Republic of Rwanda
Ministry of Education

x

RTB | RWANDA TVET BOARD

WEB APPLICATION DEVELOPMENT USING JAVASCRIPT

SPEWJ302

Develop a Web Application Using JavaScript

Competence

RQF Level: 3

Learning Hours 130

Credits: 13

Sector: ICT AND MULTIMEDIA

Trade: Software Programming and Embedded Systems

Module Type: Specific

Curriculum: TVET CERTIFICATE 3 in Software Programming and Embedded Systems

Issue Date: August 2022

<table>
<tr><td>Purpose statement</td><td colspan="6">Dynamic Website Development Using JavaScript is a module which prepares the students to develop dynamic and interactive websites. In this module, students will use the basics of JavaScript to modify the look and feel of the website, they will use advanced features of JavaScript to work with graphics, animations and backend JavaScript. At the end of this module, students will master JavaScript DOM and Event Handlers, JavaScript for Data Validation, JavaScript OOP and JavaScript Frameworks. This module provides the students with the ability to apply the principles of the user interface, and thus prepares them to become JavaScript/Web UI developers.</td></tr>
<tr><td rowspan="6">Delivery modality</td><td colspan="2">Training delivery</td><td>100%</td><td colspan="2">Assessment</td><td>Total 100%</td></tr>
<tr><td colspan="2">Theoretical content</td><td>40%</td><td rowspan="4">Formative assessment</td><td>40%</td><td rowspan="4">50%</td></tr>
<tr><td colspan="2">Practical work:</td><td rowspan="3">60%</td><td rowspan="3">60%</td></tr>
<tr><td>● Group project and presentation</td><td>20%</td></tr>
<tr><td>● Individual project /Work</td><td>40%</td></tr>
<tr><td colspan="3"></td><td colspan="2">Summative Assessment</td><td>50%</td></tr>
</table>

Elements of Competency and Performance Criteria

Elements of competency	Performance criteria
1. Use Fundamental Features of JavaScript	1.1 Basic Features of JavaScript are properly used
	1.2 Browser DOM is properly manipulated using JavaScript
	1.3 JavaScript Libraries are effectively applied
2. Create Backend	2.1 Nodejs fundamentals are clearly described
	2.2 RESTful API's are properly created Using Express
	2.3 Automated Testing of RESTful API's is appropriately built
3. Build Frontend	3.1 ReactJs basics are appropriately described
	3.2 Forms data are properly handled and HTTP Requests are properly sent in ReactJs

	3.3 ReactJs hooks, redux and Automated Testing are properly described

Course content	
Learning outcomes	**At the end of the module the learner will be able to:** 1. Use Fundamental Features of JavaScript 2. Create Backend 3. Build Frontend

Learning outcome 1: Use Fundamental Features of JavaScript	Learning hours: 40

Indicative content

- **Use of JavaScript Basic Features**

✓ **Usage of JavaScript in Web Browser**

- JavaScript in HTML
- External JavaScript
- JavaScript Code in Comment

✓ **Usage of Data Types and Variables**

- JavaScript Data Types
- JavaScript Variables
- JavaScript Literals

✓ **Usage of JavaScript Toolboxes**

- Alert Toolbox
- Prompt Toolbox
- Confirm Toolbox

✓ **Usage of JavaScript Functions**

- User-defined Functions
- Function Call
- JavaScript Built-in Functions

✓ **Usage of JavaScript Control Statements**

- if---else Control Statement
- Loop Statements

- Switch---case Statement

✓ **Usage of JavaScript Event Handling**

- Keyboard and Mouse Events
- Drag and Drop Events
- Input and Load Events

✓ **Usage of OOP in JavaScript**

- Data Encapsulation, Class and Constructor
- Extends, super, getters and setters
- Function Prototypes
- Inheritance and Polymorphism

✓ **Usage of JavaScript Objects**

- JavaScript Native Objects
- User-Defined Objects
- This, with, new and let in JavaScript

✓ **Usage of Data structures, modern operators and Strings**

- Destructuring Arrays
- Destructuring Objects
- The Spread Operator (...)
- Set and Maps: Iteration
- Data Transformations: map, filter, reduce

- **Manipulation of Browser DOM using JavaScript**

✓ **Usage of Document Object Model (DOM)**

- DOM Tree
- DOM Functions
- JavaScript DOM for HTML and CSS
- Inner HTML and Inner Text Properties

✓ **Usage of Advanced DOM and events**

- Selecting, Creating, and Deleting Elements
- Types of Events and Event Handlers
- DOM Traversing
- Building a Tabbed Component
- The Intersection Observer API
- Lifecycle DOM Events
- Efficient Script Loading: defer and async

- **Use of JavaScript Libraries**

- ✓ **Usage of ajax and json**
  - AJAX - Synchronous vs Asynchronous
  - AJAX - Technologies
  - AJAX - Action
  - AJAX - XMLHttpRequest
  - AJAX - Database Operations
  - AJAX - Security
  - AJAX - Issues
  - JSON vs XML
  - JSON format
  - JSON vs BSON
- ✓ **Usage of jquery**
  - JQUERY BASIC
  - JQUERY EFFECTS
  - JQUERY MANIPULATION
  - JQUERY ADVANCED
- ✓ **Usage of the Geolocation API**
  - Interfaces
  - Properties

