



RQF LEVEL 4



POSTMAN



AXIOS Axios



Mocha



Chai



SWDBD401 SOFTWARE DEVELOPMENT

Backend Application Development

TRAINER'S MANUAL

October, 2024



BACKEND APPLICATION DEVELOPMENT

AUTHOR'S NOTE PAGE (COPYRIGHT)

The competent development body of this manual is Rwanda TVET Board ©, reproduce with permission.

All rights reserved.

- This work has been produced initially with the Rwanda TVET Board with the support from KOICA through TQUM Project.
- This work has copyright, but permission is given to all the Administrative and Academic Staff of the RTB and TVET Schools to make copies by photocopying or other duplicating processes for use at their own workplaces.
- This permission does not extend to making of copies for use outside the immediate environment for which they are made, nor making copies for hire or resale to third parties.
- The views expressed in this version of the work do not necessarily represent the views of RTB. The competent body does not give warranty nor accept any liability.
- RTB owns the copyright to the trainee and trainer's manuals. Training providers may reproduce these training manuals in part or in full for training purposes only. Acknowledgment of RTB copyright must be included on any reproductions. Any other use of the manuals must be referred to the RTB.

© **Rwanda TVET Board**

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:

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 IV in Software Development, specifically for the module **"SWDBD401: Backend Application Development"**

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



Under Financial and Technical support of



COORDINATION TEAM

RWAMASIRABO Aimable

MARIA Bernadette M. Ramos

MUTIJIMA Asher Emmanuel

PRODUCTION TEAM

Authoring and Review

SEKABANZA Jean de la Paix

BUGIRUWENDA Jean Bosco

Validation

MUNYEMANA Alexis

ISHIMWE Gilbert

Conception, Adaptation and Editorial works

HATEGEKIMANA Olivier

GANZA Jean Francois Regis

HARELIMANA Wilson

NZABIRINDA Aimable

DUKUZIMANA Therese

NIYONKURU Sylvestre

NTEZIMANA Aimable

Formatting, Graphics, Illustrations, and infographics

YEONWOO Choe

SUA Lim

SAEM Lee

SOYEON Kim

WONYEONG Jeong

MINANI Aloys

Financial and Technical support

KOICA through TQUM Project

TABLE OF CONTENT

AUTHOR'S NOTE PAGE (COPYRIGHT)-----	iii
ACKNOWLEDGEMENTS-----	v
TABLE OF CONTENT -----	viii
ACRONYMS-----	x
INTRODUCTION -----	1
MODULE CODE AND TITLE: SWDBD401 BACKEND APPLICATION DEVELOPMENT-----	2
Learning Outcome 1: Develop RESTFUL APIs with Node JS -----	3
Key Competencies for Learning Outcome 1: Develop RESTFUL APIs with Node JS -----	4
Indicative content 1.1: Set up Node. Js Environnent -----	7
Indicative content 1.2: Connection of Node Js to the ES5 or ES6 Server -----	14
Indicative content 1.3: Establishment of Database Connection -----	19
Indicative content 1.4: Develop RESTFUL APIs-----	21
Learning outcome 1 end assessment -----	32
Further information to the trainer-----	40
Learning Outcome 2: Secure Backend Application-----	41
Key Competencies for Learning Outcome 2: Secure Backend Application-----	42
Indicative content 2.1: Data Encryption in Securing RESTFUL APIs -----	45
Indicative content 2.2: Integrating and Using Third-Party Libraries -----	48
Indicative content 2.3: Maintaining and Updating Third-Party Libraries-----	52
Indicative content 2.4: Implementation of Authentication-----	57
Indicative content 2.5: Implementation of Authorization-----	62
Indicative content 2.6: Implementation of Accountability-----	68
Indicative content 2.7: Secure Environment Variables -----	73
Indicative content 2.8: Monitor and Manage Environment Variables -----	79
Learning outcome 2 end assessment -----	83
Further information to the trainer-----	91
Learning Outcome 3: Test Backend Application -----	92
Key Competencies for Learning Outcome 3: Test Backend Application -----	93
Indicative content 3.1: Implementation of Unit Testing-----	96
Indicative content 3.2: Implementation of Usability Testing-----	101
Indicative content 3.3: Implementation of Security Testing-----	106

Learning outcome 3 end assessment -----	115
Further information to the trainer-----	123
Learning Outcome 4: Manage Backend Application. -----	124
Key Competencies for Learning Outcome 4: Manage Backend Application-----	125
Indicative content 4.1: Preparation of Deployment Environment -----	128
Indicative content 4.2: Implementation of Manual Deployment -----	131
Indicative content 4.3: Maintenance of NodeJS Application -----	133
Indicative content 4.4: Application of NodeJS Documentation Tools and Frameworks -----	138
Learning outcome 4 end assessment -----	142
Further information to the trainer-----	148

ACRONYMS

AAA: Authentication, Authorization and Accountability

ABAC: Attribute-Based Access Control

AES: Advanced Encryption Standard

APIs: Application Programming Interfaces

AWS: Amazon Web services

CI/CD: Continuous Integration/Continuous Deployment

GDPR: General Data Protection Regulation

HIPAA: Health Insurance Portability and Accountability Act

HTTPS: Hypertext Transfer Protocol Secure

JWT: json web token

KOICA: Korea International Cooperation Agency

NPM: Node package manager

OWASP: Open Worldwide Application Security Project

RBAC: Role-based access control

RSA: Rivest-Shamir-Adleman

RTB: Rwanda TVET Board

SSL: Secure Socket Layer

SSO: Single Sign-On

TLS: Transport Layer Security

TQUM Project: TVET Quality Management Project

TVET: Technical Vocational and Education Training

UAC: Universal Access Control

INTRODUCTION

This trainer's manual includes all the methodologies required to effectively deliver the module titled "**Backend Application Development.**" Trainees 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 labor 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 trainees review the key reading and the points to remember section.

MODULE CODE AND TITLE: SWDBD401 BACKEND APPLICATION DEVELOPMENT

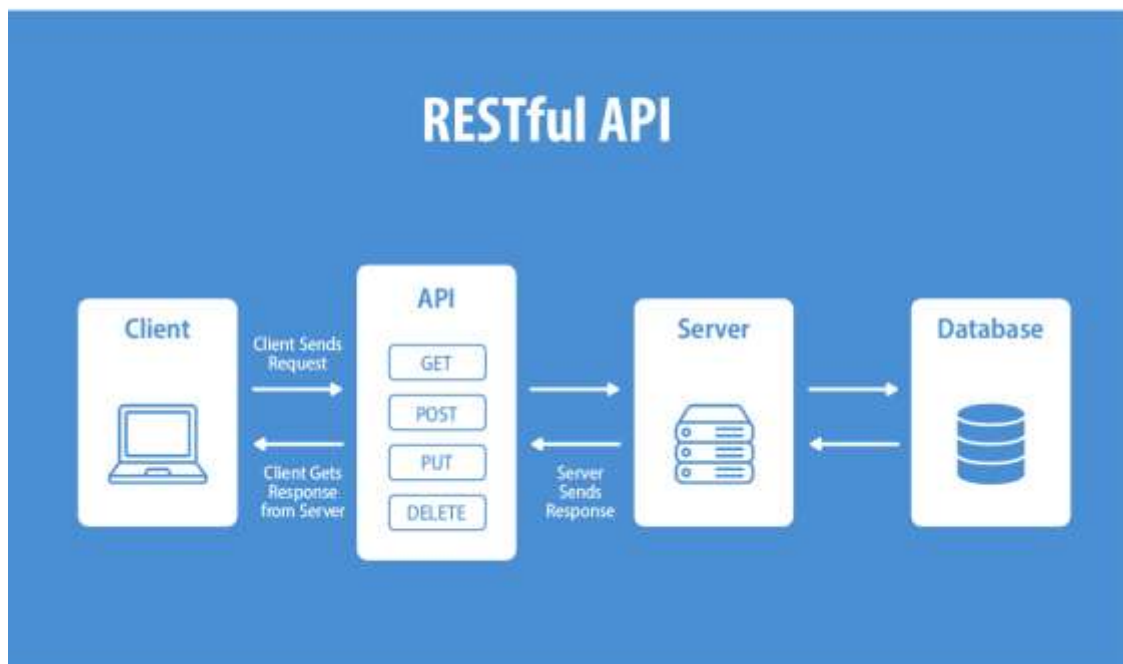
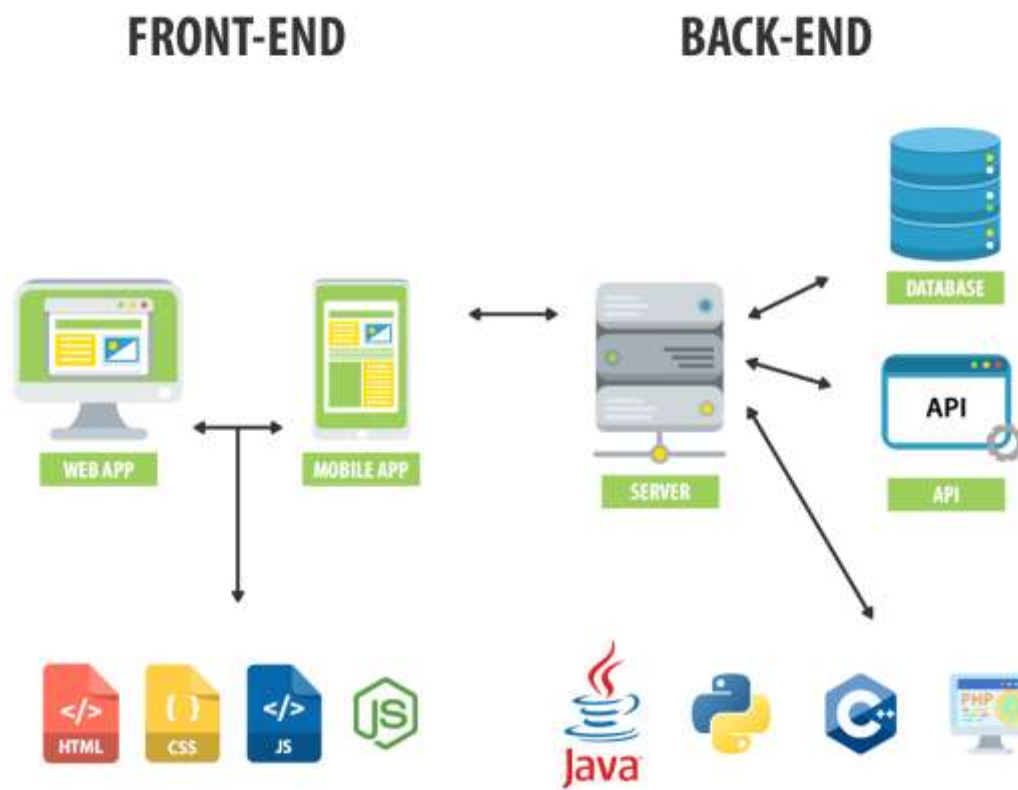
Learning Outcome 1: Develop RESTFUL APIs with Node JS

Learning Outcome 2: Secure Backend Application

Learning Outcome 3: Test Backend Application

Learning Outcome 4: Manage Backend Application

Learning Outcome 1: Develop RESTFUL APIs with Node JS



Indicative contents

1.1 Setup Node. Js Environment

1.2 Connection of Node Js to the ES5 or ES6 server

1.3 Establishment of database connection

1.4 Develop RESTFUL APIs

Key Competencies for Learning Outcome 1: Develop RESTFUL APIs with Node JS

Knowledge	Skills	Attitudes
● Description of Node.js Key Concepts ● Description of APIs ● Description of database management system	● Installing Node Js Modules and packages ● Connecting node Js to the ES5 or ES6 server ● Configuring MySQL server ● Debugging endpoints/APIs ● Installing postman ● Establishing database connection ● Developing RESTFUL APIs ● Using Middleware services ● Implementing CRUD operations using MySQL Database ● Using HTTP Status code	● Having Team work ability ● Being critical thinker ● Being Innovative ● Being creative ● Being honesty ● Having Passion for Learning ● Having Problem-Solving Mindset ● Having good Collaboration and Communication ● Being Attentive to Security ● Having Ethical Coding



Duration: 45 hrs



Learning outcome 1 objectives:

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

1. Describe correctly Node.js Key Concepts as applied in software development
2. Describe correctly database management system as they are used in backend development.
3. Install and configure properly Postman as it is used in Backend development.
4. Installing and configuring MySQL Server properly as is used in backend development.
5. Describe correctly the term API as used in software development.
6. Install correctly Node Js Modules and packages as used in backend application development
7. Connect properly node Js to the ES5 or ES6 server as applied in the server configuration
8. Develop correctly RESTFUL APIs as applied in the software development process.
9. Establish correctly database connection as applied in backend system development.
10. Use correctly Middleware services as applied in software development.
11. Perform properly CRUD operations using MySQL Database as used in the database connection.
12. Use correctly HTTP Status code correctly as required in error handling in software development



Resources

Equipment	Tools	Materials
• Computer	• Browser • NodeJS • Postman • MySQL Workbench	• Internet • Electricity



Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have prepared computer lab with internet connectivity
- Have computer with MySQL server installed
- Have visual studio code installed on all computers to be used
- Have videos to be used as didactic material.



Indicative content 1.1: Set up Node. Js Environnent



Duration: 10 hrs



Theoretical Activity 1.1.1: Description of Node.js Key Concepts



Notes to the trainer:

- While delivering this content, a small group can be used for describing Node.js Key Concepts.
- Avail sample videos to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following questions:

i. Define the following terms:

- a) Node js
- b) Npm
- c) Dependencies
- d) Backend application
- e) Routes
- f) express js
- g) Nodemon

ii. Differentiate the following terms used in function:

- a) Class
- b) Object
- c) Method
- d) Properties

iii. Describe postman as a tool used in backend system development.

- iv. Describe APIs as used in backend system development.
- v. Describe database management system.

- Step 3:** Involve trainees in presentation of their findings.
- Step 4:** Provides expert view and clarifies ideas by using didactic materials.
- Step 5:** Address any questions or concerns.
- Step 6:** Ask trainees to read the key reading 1.1.1 in the trainee manual.



Points to Remember

- Node.js is an open-source JavaScript runtime environment that allows developers to execute JavaScript code on the server side. Once you install node js it come up with it default package manager (NPM) used to install necessary packages and dependencies like express js and Nodemon.
- In development of backend application, you have to include concepts of object-oriented programming (OOP) like class, object, methods and properties.
- Backend deals with server side it has to be connected to database such as SQL based or NoSQL based one.
- The backend application has to respond to the HTTP methods via routes and once connecting to frontend there is the application of APIs and those ones are tested and documented by using postman.



Practical Activity 1.1.2: Installation of Node Js Modules and packages



Notes to the trainer

- This activity should take place in a Computer Lab.
- While delivering this content, you are required to:
 - ✓ Avail computer with VS code installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are asked to go to the computer lab to perform the followings: install node js, check if node and npm are installed, make directory as project name, navigate to created directory, initialise npm, install express, Nodemon and open the created project in VS Code.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to install node js, check if node and npm are installed, make directory as project name, navigate to created directory, initialise npm, install express, Nodemon and open the created project in VS Code.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 1.1.2.



Points to Remember

- Installation of Node Js Modules and packages you follow the following starting steps:
- Install node js
- Check if node and npm are installed
- Make directory
- Navigate to created directory using cd command
- Initialise npm
- Install express and Nodemon
- Open the created project in vs code.



Practical Activity 1.1.3: Installation and configuration of Postman



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should install and configure postman.

- While delivering this content, you are required to:
 - ✓ Avail computer lab with node js modules and packages installed.
 - ✓ Avail network connection for trainees or postman installer file on external device.
 - ✓ Avail video as didactic materials



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees do the task described below:

As a software developer, you are asked to go to the computer lab to install postman and configure basic settings.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to install postman. While demonstrating, explain the steps to install postman and configure basic settings.

Step 4: Asks trainees to install postman and configure basic settings and monitor the procedures.

Step 5: Verify whether postman is installed and configured.

Step 6: Ask trainees to read key reading 1.1.3.



Points to Remember

- To install and configure postman you follow the following steps:
- Download the Installer from official website or use it installer file on external device
- Run the Installer
- Launch Postman
- Configure the basic settings like user account (Email, Username and password) or sign up using google or sign in with SSO.



