

Software Requirements Specification 2: BookForMe

Muhammad Ahmad Humayun Hanif, Jazib Waqas, Taha Hunaid Ali, Muhammad Taqi

November 8, 2025

Contents

1	System Block Diagram	2
1.1	System Block Diagram	2
1.2	Description of Components	2
2	Project Plan	4
2.1	Objectives	4
2.2	Team Roles and Responsibilities	4
2.3	Monthly Timeline	5
2.4	Risks and Mitigation Strategies	6
3	Wireframes	8
3.1	Overview	8
3.2	Customer Registration & Authentication Flow	8
3.3	Customer Discovery & Booking Flow	11
3.4	AI Chatbot & Social Hub Flows	16
3.5	Notifications & Customer Profile	16
3.6	Vendor Registration & Management Flow	20
3.7	Integration Flows (WhatsApp & Google Sheets)	25

1 System Block Diagram

1.1 System Block Diagram

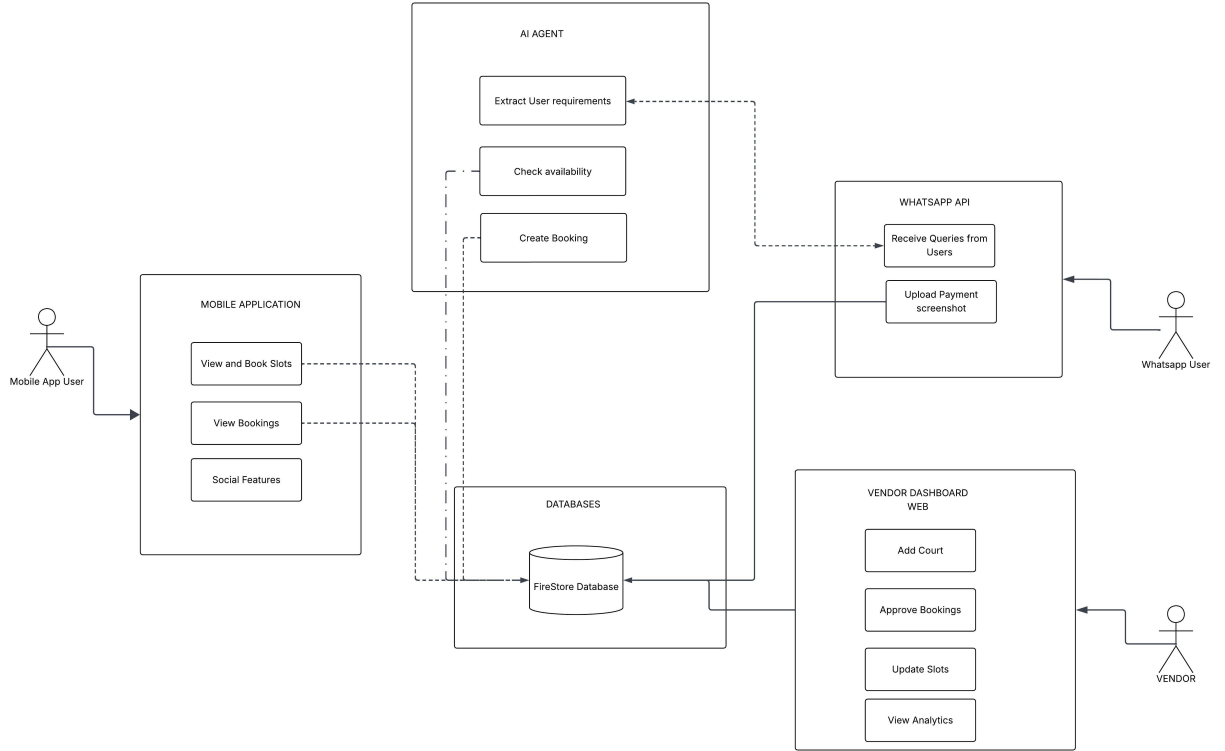


Figure 1: System Block Diagram for BookForMe Platform

1.2 Description of Components

The BookForMe system follows a modular, service-oriented architecture composed of two primary layers: **Frontend Layer** and a **Backend/API Layer**.

- **User Mobile Application (Frontend)**

Developed using **React Native**, this component is the native mobile app for customers. It allows users to browse vendors, book services, and interact with all social features (profiles, chat, forums, matchmaking). It communicates with the backend via RESTful APIs.

- **Vendor Dashboard (Frontend)**

A secure administrative mobile app, also built with **React Native**. It allows vendors to manage their business profile, view their consolidated calendar, manually **approve pending payments**, and link their external **WhatsApp API** account.

- **AI Agent Backend**

The core intelligent engine of the platform, built in **Python**. This service is responsible for:

- Managing the **WhatsApp Webhook** to receive and send messages/images.
 - Utilizing a **trained NLU model** for bilingual Natural Language Understanding (NLU).
 - Executing the core **agentic logic** (decision-making).
 - Enforcing **concurrency control** (via Firestore Transactions) for all bookings.
- **Cloud Firestore Database**

Acts as the central NoSQL data repository. It stores all persistent data: user/vendor profiles, service listings, real-time availability, bookings, social data (friends, forum posts), and the **conversation state** for the AI Agent. Its real-time capabilities are leveraged for instant updates.
 - **WhatsApp Business API**

The primary communication channel for the AI Agent. It enables two-way, asynchronous conversation with users on the vendor’s behalf. The AI Agent receives **webhooks** (for text and images) and sends replies via this API.
 - **Authentication and Security Layer**

Firebase Authentication is used to manage all user and vendor identities. **OAuth 2.0** is used for the vendor-side WhatsApp API integration. All communication between components is secured via HTTPS/TLS.

Overall, the architecture enforces modularity, concurrency control (via Firestore Transactions), and horizontal scalability via cloud deployment (e.g., Render, Firebase Cloud Functions).

2 Project Plan

2.1 Objectives

The overarching objectives of the BookForMe project are to:

1. Develop a **centralized booking platform** connecting users and vendors across informal service domains.
2. Implement an intelligent **AI Receptionist** (vendor-side) capable of processing WhatsApp-based booking requests using Natural Language Understanding (NLU).
3. Implement a **Conversational Search Agent** (user-side) to provide an advanced, query-based interface for filtering and finding vendors.
4. Foster an integrated **social community** by implementing features for user profiles, friends, direct messaging, forums, and a matchmaking/ranking system.
5. Design a scalable, secure backend integrated with **Firestore** and the **WhatsApp Business API**.
6. Deliver a seamless, user-friendly **React Native** frontend (User App & Vendor Dashboard) ensuring accessibility, reliability, and real-time synchronization.

2.2 Team Roles and Responsibilities

- **M. Ahmad Hanif (Project Lead)** — Leads system architecture and AI Receptionist integration with Firestore. Responsible for development and functioning of the NLU pipeline. Oversees project roadmap, documentation, and delivery milestones.
- **Jazib Waqas (Backend Engineer)** — Responsible for backend architecture, database integration, and implementation of booking, vendor, and authentication APIs. Leads WhatsApp API integration and ensures secure communication between the conversational agent and backend services.
- **Taha Hunaid Ali (Frontend Developer)** — Develops and maintains user-facing and vendor dashboards using **React Native**. Works on real-time interface features such as chat, leaderboard, and vendor analytics visualization. Coordinates frontend-backend linkage and deployment readiness.
- **Muhammad Taqi (AI Engineer)** — Works on the design and development of the NLU pipeline, focusing on intent detection, slot extraction, and model fine-tuning. Collaborates on AI Receptionist integration and ensures accurate handling of bilingual booking queries.

