Transport and Logistics Management System
EZTransport

A Project Report Submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

Of

Bachelor of Technology

in

Computer and Communication Engineering

by

Adriteyo Das, Riddhima Jain, Anvita Warjri

230953244, 230953108, 230953100

Under the guidance of

Dr. Diana Olivia
Associate Professor
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

Mrs. Swathi BP
Assistant Professor-Senior Scale
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

April 2025

Contents

ABSTRACT

EZTransport is a web-based Transport and Logistics Management System designed to enhance efficiency, transparency, and scalability in logistics operations. The system centralizes the management of shipments, vehicles, drivers, and waypoints, offering real-time updates and automated workflows to reduce manual effort and improve customer satisfaction.

The application features secure, role-based access tailored to three primary user roles: Administrators, Drivers, and Customers. Administrators oversee the full operation, including managing users, vehicles, and assigning drivers to shipments. EZTransport is built around modular components, including User Authentication, Shipment Management, Waypoint Tracking, Vehicle and Driver Management, and a Notification System. Each module is backed by a normalized Oracle SQL database, ensuring high data integrity and efficient query performance. The frontend, developed using React.js, integrates seamlessly with a Flask backend, leveraging RESTful APIs for communication and maintaining real-time data synchronization.

Key features include real-time waypoint tracking, automated status notifications, role-based dashboards, and strict security protocols such as JWT-based authentication and HTTPS communication. The system supports scalability to accommodate growing user and shipment volumes, and is optimized for responsiveness across desktops, tablets, and smartphones.

By integrating all logistics operations into a unified digital platform, EZTransport not only streamlines internal processes but also enhances the end-user experience through transparency and automation. It is a future-ready solution built to support the evolving needs of logistics service providers.

ACM taxonomy:
formation Systems

: Database management system engines; Data management systems; Data analytics;
Software

: Web application development; Client-server architectures;
Networks

: Network protocols; Application layer protocols;
Security and Privacy

: Authentication; Access control; Web protocol security;

Sustainable Development Goal
SDG

: Industry, Innovation and Infrastructure

List of Figures

List of Tables

1. Users(user_id, username, full_name, email, phone, user_type, status, created_at, last_login, password)
2. Admins(admin_id, user_id, access_level)
3. Customers(customer_id, user_id, company_name, tax_id, credit_limit, payment_terms)
4. Drivers(driver_id, user_id, license_number, license_expiry, medical_check_date, training_certification, status)
5. Locations(location_id, address, city, state, country, postal_code, latitude, longitude, location_type)
6. Warehouses(warehouse_id, location_id, warehouse_name, capacity, current_occupancy, manager_id, operating_hours)
7. Vehicles(vehicle_id, license_plate, make, model, year, capacity_kg, vehicle_type, status, current_location_id, last_inspection_date)
8. Routes(route_id, route_name, origin_id, destination_id, distance_km, estimated_duration_min, status, hazard_level)
9. Waypoints(waypoint_id, route_id, location_id, sequence_number, estimated_arrival, estimated_departure)
10. Shipments(shipment_id, tracking_id, customer_id, origin_id, destination_id, route_id, vehicle_id, driver_id, status, total_weight, total_volume, shipment_value, insurance_required, special_instructions, created_at, pickup_date, estimated_delivery, actual_delivery)
11. Shipment_Items(item_id, shipment_id, description, quantity, weight, volume, item_value, is_hazardous, is_fragile)
12. Tracking_Events(event_id, shipment_id, event_type, location_id, event_timestamp, recorded_by, notes)

Abbreviations

RBAC	Role-Based Access Control
API	Application Programming Interface
UI	User Interface
JWT	JSON Web Token
HTTPS	Hypertext Transfer Protocol Secure
SQL	Structured Query Language
BCNF	Boyce-Codd Normal Form
ER	Entity-Relationship
1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
REST	Representational State Transfer
CRUD	Create, Read, Update, Delete
ETA	Estimated Time of Arrival
IoT	Internet of Things
GPS	Global Positioning System
RFID	Radio Frequency Identification
3PL	Third-Party Logistics

Chapter 1

Introduction

1.1 Introduction

In today's fast-paced logistics and supply chain industry, the efficient management of shipments, vehicles, drivers, and delivery timelines is crucial for ensuring customer satisfaction, operational transparency, and business scalability. Yet, many logistics providers continue to rely on fragmented tools or manual workflows that are prone to errors, delays, and data inconsistencies. To address these challenges, our team introduces EZTransport, an integrated Transport and Logistics Management System designed to streamline shipment handling, optimize resource allocation, and deliver real-time tracking capabilities across the logistics lifecycle.

EZTransport functions as a centralized digital platform for Administrators, Drivers, and Customers, offering seamless coordination of core logistics operations including shipment requests, waypoint updates, vehicle assignments, and delivery notifications. Through role-based interfaces and modular system design, it empowers logistics teams to automate processes, minimize human errors, and ensure timely deliveries.

1.2 Effective Innovations of EZTransport

- **Secure Role-Based Access Control (RBAC)**
  EZTransport implements a robust access control model where each user: Administrator, Driver, or Customer, interacts with the system based on predefined privileges. With secure login, passwords, session control, and role-based dashboards, it ensures data protection and restricts unauthorized access, reinforcing both privacy and accountability.

- **Normalized and Scalable Database Architecture**
  Built on a normalized Oracle SQL database schema (up to BCNF), EZTransport maintains data integrity across entities like Users, Shipments, Vehicles, Drivers, and Waypoints. The schema is optimized for scalability, enabling the system to support a growing number of transactions and complex data queries while preventing redundancy and en-

suring reliable data relationships.

- **Responsive and Interactive Web-Based Interface**
  Developed using React.js and powered by Flask APIs, the user interface provides a responsive and dynamic experience across desktop and mobile platforms. Each user is greeted with a customized dashboard, whether it's for tracking shipments, updating statuses, or managing operations. Designed to support fast and intuitive workflows.

- **End-to-End Shipment Lifecycle Automation**
  EZTransport covers the entire logistics journey—from placing a shipment request to final delivery. Administrators manage assignments, drivers update real-time statuses at waypoints, and customers receive instant notifications. This automation eliminates manual follow-ups and creates a transparent delivery pipeline.

