



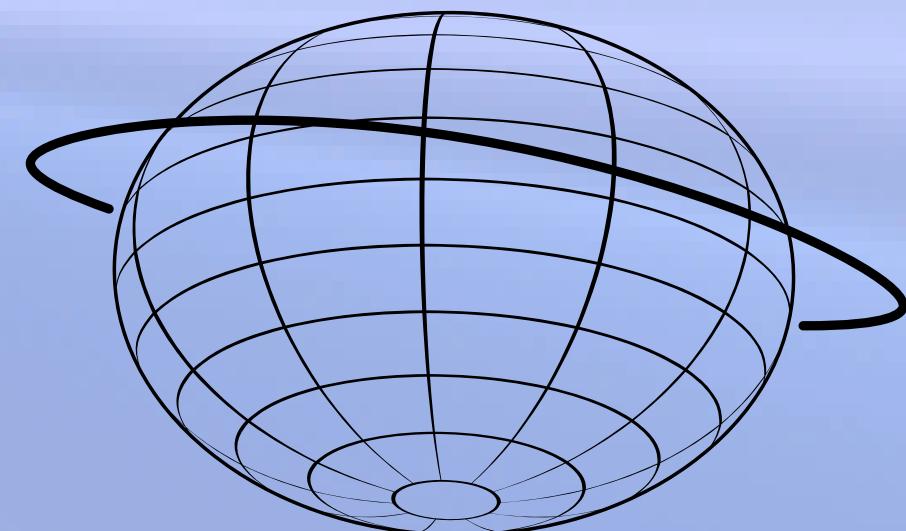
A MOBILE CLOUD BASED SMART AMBULANCE TRAFFIC MANAGEMENT SYSTEM WITH REAL-TIME TRACKING AND EMERGENCY ALERT COORDINATION

Course Code: PROJ-CSM783

Course Name: Project-I

Guide Name: Mrs.Pranashi Chakraborty

Department: CSE-AI





Presented By



Soumya Singha Roy

BWU/BTA/22/054

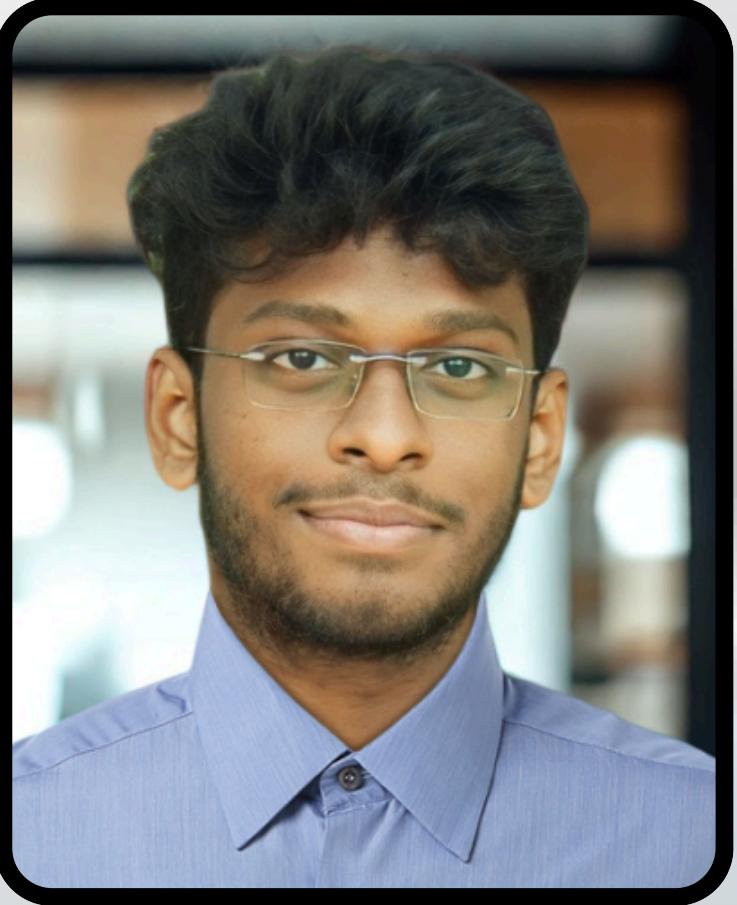
Roll: 22010332053



Subhojit Ghosh

BWU/BTA/22/019

Roll: 22010332018



Rudra Singha

BWU/BTA/22/056

Roll: 22010332055



Risita Pattanayak

BWU/BTA/22/042

Roll: 22010332041



AGENDA

INTRODUCTION



PROJECT OBJECTIVE



LITERATURE REVIEW



PROPOSED SOLUTION



IMPLEMENTATION



SYSTEM IMPLEMENTATION



FUTURE WORK



CONCLUSION



REFERENCES



INTRODUCTION



- In today's crowded cities, traffic congestion causes major delays, especially for ambulances during medical emergencies.
- Even a few minutes of delay can be life-threatening for a patient who needs immediate medical attention.
- To solve this, we propose a Smart Ambulance Navigation and Traffic Management System using a mobile application.

PROJECT OBJECTIVE

Reduce Ambulance
Delays in Traffic

Enable Smart Communication
Between Ambulances & Traffic
Authorities

Ensure Safe & Faster
Emergency Response

To minimize the time ambulances spend stuck in traffic by providing real-time navigation, priority routing, and automated traffic signal control.

To create a seamless digital link between ambulance drivers and traffic control through a mobile app, desktop dashboard, and backend server for instant alerts and updates.

To build a scalable system that supports green corridor creation, improves decision-making with KPIs, and ultimately increases patient survival by speeding up emergency medical transport.



