BSIT-S25-003

03-135221-031 Usama Rafique

03-135221-035 Kamran Ishfaq

03-135221-005 Ali Saad Sandhu

Groundify

Indoor Ground Booking App

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Information Technology

Supervisor: Hoor Fatima Yousaf

Department of Computer Science

Bahria University, Lahore Campus

January 2026

Certificate

We accept the work contained in the report titled

"Groundify (Indoor cricket ground app)"

written by

Usama Rafique

Kamran Ishfaq

Ali Saad Sandhu

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Information Technology.

Approved by:

Supervisor: Hoor Fatima Yousaf

_____ (Signature)

January 05, 2026

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-135221-031	Usama Rafique	
03-135221-035	Kamran Ishfaq	
03-135221-035	Ali Saad Sandhu	

Date: January 05, 2026

Specially dedicated to

My beloved father, mother and grandparents

(Usama Rafique)

My beloved grandmother, mother and father

(Kamran Ishfaq)

My beloved grandmother, mother and father

(Ali Saad Sandhu

ACKNOWLEDGEMENTS

We want to express our gratitude to everyone who helped us finish this project successfully. We would like to thanks Hoor Fatima Yousaf, our supervisor, for all of his help and patience over the course of the research.

We also like to thank our devoted parents and friends for their support and encouragement.

Usama Rafique
Kamran Ishfaq
Ali Saad Sandhu

Groundify (Indoor booking system)

ABSTRACT

Groundify is a digital platform designed to streamline the process of discovering and booking indoor sports facilities for athletes, teams, and recreational players. By providing a user-friendly interface, multiple sports categories, real-time slot availability, and integrated management tools, Groundify enables users to efficiently book courts for Cricket, Football, Volleyball, Basketball, Hockey, and other indoor games. The system ensures seamless communication between users and facility managers, promoting a convenient and hassle-free booking experience. Built using the Flutter framework with Firebase as the backend, Groundify delivers fast performance, secure authentication, cloud-based data storage, and instant updates. In summary, Groundify modernizes the way individuals and sports communities manage indoor ground bookings by offering a reliable, dynamic, and fully digital solution tailored to the evolving demands of sports facility management.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

API:	Application Programming Interface
UI:	User Interface
UX:	User Experience
OS:	Operating System

CHAPTER 1

INTRODUCTION

1.1 Background

Cricket arenas, futsal courts, volleyball halls, basketball gyms, and multipurpose sports complexes are examples of indoor sports facilities that are becoming more and more essential to contemporary urban life. Individuals and teams often look to reserve indoor spaces for training sessions, weekend games, tournaments, or casual games due to the growing interest in fitness, organized sports leagues, and leisure activities. In many cities, including Pakistan, the reservation process is still manual, disorganized, and ineffective despite strong demand [2].

To verify availability, a user usually has to give the establishment a call, send several WhatsApp messages, or go in person. This procedure causes a number of problems:

1. Two reservations because of antiquated manual registers
2. Inadequate communication between facility managers and users
3. Prolonged wait times for availability confirmation
4. No digital payment or reservation history
5. Having trouble changing or cancelling reservations
6. No common method for comparing prices
7. The availability of slots is opaque

Facility owners find it difficult to manage payments, handle customer inquiries, keep track of daily schedules, and maintain real-time availability. In the end, these problems lower facility utilization and customer satisfaction.

Technology-based systems, especially mobile applications, have emerged as a workable solution to this expanding issue. Developers can now easily create dynamic, scalable, and real-time booking systems thanks to cross-platform frameworks like Flutter and backend services like Firebase. While Firebase offers powerful features like Cloud Storage, Notifications, Cloud Functions, Firestore Database, and Authentication, Flutter offers a single codebase for both iOS and Android [1].

By providing a centralized, automated, and user-friendly mobile platform, the Indoor Booking System (IBS) overcomes all the drawbacks of conventional manual booking. This system allows users to view booking history, make payments, book slots in real-time, check sports availability, and receive notifications. From a linked Owner interface, facility managers can approve or reject reservations, control schedules, upload photos, monitor income, and examine usage.

1.2 Problem Statements

The majority of booking and management procedures are still done by hand, despite the rising demand for indoor sports facilities. This presents difficulties for owners as well as users: [3]

User-Side Problems

The major User-Side side Problems are show below.

1. The challenge of verifying the availability of several sports facilities in real time
2. To make a reservation, you must manually call or text the facility.
3. No centralized platform for comparing costs, schedules, and amenities
4. Uncertainty due to delayed confirmation
5. No history of reservations or payment receipts
6. Restricted or non-existent refund or cancellation procedures
7. Miscommunication between facility owners and users

Owner-Side Problems

The major Owner-Side side Problems are show below.

1. Managing multiple bookings manually leads to errors
2. Hard to update availability and schedules in real-time
3. No secure method to record payment and customer data
4. Lack of analytics regarding usage and revenue
5. Inefficient communication channels with customers
6. No automated notification or alert system
7. Unable to effectively market their available services

Core Problem

There is no **integrated**, **real-time**, **cross-platform**, and **automated system** for booking indoor grounds that supports multiple sports, user-Owner interaction, and secure cloud-based data storage [2].

The Indoor Booking System aims to directly solve these challenges using a professional Flutter + Firebase architecture.

1.3 Aims and Objectives

Aim

To develop a cross-platform Indoor Booking System using **Flutter** as the frontend and **Firebase** as the backend, enabling real-time booking, slot management, user-Owner interaction, and digital record-keeping for multiple indoor sports [2].

Objectives

Technical Objectives

The Technical Objectives of my Project are shown below.

1. Develop a mobile application using Flutter for both Android and iOS.
2. Implement Firebase Authentication for secure user and Owner login.
3. Create a real-time Firestore database for facility listings, slots, and bookings.
4. Integrate Firebase Storage for images and sports facility media.
5. Use Firebase Cloud Messaging to send real-time booking and update notifications [10].
6. Implement role-based access control (User vs. Owner workflows).
7. Provide an intuitive UI/UX for browsing and reserving indoor sports facilities.
8. Develop an Owner panel for managing schedules, bookings, and facility information.

Functional Objectives

The Functional Objectives of my Project are shown below.

1. Allow users to search sports facilities by sport type, location, time, and price.
2. Display real-time availability and prevent double-booking using Firestore transactions [5].
3. Enable booking creation, cancellation, and modification.
4. Allow Owners to approve or reject booking requests.
5. Generate booking receipts and history for users.

Provide daily, weekly, and monthly analytics for Owners.

Extended Objectives

The Expanded Objectives of my Project are shown below.

1. Introduce optional payment gateway integration.
2. Enable multi-sport support including cricket, football, volleyball, basketball, and hockey.
3. Create an automated refund or reschedule feature.

1.4 Scope of Project

In-Scope Functionalities

The In-Scope Functionalities of my Project are shown below.

1. Cross-platform (Android/iOS) mobile application.
2. User authentication (Email, Phone, Google Sign-In).
3. Facility listing with images, details, policies, and pricing.
4. Real-time slot availability and schedule overview.
5. Booking system with confirmation, cancellation, and receipts
6. Owner dashboard integrated in the same app
7. Firebase-based notifications for status updates
8. Search and filter system by sport type, time, and price
9. Feedback and rating system

Technology Scope

The Technology Scope of my Project are shown below.

1. Flutter for frontend
2. Firebase for backend
3. Firestore Database
4. Firebase Authentication
5. Firebase Storage
6. Cloud Functions for automation

Out-of-Scope Features

The Out-Scope Functionalities of my Project are shown below.

1. Loyalty points
2. Multi-currency support
3. AI-based Chat bot
4. Offline-only booking support

Integration with third-party sports services.

Significance of the Project

User Benefits

The User benefits of my Project are shown below.

1. Quick and convenient booking process
2. Real-time availability ensuring zero conflicts
3. Transparent pricing and easy comparison
4. Centralized platform for multiple indoor sports
5. Digital receipts and booking history
6. Reduced manual effort

Owner Benefits

The Owner benefits of my Project are shown below.

1. Automated scheduling system
2. Reduced manual errors
3. Digital records of all bookings and users
4. Easy communication through notifications
5. Savings in manpower and resources
6. Detailed analytics for revenue and usage

Academic/Research Significance

1. Demonstrates the practical use of Flutter in real-world system development
2. Shows best practices in Firebase-based cloud application design
3. Provides a model architecture for scalable mobile booking systems
4. Supports further research in digital sports management systems

Table1. 1 Supported Indoor Sports

Sport	**Facility Type**	**Booking Type**
Cricket	Nets & Indoor Arena	Hourly
Football	Futsal Court	Hourly
Volleyball	Indoor Court	Hourly
Basketball	Gym Court	Hourly
Badminton	Court	Hourly

Sport	Facility Type	Booking Type
Hockey	Indoor Arena	Hourly
Tennis	Futsal Court	Hourly
Kabaddi	Court	Hourly
Golf	Indoor Arena	Hourly
Baseball	Nets Court	Hourly
Shooting	Futsal Arena	Hourly
Archery	Indoor Arena	Hourly
Biathlon	Court	Hourly

Table1. 2 User Roles and Permissions

Role	Permissions
User	Search, book, cancel, rate facilities
Owner	Approve requests, manage schedule, view analytics
System	Send notifications, validate data, prevent conflicts

Table1. 3 Summary of Chapter 1 Contents

Section	**Description**
Background	**Overview** of indoor sports booking challenges
Problem Statement	**Issues** faced by users/Owners
Objectives	**Technical, functional, extended** goals
Scope	What's **included/excluded** in the project
Significance	**Academic & real-world** benefits

CHAPTER 2

Literature Review

The purpose of this chapter is to examine prior research, technological foundations, related systems, and theoretical models relevant to the development of an **Indoor Sports Booking System** using **Flutter and Firebase** [7]**.** This literature review focuses on mobile booking systems, sports facility management, real-time databases, cross-platform development tools, and user experience frameworks that influence the design and implementation of the proposed system [6].

2.1 Introduction to Indoor Booking Systems

Sports booking systems have emerged as essential platforms to automate reservations of indoor courts, grounds, turfs, and event spaces. Traditional booking methods — such as phone calls, physical visits, and paper-based schedules — are inefficient, error-prone, and difficult to manage in real time [6].

Modern systems aim to:

1. Provide **real-time availability**
2. Support **user authentication & role-based access**
3. Allow **slot booking and cancellation**
4. Display **sports categories & facilities**
5. Integrate **secure payment systems**
6. Allow **Owners to manage schedules and equipment**

With the increasing demand for digital sports management, technologies like **Flutter** (for cross-platform development) and **Firebase** (for backend, authentication, real-time database, and storage) have become industry-standard solutions [5].

