Velammal College of Engineering and Technology, Madurai – 625 009.
(An Autonomous Institution)

Department of Computer Science and Engineering

Academic Year: 2022 - 2023 (EVEN Semester)

B.E CSE - II YEAR - IV SEMESTER

21CS210 – Database Management Systems Laboratory

Mini Project

Courier Management Systems

SUBMITTED BY,

Team member1:

Roll No: 21CSE075

Name: R.MENAGA

Team member2:

Roll No: 21CSE079

Name: C.PRADEESHA

Mark Splitup

S.No.	Criteria	Maximum Marks	Marks Obtained
1.	Application Complexity	05	
2.	Database Design	20	
3.	Frontend Design	15	
4.	Presentation	05	
5.	On time submission	05	
	Total	**50**	

Course Skilled
Dr. A.M.Rajeswari
Associate Professor, CSE

Course In-charge
Dr. S. Sasikala
Associate Professor, CSE

TABLE OF CONTENT

1. Introduction

This DBMS Mini-project focuses on various routines followed in logistics which includes placing, updating and tracking delivery packages. This interactive web application provides the facility of tracking the package for the customers, who can access the delivery status of their package using an unique Tracking ID, which displays the date-time details of stages such as Dispatched and shipped from the origin office, also received by the destination office and delivered to the receiver successfully. Also provides the facility to replace the receiver's address by their neighbour's or friend's address of the same city anytime before it is out for delivery from the destination office. For the employees, it provides the facilities of placing new order and update the delivery status of the orders. The delivery charge depends upon the weight of the parcel, source and destination state.

2. Application Users

The users of the system are the courier center staff, students. admins can update the Remote tracking and delivery address update if needed and placing new a parcel. Admin can view the details of the courier status like arrival time diapatch time of the courier when it reached its destination and the user can share their feedback over our service and their expectation on our courier management system to improve in the future.

3. Functions Overview

The following functionality will be supported by the Courier management system.

1. Remote tracking and delivery address update if needed.
2. Placing new a parcel.
3. Updating the delivery status of a parcel.
4. History of consignments delivered and yet to be delivered.
5. Feedback/query support.

4. Detailed Functional Requirements

1.Remote tracking and delivery address update if needed.

In a courier management system, remote tracking and delivery address updates are important features to ensure efficient and timely delivery of packages. Here's how they can be implemented, the courier management system should have a package tracking feature that enables the customer to track the package remotely using a unique tracking ID. The system should provide real-time updates on the package's location, expected delivery time, and any changes to the delivery schedule. The system should send notifications to the customer via email or SMS when the package is dispatched, out for delivery, and delivered. The courier management system should have a customer portal where customers can log in and update their delivery address if needed. Once the delivery address is updated, the system should send a notification to the courier driver to ensure they deliver the package to the correct address. the system should also send a confirmation notification to the customer to confirm that the delivery address has been updated successfully. by implementing these features in a courier management system, customers can easily track their packages remotely and update their delivery address if needed, ensuring a smooth and hassle-free delivery experience.

2. Placing new a parcel.

The first step is to create a new shipment in the courier management system. Once a new shipment is created, the next step is to enter the shipment details, including the sender's name and address, recipient's name and address, package dimensions and weight, and any other relevant information. After the shipment details are entered, the system should generate a label and tracking number for the parcel. The label should include the sender's and recipient's addresses, as well as the tracking number. The next step is to print the label and attach it securely to the package. Once the label is attached, the parcel is ready to be dispatched fordelivery. The courier driver assigned to the shipment will pick up the parcel and scan the tracking number toupdate the shipment's status in the system. Once the shipment is dispatched, the courier management system should provide real-time tracking information, allowing the sender and recipient to track the package's progress and receive notifications when it is delivered.By following these steps, a new parcel can be placed in a courier management system, ensuring efficient and reliable delivery.

3. Updating the delivery status of a parcel.

Updating the delivery status of a parcel in a courier management system is a crucial part of ensuring that customers receive real-time updates about their shipment's progress. When the courier driver picks up the parcel, they should scan the tracking number to indicate that the package has been picked up and is in transit. As the parcel moves through the delivery network, the courier driver should update the status of the parcel in the courier management system to reflect its current location and status. For example, the parcel status may be updated to "out for delivery," "delayed," or "delivered". Once the parcel is delivered, the courier driver should confirm the delivery in the courier management system by providing the time and date of delivery, the name of the recipient who signed for the package, and any other relevant information. The courier management system should automatically send a notification to the sender and recipient to confirm that the package has been delivered successfully.

The courier management system should provide real-time tracking updates to the sender and recipient, allowing them to track the parcel's progress and receive notifications when the status changes.By following these steps, the delivery status of a parcel can be updated in a courier management system, ensuring that customers receive real-time updates about their shipment's progress and delivery.

4. History of consignments delivered and yet to be delivered.

A history of consignments delivered and yet to be delivered is an important feature of any courier management system. This feature allows users to track the status of their shipments and provides a record of all previous shipments, making it easier to manage shipping operations. Here are the basic steps involved in accessing the history of consignments delivered and yet to be delivered in a courier management system. Users should log in to the courier management system using their credentials. Once logged in, users should navigate to the shipments page, which should display a list of all active and completed shipments. Users should then filter the shipment list by delivery status to view only consignments that are yet to be delivered or already delivered. The courier management system should provide filtering options to enable users to sort shipments by status, date range, and other parameters. Users can then click on a shipment to view its details, including the sender's and recipient's details, package dimensions, weight, and delivery status. The courier management system should also display the shipment's history, showing all events related to the shipment, including pickup, transit, and delivery. This allows users to see the shipment's entire journey and any delays or issues that may have occurred. Users should be able to export shipment data to a spreadsheet or other formats for further analysis and record-keeping. By following these steps, users can easily access the history of consignments delivered and yet to be delivered in a courier management system, allowing them to track shipment status, identify delays or issues, and manage shipping operations more efficiently.

5. Feedback/query support.

Feedback and query support are essential features of a courier management system, as they enable users to provide feedback and ask questions about the system's features and functionality. Here are the basic steps involved in accessing feedback and query support in a courier management system. The first step is to contact the courier management system's support team. This can typically be done through the system's user interface or by email or phone. When contacting support, users should provide a detailed description of their feedback or query, including any relevant information or screenshots. The courier management system's support team should respond promptly to the feedback or query, providing helpful information or guidanceto resolve the issue or address the concern. Many courier management systems also have a feedback forum, where users can provide feedback, suggest new features, and share ideas with other users and the system's developers. Some courier management systems also have a knowledge base, which contains articles and guides to help users learn more about the system's features and functionality.By following these steps, users can access feedback and query support in a courier management system, enabling them to provide feedback, ask questions, and get help when needed. This helps to ensure that the system meets users' needs and is a valuable tool for managing shipping operations.

5. Software and Hardware Requirements

Front end Tool	: visual studio
ServerSide programming	: wamp server
Backend DataBase	: mysql

6. Performance Requirements

Capacity Requirements : 2 users (admin/customer)

Sample Data Base Size : 10 to 20 tuples

Response time	: 1micro seconds
Through put	: Average transaction per minute may be 3

7. Validation Criteria

We will load the database with sample data of 10 couriers details,10 staff details, 6 destination pricing and 10 user detilsfor testing, We will test the system .

8. Entities and Relationships

1. Entities

i. STAFF

ii. CREDENTIALS

iii. BRANCHES

iv. PARCEL

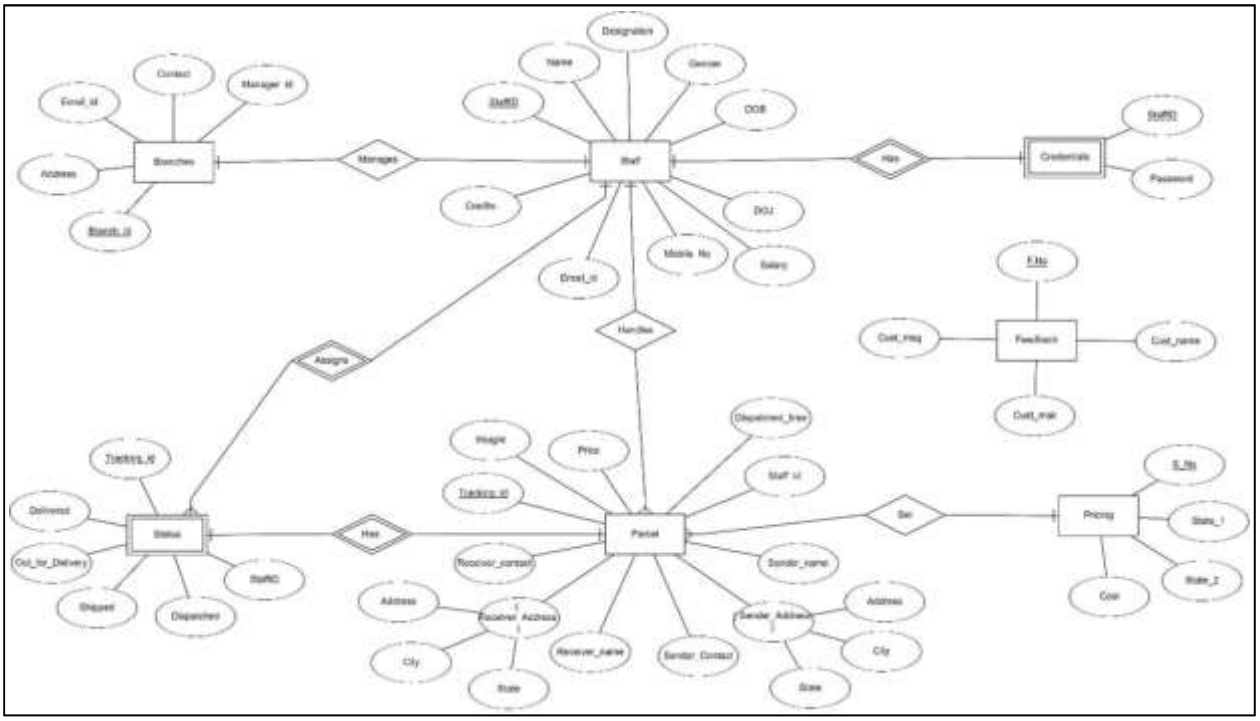
v. PRICING

vi. STATUS

2. Relationships

i. Manages
ii. Has
iii. Assigns
iv. Send
v. set

9. ER Diagram



10. Functional Dependencies and Normalization

The initial relational schema from the ER diagram.

SCHEMA

1. **STAFF** (StaffID, Name, Designation, Gender, DOB, DOJ, Salary, Mobile, Email, Credits)
2. **CREDENTIALS** (StaffID, Pwd)
3. **BRANCHES** (Branch_id, Address, Contact, Email, Manager_id)
4. **PARCEL** (TrackingID, StaffID, S_Name, S_Add, S_City, S_State, S_Contact, R_Name,R_Add, R_City, R_State, R_Contact, Weight_Kg, Price, Dispatched_Time)
5. **PRICING** (S.No, State_1, State_2, Cost)
6. **STATUS** (TrackingID, StaffID, Dispatched, Shipped, Out_for_delivery, Delivered)

FUNCTIONAL DEPENDENCIES

S.No	Relation Name	Set of FD	BCNF Checking
1.	staff	1. staffid → name 2. staffid → designation 3. staffid → DOB,DOJ 4. staffid → Salary 5. staffid → mobile 6. staffid → email 7. staffid → credits	In FDs 3 and 4 the candidate key is not superkey hence FDs 1 and 2 will be considered for the relation to be in BCNF , since dept_id is the superkey and always theyholds.
2.	Credentials	1. staffid → PWD	Among which in the FD 3 the candidate key is not superkeyhence FDs 1 and 2 will be considered for the relation to be inBCNF , since s_id is the superkey and always FDs 1and 2 holds.
3.	Branches	1. branch_id → address 2. branch_id → contact 3. branch_id → Email 4. branch_id → manager_id	In the FD 3 the candidate key sname is not superkey hence FDs 1 and 2 will be considered for the relation to be in BCNF , since staff_id is the superkey and always FDs 1and 2 holds.

4.	Parcel	1. Tracking_id→ dispatchedtime 2. staffid→ trackingid 3. Tracking_id→ price 4. Tracking_id→ destination	In FDs 3 and 4 the candidate key is not superkey hence FDs 1 and 2 will be considered for the relation to be in BCNF , since Sub_code is the superkey and always they holds.
5.	Pricing	1. State_1→ cost 2. State_1→ cost	In FDs 1,2, 3 and 4 the candidate key is not superkey hence FD 5 alone will be considered for the relation to be in BCNF , since stud_id, sub_code, date is the superkey and always they holds.
6.	status	1. trackinid, staffid →dispatched 2. trackinid, staffid →shipped 3. trackinid, staffid →delivered	In the FD 1 and 2 the candidate key sname is not superkey hence FDs 3 will be considered for the relation to be in BCNF , since stud_id, sub_code, test_no is the superkey and always FDs it holds.

