FixItNow – Neighborhood Service & Repair Marketplace

Project Statement

FixItNow is a full-stack neighborhood service platform connecting residents with nearby electricians, plumbers, carpenters, appliance repairers, and other service professionals. Residents can post service requests, browse verified service providers, and book them instantly. Service providers can manage profiles, receive booking requests, and maintain a service history. An integrated review and rating system ensures trust, and location-based filtering allows users to find the nearest help quickly.

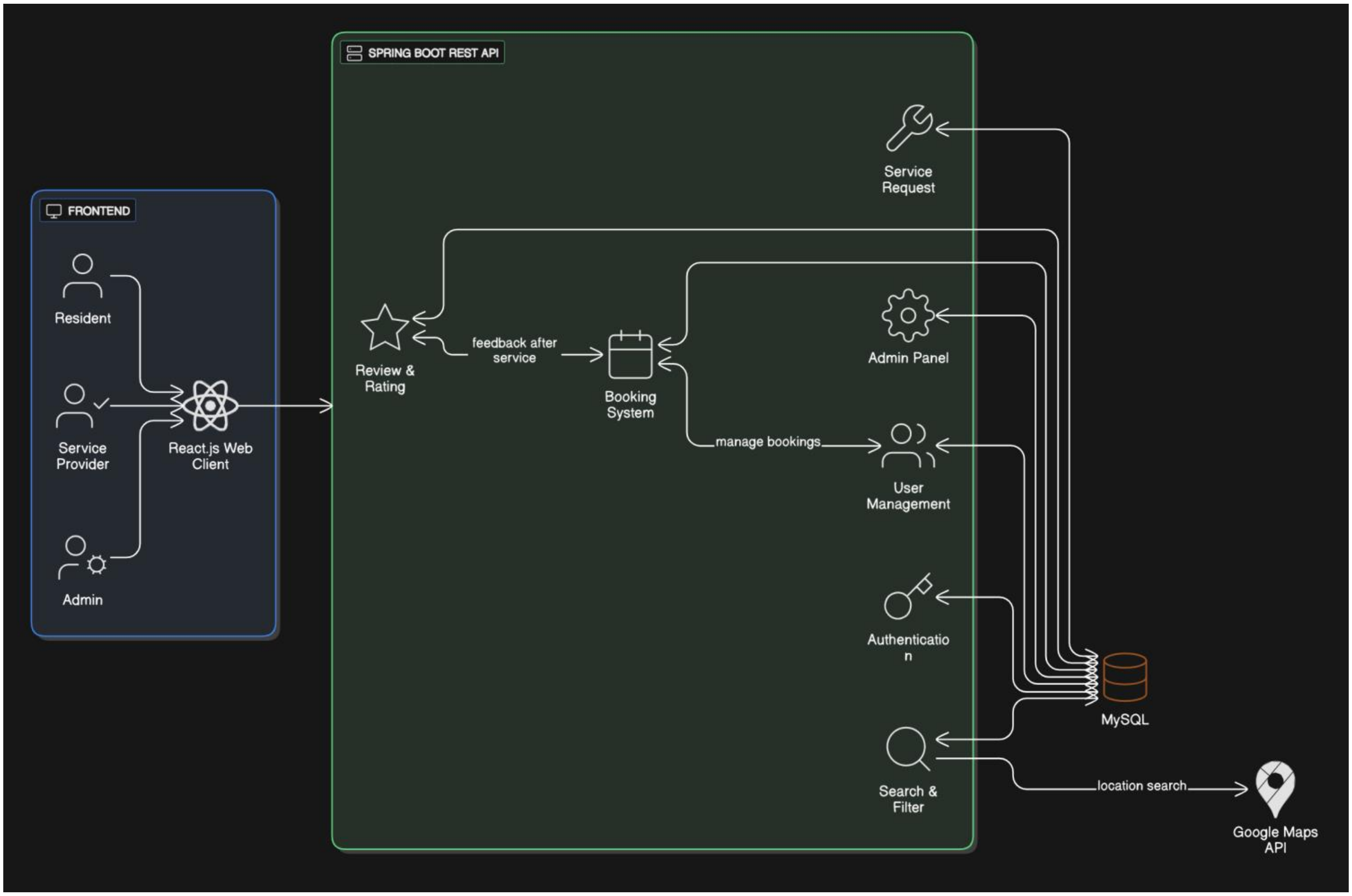
Key Features

- **Location-based search & booking** for electricians, plumbers, carpenters, etc.
- **Service categories & subcategories** for quick filtering
- **Instant booking system** with time slots
- **Service provider profiles** with ratings, reviews, and completed jobs
- **Real-time chat** between customer and provider
- **Booking management dashboard** for both users and providers
- **Admin panel** for user/service verification, dispute management, and analytics

