



*ZAID  
AHMED*

20SW003

I focused on the backend development of our Flutter project, integrating Firebase Authentication and Cloud Firestore. My ensured a secure authentication system and efficient data management.

# E-BOOK

MOBILE APPLICATION



*FARHEEN  
QAZI*

20SW055

I took charge of frontend development, designing the user interface for our Flutter application. My resulted in an intuitive and visually appealing user experience.



## MOBILE APPLICATION DEVELOPEMENT PROJECT REPORT

The purpose of this report is to document and communicate the key aspects of our Flutter project, including its functionality, contributions, and technologies used.

ASSIGNED BY:

*MA'AM MARYAM  
MEMON*

## **PROJECT TITLE**

# ***E-Book Application: Bridging the Digital Divide***

## **INTRODUCTION**

The e-book application project aims to develop a user-friendly digital platform that addresses the real-world problem of limited access to traditional printed books and educational resources, particularly in underserved communities. This report outlines the problem identification, proposed solution, and key aspects of the project.

## **REAL-WORLD PROBLEM IDENTIFICATION**

Access to traditional printed books and educational resources has long been a cornerstone of learning and personal development. However, in today's increasingly digital world, a significant portion of the population still faces a formidable challenge – limited access to printed books, especially in underserved communities, remote areas, and economically disadvantaged regions. This glaring digital divide has a profound impact on education, literacy rates, and overall access to knowledge.

## **PROPOSED SOLUTION**

In response to the pressing problem of limited access to printed books and educational resources, our e-book application project proposes a comprehensive digital solution designed to bridge the digital divide and empower individuals, communities, and educational institutions. Our solution focuses on accessibility, affordability, and user-friendliness to ensure that everyone, regardless of their geographic location or economic status, can access a vast repository of e-books and educational materials.

## **APPLICATION FEATURES**

Our e-book application encompasses a range of features tailored to address the identified problem effectively:

### **Extensive E-Book Catalog:**

- We curate an extensive catalog of e-books across various genres, from classic literature to educational textbooks.
- Users have access to a diverse selection of reading materials to suit their interests and learning needs.

### **User-Friendly Interface:**

- The application boasts an intuitive and user-friendly interface, ensuring ease of navigation and reading for users of all ages and backgrounds.
- Special attention is given to accessibility features to accommodate users with disabilities.

