**A PROJECT REPORT ON : COURIER MANAGEMENT SYSTEM(PORTAL)**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC.


**SUBMITTED BY:**

Jadhav Ajinkya Vijay Kumar

**UNDER THE GUIDENCE OF:**

Mr.Snehal Jadhav

Faculty Member

**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**Hinjawadi,PUNE


# CERTIFICATE

This is to certify that the project work under the title ‘Courier Service Management(Portal)’ is done by Jadhav Ajinkya Vijay Kumar in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Mr.Snehal Jadhav** **Mr. Yogesh Kolhe**

**Project Guide** **Course Co-Coordinator**

Date:16-08-2024

## ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Jadhav Ajinkya Vijay Kumar

0324 PG-DAC

SIIT Pune

# INDEX

## 1. Introduction

The Courier Service Management (Portal) is designed to revolutionize the way courier services operate, enhancing efficiency, reliability, and customer satisfaction. In today's fast-paced world, the demand for swift and accurate delivery services has never been higher. With e-commerce booming and businesses expanding globally, the need for an advanced courier management system is critical.

The Courier Service Portal addresses these needs by providing a digital platform where all courier-related activities can be managed seamlessly. From order creation to tracking and payment processing, the system offers a comprehensive solution that integrates the latest technologies to ensure smooth operations. The system is designed to reduce human error, streamline workflows, and provide real-time information to both employees and customers, thereby significantly improving service quality.

This project represents a significant leap forward for courier service providers, allowing them to maintain a competitive edge in a crowded market. The system is built with scalability in mind, ensuring that it can grow alongside the business, adapting to increasing volumes and changing customer demands.

The Courier Service Portal System leverages a robust technological stack that includes:

- **Front-End (React)**: React is a popular JavaScript library known for its efficiency and flexibility in building user interfaces. Its component-based architecture allows for the reuse of code and modular design, making the development process more efficient. React also provides powerful tools for managing the state of the application, ensuring that the user interface reflects the most current data at all times.
- **Back-End (Spring Boot)**: Spring Boot, a framework within the larger Spring ecosystem, is chosen for its ability to simplify the development of Java-based enterprise applications. It allows developers to create production-

ready applications with minimal configuration. The microservices architecture of Spring Boot ensures that each component of the Courier Service Portal System can be developed, deployed, and scaled independently, providing greater flexibility and resilience.

- **Database (MySQL)**: MySQL, a relational database management system, is employed to manage the vast amounts of data generated by the Courier Service Portal System. MySQL's robustness, scalability, and support for complex queries make it an ideal choice for this project. The database schema is designed to handle complex relationships between entities like Users, Orders, and Shipments, ensuring data integrity and efficient retrieval.

Each of these technologies was selected for its ability to contribute to a seamless, reliable, and scalable system that can handle the complexities of modern courier services. Together, they form a cohesive and powerful platform that meets the needs of both the business and its customers.

## 2. Project Description

The Courier Service Portal System is structured around three primary modules: Admin, Employee, and User. Each module is designed to cater to the specific needs of the different stakeholders involved in courier services.

- **Admin Module**: This module is designed for system administrators who are responsible for managing the overall operations of the courier service. It includes functionalities for managing branches, employees, and courier orders.
- **Employee Module**: Tailored for staff members, this module allows employees to handle day-to-day operations such as managing shipments, updating order statuses, and processing payments.

- **User Module**: This module is designed for customers who use the courier service to send and receive packages. It provides functionalities for creating orders, tracking shipments, and managing user profiles.

## 3. Modules And Their Functionalities

### Admin Module

The Admin module provides comprehensive control over the courier service operations. Key functionalities include:

- **Profile Management**: Admins can view and update their personal information.
- **Courier Order Management**: Admins can view and filter courier orders based on their status (e.g., pending, shipped, delivered, cancelled).
- **Branch Management**: Admins can view, add, update, or delete branches to manage service areas.
- **Employee Management**: Admins can manage employee information, including adding new employees, updating details, and deleting employees as needed.
- **Security Management**: Admins have the ability to update passwords to ensure the security of their accounts.

### Employee Module

The Employee module is designed to streamline the workflow of staff members. Key functionalities include:

- **Profile Management**: Employees can view and update their personal details, including branch assignment.
- **Order Processing**: Employees can view all courier orders, filter them by status, and update the order status as needed.

- **Shipment Management**: Employees can create new shipment orders and track orders using a tracking number.
- **Payment Processing**: Employees can process payments for shipments through an integrated payment gateway.

**User Module**

The User module focuses on providing a user-friendly interface for customers. Key functionalities include:

- **Profile Management**: Users can view and manage their personal information.
- **Order Creation**: Users can create new shipment orders and calculate the cost of shipments based on package details and destination.
- **Tracking and History**: Users can track the status of current shipments and view their shipping history.
- **Branch Information**: Users can access details about different branches, including location and contact information.

**4. System Architecture**

The system architecture of the Courier Service Portal is designed to ensure seamless integration between the front-end, back-end, and database. The front-end, built using React, provides a dynamic and responsive user interface. The back-end, developed with Spring Boot, handles business logic and interacts with the MySQL database to store and retrieve data.

**Key Components:**

- **Front-End (React)**: The front-end is responsible for rendering the user interface and handling user interactions. It communicates with the back-end via RESTful APIs.
- **Back-End (Spring Boot)**: The back-end manages the application's business logic, processes user requests, and interacts with the database.

- **Database (MySQL)**: MySQL is used to store all the data related to users, orders, branches, and employees.

### 5. Database Design(MySQL):

**Tables:**

```
mysql> show tables;
+---------------+
| Tables_in_cms |
+---------------+
| addresses     |
| branches      |
| complaints    |
| customers     |
| employees     |
| orders        |
| parcels       |
| payments      |
| users         |
+---------------+
9 rows in set (0.03 sec)
```

**Table 1:**

```
mysql> desc addresses;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| address_id  | bigint       | NO   | PRI | NULL    | auto_increment |
| address     | varchar(255) | YES  |     | NULL    |                |
| address_2   | varchar(255) | YES  |     | NULL    |                |
| city        | varchar(255) | YES  |     | NULL    |                |
| country     | varchar(255) | YES  |     | NULL    |                |
| postal_code | int          | YES  |     | NULL    |                |
| state       | varchar(255) | YES  |     | NULL    |                |
| user_id     | bigint       | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

**Table 2:**

