Web Programming 1, Semester 2 - 2025

**Angular Project Proposal**

1. **Project description**

Book Finder is an Angular web application designed to help users easily search for books using the Google Books API. The application aims to provide a simple and efficient way for users to discover books based on various criteria, such as title, author, genre, publisher, and ISBN. It is useful for book students and people who need a quick and convenient way to find books online.

**Main Features**

* Book Search – Users can search for books by entering keywords related to the title, author, publisher, genre or ISBN.
* Advanced Filtering – Search results can be refined by language, availability, sorting options and book type.
* Book Details – Clicking on a book displays additional information: description, published date, category, ISBN, availability and ratings.
* Favourites – Users can save books to their favourites, which are stored in a MongoDB database for easy access later.
* User Friendly Interface – The application is designed with a clean layout, making it easy to navigate and use, and responsible on all devices.
* Pagination – Large sets of search results are managed through pagination for better usability.

**Why is it Useful or Interesting?**

Book Finder helps users quickly find books in one place, saving time and effort. It is useful for students and casual readers by allowing them to search, filter, and preview books easily. The app also helps users discover new books.

1. **Wireframes**

The application follows a mobile-first design approach, ensuring great experience on smaller screens before scaling up to larger devices. It consists of three main pages:

* Home Page – Users can search for books, apply filters, and add books to their favourites which are stored in MongoDB. Each result displays a thumbnail, title and author, with more details available upon selection. Pagination is implemented for better navigation.
* About Page is a simple static page that includes a brief application description. Due to its simplicity, wireframes for this page were not included in the proposal.
* Favourites Page – Displays saved books from MongoDB, allowing users to view or remove them.

The wireframes have been created to reflect the API’s data structure and ensure a user-friendly layout. Below are the wireframes for the Home Page and Favourites Page for both mobile and desktop versions.

**Mobile**

Home Page Favourite Page

**A paper with lines and text

AI-generated content may be incorrect. A graph paper with lines and text

AI-generated content may be incorrect.**

**Desktop**

Home Page

**A paper with writing on it

AI-generated content may be incorrect.**

Favourites Page

**A paper with lines and symbols

AI-generated content may be incorrect.**

1. **API**

**Endpoints Used**

* The application retrieves book data from the Google Books API using the following endpoint - <https://www.googleapis.com/books/v1/volumes>

**Testing in Postman**

* The API was tested in Postman by making GET requests with different query parameters (q=harry+potter). The responses were analysed to understand the structure of the data and determine which fields are relevant for display. The API response contains 5 layers of nested data, which were considered when designing the UI.

**Search & Filtering**

* The application supports searching by: Title, Author, Genre, Publisher and ISBN
* Filtering options include: Language, Availability, Sorting (Relevance/Newest) and Book Type (Books/Magazines)

**Data Structure & Display**

High-level data:

* Thumbnail image
* Title
* Author

Book Details:

* Description
* Published Date
* Categories
* ISBN, Availability and Ratings

**MongoDB Storage**

**The database will store:**

* Book ID, Title, Image, Author and Availability Status
* Additional user-generated content, such as notes or reviews