Republic of Rwanda
Ministry of Education

RTB | RWANDA TVET BOARD

NITIW401

IoT WEB APPLICATION DEVELOPMENT USING PHP

Develop IoT Web Application Using PHP

Competence

RQF Level: 4

Learning Hours 100

Credits: 10

Sector: ICT and Multimedia

Trade: Network and Internet Technologies

Module Type: Specific

Curriculum: ICTNIT4001 TVET Certificate IV in Networking and Internet Technologies

1200

Issue Date: September, 2023

Purpose statement	This specific module describes the skills, knowledge and attitude required to develop an IoT web application. This module is intended to prepare student pursuing TVET Level 4 in Networking and Internet Technologies. Upon completion of this module, the learner will be able to develop an API using PHP, develop User Interface, Integrate API Endpoints with User Interface, develop a web application and document an IoT web application using PHP.					
Learning assumed to be in place	▪ Database Development					
Delivery modality	**Training delivery**		**100%**	**Assessment**		**Total 100%**
	Theoretical content		20%	Formative assessment	20%	50%
	Practical work:		80%		80%	
	Group project and presentation	30%				
	Individual project /Work	50%				
			Summative Assessment			50%

Elements of Competence and Performance Criteria

Elements of competence	Performance criteria
1. Develop API using PHP	1.1 Web application requirements are properly analysed based on organization requirements
	1.2 PHP environment is appropriately prepared based on web application requirements
	1.3 PHP concepts are properly applied based on PHP standards
	1.4 API CRUD endpoints are properly developed based on IoT system requirements.
	1.5 API endpoints are properly secured based on PHP data security standards.
	1.6 API endpoints are effectively tested based on IoT system requirements
	1.7 Application Programming Interface (API) are properly documented according to its functionality
2. Develop user interface	2.1 User Interface (UI) requirements are properly identified based on IOT system requirements
	2.2 HTML tags are appropriately used based on wireframe
	2.3 CSS is properly applied to the HTML tags based on system layout
3. Integrate API Endpoints with User Interface	3.1 Application Programming Interface (API) endpoints are correctly interpreted according to API documentation.
	3.2 Application Programming Interface (API) data is properly presented on the user interface in line with http communication protocol
	3.3 IOT web application is effectively documented based on system functionality

Intended Knowledge, Skills and Attitude

Knowledge	Skills	Attitude
✓ Describe PHP concepts ✓ Identify system requirements ✓ Describe API concepts ✓ Knowledge of CRUD ✓ Identify UI requirements ✓ Security Practices	✓ Install software ✓ Develop and secure Application Programming Interface Endpoints ✓ Develop UI ✓ Display API data on User Interface ✓ Interpret API Endpoints ✓ Document IoT Web Application	✓ Exploratory ✓ Critical thinking ✓ Innovativeness ✓ Persistence ✓ Self confidence ✓ Team work spirit ✓ Adaptability

Course content

Learning outcomes	**At the end of the module the learner will be able to:** 1. Develop Application Programming Interface (API) using PHP. 2. Develop user interface 3. Integrate Application Programming Interface (API) Endpoints with User Interface

Learning outcome 1: Develop Application programming interface (API) using PHP.	**Learning hours: 40**

Indicative content

- **Analysis of IoT Web application requirements**
  - Identify system requirement.
    - Collect the data.
    - Select the required devices while developing and displaying
    - Identify source of data (from sensors, from database, from APIs)
    - Describe types of system architecture
  - Functional requirements
  - Non-functional requirements
  - Code management
    - Version control and collaboration
    - Automation and continuous delivery
    - Code minimization and optimisation
- **Prepare PHP environment**
  - Description of key terms
    - PHP (purpose, characteristics)
    - Interpreter
    - Compiler
    - Open Source
    - Web Server
    - Database (DBMS, MySQL, NoSQL)

        - Browser
        - Text Editor
        - Integrated Development Environment(IDE)
    - Installation of environment tools
        - Install Apache, mysql, php (xampp, mamp, wamp,lamp)
        - Install Browser
        - Install Text Editor
        - Install Integrated Development Environment (IDE)
    - Configure Apache, MySQL, php (xampp, mamp, wamp,lamp)
    - Test the environment using phpinfo() function

