

WANDERPLAN

INTERDISCIPLINARY PROJECT

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in Information Technology

by

MIRUDHUBASHINI B S (41120110)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Category-1 University by UGC

Accredited with Grade “A++” by NAAC

12B Status by UGC |Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

May 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

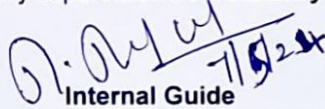
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Mirudhubashini B S (41120110)** who carried out the project entitled "**WanderPlan**" under my supervision from **January 2024 to May 2024**.

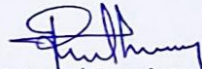

Internal Guide

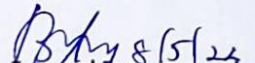
Ms. S. SAGEENGRANA, M.E.,



Head of the Department
Dr. R. SUBHASHINI M.E., Ph.D.

Submitted for Viva voce Examination held on 8/5/2024


Internal Examiner


External Examiner

DECLARATION

I, **MIRUDHUBASHINI B S (41120110)**, hereby declare that the Project Report entitled "**WANDERPLAN**" done by me under the guidance of **Ms. S. Sageengrana, M.E.**, is submitted in partial fulfillment of the requirements for the award Bachelor of Technology degree in **Information Technology**.

DATE: **8.5.2024**

PLACE: **CHENNAI**



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. **T. Sasikala, M.E., Ph. D, Dean**, School of Computing, **Dr. R. SUBHASHINI, M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. S. Sageengrana, M.E.**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

CERTIFICATION OF COMPLETION



HACKWIT TECHNOLOGIES PVT LTD
Redefine by Technologies

149, 1C, 2nd Floor, Mount Poonamallee Rd,
Ramapuram, Chennai - 600089.
admin@hackwittechnologies.com
www.hackwittechnologies.com

10.02.2024

CERTIFICATE FOR INTERNSHIP

TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **Mirudhubashini B S (Reg No: 41120110)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22nd January to 30th January 2024.

During the internship, **Mirudhubashini B S** has closely worked as a part of the **Flutter** by carrying out the project entitled "**Wander Plan**".

She demonstrated good developing and learning skill with self-motivated attitude to learn new things. Her contribution to implement the Flutter was good. We wish her all the best for her future endeavors.

Best Regards,

Ravikumar S
CEO & Managing Director
Hackwit Technologies Pvt Ltd



CERTIFICATION OF COMPLETION



ABSTRACT

WanderPlan is a groundbreaking platform developed using Flutter, aimed at redefining the landscape of travel planning through the integration of social media networking and itinerary management. This innovative application empowers users to create, share, and explore travel itineraries in a collaborative and interactive environment. By harnessing the capabilities of Flutter, WanderPlan offers a seamless and engaging user experience across web and mobile platforms. WanderPlan enables users to effortlessly craft detailed itineraries, discover new destinations, and connect with fellow travellers who share similar interests. Through its intuitive interface, users can easily collaborate with friends and family, leveraging their collective knowledge and experiences to curate memorable travel experiences. At its core, WanderPlan seeks to foster a vibrant community of travellers, facilitating meaningful connections and sharing of valuable insights and recommendations. By merging the functionalities of social media networking and itinerary planning, WanderPlan revolutionizes the way individuals approach travel planning, offering a dynamic platform that enhances engagement and enriches the overall travel experience. With WanderPlan, users can embark on their next adventure with confidence, knowing that they have access to a robust and supportive community that is passionate about exploring the world and creating unforgettable memories together.

TITLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	vii
	LIST OF ABBREVIATIONS	x
	LIST OF FIGURES	xi
1	INTRODUCTION	1
	1.1 FLUTTER	2
	1.2 DART	3
	1.3 FIREBASE	4
	1.4 PROBLEM STATEMENT	5
	1.5 OBJECTIVES	7
2	METHODOLOGY	10
	2.1 INITIAL PROJECT REQUIREMENTS	10
	2.2 DATA FLOW DIAGRAM	13
	2.3 FLUTTER ARCHITECTURE	14
	2.4 MODULES	15
	2.4.1 USER MANAGEMENT	15
	2.4.2 PROFILE MANAGEMENT	17
	2.4.3 SESSION MANAGEMENT	19
	2.5 DEVELOPMENT AND DEPLOYMENT SETUP	21

3	RESULT AND DICUSSION	25
4	CONCLUSION	30
4.1	CONCLUSION	30
4.2	FUTURE WORK	31
	REFERENCES	32
	APPENDICES	34
A	SAMPLE CODE	34
B	SCREENSHOTS	36

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
UI	- User Interface
iOS	- iPhone Operating System
UX	- User Experience
UAT	- User Acceptance Testing
IDE	- Integrated Development Environment

LIST OF FIGURES

Figure No	Name of the Figure	Page No.
2.1	Data Flow Diagram	13
2.2	Architecture of Flutter	14
2.3	Project Creation	22
2.4	Firebase Connection	23
2.5	Firebase Authentication	23
2.6	Flutter Run	24
3.1	Main Dart	25
3.2	Login Page	26
3.3	Signup Page	26
3.4	Dashboard Page	27
3.5	Profile Page	28
3.6	Profile Update Page	28
3.7	Firebase Configuration	29

CHAPTER 1

INTRODUCTION

In today's dynamic and fast-paced world, effective task management is paramount for individuals and teams alike to stay organized, meet deadlines, and accomplish their objectives efficiently. Traditional methods of task management, such as handwritten to-do lists or basic digital reminders, often lack the comprehensive features and flexibility required for modern-day productivity. Recognizing the need for a more sophisticated and streamlined approach to task management, WanderPlan emerges as a revolutionary solution. WanderPlan is not just another task management application; it represents a paradigm shift in how tasks are organized, tracked, and executed.

At its core, WanderPlan is designed to simplify the complexities of task management, providing users with a robust and feature-rich platform to optimize their productivity. By leveraging the innovative Flutter framework, WanderPlan delivers a seamless and intuitive user experience across various platforms, including Android and iOS devices. Flutter's versatility and efficiency enable WanderPlan to offer a consistent and responsive interface, ensuring that users can access their tasks anytime, anywhere, without compromising on performance or functionality.

Moreover, WanderPlan goes beyond basic task tracking and organization. It offers a comprehensive suite of tools and features to enhance productivity, including customizable task lists, priority settings, deadline reminders, and collaboration capabilities. Whether it is managing personal tasks, coordinating team projects, or organizing complex workflows, WanderPlan provides users with the flexibility and agility they need to adapt to their unique requirements and workflows.

1.1 FLUTTER

Flutter, a robust cross-platform UI toolkit developed by Google, is at the heart of WanderPlan, shaping the application's foundation and user interface. The utilization of Flutter brings a paradigm shift to the traditional app development approach, offering a single codebase that transcends platform-specific limitations. This innovation allows WanderPlan to deliver a visually captivating and responsive user interface, adeptly adjusting to both iOS and Android environments.

