

A Software requirements specification
On

Himalayan Trekking Application

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in
Computer Science and Engineering from Rajasthan Technical University, Kota



MENTOR:

Mrs.Archika Jain

(Dept. of Computer Science & Engineering)

COORDINATOR:

Mr.Sumit Mathur

(Dept. of Computer Science & Engineering)

SUBMITTED BY:

Tushar Singhal (21ESKCS213)

Mihir Jain (21ESKCS206)

Siddharth Goyal (21ESKCS212)

Anurag Sharma (21ESKCS202)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**SWAMI KESHVANAND INSTITUTE OF TECHNOLOGY,
MANAGEMENT & GRAMOTHAN
Ramnagar (Jagatpura), Jaipur – 302017**

SESSION 2023-24

Table of Contents

1. Introduction	3
1.1. Purpose	3
1.2. Motivation	4
2. Literature Survey	5
2.1. Objectives	6
3. Requirements	7
3.1. Functional Requirements	7
3.2. Non-Functional Requirements	8
3.3.1 Safety Requirements	8
3.3.2 Security Requirements	8
3.3.3 Software Quality Attributes	9
3.3. Hardware Requirements	9
3.4. Software Requirements	9
3.5. Waterfall Model	10
4. System Architecture	11
4.1. Client-Server Architecture	12
5. Design and Implementation	14
5.1. Product Features	15
5.2. Data Flow Diagram	16
5.3. Firebase and Android Studio Integration	17
5.4. Screenshots of Projects	19
6. Conclusion & Future Scope	22

1. Introduction

The Trek Mapping and Tourist Information Application is an innovative software solution designed to cater to the rapidly growing popularity of trekking and adventure tourism across various mountain ranges, including the Indian Himalayas, Western Ghats, Eastern Ghats, Garwahl, Kumaon, Vindhya, Satpura, and Aravalli ranges. Trekking has emerged as a favored form of vacationing, attracting travelers seeking unique and thrilling experiences. To address the evolving needs of trekkers and adventure enthusiasts, this application aims to provide a comprehensive platform for planning and navigating trekking routes, offering critical information such as trail difficulty, length, notable attractions, and the most suitable seasons for trekking. By integrating real-time updates on weather conditions, nearby accommodation options, banking services, medical facilities, eateries, and more, the application seeks to enhance safety and convenience, enabling users to make informed decisions during their trekking endeavors. This introduction sets the stage for an in-depth exploration of how the application fulfills these objectives, ultimately enhancing the trekking experience for adventure seekers.

1.1 Purpose

The purpose of developing the "Trek Mapping and Tourist Information Application" is to provide trekkers and adventure enthusiasts with a comprehensive tool for planning, navigating, and enhancing their trekking adventures across diverse mountain ranges, including the Indian Himalayas, Western Ghats, Eastern Ghats, Garwahl, Kumaon, Vindhya, Satpura, and Aravalli ranges. The application aims to facilitate efficient route exploration and planning, offering detailed trekking information such as difficulty levels, length, attractions, and recommended trekking seasons. It also focuses on safety and convenience by integrating real-time weather updates, nearby accommodation options, banking services, medical facilities, and eateries. By improving user experience and usability, the application encourages adventure tourism, fosters a connected trekking community, and contributes to environmental awareness by promoting responsible trekking and sustainable tourism practices. Ultimately, the application seeks to align with the

burgeoning popularity of trekking tourism, elevating the overall trekking experience for enthusiasts.

1.2. Need/Motivation

The motivation behind developing the Trek Mapping and Tourist Information Application stems from the growing allure of trekking and adventure tourism in diverse mountain ranges. With an increasing number of individuals seeking unique and immersive travel experiences, adventure tourism, particularly trekking, has garnered substantial interest. However, the experience of trekking can be greatly enhanced with the aid of technology, making it safer, more enjoyable, and accessible to a wider audience. The application aims to tap into this trend and provide a technological solution that caters to the specific needs of adventure enthusiasts. By offering a centralized platform that simplifies route planning, provides critical trekking details, and ensures safety through real-time updates and local information, the application seeks to empower trekkers, making their adventures more informed, exciting, and memorable. Additionally, the application strives to foster a sense of community among adventure seekers, encouraging responsible trekking practices and promoting environmental awareness, thus contributing positively to the adventure tourism ecosystem. Ultimately, the motivation is to create a tool that not only facilitates exploration but also nurtures a passion for adventure while upholding principles of sustainability and responsible tourism.

