SYMBIOSIS INSTITUTE OF TECHNOLOGY (SIT)
Constituent of Symbiosis International (Deemed University), Pune
(Established under Section 3 of the UGC Act of 1956 vide notification number F-9-12/2001-U-3 of the Government of india)
Re-Accredited by NAAC with 'A++' Grade
॥वसुधैव कुटुम्बकम्॥

E-Post

(Efficient Postal Management System)

Group Members:
1. Chris Vinod (22070126028)
2. Drishtti Narwal (22070126037)
3. Gaurav Prakash (22070126039)

Abstract

E-Post is an intelligent postal management system that aims to transform traditional mail services through the use of QR code-based tracking and automated sorting. The system works by assigning a specific QR code for every letter and parcel, thus making it possible to have instant status updates and also have real-time location tracking with a simple scan. This improves transparency and makes the work easier for postal staff. Developed using an intuitive web interface and mobile capability, E-Post provides the people role-based access based on their designation such as postmen, sorting officers, and administrators. It also minimizes human errors and delivery time.

Keywords

- QR Based Postal Tracking System
- Real-Time location Update
- Delivery Performance Monitoring
- Data Analytics
- Role Based Access

1. Introduction

Postal services are still a core aspect of logistics and communication especially in areas where digital solutions are not available. Nonetheless, the traditional postal systems continue to be based largely on manual sorting and tracking, resulting in inefficiencies, lost packages, and delayed deliveries. In a world of quick and transparent logistics, customers and postal staff increasingly look for real-time information and smooth service. This project, E-Post, was driven by the idea of harnessing the speed, accuracy, and convenience of existing technologies and integrating it to the postal sector. With the addition of QR-based tracking and smart automation, E-Post can revolutionize the way letters and parcels are handled, tracked, and delivered.

Traditional postal systems are primarily based on manual processes; these often can lead to delays, errors and even restricted tracking visibility for customers. With the growth of delivery volumes, there is a crucial requirement for a digital solution that can have efficient real-time updates, automated sorting, and effective communication among postal workers.

The Objectives of this project are:

- To create a web-based postal tracking and management system utilizing QR codes for real-time parcel and letter tracking.
- To give a customer-facing interface for tracking items and getting timely updates.
- To create an analytics dashboard for tracking performance indicators such as delivery effectiveness and delays
- To make the sorting process automatic by providing optimal next locations based on predefined postal routes.

The main scope of this project is:

- The system has frontend and backend parts developed using the MERN stack
- It provides QR code generation, scanning, and decoding
- The project is designed to handle different user roles like postmen, sorting officers, transporters, and admins.
- The initial development is centered around national-level postal services but can be scaled for wider deployment.

2. Literature Review

The article [1] "System optimization courier and parcel in cities" targets its efforts on maximizing courier and parcel delivery networks within urban centers to cut costs, and focuses in particular on the last stage of distribution, which represents ~50% of postal service expenses. It examines various delivery techniques (mailbox vs. hand), points (single vs. mass), and transportation (pedestrian vs. motorized), using MILP and Dijkstra's algorithm to plan routes over five subzones.

The Research [2] "Poliku Parcel System" unveiled at ICEISR 2024, is an end-user-centered parcel management solution based on Agile method with Visual Studio Code, Virtual Reality, and Augmented Reality technologies to mechanize academic parcel sorting and distribution. It uses real-time tracing, secure pickup, and customizable dashboards for users, workers, and administrators, decreasing wait times, missteps, and theft and raising satisfaction, modelled after Parcelhub. Its originality is in its academically tailored design, cutting-edge technology integration, and iterative development, filling campus logistics needs. Limitations are the lack of generalizability, quantitative measures, predictive analytics, and campus system integration, along with dependence on stable internet, recommending future research into scalability, offline operation, and data-driven optimization.

The paper [3] "integrating GIS, GPS, wireless communication, and optimization technologies for pick up/delivery service " consists of three main subsystems, they include sequence planning for optimal routes, real-time vehicle tracking, and PDA-based data transmission. It helps in reducing the service time by about 2 hours per day and increases the efficiency. Although cutting-edge in its day, it uses outdated technology (e.g., CDMA, Windows CE) and does not have QR-specific functionalities. Research gaps such as scalability and environmental concerns indicate future development with modern tech upgrades and sustainability emphasis.

The research paper [4] titled "The Object Tracking System At The Service Delivery Center Of The Traveling Salesperson Problem Method". It introduces a research focused on optimizing package delivery routes for Pos Indonesia, a state shipping enterprise, It suggests an object tracking system based on the Traveling Salesperson Problem (TSP) approach to find the shortest and most efficient delivery routes between post offices or cities.Its novelty primarily consists of incorporating real-time tracking of packages and delivery staff, ensuring route compliance and live status updates. Used in the context of a courier service, the system maximizes efficiency by reducing delivery times, lowering operational costs, and increasing service reliability.

