# Problem Statement:

Develop a minimal yet functional book review platform (**UI and Backend**) with basic user authentication, CRUD operations for book reviews, rating aggregation, and a basic recommendation feature.

## Core Features and Requirements:

### 1. User Authentication

* **Signup/Login/Logout** (email + password)
  + Use token-based authentication (e.g., JWT)
  + Data model: User ID, email, hashed password, name

### 2. Books Listing & Search

* Users should be able to:
  + View a paginated list of all books
  + Search books by title or author
  + Data model:Book ID, title, author, description, cover image (URL), genres, published year

### 3. Reviews & Ratings (CRUD)

* Users can:
  + Create, read, update, delete their own reviews
  + Rate books on a 1–5 scale
  + Data model: Review ID, book ID, user ID, text, rating (1–5), timestamp

### 4. Average Rating Calculation:

* Each book should display:
  + Average rating (rounded to 1 decimal)
  + Total number of reviews
* Automatically updated when a review is added/edited/deleted

### 5. User Profile:

* Shows:
  + List of reviews written
  + Favourite books (user can mark/unmark)

### 6. Recommendations

* Return a list of books the user may like based on:
  + Top-rated books (default for MVP)
  + Books similar to user's favourites or genres
  + Use LLM based service APIs like OpenAI to provide recommendations.

### 7. Spec driven development:

* As Ritesh showed in the session ([Recording](https://teams.microsoft.com/l/meetingrecap?driveId=b%21BzO5zs43rEmWpG25eKQD7bqKlh66NjBDlMLdXux4uCQwU4mQm6FZQp0p3vk3rdB5&driveItemId=01HNHWGYRFXJPZWOK7DNAKN6B6UICZTCF4&sitePath=https%3A%2F%2Ftalenticaall-my.sharepoint.com%2F%3Av%3A%2Fg%2Fpersonal%2Fdipen_mistry_talentica_com%2FESW6X5s5XxtApvg-ogWZiLwB3z72FdugBC54AYb93KUPKQ&fileUrl=https%3A%2F%2Ftalenticaall-my.sharepoint.com%2Fpersonal%2Fdipen_mistry_talentica_com%2FDocuments%2FRecordings%2FAI-Native%2520Engineering%2520Pre-assignment%2520session%2520on%2520leveraging%2520coding%2520assistants-20250710_150132-Meeting%2520Recording.mp4%3Fweb%3D1&iCalUid=040000008200E00074C5B7101A82E00800000000EEC7DA3707F0DB010000000000000000100000003946CF108BE7F14789355DBB6023D360&threadId=19%3Ameeting_YjJhNGI5NjMtZDY2MC00NjYzLTllZmYtOGQ1MGFlNzc5NDRj%40thread.v2&organizerId=fa28b5ab-1b8c-4bcf-adbc-3c29187d7261&tenantId=1f8e1b2e-5d9b-44a7-91fc-dbe7d9a51b15&callId=e603e1f9-c478-46fb-a080-6bf82f265ebc&threadType=Meeting&meetingType=Scheduled&subType=RecapSharingLink_RecapCore)), we need to do this development via spec driven.
* You should first generate PRD document md file, covering most of the PRD aspects like functional requirements, goals, target users, constraints, etc.
* Generate design document with a high-level component diagram in md file. Cover design document aspects like non-functional aspects, tech stack, etc.
* Based on PRD and design documents, create a task breakdown into a md file. Get the project development done via Copilot/Cursor based on this task breakdown file.

### 8. Unit Testing:

* Write unit test cases for the backend service and make sure overall code coverage is more than 80%.

### 8. Deployment:

* Write terraform script to create infra for both frontend and backend services.
* Create an infra pipeline to create application resources.
* Create a pipeline to deploy the code to frontend and backend services.