```
mysql> desc branches;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| branch_id   | bigint       | NO   | PRI | NULL    | auto_increment |
| branch_name | varchar(255) | YES  |     | NULL    |                |
| email       | varchar(255) | YES  |     | NULL    |                |
| phone       | varchar(255) | YES  |     | NULL    |                |
| address_id  | bigint       | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

**Table 3:**

```
mysql> desc complaints;
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| complaint_id     | bigint       | NO   | PRI | NULL    | auto_increment |
| complaint_date   | date         | YES  |     | NULL    |                |
| complaint_status | varchar(255) | YES  |     | NULL    |                |
| description      | varchar(255) | YES  |     | NULL    |                |
| remark           | varchar(255) | YES  |     | NULL    |                |
| order_id         | bigint       | YES  | MUL | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

**Table 4:**

```
mysql> desc customers;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| customer_id   | bigint       | NO   | PRI | NULL    | auto_increment |
| customer_type | varchar(255) | YES  |     | NULL    |                |
| email         | varchar(255) | YES  |     | NULL    |                |
| first_name    | varchar(255) | YES  |     | NULL    |                |
| last_name     | varchar(255) | YES  |     | NULL    |                |
| phone         | varchar(255) | YES  |     | NULL    |                |
| address_id    | bigint       | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

**Table 5:**

```
mysql> desc employees;
+-----------+--------+------+-----+---------+-------+
| Field     | Type   | Null | Key | Default | Extra |
+-----------+--------+------+-----+---------+-------+
| hire_date | date   | YES  |     | NULL    |       |
| salary    | double | NO   |     | NULL    |       |
| user_id   | bigint | NO   | PRI | NULL    |       |
| branch_id | bigint | YES  | MUL | NULL    |       |
+-----------+--------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

**Table 6:**

```
mysql> desc orders;
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| order_id           | bigint      | NO   | PRI | NULL    | auto_increment |
| declared_value     | double      | YES  |     | NULL    |                |
| delivery_date      | date        | YES  |     | NULL    |                |
| order_date         | date        | YES  |     | NULL    |                |
| service_type       | varchar(255)| YES  |     | NULL    |                |
| status             | varchar(255)| YES  |     | NULL    |                |
| tracking_number    | varchar(255)| YES  |     | NULL    |                |
| parcel_id_parcel_id| bigint      | YES  | MUL | NULL    |                |
| payment_id         | bigint      | YES  | MUL | NULL    |                |
| receiver_id        | bigint      | YES  | MUL | NULL    |                |
| sender_id          | bigint      | YES  | MUL | NULL    |                |
| user_id            | bigint      | YES  | MUL | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
12 rows in set (0.01 sec)
```

**Table 7:**

```
mysql> desc parcels;
+-----------+--------+------+-----+---------+----------------+
| Field     | Type   | Null | Key | Default | Extra          |
+-----------+--------+------+-----+---------+----------------+
| parcel_id | bigint | NO   | PRI | NULL    | auto_increment |
| height    | int    | NO   |     | NULL    |                |
| length    | int    | NO   |     | NULL    |                |
| weight    | float  | NO   |     | NULL    |                |
| width     | int    | NO   |     | NULL    |                |
+-----------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