### **Educational Resources:**

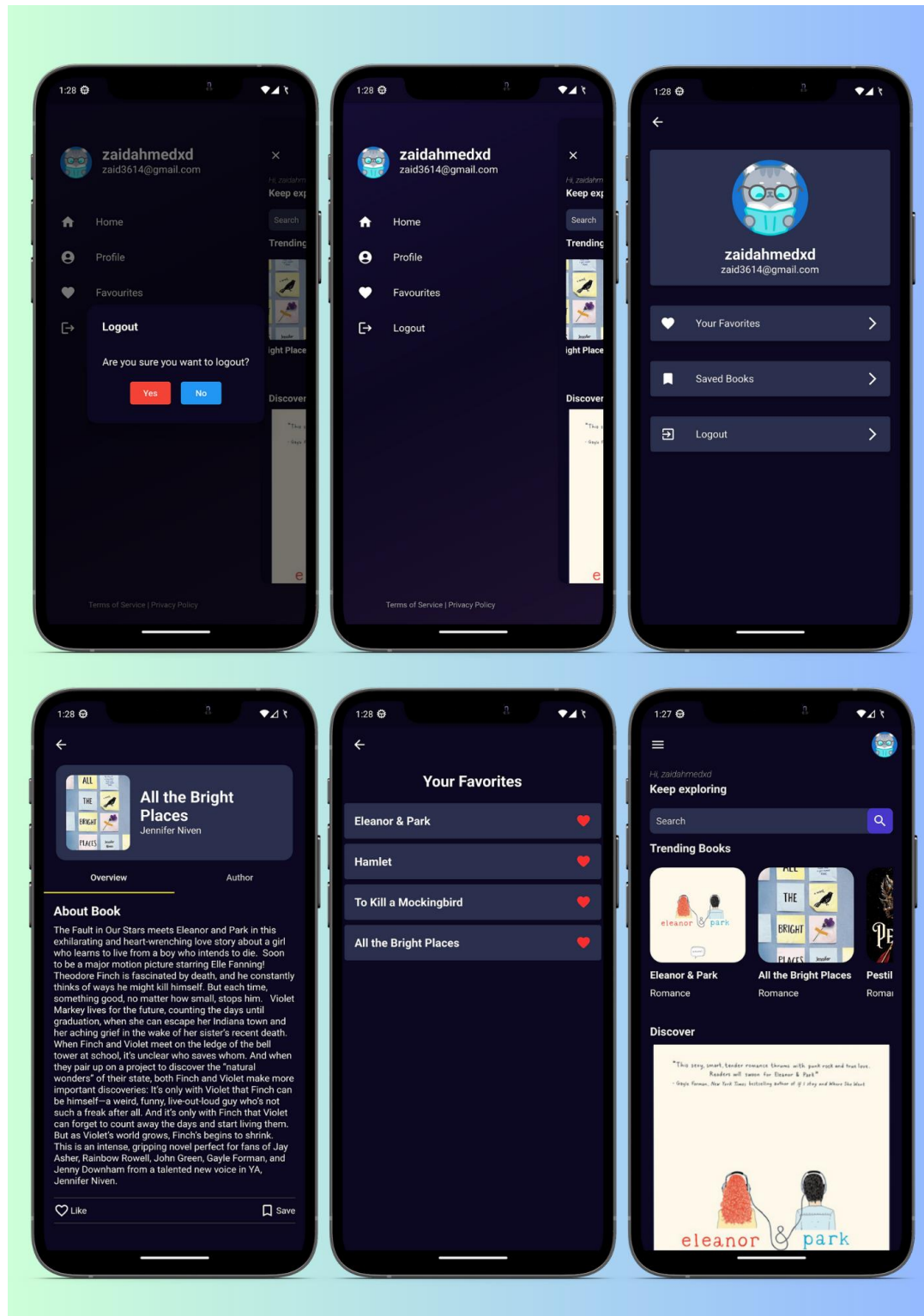
- Beyond e-books, our application offers supplementary educational resources, including interactive quizzes, video tutorials, and study guides, enhancing the learning experience.

