Republic of Rwanda
Ministry of Education

RTB | RWANDA TVET BOARD

SWDBD 401

BACKEND APPLICATION DEVELOPMENT

Develop a Backend Application using Node Js

Competence

RQF Level:	4	Learning Hours 100
Credits:	10	
Sector:	ICT and Multimedia	
Trade:	Software Development	
Module Type:	Specific	
Curriculum:	ICTSWD4002 – TVET Certificate IV in Software Development	
Copyright:	© Rwanda TVET Board, 2023	

Issue Date: September 2023

Purpose statement	This module describes the skills, knowledge and attitude required to develop a backend application using NodeJS. This module is intended to prepare students pursuing TVET Level 4 in Software Development. At the end of this module the student will be able to develop RESTFUL APIs with Node JS, secure, test and manage backend application					
Learning assumed to be in place	▪ Apply Network Fundamentals ▪ Develop Database ▪ Backend Application Design ▪ Apply Data Structure and Algorithm Fundamentals using JavaScript					
Delivery modality	**Training delivery**		**100%**	**Assessment**		**Total 100%**
	Theoretical content		**30%**	**Formative assessment**	**30%**	**50%**
	Practical work:		**70%**		**70%**	
	Group project and presentation	**20%**				
	Individual project /Work	**50%**				
				Summative Assessment		**50%**

Elements of Competence and Performance Criteria

Elements of competence	Performance criteria
1.Develop RESTFUL APIs with Node JS	1.1 Development environment is properly arranged based on coding architecture methodology
	1.2 Server and database connection are properly established according to development environment
	1.3 RESTFUL APIs are effectively implemented based on backend functionalities
2.Secure Backend Application	2.1 Data encryption is correctly applied based on system security
	2.2 Third-party libraries are carefully checked based on system security
	2.3 User Authentication, Authorization and Accountability (AAA) are carefully applied based on NPM Universal Access Control (UAC)
	2.4 Environment variables are carefully Secured according to system security
3. Test Backend Application	3.1 Unit tests are appropriately conducted based on software testing techniques
	3.2 Usability is correctly tested according to expected results
	3.3 Security is properly tested based on system threats
4.Manage Backend Application	4.1 Application is appropriately deployed based on FURPS requirements
	4.2 Backend is effectively maintained according to the system Functionalities
	4.3 Application documentation is properly generated according to the system backend

Course content

Learning outcomes	At the end of the module the learner will be able to:
	1. Develop RESTFUL APIs with Node JS 2. Secure Backend Application 3. Test Backend Application 4. Manage Backend Application

Learning outcome 1: Develop RESTFUL APIs with Node JS	Learning hours: 45

Indicative content

- **Setup Node. Js Environment**
  - ✓ Description of Node.js Key Concepts
    - Node.Js
    - Routes
    - NPM
    - Express Js
    - Backend Application
    - Class
    - Object
    - Method
    - Properties
    - Dependencies
    - APIs
    - Postman

        - Nodemon
        - DBMS (SQL Based, NoSQL Based)
    - ✓ Installation of Node Js Modules and packages
        - Node.Js and NPM
        - Express Js
        - Postman
        - Nodemon
    - ✓ Configuration of MySQL Server

- **Connection of Node Js to the ES5 or ES6 server**
    - ✓ Creation of basic server with Express Js
    - ✓ Application of Client Libraries
        - HTTP
        - HTTPs
        - Axios
        - Request
    - ✓ Establishment of server connection
        - Setup Connection parameters
        - Create / send Request
        - Handle the response
    - ✓ Test of Server Connection

- **Establishment of database connection**
    - ✓ Create Database
    - ✓ Schema Setup
    - ✓ Configure Database Connection