The research [5] "Courier Service Tracking System: NIPOST Adamawa State Nigeria" helps in addressing the defects of the Nigerian Postal Service (NIPOST) in the state of Adamawa, where a manual tracking system usually results in delays, errors, and customer dissatisfaction. The research suggested an internet-based tracking system that is created and based on GPS and GIS technologies for the real-time tracking of vehicles and parcels, enhancing pickup and delivery scheduling, security, and loss minimization. Built with HTML, PHP, and MySQL, the system enables online monitoring by customers of their parcels, increasing transparency and confidence. The research emphasizes its ability to assist NIPOST in competing with international couriers such as DHL and UPS by streamlining operations, reducing expenses, and enhancing service quality, ending with a firm recommendation for its implementation to transform NIPOST's operations.

E-Post introduces a revolutionary innovation to the postal industry through the use of its technology-based method of tracking and delivery. By combining various different technologies such as QR-based tracking, auto-sorting, and real-time status, it overcomes inefficiencies common in conventional postal systems. The project increases the accuracy of operations, enhances customer and postal worker experience, and provides an expandable platform that can be implemented by other delivery and logistics companies. E-Post helps make public service systems easy to use, efficient and enhances overall service quality.

3. Methodology

3.1 Tools and technologies used

- **MERN Stack**

  The project is built using the MERN Stack , a full stack javascript which is known for its scalability and performance.

  - React.js -> ensures a reactive and dynamic UI
  - Node.js and Express.js -> allows for high performance backend with restful APIs
  - MongoDB -> efficiently stores and retrieves tracking data with its flexible document-based structure.

4. Implementation and Experimentation

4.1 Description of how the project was implemented

4.1.1 Backend

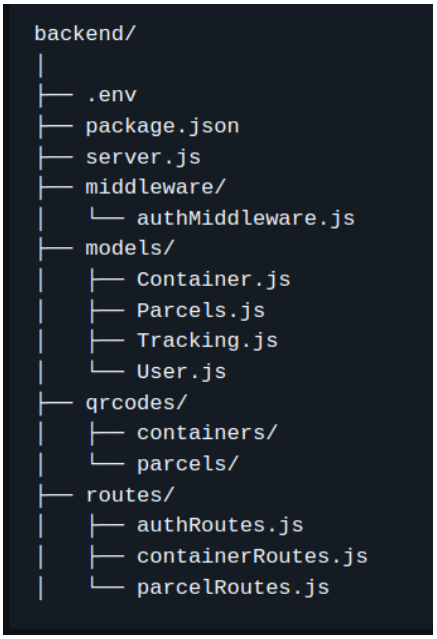

Fig.1 Backend structure

server.js: Entry point for the backend server. It loads environment variables, connects to MongoDB, sets up middleware (CORS, JSON parsing), and mounts all the API routes. It also starts the Express server.

routes/:

- **authRoutes.js**: Handles user registration, login, logout, and customer tracking ID management. Generates JWT tokens and manages user roles.
- **parcelRoutes.j**s: Handles all parcel-related endpoints: create, update, delete, fetch, and public tracking. Generates QR codes for parcels and manages tracking logs.
- **containerRoutes.js**: Handles container creation, updating, tracking, and management. Also generates QR codes for containers.

models/:

- **User.js**: Mongoose schema for users (admin, staff, customer). Stores user info, roles, and tracking IDs for customers.
- **Parcels.js:** Mongoose schema for parcels. Includes sender/receiver info, tracking ID, status, location, and references to containers and users.

- **Tracking.js**: Mongoose schema for tracking logs. Stores parcel/container status, location, and timestamps.
- **Container.js**: Mongoose schema for containers. Contains container ID, destination, parcels, logs, and status.

middleware/:

- authMiddleware.js: Express middleware to authenticate users via JWT. Sets req.currentUser for use in protected routes.

qrcodes/:

- containers/ and parcels/: Stores generated QR code images for containers and parcels, respectively.

.**env**: Stores environment variables such as MONGO_URI and JWT_SECRET.

4.1.2 Frontend

Frontend (e-post)

```
e-post/
└── src/
    ├── App.tsx
    ├── index.tsx
    ├── pages/
    │   ├── ParcelsPage.tsx
    │   ├── ContainersPage.tsx
    │   ├── TrackingPage.tsx
    │   ├── LoginPage.tsx
    │   ├── SignupPage.tsx
    │   ├── DashboardPage.tsx
    │   ├── Home.tsx
    │   └── NotFoundPage.tsx
    ├── components/
    │   ├── auth-provider.tsx
    │   ├── parcel-provider.tsx
    │   ├── parcels/
    │   │   └── ParcelList.tsx
    │   ├── tracking/
    │   │   ├── TrackingSearch.tsx
    │   │   └── TrackingResult.tsx
    │   └── ui/
    │       ├── button.tsx
    │       ├── card.tsx
    │       ├── input.tsx
    │       ├── label.tsx
    │       ├── tabs.tsx
    │       └── ... (other UI components)
    ├── layouts/
    │   └── DashboardLayout.tsx
    └── lib/
        └── api.ts
```