Tech Stack

- **Frontend:** React.js + Tailwind CSS
- **Backend:** Spring Boot (Java)
- **Database:** MySQL
- **Authentication:** JWT (access + refresh tokens)
- **Maps & Location:** Google Maps API + Geolocation API
- **Real-time Messaging:** WebSockets (Spring WebSocket + STOMP)

System Architecture



Modules

- **Module A:** User & Service Provider Management (Registration, Authentication, Roles)
- **Module B:** Service Listing & Location-based Search
- **Module C:** Booking & Scheduling System
- **Module D:** Reviews, Ratings & Chat
- **Module E:** Admin Dashboard (Verification, Dispute Management, Analytics)

8-Week Milestone Plan

Milestone 1: Weeks 1–2 – Authentication & Basic Setup

Week 1:

- Set up project architecture (React frontend + Spring Boot backend)
- Implement JWT authentication (login/register)
- Create user model in DB (roles: customer, provider, admin)

Week 2:

- Role-based routing (separate dashboards for customers, providers, admins)
- Location capture using browser Geolocation API
- Service provider registration form (category, skills, service area)

Expected Output:

- Login/Register UI + backend
- Role-based navigation
- Location-aware registration

Milestone 2: Weeks 3–4 – Service Listings & Search

Week 3:

- Create service category/subcategory structure in DB
- Provider can list services with pricing and availability
- Customer can browse services filtered by category and location

Week 4:

- Map-based service provider search using Google Maps API
- Service detail page with booking form
- Display provider ratings & reviews

Expected Output:

- Functional service listing with filters
- Map search integration

Milestone 3: Weeks 5–6 – Booking & Interaction

Week 5:

- Booking request system with time slots
- Providers can accept/reject bookings
- Booking status updates (Pending, Confirmed, Completed, Cancelled)

Week 6:

- Real-time chat between customer and provider (WebSockets)
- Review and rating system after service completion

Expected Output:

- End-to-end booking flow
- Live chat feature

Milestone 4: Weeks 7–8 – Admin Panel & Final Enhancements

Week 7:

- Admin verification of providers (document upload & approval)
- Dispute resolution workflow (report service, refund handling)

Week 8:

- Analytics dashboard (most booked services, top providers, location trends)
- Deployment & final QA

Expected Output:

- Fully functional admin panel
- Service verification & dispute tools

Expected Project Outcome

By Week 8, FixItNow should allow:

- Secure authentication and role-based dashboards
- Location-based service search and booking
- Real-time provider-customer communication
- Review and rating system
- Admin verification, moderation, and analytics

Database Schema (MySQL)

Users

- id (INT, PK)
- name (VARCHAR)
- email (VARCHAR, UNIQUE)
- password (VARCHAR)
- role (ENUM: 'customer', 'provider', 'admin')

- location (VARCHAR)
- created_at (TIMESTAMP)

Services

- id (INT, PK)
- provider_id (FK to Users.id)
- category (VARCHAR)
- subcategory (VARCHAR)
- description (TEXT)
- price (DECIMAL)
- availability (JSON or VARCHAR)
- location (VARCHAR)
- created_at (TIMESTAMP)

Bookings

- id (INT, PK)
- service_id (FK to Services.id)
- customer_id (FK to Users.id)
- provider_id (FK to Users.id)
- booking_date (DATE)
- time_slot (VARCHAR)
- status (ENUM: 'pending', 'confirmed', 'completed', 'cancelled')
- created_at (TIMESTAMP)

Reviews

- id (INT, PK)
- booking_id (FK to Bookings.id)
- customer_id (FK to Users.id)
- provider_id (FK to Users.id)
- rating (INT)
- comment (TEXT)
- created_at (TIMESTAMP)