- **Smart Package Classification for Safety-Sensitive Shipments**
  EZTransport introduces a smart classification feature that allows Administrators to mark packages as hazardous or fragile during shipment creation. This tagging helps automate safety checks and handling procedures by alerting Drivers and system modules to take necessary precautions. This feature reduces risk, ensures regulatory compliance, and improves overall shipment reliability.

- **Real-Time Waypoint Tracking and Notification Sync**
  Drivers log shipment progress at each waypoint, triggering automatic updates and timestamped notifications to customers. This real-time sync ensures delivery transparency, reduces support queries, and builds trust with clients expecting timely deliveries.

Chapter 2

Literature Survey

2.1 Database Design and Normalization

Database design is a fundamental aspect of any data-driven application. Effective schema design ensures data integrity, reduces redundancy, and optimizes query performance. Codd's relational model [1] laid the foundation for normalization, which involves decomposing complex relations into simpler, well-structured forms like 1NF, 2NF, and 3NF. Elmasri and Navathe provide a comprehensive approach to normalization and design methodologies in their widely used textbook [2]. Recent works also highlight the importance of entity-relationship (ER) modeling tools and automated schema generation systems [3].

2.2 Web Development with Flask and React

Flask is a micro web framework for Python, known for its flexibility and minimalism. It is widely used in applications where rapid development and simplicity are required. Grinberg's "Flask Web Development" [4] offers practical implementation strategies for REST APIs and templating. On the frontend, React stands out for its component-based architecture and virtual DOM efficiency, making it a go-to for dynamic UIs [5]. Combining Flask (backend) and React (frontend) in a decoupled architecture has become a common full-stack approach for modern applications [6].

2.3 Tutorials on HTML, CSS, and JavaScript

Foundational web technologies such as HTML, CSS, and JavaScript are essential for frontend development. The Mozilla Developer Network (MDN) provides one of the most authoritative and well-maintained tutorials for web standards [7]. JavaScript's evolution into ES6+ has significantly enhanced its capabilities, and learning platforms like freeCodeCamp and W3Schools [8], [9] offer structured learning paths with real-world coding examples. Responsive design using CSS3 and DOM manipulation via JavaScript are crucial for interactive UI development.

2.4 MySQL and SQL

SQL is the standard language for relational database operations, while MySQL is one of the most popular open-source RDBMSs used worldwide. Paul Dubois's "MySQL Cookbook" [10] and "Learning SQL" by Alan Beaulieu [11] provide practical examples ranging from simple queries to complex joins and stored procedures. With the rise of data-driven applications, understanding indexing, transactions, and performance tuning in MySQL is more relevant than ever [12].

2.5 General Transport and Logistics Management

The transport and logistics domain deals with the movement, storage, and flow of goods, often involving complex networks of stakeholders. Modern systems rely on real-time tracking, optimization algorithms, and integration with IoT devices [13]. Rushton et al. [14] provide a detailed understanding of logistics processes, including inventory control, warehouse management, and transportation planning. Recent developments focus on digitizing operations for better visibility and efficiency in the supply chain [15].

Chapter 3

Problem Statement

3.1 Problem Statement

In today's fast-paced, digitally connected world, individuals and businesses increasingly rely on efficient and reliable transportation services. However, many existing transport booking systems suffer from fragmented processes, lack of real-time tracking, poor scalability, and suboptimal user experience. These challenges hinder both end users seeking quick and reliable transportation and service providers managing driver fleets and operations.

Traditional transportation services or manual booking methods fail to meet the growing demand for transparency, automation, and instant availability. From inconsistent driver allocation to outdated location tracking systems, users often face delays, confusion, and limited control over their journey. Likewise, transport operators struggle with inefficient route planning, unclear driver accountability, and the absence of centralized management for users, trips, and feedback.

To address these inefficiencies, the EZ Transport System has been designed as a unified web-based platform that connects passengers and drivers through a secure, scalable, and real-time interface. With role-based access, geolocation services, and automated ride management, EZ Transport delivers a smarter, faster, and more transparent transportation solution for users and administrators alike.

3.2 Why is EZ Transport Needed?

Current urban and regional transport systems often rely on outdated, disconnected tools for booking, tracking, and managing rides. This results in:

- Long wait times for passengers due to inefficient driver matching
- Difficulty in locating available transport options in real time
- Manual fare handling or lack of price transparency

- Inability for administrators to manage drivers, routes, and trip data from a centralized system

EZ Transport fills this gap by offering a web-based, location-aware, user-role driven transport platform that handles everything from trip booking and live tracking to driver assignment and fare calculation—all in one system.

By integrating core transport functions and automating time-consuming processes, EZ Transport improves accessibility, reduces operational costs, and enhances the user experience for passengers, drivers, and system administrators.

3.3 Problem-Solving Approach of EZ Transport

1. **Centralized Ride Management System**
   All operations—user registration, trip requests, driver assignment, fare tracking, and trip history—are handled in one platform, reducing fragmentation and boosting efficiency.

2. **Role-Based Access Control (RBAC)**
   Three key user roles—Admin, User (Passenger), and Driver—ensure that each user has access to only the features and data relevant to their responsibilities, improving system security and usability.

3. **Real-Time Location Tracking**
   Using geolocation APIs, users can book nearby drivers, view real-time driver locations, and track their trips seamlessly through the web interface.

4. **Dynamic Fare Calculation**
   EZ Transport incorporates a smart pricing model that considers distance and base fare, calculating trip costs automatically to eliminate disputes and manual errors.

5. **Secure Authentication and Session Handling**
   The system uses secure login with JWT (JSON Web Tokens), encrypted passwords, and session management to protect user data and prevent unauthorized access.

6. **Responsive User Interface**
   Developed with React.js for the frontend and Flask (Python) for the backend, the system features a clean, intuitive interface optimized for desktop and mobile web usage.

7. **Automated Trip History and Feedback Logging**
   Each completed trip is recorded in a searchable, filterable log for users and admins. Passengers can rate and review trips, helping maintain quality of service.

8. **Scalable Architecture**
   Designed with modular components and RESTful APIs, EZ Transport is ready to scale across regions, fleets, and user bases without performance bottlenecks.