2.3 Monthly Timeline

Month	Tasks	Deliverables
November: Foundation Phase	<ul style="list-style-type: none"> • Finalize <i>SRS 1</i>, <i>SRS 2</i> & <i>SDS</i> and complete the literature review. • Build an initial dataset for booking requests and vendor replies. • Design and implement a low-fidelity UI prototype for early validation. • Set up GitHub repositories and development environments. 	<ul style="list-style-type: none"> • Approved SRS Document. • Preliminary Bilingual Dataset. • Prototype User Interface (Login, Vendor Discovery).
December: Baseline Model and Backend Setup	<ul style="list-style-type: none"> • Train baseline NLU models for intent detection and slot extraction. • Implement backend integrations using Firestore. 	<ul style="list-style-type: none"> • Baseline NLU Model. • Defined Database Schema. • Minimal Working Booking System (Storage & Retrieval).
January: AI Receptionist and WhatsApp Integration	<ul style="list-style-type: none"> • Fine-tune the NLU pipeline for improved bilingual performance. • Implement the AI Receptionist for WhatsApp-based booking processing. • Introduce concurrency control, safety checks, and backend error handling. • Refine frontend booking and profile management interfaces. 	<ul style="list-style-type: none"> • Functional AI Receptionist for End-to-End Conversational Booking. • Working WhatsApp-to-Database Data Flow. • Enhanced User Dashboard with Profile Management.
February: Vendor Tools and Social Features	<ul style="list-style-type: none"> • Develop vendor-side tools: analytics dashboard, calendar availability, and payment confirmation. • Implement social features: match-making, chat, and leaderboard. • Optimize bilingual model responses through retraining and error analysis. 	<ul style="list-style-type: none"> • Fully Functional Vendor Dashboard. • Active Chat, Matchmaking, and Leaderboard Features. • Optimized Bilingual NLU Model.

Month	Work (Objectives)	Deliverables
March: Integration, Optimization, and Testing	<ul style="list-style-type: none"> • Integrate user and vendor interfaces with backend services. • Implement Google Sheets API for vendor booking synchronization. • Conduct unit, integration, and system testing; optimize for performance. • Perform UX testing and benchmarking. 	<ul style="list-style-type: none"> • Fully Integrated BookForMe Platform. • Stable End-to-End Workflow (User ↔ AI Receptionist ↔ Vendor). • Detailed Test Reports and Benchmark Metrics.
April: Final Deployment and Submission	<ul style="list-style-type: none"> • Conduct pilot testing with selected vendors. • Deploy production-ready system on Firebase with secured integrations. • Prepare final documentation, technical report, and presentation. 	<ul style="list-style-type: none"> • Production-Ready Deployment. • Final Project Report and Presentation Deck. • Complete Submission Package for Evaluation.

Table 1: Project Timeline and Deliverables Overview

2.4 Risks and Mitigation Strategies

The following table outlines the major technical, operational, and organizational risks identified during the development of the *BookForMe* system, along with corresponding mitigation strategies and recommended timelines.

Risk	Mitigation Strategy	Timeline
API Rate Limits (WhatsApp)	Implement exponential backoff and retry policies; throttle outbound requests; batch operations where possible; cache responses; request higher quotas in advance.	Ongoing
Data Consistency and Race Conditions (Concurrent Bookings)	Use Firestore transactions and idempotency keys; apply optimistic locking; implement short-lived reservation holds to avoid double-booking.	Jan-Feb
Bilingual NLU Failure (English-Urdu / Roman Urdu)	Collect diverse bilingual data; use transfer learning; continuous retraining with error logs; add fallback rule-based handlers.	Nov-Dec
Security Breaches / Key Exposure	Store credentials in Google Secret Manager; rotate keys; restrict IAM roles; enforce HTTPS and TLS; audit repository commits for sensitive data.	Ongoing
System Latency / NLU Inference Delay	Profile model inference; cache frequent queries; use async jobs for long responses; horizontally scale backend on Firebase Cloud Functions.	Jan-Feb
Cloud Cost Overrun (Firestore Reads / Functions Invocations)	Estimate cost per call; use batched reads; denormalize where appropriate; add billing alerts and caps in Firebase.	Ongoing
UX / Usability Shortcomings	Conduct iterative usability testing; analyze clickstream and drop-off metrics; refine prototype flows accordingly.	Mar-Apr

Table 2: Project Risks, Mitigation Strategies, and Recommended Timelines

3 Wireframes

3.1 Overview

These mockups are created using HTML/CSS and serve as a blueprint for frontend implementation. They are grouped by user flow and include the customer registration, booking, social, vendor management, and integration processes.

3.2 Customer Registration & Authentication Flow

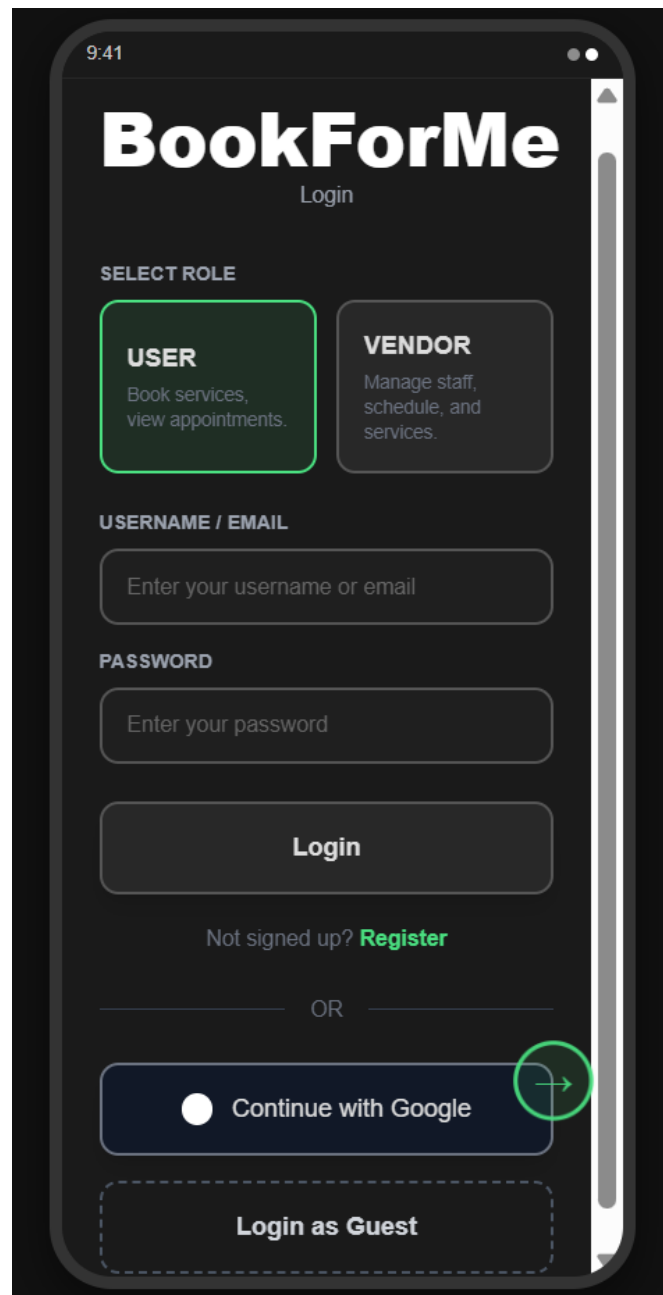


Figure 2: Customer Registration Flow (1/3): **Login / Role Selection**. Unified sign-in with role toggle and quick auth options.

9:41

← **Create Account** Step 2 of 3
Customer onboarding

BASIC DETAILS

First Name Last Name

Email Address

Mobile Number

Password

PREFERENCES

Preferred Sports

Location

AGREEMENTS

☐ I agree to the Terms & Privacy Policy.

← **Create Account** →

Home Chat Social Alerts

Figure 3: Customer Registration Flow (2/3): **Customer Registration Form.** Collects personal details, contact info, and preferences.