The efficiency of Flutter lies in its ability to streamline the development process, significantly reducing the complexities associated with managing separate codebases for different platforms. Developers benefit from a cohesive workflow, resulting in accelerated development cycles and enhanced productivity. Moreover, Flutter's hot reload feature enables real-time code changes and instant visual feedback, facilitating quick iterations and adjustments.

One of the key advantages of Flutter is its capability to ensure a consistent and high-quality user experience across diverse devices. By leveraging Flutter's rich set of customizable widgets, WanderPlan achieves a cohesive design language and smooth interactions, irrespective of the underlying operating system. Users can seamlessly transition between devices without compromising on the application's performance or aesthetics.

In essence, Flutter empowers WanderPlan to provide a unified cross-platform experience, where the focus remains on delivering an exceptional user interface and optimal performance. The adoption of Flutter aligns with the project's commitment to technological excellence, enabling WanderPlan to stand out as a modern and user-friendly platform.

1.2 DART

Unified Codebase:

WanderPlan relies on Dart, a client-optimized programming language, for implementing backend logic and server-side operations. Dart's versatility enables developers to maintain a unified codebase for both the frontend (Flutter) and backend, fostering consistency and ease of collaboration.

Asynchronous Programming:

Dart's support for asynchronous programming is crucial for handling concurrent operations efficiently. This feature ensures that WanderPlan can seamlessly manage multiple tasks, such as handling user requests, database operations, and external API calls, without causing delays or disruptions in the application's responsiveness.

Scalability:

Dart's scalability is instrumental in accommodating the potential growth of WanderPlan. As the user base expands and the demand for usage increases, Dart's ability to scale allows the platform to handle higher traffic, ensuring a smooth and responsive experience for all users.

Strong Typing System:

Dart's strong typing system enhances code reliability and maintainability in WanderPlan. By catching potential errors during the development phase, Dart enables developers to write robust and error-resistant code, reducing the likelihood of runtime issues and enhancing the overall stability of the application.

Developer-Friendly Syntax:

Dart's clean and developer-friendly syntax contributes to the efficiency of WanderPlan's codebase. With readable and expressive code, developers can easily comprehend and maintain the logic, fostering a collaborative development environment and facilitating the onboarding of new contributors to the project.

1.3 FIREBASE**Real-time Data Synchronization:**

Firebase Cloud Firestore supports real-time data synchronization, allowing seamless updates across multiple devices. This feature is crucial for WanderPlan, enabling users to experience instant changes in discussions, travel plans, and collaborative interactions.

Firebase Authentication for User Security:

Firebase Authentication provides a secure and straightforward user authentication process. WanderPlan leverages this feature to ensure that only authorized users can access the platform. Multi-factor authentication and identity verification contribute to a robust security layer.

Serverless Cloud Functions:

Firebase offers serverless cloud functions that enable the execution of backend logic without managing servers. WanderPlan utilizes these functions for various tasks, such as sending notifications, updating user data, and handling asynchronous operations. This serverless architecture enhances scalability and efficiency.

Firebase Cloud Storage for Media Assets:

Storing media assets, such as profile pictures and discussion attachments, is streamlined with Firebase Cloud Storage. WanderPlan utilizes this feature to efficiently manage and serve multimedia content, ensuring a smooth and responsive user experience for users.

Analytics and Performance Monitoring:

Firebase Analytics provides valuable insights into user behavior and engagement within WanderPlan. Additionally, Firebase Performance Monitoring tracks the app's performance, identifying and addressing bottlenecks or issues. These analytics contribute to continuous improvement and optimization of the platform.

1.4 PROBLEM STATEMENT

In today's digital age, the process of planning and organizing travel experiences remains a daunting task for many individuals and groups. Traditional methods of travel planning often involve cumbersome spreadsheets, disjointed communication channels, and an overwhelming amount of information from various sources.

This fragmented approach not only leads to inefficiencies and frustration but also diminishes the overall enjoyment of the travel experience. Moreover, while there are numerous itinerary management tools available, they primarily focus on individual planning and lack robust features for collaborative itinerary creation and sharing.

Additionally, the absence of a centralized platform that seamlessly integrates itinerary planning with social networking functionalities further exacerbates the problem. Travelers are forced to navigate through multiple platforms and tools, leading to user fatigue and decreased engagement. The lack of a comprehensive solution that addresses these pain points highlights a significant gap in the travel planning landscape. There is a pressing need for a platform that not only streamlines the itinerary creation process but also fosters community engagement and facilitates meaningful connections among travelers.

WanderPlan seeks to bridge this gap by offering a holistic solution that combines the power of itinerary management with social networking features. By providing users with a unified platform for collaborative itinerary planning, sharing, and exploration, WanderPlan aims to revolutionize the way individuals and groups plan, experience, and share their travel adventures.

1.5 OBJECTIVES

The primary objective of WanderPlan is to address the fragmented nature of travel planning by offering a comprehensive platform that seamlessly integrates itinerary creation with social networking functionalities.

Firstly, WanderPlan aims to streamline the itinerary creation process by providing users with intuitive tools and customizable templates, enabling them to craft detailed travel plans effortlessly. Secondly, the app seeks to facilitate collaborative planning among users by allowing them to invite friends, family, and fellow travelers to contribute to itinerary creation, share insights, and coordinate travel plans in real-time. Additionally, WanderPlan aims to foster a vibrant community of travelers by integrating social networking features that enable users to connect with like-minded individuals, follow their travel activities, and discover new destinations based on recommendations and insights shared within the platform. Through these objectives, WanderPlan aspires to revolutionize the travel planning experience, empowering users to create memorable adventures and fostering a sense of belonging within a supportive and engaging community of travelers.

Key Challenges

Fragmented Travel Planning:

Current travel planning processes are disjointed and fragmented, with users relying on multiple platforms and tools to organize their trips. This results in inefficiencies and confusion during the planning phase.

Limited Collaboration:

Traditional itinerary management tools lack robust collaboration features, making it difficult for users to plan and coordinate trips with friends, family, or travel companions in real-time.

Security and Privacy:

Ensuring robust security measures to protect user data, particularly in the context of user interactions. Adhering to privacy regulations and standards to build trust among users regarding the confidentiality of their information.

Flutter Codebase Maintenance:

Managing the complexity of a Flutter-based codebase to ensure code readability, maintainability, and scalability. Keeping the codebase up-to-date with the latest Flutter releases and best practices to leverage new features and improvements

Scalability and Performance:

Designing the app architecture to accommodate potential growth in user base and content without compromising performance. Optimizing the performance of WanderPlan to deliver a responsive and efficient experience, especially as the user community expands.

Deliverables

Integrated Travel Planning Platform:

Develop a comprehensive platform that consolidates itinerary creation, collaboration tools, social networking features, and personalized recommendations.

User-Friendly Interface:

Design an intuitive and visually appealing interface across web and mobile platforms, ensuring ease of use and accessibility for users of all levels of tech-savviness.