11. Data Dictionary

BRANCHES relation:

Attribute	Data Type	Constraints	Remarks
Branch_id	Varchar(5)	Primary key	Branch id
Address	Varchar(50)	Not null	Name of Department
contact	Varchar(10)	Not null	Location of Department
Email	Varchar(10)	Not null	Mail address of branches
Manager_id	Varchar(5)	Foreign key	Manager of the branch

Feedback relation:

Attribute	Data Type	Constraints	Remarks
F.no	int	Primary key	Feedback number
Cust_name	Varchar(15)	Not null	Name of customer
Cust_mail	Varchar(20)	Not null	Mail id of the customer
Cust_msg	Varchar(50)	Not null	Msg from the customer

STAFF relation:

Attribute	Data Type	Constraints	Remarks
Staff_id	Varchar(5)	Primary key	Staff id
Sname	Varchar(15)	Not null	Name of staff
designation	Varchar(10)	Not null	Designation of the stall
gender	Varchar(5)	Not null	Gender of the staff
DOB	Date	Not null	Date of birth of the staff
DOJ	Date	Not null	When the staff joined s
Salary	Integer	Not null	Salary of the staff
Mobile_num	Bigint	Not null	Contact number of the staff
Email_id	Varchar(20)	Not null	Mail id of the staff
Credits	Integer	Not null	Credits of attained by the staff

PRICING relation:

Attribute	Data Type	Constraints	Remarks
S_no	Varchar(5)	Primary key	No of price list
State_1	Varchar(15)	Not null	Name of the current location
State_2	Varchar(15)	Not null	Name of the destination
Cost	Integer	Not null	Cost for the courier

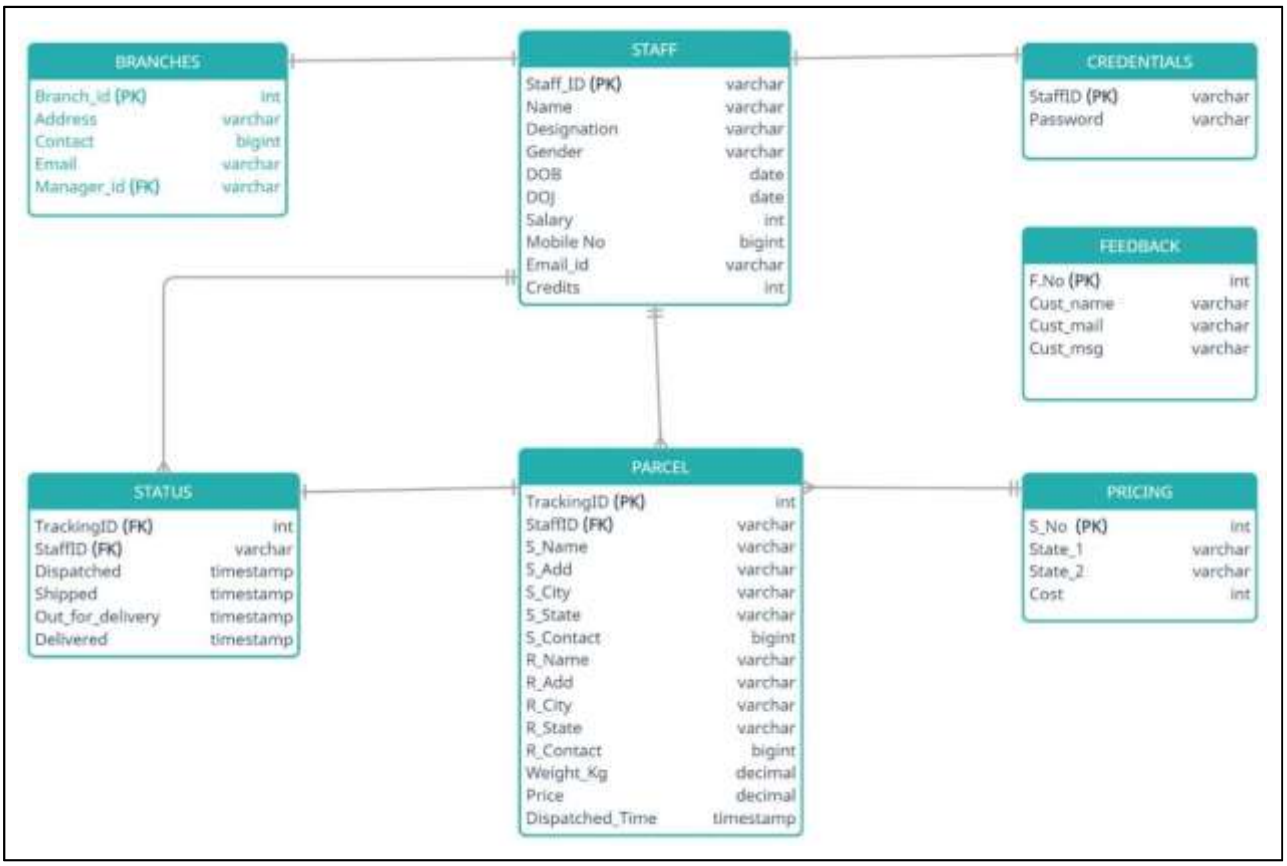
STATUS relation:

Attribute	Data Type	Constraints	Remarks
TrackingID	Varchar(5)	Foreign key	Tracking id
staffid	Varchar(5)	Primary key	Staff id
Dispatched	Timestamp	Not null	Date of the dispatched time
Shipped	Timestamp	Not null	Date of the shipped timdd
Out_of_delivery	Timestamp	Not null	Date of the shipped time
Delivered	Timestamp	Not null	Date of the shipped time

CREDENTIALS relation:

Attribute	Data Type	Constraints	Remarks
Staff_id	Varchar(5)	Primary key	References Staff
Password	Varchar(8)	Not null	Password of the staff account

SCHEMA DIAGRAM:



This schema design elucidates about the relationship each table is having with others. It also says about how the primary key and foreign key constraints are usedin order to provide relationship between the tables.

All the tables have been normalized and are present in **Boyce – Codd NormalForm** since all the transitive functional dependencies have been removed by decomposing the tables**.**

12. Database Creation

CREATING RELATIONS :

The various relations(tables) used to represent the data in the database include :

STAFF (StaffID, Name, Designation, Gender, DOB, DOJ, Salary,Mobile, Email, Credits)
CREDENTIALS (StaffID, Pwd)
BRANCHES (Branch_id, Address, Contact, Email, Manager_id) **PARCEL** (TrackingID, StaffID, S_Name, S_Add, S_City, S_State,S_Contact, R_Name, R_Add, R_City, R_State, R_Contact, Weight_Kg, Price, Dispatched_Time)
PRICING (S.No, State_1, State_2, Cost)
STATUS (TrackingID, StaffID, Dispatched, Shipped,Out_for_delivery, Delivered)
FEEDBACK (F.No, Cust_name, Cust_mail, Cust_msg)

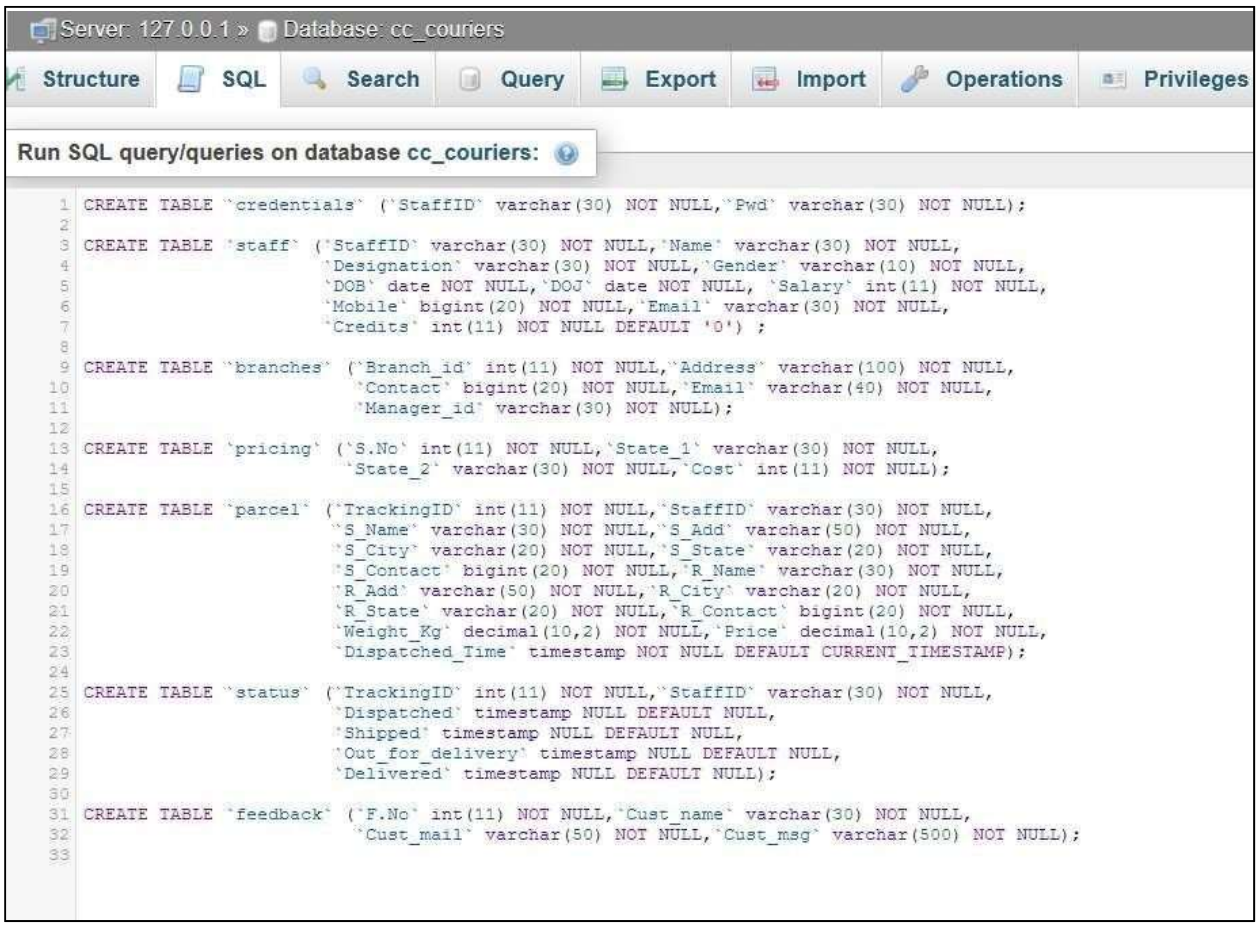


Fig 1 : SQL queries to create tables

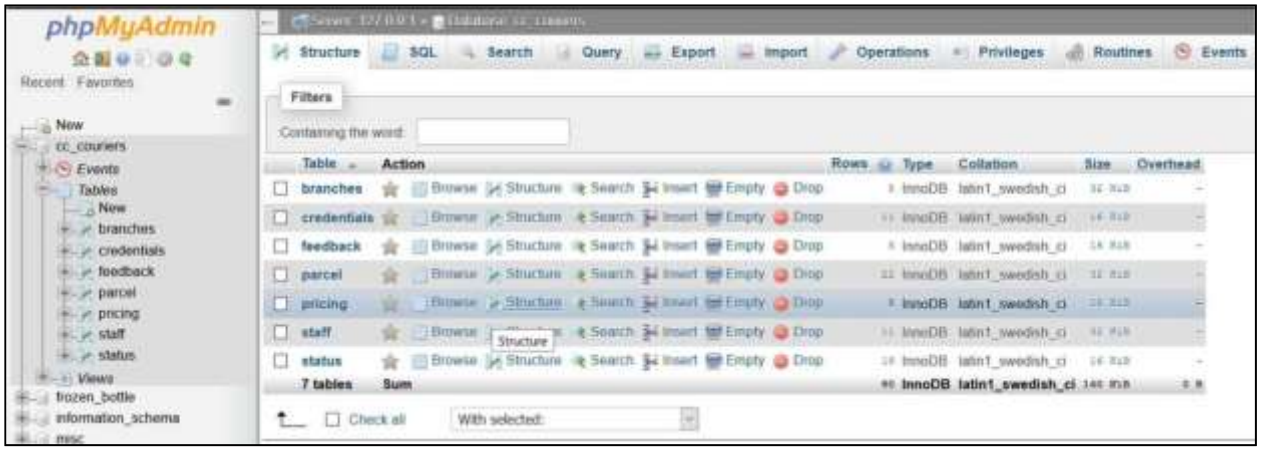


Fig 2 : Created tables in the database

CREATING VIEWS :

A view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fieldsfrom one or more real tables in the database.

Here two different views are created on the relations parcel and status.

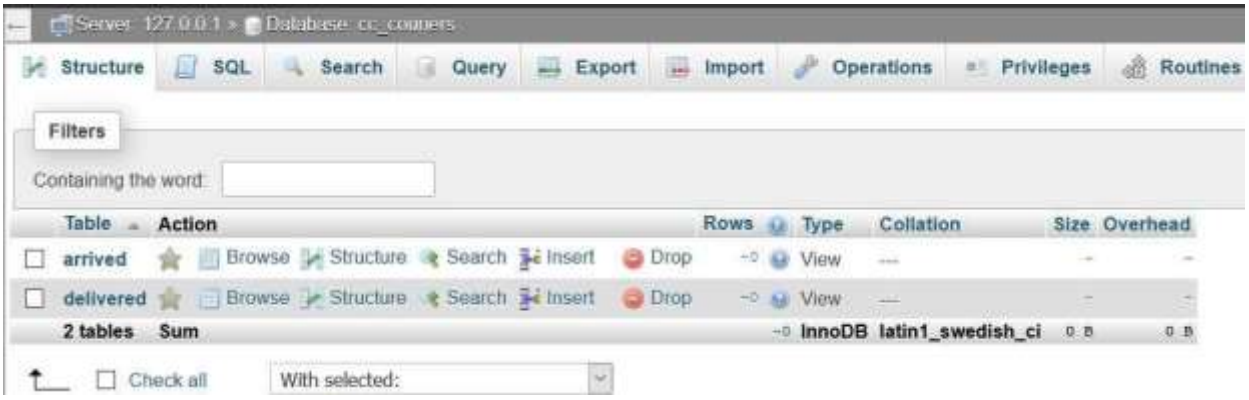


Fig 3 : List of views in the database

i. **Arrived view**: stores the details of consignments in transit.

