Ruby on Rails - Technical Interview Exercise

Project Overview

Read carefully all the requirements and create a product that fits the expectations. Please create a library management system with the requirements described in the next section.

Your development should be driven by an informal user story that you will create, and which should be included in your presentation.

Requirements

Backend

- **Authentication** and **Authorization**:
    - Users should be able to register, log in, and log out.
    - Two types of users: **Librarian** and **Member**.
    - Only Librarian users should be able to add, edit, or delete books.
- **Book Management**:
    - Ability to add a new book with details like title, author, genre, ISBN, and total copies.
    - Ability to edit and delete book details.
    - **Search functionality**:
        - Users should be able to search for a book by title, author, or genre.
- **Borrowing and Returning**:
    - Member users should be able to borrow a book if it's available. They can't borrow the same book multiple times.
    - The **system should track** when a book was borrowed and when it's due (2 weeks from the borrowing date).
    - Librarian users can mark a book as returned.
- **Dashboard**:
    - Librarian:
        - A dashboard showing total books, total borrowed books, books due today, and a list of members with overdue books.
    - Member:
        - A dashboard showing books they've borrowed, their due dates, and any overdue books.
- **API Endpoints**:
    - Develop a **RESTful API** that allows **CRUD** operations for books and borrowings.
    - Ensure proper status codes and responses for each endpoint.
    - Testing should be done with **RSPEC**.
    - Spec files should be included for all the requirements above.
- **Submission Guidelines:**

  - Include a **README** with setup instructions and any other documentation you deem necessary.
  - The application should have seeded data / credentials for demo purposes.

Frontend

- While the main focus is on Ruby on Rails, we need a decent level of **full stack development** skills, so:
  - Please integrate the backend you just built with a frontend framework of your choice (React, Vue, etc.).
  - Key criteria:
    - The frontend should be responsive and user-friendly.
    - Implement CRUD operations associated with the implemented use case.
    - Structured code: Organize your components and state cleanly.
- **Submission Guidelines:**
  - Include a **README** with setup instructions and any other documentation you deem necessary.
  - The application should have seeded data / credentials for demo purposes.

Generative AI tools

Imagine you're tasked with generating a **RESTful API** for a simple **task management system** using your preferred language. The system should support the following functionality:

- Create, read, update, and delete tasks (CRUD)
- Each task has a `title`, `description`, `status`, and `due_date`
- Tasks are associated with a user (assume basic User model exists)

Instructions:

- Using your preferred **GenAI coding tool** (e.g., Cursor, Claude Code, Windsurf, GitHub Copilot, etc.), write the **prompt you would use** to generate the API scaffold or full implementation.
- Show the **output code** (or a representative sample of it).
- Describe how you:
  - Validated the AI's suggestions
  - Corrected or improved the output, if necessary
  - Handled edge cases, authentication, or validations
  - Assessed the performance and idiomatic quality of the code

Presentation and Code Review

Ruby on Rails - Technical Interview Exercise

You will be required to present your project to the technical interview panel. During the presentation, you should explain your user story, design choices, the technical architecture, and demonstrate the functionality of the application. This will be done over Google Meets or Zoom and you will screen share either your GitHub repository or IDE.

After the presentation, the interview panel will conduct a code review of your project. You will be asked to explain your coding decisions and answer any questions related to the code. The interview panel will evaluate your project based on the following criteria:

- **Clean Architecture**: Your architecture should adhere to Clean Architecture principles, including separation of concerns and independence of components.
- **Application testing:** Your project should have sufficient test coverage. Use of TDD is preferable.
- **Code quality**: Your code should be well-organized, readable, and adhere to best practices.
- **Functionality**: Your application should perform as expected in the requirements without errors or bugs. Optional but desired: no warnings in the browser console.
- **Presentation**: Your presentation should be clear, concise, and demonstrate a good understanding of the project and of the main ***backend*** and ***frontend*** best practices.
- **GenAI tools**: Your answers and presentation must show fluency with GenAI tools and prompt engineering, and critical thinking when evaluating AI-generated code.