Data Security and Privacy Measures:

Implement robust security measures to protect user data and privacy, including encryption protocols, secure authentication mechanisms, and compliance with relevant data protection regulations.

Community Moderation and Engagement:

Implementing effective community moderation features to foster a positive and inclusive environment. Encouraging active user engagement, discussions, and contributions while preventing any misuse or inappropriate content within the platform.

CHAPTER 2

METHODOLOGY

2.1 INITIAL PROJECT REQUIREMENTS

The initial project requirements for WanderPlan encompass various aspects such as functionalities, technologies, and packages.

I. Project Scope:

- 1) Develop a mobile application named "WanderPlan."
- 2) Platform aimed at providing users with a comprehensive itinerary planning and social networking experience.
- 3) The application targets travelers, adventurers, and explorers who seek personalized itinerary planning, recommendations, and social interaction with like-minded individuals.

II. Technologies:

- 1) Flutter: Utilize Flutter for cross-platform mobile app development, ensuring a consistent user experience on both iOS and Android.
- 2) Dart: Employ the Dart programming language as the primary language for Flutter app development.
- 3) Firebase: Integrate Firebase for backend services, including authentication, real-time database (Firestore), and cloud storage.

III. Functionalities:

- 1) User Registration and Authentication: Allow users to create accounts, log in securely, and manage their profiles.
- 2) Itinerary Creation and Customization: Enable users to create personalized travel itineraries by adding destinations, activities, attractions, accommodations, and notes.
- 3) Data Privacy Controls: Implement privacy settings and consent mechanisms to give users control over their data and ensure compliance with data protection regulations.
- 4) Offline Access: Allow users to access and update their itineraries, view recommendations, and interact with the app's core features even when offline or in areas with limited connectivity.
- 5) Notifications and Reminders: Send notifications and reminders to users about upcoming trips, itinerary updates, friend requests, and relevant recommendations.
- 6) Feedback and Support: Provide users with channels to submit feedback, report issues, and seek assistance from customer support.

VI. PACKAGES USED

1) cloud_firestore

Version: 4.15.3

Purpose: Provides Firestore, a cloud-hosted, NoSQL database provided by Firebase.

2) cupertino_icons

Version: 1.0.2

Purpose: Provides Cupertino Icons font for iOS style icons.

3) firebase_auth

Version: 4.17.3

Purpose: Provides authentication services for Firebase.

4) firebase_core

Version: 2.25.3

Purpose: Provides the core functionality for Firebase services.

5) firebase_database

Version: 10.4.9

Purpose: Provides access to the Firebase Realtime Database service.

6) fluttertoast

Version: 8.2.4

Purpose: Provides toast messages for Flutter applications.

7) get

Version: 4.6.6

Purpose: Provides state management and dependency injection for Flutter applications.

8) toast

Version: 0.3.0

Purpose: Displaying simple toast notifications.

2.2 DATA FLOW DIAGRAM

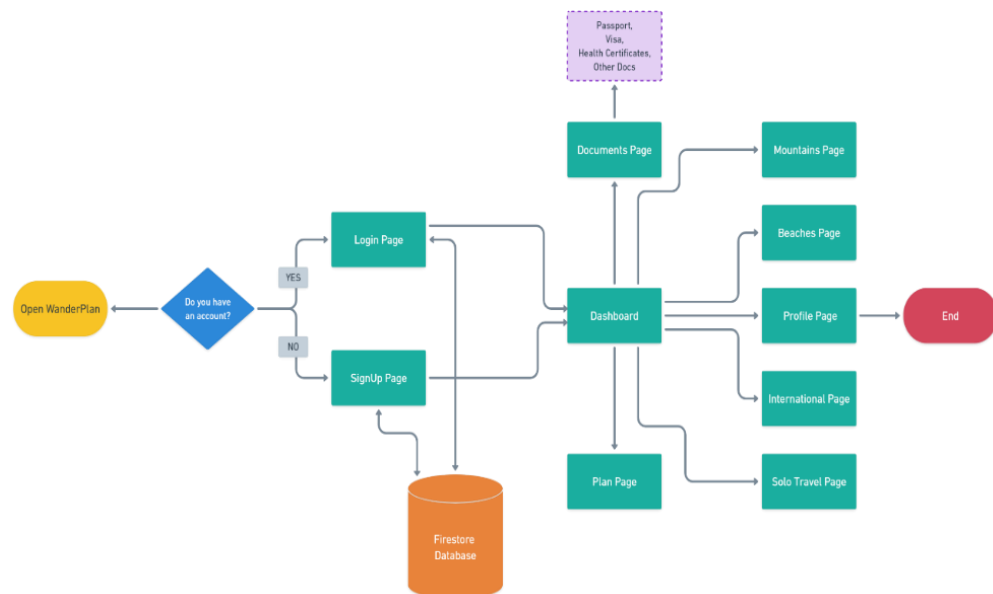


Fig.2.1: Data Flow Diagram

The following steps are used to describe the Fig 2.1 Data Flow Diagram:

- i)The user opens the mobile app.
- ii)The app prompts the user to log in or sign up if they do not have an account yet.
- iii)If the user chooses to sign up, they will be taken to the sign-up page.
- iv)If the user is already registered, they can log in directly.
- v)Once the user is logged in, they will be taken to the dashboard.
- vi)From the dashboard, the user can access various features of the app such as their Profile page, Documents page, Plan page, Mountains page, Beaches page, Solo Travel Page.
- vii)The Profile page shows the details of the user
- viii) The Document Page is used to store Documents needed for travel digitally.
- ix)The Plan Page contains our personal itinerary.
- x)The Mountain, Beaches, International and Solo Travel Page respectively contain itinerary shared by people on the category.
- xi)The app uses Firebase as its backend database to store user data.