Fig 4: SQL query to create the view – Arrived

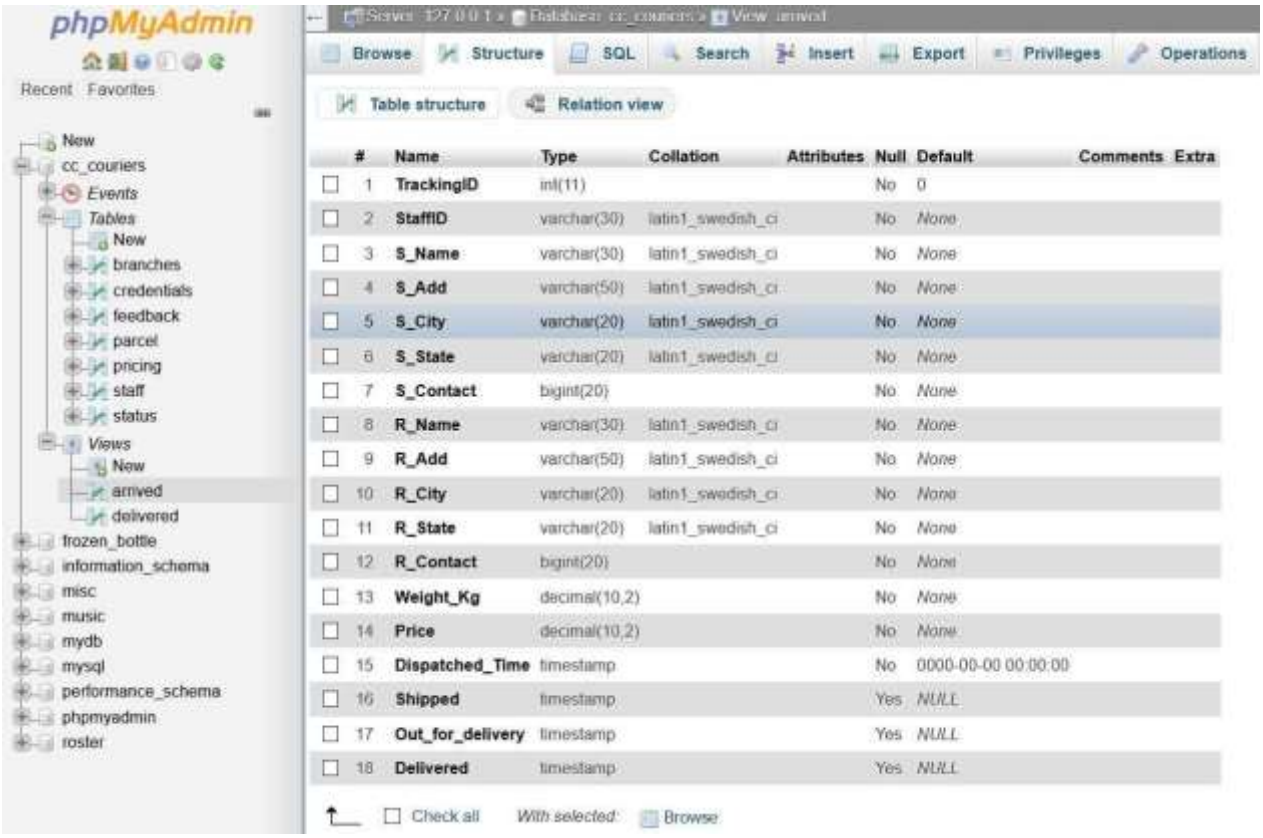


Fig 5 : Structure of the view - Arrived

i. **Delivered view** : stores the details of the consignments delivered.

Fig 6 : SQL query to create view – Delivered

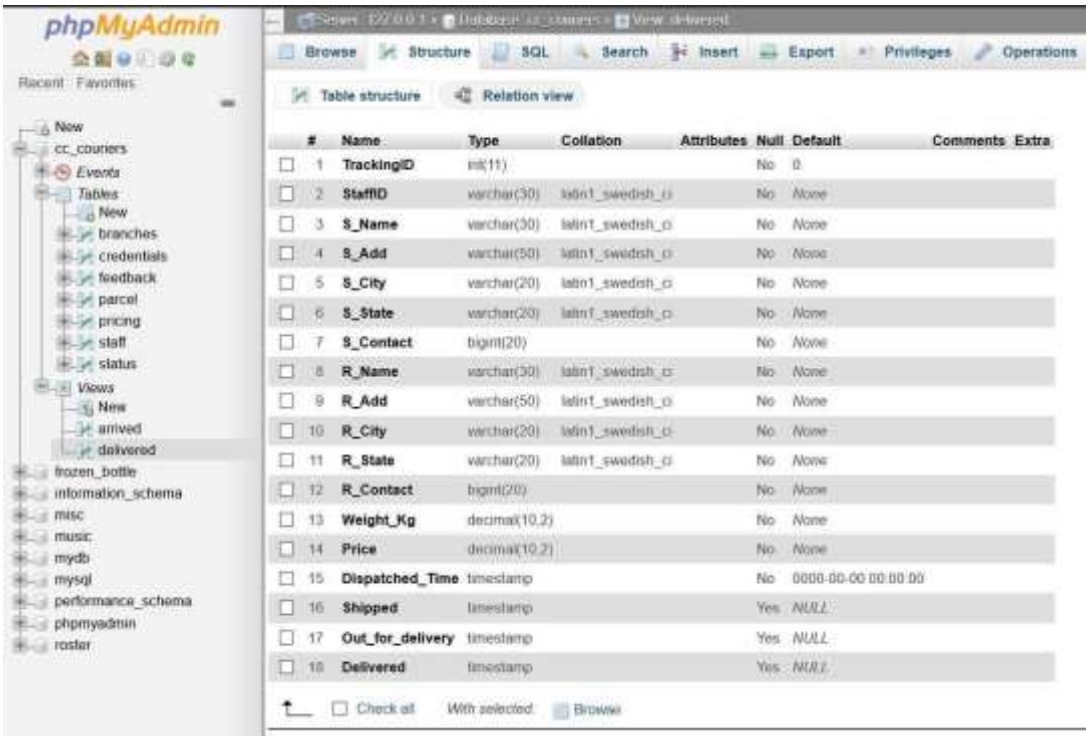


Fig 7 : Structure of the view – Delivered

CREATING TRIGGERS :

A trigger is a stored procedure in database which automatically invokes whenevera special event in the database occurs.

placeParcel : Trigger to insert the values of the attributes -TrackingId and dispatched_time - of parcel relation into the status relation after inserting a newtuple in the parcel relation.

Fig 8 : SQL query to create the trigger - placeParcel

Fig 9 : Result of successful creation of the trigger

CREATING EVENTS :

MySQL Events are named object which contains one or more SQL statement. Theyare stored in the database and executed at one or more intervals, say once every weekor month.

resetCredits: Event which resets the credits received by each staff to 0 at the beginning of every month in order to award - the Employee of the Month.

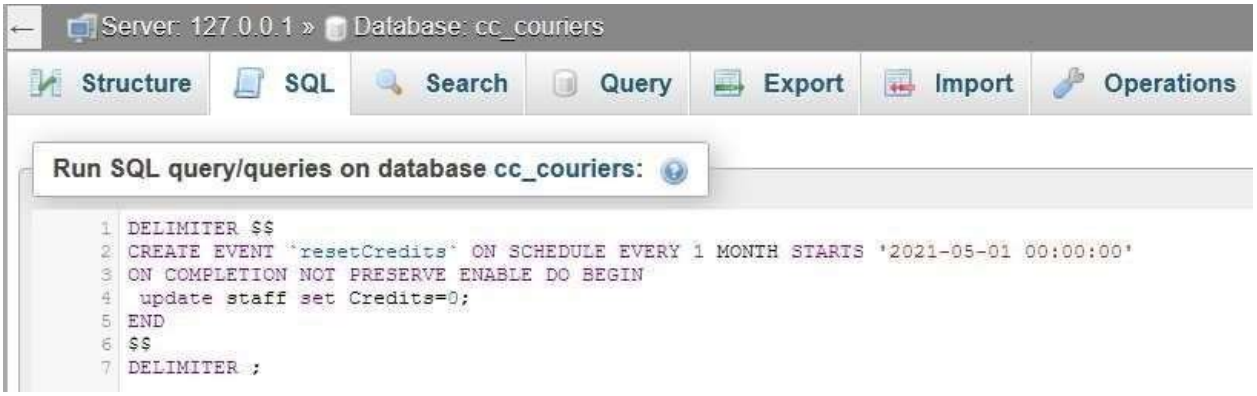


Fig 10: SQL query to create event - resetCredits

Fig 11 : Result of successful creation of the event

INSERTING VALUES INTO THE TABLES :

Data is populated into credentials, staff, branches and pricing tables by thedatabase administrator.

Data in the remaining tables – parcel, status and feedback – are populated as the staff places and updates the parcel details and the customer files a feedback/query.

1. **Credentials relation :** Stores the login credentials (Staff Id and password) ofeach of the employee.

Fig 12 : SQL query to insert data into credentials table

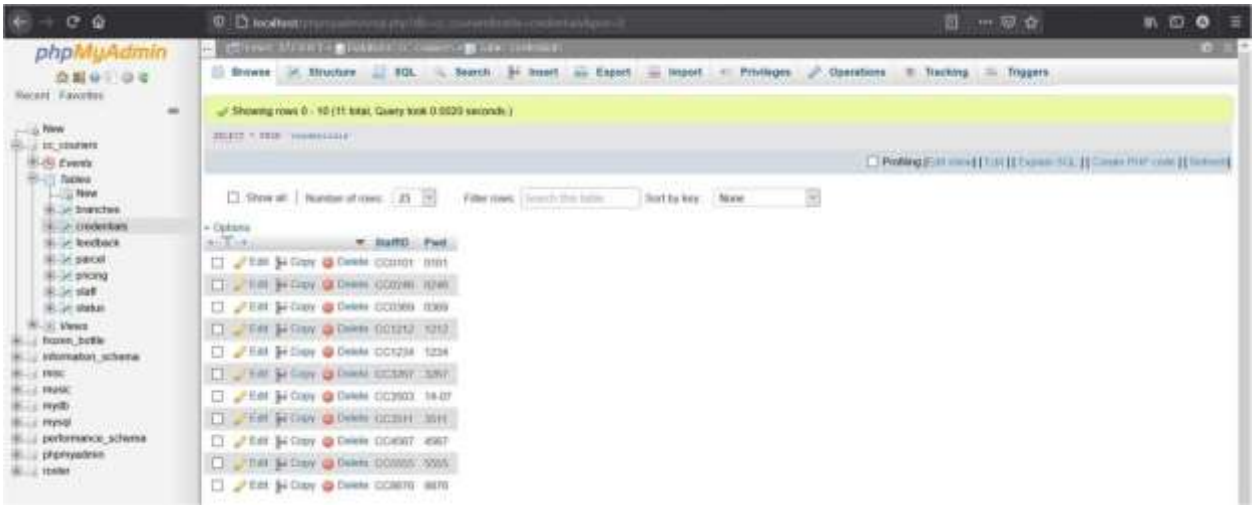


Fig 13: Data stored in credentials table

2. **Staff relation :** Stores various details regarding the staff such as their id, name,salary, contact details, etc.