Chapter 4

Database Design

4.1 ER DIAGRAM

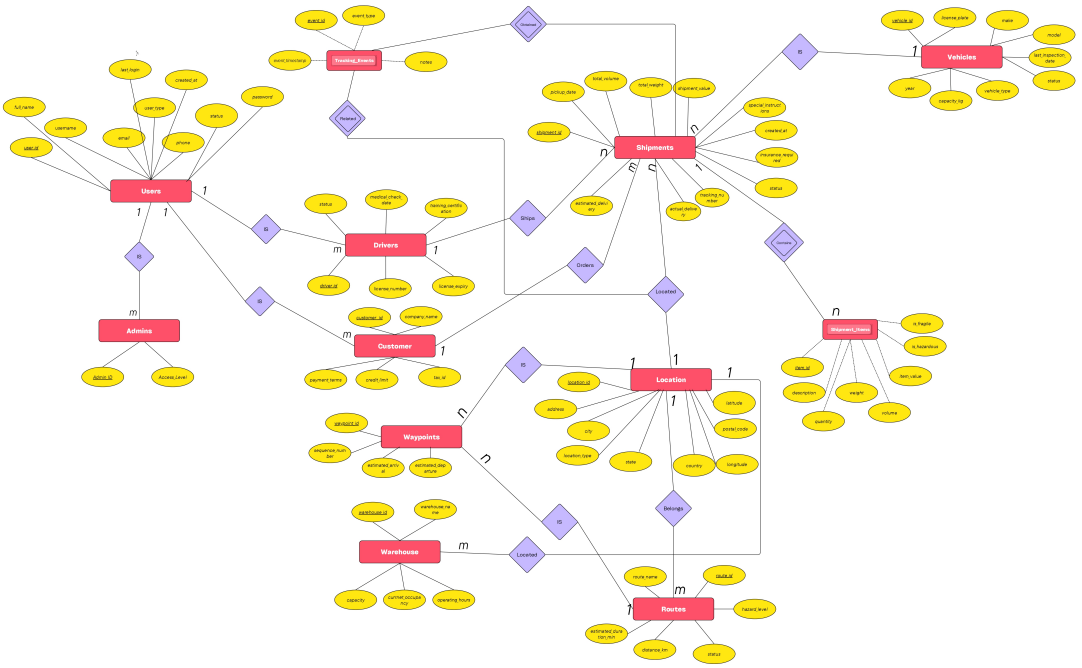

Figure 4.1: ER Diagram

4.2 SCHEMA DIAGRAM

4.3 SCHEMA TABLE

4.4 Normalization

The database schema has been normalized to the Boyce-Codd Normal Form (BCNF). The normalization process successfully transformed the original schema into a well-structured and optimized design that:

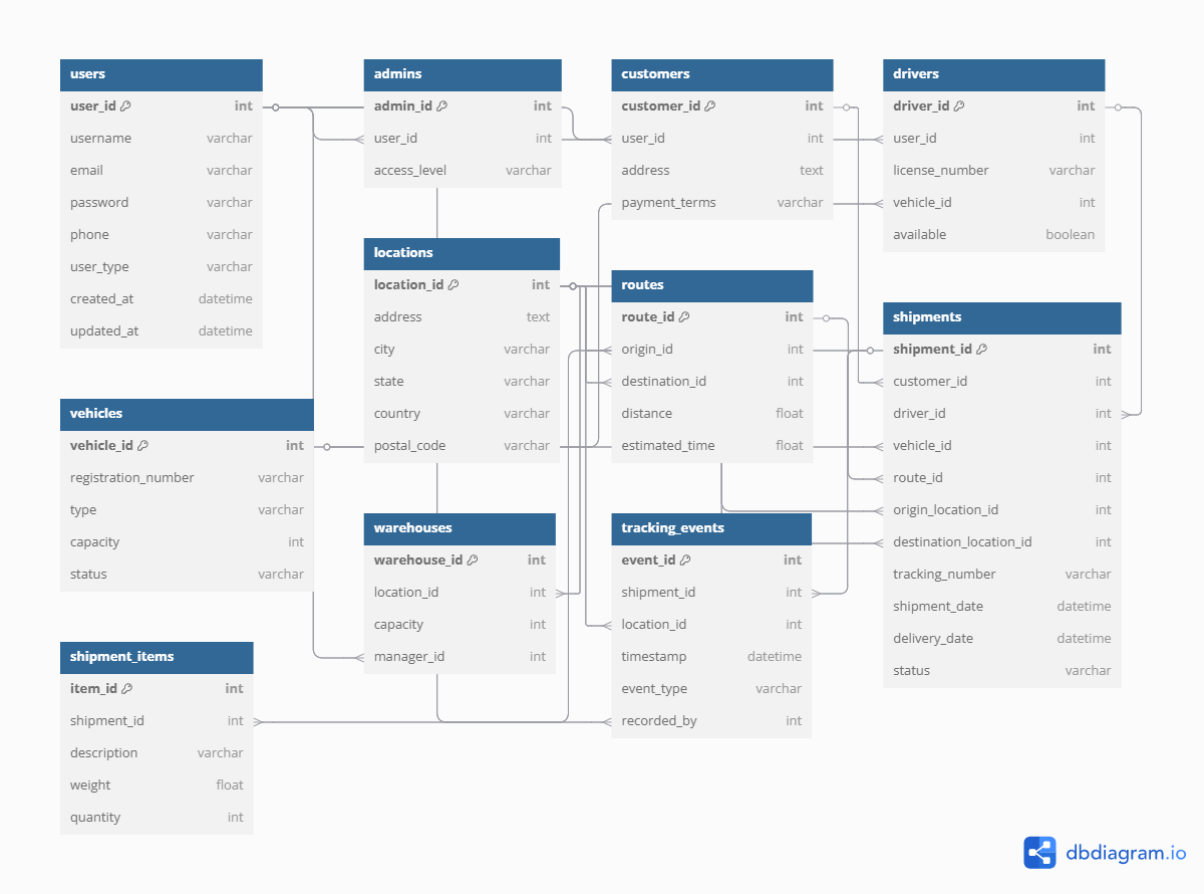

Figure 4.2: Schema Diagram

- Eliminates data redundancy
- Ensures data integrity
- Follows relational database best practices
- Provides flexibility for future schema evolution and scaling

4.4.1 First Normal Form (1NF)

To achieve 1NF:

- All tables were designed to have a primary key.
- Every column holds atomic (indivisible) values.
- No repeating groups or arrays exist in any table.