Practical Activity 1.1.4: Configuration of MySQL Server



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should configure MySQL Server.
- While delivering this content, you are required to:
 - ✓ Avail computer with MySQL installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go to the computer lab to configure MySQL Server.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to configure MySQL Server. While demonstrating, explain the steps to follow while configuring MySQL Server.

Step 4: Asks trainees to configure MySQL and monitor the procedures.

Step 5: Verify whether the configuration of MySQL Server is clearly performed.

Step 6: Ask trainees to read key reading 1.1.4.



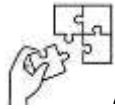
Points to Remember

- To configure MySQL Server in windows you follow those steps:
 - ✓ Set up the server configuration details like:
 - ✚ Config Type
 - ✚ Connectivity
 - ✚ Authentication

 Accounts and Roles

 Windows Service

- Apply Configuration
- Complete the Installation
- Verify Installation



Application of learning 1.1.

STU LTD is a software development company located in Musanze district, it develops software both frontend and backend for different companies, due to different activities that they have, you are hired as backend developer in node js responsible for setting the working environment by creating the project folder (**OurWebinfo**) on local disk D, and set all other necessary settings in order to be ready for development of back end in node js, testing and documenting APIs.

All tools, materials and equipment will be provided by the company.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Node Js Modules and packages are well installed	1.1Node js is Installed		
		1.2Node and npm version are verified		
		1.3Project name is created		
		1.4The project folder have been navigated using cd command		
		1.5Npm is initialised		
		1.6Express js is Installed		
		1.7Nodemon is installed		
		1.8The project is opened in vs code		
2	Postman is well installed and configured	2.1 Installer file is downloaded or accessed on external drive		
		2.2 Run the Installer		
		2.3 Launch Postman		
		2.4 The account is created or signed in using google account		

		2.5 Postman access by using lightweight API client is selected		
		2.6 Postman desktop App is displayed		
3	MySQL Server is well configured	3.1 Standalone MySQL Server option is selected		
		3.2 Port number is selected		
		3.3 TCP/IP is enabled		
		3.4 Authentication method is selected (Use strong password)		
		3.5 User account and roles are created		
		3.6 MySQL Server is tested in Command line interface		



Indicative content 1.2: Connection of Node Js to the ES5 or ES6 Server



Duration:10 hrs



Theoretical Activity 1.2.1: Description of client libraries and ES5 or ES6



Notes to the trainer:

- While delivering this content, small groups can be used for describing client libraries and ES5 or ES6.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to Discuss on the followings:

- i. Categories and Usability of Node.js Client Libraries
- ii. Difference between ES5 from ES6

Step 3: Involve trainees in presentation of their findings.

Step 4: Provide expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 1.2.1 in the trainee manual.



Points to Remember

- Node.js has a vast ecosystem of client libraries available through NPM (Node Package Manager). These libraries cover a wide range of functionalities, from HTTP requests to database access, and even utility functions.
- The number of libraries is enormous, with over a million packages available on NPM, and new ones being added regularly.

- **ES5:** Uses ``var`` for variable declarations, which has function scope.
- **ES6:** Introduces ``let`` and ``const``. ``let`` has block scope, while ``const`` is used for constants that cannot be reassigned.



Practical Activity 1.2.2: Creation of basic server with Express Js



Notes to the trainer

- This activity should take place in the computer lab where trainees should create basic server with express.js.
- While delivering this content, you are required to:
- Avail computers with node js and visual studio code installed.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to create basic server with express js.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to create basic server with express js. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to create basic server with express js and monitor the procedures.

Step 5: Verify whether create basic server with express js have been performed.

Step 6: Ask trainees to read key reading 1.2.2.



Points to Remember

- Creating a basic server with Express.js involves different steps.
- Here are steps to Create a Basic Server with Express.js
- Install Node.js and NPM
- Initialize a New Node.js Project

- Install Express.js
- Create the Server File
- Write Basic Express.js Server Code
- Run the Server
- Test the Server



Practical Activity 1.2.3: Application of Client Libraries



Notes to the trainer

- This activity should take place in the computer lab where trainees should apply client libraries.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees do the task described below:

As a software developer, you are asked to go to the computer lab to apply client libraries in existing project developed on previous practical activity.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to apply client libraries in existing project. While demonstrating, explain the steps to apply client libraries in existing project.

Step 4: Asks learners to apply client libraries in existing project and monitor the procedures.

Step 5: Verify whether application of client libraries in existing project have been performed.

Step 6: Ask trainees to read key reading 1.2.3



Points to Remember

- There are different libraries in express js but as developer you have to apply them depending on the project structure. Here are the basic ones:
 - ✓ Use the built-in ``http`` and ``https`` modules for basic requests.
 - ✓ Use **Axios** for a more modern and flexible approach to making HTTP requests.
 - ✓ The **Request** library is deprecated, but you may encounter it in older projects.



Practical Activity 1.2.4: Establishment and Test of server connection



Notes to the trainer

- This activity should take place in the computer lab where trainees should establish and test server connection.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As a backend developer, you are asked to go to the computer lab to establish and test server connection.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to establish and test server connection. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to establish and test server connection and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.2.4



Points to Remember

- While establishing and testing the server you have to follow those steps.
 1. Set up your Node.js environment and create a new project.
 2. Create a simple server using the built-in `http` module.
 3. Run the server and test the connection using a browser, cURL, Postman, or Axios.
 4. Handle any errors and debug as necessary.



Application of learning 1.2.

STU LTD is a software development company located in Musanze district, it develops software both frontend and backend for different companies, due to different activities that they have, you are hired as backend developer in node js responsible for creating the project folder (**OurWebinfo**) on local disk D and establish server file that will be used as entry point including http and axios as client libraries.

All tools, materials and equipment will be provided by the company.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Basic server with Express.js is correctly created	1.1. Node.js and NPM are installed		
		1.2 New Node.js Project is initialised		
		1.3 Express.js is installed		
		1.4 Server File is created		
		1.5 Express.js Server Code are written		
		1.6 Server file is runned		
		1.7 Server file is tested		
		1.8 Errors and debug are handled		
		1.9 Axios client library is installed		



Indicative content 1.3: Establishment of Database Connection



Duration: 10 hrs



Practical Activity 1.3.1: Establishment of database connection



Notes to the trainer

- This activity should take place in the computer lab where trainees should establish database connection.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As a backend developer, you are asked to go to the computer lab to establish database connection in node js.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to establish database connection in node js. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to establish database connection and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.3.1.



Points to Remember

- In Establishment of database connection, you have to follow those steps:
- Install MySQL package: Use `npm install mysql2`.
- Create a database: Use SQL commands in a MySQL client.
- Set up a schema: Create tables using SQL commands.
- Configure database connection: Use `mysql2` to set up a connection in a Node.js file.
- Test the connection: Run a simple query to ensure the connection works.



Application of learning 1.3.

Tela Tech Ltd is a company that develop websites for different institutions they have tasked you to develop for them a database (**Kigali innovation DB**) and inside create a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and set the basic connections that will be used while creating APIs.

Checklist:

SN	Elements	Indicators	Observation	
			Yes	No
1	Establishment of database connection is well performed	1.1 MySQL package is installed		
		1.2 Database is created		
		1.3 Schema is settled up (Required table is created)		
		1.4 Database connection is configured		
		1.5 the connection is tested		



Indicative content 1.4: Develop RESTFUL APIs



Duration: 15 hrs



Practical Activity 1.4.1: Develop endpoints and HTTP Methods



Notes to the trainer

- This activity should take place in the computer lab where trainees should establish database connection.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer who has skills of database development you are requested to go to the computer lab to define endpoints and HTTP Methods

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to define endpoints and HTTP Methods in node js. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to define endpoints and HTTP Methods and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.4.1.



Points to Remember

- Defining endpoints and HTTP methods using Express.js in Node.js. you follow different steps.

Step 1: Set Up Your Express Server

Step 2: Define Endpoints

Step 3: Testing the Endpoints



Practical Activity 1.4.2: Testing endpoints using postman



Notes to the trainer

- This activity should take place in the computer lab where trainees should test endpoints using postman.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to test endpoints using postman.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to test endpoints using postman. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to test endpoints using postman and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.4.2.



Points to Remember

- Using Postman, you can easily test all the endpoints of your Express.js application by performing the following actions:
 1. Check if postman is installed
 2. Start your express server
 3. Open postman
 4. Test the endpoints
 5. Check response



Practical Activity 1.4.3: Implementation of API endpoints



Notes to the trainer

- This activity should take place in the computer lab where trainees should implement API endpoints.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to implement API endpoints.

Step 2: Explain the task and provide clear work instruction

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

Step 4: Asks learners to implement API endpoints and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.4.3



Points to Remember

- Implementing API endpoints in Node.js using Express involves several steps, from setting up your environment to writing the code for your endpoints.



Theoretical Activity 1.4.4: Description of middleware services



Notes to the trainer:

- While delivering this content, small groups can be used for describing middleware services.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following questions:

- Explain the term middleware.
- Describe the use of middleware services
- Describe types of middleware services

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 1.4.4 in the trainee manual.



Points to Remember

- Middleware services in Express.js are functions that execute during the request-response cycle.
- They can modify the request and response objects, end the request-response cycle, and call the next middleware function in the stack.



Practical Activity 1.4.5: Use middleware services



Notes to the trainer

- This activity should take place in the computer lab where trainees should use middleware services.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab use middleware services.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to use middleware services. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to use middleware services and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.4.5.



Points to Remember

- In node js we have different types of middleware services but that are used depending on the application to be developed, among them we can use the followings:
 - ✚ **Logging Middleware** records the details of the incoming request.
 - ✚ **Input Validation Middleware** checks the data in the request and ensures it meets the necessary requirements.
 - ✚ If an error occurs (e.g., due to invalid input or server issues),
 - ✚ **Error Handling Middleware** catches the error and sends an appropriate response to the client.



Practical Activity 1.4.6: Perform CRUD operations using MySQL Database



Notes to the trainer

- This activity should take place in the computer lab where trainees should perform CRUD operations using MySQL Database.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As a backend developer, you are requested to go to the computer lab to perform CRUD operations using MySQL Database.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to perform CRUD operations using MySQL Database. While demonstrating, explain the steps to be followed.

- Step 4:** Asks learners to perform CRUD operations using MySQL Database and monitor the procedures.
- Step 5:** Verify whether the task have been performed.
- Step 6:** Ask trainees to read key reading 1.4.6.



Points to Remember

- To perform CRUD operation in node js you have to follow the following steps
 - ✓ Setup Your Node.js Environment
 - ✓ Initialize a Node.js Project
 - ✓ Install Required Packages:
 - ✓ Create a MySQL Connection
 - ✓ Set Up Express Server that contains all operations included in CRUD (Create, Read, Update and Delete as endpoints)
 - ✓ Run Your Application
 - ✓ Test the CRUD operations using tools like Postman.



Practical Activity 1.4.7: Use HTTP Status code



Notes to the trainer

- This activity should take place in the computer lab where trainees should use HTTP status code.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask learners do the task described below:

As a backend developer you are requested to go to the computer lab to use HTTP status code in an existing project developed on activity 1.4.6.

- Step 2:** Explain the task and provide clear work instruction
- Step 3:** Demonstrate how to use HTTP status code. While demonstrating, explain the steps to be followed.
- Step 4:** Asks learners to use HTTP status code and monitor the procedures.
- Step 5:** Verify whether the task have been performed.
- Step 6:** Ask trainees to read key reading 1.4.7.



Points to Remember

- HTTP Status Codes can be classified as the followings:
 - ✓ **201 Created:** Used when a new resource (developer) is successfully created.
 - ✓ **200 OK:** Used for successful requests that return data or indicate successful updates/deletions.
 - ✓ **404 Not Found:** Used when a requested resource (developer by ID) does not exist in the database.
 - ✓ **500 Internal Server Error:** Used when there is a server-side error, such as a database query failing.



Practical Activity 1.4.8: Debugging RESTFUL APIs



Notes to the trainer

- This activity should take place in the computer lab where trainees should debug RESTFUL APIs.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail computer with xampp or MySQL server installed
 - ✓ Avail sample project to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend, developer you are requested to go to the computer lab to debug RESTFUL APIs existing project developed on activity 1.4.7.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to debug RESTFUL APIs. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to debug RESTFUL APIs and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 1.4.8.



Points to Remember

- Debugging a RESTful API can be an essential part of development, ensuring that your API behaves as expected. Here's a step-by-step guide to help you debug a RESTful API effectively:
 1. Set up your development environment
 2. Implement logging.
 3. Use a debugger to set breakpoints and inspect your code.
 4. Test API endpoints with tools like Postman.
 5. Validate input data
 6. Monitor network traffic for issues.
 7. Review and update your API documentation.
 8. Use automated tests for your API endpoints.
 9. Analyse performance to identify bottlenecks.



Application of learning 1.4.

Tela Tech Ltd is a company that develop websites for different institutions they have tasked you to develop for them a database (**Kigali innovation DB**) and inside create a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and develop APIs that will be used to insert, update, select and delete client's records via frontend where you will be provide the developed APIs to be integrated with frontend. Include http status codes in order to overcome error handling issues.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Development of endpoints and HTTP Methods is effectively done	1.1 Set Up Your Express Server is performed		
		1.2 Define Endpoint is performed		
		1.3 Endpoints are tested		
2	Endpoints using postman are well tested	2.1Node js is Installed		
		2.2Node and npm version are verified		
		2.3Project name is created		
		2.4The project folder have been navigated		
		2.5cd command is used		
		2.6Npm is initialised		
		2.7Express js is Installed		
		2.8Nodemon is installed		
		The project is opened in vs code		
3	Middleware services are correctly used	3.1Logging Middleware have been used		
		3.2Input Validation Middleware have been used		
		3.3Error Handling middleware have been used		

4	CRUD operations using MySQL Database is well performed	4.1 Required Packages are Installed		
		4.2 MySQL Connection is created		
		4.3 CREATE endpoints is created		
		4.4 Read endpoints is created		
		4.5 Update endpoints is created		
		4.6 Delete endpoints is created		
		4.7 CRUD operations is tested using Postman.		
5	HTTP Status code is well used	5.1 201 have been Created		
		5.2 200 OK is created		
		5.3 404 Not Found is used		
		5.4 500 Internal Server Error is used		



Learning outcome 1 end assessment

Theoretical assessment

I. Answer by True to the correct statements or False to the wrong statements

1. The POST HTTP method is typically used to retrieve data from a server.
2. Middleware in Express.js can be used to modify the request and response objects.
3. A 404 Not Found status code indicates that the requested resource was successfully created.
4. In Express.js, the next () function is used to pass control to the next middleware function in the stack.
5. Debugging a RESTful API involves reviewing and updating the API documentation as one of the steps.
6. CRUD operations in Node.js using Express.js include creating, reading, updating, and deleting resources.

II. Complete the following with correct words

7. To define endpoints in an Express.js application, you must first _____.
8. 2. The HTTP status code _____ is used when a server-side error occurs, such as a database query failing.
9. In Node.js, the _____ middleware checks if the data in the request meets necessary requirements.
10. When testing API endpoints with Postman, you should first select the _____.
11. A _____ is a tool in Express.js used to perform tasks like logging, input validation, and error handling during the request-response cycle.
12. The 201 Created status code is used when _____.
13. Once you want to develop a back end application the followings are requirements except select one option?
 - a) Node js installed on your computer
 - b) Npm
 - c) Nodemon
 - d) Install MS word
 - e) Express

f) Postman

14. Refers to postman in backend application development Respond by TRUE or False to the followings?

a) Postman is a popular tool for testing and documenting APIs.

b) It provides a user-friendly interface for sending HTTP requests to APIs, inspecting responses, and automating API testing.