- ✓ Test Database Connection

- **Develop RESTFUL APIs**
  - ✓ Define endpoints and HTTP Methods
    - Create POST End Point
    - Create all Items GET endpoint
    - Create specific ID GET endpoint
    - Create PUT endpoint
    - Create DELETE endpoint
  - ✓ Implementation of API endpoints
  - ✓ Use of Middleware services
    - Types of middleware services
    - Error Handling
    - Logging
    - Input validation
  - ✓ Perform CRUD operations using MySQL Database
  - ✓ Use HTTP Status code

- **Debugging RESTFUL APIs**

Resources required for the learning outcome	
Equipment	▪ Computer
Materials	▪ Internet ▪ Books ▪ Tutorials ▪ Code samples ▪ Online communities

Tools	▪ Browser ▪ NodeJS ▪ ExpressJs IDE ▪ Text Editor ▪ Node Packages ▪ MySQL Workbench ▪ Postman ▪ Swagger ▪ MochaNodemon
Facilitation techniques	▪ Brainstorming ▪ Group Discussion ▪ Jig Saw ▪ Demonstration Visual Aids
Formative assessment methods /(CAT)	▪ Written assessment ▪ Performance ▪ Oral assessment

Learning outcome 2: Secure Backend Application	**Learning hours: 20**
Indicative content	
• **Data encryption in securing RESTFUL APIs** ✓ Introduction to data encryption ⁜ Types of data encryption ⁜ Encryption techniques ⁜ Benefits and importance of data encryption ✓ Steps in securing RESTFUL APIs ⁜ 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

- **Integrating and Using Third-Party Libraries**
  - ✓ Installing Node Js Package Manager (NPM)
  - ✓ Incorporating common Node.js third-party libraries
    - Express
    - Lodash
    - Moment.js
  - ✓ Interacting with third-party libraries
    - Callbacks
    - Promises
    - async/await
- **Maintaining and Updating Third-Party Libraries**
  - ✓ Monitoring of library dependencies and version numbers
    - Package. Json
    - Npm-shrinkwrap. json
  - ✓ Checking for library updates and security vulnerabilities using tools
    - NPM outdated
    - NPM audit
    - Snyk
  - ✓ Updating third-party libraries safely
    - Versioning
    - semver rules
  - ✓ Strategies for managing and testing library updates
    - staging environments
    - Version control systems.
- **Implementation of Authentication**
  - ✓ Principles of authentication
  - ✓ Role of authentication in system security
  - ✓ Implementing user authentication in Node.js using frameworks
    - Passport

        - JWT (JSON Web Tokens)
        - Social Auth. (Google, Facebook, …)
    - ✓ Using authentication middleware to protect routes and resources
    - ✓ Best practices for password storage and handling sensitive data
- **Implementation of Authorization**
    - ✓ Principles of authorization
    - ✓ Role of authorization in system security
    - ✓ Implementing role-based and attribute-based access control in Node.js
    - ✓ Using authorization middleware to manage user permissions
    - ✓ Implementing custom authorization logic for specific use cases
- **Implementation of Accountability**
    - ✓ Principles of accountability
    - ✓ Roles of Accountability in system security
    - ✓ Implementing logging and auditing features in Node.js using popular libraries
        - Winston
        - Morgan
    - ✓ Logs management
        - Best practices for securely storing log data and protecting it from unauthorized access
        - Audit logs to detect security events and system errors
- **Secure Environment Variables**
    - ✓ Types of information stored in environment variables
        - Database credentials
        - API keys
        - Encryption keys
    - ✓ Potential security risks of storing sensitive information in environment variables
    - ✓ Best practices for managing and securing environment variables in Node.js
    - ✓ Implementing security measures for protecting environment variables
        - Encrypting secrets
        - Decrypting secrets
    - ✓ Storing environment variables in a secure location

- key management service
- a. env file

✓ Management and loading environment variables in Node.js applications using dotenv

✓ Best practices for safely passing environment variables to other services and applications

- **Monitor and Manage Environment Variables**

✓ Implementing logging and auditing features to detect unauthorized access to environment variables