Transformations applied:

- Warehouses.operating_hours was split into a new Warehouse_Operating_Hours table.
- Drivers.training_certification became the Driver_Certifications table.

Table Name	Primary Key	Foreign Key(s)	Purpose
Users	user_id	—	Stores account details of all users (admin, driver, customer)
Drivers	driver_id	user_id → Users(user_id)	Stores driver-specific info like license and medical status
Customers	customer_id	user_id → Users(user_id)	Stores customer company, credit, and payment info
Vehicles	vehicle_id	—	Stores vehicle details and availability status
Locations	location_id	—	Stores geographical location data for origin, destination, warehouses, etc.
Shipments	shipment_id	customer_id → Customers origin_id → Locations destination_id → Locations route_id → Routes vehicle_id → Vehicles driver_id → Drivers	Tracks shipment lifecycle including origin, destination, route, vehicle, driver
Shipment_Items	item_id	shipment_id → Shipments(shipment_id)	Stores item-wise details for each shipment
Tracking_Events	event_id	shipment_id → Shipments location_id → Locations recorded_by → Users	Logs pickup, delivery, delays, and other shipment events with time/location
Routes	route_id	—	(Optional) Stores pre-defined routes between locations

Figure 4.3: Schema Table

- Shipments.special_instructions was normalized into Shipment_Instructions.

4.4.2 Second Normal Form (2NF)

The schema satisfies 1NF and was further improved by eliminating partial dependencies:

- All non-key attributes depend on the entire composite primary key, where applicable.

Example:

- In the original Waypoints, sequence_number was dependent only on route_id. This was corrected by introducing a separate Route_Sequence table.
- Additionally, a new Event_Locations table was created to separate location dependencies from the Tracking_Events table.

4.4.3 Third Normal Form (3NF)

To conform to 3NF:

- The schema satisfies 2NF.

- It removes transitive dependencies, where non-key attributes depended on other non-key attributes.

Examples:

- Locations: city, state, and country were originally dependent on postal code. This was normalized into a new Postal_Regions table.
- Vehicles: make, model, and year were moved into a Vehicle_Models table.
- Users: email and username were separated into User_Credentials and User_Contact tables.

4.4.4 Boyce-Codd Normal Form (BCNF)

The final step ensured that:

- For every functional dependency X $\rightarrow$ Y, X is a superkey.

Transformations:

- Drivers: Since license_number is unique, a Driver_Licenses table was introduced with license_number as a key.
- Shipments: The unique tracking_number was extracted into a new Shipment_Tracking table.

4.4.5 Schema Design Justification

The final schema comprises 22 normalized tables, designed for optimal relational structure:

- Reduces redundancy by isolating repeating or multi-valued fields into separate entities.
- Maintains strong referential integrity using well-defined foreign keys.
- Improves scalability, allowing individual components like shipment tracking, driver records, or location data to be updated or expanded independently.
- Enhances data quality and consistency through enforcement of functional dependencies and normalization principles.

4.4.6 Final Reduced Schema

- Users: user_id, full_name, user_type, status, created_at
- User_Credentials: user_id, username, password, last_login
- User_Contact: user_id, email, phone
- Admins: admin_id, user_id, access_level
- Customers: customer_id, user_id, company_name, tax_id, credit_limit, payment_terms
- Drivers: driver_id, user_id, medical_check_date, status
- Driver_Licenses: license_number, driver_id, license_expiry, issuing_authority
- Driver_Certifications: certification_id, driver_id, certification_name, issuing_authority, issue_date, expiry_date
- Postal_Regions: postal_code, city, state, country
- Locations: location_id, address, postal_code, latitude, longitude, location_type
- Warehouses: warehouse_id, location_id, warehouse_name, capacity, current_occupancy, manager_id
- Warehouse_Operating_Hours: operating_hours_id, warehouse_id, day_of_week, opening_time, closing_time
- Vehicle_Models: model_id, make, model, year, capacity_kg, vehicle_type
- Vehicles: vehicle_id, license_plate, model_id, status, current_location_id, last_inspection_date
- Routes: route_id, route_name, origin_id, destination_id, distance_km, estimated_duration_min, status, hazard_level
- Route_Sequence: route_id, sequence_count, is_circular
- Waypoints: waypoint_id, route_id, location_id, estimated_arrival, estimated_departure
- Shipments: shipment_id, customer_id, origin_id, destination_id, route_id, vehicle_id, driver_id, status, total_weight, total_volume, shipment_value, insurance_required, created_at, pickup_date, estimated_delivery, actual_delivery
- Shipment_Tracking: tracking_number, shipment_id, issued_date
- Shipment_Items: item_id, shipment_id, description, quantity, weight, volume, item_value, is_hazardous, is_fragile

- Shipment_Instructions: instruction_id, shipment_id, priority, instruction_type, details
- Tracking_Events: event_id, shipment_id, event_type, recorded_by, event_timestamp, notes
- Event_Locations: event_id, location_id

Chapter 5

Methodology

The EZTransport System is a centralized, web-based solution developed to efficiently manage shipments, vehicles, drivers, customers, waypoints, and notifications for logistics service providers. The methodology adopted in building this system follows a structured approach involving requirement analysis, database design, system implementation, and rigorous testing to ensure functionality, scalability, and reliability.

5.1 Requirement Analysis

The core objective was to build a unified logistics management system that allows:

- Customers to place shipment requests and receive real-time tracking updates.
- Administrators to assign vehicles and drivers, manage shipment statuses, and oversee system operations.
- Drivers to view assigned shipments and update waypoint statuses as deliveries progress.

Requirements were gathered through analysis of modern logistics workflows, transport safety standards, and feedback from hypothetical logistics company operations.

5.2 Database Design

A relational database model was adopted using Oracle SQL or MySQL, offering robust support for transactions and complex queries. The schema was normalized to Third Normal Form (3NF) to eliminate data redundancy and ensure consistent, accurate data relationships.