LITERATURE REVIEW

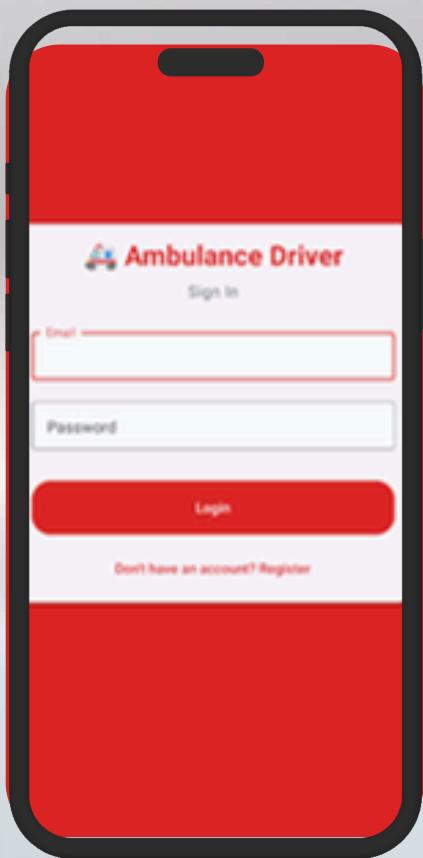
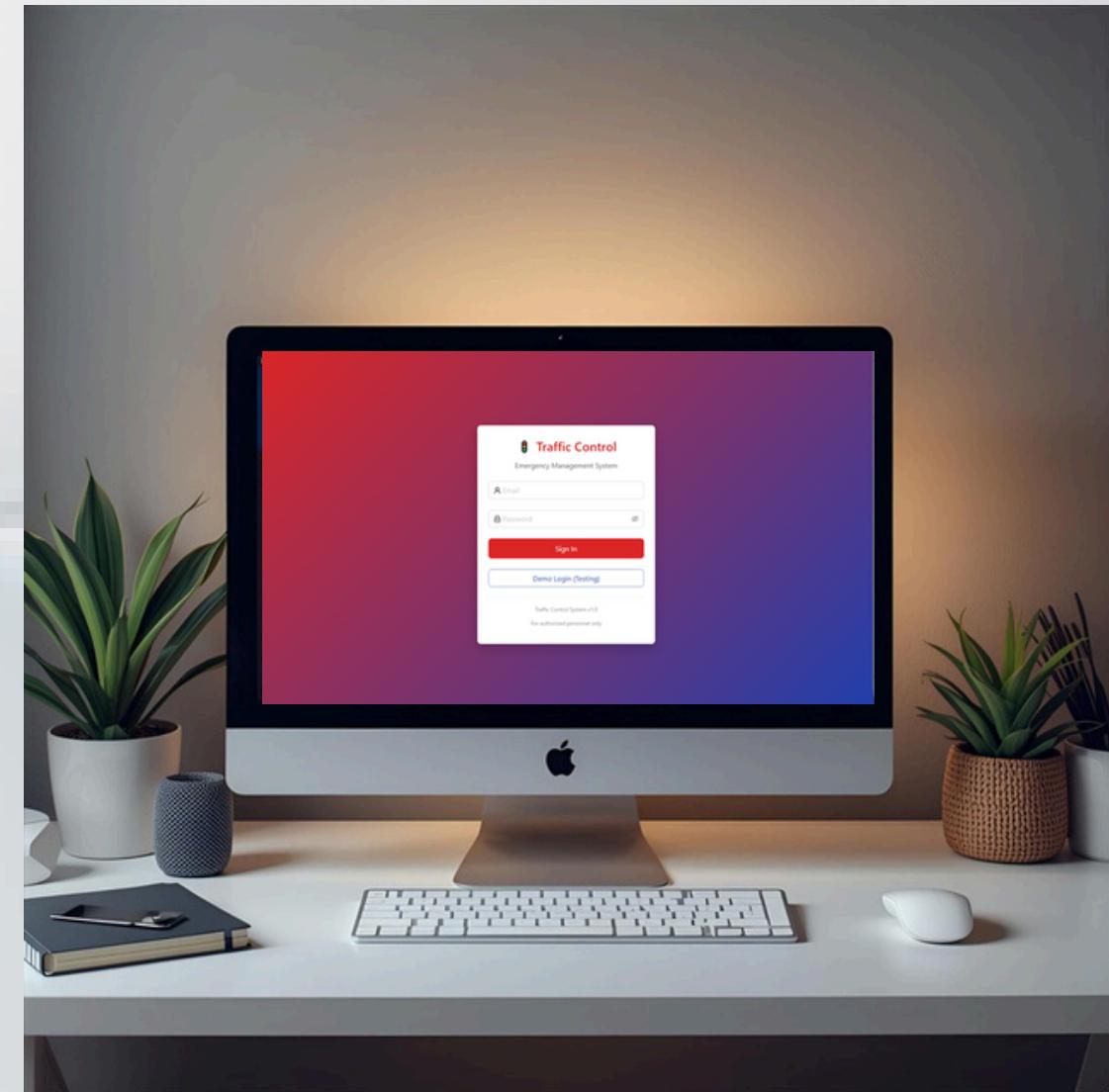
[Paper 1] ,This paper explains how a smart system can help ambulances move faster in heavy traffic. The main idea is to use modern technologies like GPS, mobile apps, and real-time traffic data to guide ambulances through the quickest route. It also highlights the importance of connecting ambulances with traffic control centers so that signals can be changed automatically to create a clear path. The study shows that such systems can reduce delays and help save lives by improving communication between the ambulance and traffic authorities.



[Paper 2] ,This paper focuses on making traffic management more efficient by using intelligent traffic signal systems. It explains how real-time monitoring can help traffic officers manage congestion better and give priority to emergency vehicles. The research also discusses how improving the backend system can make communication faster and more reliable. The goal is to reduce the time an ambulance spends waiting at signals and ensure a smooth, fast, and safe journey during emergencies.



