

Mehran University of Engineering and Technology, Jamshoro

**MAD CEP**

**Book Blossoms E-Bookstore**

**Name:** Alishba Abdul Aleem

**Roll No:** 21SW001

**Section:** 01

**Submission Deadline:** 25-October-2024 **Submitted To:** Ms. Mariam Memon

1. **Real World Problem Identification**
   1. **Problem Statement:**

When it comes to book shopping, the traditional bookstore will usually not suffice for many consumers with needs that have become ever more diverse – craving convenience and personalization in their purchasing journey.

* 1. **Concerns**:

**Accessibility Issue:** Most of the bookstores are concentrated in certain places which is also a considerable challenge for those living at distant or less developed areas to avail wide array of books.

**Lack of Personalization:** Consumers often struggle to find books that align with their interests, leading to a generic shopping experience.

**Time Constraints**: Life gets busy—between work, family, and other commitments, finding time to browse through shelves can feel impossible. Many readers end up sticking to familiar titles instead of exploring new ones

**Poor inventory:** Bookstores may not know how to inventory properly so they get out-of-stock for popular books but still have a bunch of less well-known titles.

**Making Digital Transition:** Even as a growing number of consumers prefer digital solutions, most existing bookstores still do not effectively utilize technology to improve their shopping experience.

1. **Proposed Solution:**

**Easier Wishlist Management:** No need to write down the book titles — users can go ahead and create wish lists within the app. This allows users to keep things on their wish list and be alerted when they are available or listed at a discount.

**More Titles:** An e-bookstore, as opposed to a physical bookstore which has limited shelf space, contains more titles of eBooks. All of this combines to allow for a massive breadth and depth in the catalog, meaning readers can find works by different writers and within multiple genres available at just their fingertips.

**Community Reviews and Ratings:** Traditional bookstores typically offer staff recommendations as a form of community review. For example, in an E-Bookstore app the customers can read and write reviews that generate a community to discuss books.

**Smart Search & Filter:** It takes only a fraction of time to retrieve the exact book instead of perusing racks. The E-bookstore app makes this easy with powerful search and filter tools that allow users to find just what they want in seconds.

**Fast and Easy Checkout:**. The e-bookstore app is designed to help users finish on-line e book retail purchases which are initiated out of your webpage

1. **Data Storage**

In my Book Blossoms application, I chose Firebase as the primary storage solution for several key reasons:

**Firebase Realtime Database** — A real-time database to perfectly sync data across all clients. I took advantage of this to ensure we always present the current book availability, which is key to a good user experience; it’s just simple better if users have access to up-to-date information.

**Easy Authentication:** Firebase Authentication makes it easy to log your users in with an email/password-based credential. I then used this versatility to provide an easily approachable way for users to make their accounts mud, in turn boosting engagement and retention.

**Scalability**: Firebase grows with my application. When we have more and more users on board Book Blossoms, who continue to add books one after the other, Firebase will scale without me having to make major changes in infrastructure. This scalable nature helps the application to advance without challenges in efficiency as it gets a larger user base.

**Security Features**: Firebase provides robust security measures, including user authentication and data encryption, to protect sensitive user information and book data. I utilized these features to maintain user trust and ensure compliance with data protection regulations.

1. **APIs/Packages/Plug-ins**

**Carousel Slider (carousel\_slider: ^5.0.0)**

* **Justification**: I selected the Carousel Slider package to create a visual and interactive display of featured books, as well as promotions in the Book Blossoms app.

**FlutterRatingBar (flutter\_rating\_bar: ^4.0.1)**

* **Justification**: I use a Flutter Rating Bar package to show book ratings in star format.

**Firebase Core (firebase\_core: ^3.4.0)**

* **Justification**: I have added the Firebase package to initialize firebase in my app.

**Firebase Auth (firebase\_auth: ^5.2.0)**

* **Justification**: I used the Firebase Auth package for user authentication process via email/password.

**cloud\_firestore (cloud\_firestore: ^5.4.0)**

* **Justification**: For the real-time collection and document storage, as well as live synchronization cross all supported device types`. I chose Cloud Firestore. Using this package I can efficiently deal with book data & changes of schemas which immediately appear in all devices leading to a smooth user experience.