Key tables include:

- Users: Stores login credentials and role references.
- User_Roles: Defines roles such as Admin, Driver, and Customer.
- Shipments: Records all customer shipment requests.

- Drivers: Maintains information on available drivers and qualifications.
- Vehicles: Stores vehicle details and their operational status.
- Waypoints: Tracks shipment journey updates.
- Notifications: Stores automated messages triggered by shipment progress.
- Packages: Includes package details like weight, dimensions, and safety tags (hazardous/fragile).
- Locations: Maps cities, warehouses, and delivery points.

Foreign keys were used extensively to preserve referential integrity across related entities.

5.3 Data Relationships

- users are associated with user_roles.
- shipments link to drivers, vehicles, customers, and packages.
- waypoints track a shipment's progress through specific locations.
- notifications are triggered by changes in waypoint statuses.
- vehicles and drivers can be reused, with checks for availability.
- packages can be marked as hazardous or fragile, prompting special handling.

5.4 Technology Stack

- Backend: Python (Flask) or Node.js for server-side logic
- Frontend: React.js with HTML/CSS and Bootstrap for responsive UI
- Database: Oracle SQL or MySQL
- Authentication: Secure login system with hashed passwords using bcrypt

5.5 Implementation Steps

1. Database Setup: SQL scripts to define schema, keys, and constraints.
2. Role Management Panel: Admin UI for managing users, vehicles, and drivers.
3. Shipment Module: Interfaces for customers to place requests and for admins to assign resources.
4. Waypoint Tracking: Driver interface for logging shipment progress and timestamping.

5. Notification Engine: Auto-generates updates sent to customers upon status changes.
6. Sensitive Package Handling: Admin feature to flag items as hazardous or fragile.
7. Reports and Logs: Exportable summaries of shipment histories and driver performance.

5.6 Testing & Validation

- Unit Testing: Verified that individual components (e.g., form submissions, role actions) function correctly.
- Integration Testing: Checked coordination between modules like shipments and waypoints, or users and roles.
- User Acceptance Testing (UAT): Feedback from hypothetical logistics coordinators and customers was used to refine features and improve usability.

Chapter 6

Results

This section presents the key functionalities of EZTransport through screenshots and descriptions, demonstrating the system's role-based access control, tracking features, and CRUD (Create, Read, Update, Delete) operations.

6.1 Login and User Registration

Users can log in as Admin, Customer, or Driver using secure authentication.

Role-based redirection ensures access to appropriate dashboards.

Customers and drivers can register via a form with validation.

Admins approve driver accounts to ensure legitimacy.

6.2 Admin Dashboard & CRUD Operations

Displays real-time metrics (shipments, drivers, vehicles).

Quick actions for creating shipments, adding drivers, and managing vehicles.

Admins can:

- Create/Edit Shipments: Assign drivers, vehicles, and routes.
- Track Status: Monitor "Pending," "In Transit," and "Delivered" shipments.
- Filter/Search: Use tracking IDs or dates for quick access.

Admins add/edit vehicles with details like license plate, capacity, and inspection dates.

6.3 Customer Dashboard & Tracking

- Personal Info: Edit contact/company details.
- Shipment Summary: View delivery statuses (e.g., "In Transit").

Customers track shipments in real-time with waypoint updates (e.g., ”Departed Warehouse”). Displays estimated vs. actual delivery dates.

6.4 Driver Dashboard & Shipment Updates

- Assigned Shipments: List of pending/in-transit deliveries.
- Vehicle Info: License, capacity, and inspection status.

Drivers update shipment status (e.g., ”Departed,” ”Delivered”) via a mobile-friendly form. Triggers automated notifications to customers.