Messages

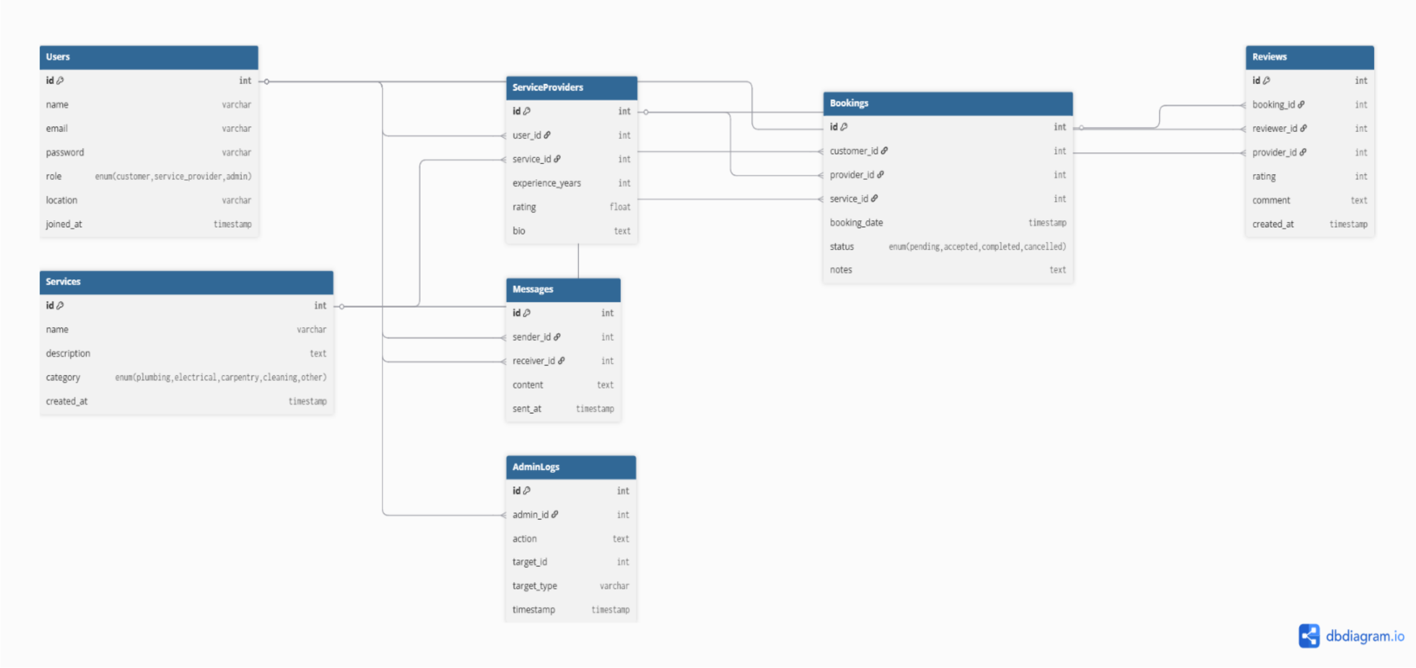
- id (INT, PK)
- sender_id (FK to Users.id)
- receiver_id (FK to Users.id)
- content (TEXT)
- sent_at (TIMESTAMP)

Reports

- id (INT, PK)
- target_type (ENUM: 'booking', 'provider', 'customer')
- target_id (INT)
- reported_by (FK to Users.id)
- reason (TEXT)
- created_at (TIMESTAMP)

AdminLogs

- id (INT, PK)
- admin_id (FK to Users.id)
- action (TEXT)
- target_id (INT)
- target_type (VARCHAR)
- timestamp (TIMESTAMP)


Users
id int
name varchar
email varchar
password varchar
role enum(customer,service_provider,admin)
location varchar
joined_at timestamp
Services
id int
name varchar
description text
category enum(plumbing,electrical,carpentry,cleaning,other)
created_at timestamp
ServiceProviders
id int
user_id int
service_id int
experience_years int
rating float
bio text
Messages
id int
sender_id int
receiver_id int
content text
sent_at timestamp
AdminLogs
id int
admin_id int
action text
target_id int
target_type varchar
timestamp timestamp
Bookings
id int
customer_id int
provider_id int
service_id int
booking_date timestamp
status enum(pending,accepted,completed,cancelled)
notes text
Reviews
id int
booking_id int
reviewer_id int
provider_id int
rating int
comment text
created_at timestamp
dbdiagram.io