2. Literature Survey

2.1 Objectives

The objective of the Trek Mapping and Tourist Information Application is to develop a comprehensive digital solution tailored to the needs of trekkers and adventure enthusiasts exploring various mountain ranges. The project aims to achieve the following key objectives:

1. Enhanced Trekking Experience:

- Develop an intuitive application that enhances the overall trekking experience for users by providing essential trekking information, including trail details, difficulty levels, and attractions.

2. Real-time Information Access:

- Enable users to access real-time information on weather, accommodation, banking facilities, medical services, eateries, and more, ensuring informed and safe trekking experiences.

3. User-Friendly Interface:

- Design a user-friendly interface that allows trekkers, irrespective of their technical proficiency, to easily navigate the application, plan their treks, and access necessary information.

4. Community Building and Interaction:

- Facilitate a sense of community among trekkers by allowing them to share their experiences, reviews, and recommendations, fostering a supportive network of adventure enthusiasts.

5. Promotion of Responsible Trekking:

- Promote responsible trekking practices and environmental awareness by incorporating features that educate users on sustainable trekking, ethical travel, and eco-friendly behavior.

6. Multi-Range Coverage:

- Cover a wide range of mountain terrains, including the Indian Himalayas, Western Ghats, Eastern Ghats, Garwahl, Kumaon, Vindhya, Satpura, and Aravalli ranges, to cater to a diverse group of trekkers.

7. Accessibility and Inclusivity:

- Ensure the application is accessible to a global audience by providing multi-language support and integrating features that cater to different cultural and geographical preferences.

8. Data Accuracy and Reliability:

- Ensure that the information provided in the application, such as trekking routes, accommodation details, and other services, is accurate, reliable, and regularly updated to maintain user trust and satisfaction.

These objectives guide the development and implementation of the Trek Mapping and Tourist Information Application, aligning with the growing interest in trekking tourism and the need for a digital platform that enhances the trekking experience for adventure enthusiasts.

3. Requirements

3.1 Functional Requirements

Functional requirements define specific functionalities the Trek Mapping and Tourist Information Application must offer to meet its objectives. These requirements drive the development and design of the application.

1. User Authentication and Profile Management:

- Users should be able to create accounts and log in securely.
- Users should be able to update and manage their profiles.

2. Trekking Routes Display:

- Display a list of available trekking routes based on selected mountain ranges.
- Show essential information for each trekking route (difficulty level, length, key attractions, etc.).

3. Accommodation Information:

- Provide a list of nearby accommodation options for each trekking route.
- Display details of each accommodation option (contact information, facilities, etc.).

4. Banking and ATMs Information:

- Offer information about nearby banks and ATMs along trekking routes.

5. Medical Facilities Information:

- Present information about nearby hospitals or medical facilities along trekking routes.

6. Eateries Information:

- Display details about nearby eateries or restaurants along trekking routes.

7. Real-time Weather Updates:

- Integrate with a weather API to provide real-time weather updates for trekking locations.

8. Search and Filtering:

- Allow users to search for specific trekking routes based on various criteria.
- Enable users to filter trekking routes based on difficulty level, length, and more.

9. Notifications and Alerts:

- Send notifications to users regarding route updates, weather changes, or important information.

10. Feedback and Reviews:

- Allow users to provide feedback and reviews for trekking routes and services.

3.2 Non-Functional Requirements

Non-functional requirements define the quality attributes and constraints that the application should adhere to, influencing its overall performance, security, and usability.

3.3.1 Safety Requirements

Safety requirements ensure that the application minimizes risks and ensures the well-being of users during their trekking experience.

- The application should emphasize safety by providing information about trek difficulty levels and recommending appropriate routes based on user capabilities.
- The application should advise users on safety measures and precautions for specific trekking routes.

3.3.2 Security Requirements

Security requirements are crucial to protect user data and ensure secure interactions within the application.

- Users' login and personal information should be encrypted and stored securely.

- Access to sensitive user data should be restricted to authorized personnel only.

