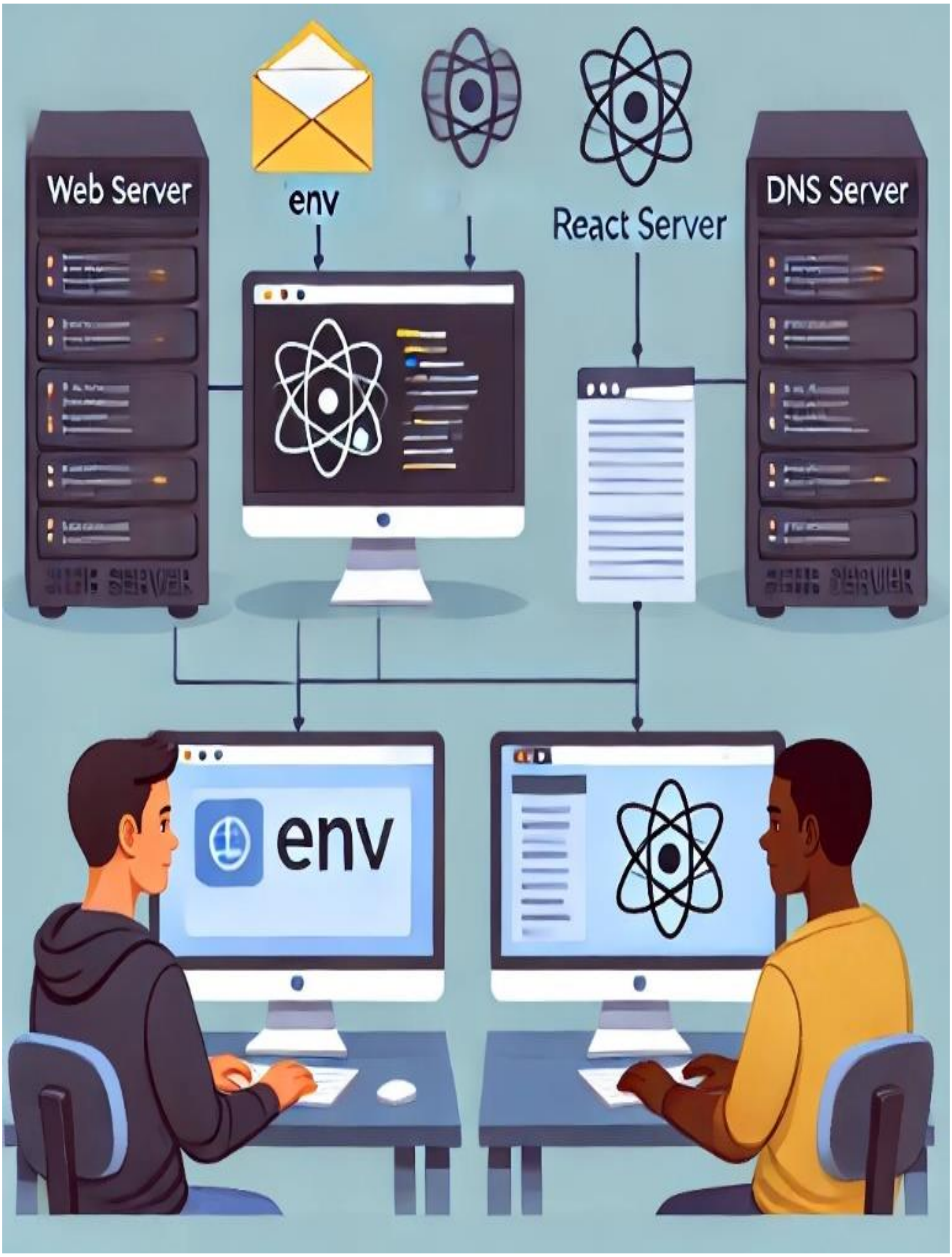
Further information to the trainer

DhiWise. (n.d.). *The ultimate guide to achieving React mobile responsiveness*. DhiWise. https://www.dhiwise.com/post/the-ultimate-guide-to-achieving-react-mobile-responsiveness

Google Developers. (n.d.). *Add a web app manifest*. web.dev. Retrieved October 11, 2024, from https://web.dev/articles/add-manifest

MDN Web Docs. (n.d.). *Progressive web apps (PWAs)*. Mozilla. Retrieved October 11, 2024, from https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps

Uploadcare. (n.d.). *Service workers: A complete guide with examples*. Uploadcare Blog. https://uploadcare.com/blog/service-workers-tutorial/


Web Server
env
React Server
DNS Server
env

Indicative contents

5.1 Configuration of Environment Variables

5.2 Deploying React Application

5.3 Setup Custom Domain

Key Competencies for Learning Outcome 5: Publish the Application

Knowledge	Skills	Attitudes
● Identification of variables ● Description of application files of React application ● Description of DNS	● Configuring the environment variables ● Deploying React application ● Setting up the custom domain	● Being problem-solving oriented during deployment of web application ● Being collaborative while publishing the application ● Being adaptable on different hosting platforms

Duration: 20 hrs

Learning outcome 5 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Identify correctly variables that are used in configuring the environment.
2. Describe clearly the application files that are used in deploying a React application.
3. Describe clearly DNS in publishing a React application
4. Configure correctly the environment variables in publishing the React application.
5. Deploy correctly React application.
6. Setup correctly custom domain in publishing the React application.

Resources

Equipment	**Tools**	**Materials**
● Computer ● DNS server	● Browser ● Terminal	● Internet ● Electricity

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have prepared computer lab with Internet connectivity
- Have browser installed on all computers to be used.
- Have Vercel account.
- Have Github account.
- Have an account to domain registrar
- Have videos to be used as didactic material.
- Have React.JS application developed to be published.

Indicative content 5.1: Configuration of Environment Variables

Duration: 5 hrs

Practical Activity 5.1.1: Configuring environment variables

Notes to the trainer

- The trainer asks trainees to configure the environment variables.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to configure the environment variables.

Step 2: Provide the clear work instructions.

Step 3: Demonstrate how to configure the environment variables. While demonstrating, explain the steps to be followed.

Step 4: Ask trainees to configure the environment variables and monitor the activity.

Step 5: Verify whether environment variables are correctly configured.

Step 6: Ask trainees to read the key readings 5.1.1 in their manual

Step 7: Ask trainees to perform the task provided in the application of learning 5.1

Points to Remember

- To configure the environment variables, you do the followings:
  - ✓ Create .env files
  - ✓ Define variables
  - ✓ Access the variables in the code.
  - ✓ Setup the storage environment.
  - ✓ Use platform-specific options.

Application of learning 5.1.

ABC is a e commerce company located in Kigali city. It has an isolated website which does not interact with payment websites. The company wants a full stack developer to integrate it with other application. Assume you have been hired and tasked with setting up environment variables for a ReactJS application that needs to interact with several backend services and third-party platforms. The project involves connecting to an FTP server to upload files and making API requests to a backend server. The following environment variables are required:

- Backend Host URL
- FTP Host URL
- FTP Username
- FTP Password

Additionally, you need to configure storage options depending on whether the app is running in a development, production, or testing environment. You also need to ensure platform-specific configurations are handled correctly.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Environment variables are appropriately configured	Variables are used		
		Platform options are used		
		.env files are separated		

Indicative content 5.2: Deploying React Application

Duration: 8 hrs

Practical Activity 5.2.1: Deploying React application

Notes to the trainer

- The trainer asks trainees to deploy React application.
- The use of videos as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to deploy the React application.
Step 2: Provide the clear work instructions.
Step 3: Demonstrate how to deploy React application. While demonstrating, explain the steps to be followed.
Step 4: Ask trainees to deploy React application and monitor the activity.
Step 5: Verify whether React application is properly deployed.
Step 6: Ask trainees to read the key readings 5.2.1 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 5.2

Points to Remember

- To deploy React application, you do the followings:
  - ✓ Create build folder.
  - ✓ Configure the deployment platform
  - ✓ Migrate the necessary files
  - ✓ Test the React application deployed.

Application of learning 5.2.

ABC company is a company located in Kigali City. It hired a React.js developer to develop a customer feedback system. The company has the application on its local machines and needed to deploy it for users to access. Assume you are hired by the company and tasked to deploy a React application that has been developed and is ready for production. The deployment will be done using Vercel as the hosting platform. You will be responsible for building the project, configuring the deployment platform, migrating the files, and testing the deployed application to ensure it is running smoothly.