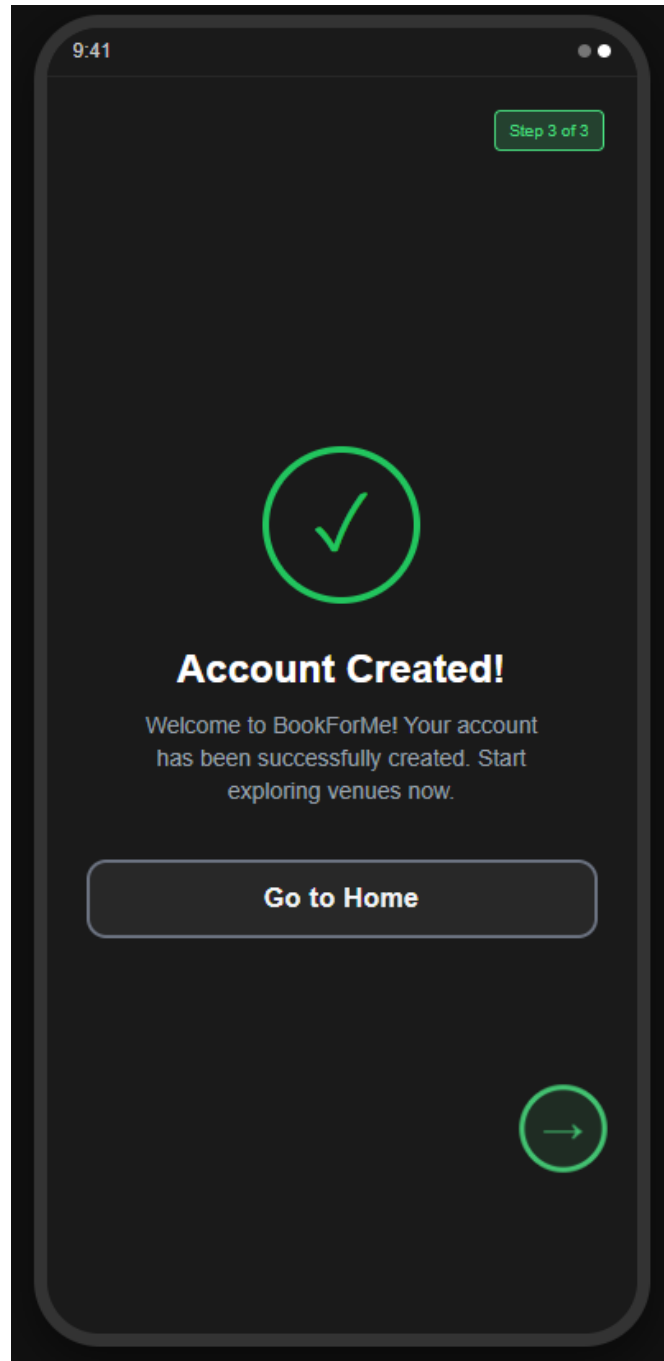


Figure 4: Customer Registration Flow (3/3): **Registration Success.** Account creation confirmation with navigation options.

3.3 Customer Discovery & Booking Flow

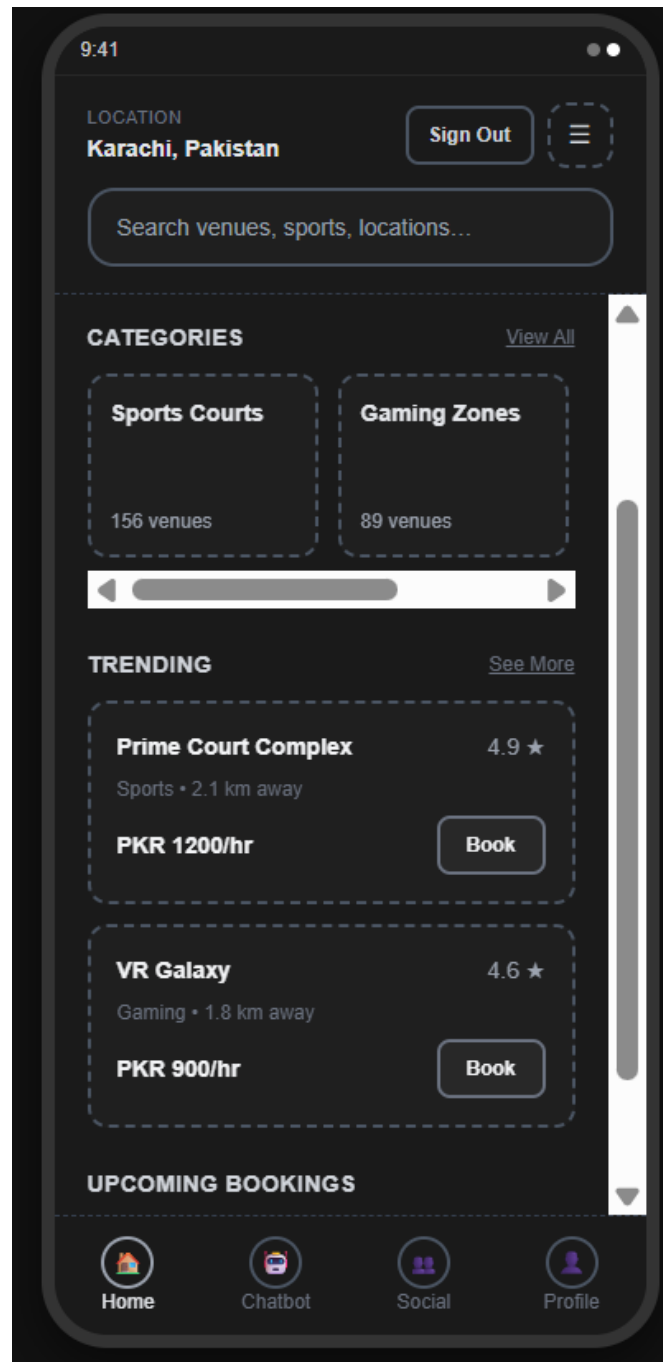


Figure 5: Customer Booking Flow (1/4): **Customer Home.** Discovery hub with search, categories, and trending venues.



Figure 6: Customer Booking Flow (2/4): **Category Listing**. Filterable list of venues with price and amenity filters.

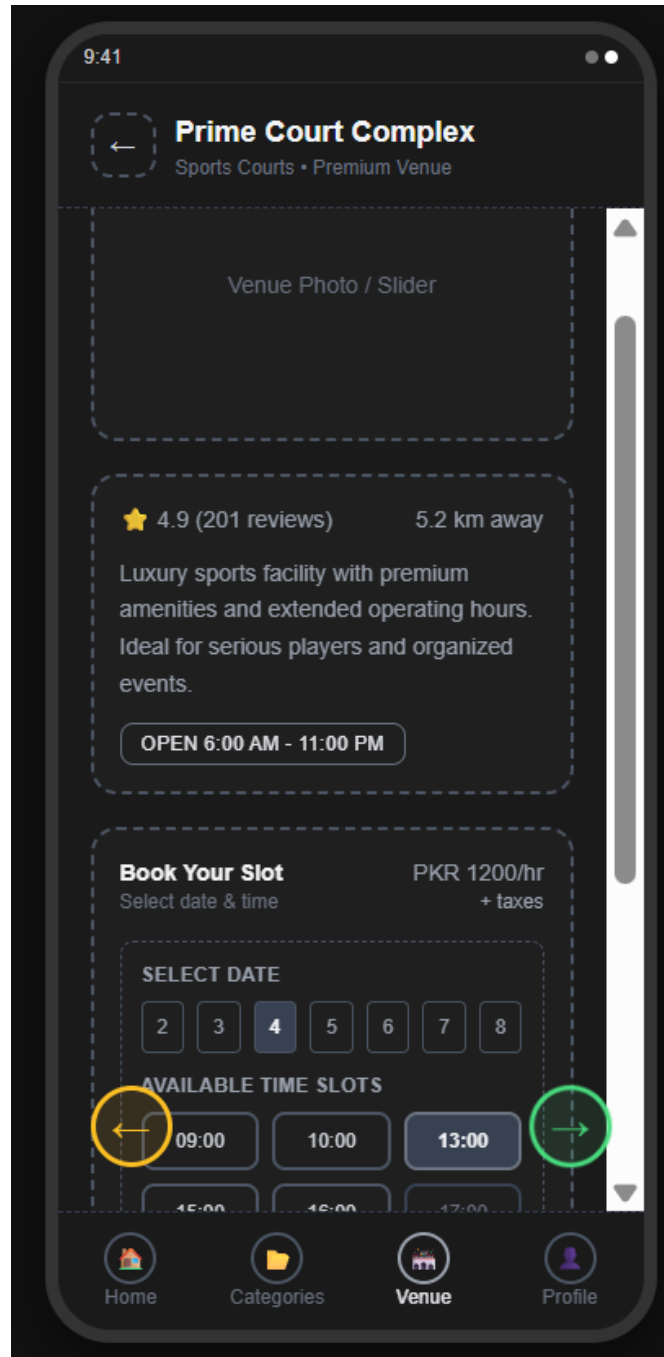


Figure 7: Customer Booking Flow (3/4): **Vendor Detail**. Individual venue profile with reviews, amenities, and slot selection.