- **Identification of PHP concepts**
    - Tags
    - Variable
        - Naming rules
        - Types
        - Scopes
    - Super Global variables
    - Operators
    - Data types
    - Variable scope
    - Constants
    - Comment
    - String concatenation
    - Control structures
        - Conditional statements
        - Iterating statement

- Switch statement
- Jump statements

✓ Arrays

- Indexed array
- Associative array
- Multidimensional array

✓ Functions

- Built-in functions
- User-defined functions
- Anonymous functions
- Calling function

✓ File handling

- Opening a file
- Reading a file
- Writing a file
- Closing a file
- Deleting a file

✓ Data Validations

- **Develop Application Programming Interface CRUD Endpoints**

✓ Description of Application Programming Interface concepts

- Definition
- Types (Representational state Transfer (REST), Simple Object Access Protocol(SOAP)
- API working principles

  - Request methods
  - Endpoint
- ✓ Application of data formats (JSON)
- ✓ Execute CREATE query
  - Register IoT device
  - Validate data from IoT device
  - Store data in database
- ✓ Execute READ Query
  - Retrieve device information
  - Fetch data from external Application Programming Interface
  - Check device status
  - Access alert records
- ✓ Execute UPDATE Query
  - Update device information
  - Modify sensor settings
  - Update device firmware
  - Update access permission
  - Update alert and notification settings
- ✓ Execute DELETE Query
  - Remove sensor data
  - Delete command history
  - Purge inactive devices
  - Delete alert configurations
- ✓ Error Handling

- **Secure API endpoints**
  - ✓ Implementing Application Programming Interface authentication
  - ✓ Apply Application Programming Interface security Best practices
  - ✓ Managing Application Programming Interface security
- **Test API endpoints**
  - ✓ Description of Application Programming Interface testing concepts
    - Testing tools (Postman, Sandbox, PHPUnit, Guzzle, Pest, Dredd-PHP)
    - API testing best practices
    - Benefits of Application Programming Interface testing
    - Testing steps
  - ✓ Types of Application Programming Interface endpoints testing
    - Unit testing
    - Integration testing
    - End to end testing
    - Load testing
  - ✓ Testing of Endpoint accessibility
- **Documentation of the developed API**
  - ✓ Choose documentation tool (Postman/swagger/ApiDoc/Spotlight)
  - ✓ Document API specifications(information)
  - ✓ Document Application Programming Interface Endpoints
    - Include authentication information
    - Include Request parameters
    - Include Response structure
    - Include Error codes and error response formats

✚ Add usage guidelines ✓ Document Application Programming Interface Versioning	
Resources required for the learning outcome	
Equipment	▪ Computer ▪ Sensors ▪ Arduino boards ▪ Node MCU boards ▪ Raspberry pi
Materials	▪ Electricity ▪ Internet
Tools	▪ XAMPP ▪ WAMP ▪ LAMP ▪ MAMP ▪ DBMS ▪ MSQL ▪ NoSQL Postman ▪ Swagger ▪ ApiDoc ▪ Spotlight ▪ Sandbox ▪ PHPUnit ▪ Guzzle ▪ Behat ▪ Pest ▪ Dredd-PHP ▪ Google chrome ▪ visual studio

	▪ Sublime text ▪ Notepad++
Facilitation techniques or Learning activity	▪ Demonstration ▪ Individual and group work ▪ Practical exercise ▪ Group discussion
Formative assessment methods /(CAT)	▪ Written assessment ▪ Oral presentation ▪ Practical Assessments

Learning outcome 2: Develop user interface	**Learning hours: 25**
Indicative content	
● **Identification of UI requirements** ✓ Review System requirements ✓ List key user services ✓ Define user personas ✓ Description of user experience (UX) best practices ✚ Define user interface ✚ Clear and Intuitive Navigation ✚ Consistent Design ✚ Use Familiar UI Patterns ✚ Choose an attractive colour scheme ✓ Features of a good user interface ✓ Draw Wireframe ● **Use HTML tags**	

- ✓ Description of HTML
  - Definition
  - Versions
  - Tag attributes
- ✓ HTML tag categories
  - HTML Structural tags
  - Formatting tags
  - Table tags
  - Form tags
  - Heading tags
  - List tags
  - Media tags
  - Code tags
  - HTML frame tags
  - HTML Comment
  - Grouping tags (div, span)
  - Hyperlink tag
  - Semantic tags

