APSCHE SMART INTERNZ

Internship Title: Backend development with node js and Mongo DB – APSCHE LTB3

Project Title: Emergency Response Network

Team ID: LTVIP2025TMID21921

Team Leader: Sadvika Tammineni

Team members: Adabala Vivek

Chintala Lekhiya

D Nithin

PROJECT DESCRIPTION:

The ERN backend is built using Node.js, Express.js, and MongoDB (Mongoose) to handle emergency requests, assign responders and volunteers, and manage real-time data flow. APIs are developed to allow seamless integration with mobile applications, web dashboards, and IoT devices.

Key functionalities include:

- **Emergency request registration**
- **Real-time alerts to response teams**
- **Automated assignment of responders based on proximity and availability**
- **Incident status updates**

PROJECT OUTCOME:

The project outcome for an Emergency Response Network (ERN) would be:

1. **Faster Response Times: Quick mobilization of emergency services, saving lives and minimizing damage.**
2. **Enhanced Coordination: Seamless collaboration between agencies, improving resource allocation and efficiency.**

3. **Real-Time Data Use: Effective decision-making supported by up-to-date information and situational awareness.**
4. **Efficient Resource Management: Optimal use of medical supplies, personnel, and equipment.**
5. **Public Awareness: Timely communication with the public, ensuring safety and preparedness.**
6. **Better Preparedness: Continuous training for responders and the public, improving overall crisis management.**
7. **Effective Recovery: Swift restoration of services and infrastructure post-emergency.**
8. **Scalability and Sustainability: A resilient and adaptable network that can expand to meet future needs.**

KEY FEATURES:

- **Automated Emergency Response: Alerts responders based on proximity**
- **Real-time Updates: Updates on responder locations and incident status.**
- **User Authentication & Role Management: Secure access for users, volunteers, and admins.**
- **Incident Logging & Reports: Maintains history for future analysis.**
- **Multi-Platform API Integration: Supports mobile, web, and IoT applications.**

PROJECT OVERVIEW:

An Emergency Response Network (ERN) is a system designed to coordinate and manage resources, personnel, and communication during emergencies or crises, such as natural disasters, medical emergencies, or large-scale accidents. The goal of an ERN is to facilitate a rapid, effective response to mitigate damage, save lives, and restore normalcy as quickly as possible.

Scenario Based Case Study

Scenario: Fire Emergency in a Residential Area

- **A fire breaks out in a residential area. A resident reports the emergency via a mobile app.**
- **The request is received by the backend and logged into the database.**
- **The system identifies the nearest fire stations and available fire trucks.**
- **Responders are notified with location tracking.**
- **Authorities receive live updates on the incident status.**

This case study demonstrates how ERN streamlines emergency response through efficient data handling.

Technical Architecture

The system follows a Microservices-based REST API architecture, ensuring modularity and scalability.

Technology Stack:

- **Backend: Node.js, Express.js**
- **Database: MongoDB (Mongoose ODM)**
- **Authentication: JWT-based authentication**
- **API Testing: Postman / Thunderclient**

Backend Development:

- **Create RESTful APIs for handling emergency requests, user authentication, and incident updates**
- **mplement JWT-based authentication for security.**
- **Define controllers and middleware for data validation and request handling.**

Database:

- **Design MongoDB schema for users, reports and emergencies.**
- **Users Collection in MongoDB Compass**
- **Emergencies Collection in MongoDB Compass**
- **Reports Collection in MongoDB Compass**
- **Implement Mongoose models and database operations (CRUD).**
- **Set up MongoDB Atlas (if using cloud storage).**

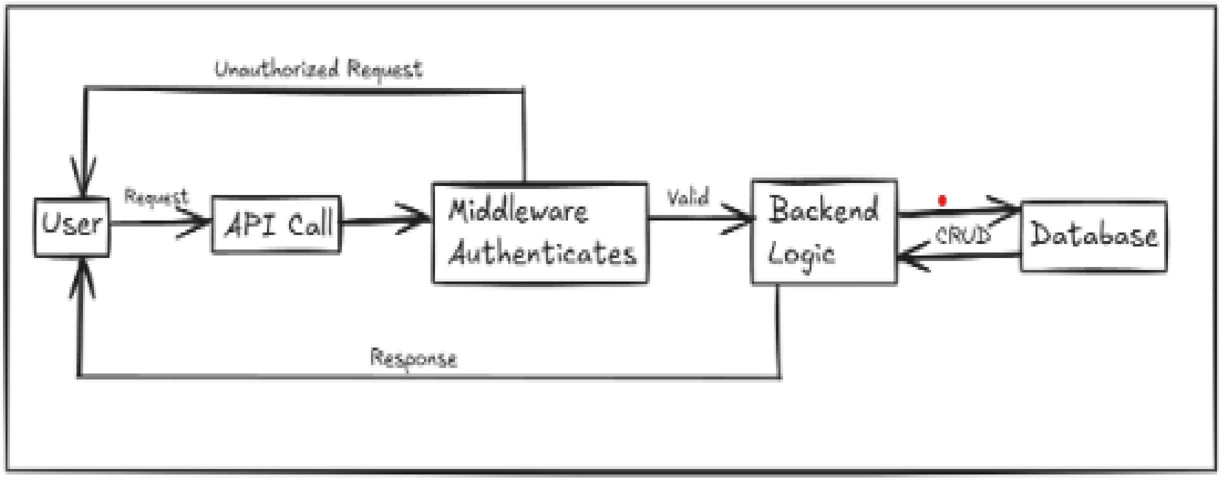
API Testing:

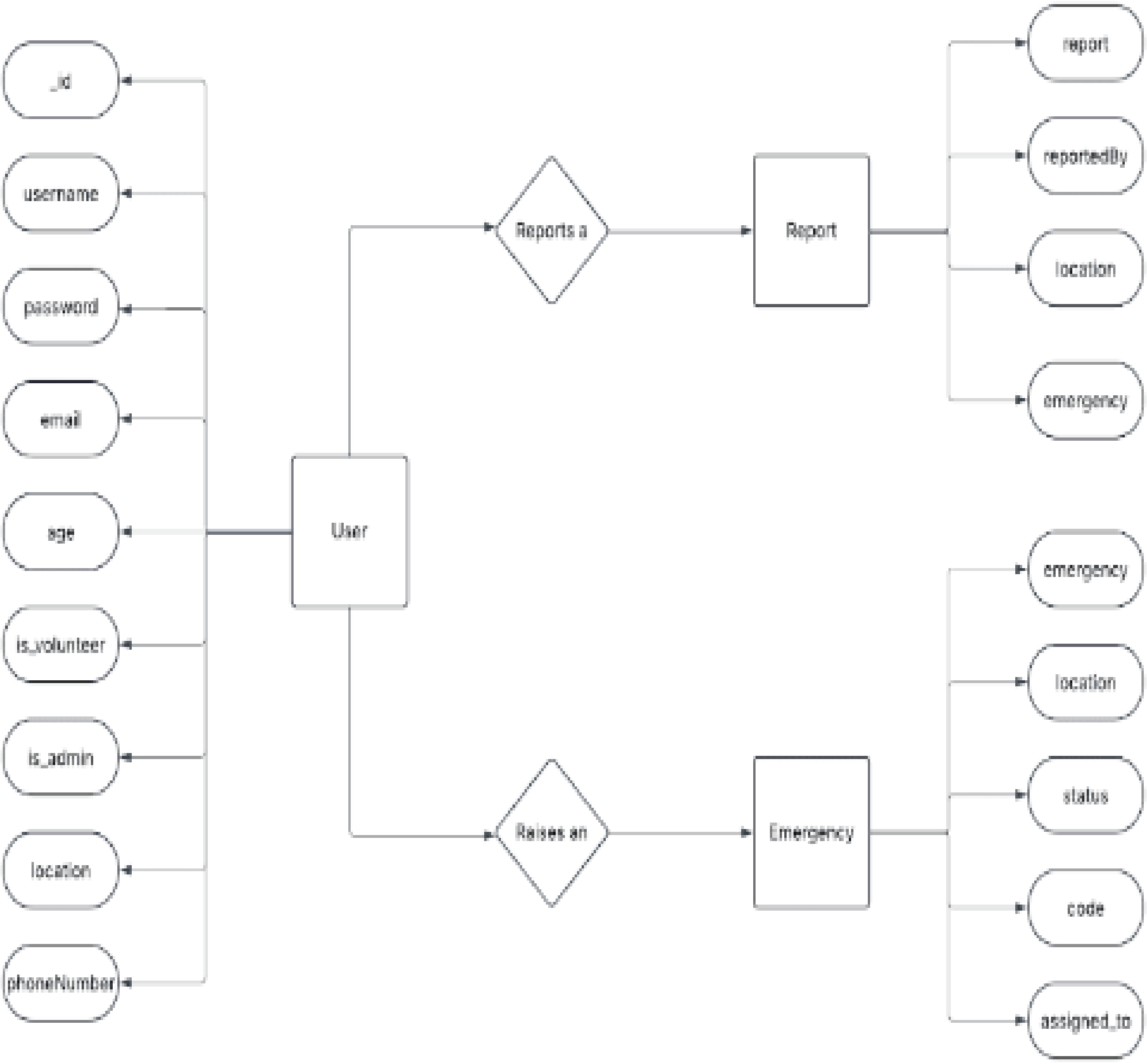
- **Use Postman(Download Application) or Thunderclient(VS Code Extension) to test API endpoints.**
- **Validate responses for:**
    - **Creating and updating reports, emergencies**
    - **Alerting volunteers and admins**
    - **Authentication and role-based access**
- **Debug issues and optimize performance.**

Conclusion:

The Emergency Response Network provides a robust backend solution for managing crisis situations effectively. With easy reporting, requesting help in emergencies, volunteers to help and secure data management, this system enhances emergency response efficiency, ensuring timely help for those in need.

PROJECT FLOW DIAGRAM:


Unauthorized Request
User
Request
API Call
Middleware Authenticates
Valid
Backend Logic
CRUD
Database
Response


_id
username
password
email
age
is_volunteer
is_admin
location
phoneNumber
User
Reports a
Report
report
reportedBy
location
emergency
Raises an
Emergency
emergency
location
status
code
assigned_to