BookNexus Report

Real World Problem Identification

In today's fast-paced world, access to physical books can be limited due to several constraints such as high costs, availability, and location. Many individuals, especially students and book lovers, accumulate books over time, but these often sit unused once read. At the same time, people nearby may be searching for the same books but lack affordable or convenient access. The disconnect between individuals who have books and those looking for them leads to inefficiencies in resource utilization. Traditional book borrowing platforms, such as libraries, often involve long waiting times or limited reach in certain areas.

Proposed Solution

BookNexus is a robust and well-developed mobile application that enables book lovers to find and share books within their local community. By leveraging Firebase's real-time data capabilities, secure authentication, and scalability, the app provides a seamless user experience. Despite encountering various technical challenges during development, each was addressed through effective optimizations, resulting in a stable and reliable platform for local book sharing. This solution not only promotes sustainable book use but also strengthens community ties by enabling book lovers to connect through literature.

1. Introduction

This report outlines the development of a mobile application called BookNexus, aimed at providing a platform for users to explore, buy, sell, and share books. The application features an onboarding experience, user authentication (sign-up, login, Google sign-in, and forgot password), and a marketplace for books in different categories such as Anime, Romance, and Horror. A key feature of BookNexus is its ability for users to list their own books for sale, including uploading book details, photos, and even PDFs for others to read. The project follows Clean Architecture principles to maintain an understandable, scalable, and maintainable codebase. We have implemented reusable components to ensure the app remains lightweight and efficient. The application uses Firebase for backend services and GetX for state management.

2. Project Structure

2.1. Clean Architecture

To ensure the project is structured in a scalable and maintainable way, the app was built using Clean Architecture principles. This approach divides the application into distinct layers: Presentation, Domain, and Data. This structure separates concerns, making the app easier to test, maintain, and scale.

Presentation Layer: Handles the UI of the app, including the screens, widgets, and animations.

Domain Layer: Contains business logic, use cases, and domain models.

Data Layer: Responsible for handling data from Firebase (both Firestore and Firebase Authentication).

The clean architecture approach ensures that each module in the project is independent, easily replaceable, and testable.

2.2. Reusability

A key focus of BookNexus is reusable code. We have implemented reusable widgets such as custom buttons, input fields, and cards for books to keep the app’s size minimal and improve performance. This approach also promotes DRY (Don’t Repeat Yourself) principles, ensuring that common components are only written once and can be used across the entire application.