Fig 2. Frontend structure

App.tsx: Main React component that sets up routing using react-router-dom. Uses a ProtectedRoute component to guard routes based on authentication and user roles. Mounts all main pages and layouts.

index.tsx: Entry point for the React app. Renders the root component.

pages/:

- **ParcelsPage.tsx**: Main page for managing parcels. Includes tabs for adding and updating parcels, QR scanning, and listing parcels.
- **ContainersPage.tsx:** Page for managing containers.
- **TrackingPage.tsx**: Page for tracking parcels by tracking ID or QR code.
- **LoginPage.tsx / SignupPage.tsx**: Authentication pages. DashboardPage.tsx: Main dashboard after login.
- **Home.tsx**: Public landing page.
- **NotFoundPage.tsx:** 404 page.

components/:

- **auth-provider.tsx**: React context/provider for authentication state and user info.
- **parcel-provider.tsx**: React context/provider for parcel state and API calls.
- **parcels/ParcelList.tsx**: Component to display a list of parcels.
- **tracking/TrackingSearch.tsx**: Component for searching/tracking parcels by ID or QR code.
- **tracking/TrackingResult.tsx**: Displays tracking results.
- **ui/**: Shared UI components (Button, Card, Input, Label, Tabs, etc.).

layouts/:

- **DashboardLayout.tsx:** Layout wrapper for dashboard pages, includes navigation and outlet for nested routes.

lib/:

- **api.ts**: Axios instance configured for API requests, including JWT token handling.

4.2 Challenges Faced

- Syntax errors in react causing difficult to understand trace output.
- Errors which are not shown in the terminal and only shown in the browser console, making it difficult to debug code.
- API errors which resulted in api's not responding to the HTTP requests.
- Logic errors, when API gave only error responses to the http requests due bad logic in the code.
- Dependency file errors due to mismatch of versions of files and due to some modules not being updated to support react v.18

- Errors due to schematic mistakes in mongoDB schema made in the backend, primarily due to primary key type mismatches.
- Had to learn tailwind css to beautify components efficiently which took time.
- Errors due to jwtToken not being shared among the files within the protected routes resulting in logout due to lack of authentication token.