## **Note: You need to use AI assistant Copilot/Cursor/Windsurf/Kiro only to build this portal.**

# Acceptance Criteria:

* A public backend repo with code, PRD, design, task breakdown md files. terraform script and pipeline scripts for deployment, commit code coverage repo.
* A public frontend repo with code, PRD, design, task breakdown md files. terraform script and pipeline scripts for deployment.
* A working demo video covering all the must have features like auth, book listing and search, reviews, rating, user profile, Avg rating calculations and recommendations.
* A text file containing all your prompts that you used in Copilot/Cursor to develop the app. You can export the chat conversation from both Copilot and Cursor.
* Email the assignment output detail to me to submit your assignment.
  + Public repo links
  + Prompts text file
  + Recording video link

# Points to consider:

* You can use your client provided Copilot/Cursor if there are no concerns on the usage, use default model like GPT 4o,4.1.
* In case you don’t have client provided Copilot/Cursor, you can purchase a $10 plan for Copilot or $20 plan for Cursor. Prefer $10 Copilot plan it comes with unlimited GPT 4.1, even offer unlimited GPT-5-mini. Cursors provide limited agent requests with $20 plan, find cursor details [here](https://docs.cursor.com/en/account/pricing)
* There shouldn’t be any commit in the repo after the submission date till it is evaluated.
* Video must cover the required features demo; add you voice to explain the features.

# Reading Material:

## Copilot:

Building a REST API with GitHub Copilot: A beginner's guide

<https://www.youtube.com/watch?v=CJUbQ1QiBUY>

Building a React app with GitHub Copilot: A beginner's frontend guide

<https://www.youtube.com/watch?v=Nw4y5XQyugc>

GitHub Copilot 101 - Essential features | Tutorial

<https://www.youtube.com/watch?v=b5xcWdzAB5c&t=10s>

Getting Started with Visual Studio Co-Pilot :-

<https://code.visualstudio.com/docs/copilot/chat/getting-started-chat>

<https://code.visualstudio.com/docs/copilot/getting-started>

Ask Mode:-

<https://code.visualstudio.com/docs/copilot/chat/chat-ask-mode>

Edits Mode:-

<https://code.visualstudio.com/docs/copilot/chat/copilot-edits>

Agent Mode :-

<https://code.visualstudio.com/docs/copilot/chat/chat-agent-mode>

How to Craft Prompts:-

<https://code.visualstudio.com/docs/copilot/chat/prompt-crafting>

Copilot Context :-

<https://code.visualstudio.com/docs/copilot/chat/copilot-chat-context>

Copilot smart actions in Visual Studio Code:

<https://code.visualstudio.com/docs/copilot/copilot-smart-actions>

Customize AI responses in VS Code:

<https://code.visualstudio.com/docs/copilot/copilot-customization>

## Cursor:

<https://docs.cursor.com/get-started/quickstart>

<https://docs.cursor.com/get-started/concepts>

More docs in the same menu to dig deep.

**Ritesh Session Recording in AI assistant:** [Link](https://teams.microsoft.com/l/meetingrecap?driveId=b%21BzO5zs43rEmWpG25eKQD7bqKlh66NjBDlMLdXux4uCQwU4mQm6FZQp0p3vk3rdB5&driveItemId=01HNHWGYRFXJPZWOK7DNAKN6B6UICZTCF4&sitePath=https%3A%2F%2Ftalenticaall-my.sharepoint.com%2F%3Av%3A%2Fg%2Fpersonal%2Fdipen_mistry_talentica_com%2FESW6X5s5XxtApvg-ogWZiLwB3z72FdugBC54AYb93KUPKQ&fileUrl=https%3A%2F%2Ftalenticaall-my.sharepoint.com%2Fpersonal%2Fdipen_mistry_talentica_com%2FDocuments%2FRecordings%2FAI-Native%2520Engineering%2520Pre-assignment%2520session%2520on%2520leveraging%2520coding%2520assistants-20250710_150132-Meeting%2520Recording.mp4%3Fweb%3D1&iCalUid=040000008200E00074C5B7101A82E00800000000EEC7DA3707F0DB010000000000000000100000003946CF108BE7F14789355DBB6023D360&threadId=19%3Ameeting_YjJhNGI5NjMtZDY2MC00NjYzLTllZmYtOGQ1MGFlNzc5NDRj%40thread.v2&organizerId=fa28b5ab-1b8c-4bcf-adbc-3c29187d7261&tenantId=1f8e1b2e-5d9b-44a7-91fc-dbe7d9a51b15&callId=e603e1f9-c478-46fb-a080-6bf82f265ebc&threadType=Meeting&meetingType=Scheduled&subType=RecapSharingLink_RecapCore)