Resources required for the learning outcome	
Equipment	▪ Computer
Materials	▪ Internet ▪ Didactic materials
Tools	▪ Text editors
Facilitation techniques	▪ Individual and group work ▪ Trainer guided ▪ Group discussion ▪ Practical works
Formative assessment methods	▪ Written assessment ▪ Performance assessment ▪ Presentations

Learning outcome 2: Create Backend	Learning hours: 45

Indicative content

- **Description of NodeJs fundamentals**
  - ✓ Node modules
    - Node Architecture
    - Types of modules
    - Creating a module
    - Loading a module
  - ✓ Usage of Node Package Manager
    - Usage of NPM
    - Semantic versioning
    - Publishing a custom package
  - ✓ Usage of Asynchronous in JavaScript
    - Synchronous vs Asynchronous
    - Callbacks
    - Promises
    - Async and await
- **Creation of RESTful API's using Express**
  - ✓ Introduction of RESTFul services
    - RESTFul services
    - Introducing Express
    - Building web server
    - Nodemon and Environment variables
    - Routing paramaters
    - Handling HTTP GET requests
    - Handling HTTP POST requests
    - Handling HTTP PUT and DELETE requests
    - Using postman to call endpoints
    - Validation of inputs
  - ✓ Description of Advanced Express concepts
    - Middleware concept
    - Environment and profile setting
    - Configurations externalization
    - Debugging setup on different profiles
    - Database integration
    - Authentication and authorization
    - Handling and Logging errors
    - Documenting Endpoints with Swagger-ui
  - ✓ Usage of Mongoose in Databases operations

- Connecting to MongoDB
- Schemas and Models
- NoSQL query operations
- Pagination
- Data validations
- Modelling relationship in NoSQL

- **Building Automated Tests of RESTful API's**

✓ Building of Unit tests

- Automated Testing
- Benefits of Automated Testing
- Types of Tests
- Tooling and Test Pyramid
- Writing Unit Tests
- Grouping Tests
- Mocking

✓ Building of Integration Tests

✓ Description of Test-Driven Development

- Describe TDD
- Write Functionalities from Tests

Resources required for the indicative content	
Equipment	▪ Computer
Materials	▪ Internet ▪ Didactic materials
Tools	▪ Text editors
Facilitation techniques	▪ Individual and group work ▪ Trainer guided ▪ Group discussion ▪ Practical works
Formative assessment methods	▪ Written assessment ▪ Performance assessment ▪ Presentations

Learning outcome 3: Build Frontend	Learning hours: 45

Indicative content

- **Description of ReactJs Basics**
  - ✓ Description of React Basics and working with components
    - How react works
    - JSX in react
    - Components in react
    - Outputting dynamic data and working with expressions
    - Passing data via props
    - The concept of composition of components (children props, connecting multiple components)
  - ✓ Description of React state and working with events
    - Listening to events and working with event handlers
    - Working with state
    - Listening to user input
    - Working with multiple states
    - Updating state that depends on previous state
    - Handling form submission
    - Adding two-way binding
    - Child-to-parent component communication(bottom-up)
    - Lifting the state up
    - Controlled vs uncontrolled components, stateless vs stateful components
  - ✓ Description of how to Render Lists and conditional content
    - Rendering list of data
    - Using stateful list
    - Understanding keys
    - Out putting conditional content
    - Adding Conditional Return Statements
    - Adding Dynamic Styles
  - ✓ Description of how to debug a react app
    - Understanding React Error Messages
    - Analyzing Code Flow & Warnings
    - Working with Breakpoints
    - Using the React DevTools
  - ✓ Description of how to work with fragments, portals and refs
    - JSX Limitations & Workarounds
    - Creating a Wrapper Component
    - React Fragments
    - Working with Portals
    - Working with "ref"s
    - Controlled vs Uncontrolled Components

- **Handling of forms data and Sending of HTTP requests in ReactJs**
  - ✓ Sending of Http request
    - Sending a GET Request
    - Handling Loading & Data States
    - Handling Http Errors
    - Using useEffect() For Requests
    - Sending a POST Request
    - Using third party library to call backend axios
  - ✓ handling forms and user inputs
    - Dealing with Form Submission & Getting User Input Values
    - Adding Basic Validation
    - Providing Validation Feedback
    - Handling the "was touched" State
    - React to Lost Focus
    - Managing the Overall Form Validity
    - Adding A Custom Input Hook
    - Applying Our Hook & Knowledge to A New Form
  - ✓ Building a multi-page SPA(Single Page Application) using React router
    - What is Routing & Why?
    - Installing React Router
    - Defining & Using Routes
    - Working with Links
    - Using NavLinks
    - Adding Dynamic Routes with Params
    - Using "Switch" and "exact" For Configuring Routes
    - Working with Nested Routes
    - Redirecting the User
    - Adding a Layout Wrapper Component
- **Description of ReactJs hooks, redux and Automated Testing**
  - ✓ Handling side effects using reducers and context APIs
    - What are "Side Effects" & Introducing use Effect
    - Using the use Effect () Hook
    - Using the use Effect Cleanup Function
    - Using the use Reducer () Hook
    - Use Reducer & use Effect
    - Use Reducer vs use State for State Management
    - Using the React Context API
    - React Context Limitations
    - Tapping into Context with the use Context Hook
    - Diving into "Forward Refs"
  - ✓ Description of Optimization techniques
    - Child Component Re-Evaluation
    - Preventing Unnecessary Re-Evaluations with React.memo()
    - Preventing Function Re-Creation with use Callback ()