5. Results and Discussion

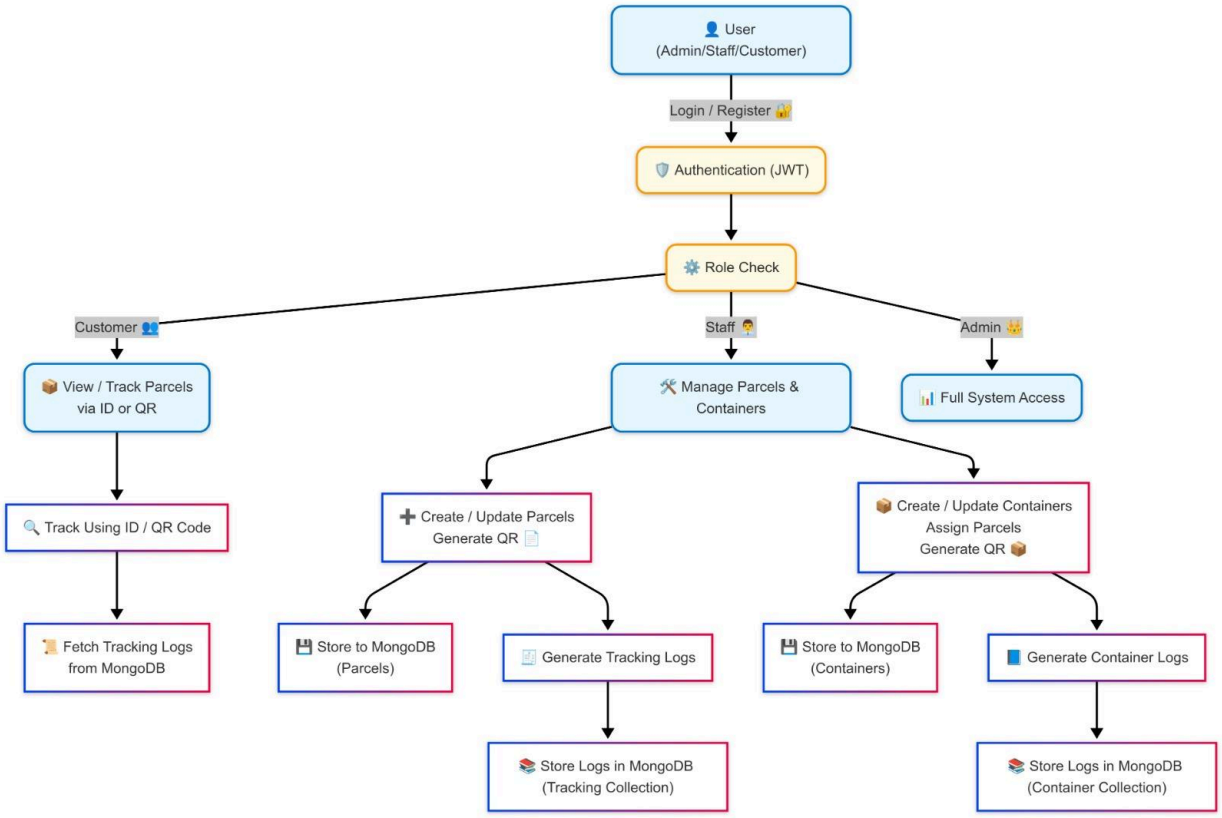

Fig 3. Workflow Diagram

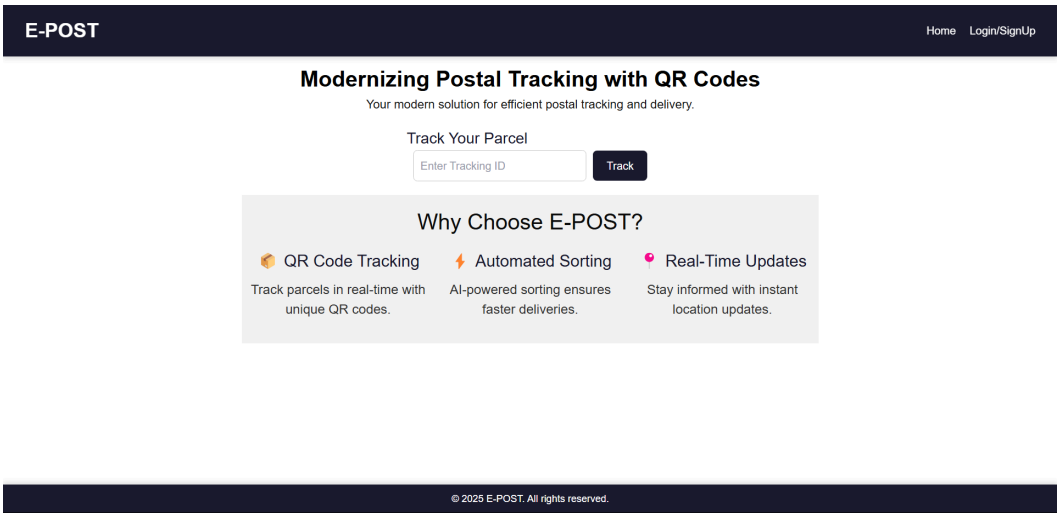