3.3.3 Software Quality Attributes

Software quality attributes define the performance, reliability, and usability expectations for the application.

- **Usability:**
 - The application should have an intuitive user interface, making it easy for users to navigate and access information.
- **Performance:**
 - The application should respond promptly to user interactions, ensuring a smooth and efficient user experience.
- **Reliability:**
 - The application should be available and functional at all times, minimizing downtime.
- **Scalability:**
 - The application should be designed to handle a growing number of users and data without compromising performance.

3.3 Hardware Requirements

The hardware requirements for the Trek Mapping and Tourist Information Application include:

- A device (e.g., smartphone, tablet, computer) with internet connectivity and GPS capabilities.

3.4 Software Requirements

The software requirements for the application encompass:

- An operating system compatible with the application (iOS, Android, Web browsers).
- Modern web browsers for web access (Chrome, Firefox, Safari).
- Integrated development environment for software development and testing.

3.5 Waterfall Model

The development of the Trek Mapping and Tourist Information Application will follow the Waterfall Model, a linear and sequential approach that consists of distinct phases: Requirements, Design, Implementation, Testing, Deployment, and Maintenance. Each phase must be completed before moving on to the next, ensuring a structured and well-planned development process.

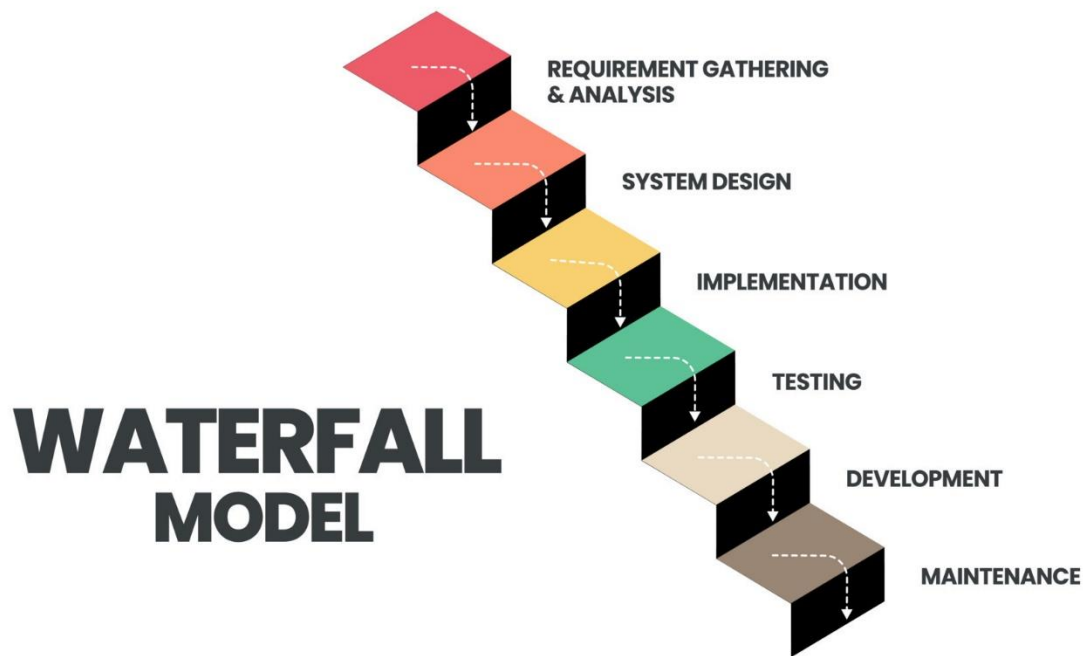


Fig 1:- Waterfall Model

The defined requirements, both functional and non-functional, lay the foundation for the development of the Trek Mapping and Tourist Information Application, providing clear guidelines for functionality, safety, security, and overall quality of the system. The choice of the Waterfall Model sets the development approach, emphasizing a systematic and well-structured progression through each development phase.

4. System Architecture

The system architecture for the Trek Mapping and Tourist Information Application is designed to provide a robust and efficient framework that meets the project's objectives of enhancing the trekking experience and providing comprehensive information to users. The architecture follows a scalable and modular approach, ensuring ease of maintenance and future expansions.