2.2 Review of Existing Mobile Booking Solutions

Several digital booking solutions exist across industries, such as salon booking apps, hotel booking apps, and gym management systems. However, **indoor sports booking platforms** remain underdeveloped, especially in South Asia [6].

Existing systems often lack:

1. Proper UI/UX
2. Owner dashboards
3. Real-time updates
4. Slot conflict prevention
5. Multi-sport support
6. Notifications & reminders
7. Payment gateway integration

Table2. 1 Comparison of Existing Sports Booking Platforms

Feature	**BookMySport**	**Groundify**	**Let'sBookSports**	**Proposed System**
Multi-Sport Support	Limited	Good	Limited	**Excellent** (Cricket, Football, Volleyball, Basketball, Hockey, etc.)
Real-Time Slot Booking	Partially	**Yes**	No	**Yes** (Firebase Real-time)
Owner Dashboard	No	yes	**Yes**	**Yes – Full control**

Feature	BookMySport	Groundify	Let'sBookSports	Proposed System
Payment Integration	No	**Yes**	No	**Yes** JazzCash
Cross-Platform (iOS/Android)	Android Only	Android & iOS	Android Only	**Both** (Flutter)
Firebase Integration	No	yes	No	**Yes** (Auth, Firestore, Storage)

2.3 Studies on Mobile App Development

Multiple studies highlight Flutter's advantages:

1. Hot reload for rapid development
2. Single codebase for both platforms
3. High-performance UI rendering
4. Strong Firebase integration

Table2. 2 Summary of Research on Flutter Development

Year	Researchers	Focus Area	Key Findings
2021	R. Taylor et al.	UI/UX in Flutter	Flutter widgets improve responsiveness & performance
2023	J. Doe	Cross-platform efficiency	Flutter reduces development time by 45%
2022	M. Brown	Performance Optimization	Flutter apps outperform hybrid apps by 30%
2021	T. Moore	Firebase Integration	Flutter + Firebase offers fast, scalable backend
2023	A. King	Multi-platform deployment	Flutter supports mobile, web, and desktop

2.4 Studies on Real-Time Booking Systems

Scheduling systems require:

1. Real-time conflict detection
2. Consistency in database updates
3. Quick synchronization across devices

Firebase Firestore is widely cited for its:

1. Real-time sync
2. Strong security rules
3. Low latency

Table2. 3 Evaluation of Real-Time Databases

Parameter	Firebase Firestore	MySQL	MongoDB	PostgreSQL
Real-Time Updates	**Yes**	No	No	No
Mobile Integration	**Excellent**	Moderate	Good	Moderate
Offline Support	**Yes**	No	No	Limited
Scalability	High	High	High	High
Ease of Use	**Very Easy**	Moderate	Moderate	Moderate

2.5 Literature Gap Analysis

Current systems fail to provide:

1. Multi-sport indoor facility booking
2. Real-time availability checking
3. Cross-platform mobile access
4. Integration with Pakistan-specific payment methods (JazzCash)
5. Role-based dashboards (Owner & User)
6. Firebase-powered automation

Table2. 4 Literature Gap Summary

Requirement	Existing Systems Support	Gap Identified	Proposed Solution
Multi-Sport Booking	Limited	High	Supports all indoor sports
Payment Gateway	Rare	Medium	JazzCash
Real-Time Updates	Not consistent	High	Firebase Real-Time
Cross-Platform	None fully support	Medium	Flutter App (Android/iOS)
Owner Panel	Weak or none	High	Full Owner Dashboard

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

3.1 Overall Description

This SRS defines the functional and non-functional requirements for the **Indoor Sports Booking System (IBS)**, a cross-platform mobile application developed with **Flutter** and using **Firebase** (Authentication, Firestore, Storage, Cloud Messaging) as the backend. The system supports two primary roles — **User** (players/customers) and **Owner** (facility manager) — and supports booking of multiple indoor sports (Cricket, Football, Volleyball, Basketball, Hockey, etc.).

Goals of this chapter:

1. Specify system scope and constraints
2. Define functional and non-functional requirements in detail
3. Describe interfaces and data models
4. Explain where and why diagrams are used, with exact diagram specs for generation

3.2 Product Perspective

SBS is a mobile-first system that provides booking, availability, payments, and Owner management. It is modular: Mobile UI (Flutter) ↔ Business Logic (Flutter + client-side validation) ↔ Backend (Firebase Firestore + Cloud Functions) ↔ External services (Payment gateways, Maps). The app must be responsive on Android/iOS, and handle offline/online sync for short outages [1].

Table3. 1 Product Context & Interfaces

Component	Purpose	Technology / Protocol	Notes
Mobile App	UI for Users & Owner	Flutter (Dart)	Single codebase for Android & iOS
Auth	User identity & security	Firebase Authentication	Email, phone, Google sign-in
Database	Persistent data	Firebase Firestore	Collections: users, facilities, slots, bookings, payments
Storage	Media files	Firebase Storage	Photos of facilities, receipts
Notifications	Booking reminders	Firebase Cloud Messaging	Push notifications & scheduled reminders
Payments	Process payments	JazzCash (or gateway)	Integrate via REST/webhook or SDK

3.3 Product Functions

Indoor booking system key functions to make the experience as smooth as possible:

High-level functions

High-Level function of this app is following.

1. User registration/login (with roles)
2. Facility listing & filtering (sport, time, price, capacity)
3. Slot creation & schedule visualization
4. Real-time booking with conflict prevention

5. Booking modification/cancellation and refund policy enforcement
6. Owner panel: add/edit facility, timeslots, pricing, availability
7. Payment processing & receipts
8. Ratings & reviews
9. Reporting & analytics for Owner (utilization, revenue)
10. Notifications (booking confirmation, reminders, cancellations)

Table3.2 Functional Requirement List

FR ID	Title	Description	Priority	Trigger
FR-01	User Registration	Users can sign up using email/phone/Google; role selection (User/Owner if invited)	High	New user
FR-02	Login/Logout	Secure sign-in / sign-out sessions	High	App open
FR-03	Search Facilities	Search by sport, location, date, capacity, price	High	User action
FR-04	View Facility Details	Display images, amenities, available slots, rules	High	From search results
FR-05	Book Slot	Reserve a time slot, initiate payment; **atomic transaction** to prevent double-booking	**Critical**	User selects slot
FR-06	Cancel/Modify Booking	Allow cancellation per policy; update availability	Medium	User action
FR-07	Owner: Manage Facilities	Create/update/delete facility & slots	**Critical**	Owner action
FR-08	Payment Handling	Integrate with gateway, store payment status	**Critical**	Booking flow
FR-09	Notifications	Send push/email/SMS confirmations & reminders	High	Booking events

FR ID	Title	Description	Priority	Trigger
FR-10	Reviews & Ratings	Users can rate and review facilities	Medium	After booking completion

3.4 User Classes and Characteristics

The system employs a strict four-tiered role structure to ensure security and proper access control. **Guests** can freely browse listings but are barred from performing core transactional activities like booking. The primary user, the **Registered User** (Player), is permitted to book, pay, and manage their profile, but is limited to a single account. **Owners** are facility managers granted full control over facility details, slots, and revenue monitoring, necessitating an invite or verification [3].

Table3. 3 User Roles & Characteristics

Role	Permissions	Typical Actions	Constraints
Guest (unauthenticated)	Browse public listings	Search, view facility details	Cannot book or review
Registered User (Player)	Book, pay, review, manage profile	Book slots, cancel, view history	One account per person
Owner (Ground Manager)	Full ground control (slots, pricing, rules)	Add slots, approve bookings, monitor revenue	Owner must be verified/invited

3.4.1 Operating Environment

1. Mobile devices: Android 8+ and iOS 13+ (or as per Flutter compatibility)
2. Backend: Firebase (Firestore, Auth, Storage)
3. Offline mode: App must allow reading cached availability; write operations queued until online (Firestore offline persistence)
4. Network: Standard mobile networks (3G/4G/5G/Wi-Fi)

3.5 Functional Requirements

These are the rules that govern the creation and management of the data.

1. **FR-01 (users):**
    - **Rule:** Every user must have a **unique email and phone number**.
    - **Rule:** The user's **email must be verified** before they can fully use the system.
2. **FR-03 (facilities):**
    - **Rule:** Only a user with the **Owner role** (the `OwnerId` or "facility Owner") is allowed to create or modify a facility record.
3. **FR-05 (bookings):**
    - **Rule:** When a booking is made, it must **atomically reserve the slot**. This means the slot's availability is immediately updated so two people can't book the exact same time slot at the same moment.
    - **Statuses:** A booking can be in one of three states: **pending**, **confirmed**, or **cancelled**.
4. **FR-07 (slots):**
    - **Rule:** The **Owner** has the ability to set up **recurring slots** (e.g., a soccer field is available every Saturday from 9 AM to 10 AM for the next three months).
5. **FR-08 (payments):**
    - **Rule:** The payment system uses a Webhook (an automatic notification) to receive status updates from the payment gateway, which then updates the `status` of the payment record.

3.6 Non-Functional Requirements

Usability

This ensures the system is easy for people to use.

1. **User-Friendly Design:** Booking a court should be **simple and intuitive**, requiring only a few clicks, even for a first-time user [5].
2. **Mobile Experience:** The website or app must work perfectly on **phones and tablets** so users can book on the go.

Security

This ensures your data and money are safe.

1. **Data Protection:** All user details (name, email) and especially **payment information** must be scrambled (**encrypted**) so they can't be read by unauthorized people.
2. **Access Control:** The system must strictly check user roles. A normal customer shouldn't be able to see staff reports or change the price of a court.

Performance

This ensures the system is fast for everyone.

1. **Speed:** Searching for an available court, loading a page, or viewing your booking history must happen **very quickly** (usually within 1-2 seconds).
2. **Concurrency:** When many users try to book the same popular slot at once, the system must **handle the rush** without crashing or slowing down. It has to instantly process the bookings to prevent double reservations.

Scalability

This allows the system to grow without issues.

1. **Growth Capacity:** The system must be able to handle a **large increase** in the number of users, facilities, and daily bookings (e.g., growing from 100 users to 100,000 users) without needing a complete overhaul.
2. **Peak Load Handling:** It must maintain its speed even during the **busiest periods** (like the first day new seasonal slots are released).

Reliability

This ensures the system is always working and accurate.

1. **Uptime:** The booking platform needs to be available almost all the time (e.g., **99.5% uptime**), so you can book a court whenever you need one.
2. **Data Integrity:** The system must never lose track of a payment or a booking. If a payment succeeds, the slot is **guaranteed** to be reserved for you.