**Table 8:**

```
mysql> desc payments;
+----------------+--------+------+-----+---------+----------------+
| Field          | Type   | Null | Key | Default | Extra          |
+----------------+--------+------+-----+---------+----------------+
| payment_id     | bigint | NO   | PRI | NULL    | auto_increment |
| amount         | float  | YES  |     | NULL    |                |
| payment_date   | date   | YES  |     | NULL    |                |
| payment_method | int    | YES  |     | NULL    |                |
+----------------+--------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

**Table 9:**

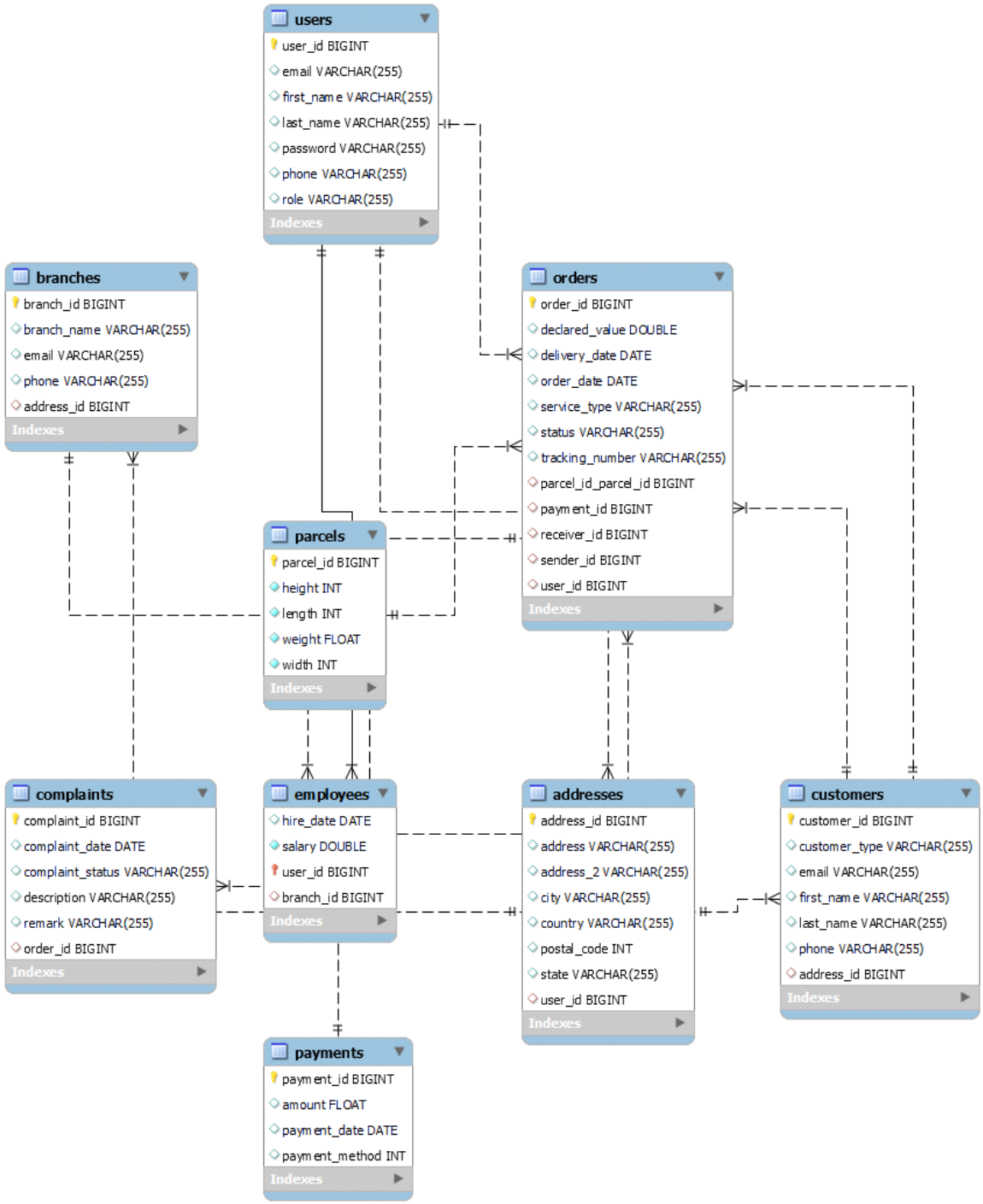
```
mysql> desc users;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| user_id    | bigint       | NO   | PRI | NULL    | auto_increment |
| email      | varchar(255) | YES  | UNI | NULL    |                |
| first_name | varchar(255) | YES  |     | NULL    |                |