2.3 FLUTTER ARCHITECTURE

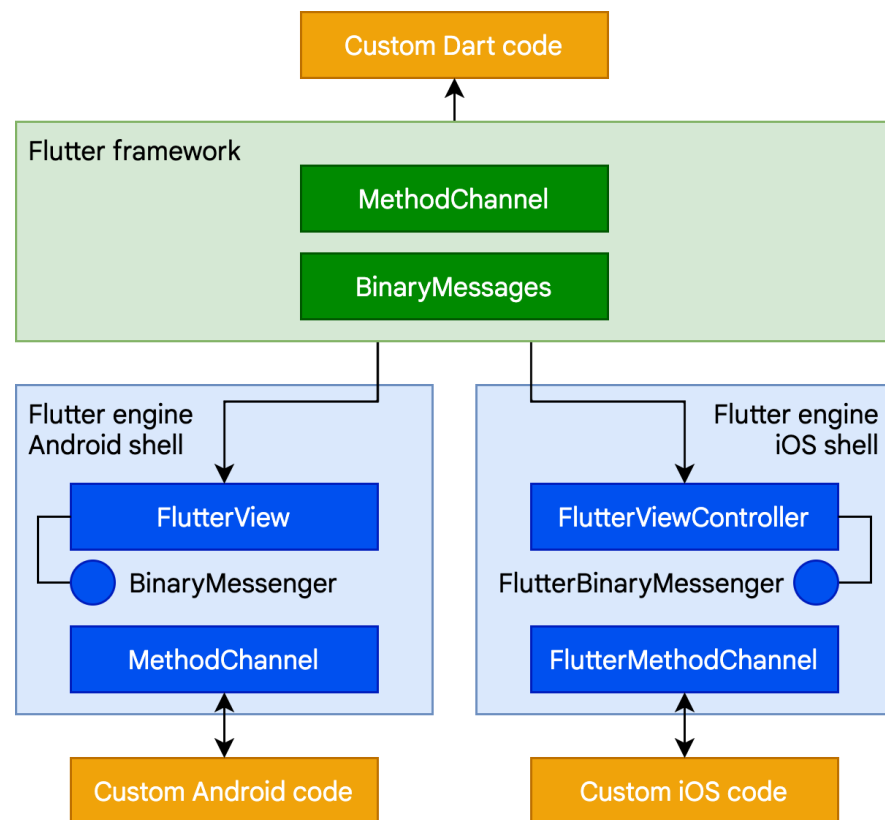


Fig 2.2: Architecture of Flutter

The Fig 2.2 represents the architecture of Flutter for mobile development. Here is a breakdown of the components:

Custom Dart code:

This is the code that you write for your Flutter app using Dart language. It is represented by the yellow box at the top of the diagram.

MethodChannel and BinaryMessages:

These are the communication channels between your Dart code and the platform-specific code. They are represented by the green boxes connected to the Custom Dart code.

Android shell and iOS shell:

These are the parts of the Flutter engine that interact with the specific mobile platform (Android or iOS). They are represented by the middle section of the diagram, which is divided into two parts- Android shell on the left and iOS shell on the right. Each shell contains its own set of components (FlutterView and MethodChannel for Android, FlutterViewController and FlutterMethodChannel for iOS) and a BinaryMessenger.

BinaryMessenger:

This is the message channel that allows Flutter and the host operating system to communicate with each other.

Custom Android code and Custom iOS code:

These are the platform-specific implementations for your app, represented by the orange boxes at the bottom of the diagram. This is where you would write native code for specific functionalities that are not available in Flutter or Dart.

2.4 MODULES

2.4.1 User Management

The User Management module in WanderPlan is essential for maintaining and managing user details and regulations within the platform. Leveraging Firebase as the backend infrastructure and Flutter for the user interface, this module ensures seamless user authentication, registration, and account management.

This module in WanderPlan, responsible for handling user accounts, authentication, and profile management. By utilizing Firebase Authentication services and integrating them into the Flutter user interface, WanderPlan delivers a robust and user-friendly experience for account creation, login, and customization.

The User Management module architecture is built upon the Firebase Authentication system, which provides a secure and scalable solution for user authentication and management. Firebase Authentication offers various authentication methods, including email/password, phone number, and social media authentication (e.g., Google Sign-In, Facebook Login), ensuring flexibility and accessibility for users. The Flutter framework serves as the front-end interface for interacting with Firebase Authentication services, enabling seamless integration and a cohesive user experience.

Functionality

User Registration:

The User Management module facilitates user registration through a streamlined process, allowing users to create accounts using their email address or phone number. Firebase Authentication handles account creation securely, generating unique user identifiers and securely storing user credentials.

User Login:

Once registered, users can log in to their accounts securely using their chosen authentication method. The User Management module verifies user credentials against Firebase Authentication, granting access to authenticated users while maintaining the integrity and security of user sessions.

Account Management:

WanderPlan empowers users to manage their accounts effectively, providing features such as password reset, profile customization, and privacy settings. Through the Flutter user interface, users can update their profile information, modify account preferences, and configure privacy settings to control data sharing and visibility.

Implementation

The implementation of the User Management module involves integrating Firebase Authentication services into the Flutter application using Firebase SDKs and plugins. Flutter's reactive framework facilitates real-time updates and seamless user interactions, ensuring a responsive and intuitive user experience. Firebase Authentication handles the backend processes of user authentication, session management, and account verification, while Flutter manages the presentation layer, rendering user interfaces and handling user interactions.

2.4.2 Profile Management

The Profile Management module in WanderPlan is crucial for handling user-generated content and interactions within the platform. This module allows users to create and customize their profiles, share travel experiences, and engage with other users through posts and comments.

Profile Management is an integral component of WanderPlan, enabling users to personalize their profiles, share travel experiences, and connect with fellow travelers.

By facilitating the creation and management of user-generated content, the Profile Management module enriches the platform's social networking features, fostering engagement and collaboration among users.

The module architecture revolves around the storage and retrieval of user-generated content, including posts, comments, and profile information. WanderPlan utilizes cloud-based storage solutions, such as Firebase Firestore, to store user data securely and efficiently. The Flutter framework serves as the frontend interface for interacting with Firestore databases, enabling seamless data synchronization and real-time updates across devices.

Functionality

Profile Creation:

The Profile Management module allows users to create personalized profiles, providing details such as their name, bio, profile picture, and travel interests. Users can customize their profiles to reflect their personality and preferences, enhancing their visibility and engagement within the WanderPlan community.

Post Creation and Sharing:

Users can create and share posts detailing their travel experiences, adventures, and recommendations. The Profile Management module facilitates the creation, editing, and deletion of posts, enabling users to curate their travel memories and share them with others.

Social Interactions:

WanderPlan encourages social interactions among users through features such as likes, comments, and follows. Users can engage with posts shared by others, leave comments, and express appreciation for inspiring travel content. The Profile Management module manages social interactions, facilitating communication and collaboration within the WanderPlan community.

Implementation

The implementation of the Profile Management module involves integrating Firestore databases with the Flutter application, enabling seamless data storage and retrieval. Flutter's reactive framework facilitates real-time updates and synchronization of user-generated content, ensuring a dynamic and engaging user experience. Firestore serves as the backend infrastructure for storing and managing user profiles, posts, and interactions, while Flutter handles the presentation layer, rendering user interfaces and managing user interactions.

2.4.3 Session Management

The Session Management module in WanderPlan is responsible for ensuring secure and uninterrupted user sessions within the platform. By managing user authentication tokens, session timeouts, and session persistence, this module enhances the security and usability of WanderPlan, providing users with a seamless travel planning experience.

Session Management is a critical component of WanderPlan, tasked with managing user sessions and authentication tokens to ensure secure and reliable access to the platform. By implementing session timeouts, session persistence, and token-based authentication mechanisms, the Session Management module enhances the security and usability of WanderPlan, safeguarding user accounts and data from unauthorized access.

The Session Management module architecture revolves around the generation, validation, and expiration of user authentication tokens. WanderPlan utilizes token-based authentication mechanisms, such as JSON Web Tokens (JWT), to authenticate users and manage their sessions securely. The Flutter framework, coupled with Firebase Authentication services, facilitates token generation and validation, ensuring seamless user authentication and session management.