✓ Monitoring changes to environment variables and detecting any suspicious activity

✓ Best practices for managing and rotating environment variables to prevent data breaches

Resources required for the indicative content	
Equipment	▪ Computer
Materials	▪ Internet ▪ Books ▪ Tutorials ▪ Code samples ▪ Online communities
Tools	▪ Browser ▪ Node.Js ▪ Text Editor ▪ Express. Js ▪ Postman ▪ Git ▪ Swagger ▪ Middleware services and libraries
Facilitation	▪ Brainstorming

techniques	▪ Group Discussion ▪ Jig Saw ▪ Demonstration Visual Aids
Formative assessment methods /(CAT)	▪ Written assessment ▪ Performance ▪ Oral assessment

Learning outcome 3: Test Backend Application	**Learning hours: 20**
Indicative content	
• **Implementation of Unit testing** ✓ Introduction to unit tests + Importance of Unit Testing + Unit Testing Process + Unit Testing tools + Frameworks + Libraries ✓ Mocha Testing Framework + Installation and Configuration + Writing Unit tests + Running Tests ✓ Chai assertion library + Installation and configuration + Writing assertions + Chai Expect and Should APIs ✓ Monitor Test results • **Implementation of Usability testing** ✓ Introduction to Usability tests + Importance of Usability Testing + Usability Testing Process + Usability Testing tools	

- ✓ Best Practices for Node.js Security Testing
  - Security Testing Lifecycle
  - Reporting Security Vulnerabilities
  - Remediation and Mitigation
  - Compliance and Regulations
- ✓ Implement of Security Testing in Nodejs
  - Implement Authentication and Authorization
  - Test input validation
  - Use SSL / TLS encryption
  - Test Error Handling
  - Regularly update dependencies
- ✓ Application of Penetration Testing steps
  - Identification scope of the test
  - Gathering API Information
  - Identify Vulnerabilities
  - Perform manual testing
  - Document findings
  - Remediate Vulnerabilities
  - Re-test
- ✓ Perform penetration Testing using OWASP
  - Installation of OWASP tool
  - Perform scan
  - Exploit vulnerabilities
  - Interpret Scan report
  - Document results

Resources required for the indicative content	
Equipment	▪ Computer
Materials	▪ Internet ▪ Books ▪ Tutorials

	▪ Code samples ▪ Online communities
Tools	▪ Browser ▪ Node.Js ▪ Text Editor ▪ Express. Js ▪ Postman ▪ Mocha ▪ Chai ▪ SuperTest ▪ Sinon ▪ Istanbul ▪ Newman
Facilitation techniques	▪ Brainstorming ▪ Group Discussion ▪ Jig Saw ▪ Demonstration Visual Aids
Formative assessment methods /(CAT)	▪ Written assessment ▪ Performance ▪ Oral assessment

Learning outcome 4: Manage Backend Application	Learning hours: 15
Indicative content	

- **Preparation of deployment Environment**
  - ✓ Description of NodeJS application deployment
  - ✓ Types of NodeJS application deployment
    - Manual Deployment
    - Continuous Deployment
    - Docker-based deployment
  - ✓ NodeJS Application Deployment tools
    - NodeJS Runtime
    - Package Manager
    - Operating system
    - Webserver
    - Database
- **Implementation of Manual Deployment of NodeJS application**
  - ✓ Copy the application source code to the server
  - ✓ Installation of dependencies
  - ✓ Start the application using command line
- **Maintenance of NodeJS application**
  - ✓ Best practices for maintenance
    - Update
    - Monitor
    - Perform test
  - ✓ Developing a maintenance plan
    - Identification of maintenance requirements
    - Schedule regular updates
    - Automate maintenance tasks
    - Monitor application performance
    - Test regularly
    - Disaster recovery plan
    - Document changes