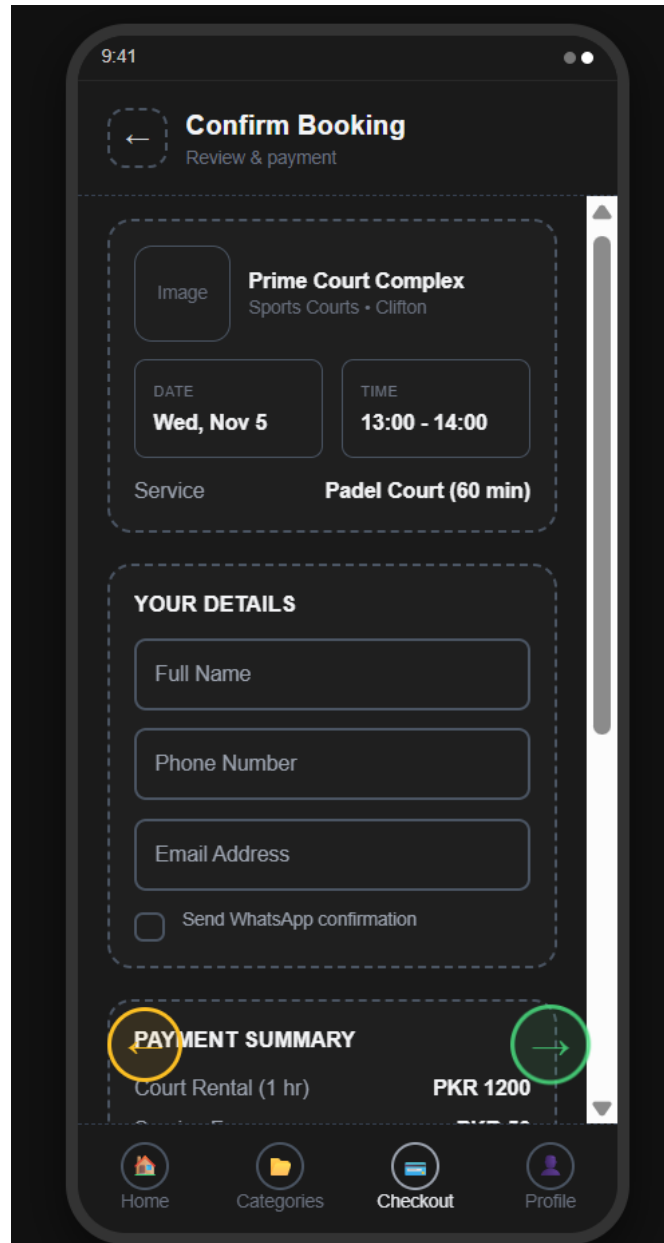


Figure 8: Customer Booking Flow (4/4): **Booking & Payment**. Booking summary, customer info entry, and payment selection.

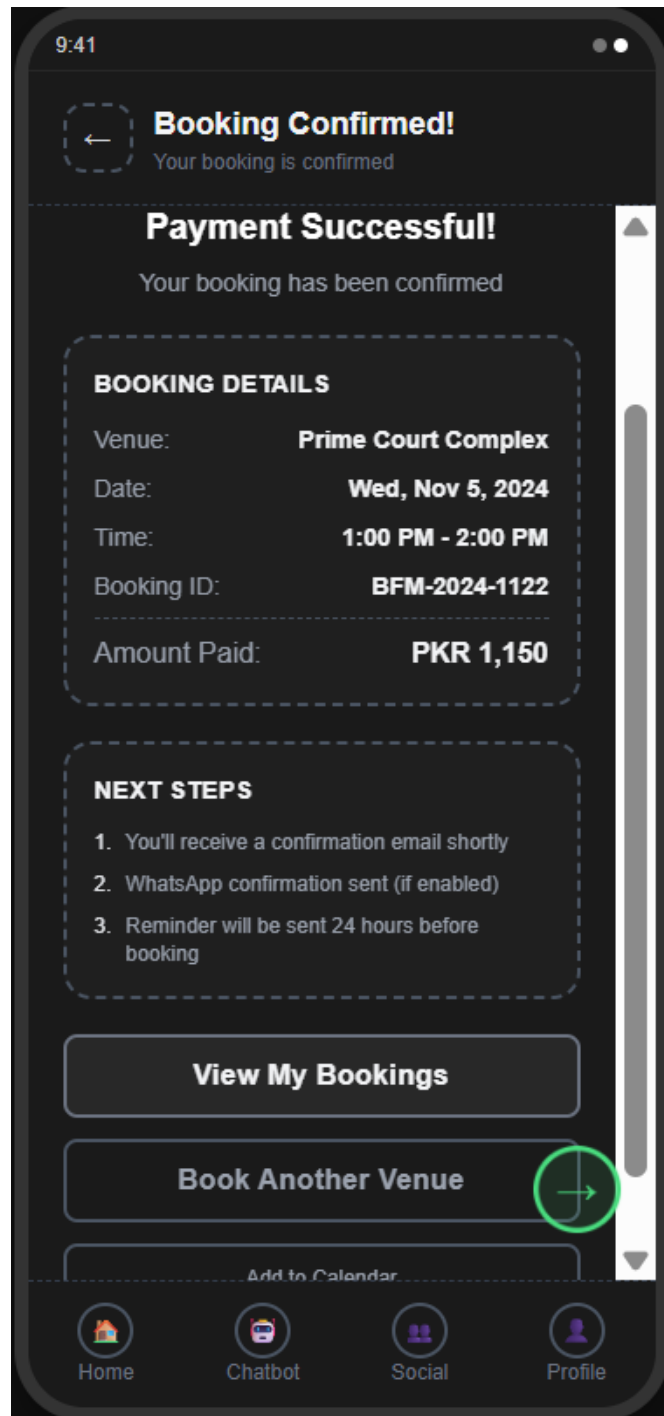


Figure 9: Payment Success (1/1): **Booking Confirmation.** Final success screen with receipt, booking details, and next steps.

