SE ZG503 FULL STACK APPLICATION DEVELOPMENT
ASSIGNMENT- WEB APPLICATION DEVELOPMENT

Submission Date: 11th November 2024

Assignment Submission Mode: ELearn LMS- File Upload

Weightage: 20 % As per course handout

OBJECTIVE:

Develop a full-stack web application for one of the given problem statements that follows modern application development principles, allowing you to design and implement a robust system.

The assignment emphasizes both design and implementation: you are expected to design a fully-featured application but develop only a working prototype focusing on core functionalities.

Two problem statements are given below. Each Learner has to choose one of the problem statements.

This is an individual take-home assignment to be carried out by each learner independently.

PROBLEM STATEMENTS

TITLE: BOOK EXCHANGE PLATFORM

Problem Statement:

Book lovers frequently accumulate a collection of books they have read and look for other recommendations. They are always eager to explore new reading material. Traditional methods of exchanging books, such as local book swaps or lending among friends, are limited in scope and accessibility. Therefore, it is imperative to have a digital platform that can facilitate book exchanges on a larger scale. This platform should connect users with similar reading interests, enabling them to trade books easily and efficiently. This project aims to develop a full-stack web application that serves as a centralized platform for users to exchange, lend, and borrow books with other users. The platform should provide a user-friendly interface, robust search and recommendation features, and secure transaction capabilities.

Features:

User Authentication: Implement a secure user authentication system to allow users to register, log in, and manage their accounts.

Book Listing: Enable users to list books they want to exchange or lend, including details such as title, author, genre, condition, and availability status.

Book Search: Provide users with advanced search and filtering options to discover books based on criteria like genre, author, title, location, and availability.

Exchange Requests: Allow users to send and receive exchange requests for specific books, including negotiation options for terms such as delivery method and duration.

Messaging System: Implement a messaging system to facilitate user communication regarding book exchanges, including negotiation details, logistics, and scheduling. (Mock API s can be used)

User Profiles: Enable users to create profiles with information about their reading preferences, favorite genres, and books they currently own or wish to acquire.

Transaction Management: Provide tools for users to track the status of their exchange transactions, including pending requests, accepted exchanges, and completed transactions.

Expected outcome

The book exchange platform should provide users with a convenient and efficient way to discover new reading material, share their favorite books with others, and connect with fellow book enthusiasts in their community. The platform aims to promote a reading culture and foster a sense of community among users by facilitating book exchanges.

User Stories and Acceptance Criteria for the Book Exchange Platform

User Story 1: User Authentication

As a user,I want to securely register, log in, and manage my account, So that I can access and use the book exchange platform.

Acceptance Criteria:

The platform must allow users to register with a valid email and password.
Passwords must be stored securely using encryption.
Users should be able to reset their password via a password recovery system.
Users should be able to log out from their account.

User Story 2: Book Listing

As a user, I want to list books that I want to exchange or lend, So that others can browse and request the books I offer.

Acceptance Criteria:

Users should be able to add a book to their list by providing details such as title, author, genre, condition, and availability status.
Each book listing must have a unique ID associated with a user’s profile.
Users should be able to edit or delete book listings at any time.
The book listing must be displayed in the user's profile and searchable by others.

User Story 3: Book Search

As a user, I want to search for books based on criteria such as title, author, genre, and location,
So that I can easily find books that interest me.

Acceptance Criteria:
The platform must provide a search bar where users can enter keywords like title, author, or genre.
The platform should allow users to filter search results by availability status, genre, and location.
Users must be able to view detailed information about a book (title, author, condition, etc.) when clicking on a search result.
The search results should be paginated or load incrementally to handle large datasets.

User Story 4: Exchange Requests
As a user,I want to send and receive book exchange requests,So that I can initiate a transaction to exchange books with others.
Acceptance Criteria:
Users must be able to send an exchange request to another user for a specific book.
The request must include the option to negotiate terms, such as delivery method and exchange duration.
The recipient of the request should be able to accept, reject, or modify the request.
Both parties should receive notifications about the status of the exchange request (pending, accepted, rejected, modified).
The platform should track ongoing exchanges in the user's transaction history.