Fig 4. Home Page

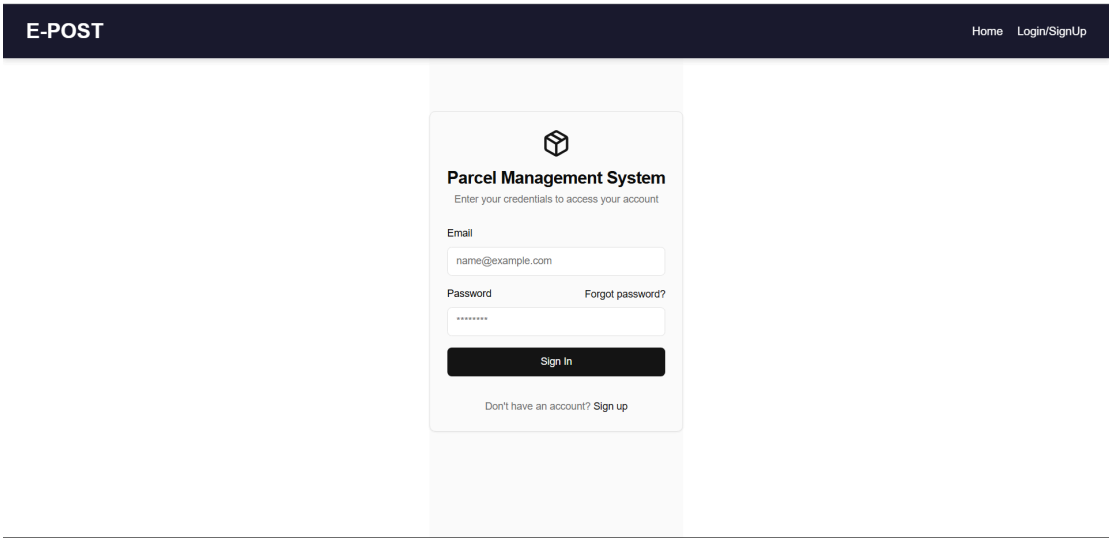

Fig 5. Login Page

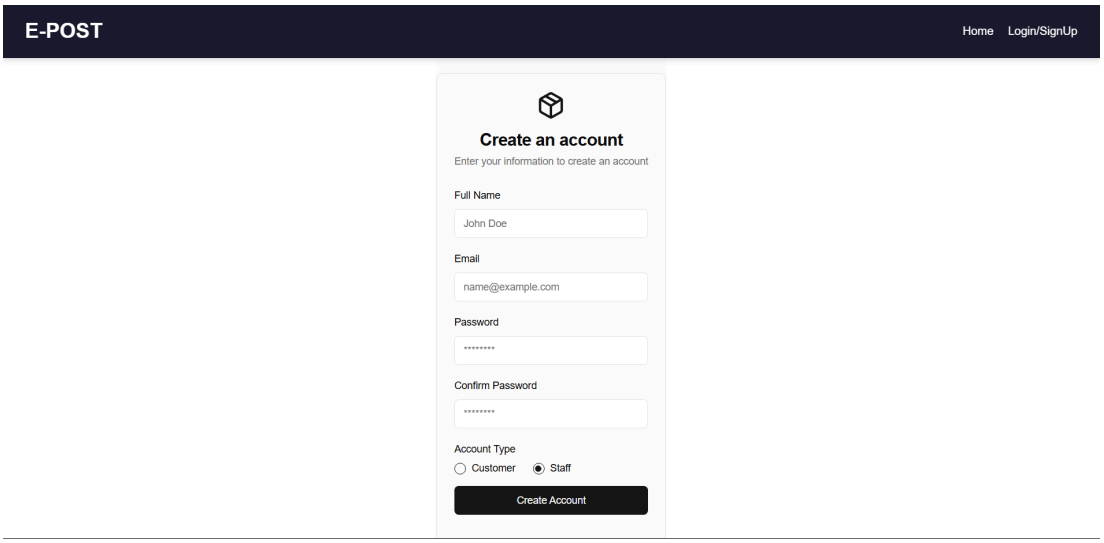

Fig 6. Signup Page