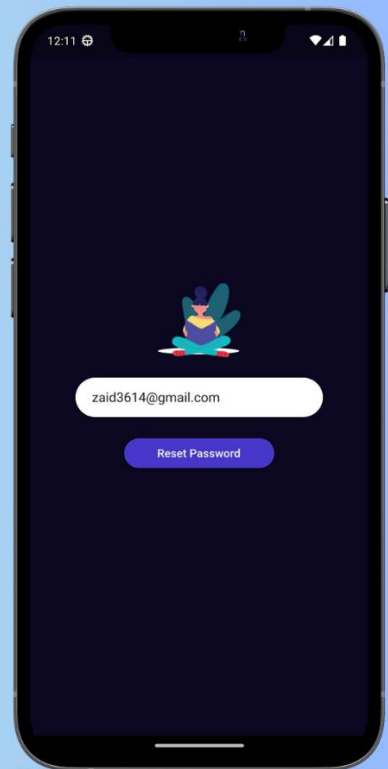
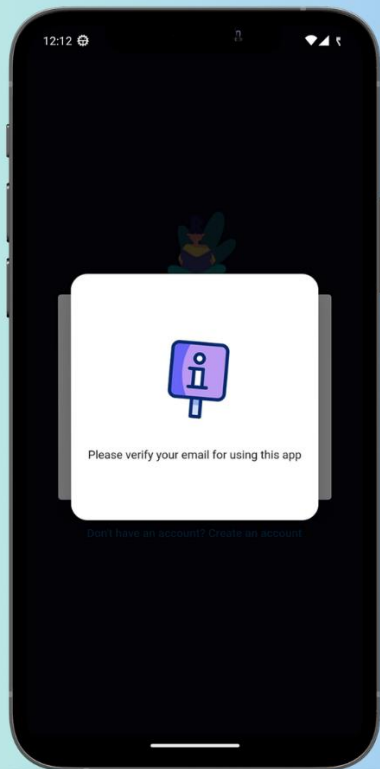
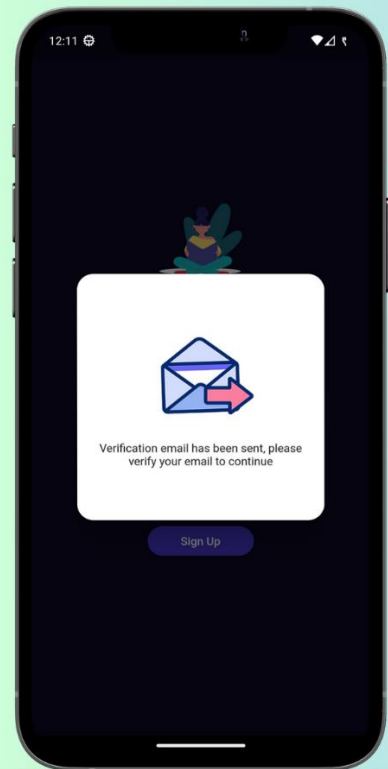