3. Functional Overview

3.1. Onboarding Screens

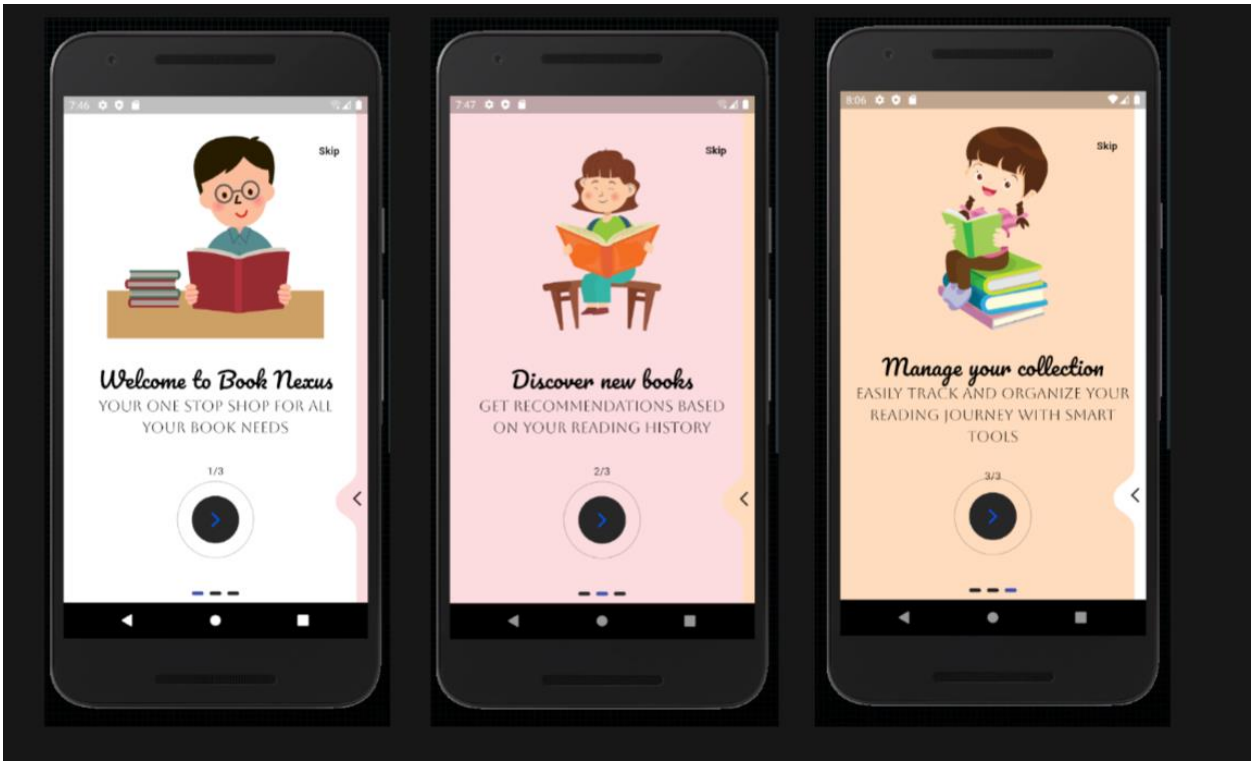
When users first install and open the BookNexus app, they are guided through three onboarding screens:

Introduction to BookNexus: An overview of the app's purpose and key features.

Explore Categories: Displays a variety of book categories such as Anime, Romance, and Horror.

Buy & Sell Books: Explains how users can buy books and sell their own books by uploading book details, photos, and even PDFs for other users to read.

These onboarding screens are designed with appealing graphics and animations to engage users.



3.2. Authentication

After onboarding, users have the option to either Log in or Sign Up for a new account. The app provides authentication through Firebase Authentication to handle the entire process.

3.2.1. Sign Up

Users can create a new account using the Sign-Up form, which collects:

Email Address

Password

Confirm Password

The form validates the inputs and creates a user account using Firebase. Once the account is created, the user is redirected to the main app interface.