4.1 Client-Server Architecture

Client-Server Architecture is a fundamental design pattern used in networking and distributed systems, forming the basis for various applications and services. This architectural model divides the system into two main components: the client, which initiates requests, and the server, which processes these requests and provides the necessary responses. This approach facilitates efficient communication and resource sharing between clients and servers.

Key Components

1. Client:

- User Interface (UI): The front-end component that interacts with users, providing a graphical interface for users to access and utilize the application.
- Application Logic: Handles the application's logic and user interactions, managing requests and responses.
- User Authentication: Responsible for authenticating users, ensuring secure access to the system.

2. Server:

- Application Server: Contains the application's core logic, processing requests from clients, applying business logic, and interacting with the database.

- **Database Server:** Stores and manages data related to trekking routes, user profiles, accommodation, and other system information.
- **External Services Integration:** Interfaces with external services such as weather APIs, accommodation booking services, and mapping APIs to fetch real-time data.

Communication Flow:

1. The client sends a request to the application server through the network, initiating the desired action (e.g., requesting trekking routes for a specific mountain range).
2. The application server processes the request, interacts with the database server to fetch required data (e.g., trekking route information), and applies the necessary business logic.
3. The application server then sends the processed data as a response back to the client.
4. The client's user interface displays the received data to the user, providing the requested information in a readable and intuitive format.

Advantages:

- **Scalability:** Client-server architecture allows for easy scaling by adding more clients or servers, enhancing system performance as demand grows.
- **Centralized Data Management:** Data is stored and managed centrally on the server, ensuring consistency and security.
- **Efficient Resource Utilization:** The server handles complex tasks and computations, relieving clients of processing burdens and ensuring efficient resource utilization.

Considerations:

- **Security:** Implementing robust security measures to protect data during transmission and ensuring authorized access to the system.
- **Load Balancing:** Employing load balancing techniques to distribute incoming client requests evenly across multiple servers, optimizing performance and availability.

Client-Server Architecture is a widely adopted model due to its scalability, efficient resource management, and clear separation of concerns between the client and server.

components. In the context of the Trek Mapping and Tourist Information Application, this architecture will enable seamless interaction between trekkers (clients) and the application's functionality (server), providing a reliable and efficient platform for trekking enthusiasts.

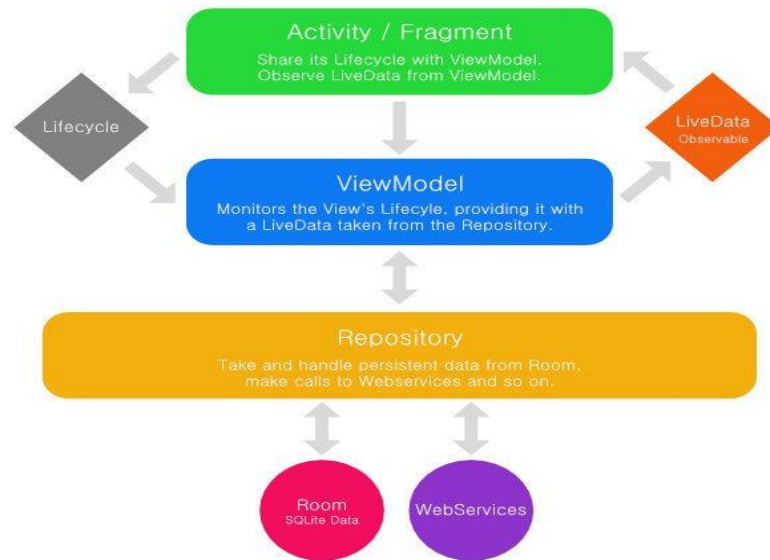


Fig 2:- Client-Server Architecture

5. Design and Implementation

In the design and implementation phase of the Trek Mapping and Tourist Information Application, we delve into the critical aspects of the system's architecture, product features, data flow diagram, class diagram, use case diagram, sequence diagram, and E-R (Entity-Relationship) diagram.

5.1 Product Features

The product features are the fundamental functionalities that the application offers to its users. These features guide the development process, ensuring that the application meets the intended objectives and requirements outlined in the Software Requirements Specification (SRS) report.

- **User Authentication and Profile Management:**

Allow users to create accounts, log in, and manage their profiles.

- **Trekking Routes Display:**

Display trekking routes for various mountain ranges with essential information.