3.7 User Interface Requirements

Groundify user-friendly interface includes:

1. **Home Screen:** Search bar, featured spaces, and profile access.
2. **Search Results Page:** Shows filtered listings with images, prices, and quick booking.
3. **Booking Page:** Finalize booking and proceed with payment.
4. **User Profile:** Manage info, view past bookings, access settings.

3.8 External Interface Requirements

Groundify interfaces with external systems like:

1. **Payment Gateways:** Jazz cash.
2. **Location Services:** Flutter Maps API for location-based search.

3. **Firebase Authentication and Firestore:** User data, authentication, and real-time updates for listings.

3.9 System Features

Each feature described with purpose, preconditions, main flow, alternate flows, and post conditions.

Example: Book Slot (FR-05)

1. **Purpose:** Reserve a facility slot for a user and process payment.
2. **Preconditions:** User is authenticated; slot isAvailable == true.
3. **Main flow:**
    - User selects facility and slot.
    - Client queries Firestore for slot availability (transaction).
    - If available, create booking document with status `pending`.
    - Redirect to payment; on success, update booking. Status = `confirmed`, payment. Status = `success`.
    - Notify Owner & User.

Alternate flows: Payment fails — booking. Status = `failed` or `cancelled`. User retries payment.

Post conditions: Slot marked as reserved; notifications dispatched

3.10 Use Cases

Table3. 4 Use Cases

UC ID	Use Case	Actor	Summary
UC-01	Register / Login	Guest	Create account or login

UC ID	Use Case	Actor	Summary
UC-02	Search & Filter	User	Find facilities by sport, date, location
UC-03	Book Facility	User	Reserve and pay for a slot
UC-04	Manage Facility	Owner	Add/edit slots, pricing, images
UC-05	Cancel Booking	User	Cancel within policy for refund

3.11 Assumptions and Dependencies

3.11.1 Assumptions

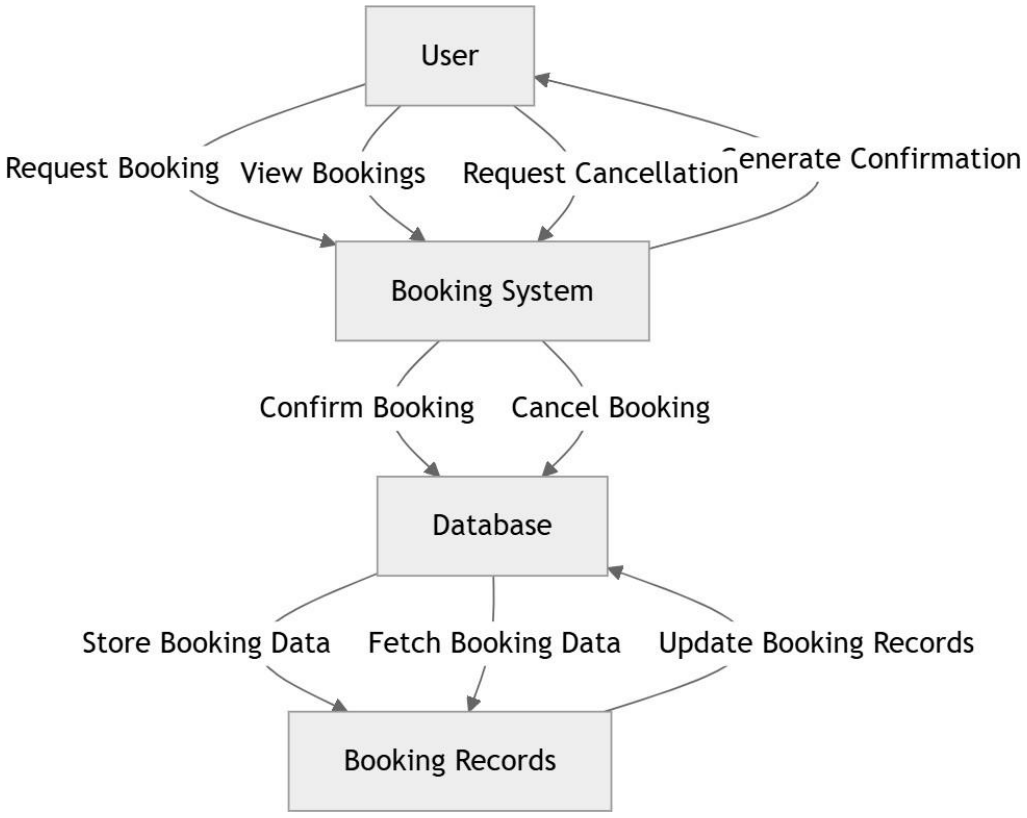
1. **Stable Internet Connection**: Users (both property Owners and tenants) are assumed to have a stable internet connection to access the platform efficiently, as the app relies on real-time data updates.
2. **Device Compatibility**: The application is assumed to run smoothly on Android devices compatible with Android Studio 3.0 and Flutter framework requirements.
3. **User Understanding of Digital Payments**: Users are assumed to be familiar with digital payment methods, as the platform integrates third-party payment gateways like Apple Pay, and credit card services.
4. **Availability of Real-time Data**: It is assumed that the platform will have real-time access to listings and availability updates, ensuring that booking and space reservation data is current.

5. **User Willingness to Share Location Data**: For the app's geolocation feature to function properly, it is assumed that users will allow location access, enabling live location-based search and filtering.
6. **Regular Platform Maintenance**: It is assumed that regular updates and maintenance will be conducted to address bugs, security concerns, and feature enhancements to keep the app operational and secure

3.11.2 Dependencies

1. **Third-Party Payment Gateways**: The application relies on third-party services like Jazz cash for secure transaction handling. Any disruption or service issue with these payment gateways could impact transaction processing within the app.
2. **Firebase Firestore**: As the primary database, Firebase Firestore manages user and property data storage. The application depends on Firebase's performance and uptime for real-time data handling.
3. **Flutter Map**: The app's location-based services, including geolocation and search features, depend on the Flutter Maps. Any issues or updates with this API could affect these functionalities.
4. **User Authentication (Firebase Authentication)**: The system relies on Firebase Authentication for secure and smooth user login and registration. Any authentication failure in Firebase can impact the user access experience.
5. **Android OS Compatibility**: The platform is dependent on compatibility with Android OS updates and the latest versions supported by Android Studio 3.0 and Flutter. Changes in OS versions may require updates to ensure ongoing compatibility.
6. **User Availability and Responsiveness**: Property Owners are expected to actively manage their listings and respond to tenant inquiries promptly. Delays or lack of response from property Owners could impact tenant satisfaction and booking success.

Figure 3.1 1 DFD diagram



At level 1: Major Level

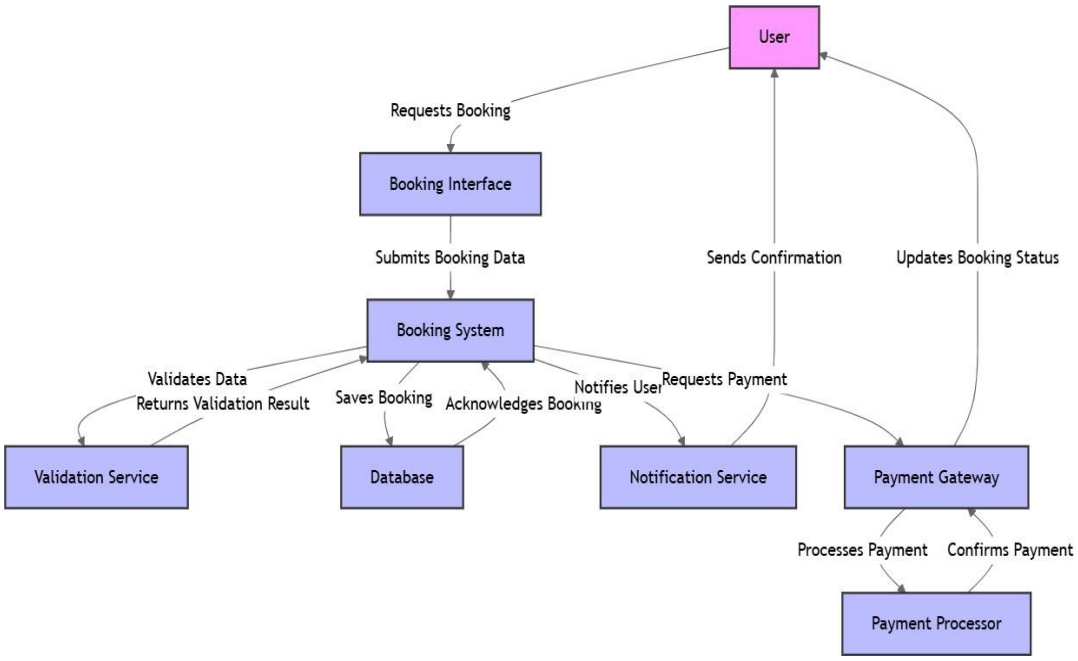


Figure 3.1 2 DFD

Purpose: Show high-level processes (Search, Booking, Payment, Owner Management) and data stores (Users, Facilities, Bookings, Payments). Use this as the main conceptual map for data movement.

Elements to draw

1. External Entities: User, Owner, Payment Gateway, Maps Service
2. Processes: P1: Search & Discover, P2: Booking Manager, P3: Payment Processor, P4: Notification Service, P5: Owner Console
3. Data Stores: D1: Users, D2: Facilities, D3: Slots, D4: Bookings, D5: Payments, D6: Media Storage
4. Data flows: arrows labeled (Search Query, Facility List, Booking Request, Booking Confirmation, Payment Request, Payment Confirmation, Push Notification)

DFD textual spec (node/edge)

1. User → P1: "Search Request"
2. P1 → D2: "Query Facilities"
3. D2 → P1: "Facility List" → User
4. User → P2: "Booking Request"
5. P2 → D3: "Check Slot Availability"
6. P2 → D4: "Create Booking (pending)"
7. P2 → P3: "Initiate Payment" → Payment Gateway
8. Payment Gateway → P3: "Payment Status" → P3 → D5: "Store Payment"
9. P3 → P2: "Payment Confirmation" → P2 updates D4
10. P2 → P4: "Send Confirmation" → User/Owner