15. Nodemon is a utility tool for Node.js that helps developers during the development process Respond by TRUE or FALSE.

16. Node.js is an open-source JavaScript runtime environment that allows developers to execute algorithm code on the server side. Respond by **TRUE** or **False**

17. Postman is a popular tool for developing and documenting APIs. Respond by **TRUE** or **False**

18. What will be performed once you use the following command

mkdir ubudeheproject

19. Expand the followings:

a) API:

b) REST:

c) NPM:

d) DBMS:

20. Dependencies are internal packages or libraries that a Node.js application relies on to perform various tasks. Respond by **TRUE** or **False**

21. What is the command that can be used to install express as node package?

22. What is command that can be used to install Nodemon as global?

Multiple Choice Questions

23. What command is used to initialize a Node.js project?

A) npm start

B) npm init -y

C) node init

D) npm create

Answer: B) npm init -y

24. Which package is used to parse incoming request bodies in Express?

A) express

- B) body-parser
- C) mysql
- D) nodemon

Answer: B) body-parser

25. What HTTP method is typically used to retrieve data from a server?

- A) POST
- B) PUT
- C) GET
- D) DELETE

Answer: C) GET

26. What status code indicates that a resource was successfully created?

- A) 404
- B) 200
- C) 500
- D) 201

Answer: D) 201

27. Which middleware function is used to log details of incoming requests?

- A) Error Handling Middleware
- B) Input Validation Middleware
- C) Logging Middleware
- D) Authentication Middleware

Answer: C) Logging Middleware

28. What is the correct method to delete a resource in REST?

- A) POST
- B) DELETE
- C) GET
- D) PUT

Answer: B) DELETE

29. Which command installs Express in your Node.js project?

- A) npm install express

- B) npm add express
- C) npm get express
- D) npm express install

Answer: A) npm install express

30. What is the purpose of Error Handling Middleware?

- A) To log requests
- B) To validate input
- C) To handle errors and send appropriate responses
- D) To connect to the database

Answer: C) To handle errors and send appropriate responses

31. Which HTTP status code indicates a resource was not found?

- A) 200
- B) 201
- C) 404
- D) 500

Answer: C) 404

32. What is the default port number for an Express server if not specified?

- A) 8080
- B) 3000
- C) 5000
- D) 4000

Answer: B) 3000

33. Match the HTTP methods of column B with their typical usage of column C

Column A Answers	Column B Methods	Column C Usage
A.....	A) GET	1) Retrieve data
B.....	B) POST	2) Update data
C.....	C) PUT	3) Delete data
D.....	D) DELETE	4) Create new resource

Answer:

- A) 1
- B) 4
- C) 2
- D) 3

34. Match the middleware of column B to its function in column C

Column A answers	Column B Middleware	Column C Function
A.....	A) Logging Middleware	1) Logs details of requests
B.....	B) Input Validation	2) Validates incoming data
C.....	C) Error Handling	3) Catches errors and sends responses

Answer:

- A) 1
- B) 2
- C) 3

Sentence Completion Questions

35. To create a new resource, you typically use the _____ HTTP method.

Answer: POST

36. In Express.js, middleware functions can modify the request and response objects, _____ the request-response cycle, and call the next middleware function.

Answer: end

37. The command to install the MySQL package in a Node.js project is _____.

Answer: npm install mysql

38. When a server encounters an unexpected issue, it typically responds with a _____ status code.

Answer: 500 Internal Server Error

39. To test API endpoints, you can use a tool called _____.

Answer: Postman

40. The _____ middleware is responsible for logging the details of incoming requests.

Answer: Logging

41. A resource that does not exist in the database should return a _____ status code.

Answer: 404 Not Found

42. The status code _____ indicates that a request was successful and data is returned.

Answer: 200 OK

43. To handle errors in an Express application, you can implement _____ middleware.

Answer: Error Handling

44. CRUD operations in Node.js require setting up an Express server that contains endpoints for _____.

Answer: Create, Read, Update, and Delete.

45. The method used to update an existing resource is typically _____.

Answer: PUT

46. An Express server can be started by calling the _____ method on the app object.

Answer: listen

47. The purpose of input validation middleware is to ensure that incoming data _____ the necessary requirements.

Answer: meets

48. A successful resource creation in a RESTful API returns a _____ status code.

Answer: 201 Created

49. To connect to a MySQL database in Node.js, you need to create a _____.

Answer: MySQL Connection

50. The _____ method is used to request data from a specified resource.

Answer: GET

51. In Express, to parse JSON data from incoming requests, you must use the _____ middleware.

Answer: body-parser

52. When debugging a RESTful API, it is important to _____ input data and monitor network traffic for issues.

Answer: validate

Practical assessment

Tela Tech Ltd is a company that develop websites for different institutions they have tasked you to develop for them a database (**Kigali innovation DB**) and inside create a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and develop APIs that will be used to insert, update, select and delete client's records via frontend where you will provide the developed APIs to be integrated with front end.

Include http status codes in order to overcome error handling issues.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Endpoints and HTTP Methods are clearly performed	1.1 Set Up Your Express Server is performed		
		1.2 Define Endpoint is performed		
		1.3 Endpoints are tested		
2	Testing endpoints using postman is well performed	2.1 Node js is Installed		
		2.2 Node and npm version are verified		
		2.3 Project name is created		
		2.4 The project folder have been navigated		
		2.5 cd command is used		
		2.6 Npm is initialised		
		2.7 Express js is Installed		
		2.8 Nodemon is installed		
		2.9 The project is opened in vs code		
3	middleware services are correctly Used	3.1 Logging Middleware have been used		
		3.2 Input Validation Middleware have been used		
		3.3 Error Handling middleware have been used		
4	CRUD operations using MySQL Database is well Performed	4.1 Required Packages are Installed		
		4.2 MySQL Connection is created		
		4.3 CREATE endpoints is created		
		4.4 Read endpoints is created		
		4.5 Update endpoints is created		
		4.6 Delete endpoints is created		
		4.7 CRUD operations is tested using Postman.		

5	HTTP Status code is well Used	5.1 201 have been Created		
		5.2 200 OK is created		
		5.3 404 Not Found is used		
		5.4 500 Internal Server Error is used		

END



Further information to the trainer

1. Doglio, F. (2018). *Rest API Development with Node.js*. Uruguay: Apress.
2. Mardan, A. (2018). *Practical Node.js*. California: Apress.
3. Mike Cantelon, M. H. (2014). *Node.js IN ACTION*. Shelter Island, NY11964: Manning Shelter Island.
4. Nael, D. (2019, March 11). Retrieved from okta: <https://developer.okta.com/blog/2019/03/11/node-sql-server>
5. Palmer, S. (2024). *how-to-build-a-secure-web-application-with-nodejs*. Retrieved from devteam: <https://www.devteam.space/blog/how-to-build-a-secure-web-application-with-nodejs/>
6. Syed, B. A. (2014). *Beginning Node.js*. New york: Apress.
7. W3schools. (2024). *nodejs_get_started.asp*. Retrieved from w3schools: https://www.w3schools.com/nodejs/nodejs_get_started.asp
8. Watmore, J. (2022, January 01). *nodejs-ms-sql-server-simple-api-for-authentication-registration-and-user-management*. Retrieved from jasonwatmore: <https://jasonwatmore.com/post/2022/07/01/nodejs-ms-sql-server-simple-api-for-authentication-registration-and-user-management>

Learning Outcome 2: Secure Backend Application



Indicative contents

- 2.1 Data encryption in securing RESTFUL APIs
- 2.2 Integrating and Using Third-Party Libraries
- 2.3 Maintaining and Updating Third-Party Libraries
- 2.4 Implementation of Authentication
- 2.5 Implementation of Authorization
- 2.6 Implementation of Accountability
- 2.7 Secure Environment Variables
- 2.7 Monitor and Manage Environment Variables

Key Competencies for Learning Outcome 2: Secure Backend Application

Knowledge	Skills	Attitudes
● Introduction to data encryption ● Describe of authentication, authorization and accountability	● Applying data encryption based on system security ● Check Third-party libraries based on system security ● Apply User Authentication, Authorization and Accountability (AAA) based on NPM Universal Access Control (UAC) ● Secure Environment variables according to system security	● Team work ● Critical thinker ● Being Innovative ● Being creative ● Practical oriented ● Detail oriented ● Being honesty ● Having Passion for Learning ● Problem-Solving Mindset ● Being collaborator and Communicator ● Attention to Security ● Ethical Coding



Duration: 20 hrs



Learning outcome 2 objectives:

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

1. Introduce correctly data encryption as used in nodejs backend application security
2. Describe perfectly authentication, authorization and accountability as used securing nodejs backend application
3. Apply correctly data encryption based on system security
4. Checking perfectly Third-party libraries based on system security
5. Apply effectively user Authentication, Authorization and Accountability (AAA) based on NPM Universal Access Control (UAC)
6. Secure correctly environment variables according to system security



Resources

Equipment	Tools	Materials
Computer	- Browser - Node.Js - Text Editor - Express. Js - Postman - Git - Swagger	- Internet - Electricity



Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have prepared computer lab with internet connectivity for trainees

- Have project with Node.js environment that contain APIs.
- Have computers with postman/ swagger installed
- Have visual studio code installed on all computers to be used
- Have videos used as didactic material.



Indicative content 2.1: Data Encryption in Securing RESTFUL APIs



Duration: 3 hrs



Theoretical Activity 2.1.1: Introduction to data encryption



Notes to the trainer:

- While delivering this content, a small group can be used on discussion for data encryption.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to respond to the following questions:

- Define the following terms:
 - Data encryption
 - Hashing
- Differentiate the types of data encryption
- Explain encryption techniques used in node js.
- Explain the benefits of data encryption

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.1.1 in the trainee manual.



Points to Remember

- Data encryption is essential in Node.js applications for safeguarding sensitive information such as user credentials and payment data.
- Encryption transforms plaintext into ciphertext, which is unreadable without the proper decryption key. There are three main types of encryption used in Node.js: symmetric encryption (e.g., AES), which uses the same key for both encryption and decryption; asymmetric encryption (e.g., RSA), which involves a public and private key pair; and hashing (e.g., SHA-256), which is a one-way encryption technique.
- Node.js provides built-in support for these through its crypto module, allowing developers to implement secure encryption and hashing techniques.



Practical Activity 2.1.2: Securing RESTFUL APIs



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should secure RESTFUL APIs.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed RESTFUL APIs.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to secure developed RESTFUL APIs on practical activity 1.4.8.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to secure RESTFUL APIs. While demonstrating, explain the steps to follow while securing RESTFUL APIs.

Step 4: Asks trainees to secure RESTFUL APIs and monitor the procedures.

Step 5: Verify whether the RESTFUL APIs are secured.

Step 6: Ask trainees to read key reading 2.1.2



Points to Remember

- While securing RESTFUL APIs you follow the following steps:
 - ✚ Install the crypto module
 - ✚ Create a key for encryption
 - ✚ Use the key to encrypt data
 - ✚ Convert the data to a buffer
 - ✚ Encrypt the data Store the encrypted data



Application of learning 2.1.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you have given developed APIs and your primary responsibility is to secure provided APIs by installing the crypto module, Creating a key for encryption, Use the key to encrypt data , , Convert the data to a buffer and Encrypt the data Store the encrypted data.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Securing RESTFUL APIs is well performed.	1.1 The crypto module is installed		
		1.2 A key for encryption is created		
		1.3 The key to encrypt data is used		
		1.4 The data to a buffer are converted		
		1.5 The data are encrypted		
		1.6 The encrypted data are Stored		



Indicative content 2.2: Integrating and Using Third-Party Libraries



Duration: 3 hrs



Practical Activity 2.2.1: Installing Node Js Package Manager (NPM)



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should Install Node Js Package Manager (NPM).
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to perform all steps to install node js package manager (NPM) in practical activity 2.2.1.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to install node js package manager (NPM).

Step 4: Ask trainees to install node js package manager (NPM) and monitor the procedures.

Step 5: Verify whether the node js package manager (NPM) installed.

Step 6: Ask trainees to read key reading 2.2.1



Points to Remember

- While installing node js package manager (NPM) you follow the following steps:
 - ✓ Download Node.js
 - ✓ Install Node.js
 - ✓ Verify Installation
 - ✓ Updating NPM (npm install -g npm@latest)
 - ✓ Initialize Nodejs project



Practical Activity 2.2.2: Incorporating Common Node.js Third-Party Libraries.



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should incorporate common Node.js third-party libraries like Express, Lodash and Moment.js
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed project in node js containing APIs.
 - ✓ Avail computer lab with nodejs installed with internet connection.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to incorporate common Node.js third-party libraries like Express, Lodash and Moment.js in project created on practical activity 2.2.1.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to incorporate common Node.js third-party libraries.

Step 4: Asks trainees to incorporate common Node.js third-party libraries like Express, Lodash and Moment.js and monitor the procedures.

Step 5: Verify whether the node js package manager (NPM) installed

Step 6: Ask trainees to read key reading 2.2.2.



Points to Remember

- While Incorporating common Node.js third-party libraries we use libraries like:

-  Express
-  Lodash
-  Moment.js



Practical Activity 2.2.3: Interacting with third-party libraries.



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should Interact with third-party libraries
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with NodeJS project created.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to interact with third-party libraries like: Callbacks, Promises and async/await in project created on practical activity 2.2.2.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to interact with third-party libraries.

Step 4: Asks trainees to interact with third-party libraries and monitor the procedures.

Step 5: Verify whether the interaction with third-party libraries done.

Step 6: Ask trainees to read key reading 2.2.3.



Points to Remember

- While interacting with third-party libraries we can use the following:

- Callbacks
- Promises
- async/await



Application of learning 2.2.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you have given developed APIs and your primary responsibility is to Integrate third-party libraries in Node.js by using NPM to manage and install packages like Express, Lodash, and Moment.js, which extend the functionality of applications. And also interact with these libraries through asynchronous operations handled via callbacks, promises and async/await in a created project.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Incorporate well the common Node.js third-party libraries	1.1 Express is used		
		1.2 Lodash is used		
		1.3 Moment.js is used		
		1.4 Server File is created		
		1.5 Express.js Server Code are written		
2	Interact well with third-party libraries	2.1 Callbacks is used		
		2.2 Promises is used		
		2.3 async/await is used		



Indicative content 2.3: Maintaining and Updating Third-Party Libraries



Duration: 3 hrs



Practical Activity 2.3.1: Monitoring of library dependencies and version numbers.



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should Monitor library dependencies and version numbers
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to monitor library dependencies and version numbers in package. Json and Npm-shrinkwrap. Json in project created on practical activity 2.2.3.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to monitor library dependencies and version numbers.

Step 4: Asks trainees to monitor library dependencies and version numbers in package. Json and Npm-shrinkwrap. json and monitor the procedures.

Step 5: Verify whether the Monitoring of library dependencies and version numbers done.

Step 6: Ask trainees to read key reading 2.3.1.



Points to Remember

- Monitor library dependencies and version numbers can be performed by using:



Package. Json



Npm-shrinkwrap. Json



Practical Activity 2.3.2: Checking for library updates and security vulnerabilities.



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should Check library updates and security vulnerabilities
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to Check library updates and security vulnerabilities using tools like NPM outdated, NPM audit and Snyk in project created on practical activity 2.3.1.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to Check library updates and security vulnerabilities.

Step 4: Asks trainees to Check library updates and security vulnerabilities and monitor the procedures.

Step 5: Verify whether the Checking library updates and security vulnerabilities done.

Step 6: Ask trainees to read key reading 2.3.2



Points to Remember

- While checking for library updates and security vulnerabilities using tools like:
 - NPM outdated
 - NPM audit
 - Snyk



Practical Activity 2.3.3: Updating third-party libraries.



Notes to the trainer

- This activity should take place in a Computer Lab where trainees should Update third-party libraries
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainee to do the task described below:

As a software developer, you are asked to go to the computer lab to Update Third-Party Libraries: Updating third-party libraries safely Versioning and semver rules

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to Update Third-Party Libraries.

Step 4: Asks trainees to Update Third-Party Libraries and monitor the procedures.

Step 5: Verify whether the Updating of Third-Party Libraries done.

Step 6: Ask trainees to read key reading 2.3.3



Points to Remember

- While updating third-party libraries safely we look for:

- Versioning
- semver rules



Theoretical Activity 2.3.4: Description of Strategies for managing and testing library updates



Notes to the trainer:

- While delivering this content, a small group can be used for Describing Strategies for managing and testing library updates
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following question:

i. How do you understand by the followings:

- a) staging environments
- b) Version control systems.

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

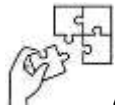
Step 6: Ask trainees to read the key reading 2.3.4 in the trainee manual.



