Community-Based Local Services Platform

Software Requirement Specification (SRS)

1. Introduction

Project Title: Community-Based Local Services Platform
Purpose: To create a web-based platform that connects users with local service providers (plumbers, electricians, tutors, cleaners, etc.) enabling booking, payments, communication, and reviews.
Scope:
- Users: Search, book, pay, chat, review.
- Providers: Register, set availability, manage bookings, receive payments.
- Admin: Approve providers, resolve disputes, analytics.

2. Functional Requirements

- User: Register/login, Search/filter providers, Booking system, Chat, Payments, Reviews, Booking history.
- Provider: Manage profile, Availability, Bookings, Chat, Earnings dashboard.
- Admin: Approve/reject providers, Monitor activity, Resolve disputes, Analytics dashboard.

3. Non-Functional Requirements

- Performance: Support 100+ concurrent users.
- Security: JWT, password hashing, encrypted payments.
- Scalability: Modular REST APIs.
- Usability: Responsive UI, dark mode.

4. System Design Overview

Frontend: React.js + Tailwind CSS
Backend: Node.js + Express
Database: MongoDB
Authentication: JWT
Payments: Razorpay/Stripe
Notifications: Firebase/Email
Deployment: Render/Netlify

UML Diagrams

1. Use Case Diagram (Textual Description)

Actors: User, Provider, Admin
Use Cases: User → Search, Book, Chat, Pay, Review; Provider → Manage profile, Accept bookings, Chat, Track earnings; Admin → Approve providers, Monitor platform, Resolve disputes

2. Class Diagram (Simplified)

User(userId, name, email, password, role)
Provider(providerId, services[], pricing, availability, rating)
Booking(bookingId, userId, providerId, date, slot, status)
Payment(paymentId, bookingId, amount, method, status)
Review(reviewId, bookingId, rating, comment)

3. Sequence Diagram (Booking Flow)

User searches → System shows providers → User selects provider → Booking request → Provider confirms → Payment → Booking created → Notifications sent

ER Diagram (Entity Relationships)

Entities:
- Users (userId)
- Providers (providerId)
- Bookings (bookingId)
- Payments (paymentId)
- Chats (chatId)
- Reviews (reviewId)

Relationships:
- Users ↔ Providers (1-1)
- Users ↔ Bookings (1-M)
- Providers ↔ Bookings (1-M)
- Bookings ↔ Payments (1-1)
- Bookings ↔ Chats (1-1)
- Bookings ↔ Reviews (1-1)