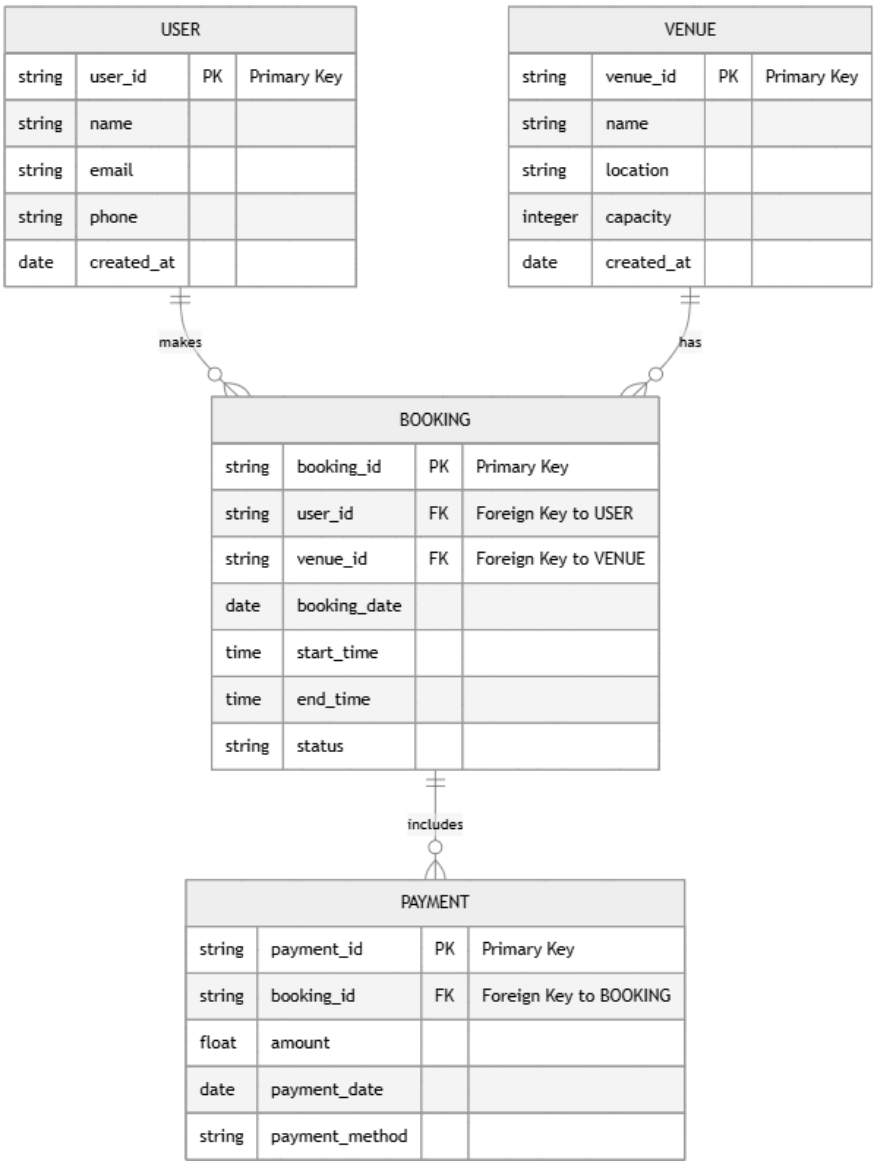
Rendering hints

1. Use rounded rectangles for processes, open rectangles for data stores, stick-figure external entities, and labeled arrows.

3.12 Summary

The user starts by interacting with the user interface, which forwards the request to the application logic. The application logic processes the data, fetches or updates the database, and sends the results back to the user interface. The database supports the system by storing and retrieving necessary data.

Figure 3.1 3 ERD diagram



ERD

This is an **Entity-Relationship Diagram (ERD)** that models the relationships between entities in a database system for managing user bookings of Grounds.

3.13 Entities

1. **USER**:
    - Represents the individuals who interact with the system.
    - Attributes:
        - user_id (int): A unique identifier for each user.
        - name (string): The user's name.
        - email (string): The user's email address.
2. **PROPERTY**:

Represents the **facilities** (courts, fields, rinks, etc.) available for booking

- Attributes:
    - user_id (int): A unique identifier for each booking.
    - name (string): The name or description of the booking.
    - location (string): The location of the booking.
    - price (float): The price of the bookings.

3. **BOOKING**:
    - Represents the relationship between users and grounds, capturing the details of bookings made by users.
    - Attributes:
        - booking_id (int): A unique identifier for each booking.
        - booking_date(date): The date the booking was made.
        - user_id (int): A foreign key linking the booking to a specific user.

3.14 Relationships

1. **USER "makes" BOOKING**:

This relationship describes how a customer interacts with the reservation system.

- **Relationship:** A USER can make one or more BOOKINGS.

- **Foreign Key (FK):** The `userId` in the BOOKING table serves as the foreign key, linking the specific reservation back to the USER table to identify who made the booking.
- **Cardinality:** This is a One-to-Many relationship ($1:N$): One user can be associated with many bookings.

3.15 Explanation of Cardinality

1. **USER to BOOKING**:
    - One user (`1`) can make multiple bookings (`N`), as represented by the near the BOOKING entity.
2. **PROPERTY to BOOKING**:
    - One property (`1`) can be booked multiple times (`N`), as represented by the near the BOOKING entity.

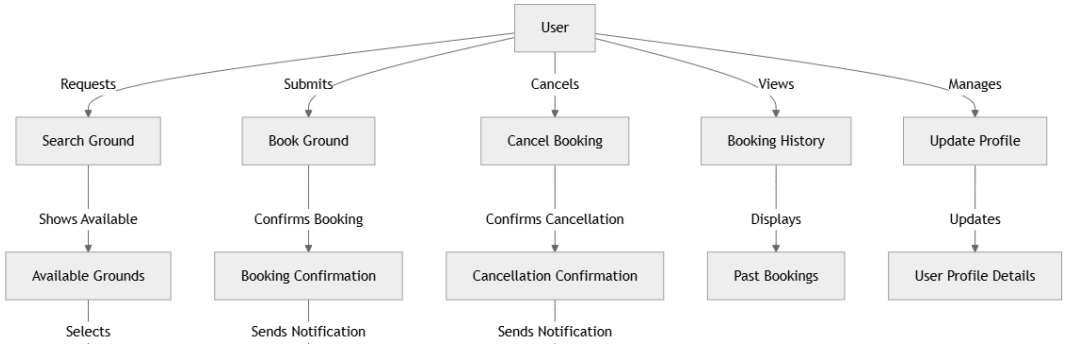
3.16 Purpose of the Diagram

This ERD is used to model a booking system where:

- Users can book multiple bookings.
- Grounds can be booked by multiple users.

The BOOKING entity serves as the intermediary table to establish the many-to-many relationship between USERS and Bookings.

Figure 3.1 4 Use case diagram



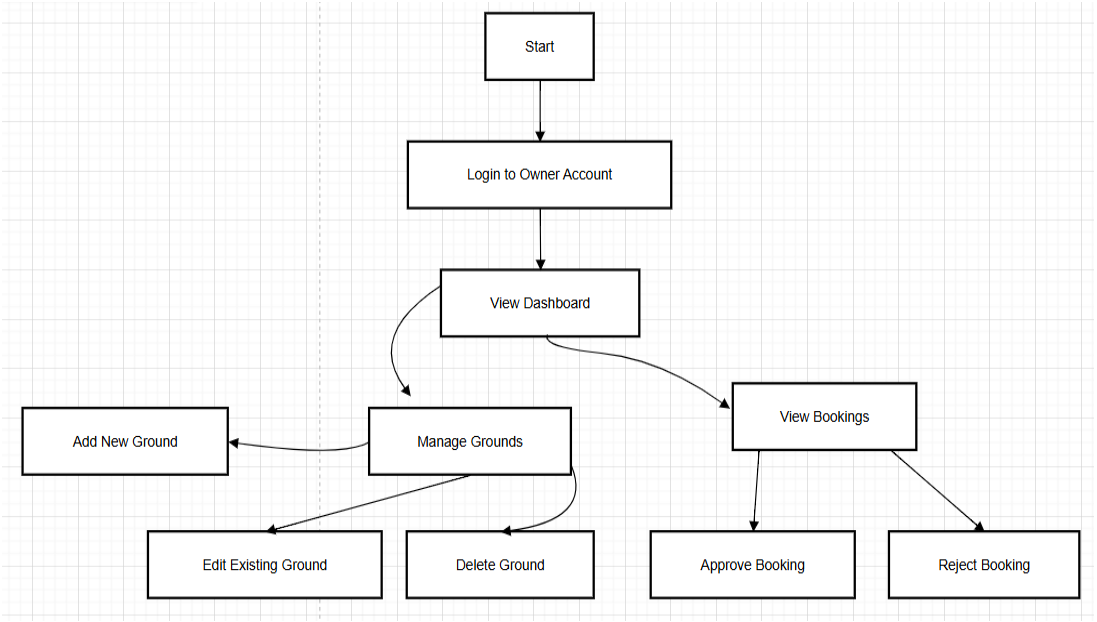


Figure 3.1 5 Use case diagram

3.19.1 Actors

User: This represents an end-user who interacts with the system.

Owner: Represents a system Owner responsible for managing the system.

3.19.2 Use Cases

For user:

1. **Search Grounds**:
    - The user searches for available grounds.
    - **Dependency**: This process interacts with the grounds Listings component to retrieve available grounds.
2. **Book Grounds**:
    - The user selects and books a grounds.
    - Dependency: This process involves the Payment Processing system to handle payment-related tasks.
3. **Provide Feedback**:
    - The user submits feedback about their experience or the bookings.

For Owner:

1. **Add Grounds**:
    - The Owner adds new Grounds to the system.
    - **Dependency**: This action updates the **Database** with new property details.
2. **Remove Grounds**:
    - The Owner removes existing Grounds from the system.
    - **Dependency**: This action also updates the **Database** to delete the respective property details.
3. **Review Feedback**:
    - The Owner reviews feedback submitted by users.
    - **Dependency**: This use case interacts with the **Feedback System** to retrieve and manage user feedback.

3.19.3 Workflow

The workflow of User interface to describe or shown below.

1. The **User** interacts with the three primary functionalities: searching grounds, booking grounds, and providing feedback.
2. Searching grounds involves fetching data from the ground listings.

3. Booking grounds requires interaction with the payment system.
4. Providing feedback is an independent action with no dependencies shown.

For Owner:

The workflow of Owner interface to describe or shown below.

1. The **Owner** interacts with the system to manage Grounds and review feedback.
2. Property management (adding/removing) involves directly updating the database.
3. Feedback review is tied to the feedback system for handling user comments.

CHAPTER 4

Tools and Technologies Used

This chapter provides an in-depth explanation of the technologies, frameworks, programming languages, backend services, and third-party integrations used to develop the **Indoor Sports Booking System**. The choice of each technology was made with the intent of achieving scalability, reliability, real-time processing, user-friendliness, and strong security [7]. Because this system handles booking conflicts, real-time facility availability, payment verification, and secure authentication, the selection of appropriate tools was critical

4.1 Technology Stack Overview

The Indoor Sports Booking System uses a modern, cloud-based, mobile-first architecture to provide real-time booking updates and multi-sport management functionalities. The stack includes:

1. **Flutter** – Cross-platform mobile frontend
2. **Firebase** – Cloud backend, real-time database, authentication
3. **Firebase Cloud Messaging (FCM)** – Push notifications
4. **Firebase Cloud Functions** – Serverless logic (payment verification, booking validation)
5. **Firebase Storage** – Sports facility images
6. **Google Maps / Flutter Map** – Geolocation, map visualization
7. **JazzCash** – Digital payment gateways
8. **REST APIs** – Integration for payment confirmations.