- ✓ 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

- **Application of NodeJS Documentation Tools and Frameworks**
  - ✓ Documentation Overview
  - ✓ The importance of documentation
  - ✓ Types of documentation
  - ✓ Overview of popular documentation tools and frameworks
    - Use Swagger/Postman for API documentation
    - Writing clear and concise comments
    - Using documentation generators
  - ✓ Best practices for documentation
  - ✓ Publishing Documentation
    - Options for hosting documentation
    - Using GitHub for collaborative documentation
    - Documentation Maintenance

Resources required for the indicative content	
Equipment	▪ Computer
Materials	▪ Internet
Tools	▪ Browser ▪ Node.Js ▪ Text Editor ▪ Express. Js ▪ Postman ▪ GitHub ▪ Swagger

	▪ OWASP ▪ Webserver ▪ MySQL Workbench ▪ Winston ▪ PM2 ▪ Redis ▪ AWS Lamda
Facilitation techniques	▪ Brainstorming ▪ Group Discussion ▪ Jig Saw ▪ Demonstration Visual Aids
Formative assessment methods /(CAT)	▪ Written assessment ▪ Performance ▪ Oral assessment

Integrated/Summative assessment for Backend Application development

Integrated situation

XM Bakeries is a Bakery Business located in Kigali City, Nyarugenge district, Gitega Sector. It deals in Producing and selling bread to different customers. The Business purchases raw products like Flour, Sugar, Food Color Paste and other ingredients for baking bread. The Sales Manager Records daily sales and inventory information in Microsoft Excel using his laptop. The file system being used does not allow him to Track inventory levels for various products, the system does not allow customers to place orders remotely and Sales reports are not generated automatically as required by the Company management.

The Company has hired you to develop a web application using Node.js and MySQL.

1. The system should enable users to add products theirs price, category and Quantity.
2. The system should enable users to Search for products based on various criteria such as price range, category and Quantity.
3. The system should provide customers the ability to place orders, track product location being delivered, view previously placed orders and purchases report.
4. The system should provide users with the ability to filter and sort the searched results based on their preferences.
5. The system should also allow users to Track inventory levels for various products, generate reports on sales data and inventory levels and allow users to manage customer information in real time

Instructions

- NodeJS shall be deployed using Firebase.
- Use NodeJS Middleware services to handle authentication, input validation and handle errors
- The above tasks should be completed in 8 Hours

Resources

Tools	▪ (Browser, Node.Js, Text Editor, Express. Js, Postman, GitHub, Swagger, OWASP, Webserver, MySQL Workbench, Winston, PM2, Redis, AWS Lamda)
Equipment	▪ Computer
Materials/ Consumables	▪ Internet

Assessable outcomes	**Assessment criteria (Based on performance criteria)**	**Indicator**	**Observation**		**Marks allocation**
			Yes	**No**	
Learning outcome 1: Develop RESTFUL APIs with Node JS **(29%)**	1.1 Development environment is properly arranged based on coding architecture methodology	Ind.1: NodeJS Development tools are selected			2
		Ind.2: NodeJS Environment is setup			3
	1.2 Server and database connection are properly established according to development environment	Ind.1: NodeJS is connected to the server			3
		Ind.2: Database connection is established			3
	1.3 RESTFUL APIs are effectively implemented based on backend functionalities	Ind 1: RESTFUL APIs is serving third party apps.			9

Learning outcome 2: Secure Backend Application **(29%)**	2.1. Data encryption is correctly applied based on system security	Ind.1: Data is encrypted			3
	2.2. Third-party libraries are carefully checked based on system security	Ind.1 Third-Party Libraries are integrated and used			3
		Ind.2 Third-Party Libraries are updated			3
	2.3 User Authentication, Authorization and Accountability (AAA) are carefully applied based on NPM Universal Access Control (UAC)	Ind.1 Authentication is implemented			3
		Ind.2 Authorization is implemented			2
		Ind.3 Accountability is implemented			2
		Ind. 4: Environment Variables are Secured			2
		Ind. 5: Environment variables are used			2
Learning outcome 3:	3.1. Unit tests are appropriately conducted	Ind.1 Unit Testing			5