User Story 7: Transaction Management
As a user, I want to manage my book exchanges, So that I can track the status of all my exchange transactions.
Acceptance Criteria:
Users must be able to view a history of their exchange requests, including pending, accepted, and completed exchanges.
The transaction management interface should allow users to cancel pending exchanges.
Users should receive notifications when a transaction status changes (e.g., request accepted, book delivered).
Transaction history should be available to users on their profile page.

TITLE: LANGUAGE LEARNING PLATFORM

Problem Statement:

Learning a new language can be challenging, but with our digital platform, we want to make it easy and accessible for everyone. Our all-in-one language learning platform offers comprehensive learning resources, interactive exercises, and a supportive community for learners of all levels. Traditional language learning methods, such as textbooks and language classes, may not always cater to individual learning styles, but our platform offers a personalized and engaging learning experience. The objective is to develop a full-stack web application that caters to learners of various languages and proficiency levels. This platform will offer various learning materials, including language proficiency assessments, interactive activities, and community features. We are confident that our platform will revolutionize the language-learning industry and provide learners with an easy and enjoyable way to learn new languages.

Features:

User Authentication: Implement a secure user authentication system to allow users to register, log in, and manage their accounts.

Language Selection: Enable users to choose the language(s) they want to learn from a list of supported languages.

Learning Materials: Provide various learning materials, including vocabulary lists, grammar explanations, audio recordings, videos, and interactive lessons.

Language Proficiency Assessments: Provide language proficiency tests and assessments to help users evaluate their current language level and track their progress over time.

Progress Tracking: Display visual representations of users' language learning progress, including charts, graphs, and statistics.

Language Challenges and Games: Offer language learning challenges, games, and competitions to make learning fun and engaging.

Expected Outcome:

The language-learning platform aims to provide users with a comprehensive and interactive learning experience that helps them achieve their language-learning goals effectively. By offering a wide range of learning materials, interactive exercises, and community features, the platform seeks to foster language proficiency and cultural understanding among its users.

User Stories and Acceptance Criteria

1. User Registration and Authentication
User Story: As a new user, I want to create an account so that I can access the platform's learning features.
Acceptance Criteria:
Users must be able to sign up using email.
Upon successful registration, the user is directed to the dashboard.
Password must meet security criteria (min. length, special characters).
Users can reset passwords through an email link if forgotten.
Validation errors are shown clearly if any fields are incorrect or missing.

2. Language Selection
User Story: As a user, I want to select a language to learn so that I can access materials specific to that language.
Acceptance Criteria:
Users can select a language from a dropdown list on the dashboard.
Users can change languages at any time via their profile or settings.
Upon language selection, appropriate learning materials for that language are loaded.

3. Access Learning Materials

User Story: As a user, I want to access various learning materials like grammar lessons, vocabulary lists, and audio exercises so that I can study a new language effectively.

Acceptance Criteria:

Users can browse learning materials by category (vocabulary, grammar, etc.).

Each lesson or resource opens on a dedicated page with interactive components (e.g., audio player, text input fields).

Resources are categorized by language and proficiency level (beginner, intermediate, advanced).

4. Language Proficiency Assessments

User Story: As a user, I want to take language tests to evaluate my proficiency and track my learning progress.

Acceptance Criteria:

Users can take assessments for different skills (e.g., reading, listening, grammar).

Test results are displayed immediately with feedback on mistakes.

The system updates users' language level based on assessment results.

Users can view past assessment scores on their profile.

5. Progress Tracking

User Story: As a user, I want to see my learning progress so that I can measure how much I have improved over time.

Acceptance Criteria:

Users' progress is displayed visually on their dashboard (e.g., graphs, percentage completed).