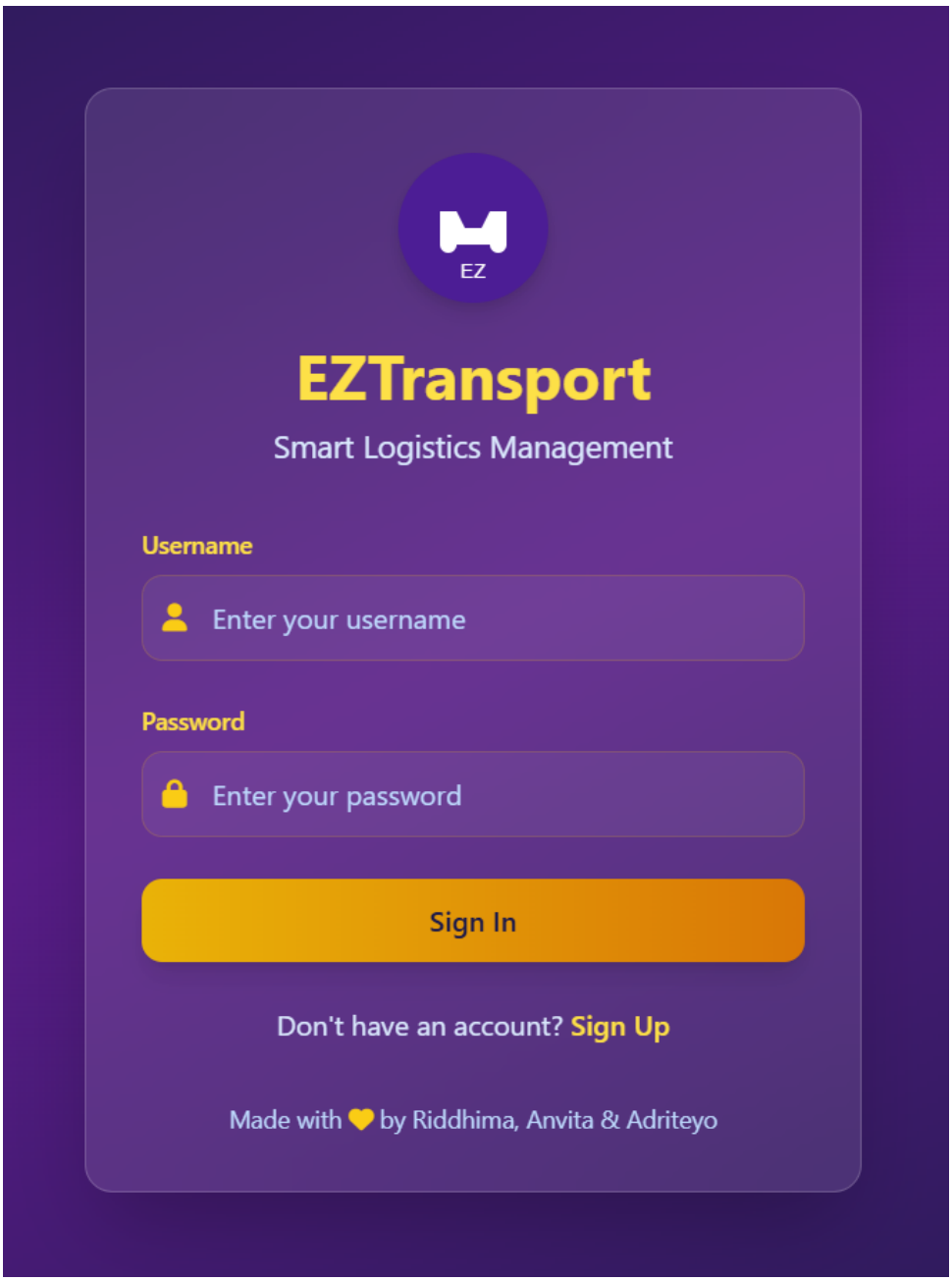

Figure 6.1: Login Page

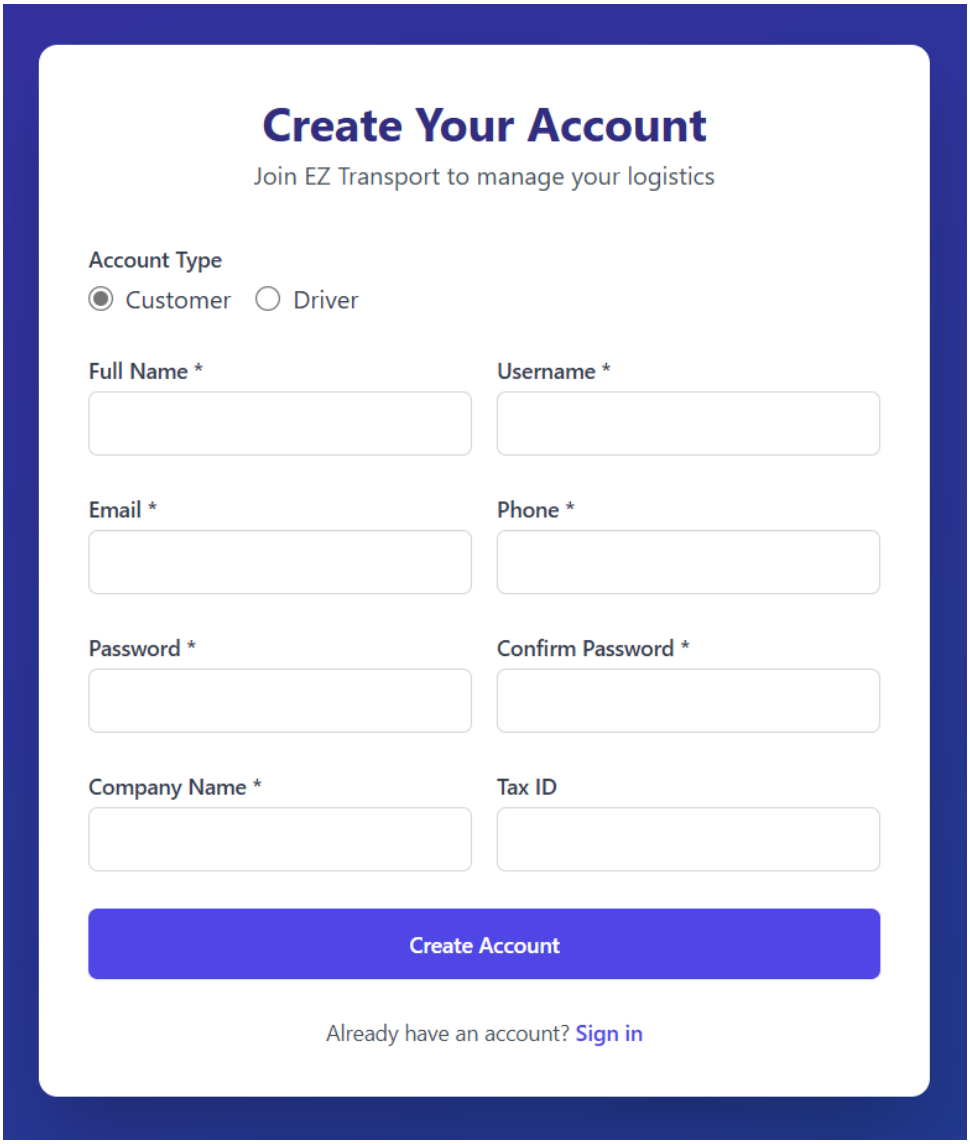

Figure 6.2: Customer/Driver Signup Page