Functionality

User Authentication:

The Session Management module handles user authentication processes, verifying user credentials and issuing authentication tokens upon successful login. These tokens serve as proof of authentication and are used to grant users access to protected resources within the platform.

Session Persistence:

WanderPlan implements session persistence mechanisms to maintain user sessions across device restarts and app relaunches. By storing authentication tokens securely and managing session state effectively, the module ensures uninterrupted access to the platform, enhancing user convenience and usability.

Session Timeout:

To enhance security, the Session Management module implements session timeout mechanisms to automatically expire inactive user sessions after a specified period of inactivity. By invalidating authentication tokens and preventing unauthorized access, session timeouts protect user accounts and data from potential security threats.

Implementation

The implementation of the Session Management module involves integrating Firebase Authentication services with the Flutter application. Firebase SDKs and plugins facilitate token generation, validation, and session management, ensuring a seamless user experience. Flutter's reactive framework enables real-time updates and synchronization of user sessions, enhancing the usability and responsiveness of WanderPlan.

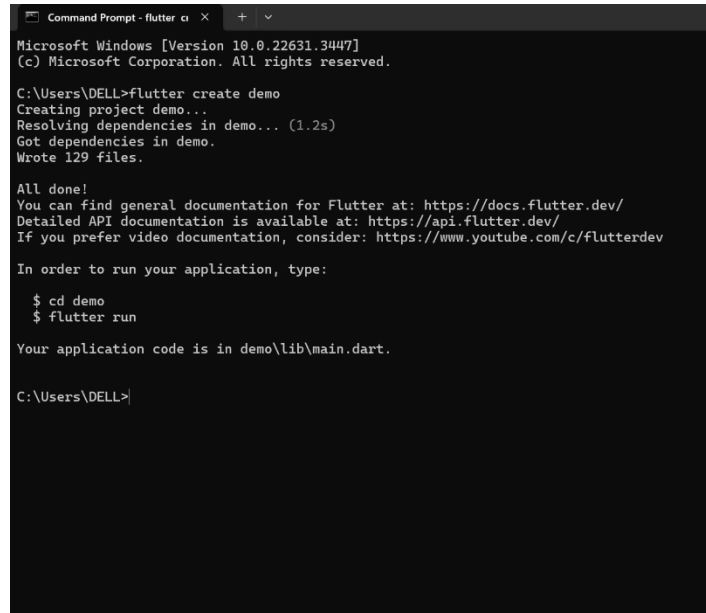
2.5 DEVELOPMENT AND DEPLOYMENT SETUP

1.Development Environment:

Flutter and Dart need to be installed on the developer's machine. IDEs like Visual Studio Code or Android Studio for coding.

2.Create Project:

>>flutter create project_name



```
Command Prompt - flutter ci X + v
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>flutter create demo
Creating project demo...
Resolving dependencies in demo... (1.2s)
Got dependencies in demo.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/Flutterdev

In order to run your application, type:

$ cd demo
$ flutter run

Your application code is in demo\lib\main.dart.

C:\Users\DELL>
```

Fig 2.3: Project Creation

The Fig 2.3 represents the command to create a new flutter project. A new flutter project named 'demo' is created in the stated figure.

3.Dependency Management:

Project dependencies specified in the pubspec.yaml file.

Key packages include:

- firebase_core for Firebase integration.
- firebase_auth for user authentication.
- cloud_firestore for real-time data synchronization.
- Other packages like get, toast, image_picker, etc., for additional functionalities.

4. Firebase Configuration:

Firebase project created and configured for the application. Firebase Hosting used for deploying the Flutter web application.

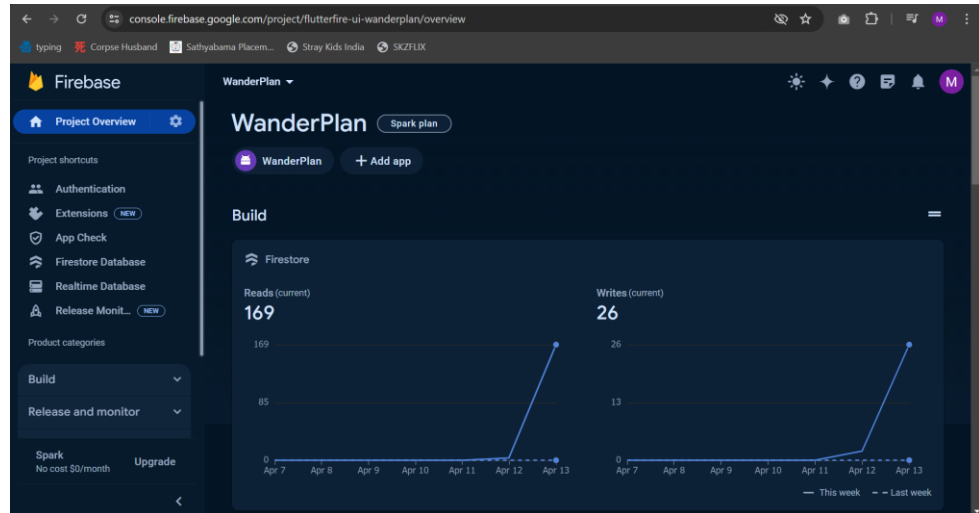


Fig. 2.4: Firebase Connection

The Fig 2.4 represents the connection between our Flutter app and Firebase Project which is named 'WanderPlan'

5. Firebase Authentication:

The Fig 2.5 represents the Firebase Authentication set up for secure user login and identity verification through the Users feature available.

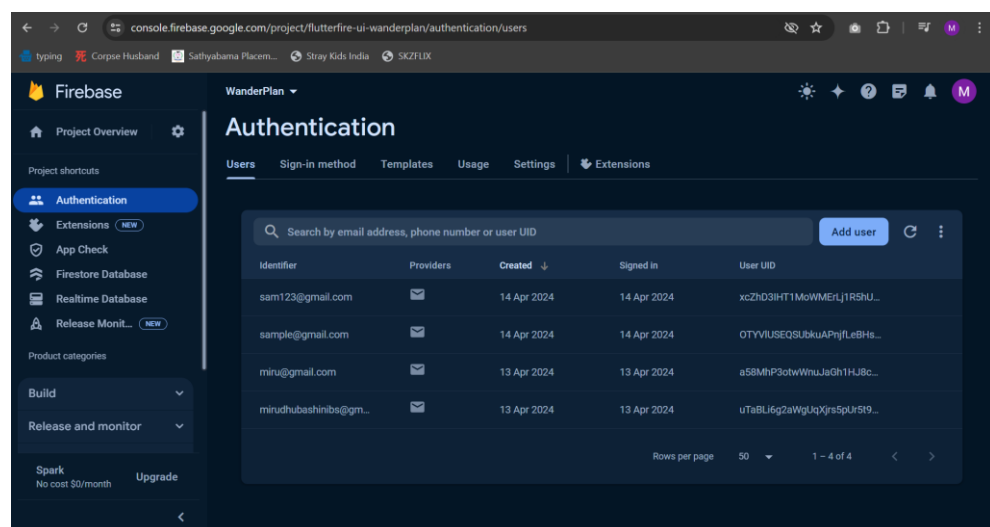
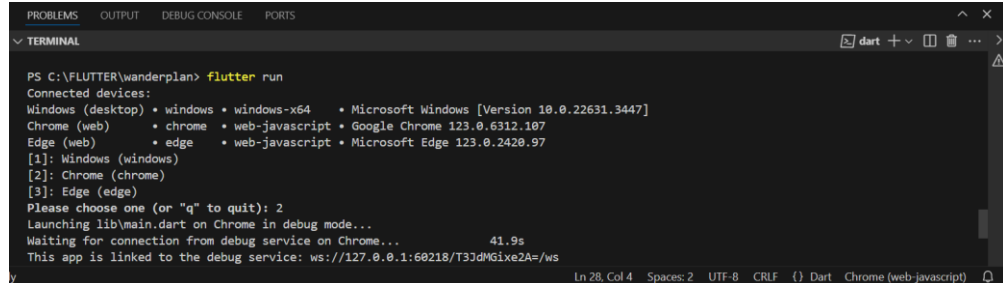