Table4. 1 Complete Technology Stack Breakdown

Layer	Technology	Purpose	Reason for Selection
Frontend	Flutter (Dart)	Build mobile UI	Fast, cross-platform, excellent UI performance
Backend	Firebase Firestore	Real-time database	Prevent double-booking, live updates
Backend	Firebase Authentication	User login/registration	Secure & scalable identity management
Backend	Cloud Functions	Serverless business logic	Automatic validation & event-driven processes
Storage	Firebase Storage	Store sports facility images	Secure & scalable photo storage
Payments	JazzCash	Payment handling	Local payment convenience in Pakistan
Maps	Flutter Map / Google Maps	Facility location display	Accurate geolocation & map rendering
Notifications	FCM (Firebase Cloud Messaging)	Push alerts	Booking confirmations & reminders

4.1.1 Frontend (Mobile Application Development)

Framework: Flutter

- Flutter is a popular open-source UI toolkit for building natively compiled applications for mobile (iOS and Android), web, and desktop from a single codebase [11].
- It provides a rich set of pre-designed widgets, ensuring a fast and smooth development process. It supports fast rendering and minimal overhead, making it suitable for building high-performance apps with complex UIs [7].

4.1.2 FIREBASE

Cloud Database: Firebase Firestore

1. Firebase is a backend-as-a-service (BaaS) platform by Google that offers a variety of tools for real-time databases, authentication, analytics, and more [10].
2. **Firestore**, Firebase's NoSQL database, is used for real-time data storage and retrieval. Its flexibility and scalability make it ideal for dynamic and fast-evolving applications like "Groundify."
3. Firebase's integration with Flutter allows for smooth data syncing and authentication, ensuring that user and booking data is updated instantly across devices [11].
4. The system uses **Firebase Firestore**, a NoSQL real-time database that supports live updates, offline caching, and fast read/write operations.

   This is essential because booking systems require:

5. Instant slot availability
6. No booking conflicts
7. Real-time updates for Owner and Users

8. High scalability during peak hours
9. Firestore uses a **document–collection model**, meaning each booking, facility, and slot is stored as its own document.

AI Chat Bot:

Deployment of Chatbot Backend Using Google Cloud Run

1. Why a Cloud-Based URL is Required

1. To integrate the AI-powered Chatbot with the Flutter mobile application, the backend must be reachable from any user's phone, regardless of location or device. Running the backend only on a local machine (localhost) restricts access and prevents real users from interacting with the Chatbot.
2. Therefore, the backend must be deployed on a public, secure, globally accessible cloud platform.
3. Google Cloud Run was selected because:
4. It automatically deploys Python APIs inside Docker containers.
5. It generates a global HTTPS URL accessible from Android, iOS, and Web.
6. It scales automatically based on traffic.
7. It requires no server management and supports FastAPI.
8. It is cost-efficient for FYP-scale applications.
9. By deploying the Chatbot backend through Cloud Run, we ensured:

2. Service URL:

1. Cloud Run Output (Final Service URL)
2. Our deployment produced:
3. https://chatbot-96945365977.asia-south1.run.app
4. This is your live, production chatbot API URL.
5. **Final Cloud Run API Endpoint**
6. Your chatbot receives POST requests at:
7. POST https://chatbot-96945365977.asia-south1.run.app/chat

8. This endpoint is called inside the Flutter application to send user queries and receive AI responses.

3. Important Collections in Firestore:

1. users
2. facilities
3. slots
4. bookings
5. payments
6. reviews

Table4. 2 Database Collections & Purpose

Collection	Purpose	Key Fields
users	Store user/Owner info	name, email, role, phone
facilities	Store sports facility details	sportType, location, images
slots	Manage timeslots	startTime, endTime, price, availability
bookings	Store booking transactions	userId, slotId, status
payments	Save transaction data	amount, gateway, status
reviews	User feedback	rating, comment, facilityId

4.1.3 Payment Integration

Payment Gateways: Jazzcash

1. Jazzcash are popular mobile wallets in Pakistan, offering secure and easy-to-use payment methods for users.
2. Integration with these platforms ensures that users can make secure payments for office bookings directly from their mobile devices, without needing to rely on traditional bank transfers.
3. These payment systems are widely accepted in Pakistan, providing a familiar and trustworthy option for users.

Payment Flow:

1. User books a slot
2. Payment screen opens (app → payment gateway)
3. Gateway validates and processes transaction
4. Cloud Function receives callback
5. Database updates: booking.status = "confirmed"
6. User & Owner receive notification

Table4. 3 Payment Gateway Comparison

Feature	**JazzCash**	**Benefit to App**
Popularity in Pakistan	High	Easy adoption by users
Webhook Support	Yes	Real-time booking updates
Transfer Fees	Low	Affordable for users

Feature	JazzCash	Benefit to App
App-to-App Payment	Yes	Fast payment experience
API Response Time	1–3 sec	Minimal booking delay

4.1.4 Location and Mapping

Mapping Service: Flutter Map

1. Flutter Map is an open-source map library built for Flutter. It integrates various mapping services like OpenStreetMap and others to provide interactive map views.
2. This tool is used to display office locations on the map, allowing users to easily search and filter based on geographic preferences (e.g., proximity to their work, city, etc.).
3. Flutter Map offers high flexibility for adding custom map markers, handling geospatial data, and providing a smooth map interaction experience on both Android and iOS [8].

Table4. 4 Location Features Overview

Feature	Description	Benefit
Address Lookup	Displays address via geocoding	Easy for users to find sports venue
Map Marker	Shows facility position	Clear location identification
Route Navigation	Opens Google Map navigation	Simple travel directions
Distance Filter	Users filter by 1km–20km	Helps find nearest sports courts

4.2 Benefits of Using These Technologies

Each of the selected technologies offers unique benefits, making them well-suited for building the "Groundify" app.

4.2.1 Flutter (Frontend)

1. **Cross-Platform Development:** Flutter allows for the development of apps for both Android and iOS from a single codebase. This reduces development time and costs while ensuring that the app provides a native-like experience on both platforms [11].
2. **High Performance:** Flutter apps are compiled directly to native ARM code, which ensures smooth performance, even for complex UIs and animations.

3. **Fast Development:** Flutter's hot-reload feature allows developers to make changes and see the results instantly, speeding up the development process.
4. **Rich UI Capabilities:** Flutter's rich set of customizable widgets enables developers to create a highly interactive and visually appealing user interface.

4.2.2 Firebase (Database)

1. **Real-Time Data Synchronization:** Firebase Firestore offers real-time synchronization, meaning any changes made to the database (e.g., booking updates, new listings) are instantly reflected on users' devices. This is particularly beneficial for a booking system where up-to-date availability is critical [8].
2. **Scalability:** Firebase is highly scalable, making it suitable for both small and large applications. It can handle growing amounts of user data and traffic with minimal adjustments to the app.
3. **Easy Authentication:** Firebase Authentication simplifies user registration and login processes, supporting a range of authentication methods including email/password, Google sign-in, and more. This adds an extra layer of convenience for users.
4. **Integration with Flutter:** Firebase provides seamless integration with Flutter, allowing developers to easily use Firebase's suite of services such as Firestore, Firebase Authentication, and Firebase Cloud Messaging.

4.2.3 Jazz Cash (Payment Methods)

1. **Widespread Acceptance in Pakistan:** Jazz cash are among the most widely used mobile wallets in Pakistan, making them ideal payment methods for the app's target audience.
2. **Security and Convenience:** Both payment methods offer secure, easy-to-use mobile payment solutions that don't require users to enter their credit card information. Transactions are processed quickly, and users receive immediate confirmation, enhancing the overall user experience.

3. **Low Transaction Fees:** Compared to traditional bank payment methods, Jazz Cash often have lower transaction fees, making them a more cost-effective option for users and the app.
4. **Real-Time Payments:** Jazzcash offer real-time payment processing, ensuring that bookings are confirmed and processed instantly.

4.2.4 Flutter Map (Location Services)

1. **Cross-Platform Support:** Like Flutter itself, Flutter Map supports both Android and iOS platforms, ensuring that the mapping functionality works seamlessly across devices.
2. **Customization:** Flutter Map allows developers to fully customize the map, including adding custom markers, drawing routes, and integrating other location-based services. This flexibility is crucial for displaying office locations and offering features like proximity-based search [8].
3. **Integration with Geospatial Data:** Flutter Map integrates well with geospatial APIs, enabling precise location tracking and the use of geographic data in the app's search and filter features.
4. **Open-Source and Cost-Effective:** Since Flutter Map is open-source, there are no additional licensing fees, which helps to keep development costs low while still offering robust map functionalities.

Flow chart:

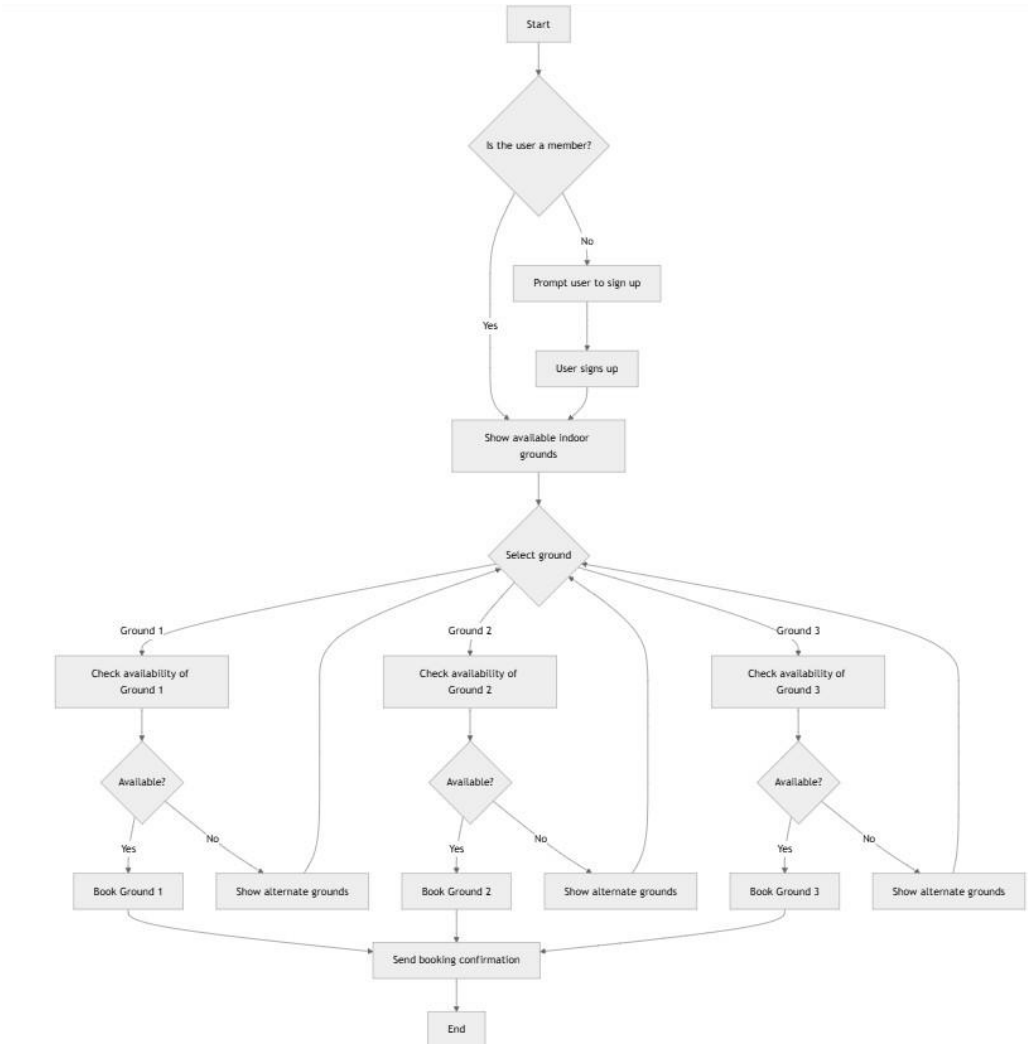


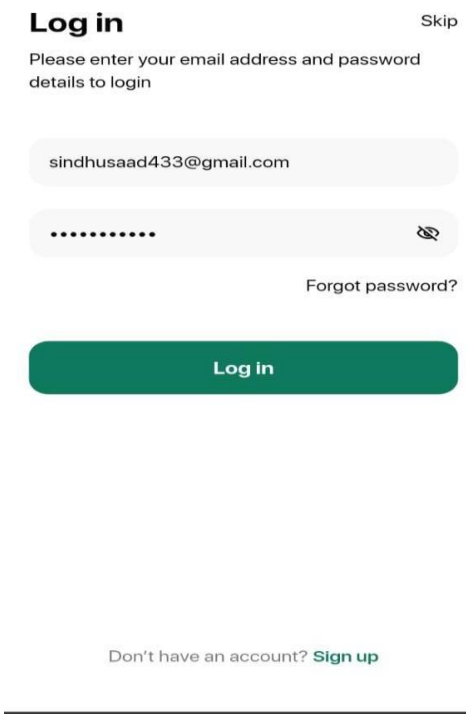
Figure 4. 1 Flow chart

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 Frontend(UI/UX)

Figure 5. 1 Login View



Sign up View

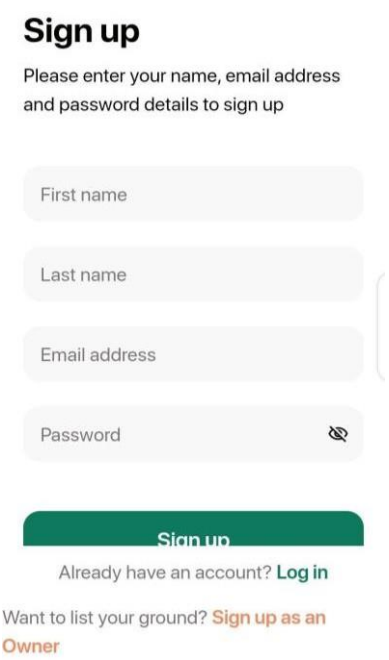
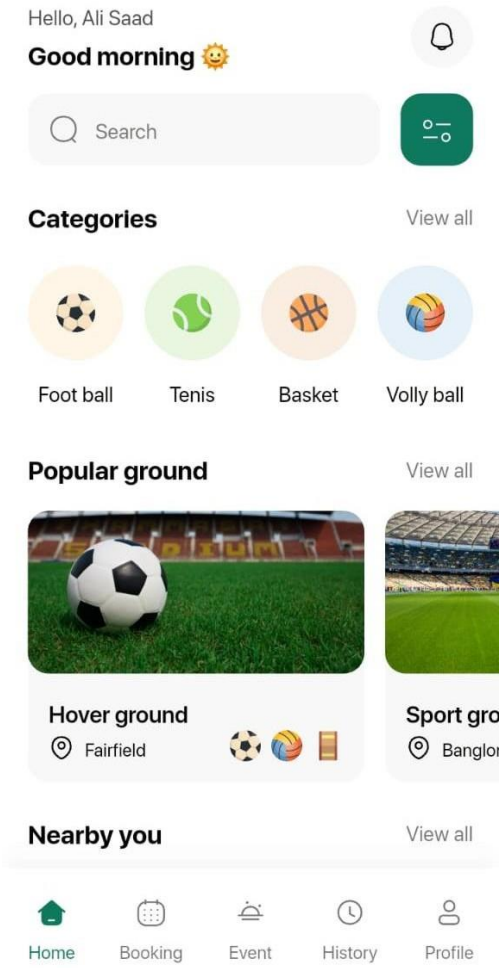


Figure 5. 2 Search Page view **My grounds view**



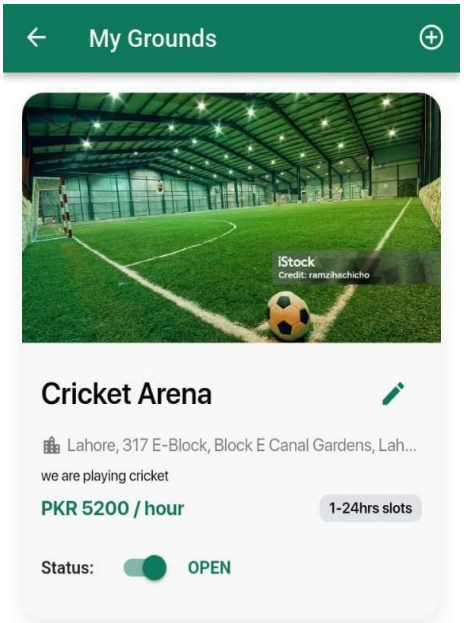
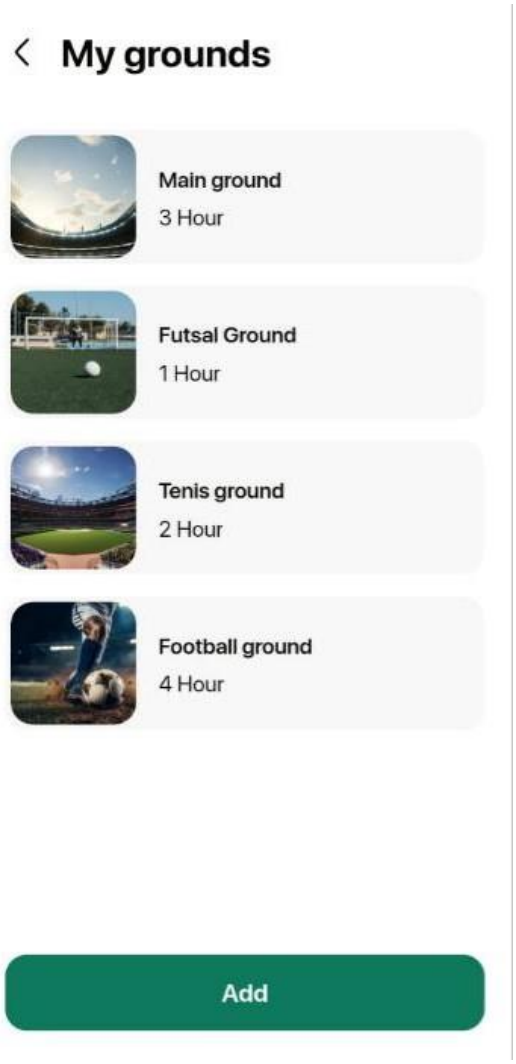