**Firebase Storage (firebase\_storage: ^12.2.0)**

* **Justification**: I utilized firebase storage to store the book documents in pdf format and later fetched them through download URL to be used in implementing (In App Reading) functionality

**PDFX package (pdfx: ^2.6.0)**

* **Justification**: I utilized pdfx package to display my book documents that were stored in pdf format at firebase storage in my flutter applicating using the package’s built in PdfViewPinch function

**Provider (provider: ^6.1.2)**

* **Justification**: I utilized provider package to create a provider class for the Cart Screen, so that every time the user clicks on (add to cart) button, the provider notifies and updates the BookCart screen

1. **Issues and Bugs Encountered and Resolved during Development**

* During the development of the Book Blossoms application, I encountered several issues that required resolution:

**SingleChildScrollView and Expanded Conflict**: I initially tried using SingleChildScrollView with Expanded, but I discovered that this combination was not compatible. To resolve this, I had to rework my layout to ensure that scrolling functionality worked effectively without causing errors.

**Firebase Auth MinSdk Issue**: In my app when I tried to run Firebase Auth on a mobile device, minSdk errors had come. I fixed this by tweaking the build. gradle file for the correct SDK version that Firebase needed to interact with authentication.

**Firestore Data Fetching Delay:** there was some delay while fetching the data from Firestore and storing it in cloud hence I temporary error. To combat this, I added a check that would create a loading indicator until the data was successfully fetched in unity, which helped with understandable user experience during load times.

**Difficulty in Navigating Between Screens**: I had difficulty navigating between different sections of the app, especially when returning to previous pages. I improved the navigation flow by implementing a bottom navigation bar, making it easier for users to switch between different sections of the app

**Debugging Firebase Connection Errors:** I faced challenges with connection errors when trying to interact with Firestore. After some troubleshooting, I discovered that it was due to incorrect configuration settings in my Firebase project. I corrected these settings, ensuring a stable connection.

**Bottom Overflow Error:** In Book Blossoms, I faced bottom overflow errors on the home screen when users scrolled through the list of featured books. The layout didn’t adjust properly when additional content was dynamically loaded, causing some items to become inaccessible, especially on smaller devices. To resolve this, I wrapped the layout in a `SingleChildScrollView` and enabled the `resizeToAvoidBottomInset` property in the `Scaffold`, ensuring all elements remained visible and accessible.

**Book Blossoms UI:**

**On Boarding Screen:**

Utilizes **PageViewBuilder** to build the onboarding screens, for the Book Blossoms Application

A box with white text and red letters

Description automatically generated with medium confidenceA person standing next to a small building

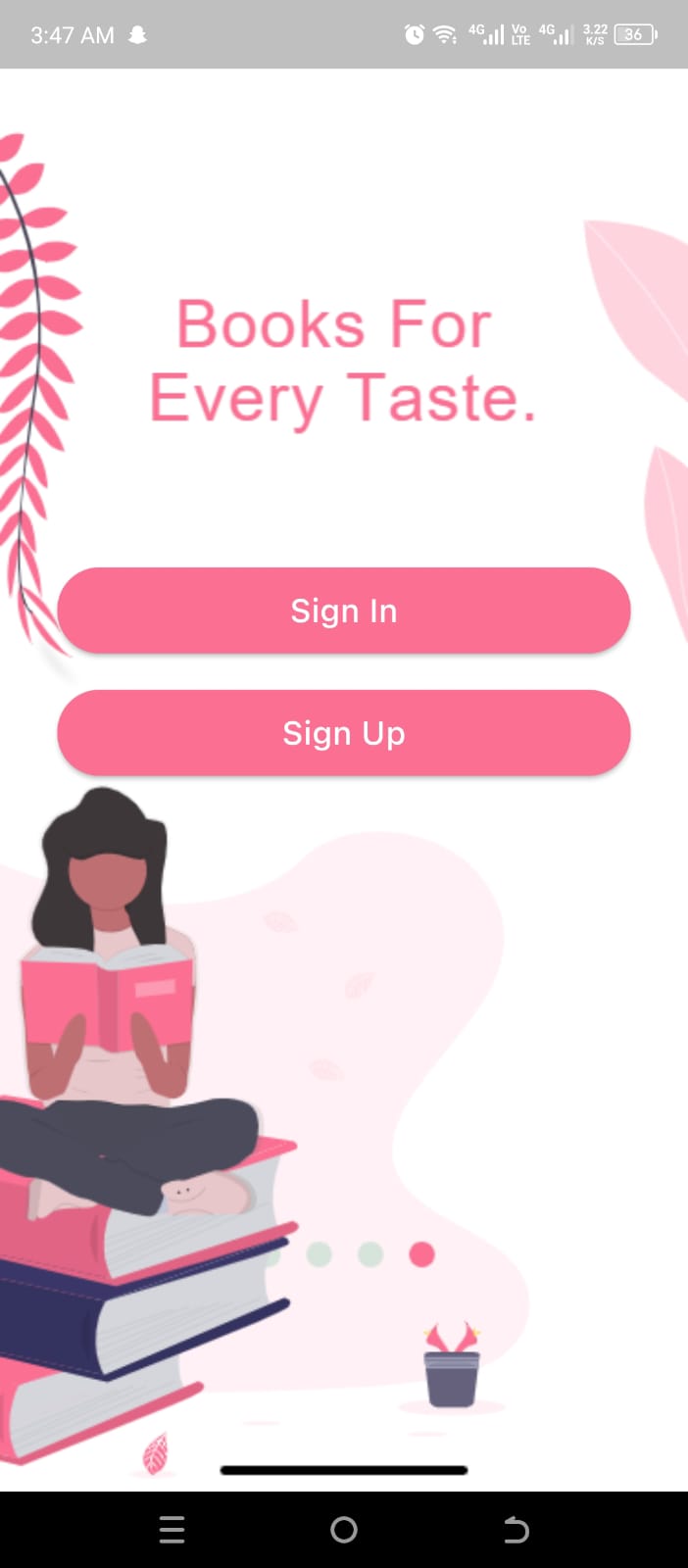
Description automatically generatedA person standing in front of a book