- Use Callback () and its Dependencies
- Understanding State Scheduling & Batching
- Optimizing with useMemo()

✓ Usage of redux in react
- Providing the Store
- Using Redux Data in React Components
- Dispatching Actions from Inside Components
- Working with Multiple State Properties
- How to Work with Redux State Correctly
- Redux Challenges & Introducing Redux Toolkit
- Adding State Slices
- Connecting Redux Toolkit State
- Working with Multiple Slices
- Reading & Dispatching from A New Slice
- Using use Effect with Redux
- Handling Http States & Feedback with Redux
- Using an Action Creator Thunk

✓ Proper building of Automated Testing in react apps
- Writing Our First Test
- Grouping Tests Together with Test Suites
- Testing User Interaction & State
- Testing Connected Components
- Testing Asynchronous Code
- Working with Mocks

✓ Deploying react application to production
- Deployment Steps
- Adding Lazy Loading
- Building the Code for Production
- Exploring Routing Issues & Finishing Deployment

Resources required for the indicative content	
Equipment	▪ Computer
Materials	▪ Internet ▪ Didactic materials
Tools	▪ Text editors
Facilitation techniques	▪ Individual and group work ▪ Trainer guided ▪ Group discussion ▪ Practical works

Formative assessment methods	▪ Written assessment ▪ Performance assessment ▪ Presentations

Integrated/Summative assessment

Integrated situation

Nyabihu TVET is a school located in the Northern Province. The school's compound consists of 3 separate buildings: administration offices, classrooms, and dining room, and 2 open spaces: a garden, and a sports area.

There is a problem that guests don't have guidance to the TVET area and sometimes can't find or locate some places in that area. For example, recently, a guest came and wanted to know where the dining room is, but they could not locate it by themselves.

TVET administration wants to hire a JavaScript developer to design and implement a solution that will be using a simple map to show all the buildings and open spaces with their location in the school where the user will have the ability to view and select each area at once on a screen. For example, the user can select the dining room by hovering (using a mouse cursor) on it, and the area will change the color to "RED". Each area will be viewed using a different color.

If you have ever used Google Maps to select and zoom in on a location and change your view mode, you were using features that were built with JavaScript. JavaScript's ability to create dynamic objects makes it a natural fit for creative interactive maps on websites or in a web app. The goal is not to recreate Google Maps, but create a simpler map built in Javascript framework jQuery (a collection of Javascript libraries with pre-written, reusable code). This project will help you to familiarize with Javascript, especially map-making capabilities.

Working Time: 3 hours.

Tools to use: Windows Operating System, Bootstrap, Geolocation API

Resources

Tools	Windows Operating Systems Internet Bootstrap Text Editor
Equipment	Computer
Materials/ Consumables	JavaScript HTML jQuery CSS Geolocation API

Assessable outcomes	Assessment criteria (Based on performance criteria)	Indicator	Observation		Marks allocation
			Yes	No	
Learning outcome 1: Use Fundamental Features of JavaScript	JavaScript Libraries are effectively applied	The picture is well displayed and partitioned depending on the areas			10
		The menus of all areas are created			10
		The color of each area is displayed			10
		Selection controls and looping controls are used			10
Learning outcome 2: Create Backend	Automated Testing of RESTful API's is appropriately built	GET,POST,DELETE and PUT endpoints are built			10
		Unit, Integration and End-to-end Tests are built			10
		Project APIs are documented using Swagger-ui			10
Learning outcome 3: Build Frontend	Forms data are properly handled and HTTP Requests are properly sent in ReactJs	Backend APIs are consumed and result frontend			10
	ReactJs hooks, redux and Automated Testing are properly described	All areas are sectioned			10
		The menu matches the area section			10
Total marks					**100**
Percentage Weightage					**100%**
Minimum Passing line % (Aggregate): 70%					

Reference books:

Books:

[1]Mark J. Collins, Pro HTML5 withCSS, JavaScript, and Multimedia, 2017
[2]Eric Freeman & Elisabeth Robson, Head First JavaScript Programming, 2014

Websites:

[1]"JavaScript Tutorial", W3schools.com, 2018. [Online]. Available: https://www.w3schools.com/js/. [Accessed: 29- Dec- 2018].
[2]"Learn JavaScript Tutorial - javatpoint", www.javatpoint.com, 2018. [Online]. Available: https://www.javatpoint.com/javascript-tutorial. [Accessed: 29- Dec- 2018].