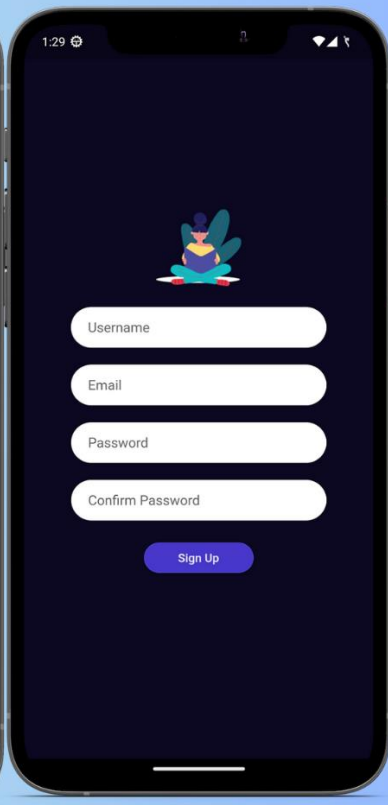
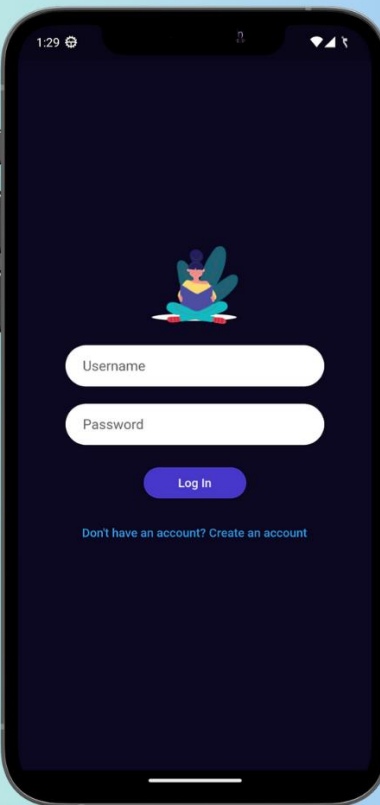
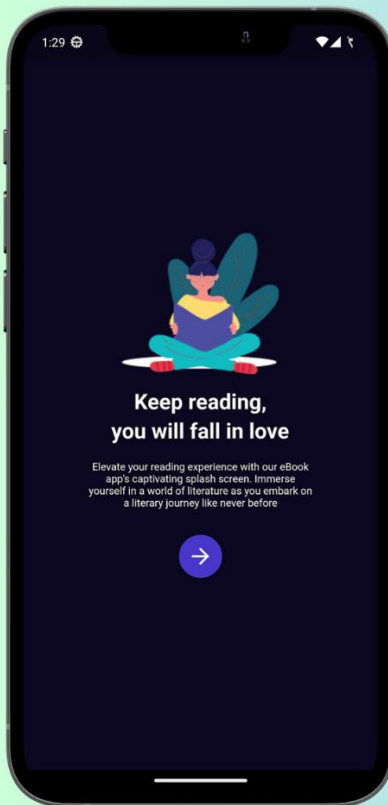
### **Target Audience and Impact:**