- **Application of CSS**
  - ✓ Description
    - Definition
    - Versions
    - Style types
  - ✓ Use CSS Visual rules

  - Font
  - Colors and background colors
  - Opacity
  - Background image
- ✓ Use CSS Display and positioning
  - Relative
  - Absolute
  - Fixed
  - Sticky
  - Block
- ✓ Use CSS Box and Grid model
  - Height and width
  - Borders
  - Border radius
  - Padding
  - Margin
  - Visibility
  - Auto

Resources required for the indicative content	
Equipment	▪ Computer ▪ Projector
Materials	▪ Electricity ▪ Internet
Tools	▪ Text editor (Notepad, Notepad++, Sublime text) ▪ Web browser

Facilitation techniques or Learning Activity	■ Group discussion ■ Demonstration ■ Practical exercise ■ Individual work
Formative assessment methods /(CAT)	■ Written assessment ■ Oral presentation ■ Practical Assessments

Learning outcome 3: Integrate API Endpoints with User Interface	**Learning hours: 35**
Indicative content	
● **Interpret API Endpoints** ✓ Read description of each endpoint HTTP method ✓ Check URL structure ✓ Check Request parameters ✓ Check the structure of API responses ✓ Check if the API requires authentication ✓ Check error codes and error response format. ● **Use API data on User Interface** ✓ URL requests ✚ Endpoints ✚ Method ✚ Headers ✚ Data (or body)	

        - Status (code and message)
    - ✓ HTTP requests
        - POST requests to create records
        - GET request to read or get a resource from the server
        - PUT and PATCH requests to update records
        - DELETE request to delete a resource from a server
    - ✓ Testing and Debugging
    - ✓ Data rendering
        - Show Loading States and Progress Indicators
        - Preserve Filtering and Sorting
        - Preserve Pagination and Infinite Scrolling
        - Ensure responsiveness design
- **Document IoT Web Application**
    - ✓ User guide documentation
    - ✓ Report on work done
    - ✓ Provide Bills of quantities

Resources required for the indicative content

Equipment	▪ Computer ▪ Projector
Materials	▪ Internet ▪ Electricity
Tools	▪ Text editor

	▪ Webserver ▪ Web browser ▪ DBMS
Facilitation techniques or Learning activity	▪ Demonstration ▪ Group discussion ▪ Practical exercise ▪ Individual work
Formative assessment methods /(CAT)	▪ Written assessment ▪ Oral presentation ▪ Practical Assessments

Integrated/Summative assessment

Integrated situation

ZOMBI Farmers ltd is a cooperative found in Rwanda, Eastern region, Nyagatare District. These farmers grow Maize. Due to prolonged dry seasons that lead to poor yields, the cooperative members adopted an irrigation system that operates manually in shifts. Recently the members realized that some members dodge their shifts which lead to drying of crops due to irregular irrigation. The cooperative decided to hire an IoT Web application developer to develop for them a web application for their irrigation system that will help them to monitor and control the soil moisture and temperature. As an IoT web Application developer, you are required to perform the following task.

1. Create API endpoints to:

   a. Consume weather parameters from irrigation system equipment.

   b. Store soil information to the database

c. Represent data on the user interface.

d. If the soil moisture levels go below 50% the water pump should start watering automatically.

e. Alert the farm manager about soil moisture level through a telephone message

f. Create a configuration interface that will allow user to set up irrigation parameters (soil moisture< = 500 for normal condition and soil moisture>500 for starting watering)

Information:

Provided is a database design.

Database name: ZOMBI

Tables:

FarmData: fd_Id (pk), Moisture_level, temp _degree, date_time

Farmer: f_id(pk), first_name, last_name, sex, phone_number, fd_Id(fk)

This work should be done in 12 hours

Resources	
Equipment	▪ Computer
Tools	▪ Web server ▪ Text editor / IDE ▪ Web browser ▪ Testing tool(telephone) ▪ API Documentation tool