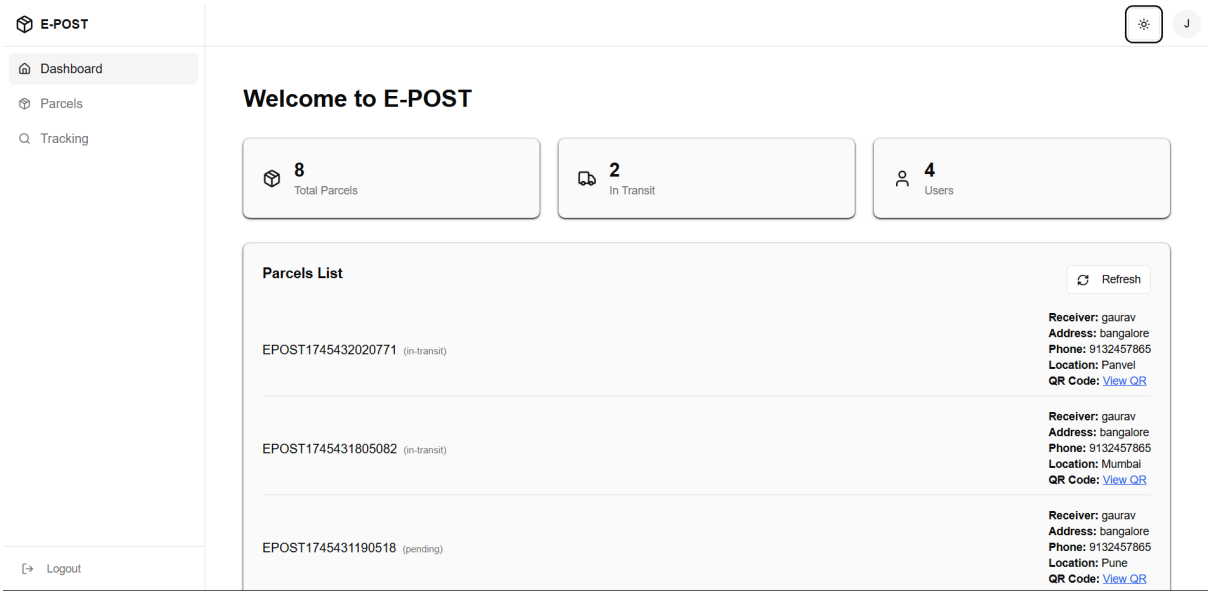

Fig 7. Dashboard Page

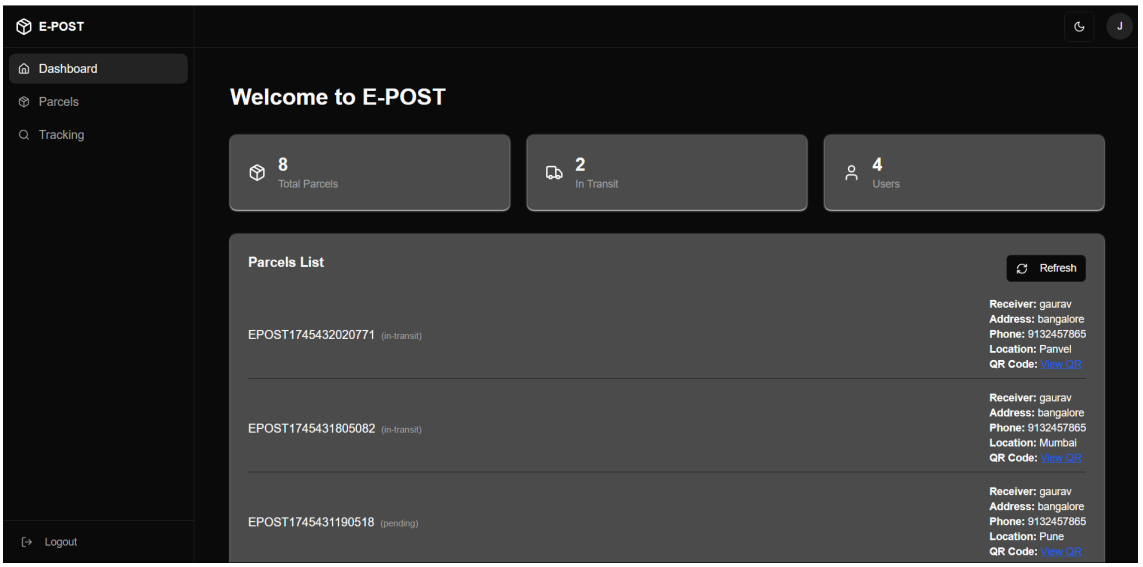

Fig 8. Dashboard (dark mode)