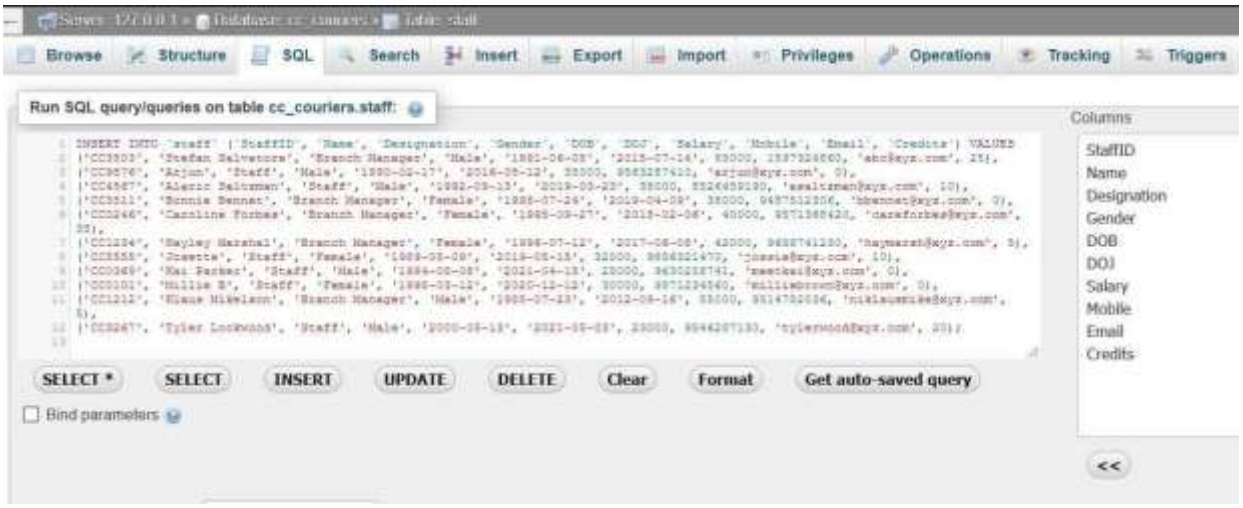


Fig 14 : SQL query to insert data into staff table

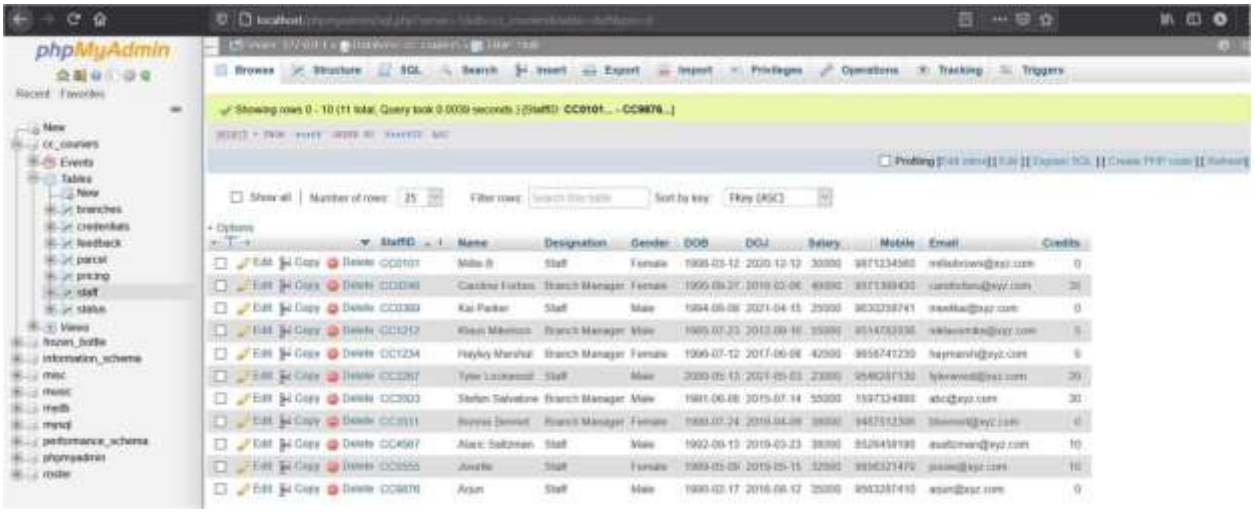


Fig 15 : Data stored in staff table

3. Branches relation :

Stores the details of the various branch offices such asbranch id, address, contact, email along with the manager id of that branch.

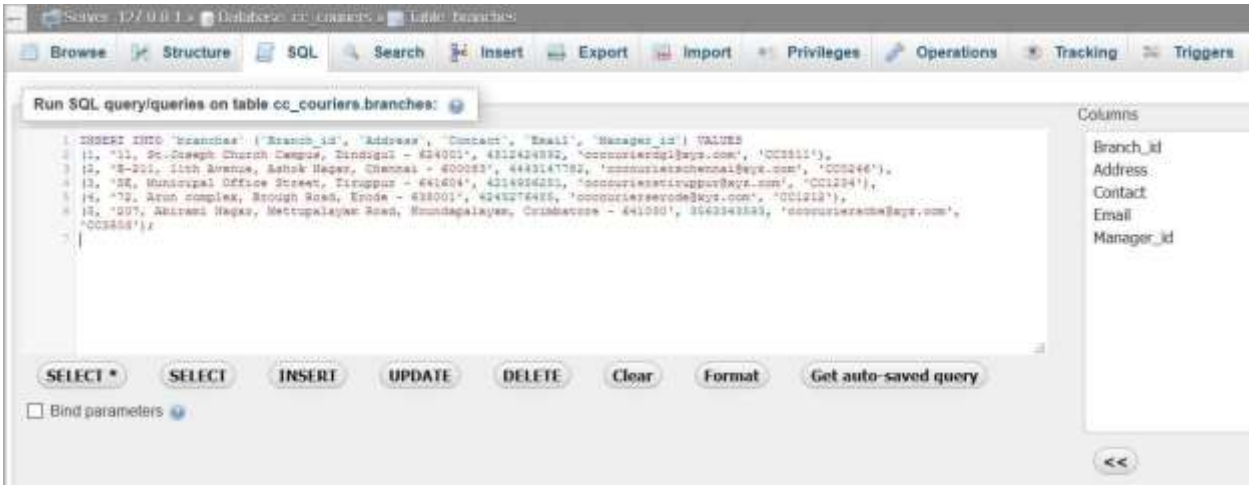


Fig 16 : SQL query to insert data into branches table

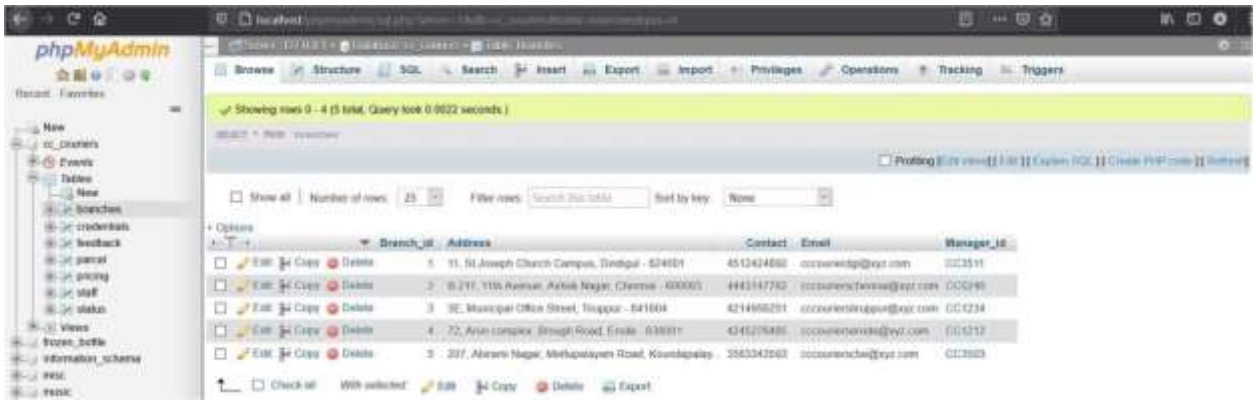


Fig 17: Data stored in branches table

4. **Pricing relation** : Stores the details of delivery charge per kg (as Cost) of thetransportation between the corresponding states. Also the states present in this relation depicts the availability of the service in various states mentioned in it.

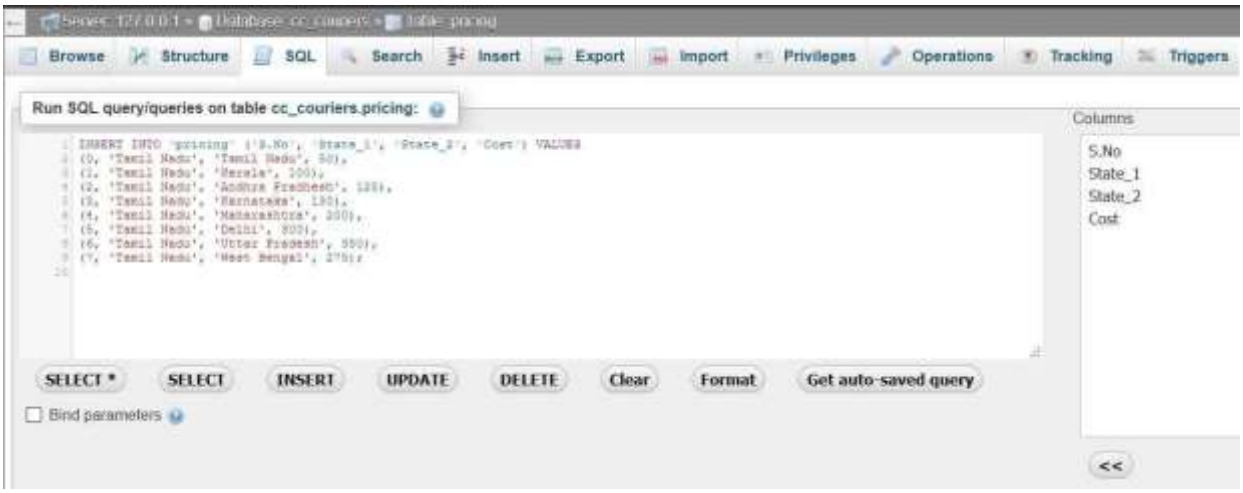


Fig 18 : SQL query to insert data into pricing table

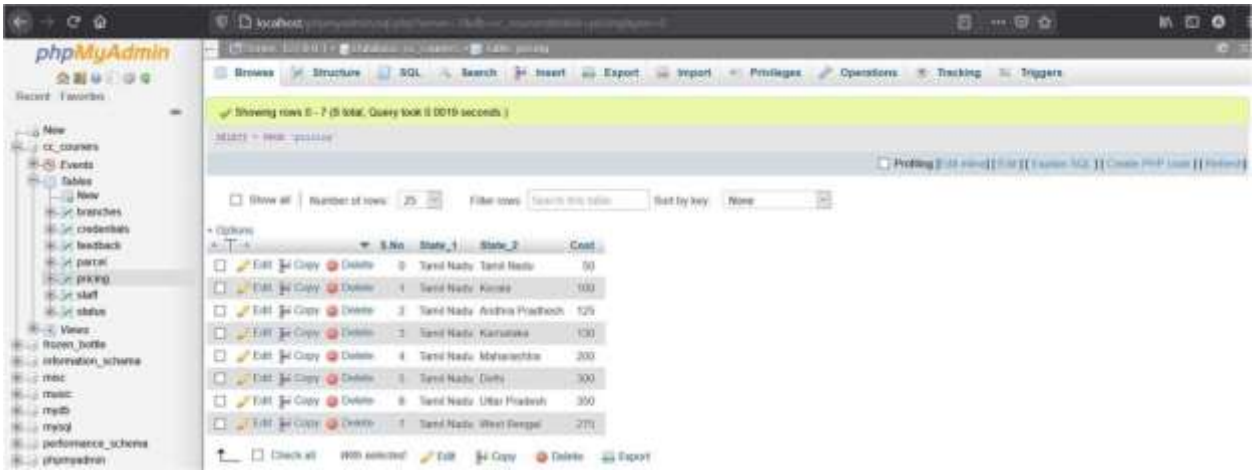


Fig 19 : Data stored in pricing table

13. DATABASE CONNECTIVITY:

```
<?php
   // connect to the database
   $servername="127.0.0.1";
   $username="root";
   $password="";
   $database='c_m_s';
   $conn=new mysqli($servername,$username,$password,$database);
   // check connection
   if(!$conn){
      echo "Connection error: ". mysqli_connect_error();
   }
?>
```

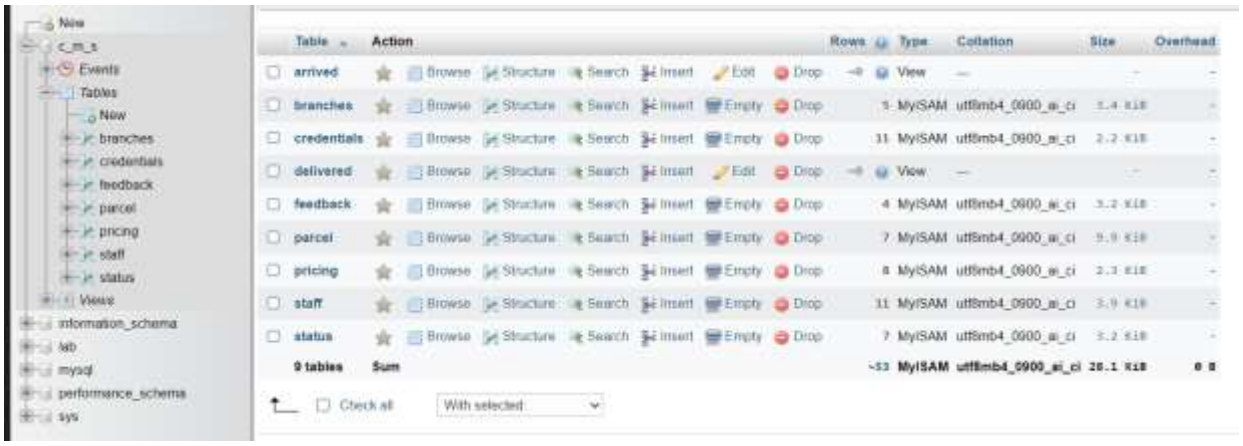


Fig 20 : MAR Couriers Database

USER INTERFACE DESIGN :

The user interface of this interactive web application is developed using HTML5, CSS3. HTML is the standard markup language for documents designed to be displayed in a web browser. CSS is the style sheet language used to style an HTML document and describes how HTML elements should be displayed.Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

HOME:

CODE:

```
<?php
   include("db_connect.php");

   $sql = "SELECT * FROM staff WHERE credits = (SELECT MAX(credits) FROM staff)";
   $result = mysqli_query($conn, $sql);
   if(mysqli_num_rows($result) > 0){
      $empmonth = mysqli_fetch_all($result, MYSQLI_ASSOC);

   }else{
      echo "Error : ". mysqli_error($conn);
   }

   $name = $email = $msg = '';
   $error = array('name' => '', 'email' => '', 'msg' => '');
   if(isset($_POST['submit'])){
      if(empty($_POST['name'])){
         $error['name'] = "*Required";
      }else{
         $name = mysqli_real_escape_string($conn, $_POST['name']);
      }
      if(empty($_POST['email'])){
         $error['email'] = "*Required";
      }else{
         if(email_validation($_POST['email'])){
            $email =  mysqli_real_escape_string($conn, $_POST['email']);
         }else{
            $error['email'] = "*Invalid email";
         }
      }
      if(empty($_POST['msg'])){
         $error['msg'] = "*Required";
      }else{
         $msg = mysqli_real_escape_string($conn, $_POST['msg']);
      }
      if(! array_filter($error)){
         $sql = "INSERT INTO feedback (Cust_name, Cust_mail, Cust_msg) VALUES ('$name', '$email', '$msg')";
         if(mysqli_query($conn, $sql)){
            echo '<script type="text/javascript">';
            echo "setTimeout(function () { swal('Thank You', 'Your response recorded successfully !!', 'success');";
            echo '}, 1000);</script>';
            $name = $email = $msg = '';
         }else{
            echo "Insert Error : " . mysqli_error($conn);
         }

      }
   }
   function email_validation($str) {
      return (!preg_match("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$^", $str)) ? FALSE
: TRUE;
   }
?>
```

```
<!DOCTYPE html>
<html>
   <head>
      <title>CC Couriers</title>
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
      <link rel="stylesheet" href="bootstrap.css">
      <link href='https://fonts.googleapis.com/css?family=Roboto' rel='stylesheet'>
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
      <script  src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
      <link rel="stylesheet" href="style/index_styles.css">
      <link rel="icon" type="image/png" sizes="32x32" href="Images/favicon-32x32.png">
      <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
      <style>
         .carousel-inner img {
          width: 100%;
          height: 100%;
         }
      </style>
   </head>
   <body style="font-family: Arial, Helvetica, sans-serif;">
      <div ><img src="logo.png" id="logo" style="height: 100px !important; margin-top: 10px !important;"
 ></div>
      <div class="background"></div>
      <nav class="navbar navbar-toggleable-md navbar-expand-lg navbar-default navbar-light mb-10"
style="background-color: rgba(255, 255, 255, 0.7); margin-bottom: 20px; margin-top:10px !important;">
         <div class="container">
            <button class="navbar-toggler text-dark" data-toggle="collapse" data-target="#mainNav">
               <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="mainNav">
               <div class="navbar-nav  " style="margin: 0 auto; font-size: large;">
                  <a class="nav-item nav-link text-dark mr-5 active" href="index.php" >Home</a>
                  <a class="nav-item nav-link text-dark mr-5" href="#about">About</a>
                  <a class="nav-item nav-link text-dark mr-5" href="tracking.php">Tracking</a>
                  <a class="nav-item nav-link text-dark mr-5" href="branches.php">Branches</a>
                  <a class="nav-item nav-link text-dark mr-5" href="#contact">Contact Us</a>
                  <a class="nav-item nav-link text-dark" href="login.php">Staff Login</a>
               </div>
            </div>
         </div>
      </nav>
      <div class = "container-fluid" style="width: 100%; padding: 0; margin: 0;">
         <div id = "carouselwithIndicators" class = "carousel slide container-fluid mt-10" data-ride = "carousel"
style="width: 85%; height: 100%; border-radius: 15px;">
            <ol class = "carousel-indicators ">
               <li data-target = "#carouselExampleIndicators" data-slide-to = "0" class = "active"></li>
               <li data-target = "#carouselExampleIndicators" data-slide-to = "1"></li>
               <li data-target = "#carouselExampleIndicators" data-slide-to = "2s"></li>
```

```
        </ol>

        <div class =" carousel-inner">
          <div class = "carousel-item active">
            <img class = "d-block "
              src = "c2.jpg"
              alt = "First slide" style="height: 80vh; width: fit-content;">
          </div>

          <div class = "carousel-item">
            <img class = "d-block "
              src = "c3.jpg"
              alt = "Second slide" style="height: 80vh; width: fit-content;">
          </div>
          <div class = "carousel-item">
            <img class = "d-block "
              src = "c4.jpg"
              alt = "Third slide" style="height: 80vh; width: fit-content;">
          </div>
        </div>

        <a class = "carousel-control-prev" href = "#carouselwithIndicators" role = "button" data-slide = "prev">
          <span class = "carousel-control-prev-icon" aria-hidden = "true" style="color: black;"></span>
          <span class = "sr-only">Previous</span>
        </a>

        <a class = "carousel-control-next " href = "#carouselwithIndicators" role = "button" data-slide = "next">
          <span class = "carousel-control-next-icon" aria-hidden = "true" style="color: black;"></span>
          <span class = "sr-only">Next</span>
        </a>
       </div>
      </div>
      <div class="container" id="about" style="margin-top: 20px; width: 85%;">
        <div class="row">
          <div class="col-md-6 p-5" style="background-color: rgba(255, 255, 255, 0.7); color: black; border-radius:
15px; ">                <h2 class="display-5 text-center mb-3 pb-2" style="border-bottom: 2px solid white;">About
Us</h2>                 <p>The launch of CC Couriers from the house of CC Cargo services is exclusively designed to
meet the commercial and personal shipment needs of our customers in both urban and rural destinations. We are
emerging as a top destination for ‘same-day’ transportation and are continuously serving our customers 24/7/365.
We constantly expand our resources to cater to our customer expectation addressing their unique market needs.</p>
            <p class="pb-3" style="border-bottom: 2px solid white;">Having created a brand in the cargo industry
we have ventured into the courier business with the same commitment. We offer flexible and faster delivery
solutions. We have spread our footprints far and wide with our bouquet of products and services. We deliver
promptly for all your time critical projects.</p>
          </div>
          <div class="col-md-6">
            <img src="abt3.jpg" style="height: 500px; width: 100%; padding-top: 5%;" >
          </div>
        </div>
      </div>        <div class="container" style="margin-top: 20px; width: 85%;">
        <div class="row">
          <div class="col-md-6 text-center p-5" style="background-color: rgba(255, 255, 255, 0.7); color: black; ">
```

```
            <img src="eom.jpg" style="width: 100%; border-top:2px solid white;" >
            <?php foreach($empmonth as $emp) : ?>
              <div style="margin:auto !important; border-bottom:2px solid white;">
                <p class="text-center pt-2" style="font-family: 'Times New Roman', Times, serif !important; font-size:x-large;"><?php echo $emp['name'] ?></p>
                <p>Staff ID : <?php echo $emp['Staffid'] ?> </p>
                <p>Credits : <?php echo $emp['credits'] ?> </p>
              </div>
            <?php endforeach; ?>
          </div>
          <div class="col-md-6 text-center p-5" style="background-color: rgba(255, 255, 255, 0.7); color: black; " id='contact'>
            <h4 style="border-bottom:2px solid white; padding-bottom:2px;">Contact Us</h4>
            <form action="index.php" class="form text-left" method = "POST">
              <div class="form-group">
                <label>Name : </label>
                <input class="form-contact"  type="text" name = "name" value=<?php echo $name; ?>>
                <span class="text-danger"><?php echo $error['name']; ?></span>
              </div>
              <div class="form-group">
                <label>Email : </label>
                <input class="form-contact"  type="text" name = "email" value=<?php echo $email; ?>>
                <span class="text-danger"><?php echo $error['email']; ?></span>
              </div>
              <div class="form-group">
                <label>Message : </label>
                <textarea class="form-contact" name = "msg" required><?php echo $msg; ?></textarea>
                <span class="text-danger"><?php echo $error['msg']; ?></span>
              </div>
              <input type="submit" name="submit" value="Submit"class="btn btn-info">
            </form>
          </div>
        </div>
            </div>
      <div class="container-fluid text-center mt-5" style="background-color: rgba(255, 255, 255, 0.7); padding: 20px; position: relative; ">
        <div class="i-bar" style="display: flex; flex-direction: row; flex-wrap: wrap; justify-content:center; margin-bottom: 2em;">
          <a class="fa fa-facebook " href="#" style="border: none; text-decoration: none; margin: 0em 1em; color:black; font-size: xx-large;"></a>
          <a class="fa fa-instagram" href="#" style="border: none; text-decoration: none; margin: 0em 1em; color:black; font-size: xx-large;"></a>
          <a class="fa fa-envelope " href="#" style="border: none; text-decoration: none; margin: 0em 1em; color:black; font-size: xx-large;"></a>
        </div>
        <p class="credit" style="font-size: 20px; font-stretch: 3px; text-align: center; color: black;">© MAR COURIERS</p>
      </div>
    </body>
</html>
```

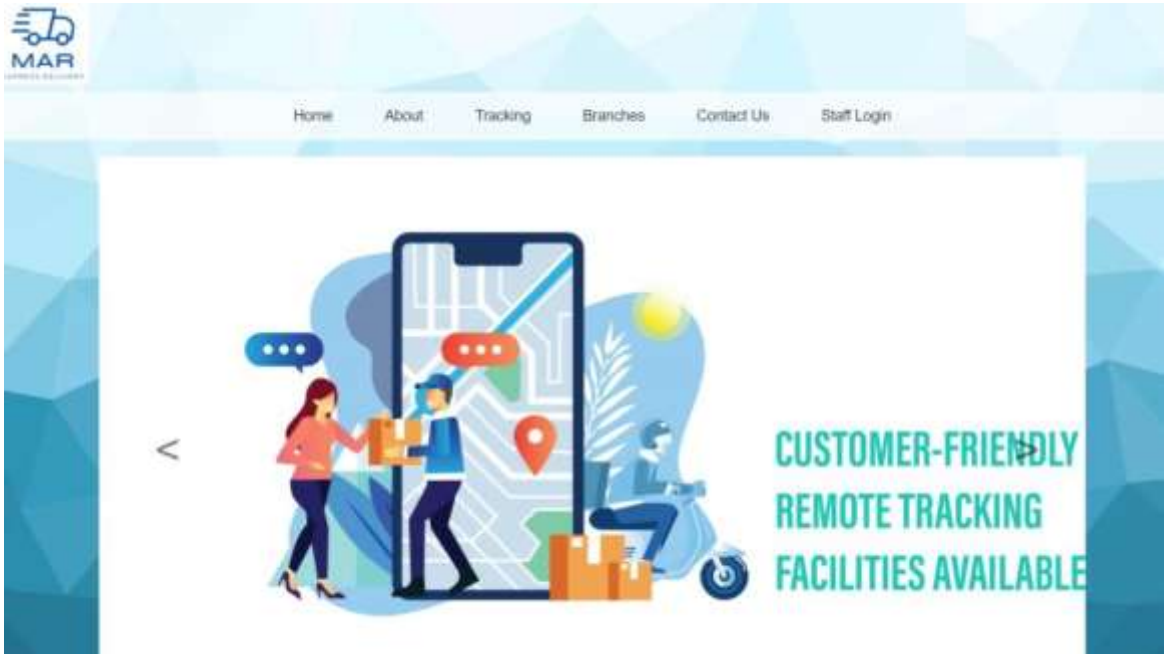


Fig 21 : Home page of the application with a carousal

Fig 22 : The page featuring the legacy of CC couriers

EMPLOYEE OF THE MONTH :

```php
$sql = "SELECT * FROM staff WHERE credits = (SELECT MAX(credits) FROM staff)";
$result = mysqli_query($conn, $sql);
if(mysqli_num_rows($result) > 0){
    $empmonth = mysqli_fetch_all($result, MYSQLI_ASSOC);

}else{
    echo "Error : ". mysqli_error($conn);
}
```

PHP code snippet with the corresponding SQL query to fetch the record of the staff with maximum credits and displayed as the Employee of the month. Note that each staff is awarded 5 credits for each parcel they place (which is added to the credits in staff table by the placeParcel trigger) and the event – resetCredits – resets the credits of all employees to 0 on every month beginning.

Fig 23 : Segment of the home page featuring the Employee of the Month and the Contact form

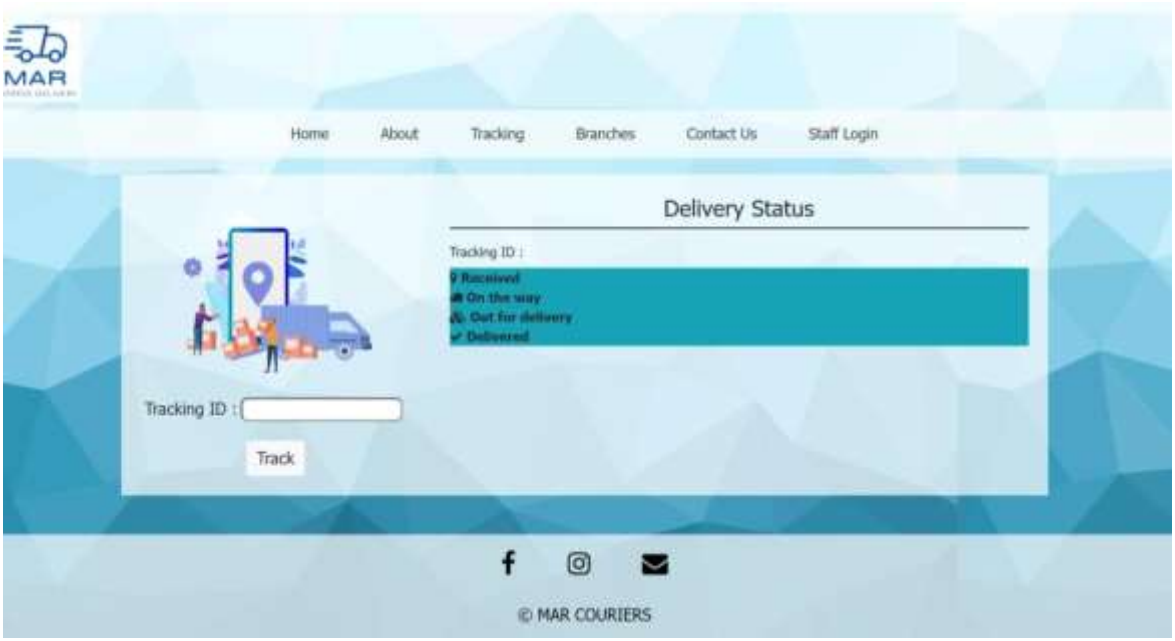


Fig 24 : Tracking page of the application

DISPLAYING BRANCH DETAILS :

branches.php

```php
<?php
    include("db_connect.php");
    $sql = "SELECT * FROM branches";
    $result = mysqli_query($conn, $sql);
    $branches = mysqli_fetch_all($result, MYSQLI_ASSOC);
?>
```