| last_name  | varchar(255) | YES  |     | NULL    |                |
| password   | varchar(255) | YES  |     | NULL    |                |
| phone      | varchar(255) | YES  | UNI | NULL    |                |
| role       | varchar(255) | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)
```

## 6. Entity-Relationship Diagram

### Entity-Relationship Diagram Overview:

The Entity-Relationship (ER) diagram provides a visual representation of the database schema used in the Courier Service Portal System. It illustrates the entities involved in the system, their attributes, and the relationships between them.

- **Entities**: The main entities in the system include User, Admin, Employee, Branch, Order, and Shipment.
- **Relationships**: The relationships between these entities, such as User creating an Order, Admin managing a Branch, and Employee processing a Shipment, are depicted in the diagram.




**Explanation of Entities and Relationships:**

- **User**: Contains attributes such as user ID, name, email, and contact details. A User can create multiple Orders.

- **Admin**: Manages the overall system, including branches and employees. An Admin can oversee multiple Branches.
- **Employee**: Associated with a Branch and responsible for processing Orders. An Employee can handle multiple Shipments.
- **Branch**: Represents the physical locations of the courier service. A Branch can have multiple Employees and handle multiple Orders.
- **Order**: Represents a shipment order created by a User. It includes attributes such as order ID, status, and destination.
- **Shipment**: Represents the actual shipment process of an order, including tracking and delivery details.

**7. Flow Diagrams**

Flow diagrams provide a step-by-step representation of the processes within the Courier Service Portal System.

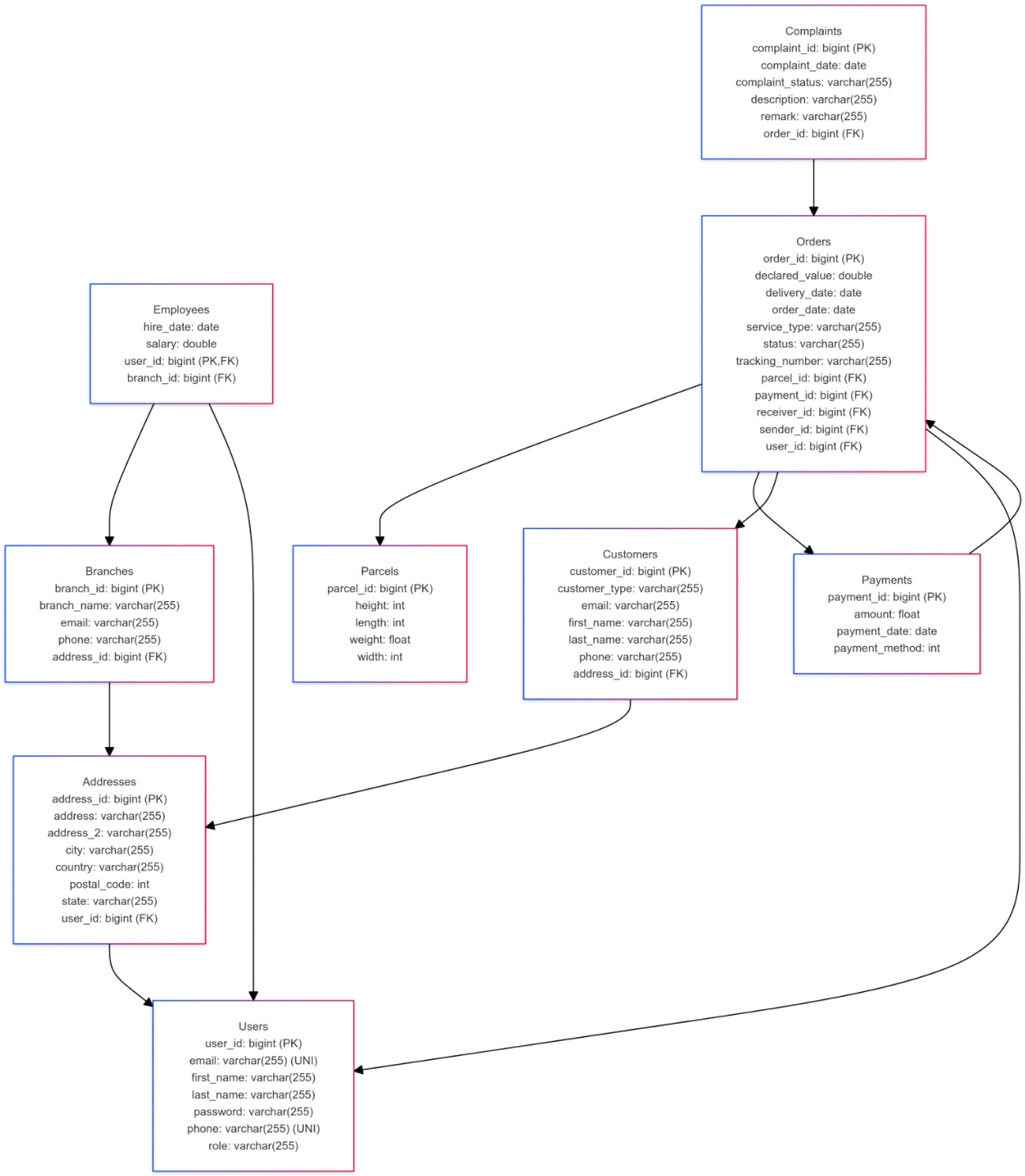