Fig.2.5: Firebase Authentication

6.Development Tools:

The Fig 2.6 represents the Flutter commands used for testing and building the application. Real-time testing on Android and iOS devices.



```
PS C:\FLUTTER\wanderplan> flutter run
Connected devices:
Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.22631.3447]
Chrome (web) • chrome • web-javascript • Google Chrome 123.0.6312.107
Edge (web) • edge • web-javascript • Microsoft Edge 123.0.2420.97
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (or "q" to quit): 2
Launching lib/main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...
This app is linked to the debug service: ws://127.0.0.1:60218/T3JdMGixe2A=/ws
```

Fig.2.6: Flutter Run

7.Deployment Process:

Firebase used for hosting the web version of the application. Seamless deployment using Flutter and Firebase commands.

8.Supported Platforms:

Cross-platform support for Android, iOS, and web.

9.Integrated Development Environment (IDE):

Developers can choose between Visual Studio Code or Android Studio for coding.

10.User Authentication:

Firebase Authentication ensures secure user login.

CHAPTER 3

RESULTS AND DISCUSSION

DART CODES

i)main.dart:

The Fig 3.1 represents file is typically the entry point of a Flutter application. It initializes the app, sets up necessary configurations, and runs the main application widget.

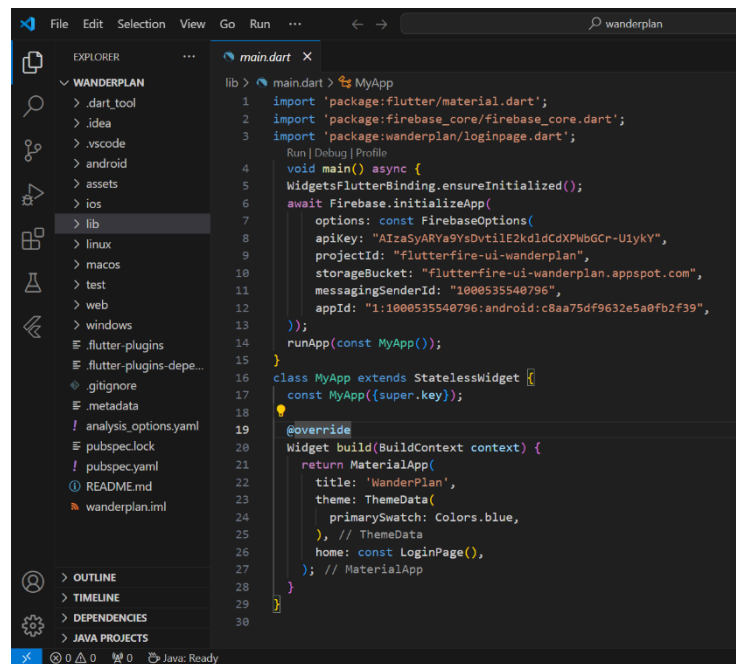
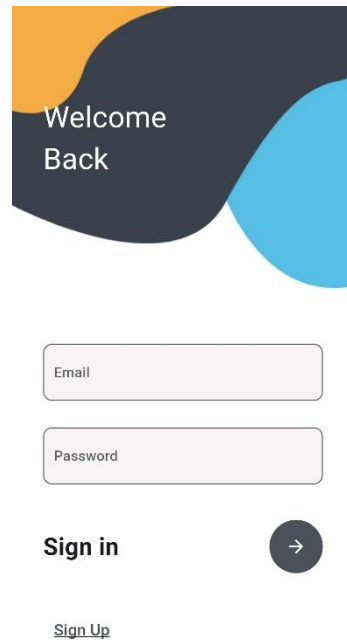


Fig.3.1: Main Dart

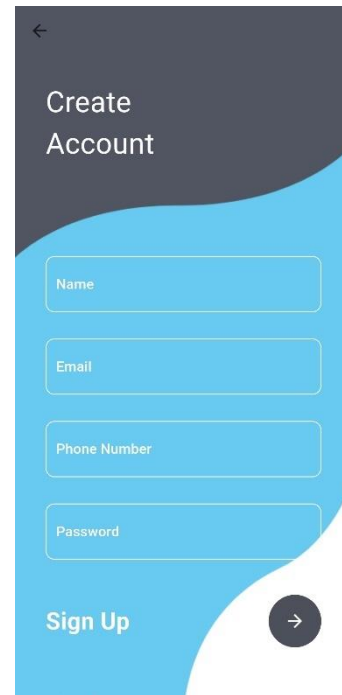
ii)loginPage.dart

The Fig 3.2 represents the UI and logic for user login. It includes form fields for entering credentials and a mechanism to authenticate users.



The login page features a dark blue header with a yellow and blue abstract shape on the left. The text "Welcome Back" is centered in white. Below the header, there are two light gray input fields labeled "Email" and "Password". At the bottom, there is a "Sign in" button with a right arrow icon and a "Sign Up" link.

Fig.3.2: Login Page



The signup page features a dark blue header with a blue abstract shape on the right. The text "Create Account" is centered in white. Below the header, there are four light blue input fields labeled "Name", "Email", "Phone Number", and "Password". At the bottom, there is a "Sign Up" button with a right arrow icon.

Fig.3.3: Signup Page

iii)signupPage.dart

The Fig 3.3 represents similar to the login page, signup page is responsible for handling user registration. It includes form fields for users to create accounts.

iv)dashboard.dart

The Fig 3.4 represents the main content of app after a successful login or signup. It contains various features.