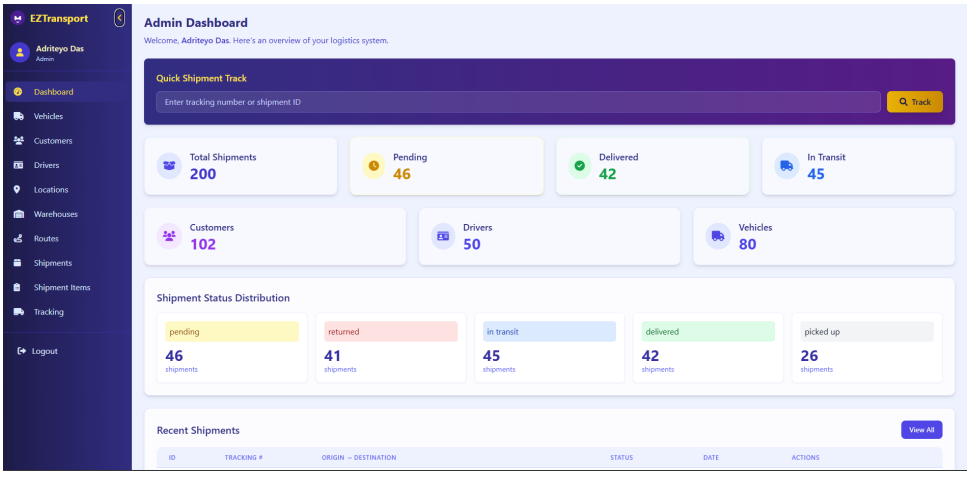

Figure 6.3: Admin Dashboard

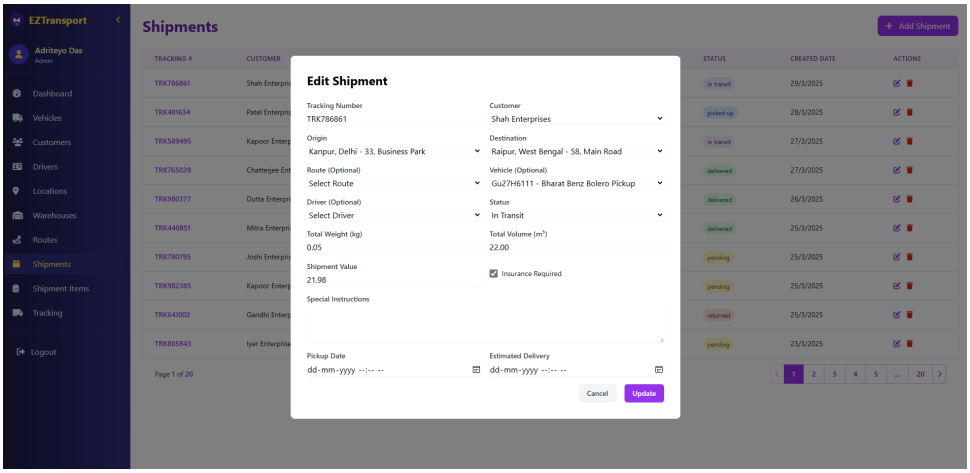

Figure 6.4: Admin Managing Shipments (CRUD)