Points to Remember

- Strategies for managing and testing library updates we can use:

- staging environments
- Version control systems.



Application of learning 2.3.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you are requested to Incorporate common Node.js third-party libraries like Express, Lodash and Moment.js and Interacting with these third-party libraries with Callbacks, Promises and async/await then Monitor library dependencies and version numbers in Package. Json and Npm-shrinkwrap. Json and also Check for library updates and security vulnerabilities using tools NPM outdated, NPM audit and Snyk and next Update third-party libraries by Versions and semver rules.

Check list:

SN	Elements	Indicators	Observation	
			Yes	No
1	Library dependencies and version numbers are well monitored.	1.1 Package. Json is used		
		1.2 Npm-shrinkwrap. Json is used		
2	Library updates and security vulnerabilities using tools are well Checked	2.1 NPM outdated is used.		
		2.2 NPM audit tool is used		
		2.3 Snyk tool is used		
3	Strategies for managing and testing library updates are well performed	3.1 Updates are tested using git		
		3.2 Environments are staged using git		
		3.3 managing them through Git branches, like merging etc. are performed		



Indicative content 2.4: Implementation of Authentication



Duration: 3 hrs



Theoretical Activity 2.4.1: Description of authentication



Notes to the trainer:

- While delivering this content, a small group can be used for Describing authentication.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following questions:

- What is Authentication
- What are the principles of authentication
- Explain role of authentication in system security

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.4.1 in the trainee manual.



Points to Remember

- Authentication is a critical process in Node.js applications, ensuring that users are who they claim to be before granting access to sensitive resources.
- The principles of authentication are:
 - Credentials Verification.
 - Multi-Factor Authentication (MFA).

 Token-Based Authentication.

- Role of Authentication in System Security

 Access Control:

 User Accountability

 Protection Against Unauthorized Access



Practical Activity 2.4.2: Implementing user authentication



Notes to the trainer

- This activity should take place in a Computer Lab for Implementing user authentication in Node.js using frameworks.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with NodeJS project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab to Implement Authentication in Node.js using frameworks like Passport, JWT (JSON Web Tokens) and Social Auth.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to Implement Authentication in Node.js using frameworks.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.4.2.



Points to Remember

- Implementing user authentication you can follow those steps:
 - ✓ Install Passport and Related Modules

- ✓ Configure Passport
- ✓ Integrate Passport into Your Application
- ✓ Run the Application
- Implementing Authentication with JWT (JSON Web Tokens)
 - ✓ Install JWT Modules
 - ✓ Create a JWT Authentication Middleware
 - ✓ Run the Application
- Implementing Social Authentication
 - ✓ Install Passport.js Social Auth Modules
 - ✓ Configure Google OAuth Strategy
 - ✓ Set Up Routes for Google Authentication
 - ✓ Run the Application



Practical Activity 2.4.3: Using authentication middleware



Notes to the trainer

- This activity should take place in a Computer Lab for authentication middleware to protect routes and resources.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab to Use authentication middleware to protect routes and resources.
- Step 2:** Explain the task and provide clear work instruction.

- Step 3:** Demonstrate how to Use authentication middleware to protect routes and resources.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.4.3.



Points to Remember

- Use authentication middleware to protect routes and resources
 1. Setting Up the Project
 2. Creating Authentication Middleware
 3. Testing Protected Routes
 4. Enhancing the Middleware



Theoretical Activity 2.4.4: Best practices for password storage and handling sensitive



Notes to the trainer:

- While delivering this content, a small group can be used for Describing Best practices for password storage and handling sensitive.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1:Involve trainees in group formulation

Step 2:Introduce the activity and request trainees to answer to the following questions:

- i. Describe the Best Practices for Password Storage
- ii. Describe the Best Practices for Handling Sensitive Data

Step 3:Involve trainees in presentation of their findings.

Step 4:Provides expert view and clarifies ideas by using didactic materials.

Step 5:Address any questions or concerns.

Step 6:Ask trainees to read the key reading 2.4.4 in the trainee manual.



Points to Remember

- Best Practices for Password Storage
 - ✚ Use Strong Hashing Algorithms
 - ✚ Salt Passwords Before Hashing
 - ✚ Apply Password Stretching
 - ✚ Role of Authentication in System Security
 - ✚ Avoid Reusing Hashing Algorithms
 - ✚ Never Store Plaintext Passwords
- Best Practices for Handling Sensitive Data
 - ✚ Encrypt Sensitive Data
 - ✚ Implement Access Control
 - ✚ Secure Data Transmission
 - ✚ Regularly Audit and Monitor Access
 - ✚ Educate and Train Users



Application of learning 2.4.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you are requested to Implementing user authentication using frameworks, Passport, JWT (JSON Web Tokens) and Social Auth. (Google, Facebook, ...) and then you will Use authentication middleware to protect routes and resources.

Checklist:

SN	Elements	Indicators	Observation	
			Yes	No
1	User authentication in Node.js using frameworks is well implemented	1.1 Passport is used		
		1.2 JWT (JSON Web Tokens) is used		
		1.3 Social Auth. Is used		
		1.4 Authentication middleware to protect routes and resources is used		



Indicative content 2.5: Implementation of Authorization



Duration: 3 hrs



Theoretical Activity 2.5.1: Description of authorization



Notes to the trainer:

- While delivering this content, a small group can be used for Describing authorization.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Introduce the activity and request trainees to answer to the following questions:

- What is Authorization
- What are the principles of authorization
- Explain role of authorization in system security

Step 2: Involve trainees in presentation of their findings.

Step 3: Provides expert view and clarifies ideas by using didactic materials.

Step 4: Address any questions or concerns.

Step 5: Ask trainees to read the key reading 2.5.1 in the trainee manual.



Points to Remember

- **Authorization** is the process of granting or denying access to resources based on the identity of the user and their assigned permissions.
- The principles of authentication are:
 - ✚ Principle of Least Privilege:
 - ✚ Role-Based Access Control (RBAC)
 - ✚ Attribute-Based Access Control (ABAC):

- Role of Authentication in System Security

- ✚ Protection of Sensitive Data
- ✚ Compliance with Regulations
- ✚ Prevention of Unauthorized Actions



Practical Activity 2.5.2: Implementing Role-Based and Attribute-Based Access Control



Notes to the trainer

- This activity should take place in a Computer Lab for Implementing Role-Based and Attribute-Based Access Control in Node.js
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab Implementing Role-Based and Attribute-Based Access Control in Node.js.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to Implement Authentication in Node.js using frameworks.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.5.2.



Points to Remember

- Implementation of Role-Based Access Control (RBAC) and Attribute-Based Access
 1. Set Up the Project
 2. Configure Sequelize

3. Define Models
4. Create Middleware for Access Control
5. Create Routes
6. Testing the Implementation
7. Database Synchronization



Practical Activity 2.5.3: Using Authorization Middleware

Notes to the trainer

- This activity should take place in a Computer Lab for Using Authorization Middleware to Manage User Permissions.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab to Use Authorization Middleware to Manage User Permissions.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to Use authentication middleware to protect routes and resources.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.5.3.



Points to Remember

- Using Authorization Middleware
 1. Set Up Authorization Middleware

2. Integrate Middleware in Routes
3. Set Up User Authentication
4. Apply Middleware to Routes
5. Start the Server and Test
6. Testing the Implementation



Practical Activity 2.5.4: Implementing Custom Authorization Logic for Specific Use Cases



Notes to the trainer

- This activity should take place in a Computer Lab for Implementing Custom Authorization Logic for Specific Use Cases.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab to Implement Custom Authorization Logic for Specific Use Cases.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to Implement Custom Authorization Logic for Specific Use Cases.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.5.4.



Points to Remember

- Implementing Custom Authorization Logic for Specific Use Cases you follow those steps:
 1. Identify the Use Case
 2. Create a Custom Authorization Middleware
 3. Apply the Middleware to Relevant Routes
 4. Test the Custom Logic
 5. Handle Complex Scenarios
 6. Optimize for Performance
 7. Document and Maintain the Custom Logic



Application of learning 2.5.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you are requested to implement role-based and attribute-based access control and Use authorization middleware to manage user permissions next you will implement custom authorization logic for specific use cases.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Role-based and attribute-based access control is well implemented	1.1 Role-based access control performed		
		1.2 Attribute-based access control performed		
2	Authorization middleware to manage user permissions well Used	1.3. Authorization middleware is used.		
3	Custom authorization logic for	1.4 Custom authorization		

	specific use cases is well implemented	logic for specific use cases is performed		
--	--	---	--	--



Indicative content 2.6: Implementation of Accountability



Duration: 3 hrs



Theoretical Activity 2.6.1: Description of accountability



Notes to the trainer:

- While delivering this content, a small group can be used for describing of accountability.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following questions:

- What is accountability
- What are the principles of accountability
- Explain role of accountability in system security

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.6.1 in the trainee manual.



Points to Remember

- **Accountability:** Accountability in system security ensures that every action in a system is attributable to a specific user, thus making users responsible for their actions.
- The principles of Accountability are:
 - Traceability
 - Non-Repudiation

Auditing

- Role of Accountability in System Security

 Enforcing Security Policies

 Detecting Malicious Behaviour

 Providing Evidence in the Event of a Security Breach



Practical Activity 2.6.2 Implementing Logging and Auditing Features



Notes to the trainer

- This activity should take place in a Computer Lab for Implement Logging and Auditing Features in Node.js using Popular Libraries.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab to Implement Logging and Auditing Features in Node.js using Popular Libraries like Winston and Morgan.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to Implement Logging and Auditing Features in Node.js using Popular Libraries.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.6.2.



Points to Remember

- While Implement Logging and Auditing Features in Node.js using Popular Libraries like:

 Winston

 Morgan



Theoretical Activity 2.6.3: Best practices for securely storing log data and protecting it



Notes to the trainer:

- While delivering this content, a small group can be used for describing Best practices for securely storing log data and protecting it.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following questions:

- i. Explain the best practices for securely storing log data and protecting it in system security

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.6.3 in the trainee manual.



Points to Remember

- Best Practices for securely storing log data and protecting it in system security:

- ✚ Encrypt Logs
- ✚ Access Controls
- ✚ Regular Audits
- ✚ Centralized Logging



Practical Activity 2.6.4 Detection of Security Events and System Errors



Notes to the trainer

- This activity should take place in a Computer Lab for Audit Logs to Detect Security Events and System Errors.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample developed internet connectivity.
 - ✓ Avail computer lab with nodejs project created with libraries installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are requested to go to the computer lab to Audit Logs to Detect Security Events and System Errors.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to Audit Logs to Detect Security Events and System Errors.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 2.6.4.



Points to Remember

- While performing Regular audits help identify security threats, reinforcing system security and ensuring compliance with security policies.



Application of learning 2.6.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you are requested to Implement logging and auditing features in Node.js using popular libraries like Winston and Morgan and then Audit logs to detect security events and system errors.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Logging and auditing features in Node.js are well implemented	1.1 logging features is implemented.		
2	Audit logs to detect security events and system errors using popular libraries is well performed	2.1 auditing features is implemented.		
		2.2 Winston libraries is used		
		2.3 Morgan libraries is used		



Indicative content 2.7: Secure Environment Variables



Duration: 1 hr



Theoretical Activity 2.7.1: Description of environment variable.



Notes to the trainer:

- While delivering this content, a small group can be used on discussion for Description of environment variable.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to respond to the following questions:

- What is the difference between types of Information Stored in Environment Variables
- What are the potential security risks of storing sensitive information in environment variables
- Explain the best practices for managing and securing environment variables in Node.js

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.7.1 in the trainee manual.



Points to Remember

- Types of information stored in environment variables are:

-  Database credentials
-  API keys

✚ Encryption keys.

- Potential Security Risks:

✚ Exposure

✚ Injection Attacks.

- Best Practices:

✚ Use .env files.

✚ Environment-Specific Variables

✚ Access Controls.



Practical Activity 2.7.2: Protecting environment variables



Notes to the trainer

- This activity should take place in the computer lab where trainees should Implementing security measures for protecting environment variables.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample nodejs project created with libraries installed.
 - ✓ Avails videos as didactic materials



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to to implement security measures for protecting environment variables by Encrypting secrets and Decrypting secrets.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to Encrypt secrets and Decrypt secrets.

Step 4: Asks learners implement security measures for protecting environment variables by Encrypting secrets and Decrypting secrets and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 2.7.2.



Points to Remember

- While Implementing security measures for protecting environment variables we:
 - ✚ Encrypting secrets
 - ✚ Decrypting secrets



Practical Activity 2.7.3: Storing environment variables



Notes to the trainer

- This activity should take place in the computer lab where trainees should Store environment variables in a secure location
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample nodejs project created with libraries installed.
 - ✓ Avails videos as didactic materials



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to Store environment variables in a secure location.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to Store environment variables in a secure location.

Step 4: Asks learners to Store environment variables in a secure location and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 2.7.3.



Points to Remember

- While we are storing environment variables, we can store in a secure location like:

- key management service
- env file



Practical Activity 2.7.4: Management and loading of environment variables



Notes to the trainer

- This activity should take place in the computer lab where trainees should Manage and load environment variables in Node.js applications using dotenv
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample nodejs project created with libraries installed.
 - ✓ Avails videos as didactic materials



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to Manage and load environment variables in Node.js applications using dotenv.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to Manage and load environment variables in Node.js applications using dotenv.

Step 4: Asks learners to Manage and load environment variables in Node.js applications using dotenv and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 2.7.4.



Points to Remember

- While we are Managing and loading environment variables in Node.js applications we can use dotenv.



Theoretical Activity 2.7.5 Best practices for passing environment variables to other services



Notes to the trainer:

- While delivering this content, a small group can be used on discussion for best practices for safely passing environment variables to other services and applications from unauthorized access.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to respond to the following questions:

- i. Explain best practices for safely passing environment variables to other services and applications

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

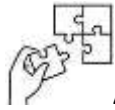
Step 7: Ask trainees to read the key reading 2.7.5 in the trainee manual.



Points to Remember

- Best Practices for safely passing environment variables to other services and applications:
 - ✚ Use Secure Channels
 - ✚ Avoid Hardcoding

Regularly Rotate Keys



Application of learning 2.7.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you are requested to Implement security measures for protecting environment variables by Encrypting secrets and Decrypting secrets then Storing environment variables in a secure location like key management service and env file next you will manage and load environment variables in Node.js applications using dotenv.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Security measures for protecting environment variables is well Implemented	1.1 Secrets is encrypted		
		1.2 Secrets is decrypted		
		1.3 Environment Variables are Stored in a Secure Location		
2	Environment variables in Node.js applications is well Managed and loaded	2.1 Environment variables managed and loaded using dotenv.		



Indicative content 2.8: Monitor and Manage Environment Variables



Duration: 1 hrs



Practical Activity 2.8.1: Implementing Logging and Auditing Features



Notes to the trainer

- This activity should take place in the computer lab where trainees should Implement Logging and Auditing Features to Detect Unauthorized Access to Environment Variables.
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample nodejs project created with libraries installed.
 - ✓ Avails videos as didactic materials



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to Implement Logging and Auditing Features to Detect Unauthorized Access to Environment Variables.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to Implement Logging and Auditing Features to Detect Unauthorized Access to Environment Variables.

Step 4: Asks trainees to Implement Logging and Auditing Features to Detect Unauthorized Access to Environment Variables.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 2.8.1.



Points to Remember

- While Implementing Logging and Auditing Features is to Detect Unauthorized Access to Environment Variables



Practical Activity 2.8.2: Monitoring Changes to Environment Variables



Notes to the trainer

- This activity should take place in the computer lab where trainees should Monitor Changes to Environment Variables and Detect Any Suspicious Activity
- While delivering this content, you are required to:
 - ✓ Avail computer lab with sample NodeJS project created with libraries installed.
 - ✓ Avails videos as didactic materials



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask learners do the task described below:

As backend developer you are requested to go to the computer lab to Monitor Changes to Environment Variables and Detect Any Suspicious Activity.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to Monitor Changes to Environment Variables and Detect Any Suspicious Activity.

Step 4: Asks learners Monitoring Changes to Environment Variables and Detecting Any Suspicious Activity and monitor the procedures.

Step 5: Verify whether the task have been performed.