**Order Creation Flow:**

- **User Logs In**: The user logs into the system using their credentials.
- **Create New Order**: The user navigates to the order creation page.
- **Input Details**: The user inputs package details, destination, and other necessary information.
- **Calculate Cost**: The system calculates the cost of shipment based on the provided details.
- **Payment**: The user proceeds to payment through the integrated payment gateway.
- **Order Confirmation**: Upon successful payment, the order is confirmed and added to the system.

**Order Tracking Flow:**

- **User Logs In**: The user logs into the system using their credentials.

- **Track Order**: The user navigates to the order tracking page.
- **Input Tracking Number**: The user inputs the tracking number associated with their order.
- **View Status**: The system retrieves the current status of the order and displays it to the user.
- **Order History**: The user can view the history of all past orders and their statuses.



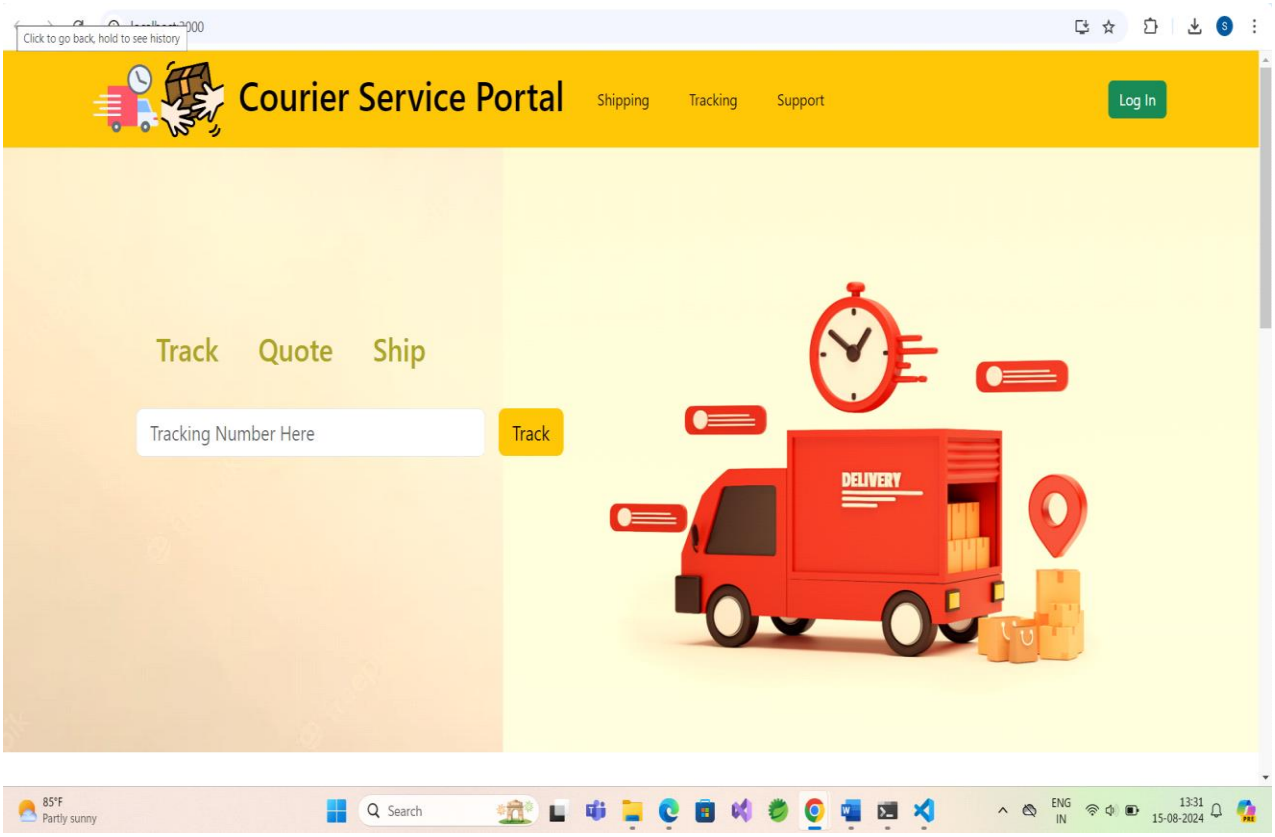

## 8. Conclusion

The Courier Service Portal System is a robust solution for managing courier services. By integrating various modules tailored to the needs of admins, employees, and users, the system ensures efficient operations and a seamless

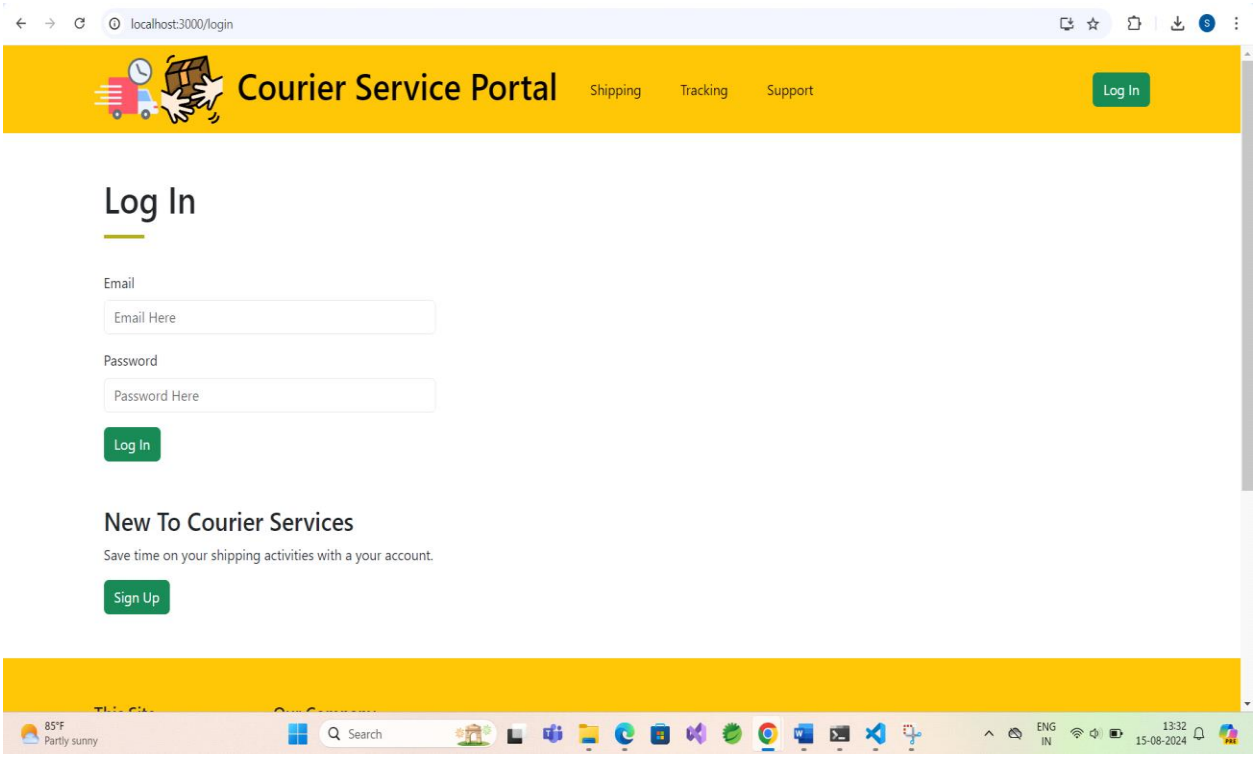
user experience. The use of modern technologies such as React, Spring Boot, and MySQL further enhances the system's performance and scalability. Future enhancements could include the integration of advanced features such as real-time tracking updates, automated notifications, and enhanced security measures.

## 9.Web Page Design:-

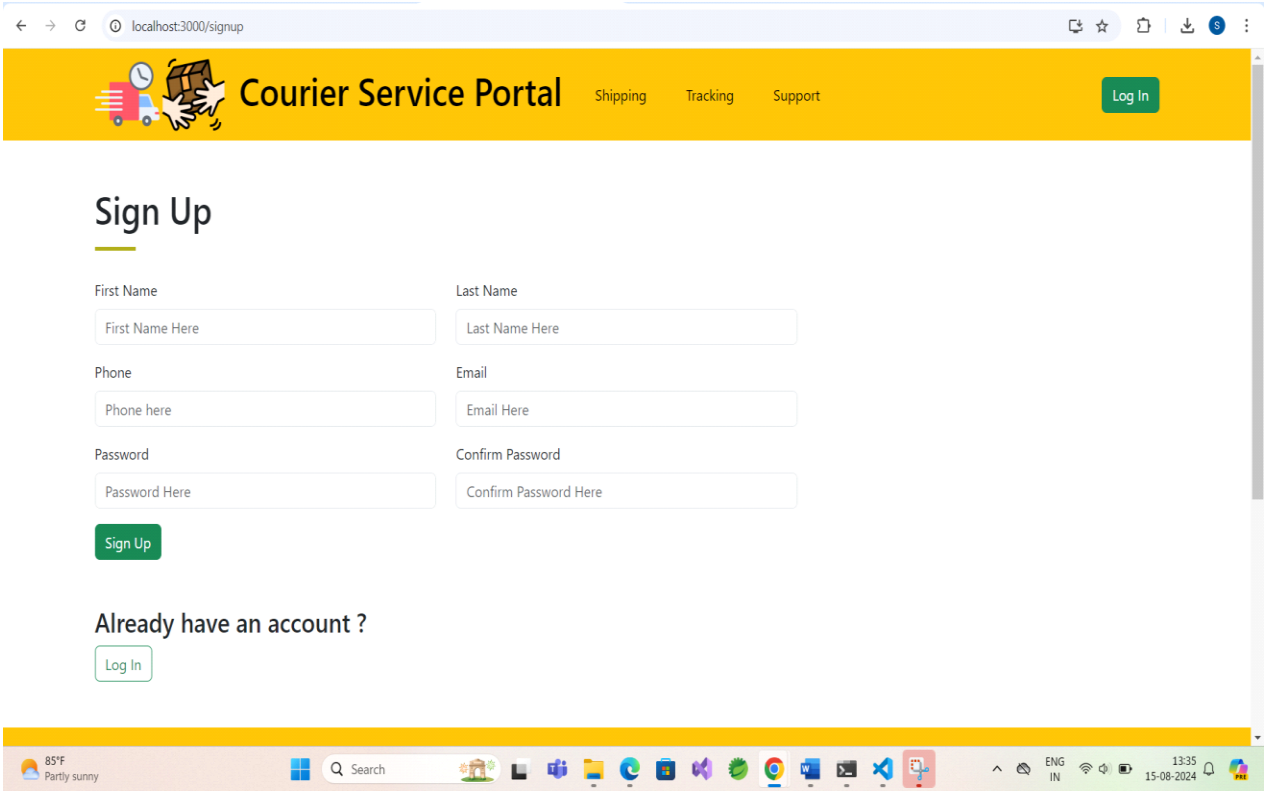
HOME PAGE:



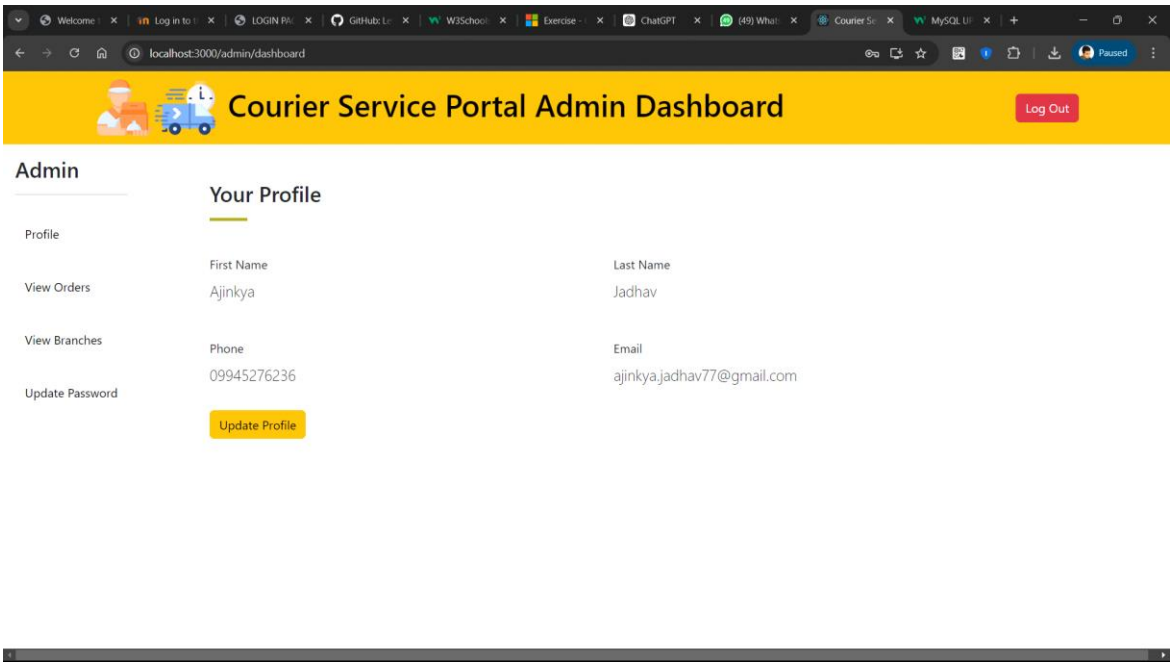
LOGIN PAGE:



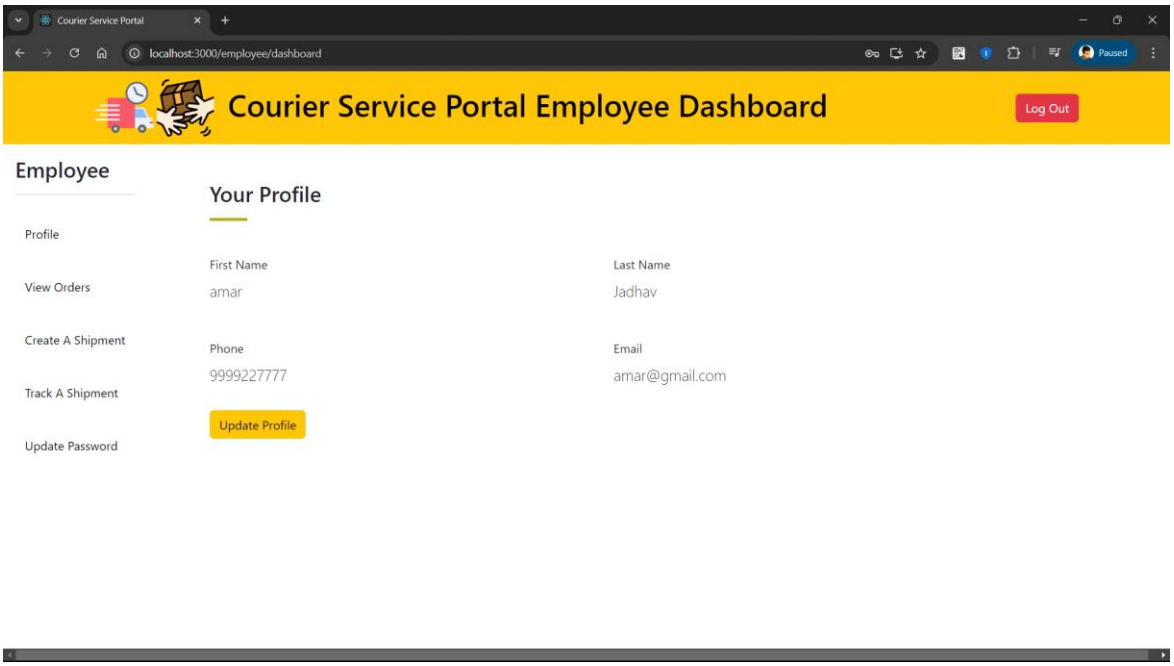

USER REGISTRATION:



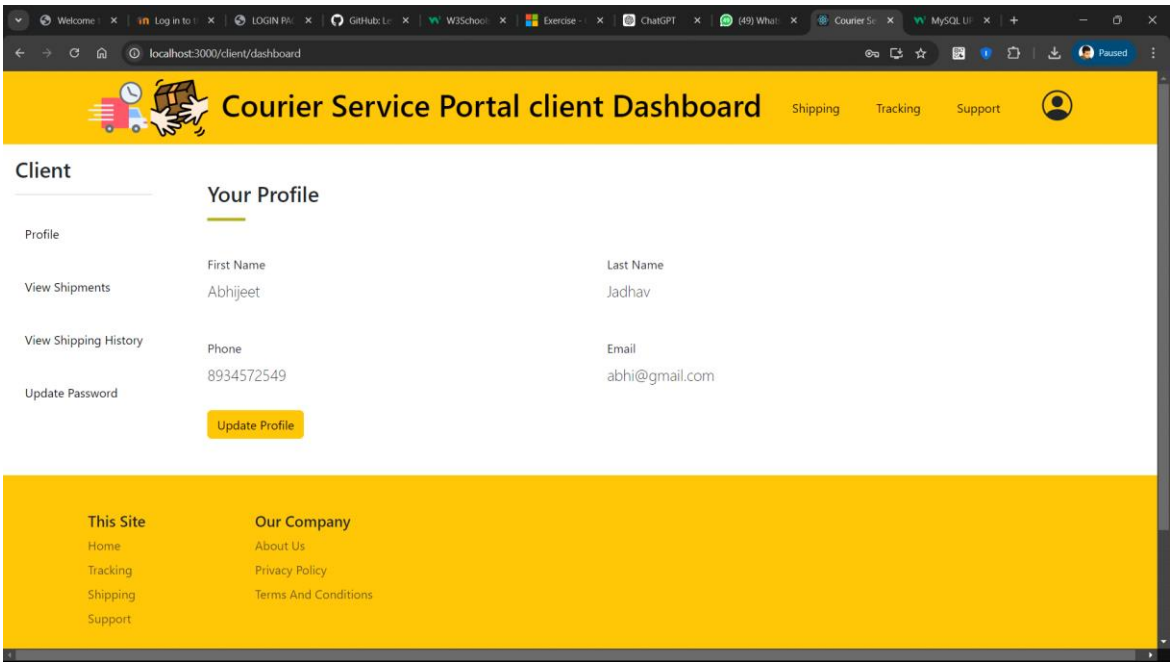

ADMIN DASHBOARD:



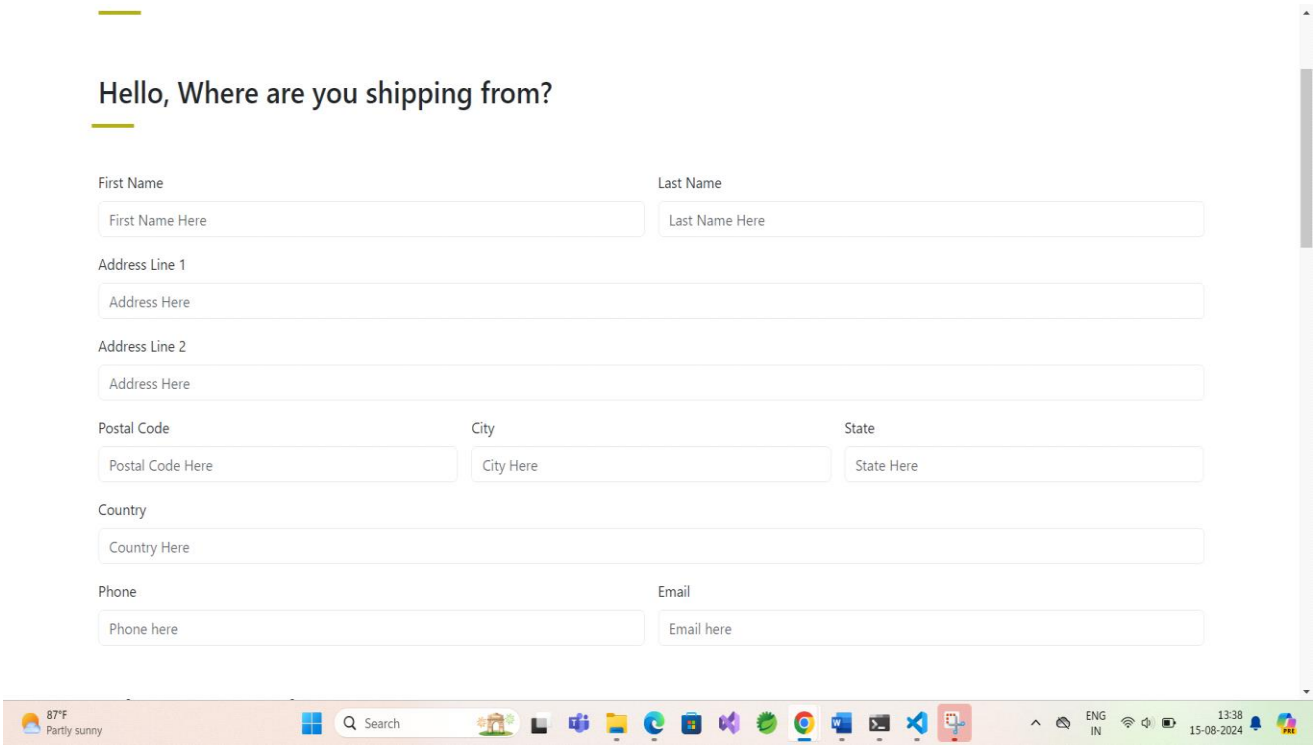