3.4 AI Chatbot & Social Hub Flows

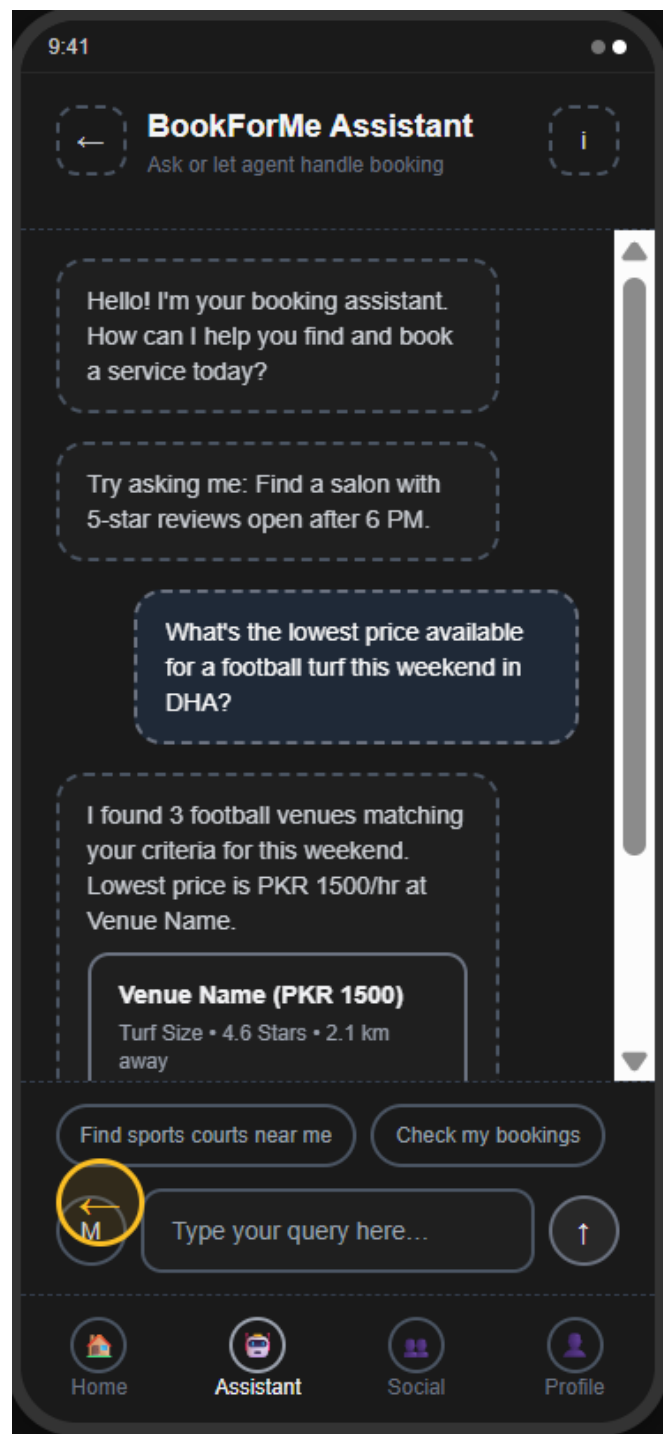


Figure 10: **AI Chatbot:** Conversational assistant for recommendations, bookings, and FAQs.

3.5 Notifications & Customer Profile

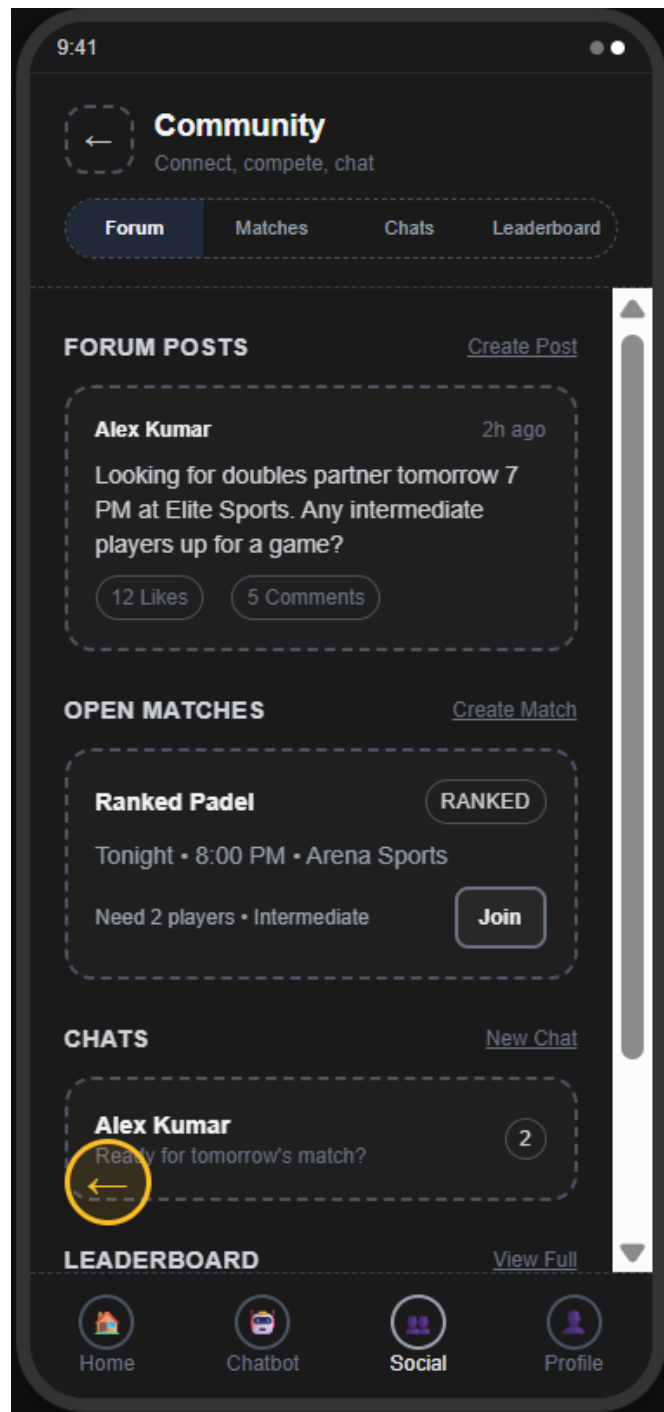


Figure 11: **Social Hub:** Tabbed community space for Forum, Open Matches, Chats, and Leaderboard.

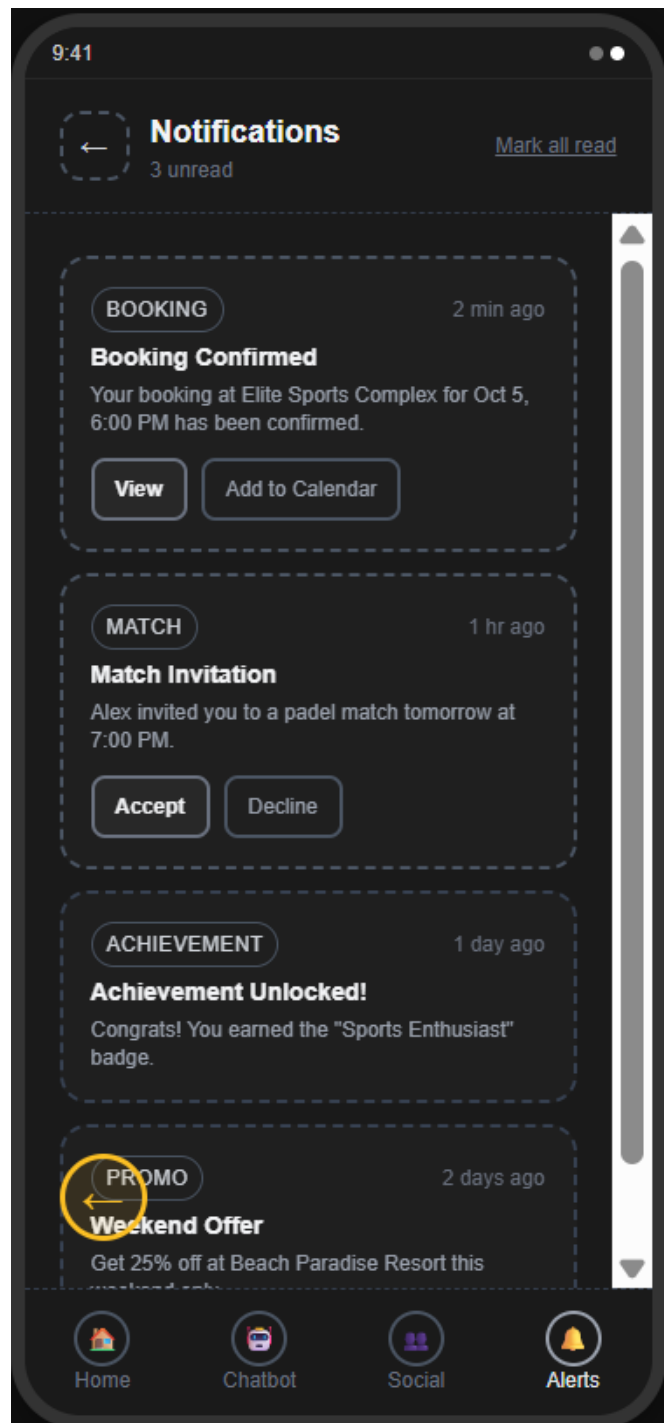


Figure 12: **Notifications:** Feed of booking, social, and promotional alerts.