Step 6: Ask trainees to read key reading 2.8.2.



Points to Remember

- We can track Changes in a .env files and Detecting Any Suspicious Activity



Theoretical Activity 2.8.3 Best Practices for Managing and Rotating Environment Variables.



Notes to the trainer:

- While delivering this content, a small group can be used on discussion for Best Practices for Managing and Rotating Environment Variables to Prevent Data Breaches.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1:Involve trainees in group formulation

Step 2:Introduce the activity and request trainees to respond to the following questions:

- I. Explain Best Practices for Managing and Rotating Environment Variables to Prevent Data Breaches

Step 3:Involve trainees in presentation of their findings.

Step 4:Provides expert view and clarifies ideas by using didactic materials.

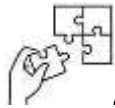
Step 5:Address any questions or concerns.

Step 6:Ask trainees to read the key reading 2.8.3 in the trainee manual



Points to Remember

- Best Practices for Managing and Rotating Environment Variables to Prevent Data Breaches:
 - ✚ Regularly Rotate Variables
 - ✚ Automated Rotation
 - ✚ Monitor Access
 - ✚ Backup Secure



Application of learning 2.8.

STU LTD is a software development company located in Musanze district, specializing in both frontend and backend solutions for various clients. As a newly hired backend developer using Node.js, you are requested to Monitor and manage environment variables by implementing logging to track changes in .env files detects unauthorized access.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	environment variables are well monitored and managed	1.1 unauthorized access are detected		
		1.2 Changes in an .env are tracked		
		1.3 changes are monitored		



Learning outcome 2 end assessment

Theoretical assessment

I: Multiple Choice Questions (MCQs)

1. Which type of encryption uses the same key for both encryption and decryption?
A) Symmetric Encryption
B) Asymmetric Encryption
C) Hashing
D) None of the above

Answer: A) Symmetric Encryption

2. Which algorithm is commonly used for symmetric encryption in Node.js?
A) RSA
B) AES
C) SHA-256
D) bcrypt

Answer: B) AES

3. In asymmetric encryption, which key is used to decrypt the data?
A) Public Key
B) Private Key
C) Symmetric Key
D) None of the above

Answer: B) Private Key

4. Which Node.js module provides built-in support for cryptographic operations?
A) fs
B) http
C) crypto
D) path

Answer: C) crypto

5. Which of the following is NOT a use case for hashing?
A) Storing passwords securely
B) Encrypting large amounts of data
C) Verifying data integrity
D) Generating fixed-size hash values

Answer: B) Encrypting large amounts of data

6. What is the output size of a SHA-256 hash?
A) 128 bits
B) 256 bits

- C) 512 bits
- D) 1024 bits

Answer: B) 256 bits

7. Which of the following is a disadvantage of symmetric encryption?

- A) Speed
- B) Key distribution
- C) Complexity
- D) Security

Answer: B) Key distribution

8. What does the bcrypt library primarily do?

- A) Encrypt files
- B) Hash passwords
- C) Generate keys
- D) Create SSL certificates

Answer: B) Hash passwords

9. Which encryption method is generally slower?

- A) Symmetric
- B) Asymmetric
- C) Both are equally fast
- D) Hashing

Answer: B) Asymmetric

10. In the context of encryption, what does IV stand for?

- A) Initialization Vector
- B) Interchangeable Variable
- C) Inner Value
- D) Independent Verification

Answer: A) Initialization Vector

II: True/False Questions

1. Asymmetric encryption uses the same key for both encryption and decryption.

Answer: False

2. The Moment.js library in Node.js is primarily used for date manipulation.

Answer: True

3. The crypto module in Node.js supports both symmetric and asymmetric encryption techniques.

Answer: True

4. Using bcrypt in Node.js automatically handles the addition of salt to the hashing process.

Answer: True

5. The package.json file in a Node.js project keeps track of dependencies and their versions.
Answer: True
6. Hash functions are reversible.
Answer: False
7. Asymmetric encryption is more suitable for bulk data encryption than symmetric encryption.
Answer: False
8. A public key can be shared openly while the private key must be kept secret.
Answer: True
9. All cryptographic algorithms are equally secure.
Answer: False
10. Hashing can be used to securely store passwords.
Answer: True

III: Fill in the Blank with Choices

1. Symmetric encryption is typically used for encrypting large amounts of data ____.
A) quickly B) slowly C) securely D) publicly
Answer: A) quickly
2. The ____ module in Node.js supports various cryptographic functions.
A) fs B) crypto C) path D) url
Answer: B) crypto
3. In asymmetric encryption, a pair of keys (public and private) is used, with one key encrypting the data and the other decrypting it.
A) symmetric B) asymmetric C) hashing D) block
Answer: B) asymmetric
4. The bcrypt library is commonly used in Node.js for securely ____.
A) hashing passwords B) encrypting files C) generating keys D) decoding messages
Answer: A) hashing passwords
5. A ____-bit key and a 16-byte initialization vector (IV) are generated for AES encryption.
A) 128 B) 256 C) 512 D) 1024
Answer: B) 256
6. ____ is often used to prevent brute-force attacks on hashed passwords.
A) Salting B) Encoding C) Hashing D) Encrypting
Answer: A) Salting

7. ____ encryption is more efficient for encrypting large amounts of data.
A) Symmetric B) Asymmetric C) Hashing D) Hybrid
Answer: A) Symmetric
8. The ____ algorithm is commonly used in asymmetric encryption to generate public and private keys.
A) SHA-1 B) RSA C) AES D) MD5
Answer: B) RSA
9. To store passwords securely, we use ____, not encryption.
A) Encoding B) Hashing C) Public Key D) Decryption
Answer: B) Hashing
10. The process of converting plaintext into ciphertext is known as ____.
A) Hashing B) Decryption C) Encryption D) Encoding
Answer: C) Encryption

IV: Read the following statement related to secure backend application and write the number corresponding to the correct description

Answer	Encryption Type	Description
.....2.....	A) Symmetric	1) Uses a pair of keys (public and private)
.....1.....	B) Asymmetric	2) Uses the same key for encryption and decryption
.....3.....	C) Hashing	3) Produces a fixed-size output from input data
.....4.....	D) Encryption	4) Converts plaintext into ciphertext

V: Multiple Choice Questions

1. Which of the following is a secure hash function?
A) SHA-1
B) MD5
C) SHA-256
D) All of the above
Answer: C) SHA-256
2. What is the purpose of a salt in hashing?
A) To increase speed
B) To add randomness
C) To encrypt data
D) To create keys
Answer: B) To add randomness
3. Which is an example of a one-way hash function?
A) AES
B) RSA
C) SHA-512
D) Blowfish
Answer: C) SHA-512

4. What type of attack does salting help prevent?

- A) Replay attacks
- B) Man-in-the-middle attacks
- C) Rainbow table attacks
- D) Denial of service attacks

Answer: C) Rainbow table attacks

5. Which cryptographic operation is reversible?

- A) Hashing
- B) Encryption
- C) Salting
- D) None of the above

Answer: B) Encryption

6. What is the main purpose of the initialization vector (IV)?

- A) To hash data
- B) To ensure randomness
- C) To create keys
- D) To compress data

Answer: B) To ensure randomness

7. Which of the following is an example of a symmetric encryption algorithm?

- A) RSA
- B) AES
- C) Diffie-Hellman
- D) DSA

Answer: B) AES

8. What does SSL stand for?

- A) Secure Socket Layer
- B) Simple Security Layer
- C) Standard Security Layer
- D) Secure System Layer

Answer: A) Secure Socket Layer

11. Which of the following is NOT a feature of hashing?

- A) Deterministic
- B) Fixed size output
- C) Reversible
- D) Fast

Answer: C) Reversible

12. Which Node.js function is used to create a hash?

- A) crypto.createCipher
- B) crypto.createHash
- C) crypto.generateKeyPair

D) `crypto.createSign`

Answer: B) `crypto.createHash`

Practical assessment

Tela Tech Ltd is a company that develop websites for different institutions they have tasked you to implement symmetric and asymmetric encryption using the Node.js 'crypto' module, employing algorithms like AES and RSA. Additionally, you will explore hashing techniques for securing passwords and verifying data integrity. You will also integrate third-party libraries such as 'bcrypt' for password hashing and 'jsonwebtoken' for secure data transmission using JWT. The exam will further involve securing RESTful APIs in Node.js, focusing on using encryption methods to protect data during transmission. In addition, you will demonstrate your understanding of integrating and managing Node.js third-party libraries, handling asynchronous operations with callbacks, promises, and `async/await`. Ensuring the security of dependencies by monitoring and updating them using `npm`, Manage and load environment variables in Node.js applications using `dotenv`, monitor changes to environment variables and detecting any suspicious activity, implement logging and auditing features to detect unauthorized access to environment variables.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Securing RESTFUL APIs is well performed.	1.1 The crypto module is installed		
		1.2 A key for encryption is created		
		1.3 The key to encrypt data is used		
		1.4 The data to a buffer are converted		
		1.5 The data are encrypted		
		1.6 The encrypted data are Stored		
2	Incorporate well the common Node.js third-party libraries	2.1 Express is used		
		2.2 Lodash is used		
		2.3 Moment.js is used		
		2.4 Server File is created		
		2.5 Express.js Server Code are written		
3	Interact well with third-party libraries	3.1 Callbacks is used		
		3.2 Promises is used		
		3.3 <code>async/await</code> is used		
4	Library dependencies and version numbers are well monitored.	4.1 Package. Json is used		
		4.2 <code>Npm-shrinkwrap</code> . Json is used		

5	Library updates and security vulnerabilities using tools are well Checked	5.1 NPM outdated is used.		
		5.2 NPM audit tool is used		
		5.3 Snyk tool is used		
6	Strategies for managing and testing library updates are well performed	6.1 Updates are tested using git		
		6.2 Environments are staged using git		
		6.3 managing them through Git branches, like merging etc. are performed		
		6.4 User authentication in Node.js using frameworks is well implemented		
7	Role-based and attribute-based access control is well implemented	7.1 Role-based access control performed		
		7.2 Attribute-based access control performed		
8	Authorization middleware to manage user permissions well Used	8.1 Authorization middleware is used.		
9	Logging and auditing features in Node.js is well implemented	9.1 Logging features is implemented.		
		9.2 Auditing features is implemented.		
10	Audit logs to detect security events and system errors using popular libraries is well performed	10.1 Winston libraries is used		
		10.2 Morgan libraries is used		
11	Authorization middleware to manage user permissions well Used	11.1 Authorization middleware is used.		
12	Custom authorization logic for specific use cases is well implemented	12.1 Custom authorization logic for specific use cases is performed		
13	Security measures for protecting environment variables is well Implemented	13.1 Secrets is encrypted		
		13.2 Secrets is decrypted		
		13.3 Environment Variables are Stored in a Secure Location		
14	Environment variables in Node.js applications are well	14.1 Environment variables managed and loaded using dotenv.		

	Managed and loaded			
15	environment variables are well	15.1 Unauthorized access is detected		
16	monitored and managed	16.1 Changes in a. env are tracked		

END



Further information to the trainer

Books

Doglio, F. (2018). Rest API Development with Node.js. Uruguay: Apress.

Mardan, A. (2018). Practical Node.js. California: Apress.

Mike Cantelon, M. H. (2014). Node.js IN ACTION. Shelter Island, NY11964: Manning Shelter Island.

Nael, D. (2019, March 11). Retrieved from okta: <https://developer.okta.com/blog/2019/03/11/node-sql-server>

Web links

javatpoint. (2024). nodejs-tutorial. Retrieved from javatpoint: <https://www.javatpoint.com/nodejs-tutorial>

Palmer, S. (2024). how-to-build-a-secure-web-application-with-nodejs. Retrieved from devteam: <https://www.devteam.space/blog/how-to-build-a-secure-web-application-with-nodejs/>

Syed, B. A. (2014). Beginning Node.js. New york: Apress.

W3schools. (2024). nodejs_get_started.asp. Retrieved from w3schools: https://www.w3schools.com/nodejs/nodejs_get_started.asp

Watmore, J. (2022, January 01). nodejs-ms-sql-server-simple-api-for-authentication-registration-and-user-management. Retrieved from jasonwatmore: <https://jasonwatmore.com/post/2022/07/01/nodejs-ms-sql-server-simple-api-for-authentication-registration-and-user-management>.

Learning Outcome 3: Test Backend Application



POSTMAN

Indicative contents

1.1 Implementation of Unit testing

1.2 Implementation of Usability testing

1.3 Implementation of Security Testing

Key Competencies for Learning Outcome 3: Test Backend Application

Knowledge	Skills	Attitudes
● Description of Usability tests ● Description of Secure Coding Practices in Node.js ● Description of Testing Techniques for Node.js Security ● Identification of Best Practices for Node.js Security Testing	● Using Mocha Testing Framework ● Using Chai assertion library ● Monitoring Test results ● Using Postman Testing Tool ● Using Puppeteer Testing Tool ● Implementing Security Testing in Nodejs ● Applying Penetration Testing steps ● Performing penetration Testing using OWASP	● Having Team work ability ● Being critical thinker ● Being Innovative ● Being creative ● Practical oriented ● Detail oriented ● Being honesty ● Passion for Learning ● Problem-Solving Mindset ● Collaboration and Communication ● Attention to Security ● Ethical Coding



Duration: 20 hrs



Learning outcome 3 objectives:

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

1. Use correctly Mocha Testing Framework as tool for testing backend
2. Use correctly Chai assertion library as tool for testing backend
3. Monitor properly Test results as applied in testing backend
4. Use correctly Postman Testing Tool as tool for testing backend
5. Use correctly Puppeteer Testing Tool as tool for testing backend
6. Implement clearly Security Testing in Nodejs as applied In software development
7. Apply correctly Penetration Testing steps as applied in software testing
8. Perform clearly penetration Testing using OWASP as applied in backend testing
9. Describe correctly Usability tests as step in backend testing
10. Describe correctly Secure Coding Practices in Node.js as applied in testing the backend
11. Describe correctly Testing Techniques for Node.js Security as applied in backend testing
12. Identify clearly Best Practices for Node.js Security Testing as applied in backend



Resources

Equipment	Tools	Materials
• Computer	• Browser • Node.Js • Text Editor • Express. Js	• Internet

	• Postman • Mocha	
 Advance Preparation: Before delivering this learning outcome, you are recommended to: • Have computers with installed web browser, node js and xampp server. • Have sample backend project developed to be used for implementing testing. • Have internet connection accessibility for all computers.		



Indicative content 3.1: Implementation of Unit Testing



Duration: 7 hrs



Theoretical Activity 3.1.1: Introduction to unit tests



Notes to the trainer:

- While delivering this content, small groups can be used for introducing unit testing.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to Discuss on the followings:

- ❖ Define the term unit testing
- ❖ Discuss on importance of Unit Testing in Node.js
- ❖ Explain unit testing process in Node.js
- ❖ Discuss on unit Testing Tools and Frameworks for Node.js
- ❖ Explain unit testing Libraries for Node.js

Step 3: Involve trainees in presentation of their findings.

Step 4: Provide expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 3.1.1 in the trainee manual.



Points to Remember

- Unit testing in Node.js involves testing individual components or functions of an application to ensure they work as expected. It is crucial for identifying bugs early, improving code quality, and ensuring reliable, maintainable applications. The process typically includes writing tests for each unit, running them, and verifying

the outcomes. Popular tools and frameworks for unit testing in Node.js include Mocha, Jest, and Jasmine, while libraries like Chai and Sinon provide utilities for assertions and mocking during tests.



Practical Activity 3.1.2: Using Mocha Testing Framework



Notes to the trainer

- This activity should take place in the computer lab where trainees should use mocha testing framework.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js developed project to be used



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to use mocha testing framework.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to use mocha testing framework. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to use mocha testing framework and monitor the procedures.

Step 5: Verify whether use mocha testing framework have been performed.

Step 6: Ask trainees to read key reading 3.1.2.



Points to Remember

- While using mocha testing framework you follow those stps:
 1. Installation and Configuration

2. Writing Unit Tests

3. Running Tests



Practical Activity 3.1.3: Using Chai assertion library



Notes to the trainer

- This activity should take place in the computer lab where trainees should use Chai assertion library.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js developed project to be used



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to use Chai assertion library.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to use Chai assertion library. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to use Chai assertion library and monitor the procedures.