Our target audience includes:

- Students in underserved schools and educational institutions.
- Individuals in remote areas with limited access to physical libraries.
- Adults seeking self-improvement, career advancement, or personal enrichment through reading and learning.

## RESPONSIVE USER INTERFACE

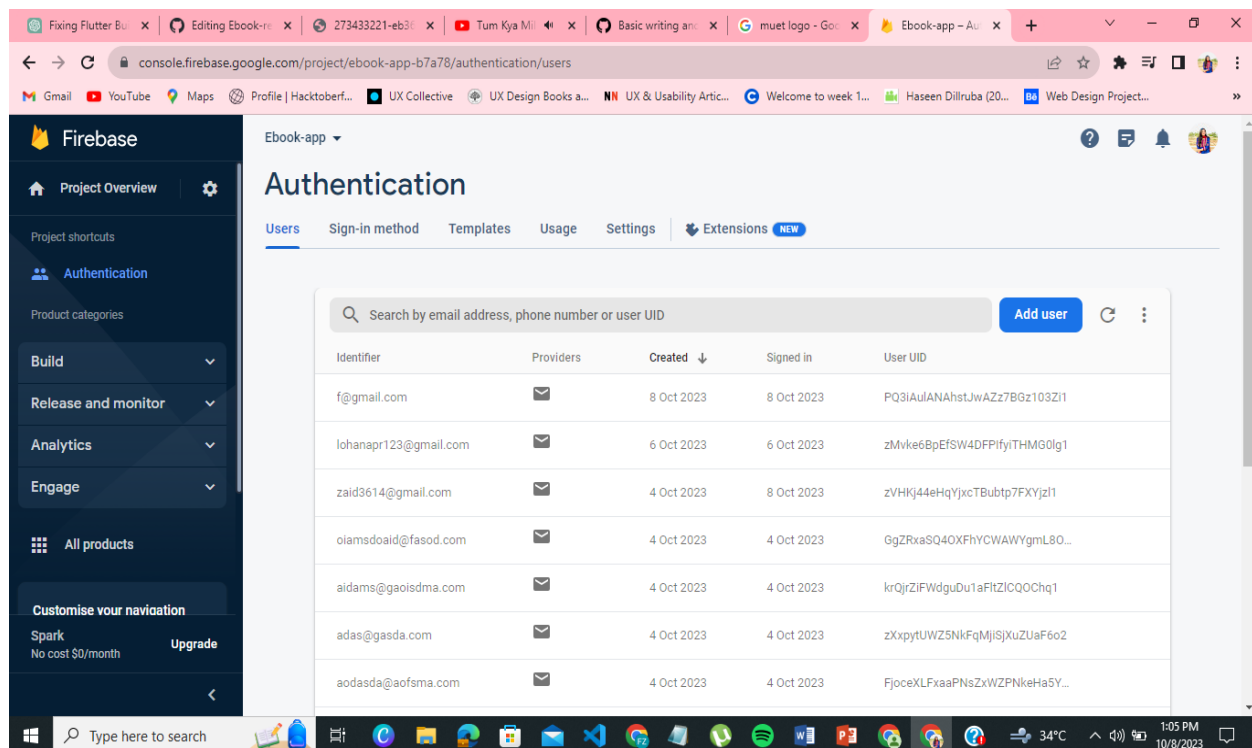




## DATA STORAGE

In our Flutter project, we have implemented a robust authentication system using Firebase Authentication. This authentication system covers key functionalities, including user registration (sign-up), login, and password recovery (forgot password). With Firebase Authentication, users can create accounts, securely log in, and easily reset their passwords if needed, ensuring a seamless and secure user experience.