Assessable outcomes	**Assessment criteria (Based on**	**Indicator**	**Observation**		**Marks allocation**
			Yes	**No**	

	performance criteria)				
Learning outcome 1: Develop API using PHP **(30%)**	**1.1:** API CRUD endpoints are properly developed based on IoT system requirements.	**1.** CREATE query is executed			**4**
		2. READ Query is executed			**4**
		3. UPDATE Query is executed			**4**
		4. DELETE Query is executed			**4**
	1.2: API endpoints are properly secured based on PHP data security standards.	**1.** Authentication is implemented			**4**
		2. Security is implemented			**3**
		3. Security is managed			**3**
	1.3: API endpoints are effectively tested based on IOT system requirements	**1.** Testing tool is selected			**3**
		2. Point to point test is performed			**4**
		3. Endpoint access test is performed			**4**
	1.4: API are properly documented according to its functionality	**1.** Documentation tool is selected			**2**
		2. API functionality is specified			**2**

		3. Endpoints are included			**2**
		4. API Version is indicated			**2**
Learning outcome 2: Develop user interface **(40%)**	**2.1:** UI requirements are properly identified based on IOT system requirements	**1.** Attractive colours are selected			**3**
		2. key user services are identified			**3**
		3. Wireframe is drawn			**4**
	2.2: HTML tags are appropriately used based on wireframe	**1.** Formatting tags are used			**4**
		2. Code tags are used			**4**
		3. Semantic tags are used			**4**
	2.3: CSS is properly applied to the HTML tags based on system layout	**1.** Colours and images are used			**3**
		2. CSS display and positioning techniques are applied			**4**
		3. CSS box or grid model is used			**4**
Learning outcome 3:	**3.1:** API data is properly presented	**1.** create record requests are performed			**3**

Integrate API Endpoints with User Interface **(30%)**	on the user interface in line with http communication protocol	**2.** Resource is accessed from server			**4**
		3. PUT and PATCH operations are performed			**4**
		4. Delete operations are performed			**3**
	3.2: IoT web application is effectively documented based on system functionality	**1.** user guide is written			**3**
		2. Report is provided			**3**
		3. Bill of quantities is provided.			**3**
Total marks		**100**			
Percentage Weightage		**100%**			
Minimum Passing line % (Aggregate): 70%					

References

author, N. ((n.d.)). (N o author. (n.d.). How to integrate API. MindSphere. https://documentation.mindsphere.io/MindSphere/howto/howto-integrate-api.html, (n.d.))No author. (n.d.). PHP for Beginners. Google Books. , (n.d.))https://www.google.rw/books/edition/.

author, N. ((n.d.)). 3.No author. (n.d.). API Gateway mapping template reference. Amazon Web Services. https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-mapping-template-reference.html.

author, N. ((n.d.)). 4.(No author. (n.d.). How to integrate API. MindSphere. https://documentation.mindsphere.io/MindSphere/howto/howto-integrate-api.html, (n.d.))No author. (n.d.). PHP for Beginners. Google Books. .

author, N. ((n.d.)). 4.No author. (n.d.). How to integrate API. MindSphere. https://documentation.mindsphere.io/MindSphere/howto/howto-integrate-api.html.

author, N. ((n.d.)). 5.No author. (n.d.). 4 IoT device application requirements - normative section. GSMA. https://www.gsma.com/iot/4-iot-device-application-requirements-normative-section/.

author, n. (n.d.). 4.(no author. (n.d.). How to integrate API. MindSphere. https://documentation.mindsphere.io/MindSphere/howto/howto-integrate-api.html, (n.d.))No author. (n.d.). PHP for Beginners. Google Books. .

author., N. (n.d.). 7. No author. (n.d.). How to secure API endpoints: 9 tips and solutions. Nordic APIs. https://nordicapis.com/how-to-secure-api-endpoints-9-tips-and-solutions/.

Kumar, S. P. ((April, 2018)). 2.S Praveen Kumar. (April, 2018). IoT-based temperature and humidity monitoring system using Raspberry Pi. ResearchGate. https://www.researchgate.net/publication/325172192_IoT-based_Temperature_and_Humidity_Monitoring_System_using_Raspberry_pi.

Schlossnagle, G. ((2004)). 1.Schlossnagle, G. (2004). Advanced PHP Programming. USA: Sams. Retrieved 8 6, 2023, from https://ptgmedia.pearsoncmg.com/images/9780672325618/samplepages/0672325616.pdf.

D9