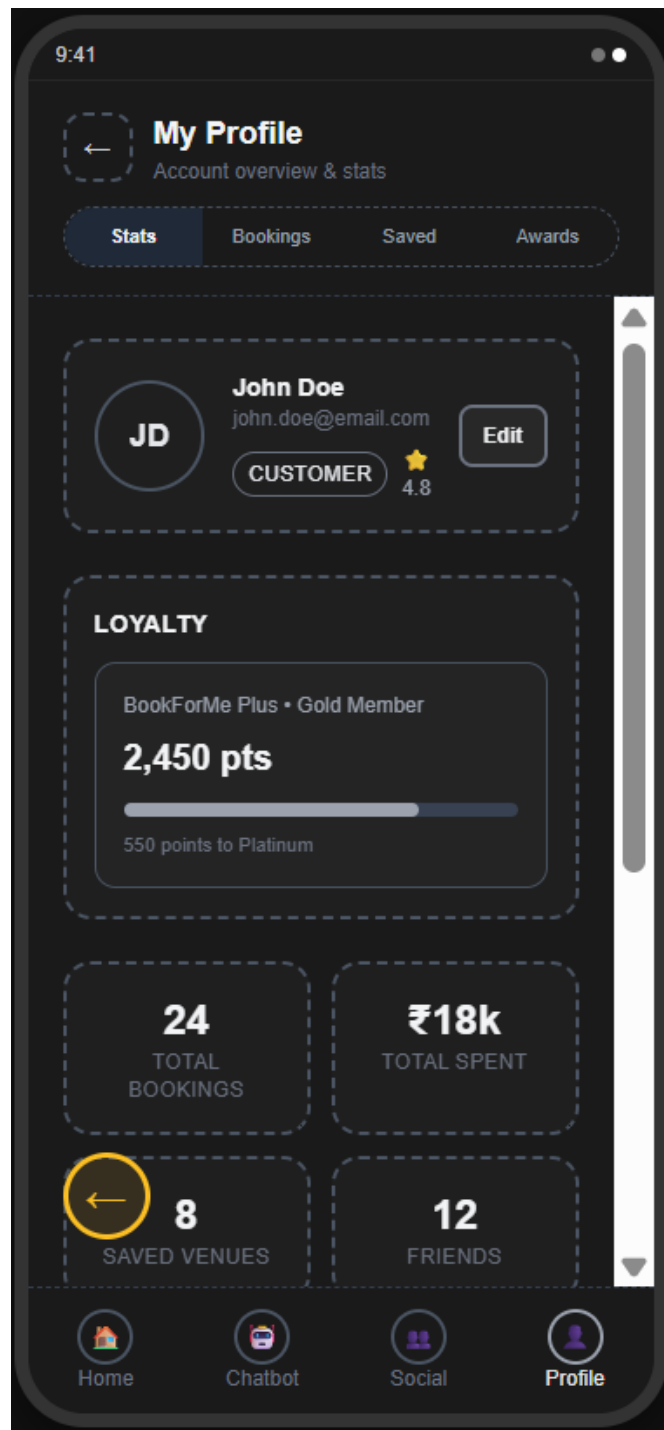


Figure 13: **Customer Profile:** Account overview with stats, bookings, saved venues, and settings.

3.6 Vendor Registration & Management Flow

The screenshot shows a mobile application interface for 'Vendor Signup'. At the top, the status bar displays '9:41'. Below it, a navigation bar contains a back arrow icon and the title 'Vendor Signup' with the subtitle 'Business onboarding'. The main content area is a form with three sections: 'BUSINESS INFO' containing fields for 'Business Name', 'Category', 'City / Area', and 'Full Address'; 'CONTACT' containing fields for 'Owner Name', 'Phone Number', and 'Email Address'; and 'OPERATIONS' containing 'Open Time' and 'Close Time' fields. A yellow circle highlights a back arrow icon to the left of the 'Open Time' field, and a green circle highlights a forward arrow icon to the right of the 'Close Time' field. A vertical scrollbar is visible on the right side of the form. At the bottom, a navigation bar features four icons: 'Home' (house), 'Dashboard' (chart), 'Calendar' (calendar), and 'Settings' (gear).

Figure 14: Vendor Flow (1/5): **Vendor Registration.** Form for business information, categories, and verification.

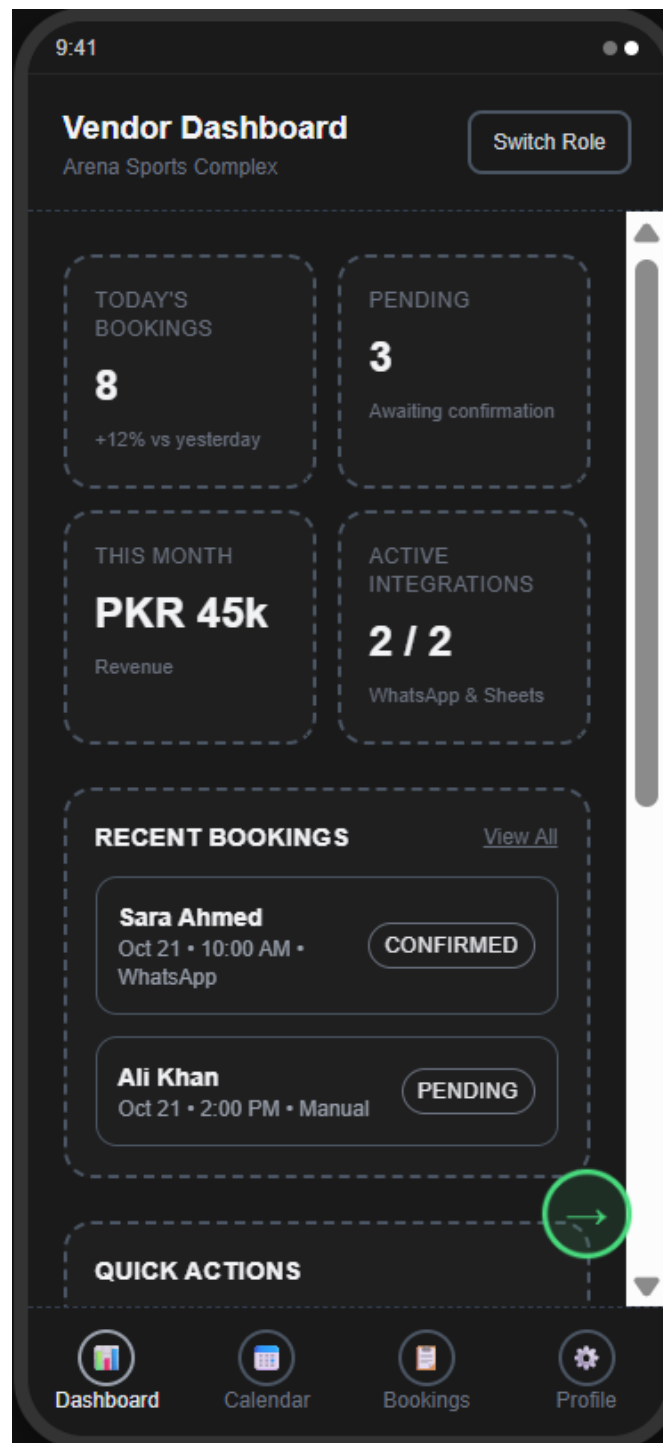


Figure 15: Vendor Flow (2/5): **Vendor Dashboard**. At-a-glance analytics, recent activity, and integration status.