Additionally, for managing and storing the data related to our project, particularly the information about books, we have integrated Firebase Cloud Firestore. Cloud Firestore serves as our database solution, enabling us to efficiently store, retrieve, and manage book data. This allows us to provide users with real-time access to the latest book information while ensuring data consistency and reliability.

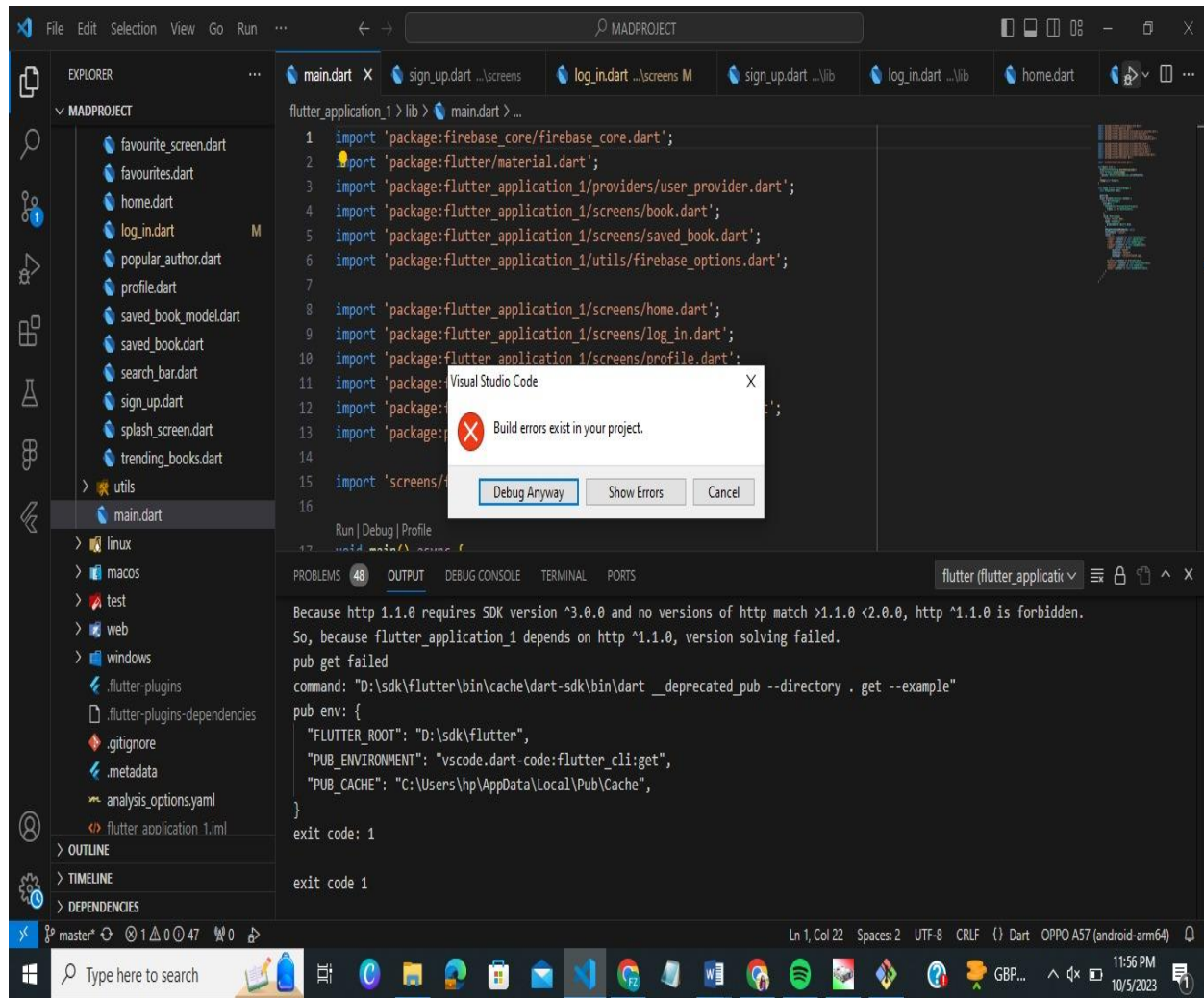


The screenshot displays the Firebase Authentication console for the 'Ebook-app' project. The left sidebar shows the 'Authentication' section under 'Project Overview'. The main area is titled 'Authentication' and includes tabs for 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. The 'Users' tab is active, showing a table of users with columns for Identifier, Providers, Created, Signed in, and User UID. A search bar at the top of the table allows searching by email address, phone number, or user UID. An 'Add user' button is located next to the search bar.

Identifier	Providers	Created	Signed in	User UID
f@gmail.com		8 Oct 2023	8 Oct 2023	PQ3iAulANahstJwAZz7BGz103Z1i
Iohanapr123@gmail.com		6 Oct 2023	6 Oct 2023	zMvke6BpEFSW4DFPifYITHMG0lg1
zaid3614@gmail.com		4 Oct 2023	8 Oct 2023	zVHKj44eHqYjxcTBubtp7FXyz1
oiamsdoaid@fasod.com		4 Oct 2023	4 Oct 2023	GgZRxaSQ40XfHYCWAwygmL80...
aidams@gaolsdma.com		4 Oct 2023	4 Oct 2023	krQjrZiFwDguDu1aFHzICQ0Chq1
adas@gasda.com		4 Oct 2023	4 Oct 2023	zXxpytUWZ5NkFqMjSjXuZuA6o2
aodasda@aofsma.com		4 Oct 2023	4 Oct 2023	FjoceXLFxaaPNsZxWZPNkeHa5Y...