3.2.2. Login

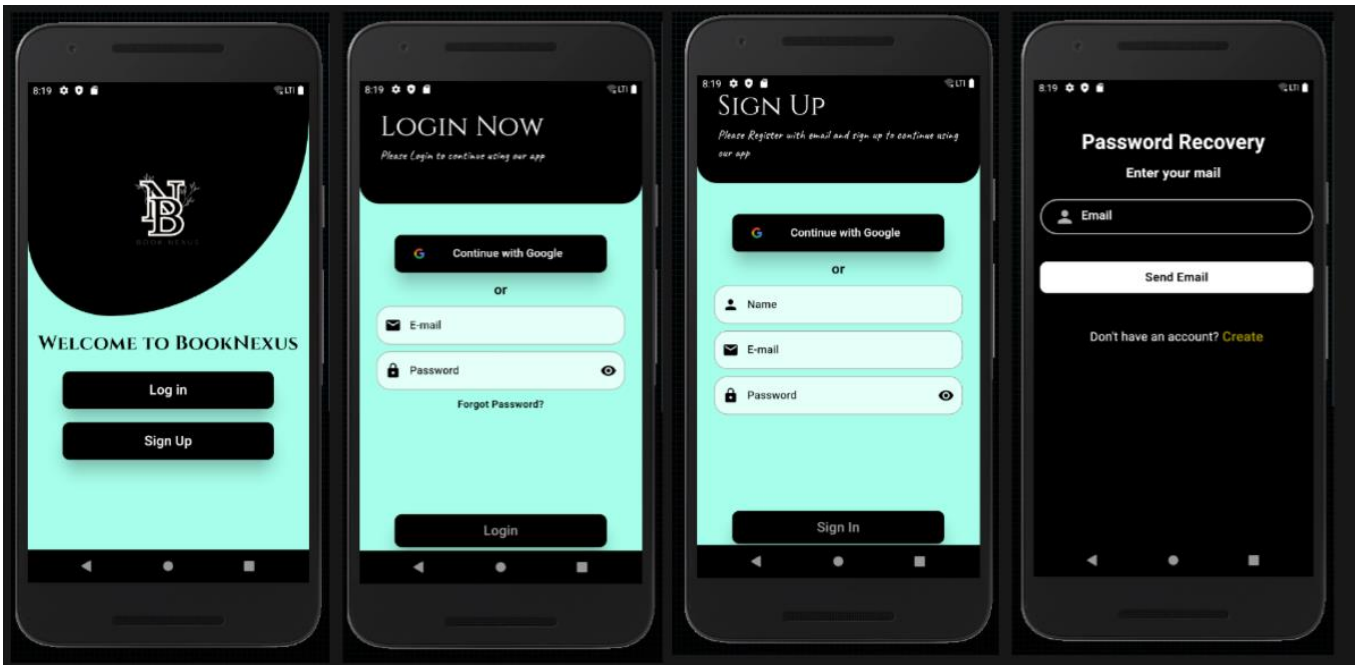
For existing users, there is a Login form where they can enter their credentials to access the app. Validation for incorrect credentials or unregistered accounts is managed by Firebase Authentication.

3.2.3. Sign in with Google

We have integrated Google Sign-In for users who prefer using their Google account for authentication. This process is simplified through Firebase, which securely manages OAuth-based authentication.

3.2.4. Forgot Password

A Forgot Password option allows users to reset their password by receiving an email link sent from Firebase. This helps users regain access to their accounts in case they forget their password.



3.3. Main Features

After a successful login or sign-up, users are taken to the main app interface, which showcases the following features:

3.3.1. Book Categories

Users can explore books from various categories, such as:

Anime

Romance

Horror

Thrillers

Sci-Fi, and more.

Each category is presented in a grid or list view for easy browsing.

3.3.2. Book Details

Upon selecting a book, users can view the detailed description, price, and other related information about the book.

3.3.3. Sell a Book

BookNexus provides users with the ability to sell their own books. This feature includes:

Capture Front Page: Users can capture or upload a photo of the book's front cover.

Set Price: Users can set a selling price for their book.

Add Description: Users can write a brief description of the book.

Upload PDF: Optionally, users can upload a PDF of their book, making it available for other users to read digitally.

This feature allows users to share their books with the community while generating some income.

3.3.4. Buy Books

Users can view books for sale by other users and make a purchase. They can sort books by price, category, and popularity. Users can also view sellers' profiles and contact them directly within the app for further details.

4. State Management

For state management, we have chosen GetX. It allows us to efficiently manage and update the UI without unnecessary overhead. GetX is ideal for this project due to its simplicity, flexibility, and high performance.

Simple State Management: Manages states like user login status, books lists, and form input fields without excessive boilerplate.

Reactive Programming: Automatically updates the UI when data changes, making the app highly responsive.

Dependency Injection: GetX allows us to easily inject services such as FirebaseAuth, Firestore, and user data into different parts of the app.