Progress is tracked based on completed lessons, test results, and time spent on the platform.

Users can filter progress by time (daily, weekly, monthly).

ASSIGNMENT EXPECTATIONS

PART 1: DESIGN

Design a full-stack architecture for the given Problem statement. For the chosen application, define the application's purpose, user roles, and core functionalities. Identify the primary features of your application and design the architecture of the application.

The design should consider scalability by using a modular architecture. The frontend should be component-based (e.g., using React components) with a structured hierarchy, while the backend should follow proper separation of concerns (e.g., routing, controllers, services) and a suitable architecture. Identify the services and the operations to be implemented

Plan your application design in a way that would make it easy to add new features or extend the architecture in the future. For example, ensure that your backend logic is decoupled from the frontend so that other applications, like a mobile app can reuse it.

PART 2: BACKEND

You are expected to implement **three important user stories** for the problem statement. Your implementation should follow modern web application development practices.

Use Node.js (or another backend framework of your choice) to handle server-side logic. Develop REST APIs or GraphQL APIs to handle CRUD operations with a database. Ensure proper routing, URL mapping, and response handling. Use any database (SQL/NoSQL) for data storage and retrieval.

PART 3. FRONTEND

Design and implement the frontend UI for the same **three important user stories**. Choose a suitable frontend framework and implement it. Use a client-side JavaScript framework (e.g., React, Angular, Vue.js) to build the user interface. Design an interactive, user-friendly UI with responsiveness and state management. Implement the key components that reflect the selected functionalities

PART 4: INTEGRATION

A complete end-to-end demonstration of the selected user stories to be implemented.

Integrate the frontend and backend components and demonstrate the capabilities. Demonstrate communication between the front end and back end using APIs (e.g., AJAX, Fetch API). Use authentication for secure access to API endpoints.

DELIVERABLES:

1. **Design Documentation**: Full application architecture (frontend-backend interaction, API endpoints, database schema). UI/UX wireframes for the platform's core features. A detailed explanation of your approach to implementing scalability and modularity.
2. **Demonstration Video of the Working Prototype**: Video recording showing the demonstrations of the user stories implemented both frontend and backend.
   Note: if you are exceeding size limits when uploading videos to the elearn portal, add it to a Google Drive and submit the link)
3. **GitHub Repository:** A well-organized repository with a clear README file that includes instructions for setting up the application.
4. **Short Presentation:** A presentation explaining your technology choices, the user stories implemented, snapshots of UI, snapshots of API response in tools like Postman

SUBMISSION INSTRUCTIONS

1. Upload the complete code to the GitHub repositories. GitHub repository access is to be kept public and open for all evaluators.

2. Create a ZIP file including all the **four deliverables (Design document, Demonstration Video, Github Link(mentioned in a notepad), Short Presentation)** and upload a single ZIP file to the LMS.
3. Academic Honesty: You can discuss with peers and refer to the internet to understand the concept better. However, you may not share code with other groups or do not do a verbatim copy from the Internet/Generative AI responses/GitHub repositories. The code will be tested for plagiarism. If found guilty, no marks will be awarded.

WEIGHTAGE AND EVALUATION:

Total: 20 Marks

Activity	**Marks**
Design:	5%
Backend:	6%
Frontend:	6%
Integration:	3%

The relevant documents to be submitted in elearn portal. After the completion of deadline, Assignment will be evaluated by a group of instructors on the submissions made.

Evaluation Criteria:

- Design and Architecture: Completeness, scalability, and modularity of the design.
- Implementation of User Stories: Functionality, accuracy, and user experience of the implemented stories.
- Code Quality: Readability, adherence to best practices, and documentation.
- Presentation: Clarity in explaining design and implementation.

NOTE:

- This is a take-home assignment to be carried out by each learner independently
- In case of any further queries, use discussion forums, or reach out to me at akshaya.ganesan@pilani.bits-pilani.ac.in.