Fig.3.4: Dashboard Page

v)profile.dart:

The Fig 3.5 represents the page responsible for displaying user profiles. It includes information about the user, such as their name, profile picture, and additional details

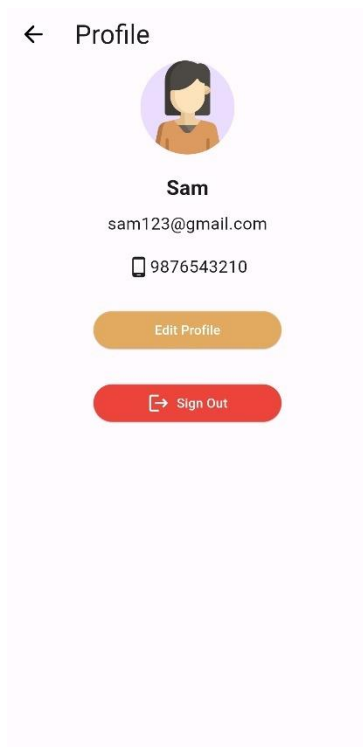


Fig.3.5: Profile Page

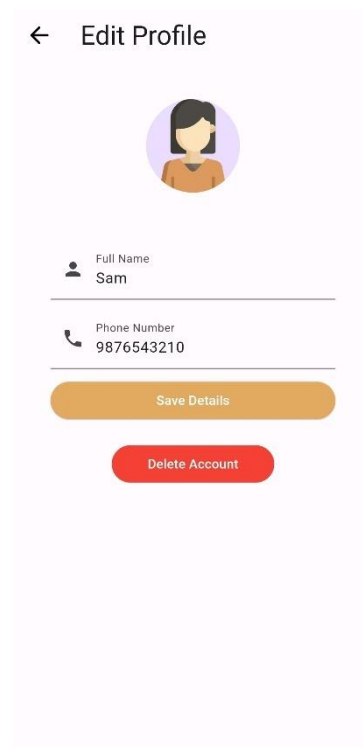


Fig.3.6: Profile Update Page

vi) profileupdate.dart:

The Fig 3.6 represents details of user to be updated and also provides the delete account option.

FIREBASE SETUP

i)Create a Firebase Project:

- Go to the Firebase Console.
- Click on "Add Project" and follow the setup instructions.

ii)Configure Flutter Project:

- Add the Firebase configuration files to your Flutter project.
- Follow the instructions provided in the Firebase Console to add googleservices.json for Android and GoogleService-Info.plist for iOS.

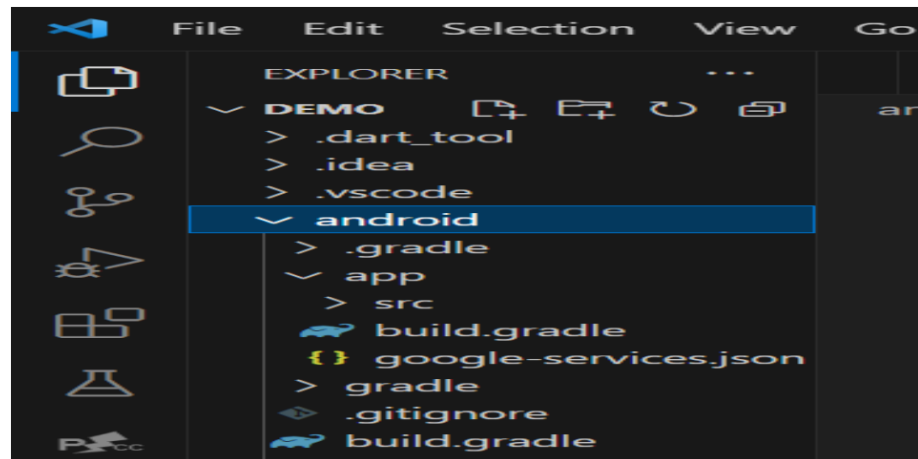


Fig.3.7: Firebase Configurations

The Fig 3.7 represents the code used to connect and configure the firebase database and our app

iii)Add Firebase Dependencies:

Open the pubspec.yaml file and add the following dependencies:

yaml code:

dependencies:

firebase_core: ^2.25.3

firebase_auth: ^4.17.5

iv)Initialize Firebase:

In your main.dart or wherever you initialize your Flutter app, make sure to initialize Firebase. If you want to implement user authentication, Firebase provides the firebase_auth package. You can use it to sign in users using various methods like email/password, Google Sign-In, etc.

CHAPTER 4

CONCLUSION AND FUTUREWORK

4.1 CONCLUSION

In conclusion, WanderPlan represents a transformative solution for modern travelers, offering a seamless blend of itinerary planning, social networking, and personalized recommendations. By streamlining the travel planning process and fostering connections among users, WanderPlan enriches the journey from start to finish.

WanderPlan empowers users to create tailored travel itineraries that reflect their unique interests, preferences, and schedules. With intuitive tools and real-time collaboration features, users can effortlessly coordinate plans with friends and family, ensuring smooth and enjoyable travel experiences.

WanderPlan's personalized recommendations provide users with curated suggestions for attractions, activities, and dining options, enhancing the exploration of new destinations. By leveraging user data and advanced algorithms, WanderPlan delivers relevant and engaging content that resonates with each user's individual tastes and preferences.

WanderPlan is more than just a travel planning tool—it is a gateway to personalized adventures, shared discoveries, and lasting memories. By combining innovation with user-centric design, WanderPlan transforms the way travelers explore the world, ensuring that every journey is filled with excitement, inspiration, and unforgettable moments.

4.2 FUTURE WORK

1.Enhanced User Profiles:

Implement additional fields and customization options in user profiles, allowing users to showcase more details.

2.Advanced Algorithm:

Continuously refining and expanding the algorithms responsible for personalized recommendations will ensure that WanderPlan remains at the forefront of delivering tailored travel suggestions. Incorporating machine learning and AI techniques can further enhance the accuracy and relevance of recommendations based on user preferences and behavior.

3. Expansion of Geographic Coverage

Expanding the geographic coverage of WanderPlan to include more destinations worldwide will cater to the needs of a broader user base. Collaborating with local tourism authorities and businesses can facilitate the integration of comprehensive information and services for a diverse range of destinations.

4.Enhanced Data Privacy and Security Measures

Continuous improvement of data privacy and security measures will remain a priority to ensure user trust and compliance with evolving regulations. Implementing robust encryption protocols, regular security audits, and user education initiatives can further enhance data protection.

5.Monetization Strategies

Exploring additional monetization strategies such as premium subscription tiers, sponsored content partnerships, and affiliate marketing programs can diversify revenue streams while ensuring continued value delivery to users.

REFERENCES

Academic Papers and Journals:

[1] C.-L. Ho, W.-L. Chen, and C.-H. Ou. (2023). Constructing a personalized travel itinerary recommender system with the internet of things. *Wireless Networks*, 321–330.

[2] P. Tiwari, R. P. Kaurav, and K. Y. Koay. (2023). Understanding Travel Apps Usage intention: Findings from PLS and NCA. *Journal of Marketing Analytics*, 12(1), 25–41.