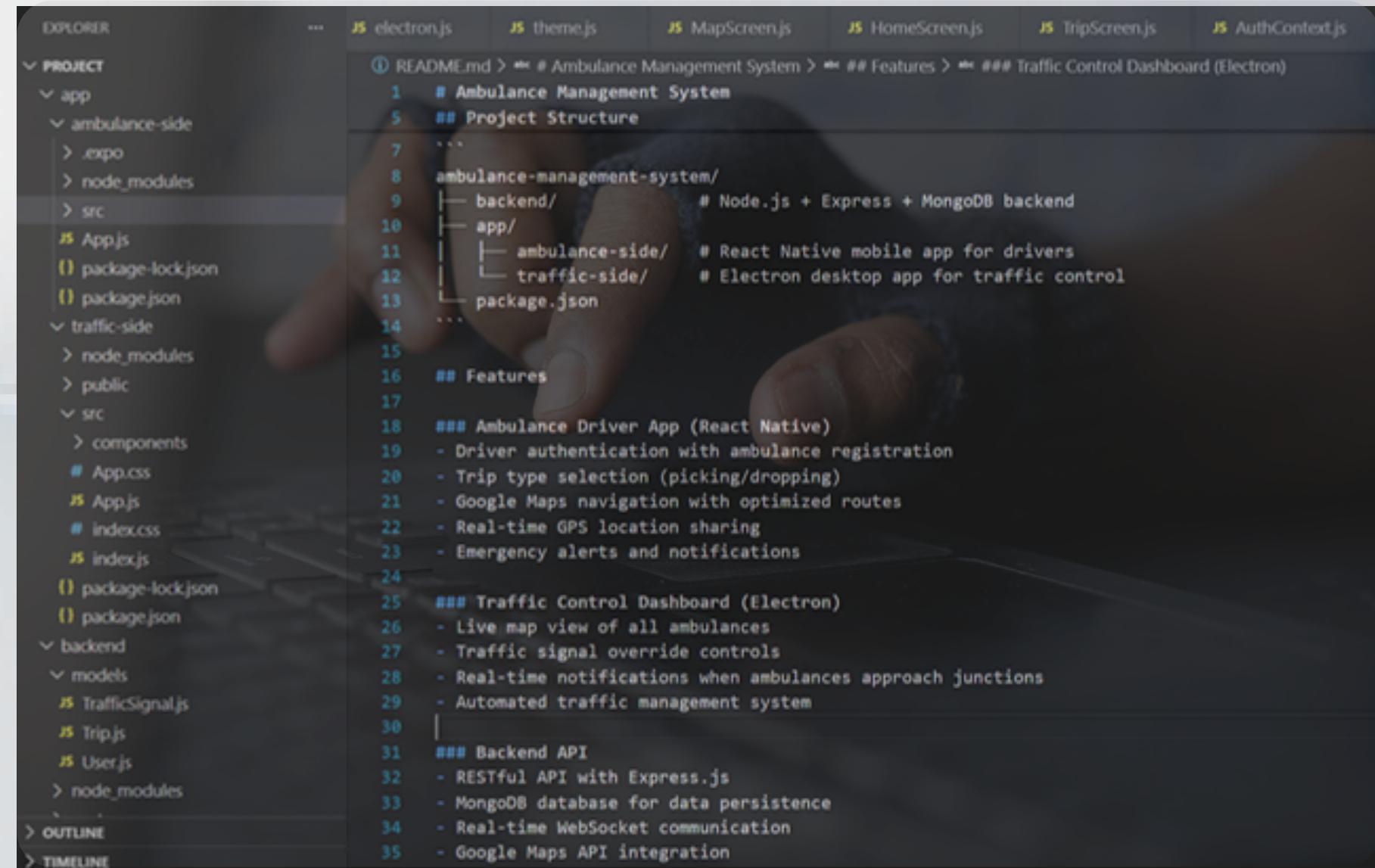
PROPOSED SOLUTION



IMPLEMENTATION

STEP-BY-STEP DEVELOPMENT

- Mobile App Development – React Native
- Control Room Dashboard Development – Electron.js
- Backend Server Development – Django
- Database Integration – MongoDB
- Mapping & Navigation Integration – Google Maps API
- Smart Alert & Green Corridor Implementation – WebSockets (Socket.io)
- System Testing & Optimization – Jest , Postman



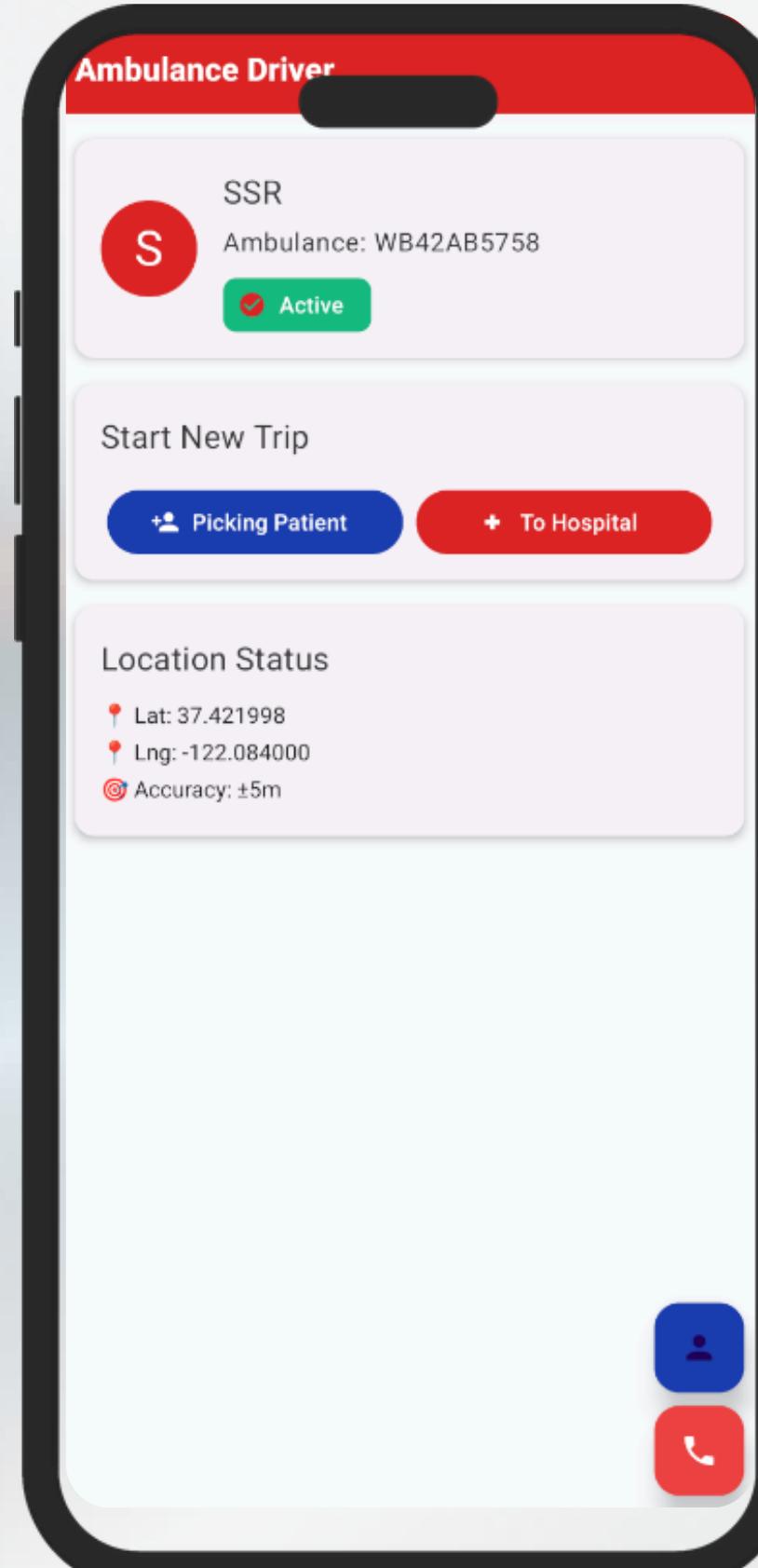
The screenshot shows a code editor interface with a dark theme. On the left is the Explorer panel, which displays the project structure:

- PROJECT
- app
 - ambulance-side
 - .expo
 - node_modules
 - src
 - App.js
 - package-lock.json
 - package.json
 - traffic-side
 - node_modules
 - public
 - src
 - components
 - App.css
 - App.js
 - index.css
 - index.js
 - package-lock.json
 - package.json
 - backend
 - models
 - TrafficSignal.js
 - Trip.js
 - User.js
 - node_modules
 - OUTLINE
 - TIMELINE

On the right is the main editor area, showing the content of the README.md file:

```
① README.md > ** # Ambulance Management System > ** ## Features > ** ### Traffic Control Dashboard (Electron)  
1  # Ambulance Management System  
5  ## Project Structure  
7  ...  
8  ambulance-management-system/  
9    |--- backend/          # Node.js + Express + MongoDB backend  
10   |--- app/  
11     |--- ambulance-side/ # React Native mobile app for drivers  
12       |--- traffic-side/  # Electron desktop app for traffic control  
13         package.json  
14  ...  
16  ## Features  
17  
18  ### Ambulance Driver App (React Native)  
19  - Driver authentication with ambulance registration  
20  - Trip type selection (picking/dropping)  
21  - Google Maps navigation with optimized routes  
22  - Real-time GPS location sharing  
23  - Emergency alerts and notifications  
24  
25  ### Traffic Control Dashboard (Electron)  
26  - Live map view of all ambulances  
27  - Traffic signal override controls  
28  - Real-time notifications when ambulances approach junctions  
29  - Automated traffic management system  
30  |  
31  ### Backend API  
32  - RESTful API with Express.js  
33  - MongoDB database for data persistence  
34  - Real-time WebSocket communication  
35  - Google Maps API integration
```

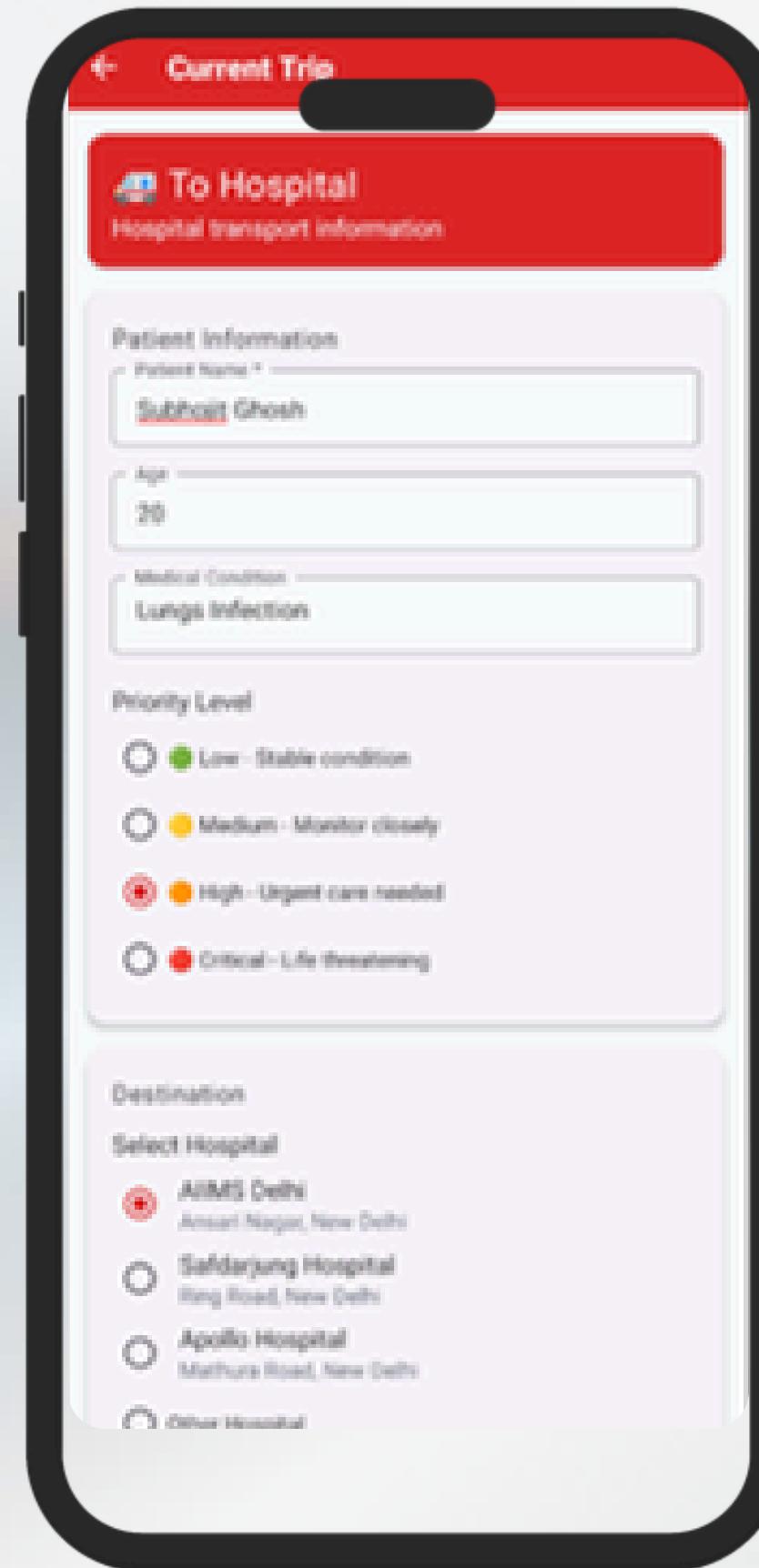
[PHASE-1] IMPLEMENTATION OF Mobile App & Control Room Dashboard Development



A screenshot of a control room dashboard titled "Traffic Control Center". It features a red header bar with the title and a "Welcome, Demo Traffic Controller | Logout" link. Below the header are four summary cards: "Active Ambulances" (5), "Ongoing Trips" (2), "Completed Today" (0), and "Override Signals" (0). A "Connected" status message indicates "Connected to ambulance tracking system". The main area contains a map titled "Real-time Ambulance Tracking" showing the locations of ambulances in Delhi and surrounding areas like Noida, Ghaziabad, and Gurgaon. The map includes major roads like NH9, NH24, and NE3. Below the map is a "Live Notifications" section with a message "No active notifications".