EMPLOYEE DASHBOARD:




CLIENT DASHBOARD:



CREATE SHIPMENT:



## Where is your shipment going?

First Name

First Name Here

Last Name

Last Name Here

Address Line 1

Address Here

Address Line 2

Address Here

Postal Code

Postal Code Here

City

City Here

State

State Here

Country

Country Here

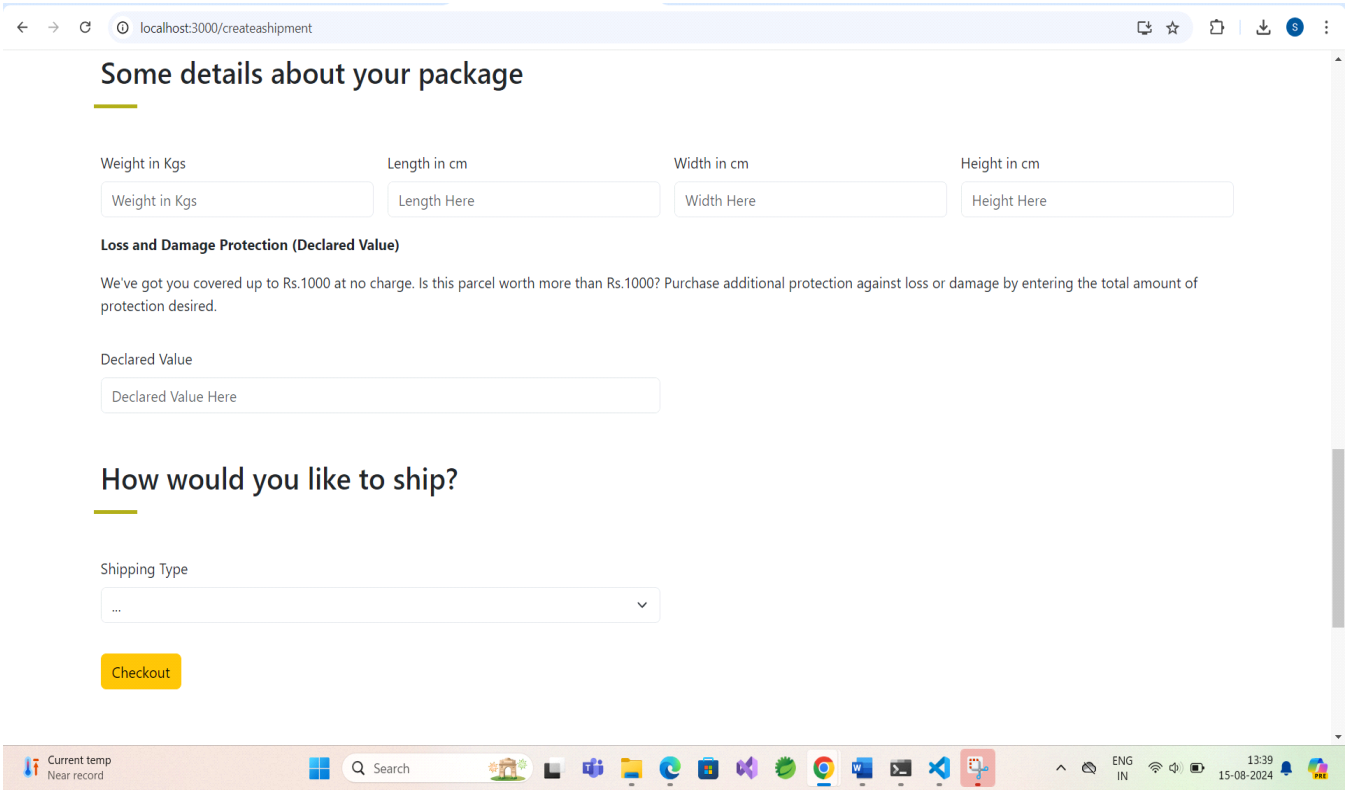
Phone

Phone Here

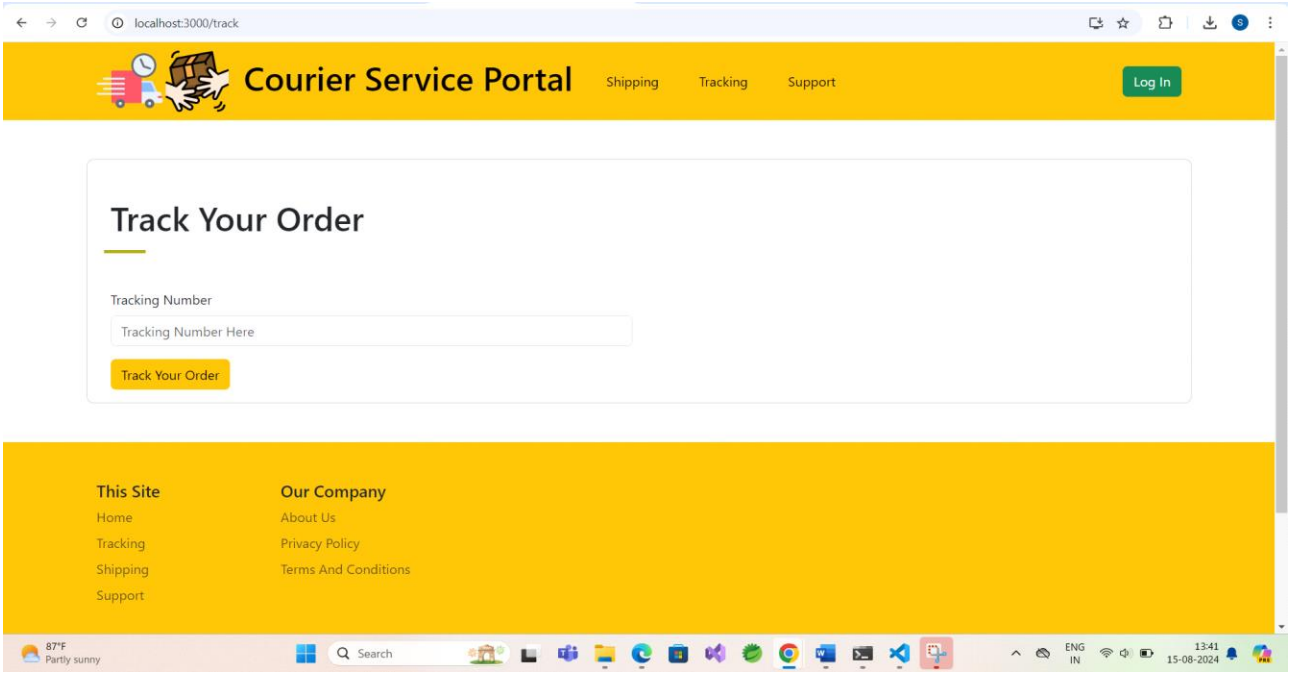
Email

Email here

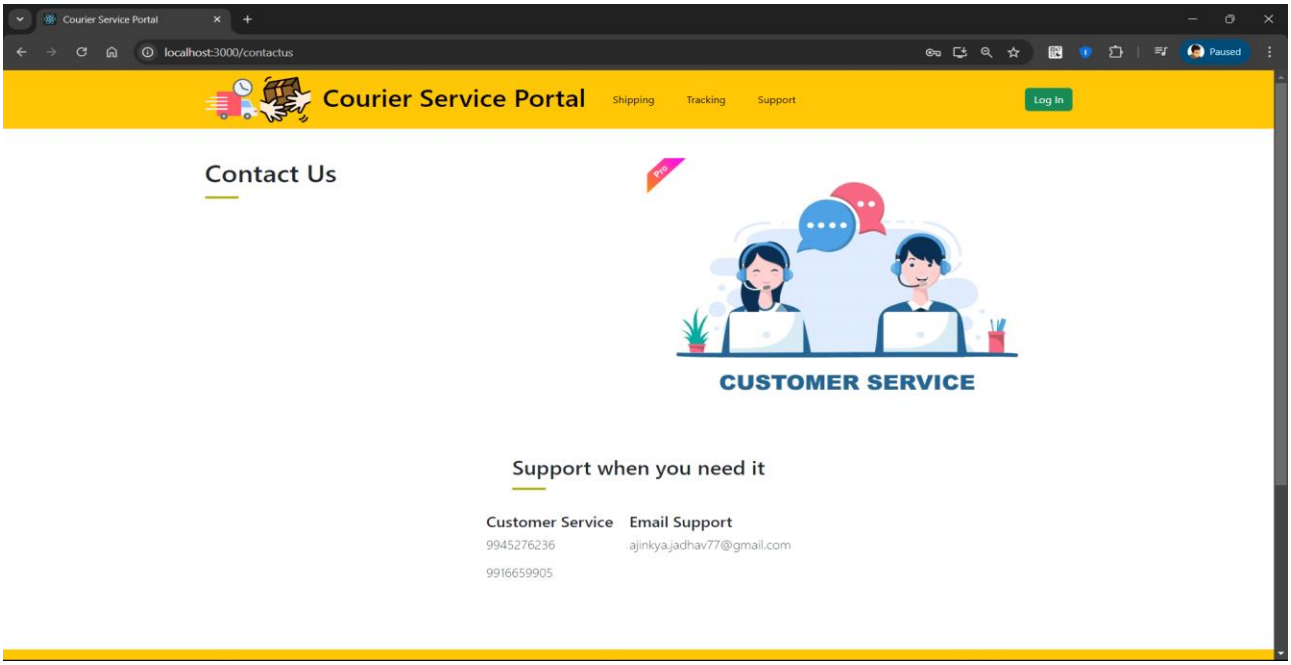
# PACKAGE DELIVERY QUOTATION:



PACKAGE TRACKING:




CUSTOMER SUPPORT :



## 9.REFERENCES


http:// HYPERLINK "http://www.xml101.com:8081/xml/"www.dhl.com

http:// HYPERLINK "http://www.xml101.com:8081/xml/"www.delhivery.com

https://docs.spring.io/spring-boot/documentation.html

https://jwt.io/

https://getbootstrap.com/

https://react.dev/learn

https://dev.mysql.com/doc/

http://www.webdevelopersjournal.com/

http://www.fedex.com

http://www.w3.org

http://www.wikipedia.org