Fig PHP code snippet with corresponding SQL query to fetch the records of the branches

Fig 25 : Branches page displaying the details of the

branch offices stored in branches relation

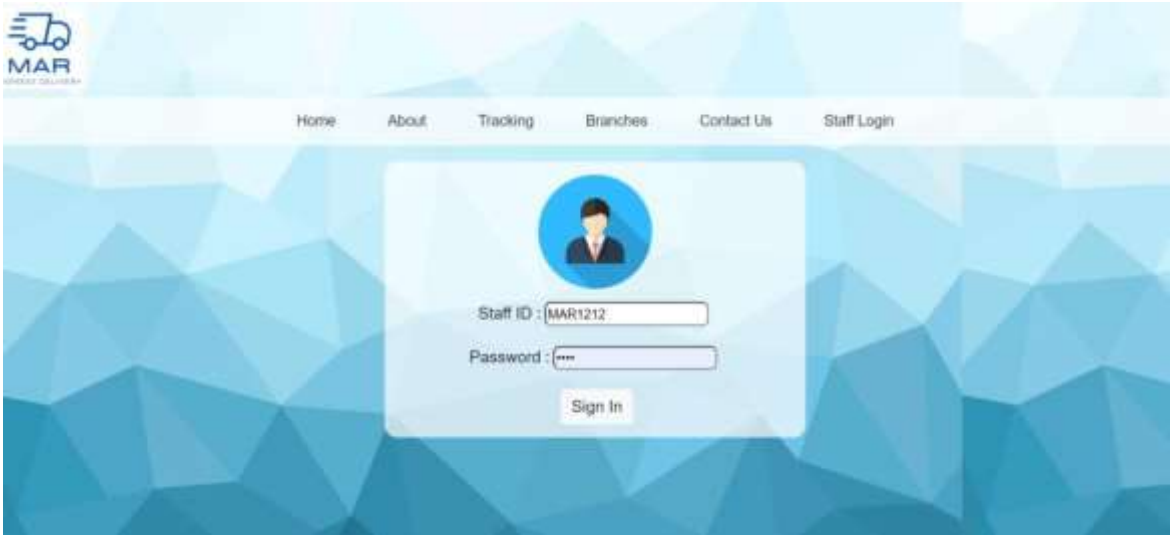


Fig 26: Staff Login page

Fig 27: Placing a new order tab of the staff page

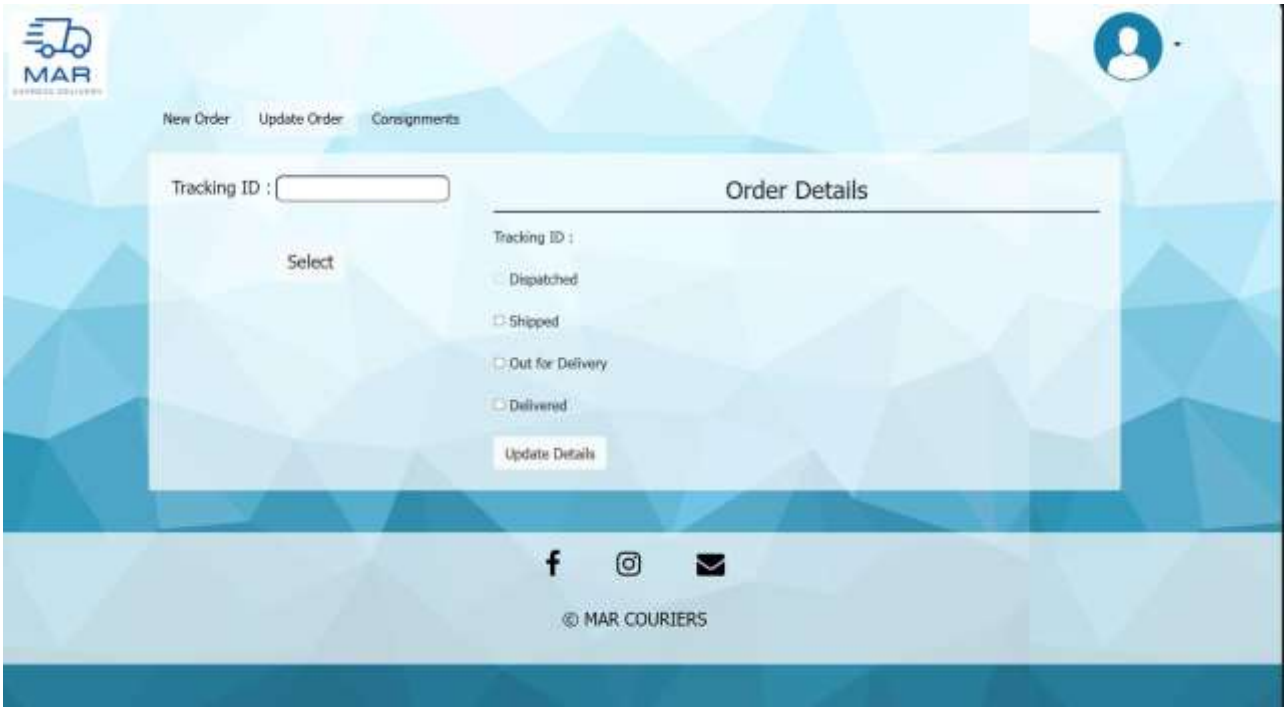
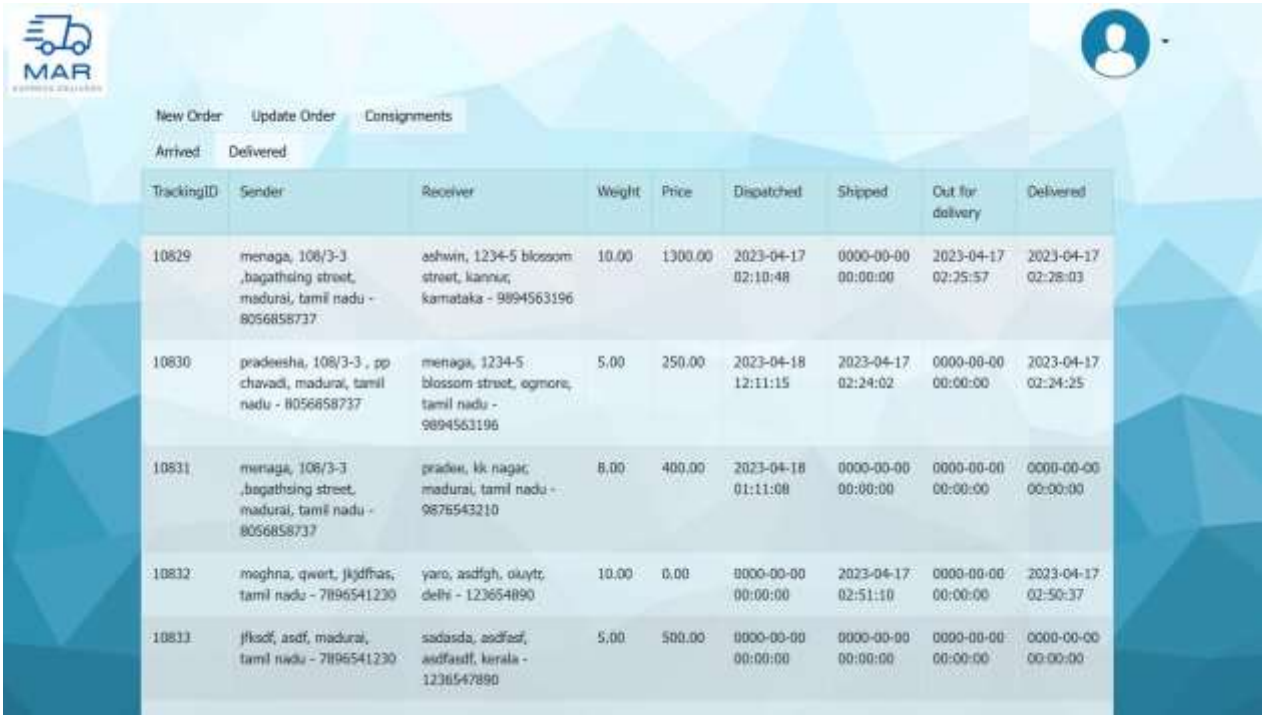


Fig 28: Updating the existing parcel details tab of staff page



TrackingID	Sender	Receiver	Weight	Price	Dispatched	Shipped	Out for delivery	Delivered
10829	menaga, 108/3-3 ,bagathsing street, madurai, tamil nadu - 8056858737	ashwin, 1234-5 blossom street, kannur, karnataka - 9894563196	10.00	1300.00	2023-04-17 02:10:48	0000-00-00 00:00:00	2023-04-17 02:25:57	2023-04-17 02:28:03
10830	pradeesha, 108/3-3 , pp chavadi, madurai, tamil nadu - 8056858737	menaga, 1234-5 blossom street, egmore, tamil nadu - 9894563196	5.00	250.00	2023-04-18 12:11:15	2023-04-17 02:24:02	0000-00-00 00:00:00	2023-04-17 02:24:25
10831	menaga, 108/3-3 ,bagathsing street, madurai, tamil nadu - 8056858737	pradee, kk nagar, madurai, tamil nadu - 9876543210	8.00	400.00	2023-04-18 01:11:08	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00
10832	meghna, qwert, jkjdfhas, tamil nadu - 7896541230	yaro, asdfgh, oiuytr, delhi - 123654890	10.00	0.00	0000-00-00 00:00:00	2023-04-17 02:51:10	0000-00-00 00:00:00	2023-04-17 02:50:37
10833	jfksdf, asdf, madurai, tamil nadu - 7896541230	sadasda, asdfasf, asdfasdf, kerala - 1236547890	5.00	500.00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00

Fig 29 : Page displaying the consignments in transit which are the contents of the arrived and derived view

FEEDBACK/QUERY FILING :

```php
$name = $email = $msg = '';
$error = array('name' => '', 'email' => '', 'msg' => '');
if(isset($_POST['submit'])){
    if(empty($_POST['name'])){
        $error['name'] = "*Required";
    }else{
        $name = $_POST['name'];
    }
    if(empty($_POST['email'])){
        $error['email'] = "*Required";
    }else{
        if(email_validation($_POST['email'])){
            $email = $_POST['email'];
        }else{
            $error['email'] = "*Invalid email";
        }
    }
    if(empty($_POST['msg'])){
        $error['msg'] = "*Required";
    }else{
        $msg = $_POST['msg'];
    }
    if(! array_filter($error)){
        $sql = "INSERT INTO feedback (Cust_name, Cust_mail, Cust_msg) VALUES ('$name', '$email', '$msg')";
        if(mysqli_query($conn, $sql)){
            echo '<script type="text/javascript">';
            echo "setTimeout(function () { swal('Thank You', 'Your response recorded successfully !!', 'success');";
            echo '}, 1000);</script>';
        }else{
            echo "Insert Error : " . mysqli_error($conn);
        }
```

PHP code snippet to store the contents from the contact form provide for feedback and queries into the feedback table.

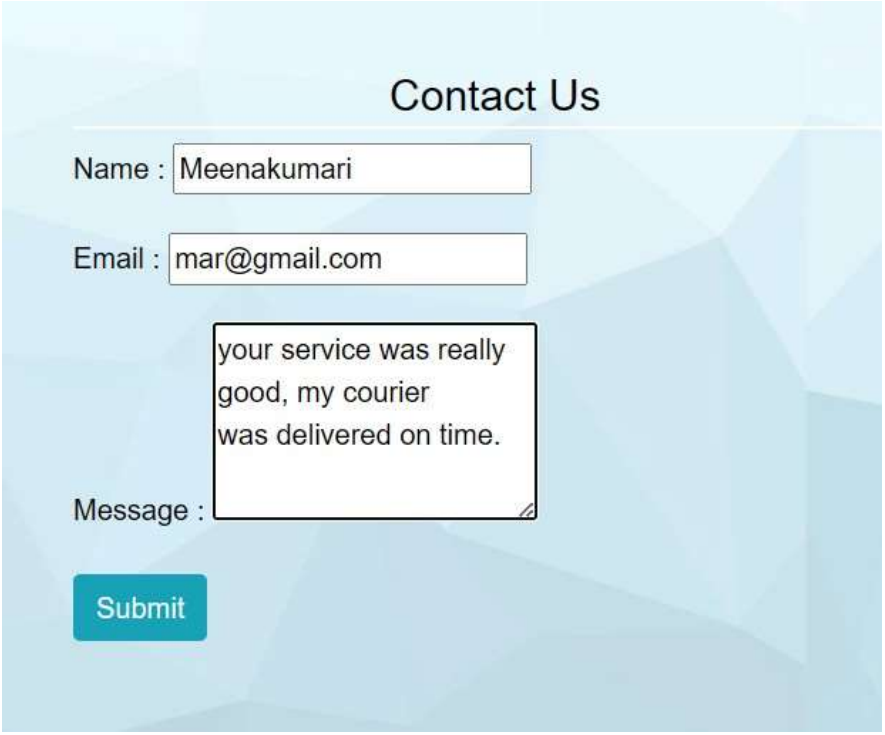


Fig 30 : The Contact form with details entered.