[PHASE-2] IMPLEMENTATION OF

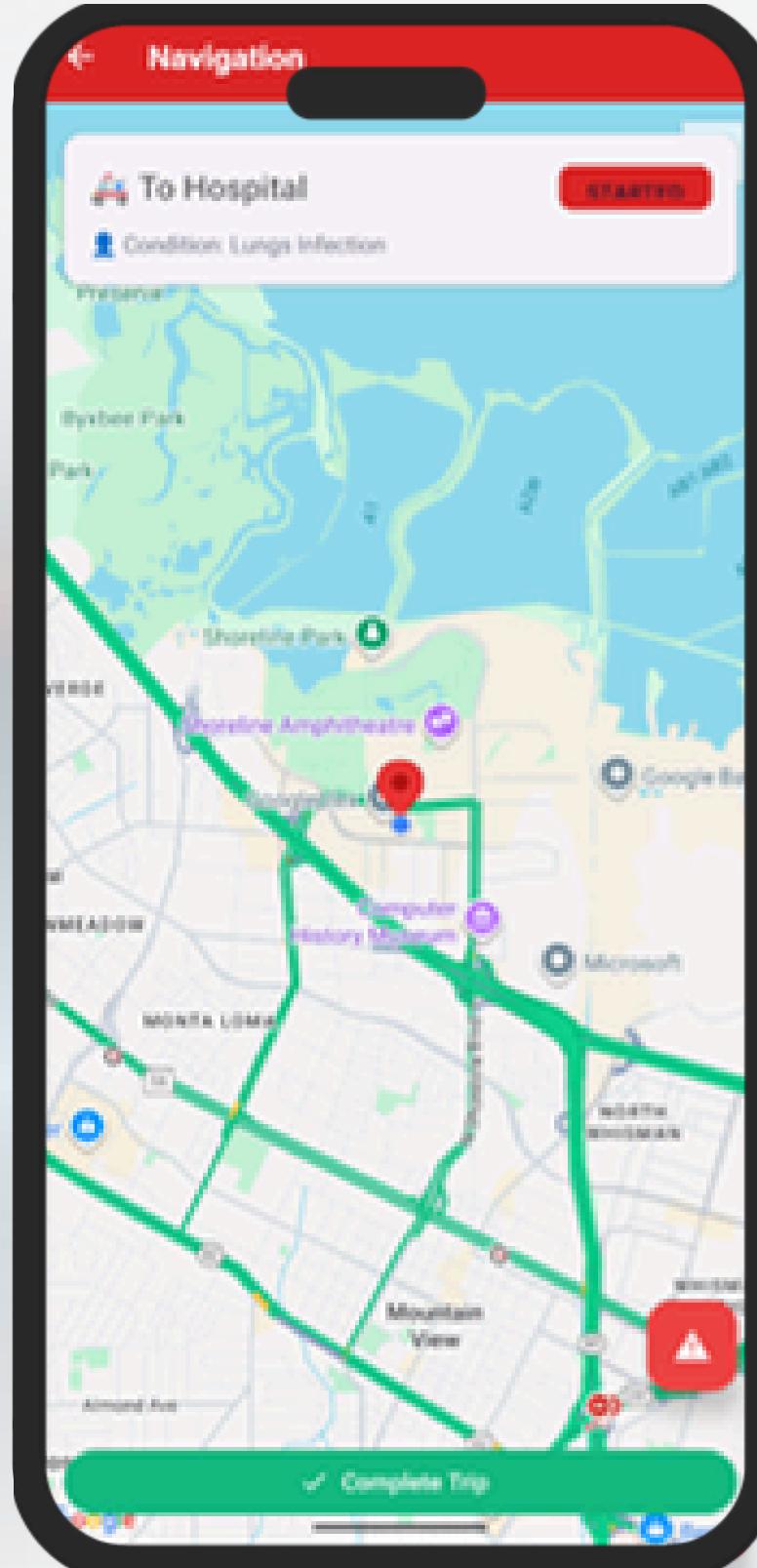
Mobile App & Control Room Dashboard Development



The control room dashboard has a red header bar with "Traffic Control Dashboard", "File View Emergency", "Welcome, Demo Traffic Controller | Logout", and a title "Traffic Control Center". It features four summary cards: "Active Ambulances" (1), "Ongoing Trips" (0), "Completed Today" (0), and "Override Signals" (0). Below these are tabs for "Live Map", "Ambulances", "Traffic Signals", and "Notifications". The "Live Map" section shows a map of New Delhi and Noida with various colored traffic routes and several blue dots representing active ambulances. The "Live Notifications" section indicates "No active notifications".