Test Backend Application **(21%)**	based on software testing techniques	is implemented			
	3.2. Usability is correctly tested according to expected results	Ind.1 Usability Testing is implemented			5
	3.3 Security is correctly tested according to expected results	Ind.1 Security Testing is implemented			5
Learning Outcome 4. Manage Backend Application **(21%)**	4.1 Application is appropriately deployed based on FURPS requirements	Ind.1: Deployment environment is prepared			3
		Ind.2: Application is Deployed			5
	4.2 Backend is effectively maintained according to the system Functionalities	Ind.1: Web Application is maintained			4
	4.3 Application documentation is properly generated according to the system backend	Ind.1: Documentation is performed using Postman			3
Total marks		**70**			
Percentage Weightage		**100%**			
Minimum Passing line % (Aggregate):		**70%**			

References

References

1. (2023, May 02). Express - Node.js web application framework (expressjs.com)
2. (2023, May 02). Documentation | Node.js (nodejs.org)
3. (2023, May 02). https://www.w3schools.com/nodejs/nodejs_get_started.asp
4. (2023, May 04). https://morioh.com/p/59e87ce2399f
5. Secure Backend Application
6. (2023, May 03). https://developer.okta.com/blog/2019/03/11/node-sql-server
7. (2023, May 04). https://jasonwatmore.com/post/2022/07/01/nodejs-ms-sql-server-simple-api-for-authentication-registration-and-user-management
8. (2023, May 04). https://www.devteam.space/blog/how-to-build-a-secure-web-application-with-nodejs/
9. Test Backend Application
10. (2023,May 04). https://brightsec.com/blog/unit-testing-in-nodejs/
11. (2023, May 04). https://www.youtube.com/watch?v=zLbsM_n1H9w
12. Manage a Backend Application
13. (2023, May 04). https://www.freecodecamp.org/news/how-to-build-a-backend-application/
14. (2023, May 04). https://adaptable.io/docs/app-guides/deploy-nodejs-app
15. (2023, May 04). https://www.youtube.com/watch?v=VStXlFxQgZg
16. Hughes, C., & Wilson, I. (2018). Node.js 8 the right way: Practical, server-side JavaScript that scales. Pragmatic Bookshelf.
17. Millspaugh, A. (2019). Beginning Node.js: Developing Web Applications and Beyond. Apress.
18. Osmani, A. (2018). Node.js design patterns: Build better software with reusable code. O'Reilly Media.
19. Pitt, A. (2019). Hands-On RESTful Web Services with TypeScript 3: Design and develop scalable RESTful APIs for your applications with TypeScript 3 and Node.js 12. Packt Publishing.
20. Rauch, G. (2018). Node.js at scale: Building distributed applications with DNode, ZeroMQ, and Node. O'Reilly Media.

21. Shaver, B. (2017). Professional Node.js: Building Javascript Based Scalable Software. Wiley.
22. Cantelon, M., Harter, T., & Holowaychuk, T. (2013). Node.js in Action. Manning Publications.
23. Ruben, T., & Teixeira, D. (2015). Beginning Node.js. Apress.
24. Hughes-Croucher, T., & Wilson, M. (2012). Node: Up and Running: Scalable Server-Side Code with JavaScript. O'Reilly Media.
25. Kärrberg, P. (2019). Node.js Design Patterns - Second Edition. Packt Publishing.
26. Mehta, S. (2015). Node.js Blueprints. Packt Publishing.
27. Hallett, G., & Winchester, S. (2016). Professional Node.js: Building Javascript-Based Scalable Software. John Wiley & Sons.
28. Bretz, M. (2014). Building Node Applications with MongoDB and Backbone. O'Reilly Media.