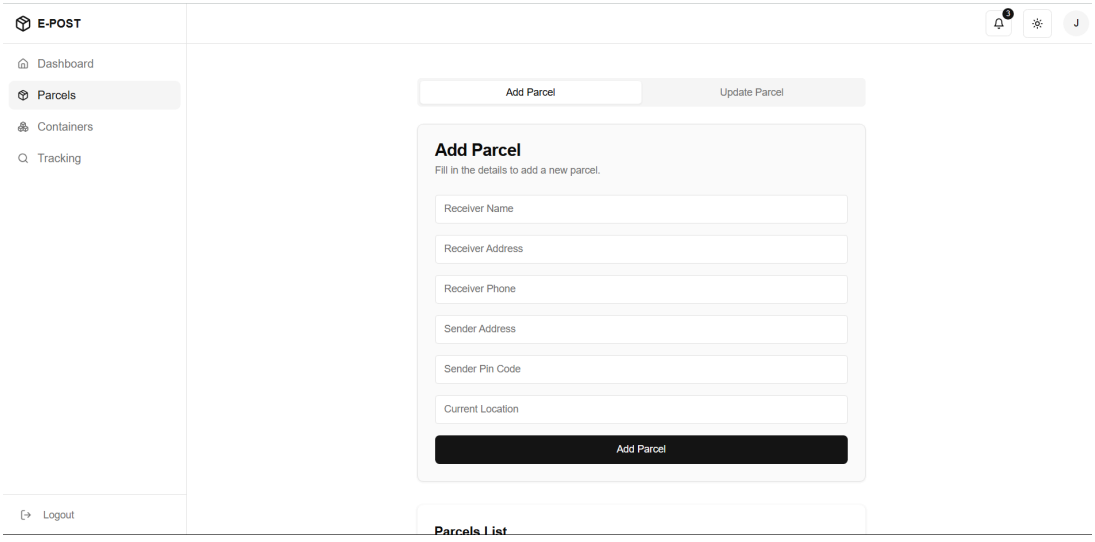

Fig 9. Add Parcel Page

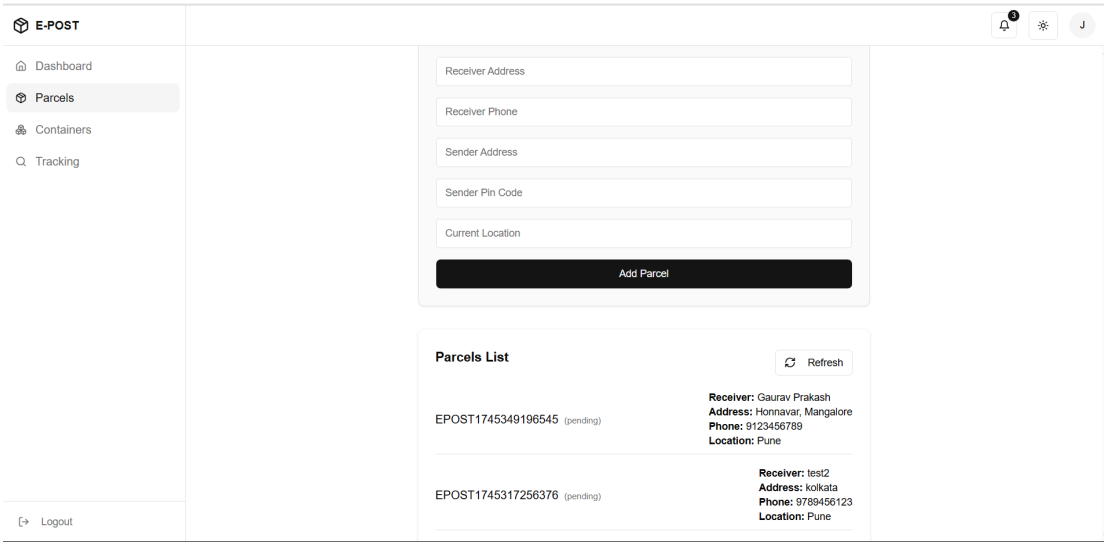

Fig 10. Parcel List in Parcel Page

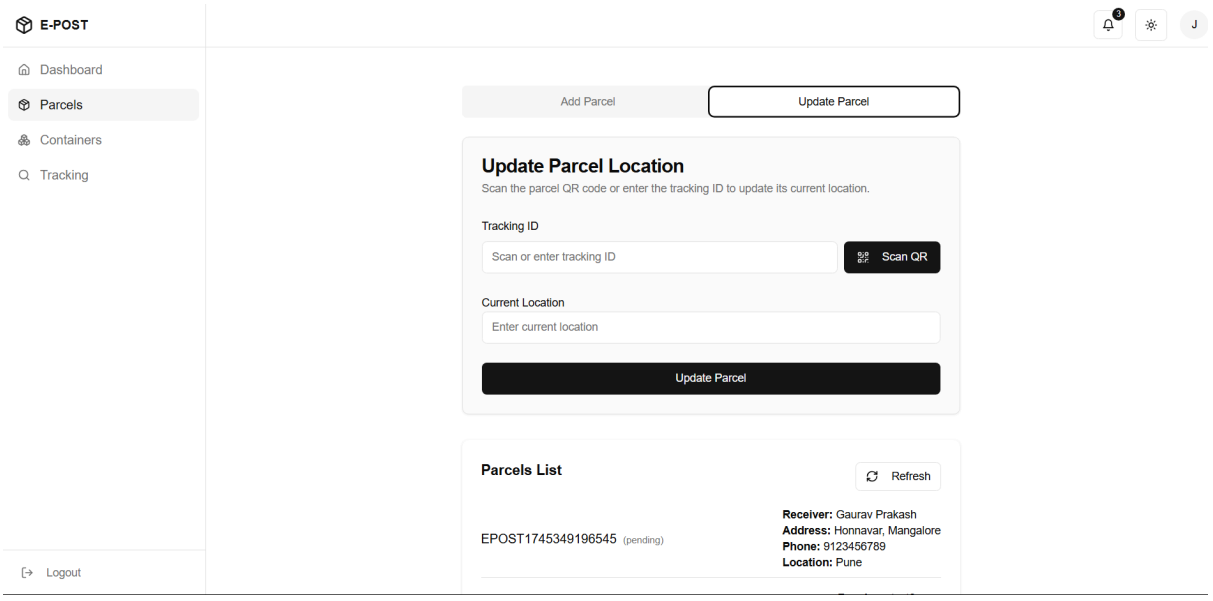

Fig 11. Update Parcel

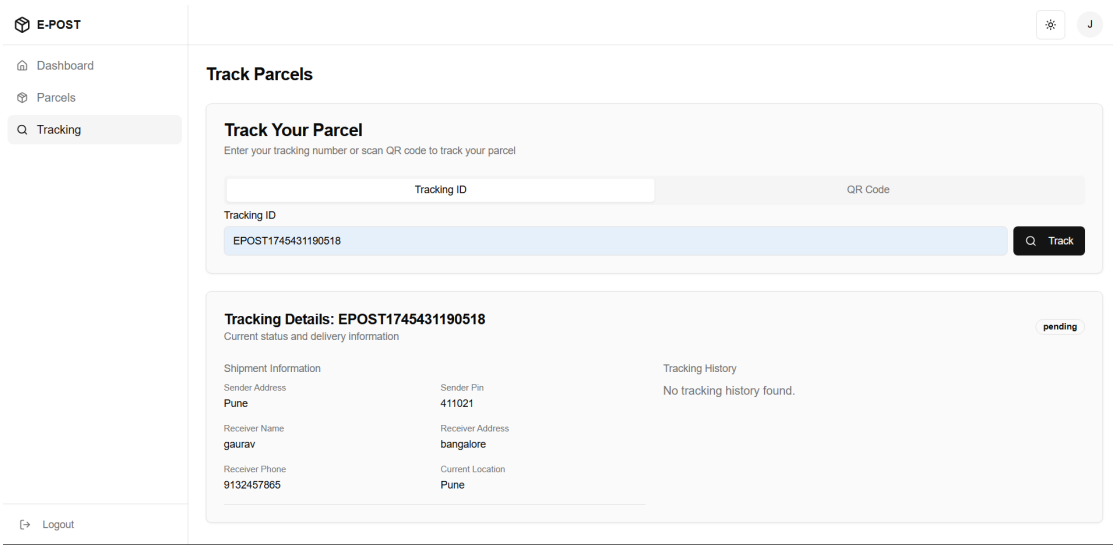

Fig 12. Track Parcel

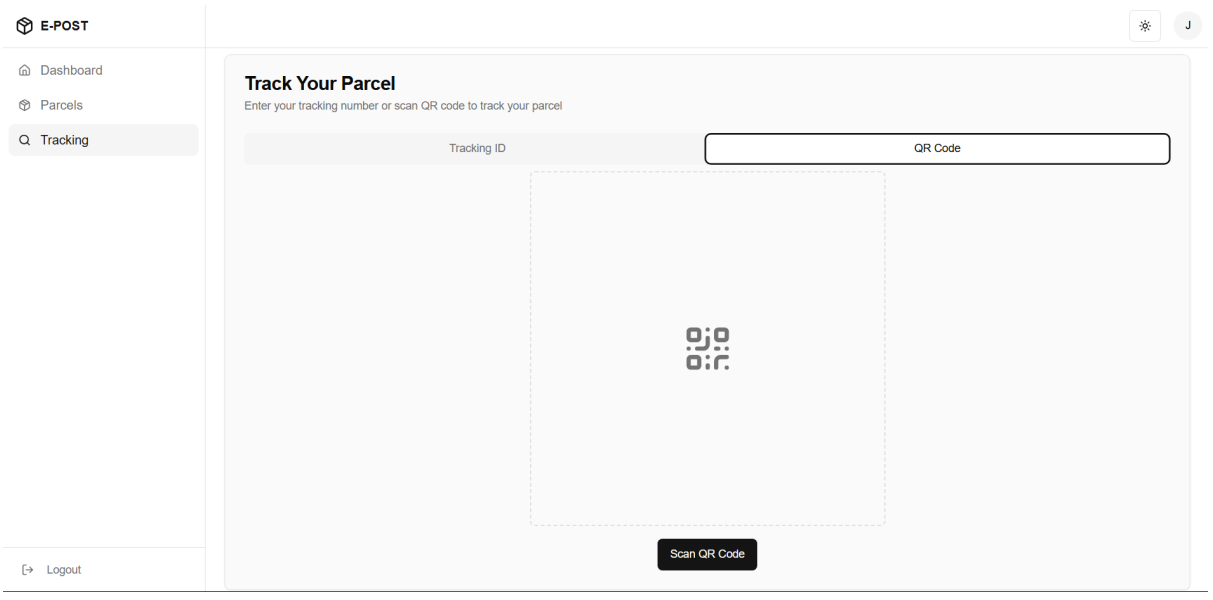

Fig 13. Track Parcel Using QR

6. Conclusion and Future Work

E-Post is an integrated solution for contemporary postal services with focus on traceability, user access control, and automation.

Limitations of the Current Implementation

- ML-based predictive logistics optimization yet to be implemented
- No integration with external APIs such as SMS/email notifications

Scope for Future Improvements

- Include ML model for best route planning
- Mobile app version
- Admin dashboard analytics and visualization
- Integration with logistics services and postal APIs

7. References

- [1] López, José Antonio Vera, et al. "System optimization courier and parcel in cities." *Procedia-Social and Behavioral Sciences* 160 (2014): 577-586
