**1. Project Title**

DevShelf – Personalized Technical Bookshelf and Learning Journal

**2. Concept Description**

DevShelf is a web application platform designed for college students, developers, and self-students to systematically manage their technical reading process. Users can search for all kinds of technical books through the integrated Google Books API and add them to their personal bookshelf.

Functions include: setting the reading status of a book (scheduled, in progress, completed), adding custom tags (e.g. ‘front-end’, ‘algorithms’), rating each book, and organising study notes.

In addition, DevShelf provides a recommendation engine based on users' tagging preferences, automatically suggesting new books that may be of interest to users to help them learn continuously and efficiently. The platform's core user groups are mainly college students, beginner developers, and self-learners who are interested in technology.

**3. API Integration Plan**

**API Used:**

Google Books API

**Purpose:**

Get meta-information on technical books such as:

* Book title
* Author(s)
* Thumbnail image
* Description
* Published date
* Preview link

**Integration Method:**

* The user searches for books using keywords.
* The system calls the Google Books API and displays results as book cards.
* When a user saves a book to their shelf, the book data is stored in the local database and linked to user-generated data like notes, tags, reading status, and rating.
* Duplicate entries are avoided using the unique book ID from the API.

1. **Custom Database Plan**

**Stores content fields:**

* User information (user\_id, email, password)
* Basic book information retrieved from the API
* User behaviour data: notes, tags, status, rating

Key Tables & Fields

Users:user\_id (PK), email, password, created\_at

Books:book\_id (API ID), title, author, thumbnail\_url, description

UserBooks:user\_id (FK), book\_id (FK), reading\_status, rating, tags

Notes:note\_id, user\_id (FK), book\_id (FK), note\_text, created\_at

**Data Relationships**

One user can save multiple books.

Each book can be saved by multiple users.

Users can add multiple notes per book.

Each saved book is customized with tags, status, and rating by each user.

User behaviour data is combined with book API data to build a personalised content system.

**5. User Service Description**

DevShelf is dedicated to transforming the user's reading process into a structured, personalised learning experience that delivers practical and ongoing value to the user.

**Core service features include:**

Personalised bookshelf management: Users can bookmark books in their personal bookshelf and set the reading status (scheduled, in progress, completed) to form a clear learning track.

Custom labelling system: each book can add custom labels (e.g. ‘Algorithm’, ‘Front-end’, ‘Interview’), which helps users to classify and manage books according to their personal learning goals. Categorise and manage books according to personal learning goals.

Advanced Filtering and Searching: Users can filter the books in the bookshelf according to the label, rating or reading status, etc., and quickly locate the resources they need to read most.

Study Notes: Each book supports users to write study notes, which can record reading thoughts, knowledge points or code samples to promote in-depth thinking and knowledge consolidation.

Book rating and review: Users can rate each book to review their reading experience and optimise future reading choices.

Intelligent Recommendation System: The platform recommends technical books that may be of interest to users based on their frequently used tags and history of favourites, expanding the boundaries of learning.

Learning Output Export: Users can export their bookshelves and learning notes to PDF documents for use as personal learning files, presentation materials, or project reports.

Through the above features, DevShelf not only helps users to organise their technical reading content, but also establishes a systematic knowledge system and enhances long-term learning through continuous recording and intelligent recommendations.

## **6. Team Collaboration Plan**

### **Roles & Responsibilities**

| Role | Team Member (Example) | Responsibilities |
| --- | --- | --- |
| Frontend Developer | Bowen Zhu | Build UI (search, shelf, notes) |
| Backend Developer | Jie Li | API integration, user auth, database logic |
| Recommender Dev | Xiyuan Zhang | Implement tag-based recommendation system |
| Project Manager | Xinchen Luo | Coordinate tasks, testing, documentation |

### **Communication Plan**

## ...

## **Development Timeline**

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Milestone | Tasks | Responsible Members |
| 3-6 | Project Planning & Setup | - Finalize project idea and scope  - Design database schema  - Set up GitHub repository and Trello board | All Members |
| 7 | API Integration & Search | - Integrate Google Books API  - Build search input & results page UI  - Display book cards dynamically | Jie Li |
| 8 | Personal Bookshelf | - Implement save-to-shelf logic  - Create “My Bookshelf” page  - Store book metadata & link to users | Bowen Zhu |
| 9 | Notes & Tagging System | - Add personal notes per book  - Allow tag input and filtering  - Display notes on book detail view | Xinchen Luo |
| 10 | Rating & Recommendation | - Enable book rating system  - Build tag-based recommendation logic  - Show recommended books section | Xiyuan Zhang |
| 11 | Testing & Bug Fixing | - Conduct unit tests and UI testing  - Fix bugs from earlier stages  - Polish user interactions | Xinchen Luo |
| 12 | Finalization & Submission | - Write documentation  - Export demo notes as PDF  - Record demo video  - Final submission | All Members |