Step 5: Verify whether use Chai assertion library have been performed.

Step 6: Ask trainees to read key reading 3.1.3.



Points to Remember

- Using chai assertion library, you follow the following steps:
 1. Installation and Configuration
 2. Writing Assertions

3. Chai `expect` and `should` APIs



Practical Activity 3.1.4: Monitoring Test results



Notes to the trainer

- This activity should take place in the computer lab where trainees should monitor test results.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js developed project to be used



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to monitor test results of practical activity 3.1.2.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to monitor test results. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to monitor test results and monitor the procedures.

Step 5: Verify whether monitoring test results have been performed.

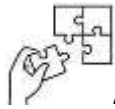
Step 6: Ask trainees to read key reading 3.1.4.



Points to Remember

- While monitoring test results you can perform the following steps:
 1. Running Tests Locally
 2. Generating Test Reports
 3. Using Continuous Integration (CI) Tools

4. Analysing Test Coverage
5. Automated Monitoring with Dashboards



Application of learning 3.1.

Tela Tech Ltd is a company that develop websites for different institutions they have provided you the developed database (Kigali innovation DB) and inside created a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and developed APIs that will be used to insert, update, select and delete client's records via frontend where you will be provide the developed APIs to be integrated with front end.

They have included http status codes in order to overcome error handling issues.

You have been tasked for:

- ✓ Using Mocha Testing Framework for unit testing
- ✓ Monitoring Test results

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Mocha Testing Framework for unit testing is well used	1.1 Installation and Configuration of mocha is done		
		1.2 Writing Unit Tests is done		
		1.3 Running Tests is performed		
2	Monitoring Test results is well performed	2.1 Running Tests Locally have been performed		
		2.2 Generating Test Reports is done		
		2.3 Continuous Integration (CI) Tool is used		
		2.3 Test Coverage is analysed		
		2.4 Automated Monitoring with Dashboards is done		



Indicative content 3.2: Implementation of Usability Testing



Duration: 7 hrs



Theoretical Activity 3.2.1: Introduction to Usability tests



Notes to the trainer:

- While delivering this content, small groups can be used for introducing usability testing.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to Discuss on the followings:

- Define the term Usability testing
- Discuss on importance of Usability Testing
- Explain usability testing process
- Identify usability Testing tools

Step 3: Involve trainees in presentation of their findings.

Step 4: Provide expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 3.2.1 in the trainee manual.



Points to Remember

- Usability testing is a technique used to evaluate a product or service by testing it with real users.

- There different importance of usability testing including User-Centered Design, Identifying Issues Early, Improving User Satisfaction, Validating Design Decisions and Competitive Advantage
- Once performing Usability testing you follow different Processes starting from Planning to reporting and Recommendations and you have to use testing tools like Nightwatch.js, Puppeteer, Cypress, WebdriverIO and TestCafe.



Practical Activity 3.2.2: Using Postman Testing Tool



Notes to the trainer

- This activity should take place in the computer lab where trainees should use postman for usability testing.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js project to be used for testing



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to test developed project on practical activity of outcome 1 by using postman.

Step 2: Explain the task and provide clear work instruction.

Step 3: Ask trainees to read key reading 3.2.2.

Step 4: Demonstrate how to use postman in testing. While demonstrating, explain the steps to be followed.

Step 5: Asks trainees to test developed project using postman and monitor the procedures.

Step 6: Verify whether project testing using postman have been performed.



Points to Remember

- Once testing usability for developed project in node js using postman you pass through the following steps:
 1. Installation of Postman
 2. Create a Collection
 3. Define Request
 4. Write Test Cases
 5. Run Tests
 6. Iterate and Improve



Practical Activity 3.2.3: Using Puppeteer Testing Tool



Notes to the trainer

- This activity should take place in the computer lab where trainees should use puppeteer testing tool for usability testing.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js project to be used for testing



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to test developed project on practical activity of outcome 1 by using Puppeteer.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to use Puppeteer in testing. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to test developed project using Puppeteer and monitor the procedures.

Step 5: Verify whether project testing using Puppeteer have been performed.

Step 6: Ask trainees to read key reading 3.2.3



Points to Remember

- Using the Puppeteer testing tool, covering everything from installation to generating reports you follow the following steps:
 1. Installation of Puppeteer
 2. Define Test Scenarios
 3. Automate User Interaction
 4. Measure Page Performance
 5. Test Accessibility
 6. Generate Report



Application of learning 3.2.

Tela Tech Ltd is a company that develop websites for different institutions they have provided you the developed database (Kigali innovation DB) and inside created a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and developed APIs that will be used to insert, update, select and delete client's records via frontend where you will be provide the developed APIs to be integrated with front end.

They have included http status codes in order to overcome error handling issues.

You have been tasked for:

- ✓ Use Postman testing tool for usability testing on APIs as provided.
- ✓ Use Puppeteer testing tool for usability testing.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Postman testing tool for usability testing on APIs is well used	1.1 Postman is installed		
		1.2 Collection is created		
		1.3 Request is defined		
		1.4 Test Cases is written		
		1.5 Run Tests is performed		
2	Puppeteer testing tool for usability testing is well used	2.1 Puppeteer tool is installed		
		2.2 Test Scenarios is defined		
		2.3 User Interaction is automated		
		2.4 Page Performance is well measured		
		2.5 Accessibility is tested		
		2.5 Report is generated		



Indicative content 3.3: Implementation of Security Testing



Duration: 6 hrs



Theoretical Activity 3.3.1: Introduction Node.js Security



Notes to the trainer:

- While delivering this content, small groups can be used for introducing node js security.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Involve trainees in group formulation
- Step 2:** Introduce the activity and request trainees to Discuss on key security issues in software development.
- Step 3:** Involve trainees in presentation of their findings.
- Step 4:** Provide expert view and clarifies ideas by using didactic materials.
- Step 5:** Address any questions or concerns.
- Step 6:** Ask trainees to read the key reading 3.3.1 in the trainee manual.



Points to Remember

- Node.js is a popular runtime environment for building scalable network applications. However, with its popularity, it becomes a target for various security threats like the followings:
 - ✓ Injection Attacks
 - ✓ Broken Authentication and Session Management
 - ✓ Cross-Site Scripting (XSS)
 - ✓ Cross-Site Request Forgery (CSRF)

- ✓ Security Misconfiguration
- ✓ Insecure Cryptographic Storage
- ✓ Insufficient Authorization
- ✓ Insufficient Logging and Monitoring



Theoretical Activity 3.3.2: Identification of Tools for Security Testing in Node.js



Notes to the trainer:

- While delivering this content, small groups can be used for identifying tools for security testing in node js.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Involve trainees in group formulation
- Step 2:** Introduce the activity and request trainees to identify tools for security testing in node js.
- Step 3:** Involve trainees in presentation of their findings.
- Step 4:** Provide expert view and clarifies ideas by using didactic materials.
- Step 5:** Address any questions or concerns.
- Step 6:** Ask trainees to read the key reading 3.3.2 in the trainee manual.



Points to Remember

- Security testing in Node.js involves using various tools to identify vulnerabilities and ensure that your application is secure. Those security testing tools and frameworks that came classified in the following types
 - ✓ Static Analysis Tools
 - ✓ Dynamic Analysis Tools
 - ✓ Testing Frameworks



Theoretical Activity 3.3.3: Description of Secure Coding Practices in Node.js



Notes to the trainer:

- While delivering this content, small groups can be used to describe secure coding practices in node js



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Involve trainees in group formulation
- Step 2:** Introduce the activity and request trainees to describe secure coding practices in node js.
- Step 3:** Involve trainees in presentation of their findings.
- Step 4:** Provide expert view and clarifies ideas by using didactic materials.
- Step 5:** Address any questions or concerns.
- Step 6:** Ask trainees to read the key reading 3.3.3 in the trainee manual.



Points to Remember

- Secure coding practices are essential to building robust and secure Node.js applications. These practices help prevent common vulnerabilities and ensure that your application is resilient against attacks. Below are key secure coding practices in Node.js.
 1. Input Validation and Sanitization
 2. Use of Parameterized Queries
 3. Authentication and Authorization
 4. Secure Error Handling
 5. Avoid Hardcoding Secrets
 6. Secure Dependencies
 7. Implement HTTPS/TLS
 8. Protect Against Cross-Site Scripting (XSS)
 9. Prevent Cross-Site Request Forgery (CSRF)
 10. Secure Session Management

11. Secure File Uploads
12. Use Security Headers
13. Conduct Regular Security Audits



Theoretical Activity 3.3.4: Description of Testing Techniques for Node.js Security



Notes to the trainer:

- While delivering this content, small groups can be used to describe Testing Techniques for Node.js Security.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Involve trainees in group formulation
- Step 2:** Introduce the activity and request trainees to describe Testing Techniques for Node.js Security.
- Step 3:** Involve trainees in presentation of their findings.
- Step 4:** Provide expert view and clarifies ideas by using didactic materials.
- Step 5:** Address any questions or concerns.
- Step 6:** Ask trainees to read the key reading 3.3.4 in the trainee manual.



Points to Remember

- Testing techniques for Node.js security is essential for identifying and mitigating vulnerabilities in your application. These techniques ensure that your Node.js application is robust, secure, and capable of withstanding various attacks. Those are testing techniques in node js.
 1. Static Application Security Testing (SAST)
 2. Dynamic Application Security Testing (DAST)
 3. Interactive Application Security Testing (IAST)
 4. Penetration Testing
 5. Fuzz Testing
 6. Security Unit Testing

- 7. Continuous Security Testing
- 8. Threat Modelling
- 9. Vulnerability Scanning
- 10. Security Regression Testing



Theoretical Activity 3.3.5: Description of best Practices for Node.js Security



Notes to the trainer:

- While delivering this content, small groups can be used to describe best Practices for Node.js Security.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Involve trainees in group formulation
- Step 2:** Introduce the activity and request trainees to describe best Practices for Node.js Security.
- Step 3:** Involve trainees in presentation of their findings.
- Step 4:** Provide expert view and clarifies ideas by using didactic materials.
- Step 5:** Address any questions or concerns.
- Step 6:** Ask trainees to read the key reading 3.3.5 in the trainee manual.



Points to Remember

- Best Practices for Node.js Security Testing include the followings:
 - 1. Integrate Security Testing Early and Often
 - 2. Use a Comprehensive Security Testing Suite
 - 3. Employ Threat Modelling
 - 4. Regularly Update and Audit Dependencies
 - 5. Perform Regular Penetration Testing
 - 6. Implement Security Regression Testing



Practical Activity 3.3.6: Implementation of Security Testing in Nodejs



Notes to the trainer

- This activity should take place in the computer lab where trainees should implement security testing in node js.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js developed to be used in implementing security testing.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to implement security testing in node js on developed project of outcome 1.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to implement security testing in node js. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to implement security testing in node js and monitor the procedures.

Step 5: Verify whether implementation of security testing in node js have been performed.

Step 6: Ask trainees to read key reading 3.3.6.



Points to Remember

- Implementing security testing in Node.js involves integrating various testing methodologies and tools throughout the development lifecycle to ensure that your application is secure from vulnerabilities and potential attacks. Below is a guide on how to implement security testing in Node.js:
 1. Setting Up the Environment
 2. Static Application Security Testing (SAST)

3. Dynamic Application Security Testing (DAST)
4. Interactive Application Security Testing (IAST)
- . Unit Testing for Security
6. Penetration Testing
7. Security Regression Testing
8. Continuous Integration and Deployment (CI/CD)
9. Logging and Monitoring
10. Reporting and Remediation
11. Compliance and Regular Audits
12. Training and Awareness



Practical Activity 3.3.7: Perform penetration Testing using OWASP



Notes to the trainer

- This activity should take place in the computer lab where trainees should implement security testing in node js.
- While delivering this content, you are required to:
 - ✓ Avail computers with node js and visual studio code installed.
 - ✓ Avail sample node js developed to be used in Performing Penetration Testing using OWASP.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As a software developer, you are asked to go the computer lab to Perform Penetration Testing using OWASP on developed project of outcome 1.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to Perform penetration Testing using OWASP. While demonstrating, explain the steps to be followed.

Step 4: Asks trainees to Perform Penetration Testing using OWASP and monitor the procedures.

Step 5: Verify whether Perform penetration Testing using OWASP have been performed.

Step 6: Ask trainees to read key reading 3.3.7.



Points to Remember

- Performing penetration testing using OWASP involves several steps to ensure thorough assessment and documentation of vulnerabilities.
 1. Installation of OWASP Tool
 2. Perform Scan
 3. Exploit Vulnerabilities
 4. Interpret Scan Report
 5. Document Results



Application of learning 3.3.

TelaTech Ltd is a company that develop websites for different institutions they have provided you the developed database (KigaliinnovationDB) and inside created a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and developed APIs that will be used to insert, update, select and delete client's records via frontend where you will be provide the developed APIs to be integrated with front end.

They have included http status codes in order to overcome error handling issues.

They have created a login form that will be used once accessing the system

You have been tasked for:

- ✓ Implementation of Security Testing in Nodejs
- ✓ Perform penetration Testing using OWASP

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Security Testing in Nodejs is well implemented	1.1 Setting Up the Environment is performed		
		1.2 Static Application Security Testing (SAST) is done		
		1.3 Dynamic Application Security Testing (DAST) have been performed		
		1.4 Interactive Application Security Testing (IAST) have been performed		
		1.5 Unit Testing for Security is performed		
		1.6 Security Regression Testing is performed		
		1.7 Continuous Integration and Deployment (CI/CD) is performed		
		1.8 Logging and Monitoring is applied		
		1.9 Reporting and Remediation is provided		
2	Penetration Testing using OWASP is well performed	2.1 Installation of OWASP Tool is done		
		2.2 Scan is performed		
		2.3 Exploitation of Vulnerabilities is done		
		2.4 Interpretation of Scan Report is performed		
		2.5 Document Results is provided		



Learning outcome 3 end assessment

Theoretical assessment

I. Read the following statement related to test backend application and choose the correct answer:

1. What is the primary purpose of unit testing?

Answer: b) To validate individual components or functions

2. Which of the following is a popular unit testing framework for Node.js?

Answer: d) Mocha

3. Which assertion library is commonly used with Mocha?

Answer: b) Chai

4. What command is used to install Mocha in a Node.js project?

Answer: a) npm install mocha

5. In Chai, which API is used for behavior-driven development (BDD) style assertions?

Answer: a) Should

6. What is the main benefit of using the Expect API in Chai?

Answer: a) It is simpler to write and understand

7. What is the first step in the unit testing process?

Answer: d) Setting up the testing environment

8. Which tool is commonly used for usability testing?

Answer: c) Usabilla

9. What does the acronym OWASP stand for?

Answer: a) Open Web Application Security Project

10. Which of the following is NOT a type of security vulnerability?

Answer: c) Unit Testing

11. Which command installs the Chai assertion library?

Answer: a) npm install chai

12. What type of testing is Postman primarily used for?

Answer: b) API Testing

13. What is the primary function of Puppeteer?

Answer: a) Simulate user interactions in web applications

14. In Mocha, which hook runs before all test cases?

Answer: a) before ()

15. Which is NOT a key benefit of unit testing?

Answer: d) Improved network performance

16. What is the purpose of the describe () function in Mocha?

Answer: a) To group related test cases

17. Which of the following is a feature of the Chai Expect API?

Answer: a) It allows chaining of assertions

18. Which method is used to run tests in Mocha?

Answer: b) npm test

19. What is an example of a dynamic analysis tool for security testing in Node.js?

Answer: a) OWASP ZAP

20. Which framework is commonly used to implement security tests in Node.js?

Answer: a) OWASP

21. What is a key benefit of using automated testing tools like Puppeteer?

Answer: a) Reduces human error

22. Which command is used to run a collection in Postman?

Answer: b) Newman run

23. In Mocha, how do you skip a test case?

Answer: b) Use it. skip ()

24. Which of the following is an example of a static analysis tool?

Answer: a) ESLint

25. What is a common use case for Chai's should API?

Answer: a) Writing test assertions in a more natural language style

II. Read the following statement related to test backend application in column B and C and write the number corresponding to the correct description

