Dear sir,

Project Overview

The Navatar system enables remote healthcare consultations by deploying Navatars on hospital floors. Each Navatar is equipped with wheels, a camera, and a microphone, moving like a car under manual control. The system is built using Django (Python) for the backend, MySQL for the database, and CSS/JavaScript for the web interface, with a mobile app for navigation.

Database Schema

1. Company Table:

- First entry is our company (project maker), followed by hospitals.

sql

CREATE TABLE company (

company\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

pincode VARCHAR(10),

country\_code VARCHAR(50)

);

- Example:

- company\_id: 1, name: SangamOne, pincode: 560001, country\_code: IN

- company\_id: 2, name: Gandhi Hospital, pincode: 500001, country\_code: IN

2. Navatar Table:

- Navatars are assigned to floors, and one floor can have multiple Navatars.

sql

CREATE TABLE navatar (

navatar\_id SERIAL PRIMARY KEY,

**company\_id INTEGER NOT NULL,**

navatar\_name VARCHAR(100) NOT NULL,

location VARCHAR(50) NOT NULL,

**status VARCHAR(20) DEFAULT 'idle',**

FOREIGN KEY (company\_id) REFERENCES company(company\_id) ON DELETE CASCADE,

);

**3. Users and Roles Tables:**

- Users can have multiple roles (e.g., doctor, admin, super\_admin, nurse).

sql

CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

user\_name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

mobileno VARCHAR(15),

company\_id INTEGER NOT NULL,

is\_admin BOOLEAN DEFAULT FALSE, is\_doctor BOOLEAN DEFAULT FALSE, is\_superadmin BOOLEAN DEFAULT FALSE, is\_nurse BOOLEAN DEFAULT FALSE,

gender

FOREIGN KEY (company\_id) REFERENCES company(company\_id) ON DELETE CASCADE

);

4. Bookings and Payments Tables:

sql

CREATE TABLE bookings (

booking\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL,

navatar\_id INTEGER NOT NULL,

date DATE NOT NULL,

start\_time TIME NOT NULL,

duration INTEGER NOT NULL,

booking\_status VARCHAR(20) DEFAULT 'pending',

payment\_id SERIAL PRIMARY KEY,

booking\_id INTEGER NOT NULL,

payment\_reference VARCHAR(100) NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

status VARCHAR(20) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE,

FOREIGN KEY (navatar\_id) REFERENCES navatar(navatar\_id) ON DELETE CASCADE,

);

User Roles and Authentication

- Roles: super\_admin (e.g., Kavya, CS at SangamOne), admin (e.g., James, Suresh at Gandhi Hospital), doctor (e.g., Dr. John), nurse. Users can have multiple roles (e.g., Dr. John as doctor and admin).

- Registration: Users register via Google OAuth, and admins (e.g., James) authorize and assign roles.

- Authentication: Handled via Google OAuth for secure login.

- Authorization: Admins must know and manage user roles, ensuring proper access (e.g., doctors book sessions, admins manage Navatars).

Workflow

1. Booking (Web App):

- Doctors (e.g., Dr. John) book a Navatar session via the web app, selecting a floor (e.g., Navatar1 on Ground Floor), date, time, duration, and patient room (e.g., Room 305).

- Payment is processed via Razorpay (UPI, GPay, QR), and the booking is confirmed in the bookings table.

- The backend notifies the doctor’s mobile app via FCM (REST API endpoint: /api/notify-booking/).

2. Navigation (Mobile App):

- The Navatar app receives the notification, displaying the booking details.

- The app fetches the camera , showing the corridor.

- The doctor navigates the Navatar using buttons (left, right, straight, stop), sending commands to

navigate

- The Navatar moves like a car (manual control) to the patient room.