RWANDA POLYTECHNIC

HUYE COLLEGE

Date: May 9, 2025

Supervisor: Assistant Lecturer. Etienne NTAMBARA

Deadline: June 6, 2025

Presentation Day: June 6, 2025 in LAB1 From 8:30 am to 12:00pm

N.B: This is an individual work

Business Information Technology Internship (UR-CBE)

Project: E-LEARNING PLATFORM FOR RWANDA MODERN AI COMPANY - E-LEARNING SYSTEM

Summary

This project encompasses a comprehensive range of tasks, including:

1. Data Collection and Management
2. Database Maintenance
3. Database Backup
4. Additional tasks beyond these core areas

System Overview

This document outlines the complete scenario for building an e-learning platform for Rwanda Modern AI Company using React.js (frontend), Node.js (backend), and MySQL (database). The system will have three main user roles: Admin, Instructor, and Learner.

Technical Stack

- **Frontend**: React.js with React Router, Material-UI/Ant Design
- **Backend**: Node.js with Express.js
- **Database**: MySQL with Sequelize ORM

- **Authentication**: JWT (JSON Web Tokens)
- **Certificate Generation**: PDF generation library (PDFKit or similar)

Database Schema

Tables

1. **Roles**
2. **Users**
3. **Modules**
4. **ModuleContent**
5. **Enrollments**
6. **Assessments**
7. **Questions**
8. **Answers**
9. **UserAttempts**
10. **Certificates**
11. **Payments (Optional)**

Development Scenarios

1. Authentication Flow

Registration:

- All users register through a common form
- Role selection (defaults to learner)
- Email verification required

Login:

- JWT token returned upon successful login

- Token stored in secure HTTP-only cookie
- Different dashboards based on role

2. Admin Dashboard

Module Management:

- Create/edit/delete modules
- Assign instructors to modules
- Publish/unpublish modules
- View all modules with statistics (enrollments, completions)

User Management:

- View all users
- Promote users to instructor role
- Disable/delete users

System Reports:

- Enrollment statistics
- Revenue reports
- Module completion rates

3. Instructor Dashboard

Module Preparation:

- View assigned modules
- Upload content (videos, PDFs, presentations)
- Organize content in sequence
- Create assessments/quizzes
- Set passing criteria

Learner Monitoring:

- View enrolled learners
- Track learner progress
- View assessment results
- Provide feedback to learners

4. Learner Dashboard

Module Browsing:

- Browse available modules
- Filter by category, difficulty, duration
- View module details and syllabus

Enrollment:

- Subscribe to modules (free/paid)
- Payment integration (if paid module)
- Access to enrolled modules

Learning Experience:

- Progress through module content
- Mark content as completed
- Take notes/highlight materials
- Ask questions (forum or direct to instructor)

Assessments:

- Take final assessment when ready
- View results immediately
- Retake if not passed (up to max attempts)

- Download certificate upon passing

API Endpoints (Node.js)

Auth Routes

- POST /api/auth/register
- POST /api/auth/login
- POST /api/auth/logout
- GET /api/auth/me

Admin Routes

- GET /api/admin/users
- PUT /api/admin/users/:id/role
- GET /api/admin/stats
- GET /api/admin/reports

Module Routes

- GET /api/modules (public)
- GET /api/modules/:id (public)
- POST /api/modules (admin)
- PUT /api/modules/:id (admin/instructor)
- DELETE /api/modules/:id (admin)

Content Routes

- POST /api/modules/:id/content (instructor)
- PUT /api/content/:id (instructor)
- DELETE /api/content/:id (instructor)

Enrollment Routes

- POST /api/enroll (learner)
- GET /api/enrollments (learner)
- GET /api/enrollments/:moduleId (instructor)

Assessment Routes

- POST /api/assessments (instructor)
- GET /api/assessments/:moduleId
- POST /api/assessments/:id/attempt (learner)
- GET /api/attempts/:id (learner/instructor)

Certificate Routes

- GET /api/certificates/:moduleId (learner)
- GET /api/certificates/download/:id (learner)

React Component Structure

Main Layout

- App
  - Navbar (role-based navigation)
  - Sidebar (role-based)
  - MainContent

Auth Components

- LoginForm
- RegisterForm
- ForgotPassword

Admin Components

- UserManagement

- ModuleManagement
- ReportsDashboard

Instructor Components

- MyModules
- ModuleEditor
- ContentUploader
- AssessmentBuilder
- LearnerProgress

Learner Components

- ModuleCatalog
- ModuleDetails
- LearningPlayer
- NotesComponent
- AssessmentComponent
- CertificateViewer

Implementation Steps

Phase 1: Setup and Authentication

1. Set up React project with create-react-app
2. Create Node.js/Express backend structure
3. Design MySQL database schema and implement with Sequelize
4. Implement JWT authentication
5. Create basic role-based routing

Phase 2: Core Functionality

1. Implement module management (CRUD)
2. Build content upload and organization
3. Create enrollment system
4. Develop learning interface with progress tracking

Phase 3: Assessments and Certification

1. Design assessment system
2. Implement quiz/question components
3. Create certificate generation
4. Build retake logic

Phase 4: Polish and Deployment

1. Add payment integration (if needed)
2. Implement responsive design
3. Add analytics and reporting
4. Deploy to production environment

Security Considerations

1. Input validation on all forms
2. Role-based access control for all endpoints
3. Secure file uploads with virus scanning
4. Password hashing with bcrypt
5. HTTPS for all communications
6. Rate limiting on authentication endpoints

Testing Strategy

1. Unit tests for all utility functions

2. Integration tests for API endpoints
3. End-to-end tests for critical user flows
4. Load testing for assessment submission
5. Security testing for vulnerabilities

Database Backup

After the implementation of your project. Please make a backup copy of your final database to the free or paid cloud systems like: https://www.hostinger.com , DigitalOcean | Cloud Infrastructure for Developers, https://www.heroku.com/ ETC