1. The Unit Testing Tools with their primary language support:

Answer	Column B	Column C
1.....	a) Mocha	1. JavaScript
2.....	b) Junit	2. Python
3.....	c) PyTest	3. Java
4.....	d) RSpec	4. Ruby

2. The Security Testing Tool with its function:

Answer	Column B	Column C
1.....	a) OWASP ZAP	1. Static code analysis
2.....	b) ESLint	2. Dynamic application security testing
3.....	c) Burp Suite	3. Automated web vulnerability scanner
4.....	d) SonarQube	4. Code quality analysis

3. The Puppeteer function with its description:

Answer	Column B	Column C
1.....	a) page.goto()	1. Simulates a browser click
2.....	b) page.click()	2. Takes a screenshot of the page
3.....	c) page.screenshot()	3. Navigates to a URL
4.....	d) page.type()	4. Enters text into an input field

4. The Chai assertion style with its description:

Answer	Column B	Column C
1.....	a) Expect	1. Classic assert style without chaining"
2.....	b) Should	2. Assert style chaining with "expect
3.....	c) Assert	3. Not a Chai assertion style
4.....	d) Chain	4. Assert style using "should"

5.The Usability Testing Tool with its feature:

Answer	Column B	Column C
1.....	a) Postman	1. Collecting user feedback on websites
2.....	b) Usabilla	2. Remote user testing and reporting
3.....	c) Optimizely	3. API testing and automation
4.....	d) Maze	4. A/B testing and experimentation

6.The type of vulnerability with its description:

Answer	Column B	Column C
1.....	a) SQL Injection	1.Executing unauthorized SQL commands
2.....	b) XSS	2. Storing sensitive data without proper encryption
3.....	c) CSRF	3. Injecting malicious scripts into web pages
4.....	d) Insecure Cryptographic Storage	4. Forcing users to perform actions they did not intend

7. The OWASP Top 10 risk with its associated attack type:

Answer	Column B	Column C
1.....	a) Injection	1. Script injection
2.....	b) Broken Authentication	2. Default passwords
3.....	c) Cross	3.Credential stuffing
4.....	d) Security Misconfiguration	4.SQL Injection

8. The Postman feature with its description:

Answer	Column B	Column C
1.....	a) Collections	1. Storing reusable data like API keys
2.....	b) Environment Variables	2. Grouping related API requests
3.....	c) Pre-request Script	3. Validating responses from API requests
4.....	d) Tests	4. Executing scripts before a request is sent

9. The Node.js Security Practice with its best practice:

Answer	Column B	Column C
1.....	a) Use of SSL/TLS	1.Keeping packages UpToDate
2.....	b) Input Validation	2. Encrypting data in transit
3.....	c)Dependency Management	3. Avoiding exposure of sensitive information
4.....	d) Error Handling	4. Sanitizing user inputs to prevent attacks

10. The unit test concept with its description

Answer	Column B	Column C
1.....	a) Test Case	1. A collection of related test cases
2.....	b) Test Suite	2. Simulating dependencies in a test
3.....	c) Mocking	3. A statement that must be true for the test to pass
4.....	d) Assertion	4. A single scenario to validate

III. Complete the following statement related to test backend application with the appropriate term used in backend application development: **Expect, SQL Injection, individual components or functions work as expected, describe (), document, automate, user experience and interface design, insecure, test, validate, before Each(),penetration testing, SSL/TLS encryption, scan, API**

1. Unit testing is primarily conducted to ensure that _____.

Answer: individual components or functions work as expected.

2. In Chai, the _____ API allows for natural language assertions.

Answer: Expect

3. The Mocha testing framework uses the _____ method to group related tests.

Answer: describe ()

4. Usability testing is important because it helps identify issues with _____.

Answer: user experience and interface design.

5. A common vulnerability in web applications, where an attacker can inject SQL queries, is known as _____.

Answer: SQL Injection

6. Postman is commonly used to _____ API endpoints and automate testing.

Answer: test

7. Puppeteer allows developers to _____ user interactions in a headless Chrome browser.

Answer: automate

8. In security testing, _____ is used to identify potential vulnerabilities before an attacker can exploit them.

Answer: penetration testing

9. Node.js applications should always use _____ to ensure data is transmitted securely.

Answer: SSL/TLS encryption

10. A test case in unit testing is designed to _____ a specific function or feature.

Answer: validate

11. The OWASP ZAP tool is used to _____ web applications for vulnerabilities.

Answer: scan

12. In Mocha, the _____ hook runs before each individual test case.

Answer: before Each ()

13. Chai's should _____ is often used for assertions in a BDD style.

Answer: API

14. Security misconfigurations often arise from _____ settings or defaults.

Answer: insecure

15. After performing a penetration test, it is crucial to _____ the findings in a detailed report.

Answer: document

Practical assessment

TelaTech Ltd is a company that develop websites for different institutions they have provided you the developed database (KigaliinnovationDB) and inside created a table called clients with the following fields ID, Names, Sex, Address, Phone and Email in node js and developed APIs that will be used to insert, update, select and delete client's records via frontend where you will be provide the developed APIs to be integrated with front end.

They have included http status codes in order to overcome error handling issues.

They have created a login form that will be used once accessing the system.

You have been tasked for:

- ✓ Use Postman testing tool for usability testing on APIs as provided.
- ✓ Use Puppeteer testing tool for usability testing.
- ✓ Using Mocha Testing Framework for unit testing
- ✓ Monitoring Test results
- ✓ Implementation of Security Testing in Nodejs
- ✓ Perform penetration Testing using OWASP for security testing

All tools' materials and equipment will be provided by the company.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Postman testing tool for usability testing on APIs is well used	1.1 Postman is installed		
		1.2 Collection is created		
		1.3 Request is defined		
		1.5 Test Cases is written		
		1.6 Run Tests is performed		
2	Puppeteer testing tool for usability testing is well used	2.1 Puppeteer tool is installed		
		2.2 Test Scenarios is defined		
		2.3 User Interaction is automated		
		2.4 Page Performance is well measured		
		2.5 Accessibility is tested		
		2.6 Report is generated		
3	Mocha Testing Framework for unit testing is well used	3.1 Installation and Configuration of mocha is done		
		3.2 Writing Unit Tests is done		
		Running Tests is performed		
4	Monitoring Test results is well performed	4.1 Running Tests Locally have been performed		
		4.2 Generating Test Reports is done		
		4.3 Continuous Integration (CI) Tool is used		
		4.4 Test Coverage is analysed		
		4.5 Automated Monitoring with		

		Dashboards is done		
5	Security Testing in Nodejs is well implemented	5.1 Setting Up the Environment is performed		
		5.2 Static Application Security Testing (SAST) is done		
		5.3 Dynamic Application Security Testing (DAST) have been performed		
		5.4 Interactive Application Security Testing (IAST) have been performed		
		5.5 Unit Testing for Security is performed		
		5.6 Security Regression Testing is performed		
		5.7 Continuous Integration and Deployment (CI/CD) is performed		
		5.8 Logging and Monitoring is applied		
		5.9 Reporting and Remediation is provided		
6	Penetration Testing using OWASP is well performed	6.1 Installation of OWASP Tool is done		
		6.2 Scan is performed		
		6.3 Exploitation of Vulnerabilities is done		
		6.4 Interpretation of Scan Report is performed		
		6.4 Document Results is provided		

END



Further information to the trainer

Books

Doglio, F. (2018). *Rest API Development with Node.Js*. Uruguay: Apress.

Mardan, A. (2018). *Practical Node.js*. California: Apress.

Mike Cantelon, M. H. (2014). *Node.js IN ACTION*. Shelter Island, NY11964: Manning Shelter Island.

Web Links

Adaptable. (2024). *deploy-nodejs-app*. Retrieved from adaptable: <https://adaptable.io/docs/app-guides/deploy-nodejs-app>.

Dizdar, A. (2022, July 27). *unit-testing-in-nodejs*. Retrieved from Bright: <https://brightsec.com/blog/unit-testing-in-nodejs/>.

Ebere, N. S. (2022, March 3). *how-to-build-a-backend-application*. Retrieved from freecodecamp: <https://www.freecodecamp.org/news/how-to-build-a-backend-application>.

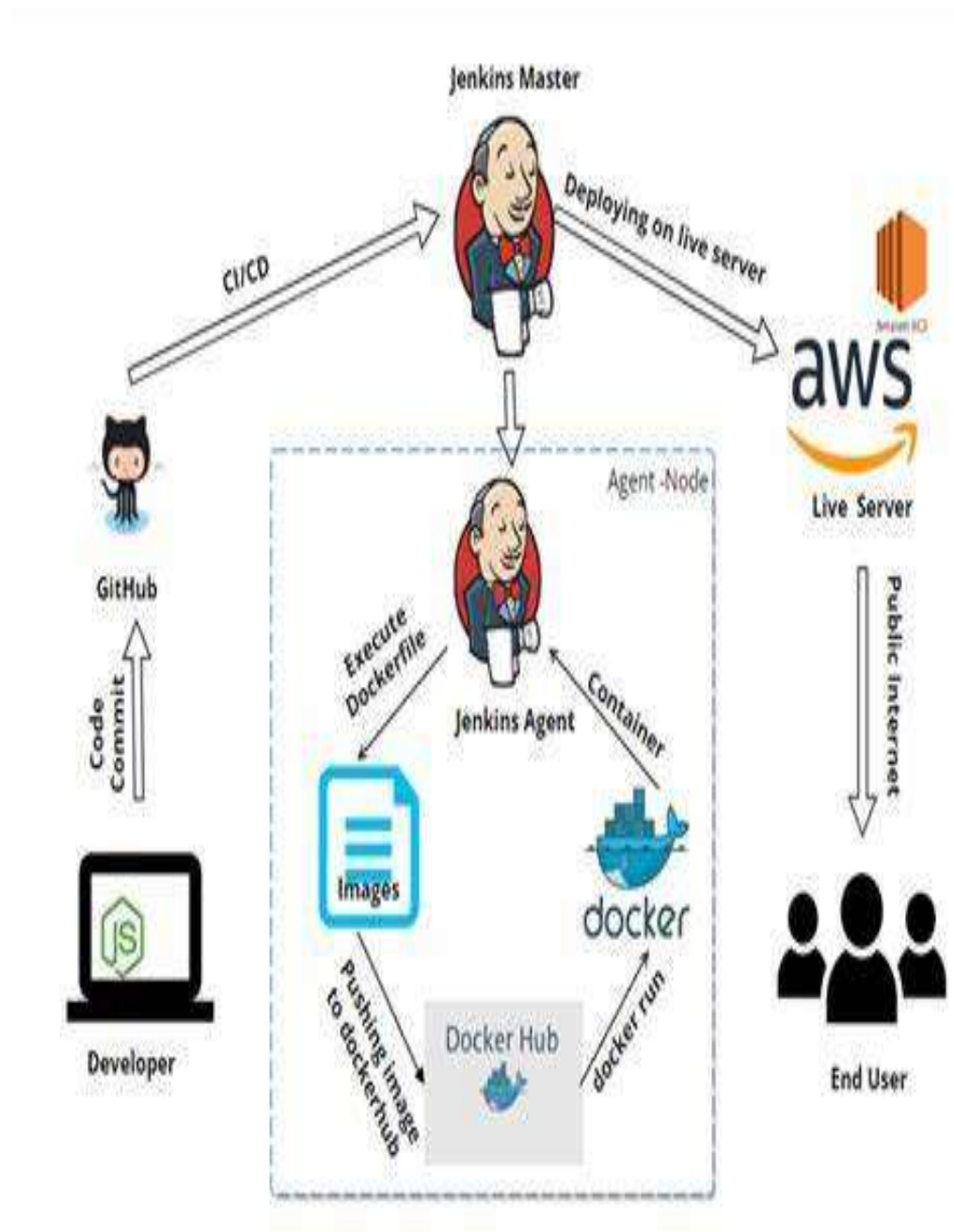
learnthinkcreatecode. (2022, October 25). *How to test a Node.js REST API using postman*.

Nael, D. (2019, March 11). Retrieved from okta: <https://developer.okta.com/blog/2019/03/11/node-sql-server>.

YouTube. (2020, September 13). *watch?v=VStXIFxQgZg*. Retrieved from youtube: <https://www.youtube.com/watch?v=VStXIFxQgZg>.

YouTube. (2022, October 25). *watch?v=zLbsM_n1H9w*. Retrieved from youtube: https://www.youtube.com/watch?v=zLbsM_n1H9w.

Learning Outcome 4: Manage Backend Application.



Indicative contents

4.1 Preparation of deployment Environment

4.2 Implementation of Manual Deployment of NodeJS application

4.3 Maintenance of NodeJS application

4.4 Application of NodeJS Documentation Tools and Frameworks

Key Competencies for Learning Outcome 4: Manage Backend Application

Knowledge	Skills	Attitudes
● Description of NodeJS application deployment ● Description of Types of NodeJS application deployment ● Describing the Application of NodeJS Documentation Tools and Frameworks	● Maintaining NodeJS application ● Preparing NodeJS Application Deployment tools ● Publishing Documentation ● Implementing Manual Deployment of NodeJS application ● Developing a maintenance plan	● Team work ● Being critical thinker ● Being Innovative ● Being creative ● Practical oriented ● Detail oriented ● Being honesty ● Having Passion for Learning ● Problem-Solving Mindset ● Being Collaborator and Communicator ● Attention to Security ● Ethical Coding



Duration: 15 hrs



Learning outcome 4 objectives:

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

1. Describe correctly NodeJS application deployment as used in backend application.
2. Describe correctly the Types of NodeJS application deployment as used in backend application
3. Develop effectively a maintenance plan based on deployed application.
4. Describe correctly the Application of NodeJS Documentation Tools and Frameworks as used in node js application deployment
5. Maintain perfectly NodeJS application deployed as used in nodejs application deployment
6. Prepare correctly the NodeJS Application Deployment tools as used in the server machine.
7. Publish correctly documentation as used in nodejs application deployment
8. Implement effectively Manual Deployment of NodeJS application as used in nodejs application deployment.



Resources

Equipment	Tools	Materials
• Computer	• Browser • Node.js • Text Editor • Express. Js • Postman • GitHub • Swagger • OWASP	• Internet

	• Webserver • MySQL Workbench • Winston • PM2 • Redis • AWS Lambda	
 Advance Preparation: Before delivering this learning outcome, you are recommended to: • Have prepared computer lab with internet connectivity for trainees • Have computer with a nodejs application developed. • Have a sever computer with NOS installed (window server OS) • Git installed • Postman installed. • Vscode editor installed • Have videos used as didactic material.		



Indicative content 4.1: Preparation of Deployment Environment



Duration: 4 hrs



Theoretical Activity 4.1.1: Description of NodeJS application deployment



Notes to the trainer:

- While delivering this content, a small group can be used for Describing NodeJS application deployment.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1:Involve trainees in group formulation

Step 2:Introduce the activity and request trainees to answer to the following questions:

- Define the following terms “Node.js application deployment”
- Differentiate the types of Node.js application deployment
- Write short notes about the following terms used in nodejs application deployment:
 - Environment Configuration
 - Dependencies
 - Build the Application
 - Testing

Step 3:Involve trainees in presentation of their findings.

Step 4:Provides expert view and clarifies ideas by using didactic materials.

Step 5:Address any questions or concerns.

Step 6:Ask trainees to read the key reading 4.1.1 in the trainee manual.



Points to Remember

- Node.js application deployment involves preparing, transferring, and running a Node.js app on a server or cloud environment, with proper setup for runtime environments, dependencies, and configurations.
- Deployment types include manual, continuous, and Docker-based.
- Manual deployment requires hands-on server management, while continuous deployment automates the process via CI/CD pipelines.
- Docker-based deployment uses containers for consistent environments.



Practical Activity 4.1.2: Preparation of NodeJS Application Deployment tools.



Notes to the trainer

- This activity should take place in a Computer Lab.
- While delivering this content, you are required to:
 - ✓ Avail servers computers with NOS (Window server OS) installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are asked to go to the computer lab to prepare NodeJS Application Deployment tools: NodeJS Runtime, Package Manager, Operating system, Webserver and Database.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to deploy node js app, check if nodejs application deployed.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 4.1.2.