- **Detailed Route Information:**

Provide detailed information about each trekking route, including difficulty level, length, and key attractions.

- **Accommodation Information:**

Display nearby accommodation options with contact information and facilities available.

- **Banking and ATMs Information:**

Provide information on nearby banks and ATMs along trekking routes.

- **Medical Facilities Information:**

Display nearby hospitals or medical facilities with contact information.

- **Eateries Information:**

Provide information about eateries or restaurants available along trekking routes.

- **Real-time Weather Updates:**

Integrate with a weather API to provide real-time weather updates for trekking locations.

- **Search and Filtering:**

Allow users to search for specific trekking routes and apply filters based on preferences.

- **Notifications and Alerts:**

Send notifications and alerts to users for important updates and reminders related to their treks.

- **Feedback and Reviews:**

Enable users to provide feedback and reviews for trekking routes and other services.

5.2 Data flow diagram:

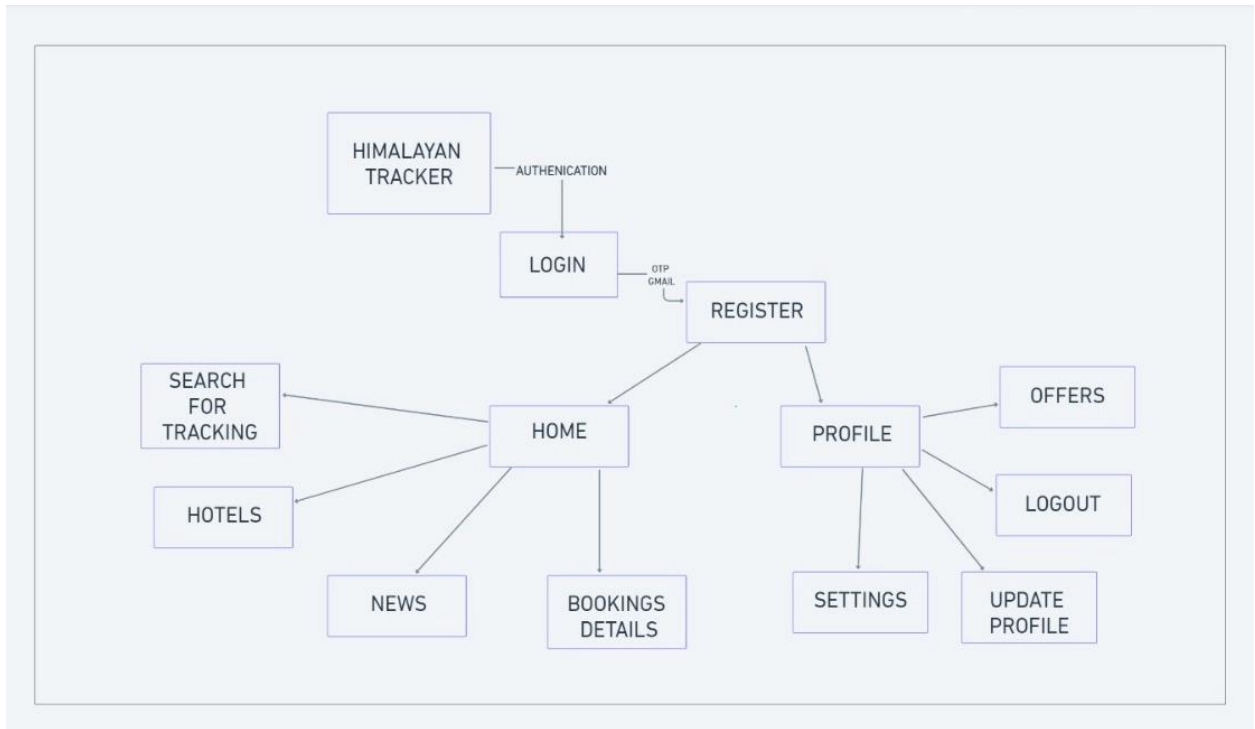


Fig 3:- Data flow diagram

5.3 Firebase and Android Studio Integration:

Firebase and Android Studio integration is a powerful combination that allows developers to build feature-rich, real-time, and scalable Android applications. Firebase is a comprehensive mobile development platform provided by Google, offering various tools and services to simplify app development and improve user engagement.