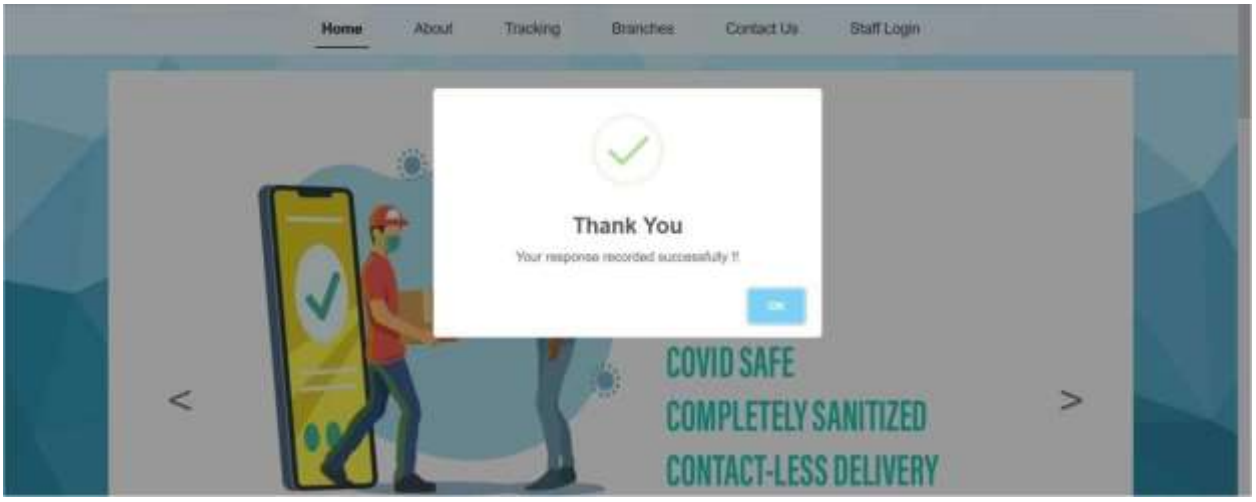


Fig 31 : On successful submission of the feedback/query success alert is prompted.

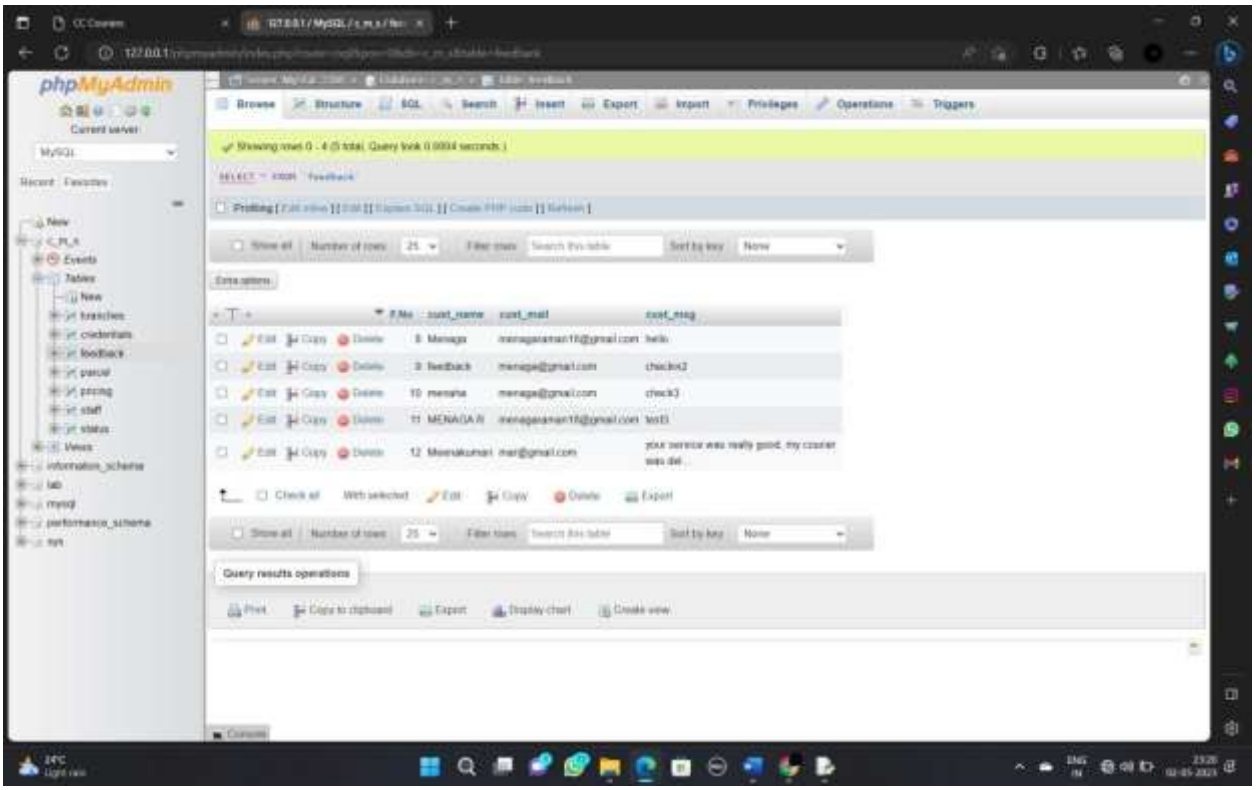


Fig 32 : The result of successful submission is reflected in the feedback table at entry f.no 12.

PLACING A NEW PARCEL :

```
72          if(array_filter($errors)){
73              //echo errors
74          }else{
75              $price = 0;
76              $sql  = "SELECT * FROM pricing WHERE State_1 = '$sstate' AND State_2 = '$rstate'";
77              $result = mysqli_query($conn, $sql);
78              if(mysqli_num_rows($result) > 0){
79                  $pricing = mysqli_fetch_assoc($result);
80                  $price = $pricing['Cost'] * $wgt;
81
82                  $sql = "INSERT INTO parcel (StaffID, S_Name, S_Add, S_City, S_State, S_Contact, R_Name, R_Add, R_City, R_State, R_Contact,
83                  if(mysqli_query($conn, $sql)){
84                      $tid = mysqli_insert_id($conn);
85                      $_SESSION['tid'] = $tid;
86                      header("Location: receipt.php");
87                  }else{
88                      echo "Error : " . mysqli_error($conn);
89                  }
90              }else{
91                  echo '<script type="text/javascript">';
92                  echo "setTimeout(function () { swal('Service Not Available', 'CC Couriers will reach your place soon !!', 'info');";
93                  echo '}, 1000);</script>';
94              }
95          }
```

PHP code snippet which stores the data collected from the new order form into the parcel table (and to the status table by the trigger placeParcel) only if the delivery details provided can be serviced. Here the delivery charge is calculated based on the values stored in thepricing table. It redirects the page to receipt page along with its unique tracking id. It also alerts the staff when the service is not available for the address details provided.

Fig 33 : Page displaying the form with details which is stored into the parcel and status table when the place order button is hit.

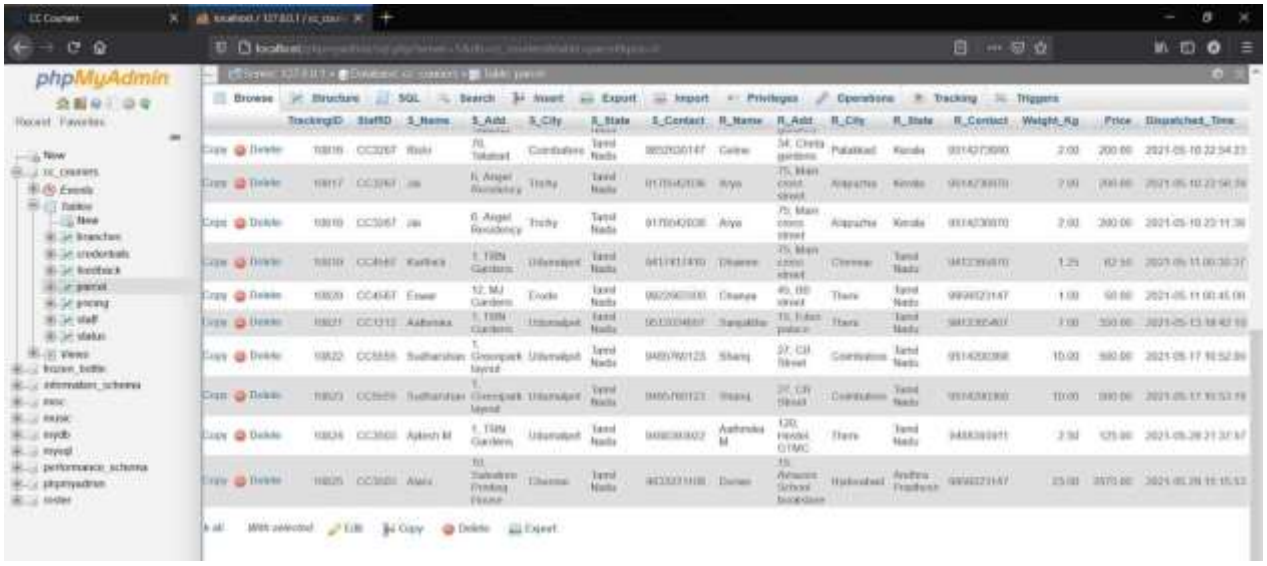
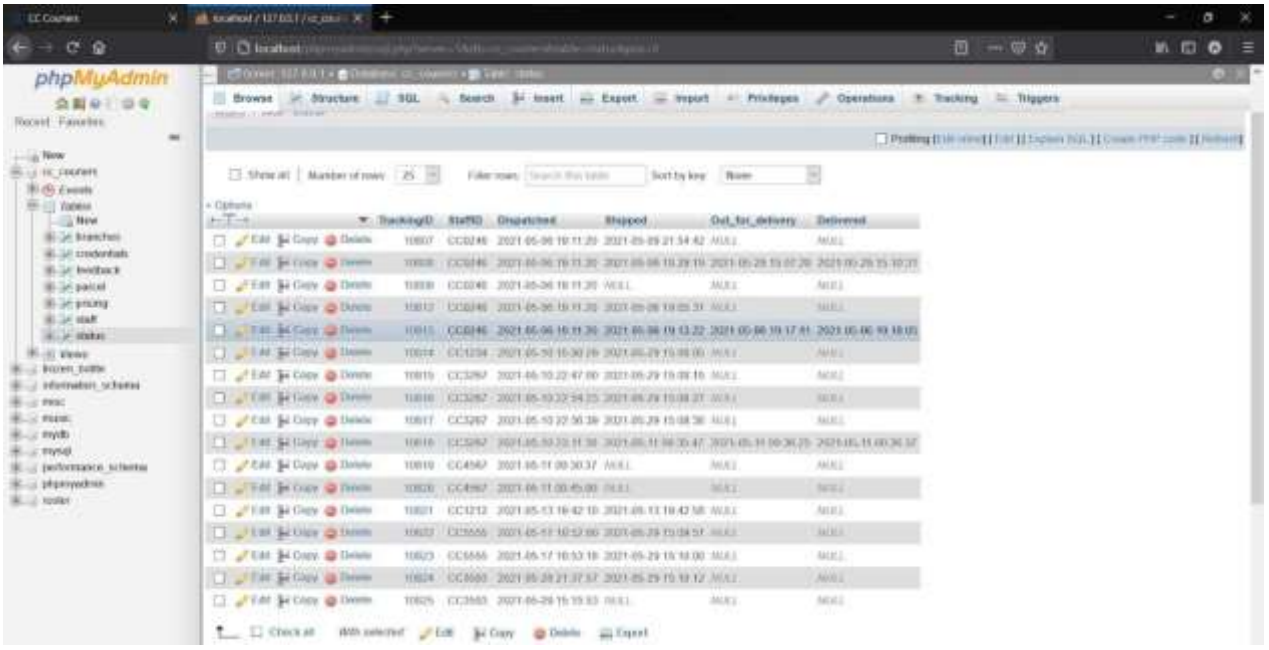


Fig 34: The details of the placed parcel along with its delivery charge (price), staff id and the dispatched time (which is the time when the parcel received by the source office = CURRENT_TIMESTAMP) in the parcel table.



FResult of the trigger – placeParcel – which stored the tracking id, staff id and dispatchedtime of the new parcel placed (in parcel table) into the status table. Also the trigger adds 5 creditsto the staff for each parcel placed by them which is used to announce the Employee of the Month.

Fig 35 : Receipt page on successful placement of parcel display its details along with its uniquetracking id and delivery charges calculated based on the values provided by the pricing table on. The back button takes us back to staff page and the print button prints the receipt out.

Fig 36 : The generated receipt is printed when the print button is hit.

Fig 37 : Page displaying the form with details where the service is not available. These servicedetails are extracted from the pricing table which depicts the states between which the service is available along with delivery charge/kg.