Points to Remember

- Deploying a Node.js application on a Windows Server with a MySQL database. You'll use IIS to serve the application, ensuring it is accessible to users via the web.
- PM2 will manage your application process, and MySQL will handle your application's data. Make sure to continuously monitor, update, and secure your application and server to keep your deployment stable and secure.



Application of learning 4.1.

STU LTD is a software development company located in Musanze district, it develops software both frontend and backend for different companies, due to different activities that they have, you are hired as backend developer and deployment of node js application responsible for Preparing the following NodeJS Application Deployment tools: NodeJS Runtime, Package Manager, Operating system, Webserver and Database.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	NodeJS Application Deployment tools are well prepared	1.1 NodeJS Runtime is prepared		
		1.2 Package Manager is prepared.		
		1.3 Operating system is installed.		
		1.4 Webserver is configured		
		1.5 Database is prepared		



Indicative content 4.2: Implementation of Manual Deployment



Duration: 4 hrs



Practical Activity 4.2.2: Implementation of manual deployment of nodeJS application.



Notes to the trainer

- This activity should take place in a Computer Lab.
- While delivering this content, you are required to:
 - ✓ Avail servers computers with NOS (Window server OS) and required features like webserver and mysql installed.
 - ✓ Avail video as didactic material.



Key steps:

While delivering this activity, pass through the following steps:

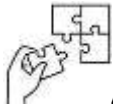
- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are asked to go to the computer lab to deploy the project Manual Deploying of NodeJS application by Copying the application source code to the server, installing of dependencies, Start the application using command line.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to deploy node js app, check if nodejs application deployed.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 4.2.1.



Points to Remember

- **While we are implementing manual deployment, to Copy the Application we** use RDP, FTP, SFTP, or SCP to transfer files to the Windows server. To **install dependencies, we** use npm install to set up dependencies on the server.
- **To Start the Application, we** run the application using node index.js OR npm start.

- Then **optional use PM2** For process management and auto-restart functionality.



Application of learning 4.2.

STU LTD is a software development company located in Musanze district, it develops software both frontend and backend for different companies, due to different activities that they have, you are hired as backend developer and deployment of node js application responsible for deploy the project Manual Deploying of NodeJS application by Copying the application source code to the server, Installing of dependencies, Start the application using command line.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Manual Deployment of NodeJS application is well Implemented	1.1 Option to be used is selected.		
		1.2 Codes are copied to sever.		
		1.3 Dependencies are installed.		
		1.4 Option to be used is selected to start application.		
		1.5 Application is started Using Command Line		



Indicative content 4.3: Maintenance of NodeJS Application



Duration: 4 hrs



Theoretical Activity 4.3.1: Description of Best practices for maintenance.



Notes to the trainer:

- While delivering this content, a small group can be used for requested to describe best practices for maintaining nodejs application.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to answer to the following questions:

- Write short notes about the following terms used in maintaining nodejs application deployed:
 - Update
 - Monitor
 - Perform test

Step 3: Involve trainees in presentation of their findings.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 4.3.1 in the trainee manual.



Points to Remember

- While performing best practices for maintenance a deployed nodejs application we consider on:
 - ✚ Update
 - ✚ Monitor

 Perform test



Practical Activity 4.3.1: Development of maintenance plan



Notes to the trainer

- This activity should take place in a Computer Lab.
- While delivering this content, you are required to:
 - ✓ Avail computer with VS code installed and nodejs application developed.
 - ✓ Git installed
 - ✓ postman installed
 - ✓ Application deployed.
- Avail video as didactic material



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are asked to go to the computer lab to Update, Monitor and Perform test as practices for maintenance for deployed nodejs application.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to maintain a deployed nodejs application
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 4.3.2.



Points to Remember

- While developing a maintenance plan for a Node.js application. You have to consider the following aspects, including:
 - ✓ **Identification of Maintenance Requirements:** Listing dependencies, auditing for vulnerabilities, and updating packages.
 - ✓ **Scheduling Regular Updates:** Setting up a CI/CD pipeline for automated updates.

- ✓ **Automating Maintenance Tasks:** Automating database backups and using Task Scheduler.
- ✓ **Monitoring Application Performance:** Utilizing PM2 for monitoring.
- ✓ **Testing Regularly:** Setting up automated tests with Mocha and Chai.
- ✓ **Disaster Recovery Plan:** Automating and testing database backups.
- ✓ **Documenting Changes:** Maintaining a changelog and using version control.



Practical Activity 4.3.2: Performing Continuous maintenance and improvement



Notes to the trainer

- This activity should take place in a Computer Lab.
- While delivering this content, you are required to:
 - ✓ Avail computer with VS code installed and nodejs application developed.
 - ✓ Git installed
 - ✓ Application deployed.
- Avail video as didactic material



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the topic and ask trainees to do the task described below:
As a backend developer, you are asked to go to the computer lab Upgrade and maintain previously developed functionalities, develop new functionalities, secure new and previously developed functionalities, Test new functionalities, Deploy new changes.
- Step 2:** Explain the task and provide clear work instruction.
- Step 3:** Demonstrate how to maintain a deployed nodejs application.
- Step 4:** Asks trainees to perform the activity of step 3 and monitor the procedures.
- Step 5:** Verify whether the tasks are clearly performed.
- Step 6:** Ask trainees to read key reading 4.3.3.



Points to Remember

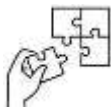
- While performing continuous maintenance and improvement NodeJS applications we:

- ✓ **Upgrade and Maintain:**

Use npm-check-updates for dependency updates.

Refactor code with ESLint and perform code reviews.

- ✓ **Develop New Features:** Plan, implement, and test new features using Git branches.
- ✓ **Secure Application:** Apply security patches and implement security measures with tools like helmet and dotenv.
- ✓ **Test:** Run automated tests with Jest and perform manual testing.
- ✓ **Deploy Changes:** Set up CI/CD pipelines, deploy to production, and monitor performance.



Application of learning 4.3.

STU LTD is a software development company located in Musanze district, it develops software both frontend and backend for different companies, due to different activities that they have, you are hired as backend developer and deployment in node js responsible for deploying the project folder (OurWebinfo) on local disk D, and Upgrade and maintain previously developed functionalities, develop new functionalities, Secure new and previously developed functionalities, Test new functionalities, Deploy new changes.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Continuous maintenance and improvement of NodeJS applications is well performed	1.1 Previous developed functionalities are Upgraded and maintained.		
		1.2 New functionalities are developed.		
		1.3 New and previously developed functionalities are Secured.		
		1.4 New functionalities are Tested.		
		1.5 New changes are deployed		



Indicative content 4.4: Application of NodeJS Documentation Tools and Frameworks



Duration: 3 hrs



Theoretical Activity 4.4.1: Description of nodejs documentation.



Notes to the trainer:

- While delivering this content, small groups can be used for Describing of nodejs documentation.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to Discuss on the followings:

- i. What is documentation?
- ii. What are the importance of documenting in Node.js application deployment?
- iii. What are the differences between types of documentation?

Step 3: Involve trainees in presentation of their findings.

Step 4: Provide expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 4.4.1 in the trainee manual



Points to Remember

- Documentation for Node.js applications is essential for detailing functionality, setup, and integration, enhancing understandability, maintenance, and user support.
- Key types include code comments, API documentation (using Swagger or Postman), user guides, and design overviews.



Theoretical Activity 4.4.2: Description of nodejs documentation tools and frameworks.



Notes to the trainer:

- While delivering this content, small groups can be used for Describing of nodejs documentation tools and frameworks.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Involve trainees in group formulation

Step 2: Introduce the activity and request trainees to Discuss on the followings:

- i. Differentiate Popular Documentation Tools and Frameworks you know.
- ii. Describe the best practice for documentation

Step 3: Involve trainees in presentation of their findings.

Step 4: Provide expert view and clarifies ideas by using didactic materials.

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 4.4.2 in the trainee manual



Points to Remember

- While documenting we use Tools like Swagger, Postman, JSDoc, and Docdash help in generating and presenting documentation. Best practices involve clear comments, consistent formatting, and regular updates



Practical Activity 4.4.1: Publishing documentation



Notes to the trainer

- This activity should take place in a Computer Lab.
- While delivering this content, you are required to:
 - ✓ Avail computer with VS code installed and nodejs application developed.
 - ✓ Postman installed

- ✓ Git installed
- ✓ Application deployed.
- Avail video as didactic material



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees do the task described below:

As a software developer, you are asked to go to the computer lab to host documentation, Use GitHub for collaborative documentation, document the Maintenance.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to host documentation, Use GitHub for collaborative documentation, document the Maintenance.

Step 4: Asks learners to host documentation, Use GitHub for collaborative documentation, document the Maintenance and monitor the procedures.

Step 5: Verify whether host documentation, Use GitHub for collaborative documentation and document the Maintenance have been performed.

Step 6: Ask trainees to read key reading 4.4.3



Points to Remember

- You can effectively deploy, maintain, and continuously improve your Node.js application using GitHub as the primary tool for collaboration, automation and documentation



Application of learning 4.4.

STU LTD is a software development company located in Musanze district, it develops software both frontend and backend for different companies, due to different activities that they have, you are hired as backend developer and deployment in node js responsible for Documenting deployed nodejs application, with GitHub facilitating collaborative updates. Regular maintenance includes updating docs, collecting feedback, and integrating updates into the CI/CD pipeline.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Documentation is clearly published	1.1 Options for hosting documentation is selected.		
		1.2 GitHub for collaborative documentation is used		
		1.3 Documentation Maintenance is performed.		



Learning outcome 4 end assessment

Theoretical assessment

I. Read the following statement related to manage backend application and choose the correct answer:

1. What is the primary purpose of Node.js application deployment?

- a) Writing JavaScript code
- b) Debugging the application
- c) Preparing, transferring, and running a Node.js application on a server or cloud environment
- d) Creating a user interface

Answer: c) Preparing, transferring, and running a Node.js application on a server or cloud environment

2. Which of the following tools is commonly used for containerizing a Node.js application?

- a) Jenkins
- b) Docker
- c) Nginx
- d) Git

Answer: b) Docker

3. Which command is used to start a Node.js application if index.js is the entry point?

- a) npm install
- b) npm run build
- c) npm start
- d) node index.js

Answer: d) node index.js

4. Which of the following is NOT a type of Node.js application deployment?

- a) Manual Deployment
- b) Continuous Deployment

- c) Docker-based Deployment
- d) Script-based Deployment

Answer: d) Script-based Deployment

5. Which process manager is commonly used to manage Node.js applications in production?

- a) npm
- b) PM2
- c) Yarn
- d) GitHub Actions

Answer: b) PM2

II. Complete the following statement related to test backend application by choosing the correct answer

1. In a CI/CD pipeline, tools like _____ are used to automate the testing and deployment process.

- a) Jenkins
- b) FTP
- c) npm
- d) Webpack

Answer: a) Jenkins

2. To manage environment variables and sensitive information like API keys in a Node.js application, you should use _____.

- a) package.json
- b) npm install
- c) Environment Variables
- d) scp

Answer: c) Environment Variables

3. Before deploying a Node.js application, it is essential to install all necessary modules using _____.

- a) npm start
- b) npm install

- c) Docker
- d) Git

Answer: b) npm install

4. The command `pm2 start index.js` is used to _____ the application using PM2.

- a) Install
- b) Monitor
- c) Start
- d) Delete

Answer: c) Start

5. For secure file transfer to a server, tools like _____ can be used.

- a) RDP
- b) SCP
- c) FTP
- d) Git

Answer: b) SCP

III. Read the following statement related to manage backend application and write the number corresponding to the correct description

1. Deployment and description

Answer	Deployment Type	Description
1.....	A) Manual Deployment	1) Automated process with CI/CD pipelines
2.....	B) Continuous Deployment	2) Containerizing the application using Docker
3.....	C) Docker-based Deployment	3) Manually transferring and starting the application on a server

2. The tool with its function:

Answer	Tool	Function
1.....	A) Webserver	1) Stores and retrieves application data
2.....	B) Database	2) Executes JavaScript code on the server
3.....	C) NodeJS Runtime	3) Serves the application and handles incoming traffic

IV. Read the following statement related to manage backend application and answer by True if the statement is correct and False if the statement is wrong

1. Node.js is a runtime environment that allows you to run JavaScript on the server side.

Answer: True

2. Continuous deployment means that every change that passes the automated tests is deployed to production automatically.

Answer: True

3. Docker is a version control system used for managing Node.js application code.

Answer: False (Docker is a tool for containerizing applications, not a version control system.)

4. PM2 is used for managing and monitoring Node.js applications in production.

Answer: True

5. SCP is used to manage Node.js dependencies and packages.

Answer: False (SCP is used for secure file transfer, while npm or Yarn is used to manage dependencies.)

6. In a CI/CD pipeline, Jenkins can be used to automate the deployment of a Node.js application.

Answer: True

7. Environment variables are used to store sensitive information, such as API keys, in a Node.js application.

Answer: True

8. A webserver is responsible for storing and retrieving data in a Node.js application.

Answer: False (A webserver handles HTTP requests, while a database is responsible for storing and retrieving data.)

Practical assessment

TelaTech Ltd is a company that develop and deploy websites for different institutions they have tasked you to deploy for them a nodejs application by Preparing deployment Environment on the windows server: NodeJS Runtime, Package Manager, Operating system, Webserver and Database. Implement Manual Deployment of NodeJS application: Copy the application source code to the server, install dependencies, Start the application using command line. Develop a maintenance plan: Identify the maintenance requirements, Schedule regular updates, automate maintenance tasks, Monitor application performance, Test regularly, Disaster recovery plan, Document changes. In additional perform Continuous maintenance and improvement of NodeJS applications: Upgrade and maintain previously developed functionalities, develop new functionalities, Secure new and previously developed functionalities, Test new functionalities, Deploy new changes. Next, apply NodeJS Documentation Tools and Frameworks: Use popular documentation tools and frameworks like Swagger/Postman for API documentation, write clear and concise comments using documentation generators then after Publish Documents and Optional Use GitHub for collaborative documentation.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	NodeJS Application Deployment tools are well Prepared	1.1 NodeJS Runtime is prepared		
		1.2 Package Manager is prepared.		
		1.3 Operating system is installed.		
		1.4 Webserver is configured		
		1.5 Database is prepared		
2	Manual Deployment of NodeJS application is well Implemented	2.1 Option to be used is selected.		
		2.2 Codes are copied to sever.		
		2.3 Dependencies are installed.		
		2.4 Option to be used is selected to start application.		
		2.5 Application is started Using Command Line		
3	Continuous maintenance and improvement of NodeJS applications is well performed	3.1 Previous developed functionalities are Upgraded and maintained.		
		3.2 New functionalities are developed.		
		3.3 New and previously developed functionalities are Secured.		
		3.4 New functionalities are Tested.		
		3.5 New changes are deployed		

4	Documentation is clearly published	4.1 Options for hosting documentation is selected.		
		4.2 GitHub for collaborative documentation is used		
		4.3 Documentation Maintenance is performed.		

END



Further information to the trainer

Books

Doglio, F. (2018). *Rest API Development with Node.js*. Uruguay: Apress.

Mardan, A. (2018). *Practical Node.js*. California: Apress.

Mike Cantelon, M. H. (2014). *Node.js IN ACTION*. Shelter Island, NY11964: Manning Shelter Island.

Web Links

Adaptable. (2024). *deploy-nodejs-app*. Retrieved from adaptable: <https://adaptable.io/docs/app-guides/deploy-nodejs-app>

Dizdar, A. (2022, July 27). *unit-testing-in-nodejs*. Retrieved from Bright: <https://brightsec.com/blog/unit-testing-in-nodejs/>

Ebere, N. S. (2022, March 3). *how-to-build-a-backend-application*. Retrieved from freecodecamp: <https://www.freecodecamp.org/news/how-to-build-a-backend-application>

learnthinkcreatecode. (2022, October 25). *How to test a Node.js REST API using postman*.

Nael, D. (2019, March 11). Retrieved from okta: <https://developer.okta.com/blog/2019/03/11/node-sql-server>

YouTube. (2020, September 13). *watch?v=VStXlFxQgZg*. Retrieved from youtube: <https://www.youtube.com/watch?v=VStXlFxQgZg>

YouTube. (2022, October 25). *watch?v=zLbsM_n1H9w*. Retrieved from youtube: https://www.youtube.com/watch?v=zLbsM_n1H9w