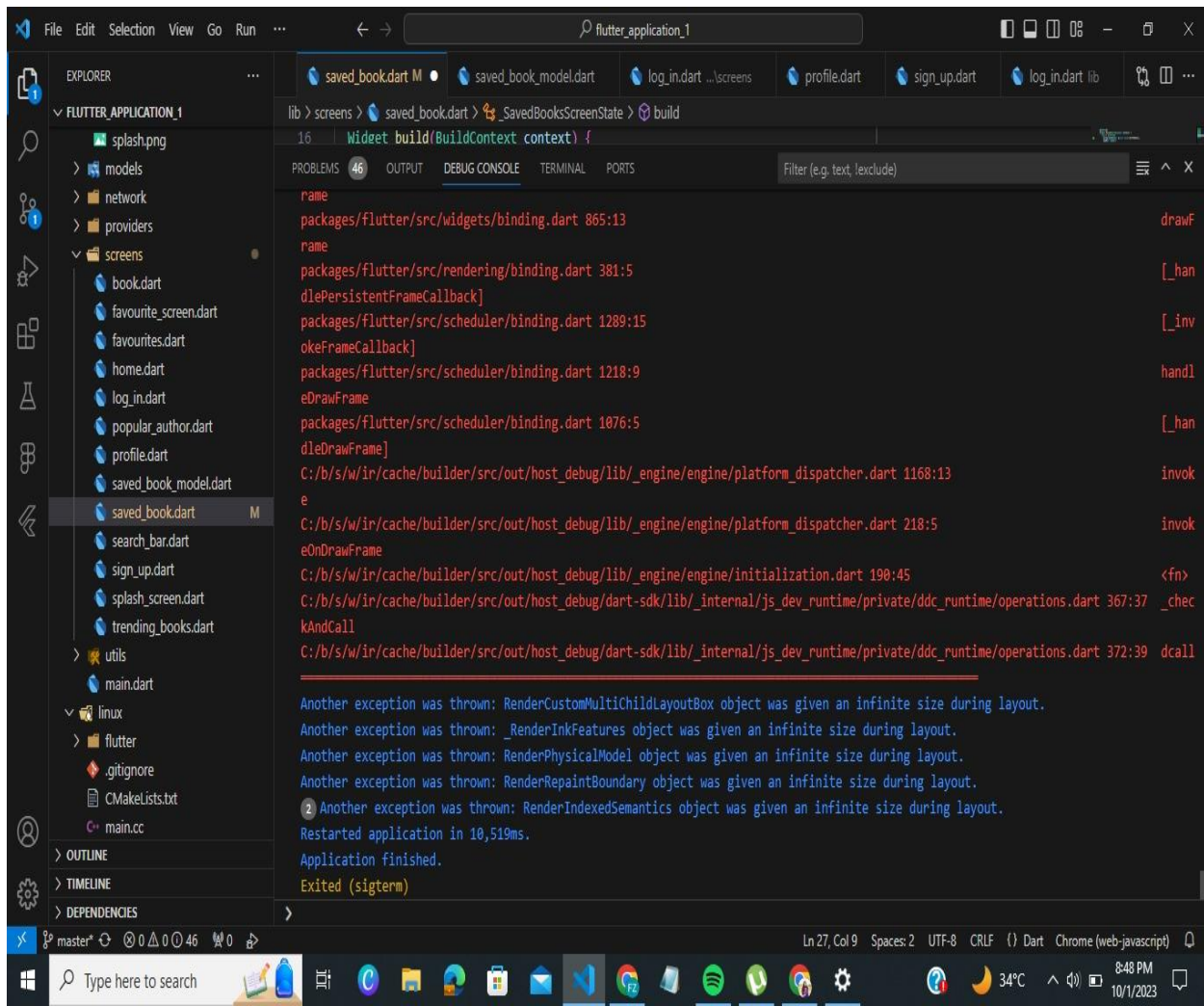


## ERRORS AND BUGS



### *How we resolved this issue?*

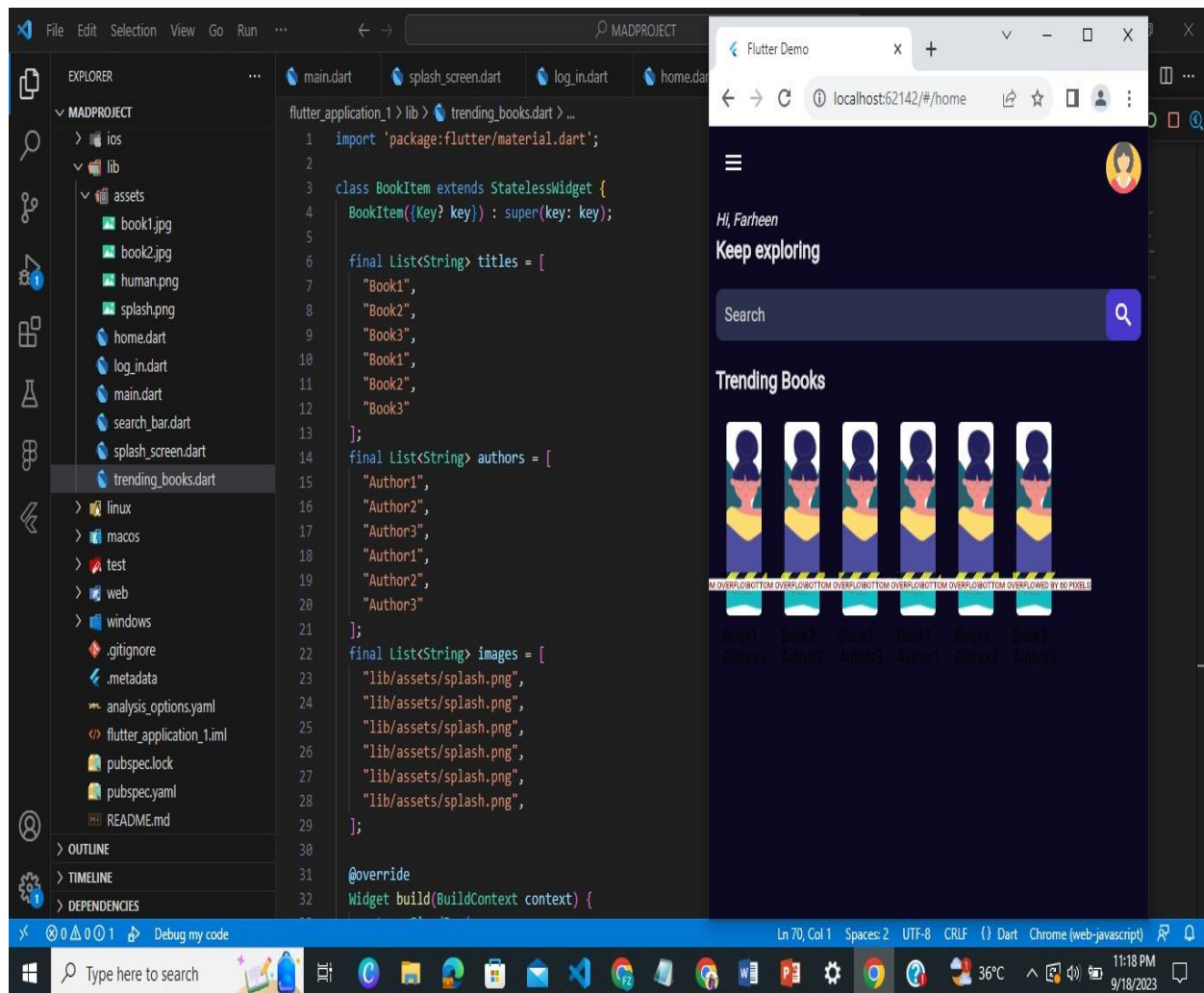
To resolve the issue in our Flutter project, we employed the "clean and rebuild" method, a common troubleshooting technique. This involved executing the flutter clean command to remove temporary build files and then running flutter pub get to ensure our dependencies were up-to-date. Finally, we used flutter build to rebuild the project from scratch. This approach helped eliminate any potential caching or dependency-related issues, ultimately resolving the problem. We have documented this process in our Flutter project report as a recommended troubleshooting step for addressing build errors efficiently.



## How we resolved this issue?

In our Flutter project, we encountered the error "Another exception was thrown: RenderCustomMultiChildLayoutBox object was given an infinite size." This error typically arises when a widget within our layout does not have constraints to determine its size, causing layout ambiguity. To resolve this issue, we carefully inspected the widget tree and identified the specific widget or widgets triggering the error. We then applied constraints or sizing parameters to ensure that these widgets had a defined size. This action allowed the layout engine to properly calculate dimensions, ultimately eliminating the "infinite size" error. In our project report, we have documented this process as a crucial step in maintaining a well-defined and constraint-aware widget hierarchy to prevent layout ambiguities and build errors.





## How we resolved this issue?

In our Flutter project, we encountered an "OverflowBottom" error specifically within the "Training Books" section, which was implemented using a **ListView**. This error occurred when the content within the ListView exceeded the available vertical space, causing overflow issues. To fix the overflow issue, we made adjustments to the sizing and layout properties of the widgets within the **ListView**. We ensured that the content fit within the available vertical space without causing overflow.

Given the nature of this section, which likely contains a large number of training books, we implemented a scrollable **ListView**. This allowed users to scroll through the list of books, ensuring that they could access all the content without any overflow issues.

The  
End