[PHASE-3] IMPLEMENTATION OF

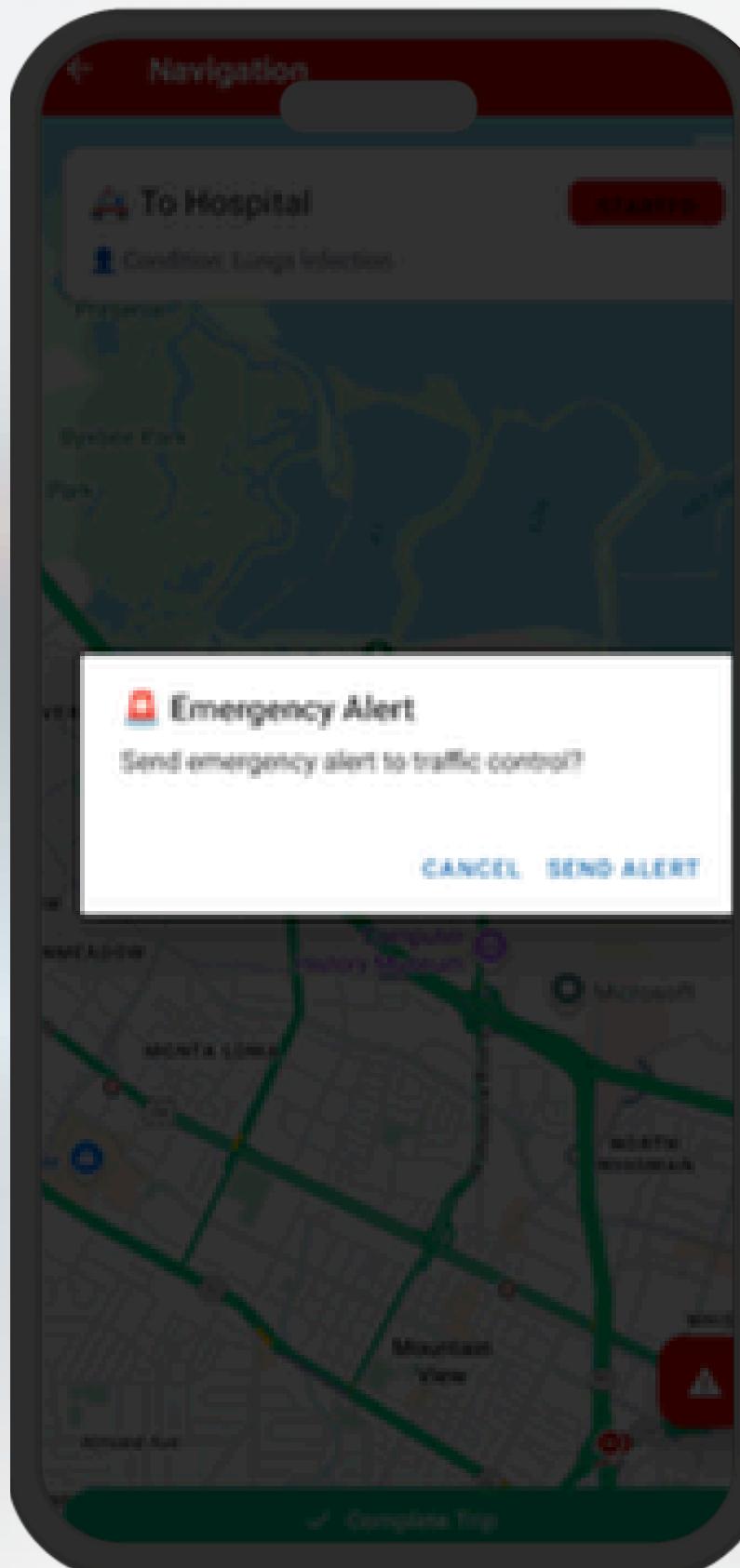
Mobile App & Control Room Dashboard Development



The dashboard has a red header bar with the title "Traffic Control Center". Below it, a red alert box shows "EMERGENCY ALERTS ACTIVE" with a message: "Emergency vehicle requires immediate assistance - Ambulance WB42P5758". The main area is divided into four sections: "Active Ambulances" (1), "Ongoing Trips" (1), "Completed Today" (0), and "Override Signals" (0). Below these are summary counts: 1 Active Ambulances, 0 Critical ETA (<10 min), 0 Patient Pickups, and 1 Hospital Drops. The "Live Ambulance Tracking" section shows a table for Ambulance ID WB42P5758, which is en route to AIIMS Delhi (Ansari Nagar, New Delhi) with a calculated ETA. It includes "Track" and "Emergency" buttons.

Ambulance ID	Trip Type	Destination	Location	ETA	Last Update	Actions
WB42P5758	TO HOSPITAL	AIIMS Delhi Ansari Nagar, New Delhi...	37.4220, -122.0640	Calculating...	14:58:06 2 minutes ago	<button>Track</button> <button>Emergency</button>

[PHASE-4] IMPLEMENTATION OF Mobile App & Control Room Dashboard Development



A screenshot of a desktop application titled "Traffic Control Dashboard". The main title bar says "Traffic Control Center". In the center, a modal window is open with the heading "Emergency Override - All Traffic Signals". It contains a warning message: "This will override ALL traffic signals in the system for 60 seconds. Use only in extreme emergencies." Below the message is a text input field with the placeholder "Enter reason for emergency override (required...)". At the bottom of the modal are "Cancel" and "ACTIVATE OVERRIDE" buttons. In the background, the dashboard shows "Active Ambulances" (1), "Live Map", "Ambulances" tab selected, "Traffic Signals" (0), "Patient" (0), and "Live Ambulance Tracking" for an ambulance with ID "WB42P5758" heading to AIIMS Delhi.

SYSTEM IMPLEMENTATION

1. Google Maps Integration

Used Google Maps API for real-time location tracking, route generation, traffic data, and map visualization in both mobile app and dashboard.

2. WebSocket (Socket.io) Communication

Implemented real-time communication for live ambulance location updates, emergency alerts, and instant dashboard notifications.

3. JWT-Based Authentication

Used JSON Web Tokens for secure login, role-based access, and protected API communication between app, dashboard, and backend.

```
op > ambulance-side > js App.js > ↴ App
 1 import React, { useEffect, useState } from 'react';
 2 import { NavigationContainer } from '@react-navigation/native';
 3 import { createStackNavigator } from '@react-navigation/stack';
 4 import { Provider as PaperProvider } from 'react-native-paper';
 5 import * as SecureStore from 'expo-secure-store';
 6 import { StatusBar } from 'expo-status-bar';
 7
 8 // Screens
 9 import LoginScreen from './src/screens/LoginScreen';
10 import HomeScreen from './src/screens/HomeScreen';
11 import TripScreen from './src/screens/TripScreen';
12 import MapScreen from './src/screens/MapScreen';
13 import ProfileScreen from './src/screens/ProfileScreen';
14
15 // Context
16 import { AuthProvider } from './src/context/AuthContext';
17 import { theme } from './src/theme/theme';
18
19 const Stack = createStackNavigator();
20
21 const express = require('express');
22 const TrafficSignal = require('../models/TrafficSignal');
23 const router = express.Router();
24
25 // Get all traffic signals
26 router.get('/signals', async (req, res) => {
27   try {
28     const signals = await TrafficSignal.find({ isActive: true })
29       .populate('overriddenBy', 'name');
30     res.json(signals);
31   } catch (error) {
32     console.error('Get traffic signals error:', error);
33     res.status(500).json({ message: 'Error fetching traffic signals' });
34   }
35 });
36
37 // Override traffic signal
38 router.post('/signals/:junctionId/override', async (req, res) => {
39   try {
40     const { operatorId, reason, duration = 60 } = req.body; // duration in se
41   }
42 }
```

SYSTEM IMPLEMENTATION

4. MongoDB Database Structure

Designed NoSQL collections for users, trips, alerts, and GPS coordinates using GeoJSON for efficient geolocation queries.