Description automatically generated

**Sign-up and Sign-in Screens:**

Here I used the **Form widget** for the Sign Up and Sign In screen to validate the user’s response.

Additionally, I have integrated **Firebase authentication**, for user authentication and personalized experience

****A screenshot of a login form

Description automatically generatedA screenshot of a login screen

Description automatically generated

**Home Screen, Book Detail screen and In App Reading Screen**

In **Home Screen,** I used **CarousalBuilder** for the top part of the screen, to display our Top Pick Books in a rotating sliding manner. For the rest of the Home Screen, **ListViewBuilder** has been used.

Additionally, every book has been enclosed in **GestureDetector, for a smooth navigation to Book Profile Screen**, where the book details are being displayed.

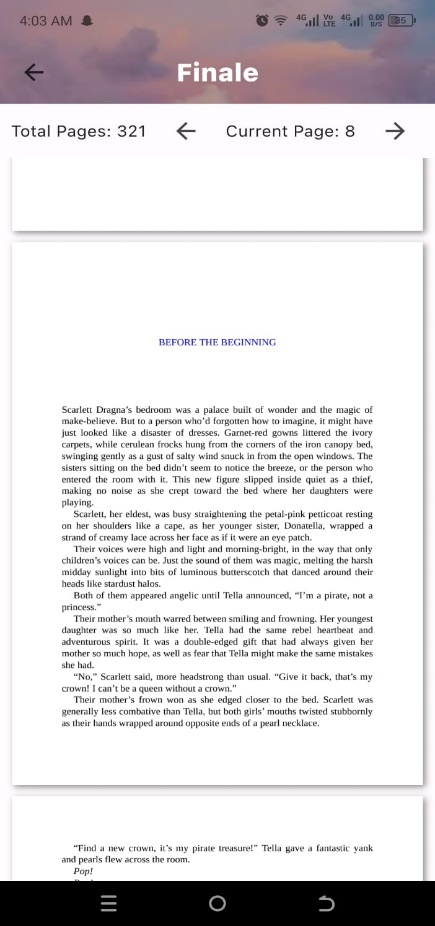
Also, the book information is not stored locally but rather on **Firebase Database,** from where it is being displayed using **firebase\_cloud** package

Furthermore, we have an **In App reading screen**, where we have used a **pdfx package**, mainly the **PdfViewPinch** function, that takes a pdf document as an argument, to display our pdf file.

The pdf documents are stored in **Firebase storage.** I utilized firebase\_storage package built in methods to fetch the DownloadUrl for the specified book, which is later on load to be passed as a argument to the PdfPinchView method

A screenshot of a book

Description automatically generatedA screenshot of a book

Description automatically generated****

**Search screens**

The **Search screen** alows user to filter the available books based on the user’s input.

We have used a **TextField,** for recieving user’s input, the inputed string is then utilized by a function logic that I implemented, where I check, if any of the books data that we have on hand conatins the inputed string, if yes then the books that qualified are stored in a list, which is later on used in **ListViewBuilder** to be displayed.

A group of books on a red background

Description automatically generatedA screenshot of a phone

Description automatically generatedA screenshot of a book

Description automatically generated

**WishList and Cart screen:**

For **Wishlish screen data,** I implemeted a logic where whenever a user pressed on (add to wishlist button), the book details are added as a **subcollection under User’s ID** in firebase database.

This logic, help us in creating a **personlized experince** for eaach of our users.

As for the **Cart screen**, we have used provider package, to maintain the cartItems list and notify the screen of any changes in the state. We have additional, **add, remove methods** too.

A screenshot of a cell phone

Description automatically generatedA screenshot of a book

Description automatically generated