5. Firebase Integration

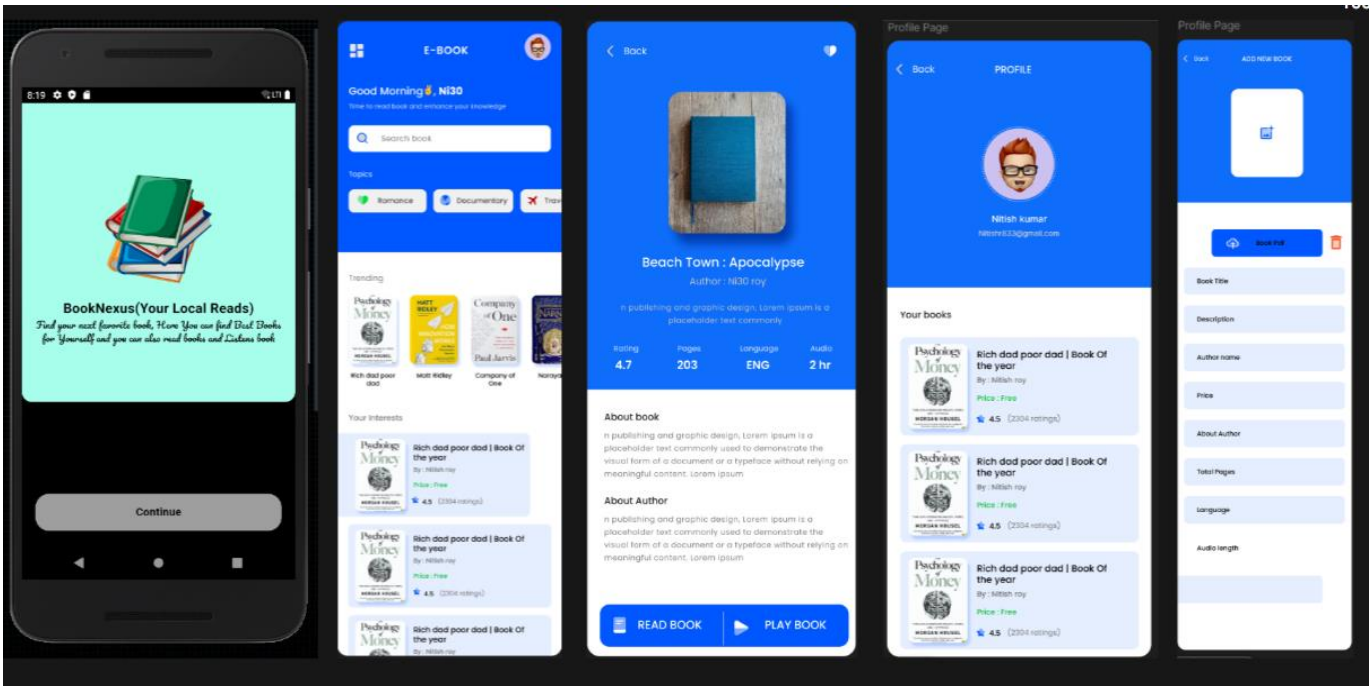
The backend of the application is powered by Firebase, providing scalable and secure services for authentication and database management.

Firebase Authentication: Handles user login, sign-up, Google sign-in, and password reset functionalities.

Cloud Firestore: A NoSQL database is used to store books, user profiles, book categories, and transactions between users.

Firebase Storage: Manages the storage of uploaded files like book cover images and PDF files.

The use of Firebase ensures real-time updates and efficient data management.



6. Issues and Bugs Encountered and Resolved during Development

During the development of our project, we encountered a few minor issues and bugs in our code, particularly with aspects like Firebase authentication. These issues required some troubleshooting but were resolved without any major delays. As Flutter developers in our group, we are pleased to report that no significant problems or critical bugs have emerged thus far. We have maintained a proactive approach in testing and debugging, ensuring that small issues do not escalate. However, if any major issues arise in the future, we are prepared to address them thoroughly and will provide a detailed explanation of any such challenges during our presentation. Our focus remains on delivering a smooth and reliable user experience, and we are committed to resolving any obstacles that come our way.

7. Future Enhancements

Chat Feature: Real-time communication between buyers and sellers for seamless transactions.

Book Recommendations: Personalized book recommendations based on user reading preferences.

Payment Integration: Enabling in-app purchases with payment gateway integration like Stripe or PayPal.

Advanced Search and Filters: Enhanced search capabilities to filter by price range, book condition, and location.