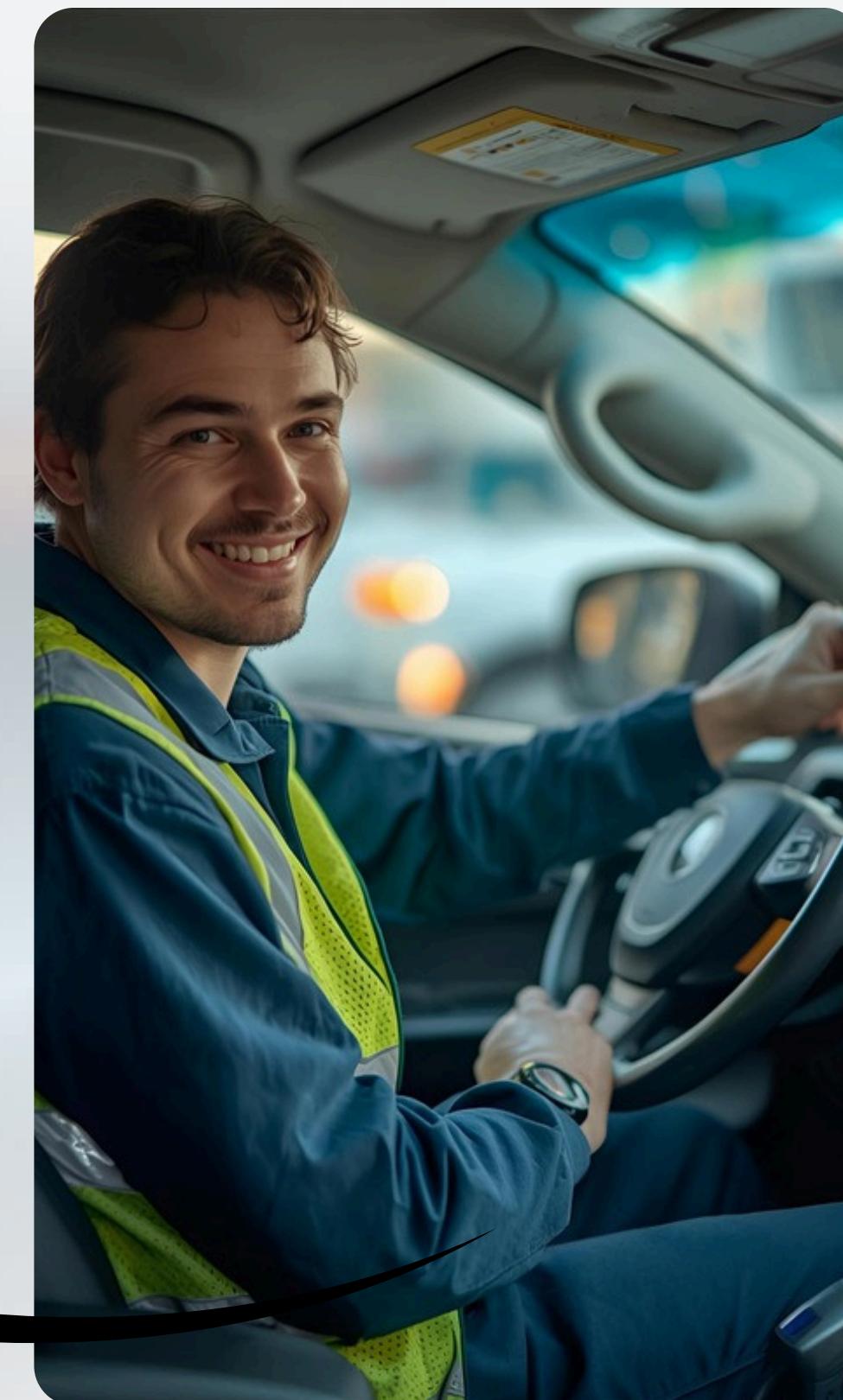
5. REST API Development

Built backend APIs for handling trips, alerts, real-time location updates, and dashboard data using Node.js and Express.js.

```
2
3 const tripSchema = new mongoose.Schema({
4   driverId: {
5     type: mongoose.Schema.Types.ObjectId,
6     ref: 'User',
7     required: true
8   },
9   ambulanceId: {
10    type: String,
11    required: true
12  },
13   tripType: {
14     type: String,
15     enum: ['picking', 'dropping'],
16     required: true
17 },
18   status: {
19     type: String,
20     enum: ['started', 'in_progress', 'completed', 'cancelled'],
21     default: 'started'
22 }
23
24 // Create trip
25 const express = require('express');
26 const jwt = require('jsonwebtoken');
27 const User = require('../models/User');
28 const router = express.Router();
29
30 // Register
31 router.post('/register', async (req, res) => {
32   try {
33     const { name, email, password, role, ambulanceId, phoneNumber } = req.body;
34
35     // Check if user exists
36     const existingUser = await User.findOne({ email });
37     if (existingUser) {
38       return res.status(400).json({ message: 'User already exists' });
39     }
40
41     // Create user
42     const user = new User({
43       name,
44       email,
45       password,
46       role,
47       ambulanceId,
48       phoneNumber
49     });
50     await user.save();
51     const token = jwt.sign({ _id: user._id, role: user.role }, process.env.JWT_SECRET);
52     res.status(201).json({ user, token });
53   } catch (error) {
54     console.error(error);
55     res.status(500).json({ message: 'Internal server error' });
56   }
57 }
58
59 // Login
60 router.post('/login', async (req, res) => {
61   const { email, password } = req.body;
62
63   const user = await User.findOne({ email });
64   if (!user || !user.comparePassword(password)) {
65     return res.status(401).json({ message: 'Invalid credentials' });
66   }
67
68   const token = jwt.sign({ _id: user._id, role: user.role }, process.env.JWT_SECRET);
69   res.json({ user, token });
70 }
71
72 // Logout
73 router.get('/logout', (req, res) => {
74   res.cookie('token', null, { maxAge: 0 });
75   res.json({ message: 'Logout successful' });
76 }
77 )
78
79 module.exports = router;
```