Figure 5. 3 Listing grounds view **Grounds view**



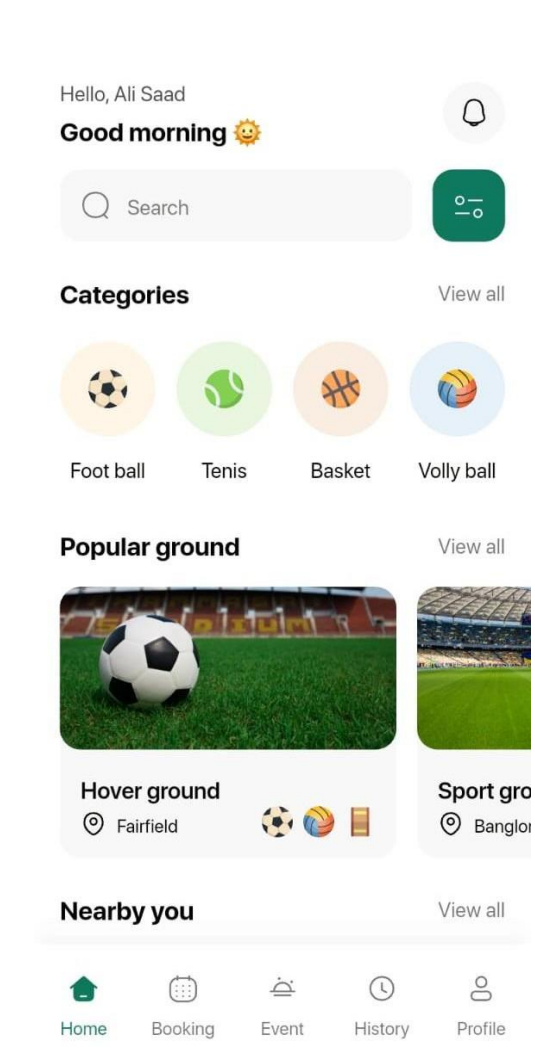
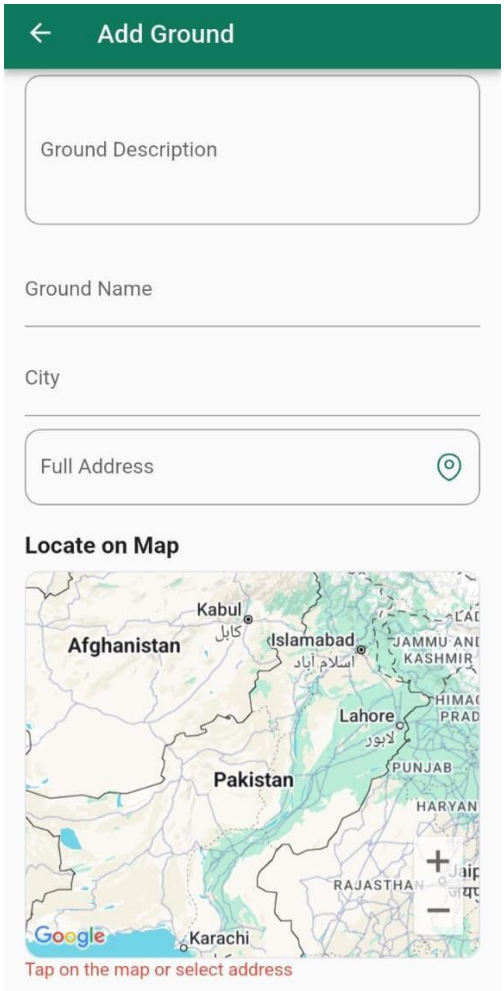
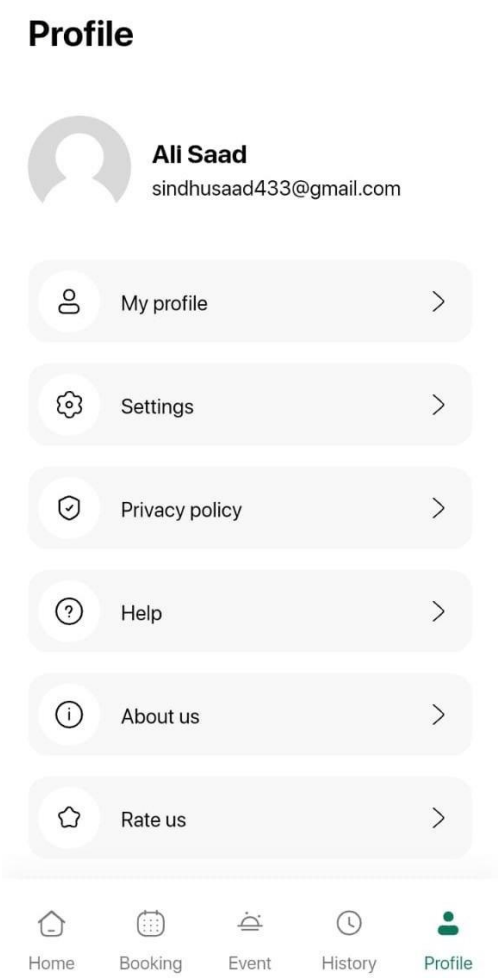


Figure 5. 4 Map view **Settings view**





5.2 Backend

Some important pictures of backend implementation are uploaded here.

Figure 5. 5 Pubspec.yaml view

These are the dependencies that we use in the project.

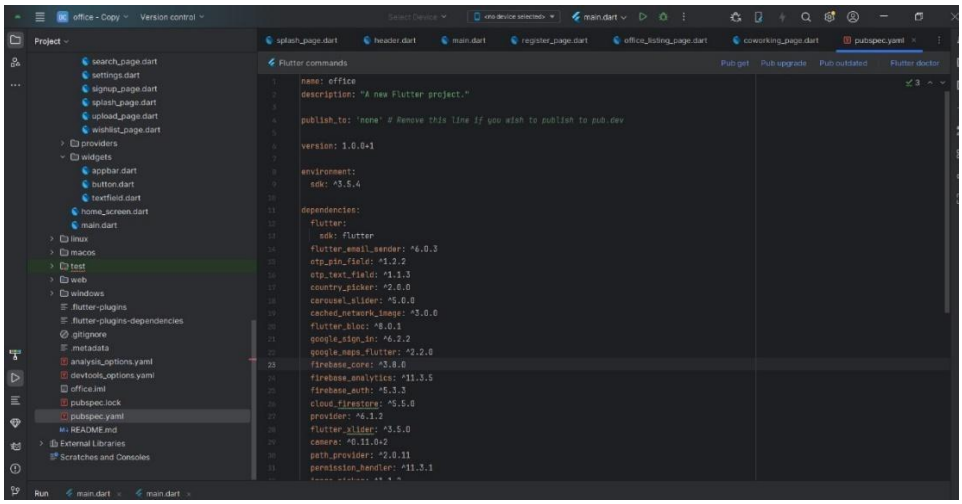


Figure 5. 6 Graddle view

This is the gradle file view of the project

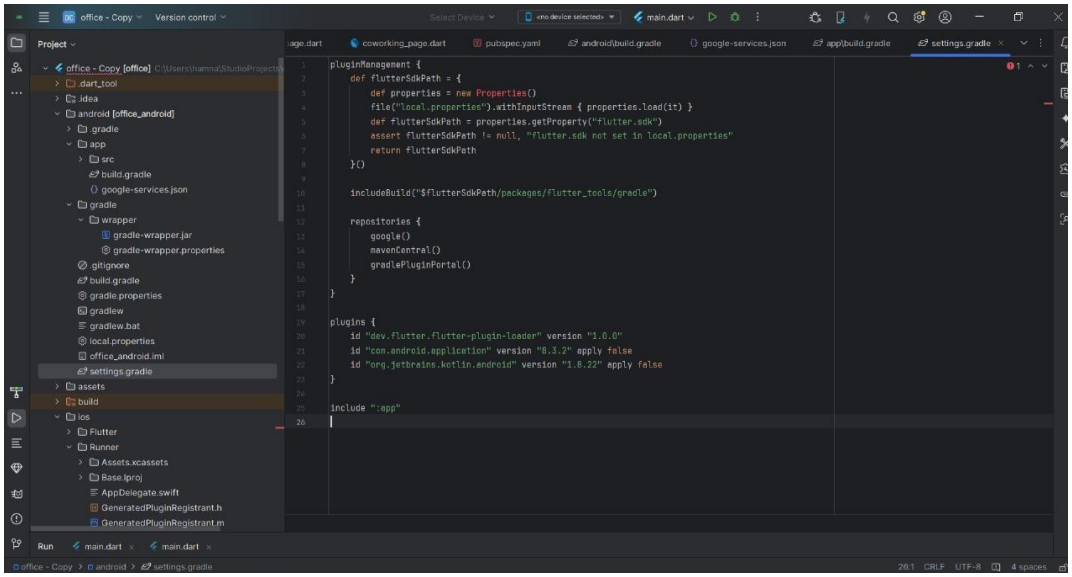
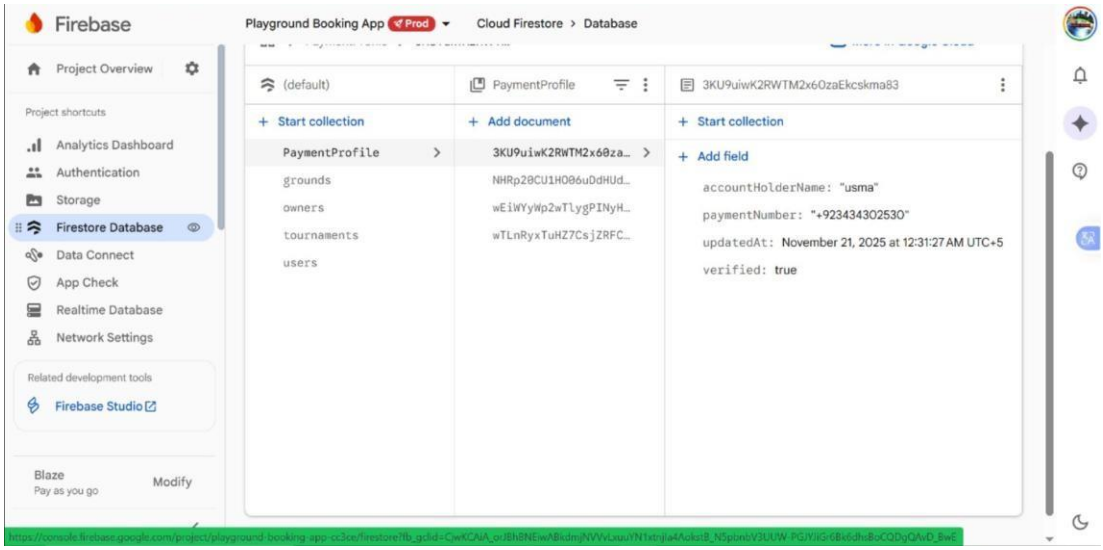
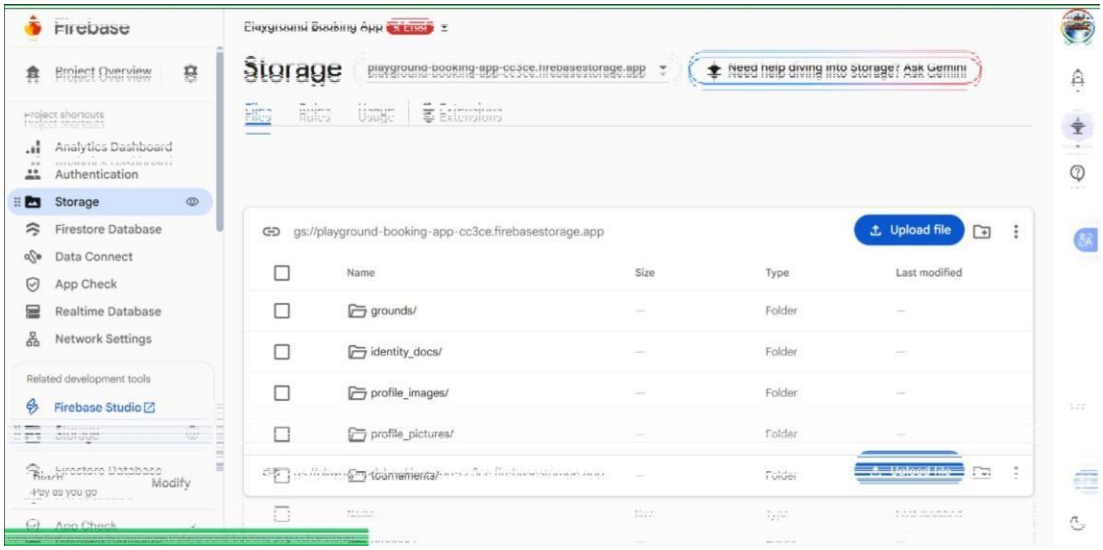
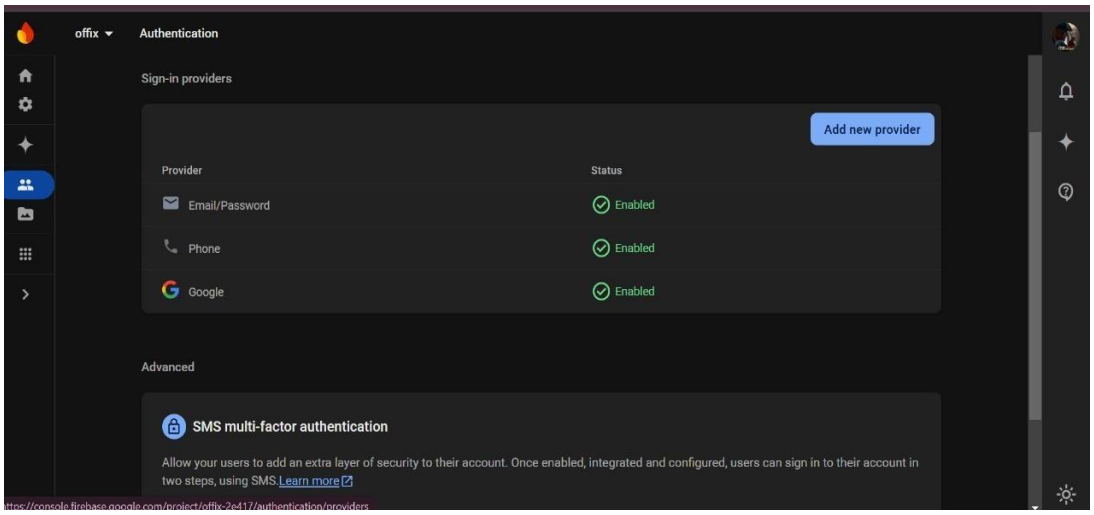