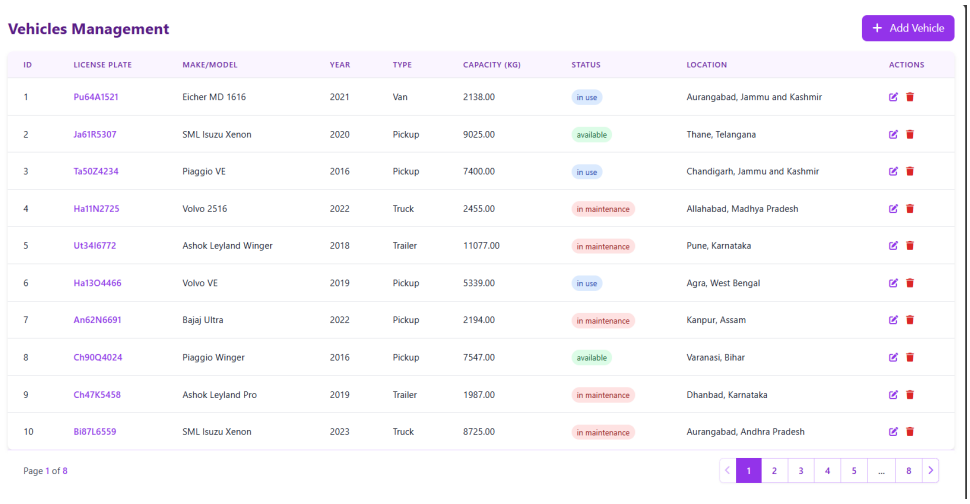

Figure 6.5: Vehicle Management

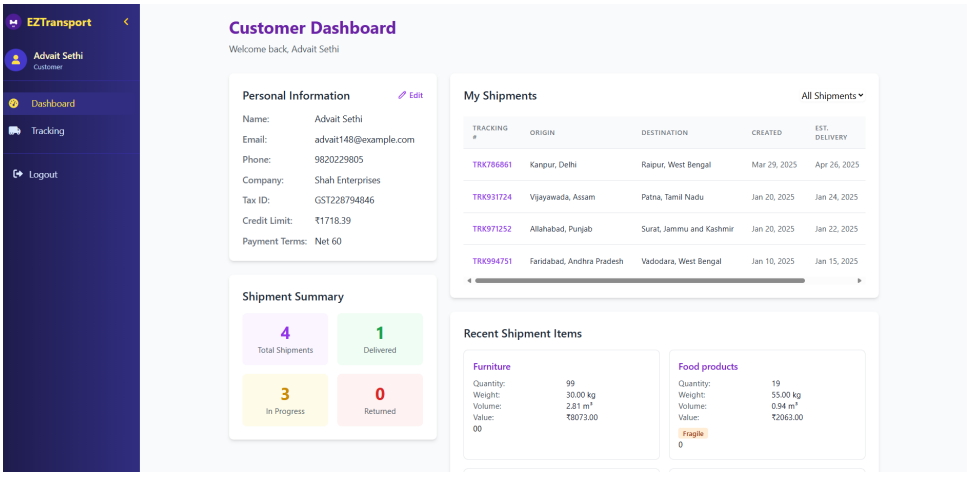

Figure 6.6: Customer Dashboard

Figure 6.7: Shipment Tracking

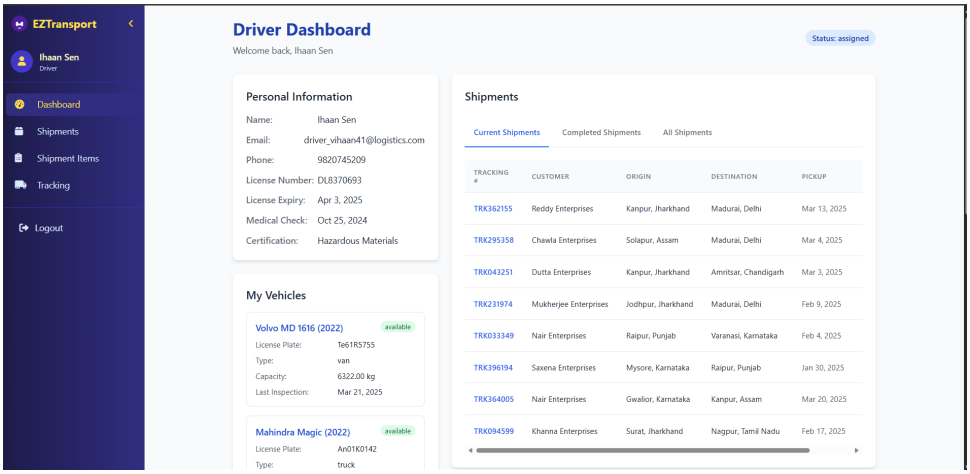

Figure 6.8: Driver Dashboard

EVENT ID	SHIPMENT ID	EVENT TYPE	LOCATION	TIMESTAMP	NOTES	ACTIONS
518	172	Departure	Ghaziabad, Haryana	20/4/2025, 11:32:08 pm		
517	172	Delay	Mysore, Jharkhand	20/4/2025, 10:29:42 pm		
477	185	Delivery	Chandigarh, Gujarat	9/4/2025, 5:30:00 am	-	
450	172	Pickup	Meerut, Tamil Nadu	8/4/2025, 5:30:00 am	-	
400	152	Departure	Hyderabad, Bihar	8/4/2025, 5:30:00 am	-	
262	100	Departure	Varanasi, Bihar	6/4/2025, 5:30:00 am		
479	185	Delay	Agra, Punjab	6/4/2025, 5:30:00 am	-	
461	176	Issue	Coimbatore, Andhra Pradesh	6/4/2025, 5:30:00 am	-	
449	172	Departure	Surat, Madhya Pradesh	6/4/2025, 5:30:00 am		
261	100	Pickup	Indore, Telangana	5/4/2025, 5:30:00 am		

Figure 6.9: Driver Waypoint Logging

Chapter 7

Conclusion

7.1 Conclusion

The successful design and implementation of EZTransport, an integrated transport and logistics management system, represents a substantial advancement in how modern logistics operations are conducted. By digitizing and automating critical processes such as shipment creation, vehicle and driver assignment, waypoint tracking, and customer notifications, EZTransport offers a robust and scalable solution for logistics companies striving for operational excellence.

A major strength of EZTransport lies in its modular and role-based architecture, which empowers Administrators, Drivers, and Customers to interact with the system in secure and intuitive ways. Whether it's assigning resources, updating shipment status, or tracking packages in real time, each user is provided with a streamlined experience tailored to their role.

The platform's real-time data synchronization, secure authentication protocols, and responsive UI make EZTransport a strategic asset for logistics providers. Its normalized database design ensures data integrity and enables efficient querying across multiple entities such as shipments, waypoints, packages, and notifications. Features like hazardous and fragile item tagging further enhance operational safety and accuracy.

In summary, EZTransport significantly improves logistics transparency, minimizes manual errors, and accelerates delivery workflows, making it an essential tool for modern supply chain ecosystems.

7.2 Future Works

- **Mobile App for Drivers and Customers**
  Developing a mobile version of EZTransport will allow drivers to update shipment status directly from the field and enable customers to track packages and receive notifications on the go. This enhancement will increase system accessibility and improve responsiveness across all user roles.

- **Geo-Tracking and Route Optimization**
  Integrating GPS-based tracking with real-time route optimization algorithms can help drivers follow the most efficient delivery paths, reduce fuel consumption, and provide precise ETA predictions to customers.

- **AI-Based Shipment Prediction and Analytics**
  Leveraging AI and machine learning can enable predictive analytics for shipment delays, driver performance metrics, and demand forecasting. These insights can support strategic planning and help companies optimize their operations further.

- **Integration with RFID and Barcode Scanning**
  Adding RFID and barcode scanning support will help automate waypoint check-ins, enhance package traceability, and reduce manual data entry errors during loading, unloading, and delivery.

- **Dynamic Pricing and Cost Estimation**
  Implementing a dynamic pricing engine that calculates shipment costs based on distance, weight, type (fragile/hazardous), and urgency will provide customers with transparent billing and help companies maintain profitability.

- **Third-Party Logistics (3PL) API Integration**
  Future updates may include APIs for integration with external 3PL services, enabling businesses to outsource or partner with regional carriers while still managing all operations via EZTransport.

- **Cloud Hosting for Multi-Branch Scalability**
  Deploying EZTransport to a cloud platform will ensure data availability, real-time collaboration across regional branches, and automatic backups, making it ideal for growing logistics networks.

- **Regulatory Compliance Automation**
  Automating compliance checks for transporting hazardous goods, vehicle inspection schedules, and driver certifications can help logistics companies maintain safety standards and avoid penalties.

Bibliography

[1] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[2] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2015.

[3] J. Teorey, S. Lightstone, and T. Nadeau, *Database Modeling and Design*, 5th ed. Morgan Kaufmann, 2011.

[4] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. O'Reilly Media, 2018.

[5] A. Banks and E. Porcello, *Learning React: A Hands-On Guide to Building Web Applications Using React and Redux*. O'Reilly Media, 2020.

[6] N. Jain, *Building a full-stack web app with flask and react*, Real Python, 2021.

[7] Mozilla Developer Network, *Html, css, and javascript tutorials*, Online. [Online]. Available: `https://developer.mozilla.org`.

[8] freeCodeCamp, *Responsive web design certification*, Online. [Online]. Available: `https://www.freecodecamp.org`.

[9] W3Schools, *Html, css, and js documentation*, Online. [Online]. Available: `https://www.w3schools.com`.

[10] P. Dubois, *MySQL Cookbook*, 4th ed. O'Reilly Media, 2020.

[11] A. Beaulieu, *Learning SQL*, 3rd ed. O'Reilly Media, 2020.

[12] MySQL Documentation, *Mysql 8.0 reference manual*, Online. [Online]. Available: `https://dev.mysql.com/doc/`.

[13] B. Montreuil, "Toward a physical internet: Meeting the global logistics sustainability grand challenge," *Logistics Research*, vol. 3, no. 2-3, pp. 71–87, 2011.

[14] A. Rushton, P. Croucher, and P. Baker, *The Handbook of Logistics and Distribution Management*, 6th ed. Kogan Page, 2022.

[15] M. Christopher, *Logistics & Supply Chain Management*, 5th ed. Pearson UK, 2016.