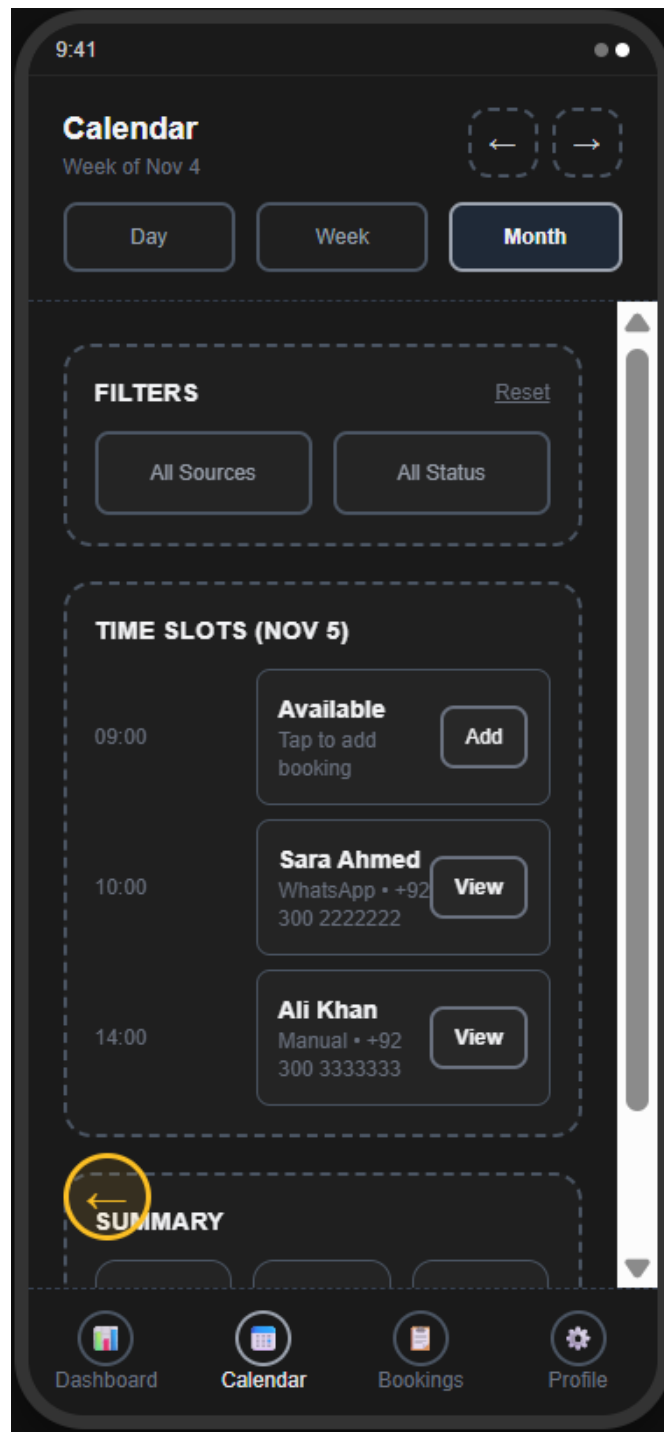


Figure 16: Vendor Flow (3/5): **Vendor Calendar.** Calendar view of bookings with source/status filters and slot management.

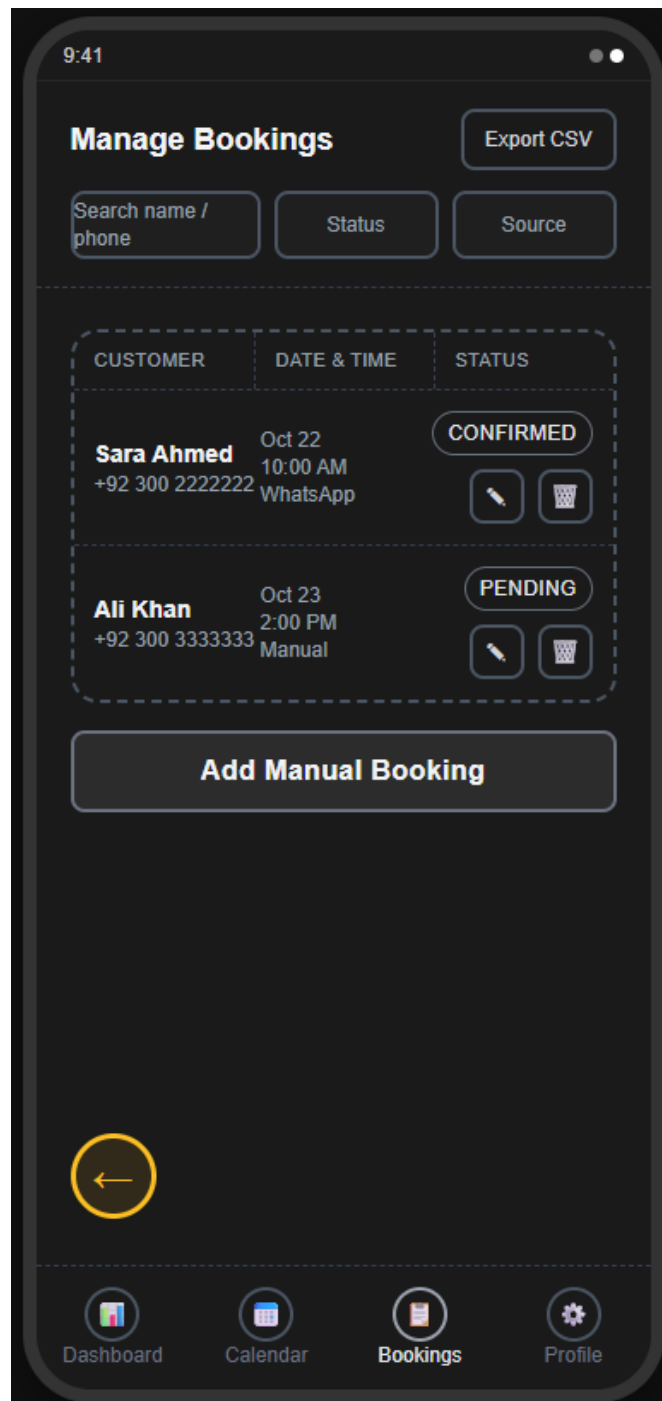


Figure 17: Vendor Flow (4/5): **Manage Bookings**. Tabular management of all bookings with search, filters, and manual entry.