[3] P. Y.-H. Sia, S. S. Saidin, and Y. H. Iskandar.(2022).Systematic review of Mobile Travel Apps and their smart features and challenges. *Journal of Hospitality and Tourism Insights*,6(5). 2115–2138.

Books:

[1] K. S. (2018).Mastering Firebase for Android Development: Build Real-Time, Scalable, and Cloud-Enabled Android Apps with Firebase. Birmingham, UK: Packt Publishing, Ltd.

[2] Miola. (2023). Flutter Complete Reference 2.0: The Ultimate Reference for Dart and Flutter. Independently published.

[3] J. H. Iversen and M. A. Eierman. (2014).Learning Mobile App Development: A Hands-on Guide to Building Apps with IOS and Android. Upper Saddle River, New Jersey: Addison-Wesley.

Internet Articles:

[1] D. Camp. (2024). Best flutter blog articles from around the web. Flutter articles. <https://www.devgla.com/programming/flutter-article-feeds>.

[2] Google, Firebase documentation. (2024). Flutter Documentations. <https://firebase.google.com/docs>.

[3] W. Wilson. 2024). The ultimate list of flutter resources (over 200!). Flutter Resources. <https://wilsonwilson.dev/articles/flutter-resources>.

APPENDICES

A. SAMPLE CODE

main.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:wanderplan/loginpage.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: const FirebaseOptions(
      apiKey: "AIzaSyARYa9YsDvtilE2kdldCdXPWbGCr-U1ykY",
      projectId: "flutterfire-ui-wanderplan",
      storageBucket: "flutterfire-ui-wanderplan.appspot.com",
      messagingSenderId: "1000535540796",
      appId:
"1:1000535540796:android:c8aa75df9632e5a0fb2f39",
    ));
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'WanderPlan',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const LoginPage(),
    );
  }
}
```

pubspec.yaml

```
name: wanderplan
description: "A new Flutter project."
publish_to: 'none'
version: 1.0.0+1

environment:
  sdk: '>=3.2.3 <4.0.0'

dependencies:
  flutter:
    sdk: flutter

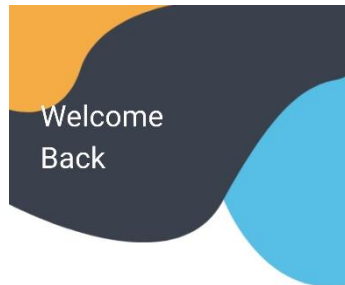
  cupertino_icons: ^1.0.2
  firebase_auth: ^4.17.3
  firebase_core: ^2.25.3
  cloud_firestore: ^4.15.3
  fluttertoast: ^8.2.4
  get: ^4.6.6
  firebase_database: ^10.4.9

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.0

flutter:
  uses-material-design: true

  assets:
    - assets/
```

B. SCREENSHOTS

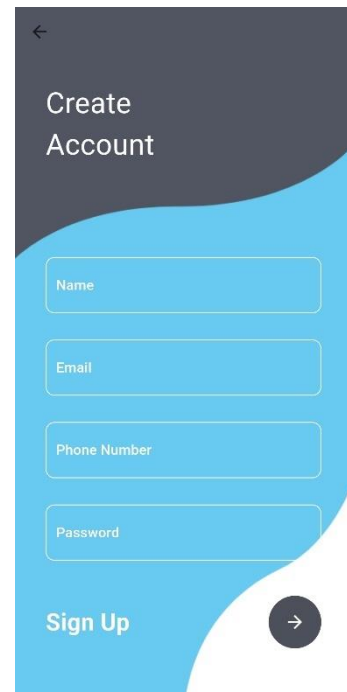


Sign in



[Sign Up](#)

SIGN IN PAGE

A mobile app screen with a dark blue background and a light blue wave-like shape at the bottom. The text "Create Account" is displayed in white. Below the title are four input fields: "Name", "Email", "Phone Number", and "Password". At the bottom, there is a "Sign Up" button and a right arrow icon.

←

Create Account

Name

Email

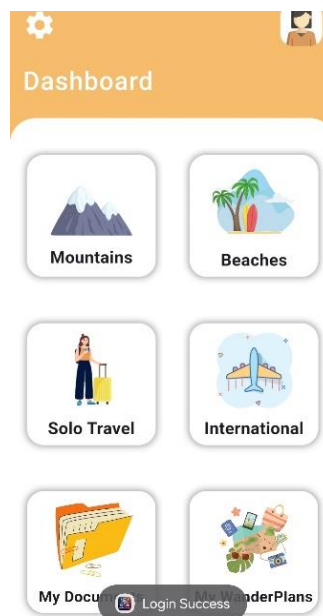
Phone Number

Password

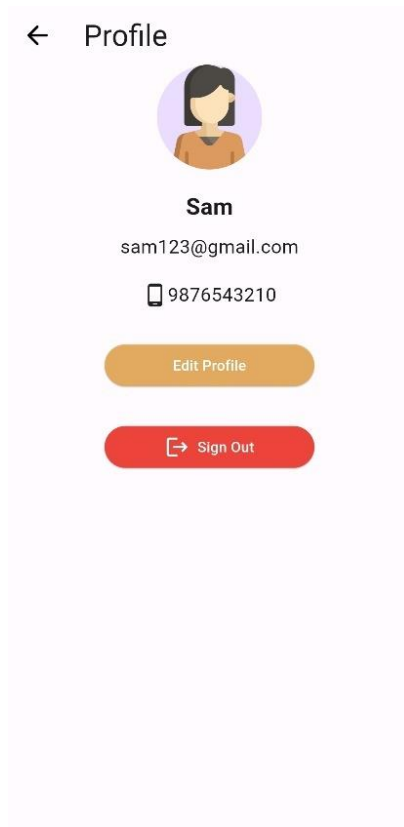
Sign Up

→

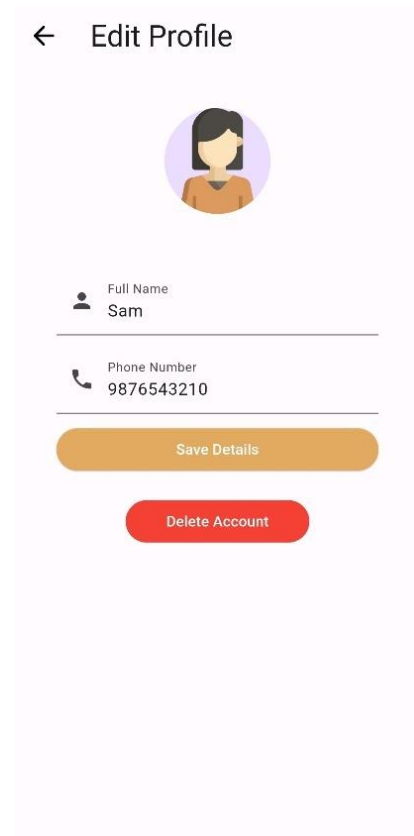
SIGN UP PAGE



DASHBOARD



PROFILE PAGE



EDIT PROFILE