

Book Review Platform

# Ibtisam Al-hitteh

## Introduction:

The development of a Book Review Platform involves creating an online space where users can view, rate, review, and discuss books they've read. This platform will enable book enthusiasts to connect, exchange ideas, and discover new books. As with any software project, several assumptions and development decisions must be documented to ensure the project's success and to maintain transparency throughout the development process.

## Development Decisions:

Development decisions are key choices made during the platform's design and implementation. These decisions include selecting technologies, frameworks, and tools that best meet the needs of the platform while addressing project requirements and constraints.

### 1. Front-End Framework: Vue.js

- a. **Decision:** Vue.js was chosen as the front-end framework for building the user interface.

### 2. Back-End Framework: Laravel (PHP):

- a. **Decision:** PHP, using the Laravel framework, was selected for building the back-end API and Admin.
- b. **Reasoning:** Laravel express is a minimal, flexible framework that allows for easy handling of HTTP requests and integration with other services, making it a good fit for building RESTful APIs for the platform.
  - Repository Design Pattern used for build Model in laravel application
  - Passport as API authentication
  - z-song for build Admin back-end

### 3. Database: SQL, MySQL Tool

- a. **Decision:** MySQL was chosen as the relational database management system for the platform.
- b. **Reasoning:** SQL is highly reliable and supports advanced querying, transactions, and data integrity, which is essential for storing user-generated content like reviews, ratings, and user profiles. Its robust support for SQL queries also ensures scalability as the platform grows.

### 4. Google Books API

- a. **Decision:** External API was chosen as the trusted source to get collection of books, which is the main point of platform introduce useful books for user.
- b. Call google books api from end point <https://www.googleapis.com/books/v1/volumes>, and get collations of book with information all information, build api to get all needed data for book.

## Headlines:

### Repository Design Pattern: An Overview

The Repository Design Pattern is a structural pattern that provides a way to manage data access in an application, abstracting the underlying data source (e.g., database, file system, external API) from the rest of the application. It acts as an intermediary layer between the application's business logic and the data source, encapsulating the logic for retrieving, storing, and querying data. This helps maintain a clean separation of concerns, making the code easier to maintain and test.

### Key Concepts of the Repository Design Pattern:

- 1- Abstraction of Data Access:
- 2- Decoupling Business Logic from Data Layer
- 3- Collection-Like Interface

### Database Classes:

- 1- **User Class** (id , name , email , password)
- 2- **Book Class** (id , title,subtitle,authors,print\_type, page\_count, publisher, published\_date, average\_rating, thumbnail,language, categories)
- 3- **Review Class** (id , content , book\_id , user\_id)
- 4- **Rating Class** (id , rating\_value, book\_id , user\_id)
- 5- **User Books** is a relationship of type "Many To Many", which is display the list of history for user

### Repository Classes:

You can find all repo on "App\Repositories" folder with the "AbstractModel" file.

### Front-end Vue:

#### Github URL

[https://github.com/ibtisamalhitteh/books\\_preview\\_frontend\\_vuew](https://github.com/ibtisamalhitteh/books_preview_frontend_vuew)

#### Follow these steps:

# Step 1: Clone the repository using the project's Git URL.

```
git clone < git@github.com:ibtisamalhitteh/books_preview_frontend_vuew.git>
```

# Step 2: Navigate to the project directory.

```
cd < books_preview_frontend_vuew>
```

# Step 3: Install the necessary dependencies.

```
npm i
```

# Step 4: Start the development server with auto-reloading and an instant preview.

```
npm run dev
```

## Back-end Laravel Application:

### Github URL:

[https://github.com/ibtisamalhitteh/book\\_preview](https://github.com/ibtisamalhitteh/book_preview)

### Follow these steps:

# Step 1: Clone the repository using the project's Git URL.

```
git clone < git@github.com:ibtisamalhitteh/book_preview.git>
```

# Step 2: Navigate to the project directory.

```
cd < book_preview>
```

# Step 3: Install the necessary dependencies.

Composer install

# Step 4: Config .env file

Change app url and Database information

# Step 5: Run Migration to create all database table

```
php artisan migrate
```

# Step 6: Run Seed to create user and admin

```
php artisan db:seed
```

# Step 7: Admin

Open <http://localhost/admin/> in browser, use **username** admin and **password** admin to login.

# Step 8: Run application

```
"php -S localhost:8000 -t public" or "php artisan serve"
```

# Step 9: Import postman collection

You can find postman collection inside root "postman" folder  
"books\_review.postman\_collection.json"