Key Integration Points:

1. Firebase Authentication:

Integrate Firebase Authentication to enable secure user authentication in the Android application. Users can sign in using email, phone number, Google, or other supported providers.

2. Firebase Realtime Database:

Utilize Firebase Realtime Database to store and synchronize data in real time across connected devices. This enables live updates and seamless collaboration within the application.

3. Firebase Cloud Messaging (FCM):

Implement FCM to enable push notifications in the Android app, allowing for personalized messaging and engagement with users.

4. Firebase Storage:

Use Firebase Storage to efficiently store and serve user-generated content such as images, videos, and other files associated with the application.

5. Firebase Crashlytics

Integrate Firebase Crashlytics for real-time crash reporting and analysis, enabling developers to identify and fix application issues promptly.

6. Firebase Analytics:

Implement Firebase Analytics to gain insights into user behavior, measure app performance, and optimize the application based on analytics data.

Integration Steps:

1. Setting up a Firebase Project:

Create a Firebase project on the Firebase Console (<https://console.firebase.google.com/>).
Add the Android app to the project by providing the app's package name.

2.Downloading Configuration Files:

Download the google-services.json configuration file for the Android app and place it in the app module.

3.Adding Firebase SDK to Android Project:

Update the app-level build.gradle file to include the Firebase SDK dependencies.

Sync the project to fetch the necessary dependencies.

4. Initializing Firebase:

Initialize Firebase in the Android app by adding the initialization code in the application class or the main activity.

5. Using Firebase Services:

Integrate Firebase services like Authentication, Realtime Database, Cloud Messaging, Storage, etc., based on the application requirements.

6. Testing:

Test the Firebase integration by using Firebase features within the Android app and ensuring proper functionality.

7. Monitoring and Optimization

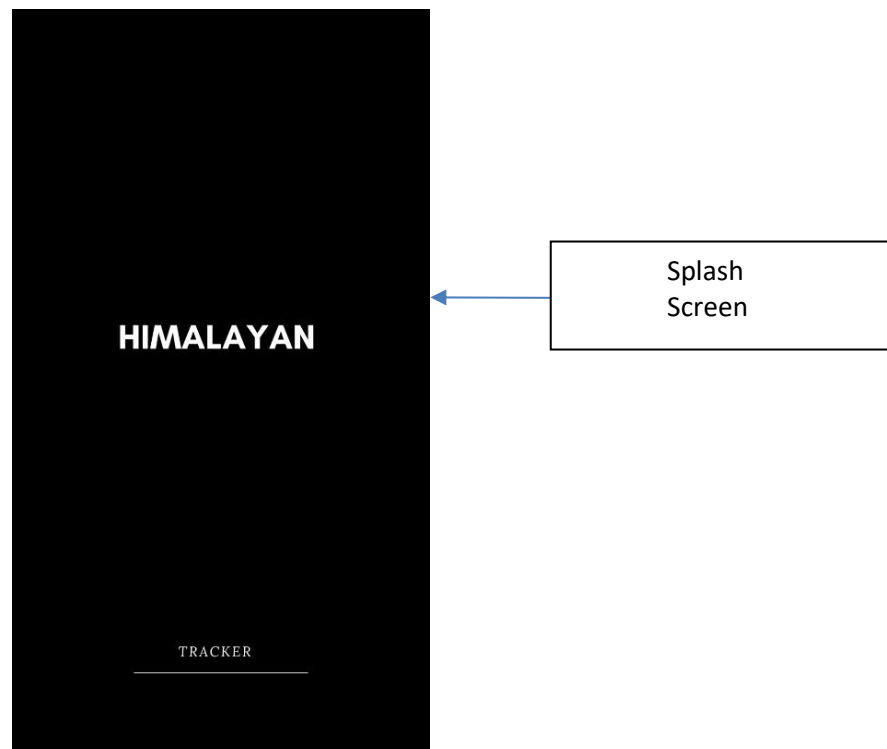
Monitor Firebase Analytics, Crashlytics, and other Firebase services to gain insights into app performance and user engagement. Optimize the app based on analytics data and user feedback.