Checklist:

SN	Criteria	Indicators	Yes	No
1	React application is well deployed	Build Script is created		
		Vercel is selected		
		Vercel account is accessed		
		New project is created		
		Repository is imported		
		Project settings are configured		
		Deployment is performed		
		Build folder is uploaded		
		Server is configured		
		Deployed application is accessed		

Indicative content 5.3: Setup Custom Domain

Duration: 7 hrs

Theoretical Activity 5.3.1: Description of DNS

Notes to the trainer:

- While delivering this content small groups can be used to describe the domain name.
- Avail videos to be used as didactic material.

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Write in full words DNS.
ii. What is the role of DNS?
iii. Give five examples of domain names.
iv. Explain the hierarchy and structure of DNS.
v. Explain the name resolution process performed by DNS.

Step 2: Ask trainees to write their findings on paper or flipchart

Step 3: Engage trainees to present their findings

Step 4: Provide an expert view and clarify the ideas

Step 5: Address any question or concern raised by the trainees.

Step 6: Ask the trainees to read key readings 5.3.1 in their manual.

Points to Remember

- **DNS** (Domain Name System) translates human-readable domain names into IP addresses that computers use to identify each other on the network.
- **Root Level:** The top of the hierarchy, represented by the dot (.) at the end of the domain (often implied).
- **TLD (Top-Level Domain):** The first part after the root, such as .com, .org, .net, etc.
- **Second-Level Domain:** The actual domain name chosen by the user, such as example in example.com.
- **Subdomains:** Optional parts of the domain that come before the second-level domain, such as www in www.example.com.

Practical Activity 5.3.2: Setting Up Custom Domain

Notes to the trainer

- While delivering this content small groups can be used to set up custom domain.
- Avail videos to be used as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to go to the computer lab to set up the custom domain.
Step 2: Provide the clear work instructions.
Step 3: Demonstrate how to set up custom domain. While demonstrating, explain the steps to be followed.
Step 4: Ask trainees to setup the custom domain and monitor the activity.
Step 5: Verify whether custom domain is properly set.
Step 6: Ask trainees to read the key readings 5.3.2 in their manual
Step 7: Ask trainees to perform the task provided in the application of learning 5.3

Points to Remember

- To set up custom domain, you do the followings:
  - ✓ Link the custom domain to the React.js application
  - ✓ Configure DNS settings
  - ✓ Configure the SSL
  - ✓ Perform testing and verification

Application of learning 5.3.

ABC company is a company located in Kigali City. It hired a React.js developer to develop a customer feedback system. The company has the application on its local machines and needed to deploy it for users to access. Assume you are hired by the company and tasked to configure it for a custom domain with DNS and SSL settings. Your client, ABC Technologies, has purchased the domain www.abctech.com. You are using Vercel to host the React app.

Your task includes the following:

1. Set up the custom domain www.abctech.com for the React application.
2. Configure the DNS settings correctly for the domain.
3. Enable SSL to ensure secure communication (HTTPS).
4. Verify that the domain is correctly configured, secured, and that the application is accessible via https://www.abctech.com.

Checklist:

SN	Criteria	Indicators	Yes	No
1	Custom domain is correctly set	www.abctech.com is added		
		DNS records are updated		
		SSL Configuration is performed		
		Domain Resolution is checked		
		SSL Configuration is checked		
		Application is accessed		

Learning Outcome 5 End Assessment

Written assessment

Q1. Read carefully the following statements about publish the application and answer by TRUE if the statement is correct and by FALSE if the statement is incorrect.

a) Environment variables in a React application can be used to store sensitive information such as backend host, FTP host, FTP username, and FTP password. **Answer: True**

b) Changing the value of an environment variable requires restarting the development server for the changes to take effect. **Answer: True**

c) Vercel supports automatic deployments from a Git repository, allowing for continuous integration. **Answer: True**

d) When migrating application files to a deployment platform, it's unnecessary to include configuration files like package.json. **Answer: False**

e) React applications can use platform-specific options for environment variable configurations. **Answer: True**

f) Testing the deployed application is important to ensure that it works as expected in the production environment. **Answer: True**

g) You can directly access environment variables in React without prefixing them with REACT_APP_. **Answer: False**

h) It is best practice to keep all environment variables in a single .env file for security purposes. **Answer: False**

i) After configuring DNS settings, it is unnecessary to wait for DNS propagation to take effect. **Answer**: **False**

j) You can create separate .env files for different environments (e.g., .env.development, .env.production) in a React application. **Answer: True**

k) Migrating application files involves copying only the src folder from your React application. **Answer: False**

l) After deploying a React application, you should avoid running any performance audits to ensure the app functions correctly. **Answer: False**

m) Verifying SSL settings can be done by accessing the website and checking if the URL starts with "http://" instead of "https://". **Answer**: **False**