- [2] Mahran, Noraimi, Jamaah Suud, and Ahmady Solong. "POLIKU PARCEL SYSTEM: AN INTELLIGENT APPROACH TO PARCEL SORTING AND DELIVERY."
- [3] Hoon, Jung, and Lim Seung-Kil. "Integrating GIS, GPS, and Optimization Technologies for Pick-up/Delivery Service." *Korean Management Science Review* 21.3 (2004): 115-127.
- [4] Hardi, R., Suryana, N., Pee, N.C., Pribadi, A.S., Rusdi, J.F. and Junaidi, A., 2021, April. The object tracking system at the service delivery center of the traveling salesperson problem method. In *Journal of Physics: Conference Series* (Vol. 1807, No. 1, p. 012034). IOP Publishing.

- [5] Inuwa, Ibrahim, and N. D. Oye. "Courier Service Tracking System: NIPOST Adamawa State Nigeria."

Plagiarism and AI similarity

- Plagiarism less than 10%
- AI similarity less than 10%

ESE Rubrics:

Total ESE marks: 15M (will be reduced to 9M)

Project Report (Late submissions and not as per template will affect the marks)	**UI Design Quality**	**Code Quality**	**Implement ation and Testing**	**Viva**
5M	**2M**	**2M**	**3M**	**3M**