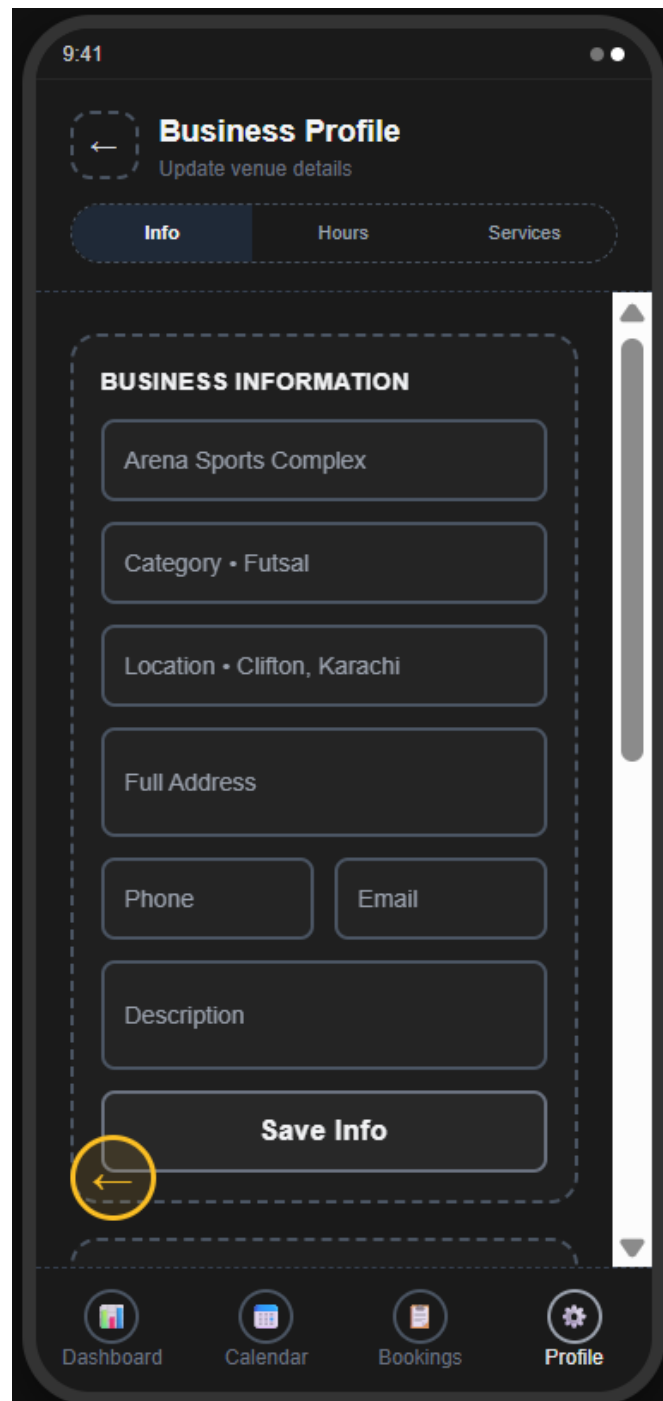


Figure 18: Vendor Flow (5/5): **Business Profile**. Configuration for business info, hours, services, pricing, and media.

3.7 Integration Flows (WhatsApp & Google Sheets)

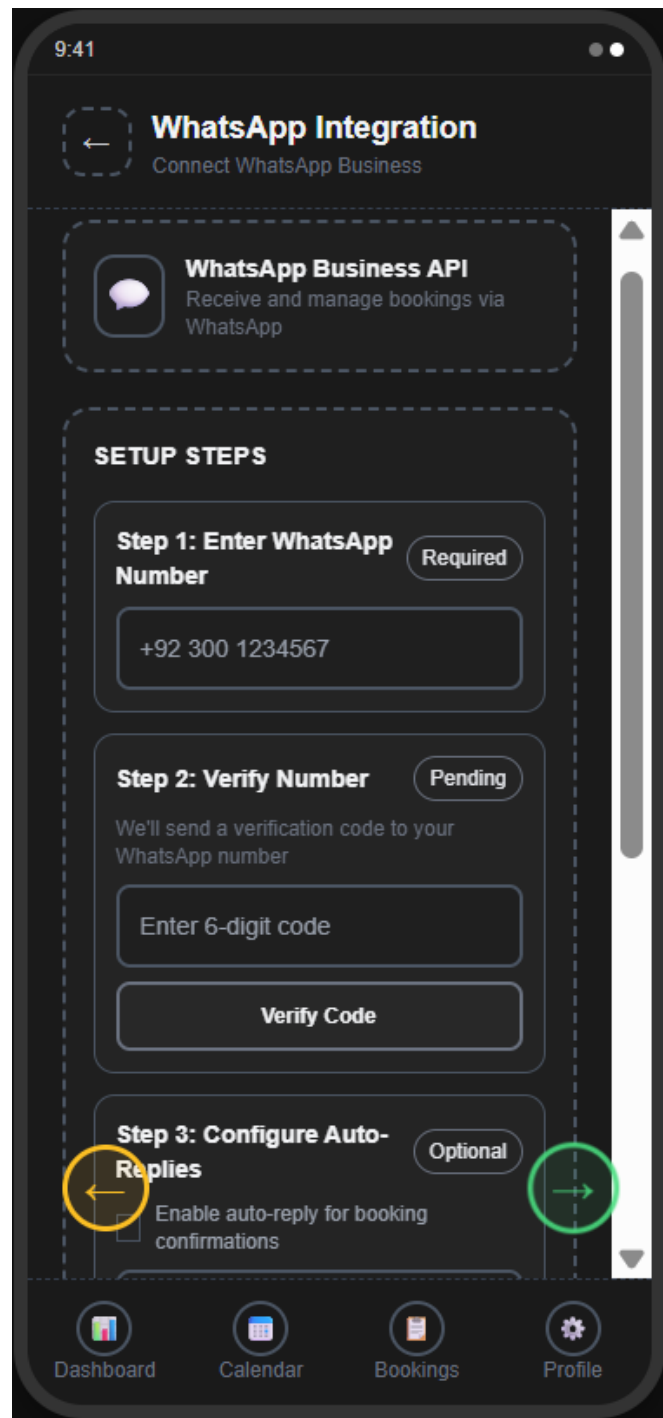


Figure 19: Integration Flow (1/2): **WhatsApp Integration Setup**. Connecting WhatsApp Business API and configuring auto-replies.

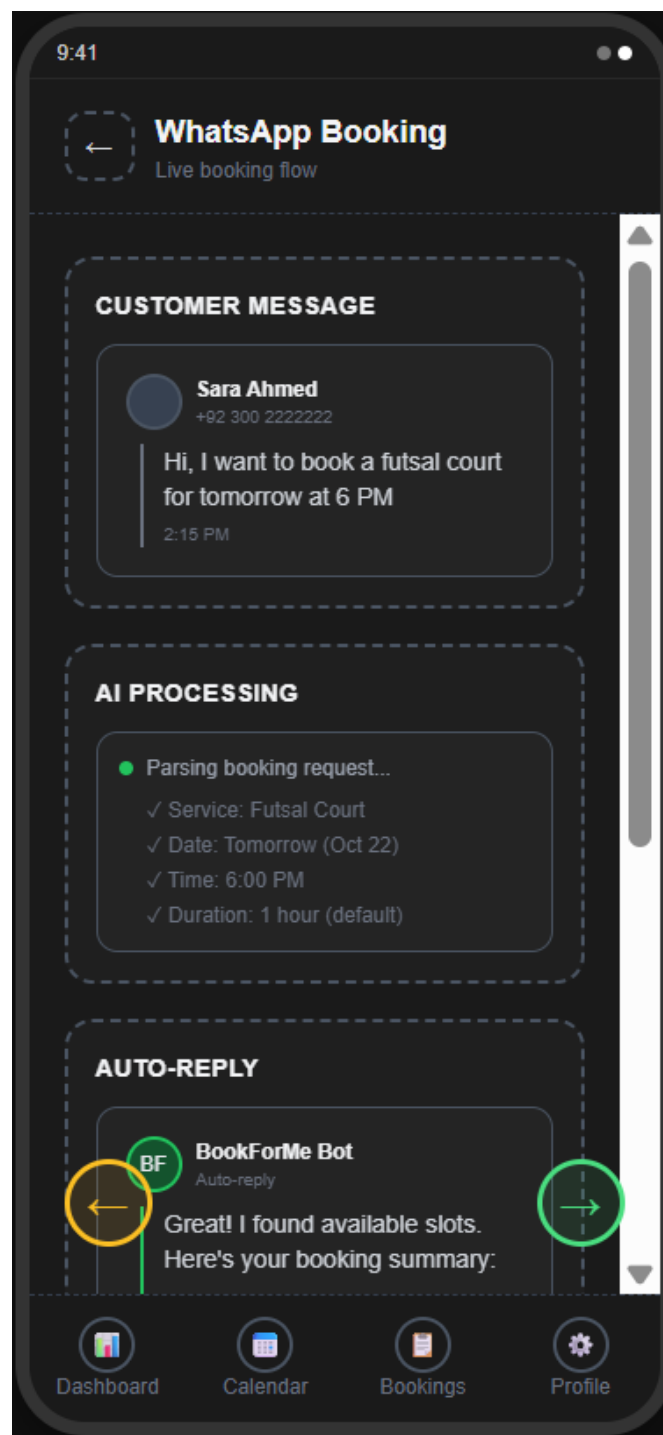


Figure 20: Integration Flow (2/2): **WhatsApp Booking Flow**. AI parsing a customer message and creating a booking.