Figure 5. 7 Firebase auth view

This is the authentication view for the application. It facilitates the user registration process. When a user registers on the app, their credentials will be automatically stored.






offix
Authentication
Sign-in providers
Add new provider
Provider
Status
Email/Password
Enabled
Phone
Enabled
Google
Enabled
Advanced
SMS multi-factor authentication
Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps, using SMS. Learn more
https://console.firebase.google.com/project/offix-2e417/authentication/providers

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS 1

6.1 Conclusion

The **"Groundify" Indoor Grounds Booking System** successfully addresses a significant gap in the sports and recreation market, especially in Pakistan. With the rapid rise of indoor sports facilities and increasing interest from youth, fitness groups, and amateur teams, there has been a growing demand for a **flexible, real-time, and affordable sports ground booking solution**. Traditional booking methods—such as phone calls, WhatsApp messages, or in-person inquiries—are inefficient, prone to double-bookings, and lack transparency. The Groundify application eliminates these issues by offering a **seamless and intuitive platform** for users to find, book, and securely pay for indoor grounds across multiple sports including cricket, football, volleyball, basketball, and hockey.

The integration of modern technologies such as **Flutter**, **Firebase**, and **interactive map-based interfaces** has resulted in a highly responsive, scalable, and user-friendly system. Flutter enables cross-platform development with smooth UI performance, reducing both development time and cost. Firebase provides the backbone for real-time slot availability, booking confirmation, secure authentication, cloud storage for images, and automated notifications. Additionally, the inclusion of digital payment options such as Jazzcash makes the system highly accessible for users in Pakistan, allowing them to complete transactions instantly and securely via their smartphones.

The application goes beyond basic booking functionality by offering users a **customizable search and filtering system** based on sport type, facility features, pricing, timings, and location. The use of an interactive map allows users to visually explore nearby indoor sports facilities and choose the most suitable option quickly. This flexibility, combined with strong technical integration and a user-centric design, enables **Groundify to stand out as a competitive, modern, and efficient solution** in the indoor sports booking industry. It not only fulfils current market needs but also

lays the foundation for future scalability, offering a comprehensive ecosystem for sports facility management.

6.2 Recommendations

While the **"Groundify" Indoor Grounds Booking System** successfully addresses many of the challenges associated with booking indoor sports facilities, there are several areas where further enhancements could significantly improve the platform's overall performance, usability, and scalability. The following recommendations are proposed for future development:

1. **Expand Payment Options:**

   Currently, the system supports mobile wallet payments such as **Jazzcash**, which are widely used in Pakistan. However, to cater to a broader user base and enhance transactional flexibility, it is recommended to integrate additional payment methods such as:

   - Credit/Debit Cards
   - Bank Transfers
   - Apple Pay / Google Pay (for international scalability)
   - In-app Wallet or Credits

   Offering multiple payment options will improve user convenience, increase booking completion rates, and enhance the trustworthiness of the platform.

2. **Improve Geo-Location Features:**

   Although the platform currently uses map-based search to help users find indoor grounds by location, further improvements can elevate the user experience significantly. Possible enhancements include:

   - Advanced **geo-fencing** to suggest nearby grounds automatically
   - Integration of **Augmented Reality (AR)** for virtual walkthroughs of cricket nets, futsal courts, or basketball gyms
   - Real-time traffic estimation to calculate estimated travel time
   - Heatmaps showing peak and off-peak hours for each ground

   These improvements will allow users to make better decisions and increase their confidence before booking a facility.

3. **Introduce a Loyalty Program:**

A dedicated loyalty program can play a major role in user retention, especially among frequent players, sports academies, and training groups. The program may include:

- Reward points per booking
- Exclusive discounts for regular users
- Membership tiers (Silver, Gold, Platinum)
- Seasonal offers for high-demand sports

Such incentives encourage repeated bookings and strengthen customer engagement with the platform

4. **Expand Market Reach:**

Currently, the Groundify platform primarily targets Pakistani users. However, indoor sports and recreational activities have strong demand in various regions including:

- Middle East (UAE, Qatar, Saudi Arabia)
- Southeast Asia (Malaysia, Indonesia, Singapore)
- South Asia (Sri Lanka, Bangladesh)

To support international expansion, the system must incorporate:

- Multi-language support
- Region-specific pricing and taxation
- Local currency support
- Integration with regional payment gateways
- Localization of sports types, ground categories, and booking rules

Expanding globally would significantly increase the platform's market potential and revenue streams.

6.3 Final Thoughts

The development of the **"Groundify" Indoor Grounds Booking System** represents a forward-thinking and highly practical solution to the increasing demand for accessible, flexible, and well-managed indoor sports facilities. As sports participation continues to grow across Pakistan—particularly among youth, fitness groups, and recreational athletes—the need for a streamlined digital platform for booking indoor grounds has become more evident than ever.

The long-term success of Groundify will depend on its ability to continuously evolve with user expectations and industry trends. Regular user feedback, timely updates, and ongoing feature enhancements will be essential for maintaining the platform's competitiveness. By consistently integrating new technologies, improving usability, and expanding service offerings, **Groundify has the potential to become the leading digital platform for indoor sports facility booking in the region**.

With its strong technical foundation, scalable architecture, and user-centric design, Groundify is well-positioned to significantly transform how sports enthusiasts discover, reserve, and enjoy indoor sports spaces. Through continuous innovation and strategic upgrades, the platform can achieve sustained growth and establish itself as a benchmark solution in the indoor sports industry.

6.4 Future Prospects

Looking ahead, **"Groundify" can significantly expand its reach and enhance its features to cater to a much wider sports audience.** The platform's true potential lies not only in providing a convenient booking solution for indoor grounds but also in building a complete digital ecosystem where ground Owners, users, trainers, academies, and service providers can seamlessly interact.

Future development may include forming partnerships with **sports academies**, **equipment rental services**, **coaching providers**, **nutrition brands**, and **event organizers** to offer users a more comprehensive indoor sports experience. Such integrations would allow Groundify to evolve from a simple booking application into a full-service sports engagement platform.

Additionally, the adoption of emerging technologies such as **Blockchain** can offer secure, immutable, and transparent booking records, digital contracts, and payment verification. This would increase user trust, especially for high-value bookings, tournaments, and long-term facility agreements. Advanced analytics, AI-driven recommendations, automated scheduling optimization, and region-based sports trends can also be implemented to take Groundify to the next level of digital transformation.

Through continuous innovation, strategic partnerships, and the integration of advanced technologies, **Groundify has the potential to become a leading platform in the indoor sports industry,** both locally and internationally**.**

REFERENCES

Journal Papers:

[1] Performance and stability Comparison of React and Flutter: Cross-platform Application Development, Kamal Kishore; Shanu Khare; Vaibhav Uniyal; Sahil Verma

[2] State Management Analyses of the Flutter Application, Hoang, Ly (2019)

[3] Technology-Based Scheduling and Event Management in Sports, Sangramsing S. More Shripatrao Chougule Arts & Science College, Malwadi-Kotoli

[4] AI Chatbot integration in SME marketing platforms: Improving customer interaction and service efficiency, Wagobera Edgar Kedi

[5] A Clean Approach to Flutter Development through the Flutter Clean Architecture Package, Shady Boukhary
Department of Computer Science, Midwestern State University, Wichita Falls, TX

Conference Papers:

[6] Introduction to UI/UX Design: Key Concepts and Principles, Nasrullah Hamidli

[7] Software Engineering Research at the International Conference on Software Engineering in 2016. Christopher Theisen,

Technical Reports and Theses:

[8] Potentials and Challenges of Chatbot-Supported Thesis Writing: An Autoethnography, Nicolas Schwenke, Heinrich Söbke and Eckhard Kraft

[10] Developing a multi-platform, notification-centric mobile application with Firebase Cloud Messaging integration Valkamo, Eetu (2026)

ORIGINALITY REPORT

10%	8%	3%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	1%
2	Submitted to Hotelschool Den Haag Student Paper	1%
3	www.coursehero.com Internet Source	1%
4	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
5	Submitted to University of New South Wales Student Paper	<1%
6	www.fdaytalk.com Internet Source	<1%
7	Submitted to University of East London Student Paper	<1%
8	Submitted to Central Queensland University Student Paper	<1%
9	hdl.handle.net Internet Source	<1%
10	Submitted to Macquarie University Student Paper	<1%

25	www.nulledfrm.com Internet Source	<1%
26	www.researchgate.net Internet Source	<1%
27	jacksonlumber.cld.bz Internet Source	<1%
28	www.amigoways.com Internet Source	<1%
29	Submitted to Universiti Teknologi MARA Student Paper	<1%
30	au.freedissertation.com Internet Source	<1%
31	calhoun.nps.edu Internet Source	<1%
32	www.logzerotechnologies.com Internet Source	<1%
33	essaywritingforkids150.blogspot.com Internet Source	<1%
34	lib.buet.ac.bd:8080 Internet Source	<1%
35	monimega.com Internet Source	<1%
36	uu.diva-portal.org Internet Source	<1%
37	www.aihirely.com Internet Source	<1%

38	Sumit Kalaria. "Securing Android Apps - A Practical Approach for Secure Development", CRC Press, 2025 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		