Benefits of Firebase and Android Studio Integration:

- **Rapid Development:** Firebase provides ready-to-use features, reducing development time.
- **Real-time Updates:** Firebase Realtime Database enables real-time synchronization of data, enhancing the user experience.
- **Improved User Engagement:** Firebase Cloud Messaging allows personalized notifications, keeping users engaged.
- **Easy Monitoring:** Firebase Analytics and Crashlytics provide valuable insights for monitoring and optimization.

Integrating Firebase with Android Studio significantly enhances the development process, providing a wide range of tools and services to build robust, engaging, and scalable Android applications.

5.4 Screenshots of project:






Login with
Mobile number

+91

SEND OTP


 **SIGN IN WITH GOOGLE**

REGISTER AS A DRIVER

Login Activity

Register
Activity

Register Yourself

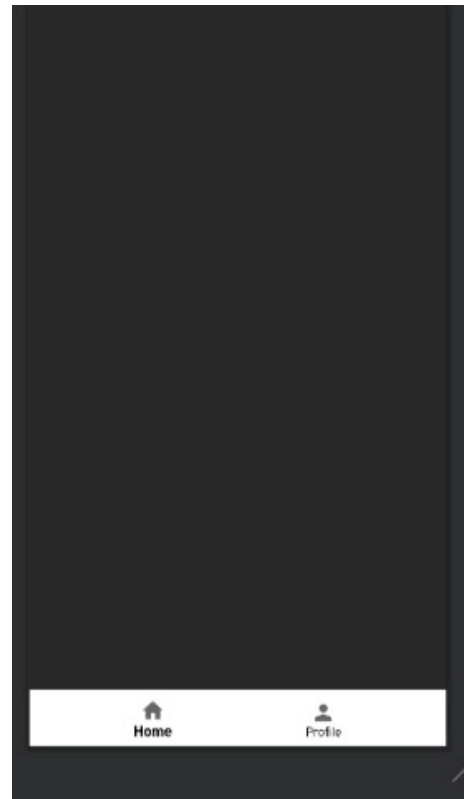


CONTINUE



OTP Authentication

Dashboard(Still Working)



6 Conclusion & Future Scope

Conclusion:

In conclusion, the Trek Mapping and Tourist Information Application is a vital tool that caters to the evolving needs of trekkers and adventure enthusiasts. With the ever-increasing popularity of adventure tourism, especially trekking across diverse mountain ranges, this application addresses the demand for a comprehensive platform that enhances the trekking experience. By integrating essential features such as real-time weather updates, nearby accommodation options, medical facilities, eateries, and detailed trekking route information, the application ensures safety and convenience for users. It provides a user-friendly interface, aiding both experienced and novice trekkers in planning and navigating trekking routes effectively. Furthermore, the potential for fostering a community of adventure enthusiasts and promoting sustainable trekking practices underlines the positive impact this application can have on the adventure tourism landscape.

Future Scope:

Looking ahead, the Trek Mapping and Tourist Information Application has vast potential for growth and enhancement. Some of the key areas for future development and expansion include:

1. Advanced Features and Functionality:

Integrate augmented reality (AR) for a more interactive and immersive trekking experience, allowing users to visualize routes and attractions.

2. Social Integration:

Incorporate social networking elements, enabling trekkers to share experiences, photos, and tips, fostering a sense of community among adventure seekers.

3. Machine Learning and Personalization:

Implement machine learning algorithms to personalize recommendations for routes, accommodation, and services based on user preferences and past interactions.

4. Integration with Wearable Devices:

Enable compatibility with wearable devices to provide real-time health and fitness monitoring, enhancing safety and overall well-being during treks.

5. Global Expansion:

Extend the application to cover trekking destinations beyond India, catering to a broader international audience and encompassing popular trekking spots worldwide.

6. Partnerships and Collaborations:

Collaborate with tourism boards, local businesses, and environmental organizations to provide users with exclusive offers, promote responsible tourism, and contribute to environmental conservation efforts.

7. Multi-Language Support:

Introduce multi-language support to make the application accessible to a global audience, ensuring language is not a barrier for international trekkers.

The future scope of the Trek Mapping and Tourist Information Application involves continued innovation and strategic partnerships, with a focus on providing an enhanced, personalized, and engaging experience for adventure enthusiasts, further cementing its position as an indispensable tool in the adventure tourism domain.