Q2. Fill in the blank spaces with appropriate word(s). Select from the given choices in the box.

dot(.), CNAME record, .env, nslookup, REACT_APP_,A Record, npm run build, comma(,) and IP address

a) The command …………………………… is used to create a production-ready build of a React application.

b) You can set environment variables in ReactJS by creating a ……………………….. file in the root of your project.

c) Add an …………………………………….. to point your domain to the IP address of the server hosting your React application.

d) Variables should be prefixed with ……………………………………… for React to recognize them

e) Use the command ………………………………to check if your domain resolves to the correct IP address.

f) The ……………………………. is returned to the browser, which uses it to connect to the server.

g) The top of the DNS hierarchy represented by a ………………………………. symbol.

Answers:

a) npm run build
b) .env
c) A Record
d) REACT_APP_
e) nslookup
f) IP address
g) dot(.)

Q3. Match term in middle column and its corresponding meaning in right column and then write the answer in the left column.

Answer	Key term	Description
1……..	1. DNS	A. It is an extension of HTTP that adds an extra layer of security.
2……..	2. SSL	B. It is a type of DNS (Domain Name System) record that maps an alias name to a true or canonical domain name.

3……..	3. HTTPS	C. It is the last part of a domain name, appearing after the final dot. It helps indicate the nature or origin of the domain.
4………	4. FTP host	D. The top of the hierarchy, represented by the dot (.) at the end of the domain.
5………	5.CNAME Record	E. It is a system that translates human-readable domain names into machine-readable IP addresses that computers use to identify each other on the network.
6……….	6.Top-Level Domain	F. It the server or system that provides the File Transfer Protocol service, allowing users to upload, download, and manage files over the internet.
7………	7.Root Level	G. It is a company that provides individuals and organizations access to the internet.
8………	8. IP address	H. It is a standard security technology used to establish an encrypted link between a server and a client, typically a web server (website) and a browser.
9………	9. ISP	I. It is the unique identifier for the account.
10…….	10. Domain registrar	J. It is a type of DNS (Domain Name System) record that maps a domain name to its corresponding IPv4 address.
		K. It is a company or organization that manages the reservation of Internet domain names.
		L. It is a unique identifier assigned to devices (such as computers, smartphones, servers) connected to a network that uses the Internet Protocol for communication.

Answers:

1.E

2.H

3.A

4.F

5.B

6.C

7.D

8.L

9.G

10.K

Practical assessment

ABC company is a company located in Kigali City. It hired a React.js developer to develop a customer feedback system. The company wants to hire a ReactJS developer to deploy the application on Vercel. The client has provided a custom domain, www.abc.com, and they want the application to be accessible via this domain. Assume you are hired as a ReactJS developer and tasked to make sure application to have SSL (HTTPS) enabled for secure access. Your tasks are to:

1. Configure the DNS settings to point the custom domain to the Vercel deployment.
2. Set up SSL to ensure that the site can be accessed via HTTPS.
3. Test and verify that the domain is properly configured and secured.

Checklist

SN	Criteria	Indicators	Yes	No
1	Environment variables are appropriately configured	Variables are used		
		Platform options are used		
		.env files are separated		
2	React application is well deployed	Build Script is created		
		Vercel is selected		
		Vercel account is accessed		
		New project is created		
		Repository is imported		
		Project settings are configured		

		Deployment is performed		
		Build folder is uploaded		
		Server is configured		
		Deployed application is accessed		
3	Custom domain is correctly set	www.abc.com is added		
		DNS records are updated		
		SSL Configuration is performed		
		Domain Resolution is checked		
		SSL Configuration is checked		
		Application is accessed		

Further information to the trainer

Appwrk. (n.d.). *How to set up the .env file in ReactJS*. Retrieved October 14, 2024, from https://appwrk.com/reactjs-environment-variables#h-how-to-set-up-the-env-file-reactjs

Create React App. (n.d.). *Deployment*. https://create-react-app.dev/docs/deployment/

Vercel. (n.d.). *Deploying a React application with Vercel*. Vercel. https://vercel.com/guides/deploying-react-with-vercel

Vercel. (n.d.). *Add a domain*. Vercel. https://vercel.com/docs/projects/domains/add-a-domain.

RTB | RWANDA TVET BOARD

October, 2024