FUTURE WORK



- 1. AI-Based Route Prediction**
- 2. Integration With Real Traffic Signals**
- 3. Public Vehicle Alert System**
- 4. Multi-Ambulance Priority Handling**
- 5. Voice-Control Emergency Trigger**
- 6. Hospital Dashboard Integration**
- 7. Offline Navigation Enhancement**

CONCLUSION

This project has successfully moved from design to a functional Smart Ambulance Traffic Control System. The mobile app, web dashboard, and backend server have been fully built and integrated. Core features such as real-time tracking, trip management, and the emergency alert system have been implemented and tested.

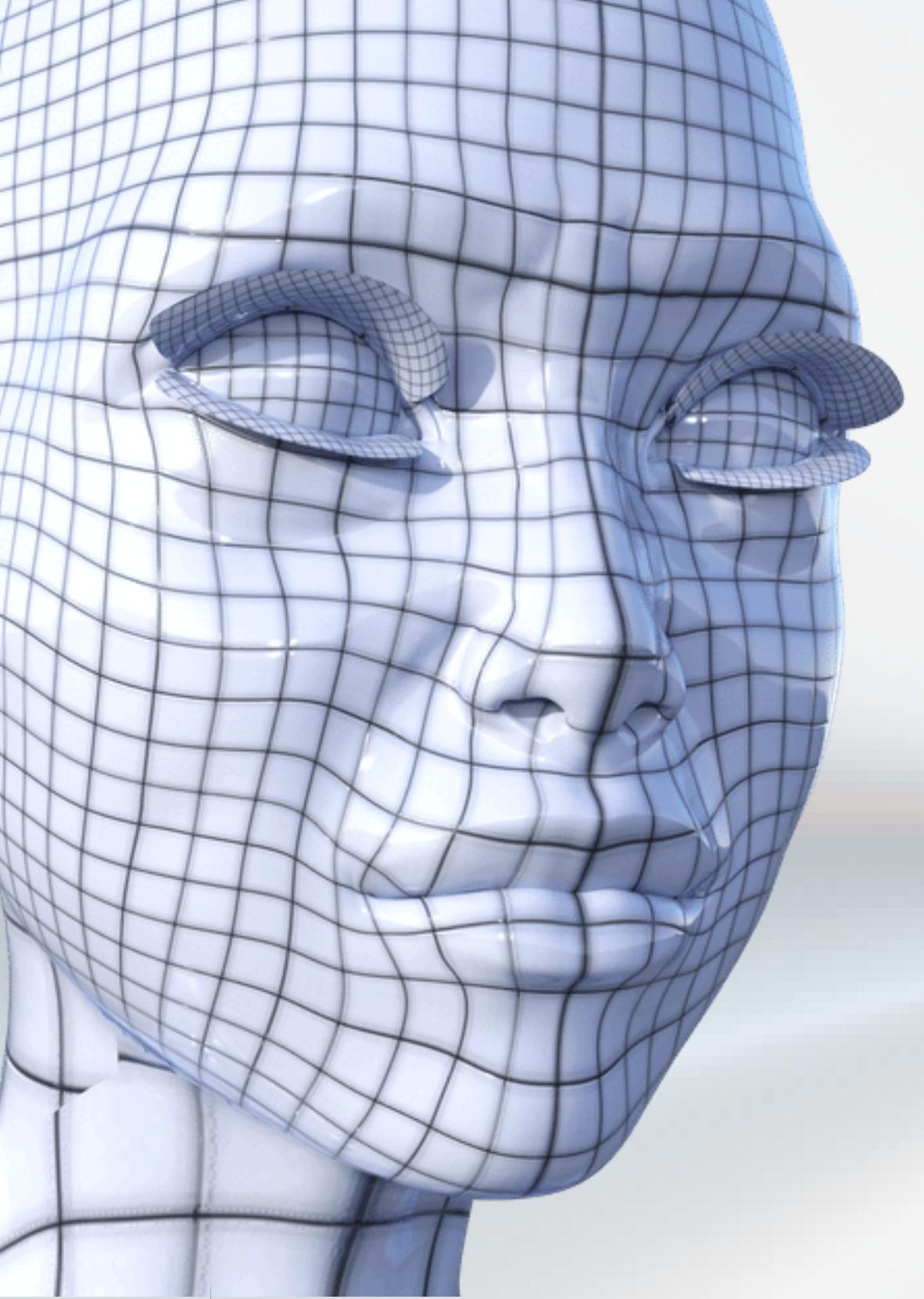
The newly designed inter-controller handoff mechanism further strengthens its real-world applicability. Overall, the project is on track to deliver a complete solution that effectively reduces ambulance delays.



REFERENCES

- Thanishka K, Sarala T, Kiran K N , Tanzeem Pasha, Ujwal B S, Varun Sameeran Namaval, 2025, An Intelligent Traffic Control System for Emergency Vehicles, <https://www.ijraset.com/research-paper/an-intelligent-traffic-control-system-for-emergency-vehicles>
- Gowtham .M, 2023, RFID BASED TRAFFIC CONTROL SYSTEM FOR EMERGENCY VEHICLE, https://www.researchgate.net/publication/370760738_RFID_BASED_TRAFFIC_CONTROL_SYSTEM_FOR_EMERGENCY_VEHICLE
- Abdel Ghani Karkar, 2019, Smart Ambulance System for Highlighting Emergency-Routes, <https://ieeexplore.ieee.org/abstract/document/8903948>
- P. Devigayathri, R. Amritha Varshini, Mi Pooja, S. Subbulakshmi, 2020, Mobile Ambulance Management Application for Critical Needs, <https://ieeexplore.ieee.org/abstract/document/9076395/>
- Ankur Dumka, Anushree Sah, 2020, Smart ambulance traffic management system (SATMS)—a support for wearable and implantable medical devices, <https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780128153697000094>





Thank You

FOR YOUR ATTENTION