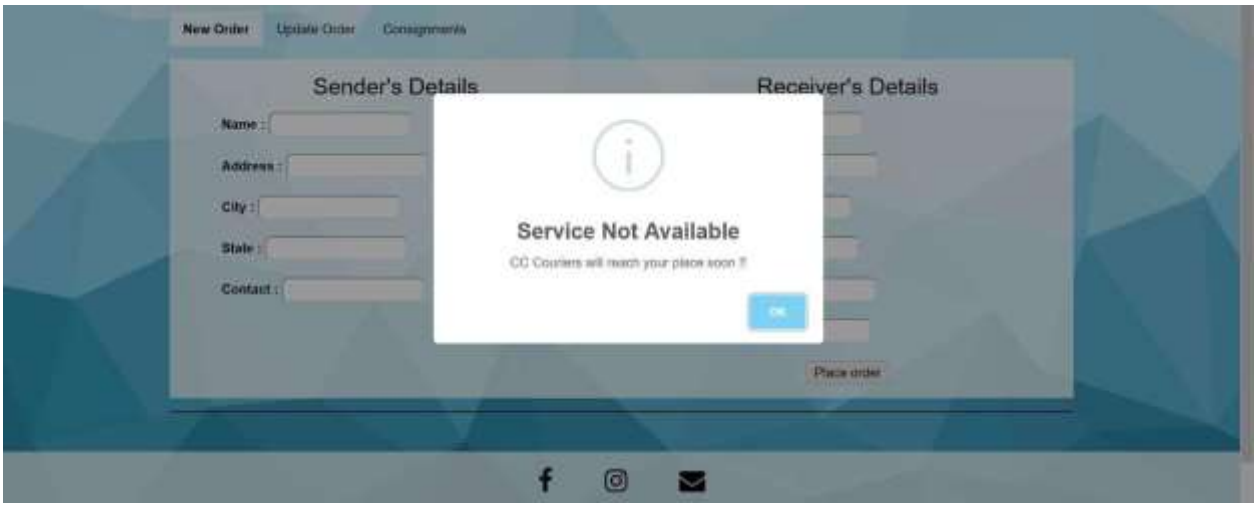


Fig 38 : Alert prompting that the delivery service is not available for the address details provided.

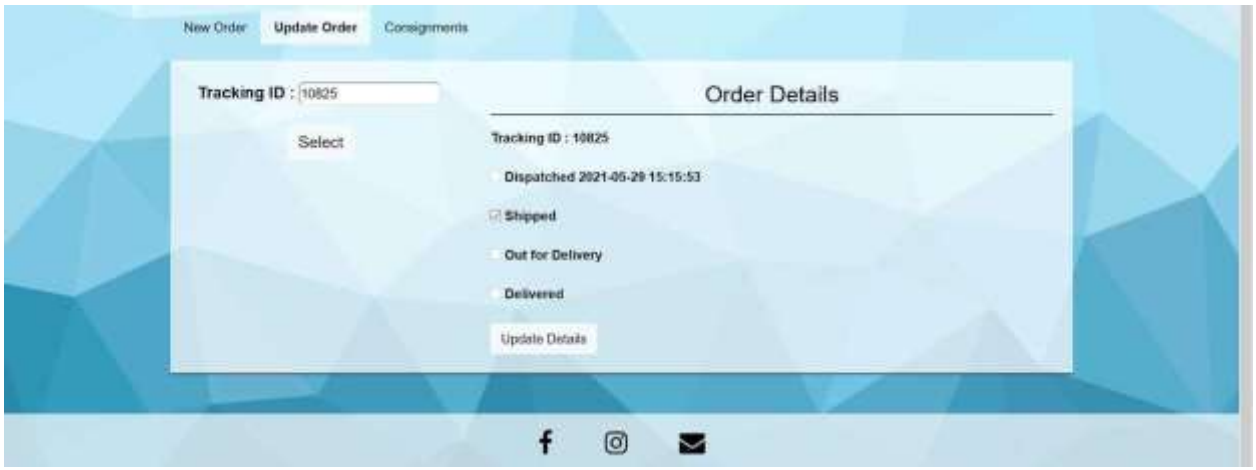


Fig 39: The record of the entered tracking id is fetched from the status table is displayed and the subsequent status is checked for update. Also note that only one status track is enabled, thelower and higher levels are disabled to ensure the hierarchy of the delivery statuses.

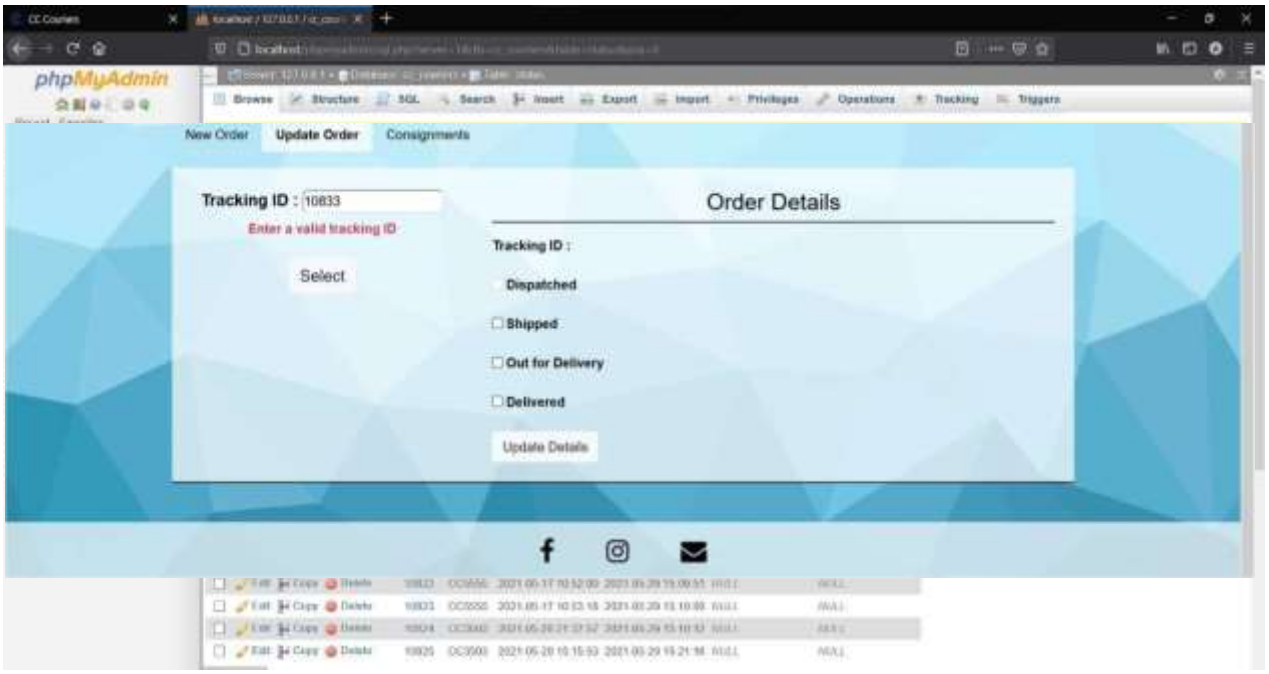


Fig 40 : Note that the record of the tracking id 108025 is modified with Shipped = current time.

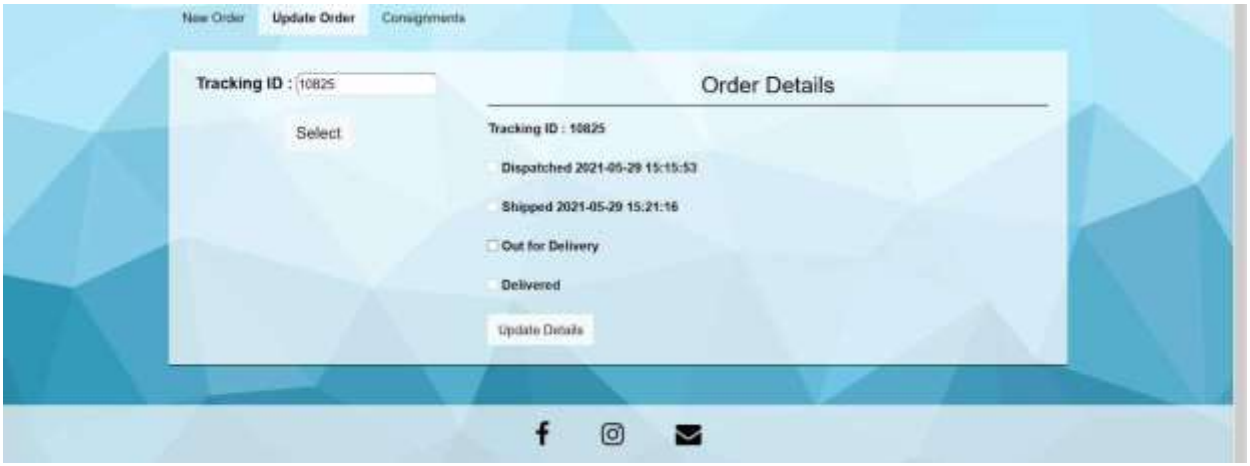


Fig 41 : The successful status update is reflected when the record of the same tracking id is fetched.

STAFF LOGIN :

login.php

```php
<?php
    include("db_connect.php");

    $id = $pwd = '';
    $errors = array('id' => '', 'pwd' => '', 'login' => '');

    if(isset($_POST['submit'])){
        if(empty($_POST['id'])){
            $errors['id'] = "*Required";
        }else{
            $id = $_POST['id'];
        }
        if(empty($_POST["pwd"])){
            $errors['pwd'] = "*Required";
        }else{
            $pwd = $_POST['pwd'];
        }
        if(array_filter($errors)){
            //echo errors
        }else{
            $id = mysqli_real_escape_string($conn, $id);
            $pwd = mysqli_real_escape_string($conn, $pwd);

            $sql = "SELECT * FROM credentials WHERE StaffID='$id' AND Pwd='$pwd'";
            $result = mysqli_query($conn, $sql);
            if(mysqli_num_rows($result) > 0){
                $user = mysqli_fetch_assoc($result);
                session_start();
                $_SESSION['id'] = $user['StaffID'];
                header("Location: staff.php");
            }else{
                $sql = "SELECT * FROM credentials WHERE StaffID='$id'";
                $result = mysqli_query($conn, $sql);
                if(mysqli_num_rows($result) == 0){
                    $errors['login'] = 'Enter valid Staff ID';
                }else{
                    $user = mysqli_fetch_assoc($result);
                    if($pwd != $user['Pwd']){
                        $errors['login'] = 'Incorrect Password';
                    }
                }
            }
        }
    }

```

PHP code snippet for validating the login credentials and redirecting to the staff pageon successful login. It stores the credentials in the session on successful login.

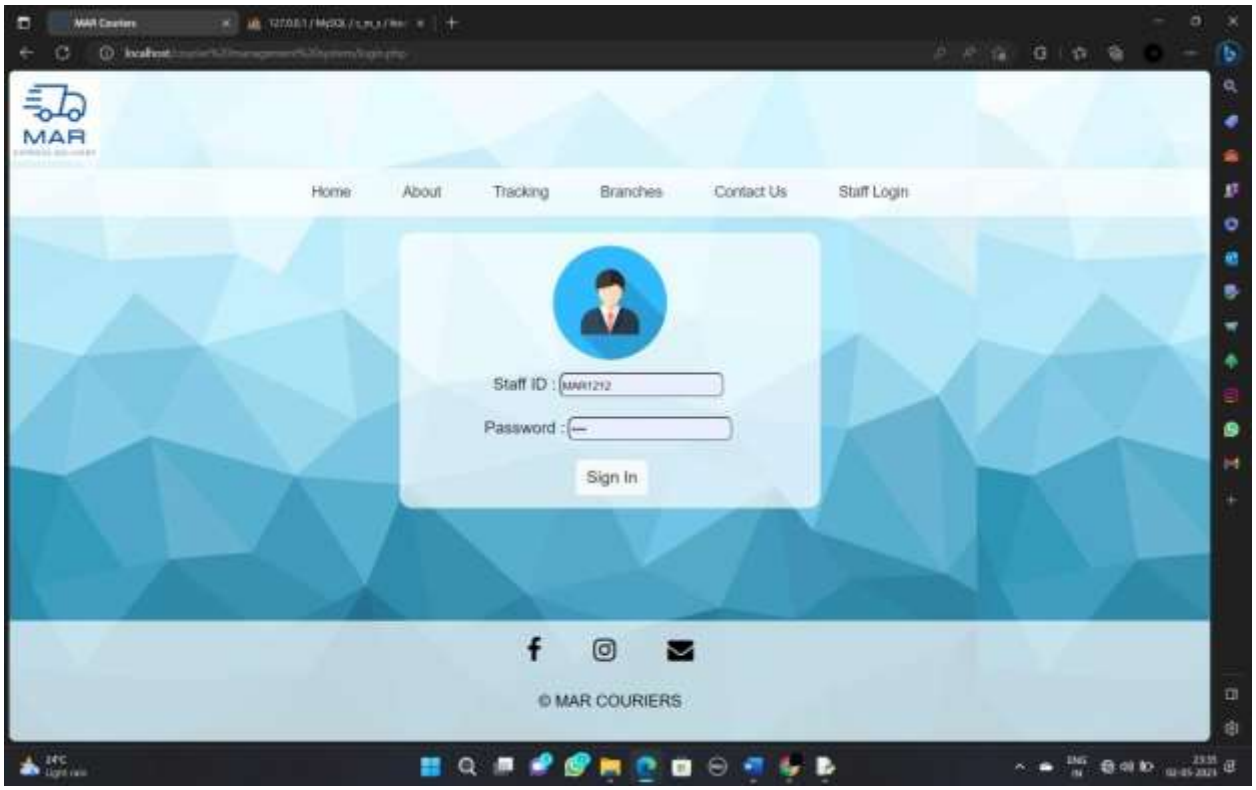


Fig 42 : Staff login page with login credentials as per the credentials table

Fig 43 : Staff login page displaying the prompt – Incorrect Password

STAFF DETAILS :

```php
<?php
    include("db_connect.php");
    session_start();
    $id = $_SESSION['id'];
    $sql = "SELECT * FROM staff WHERE StaffID='$id'";
    $result = mysqli_query($conn, $sql);
    $staff = mysqli_fetch_assoc($result);
?>
```

PHP code snippet that executes the corresponding SQL query to fetch the details of thelogged in staff

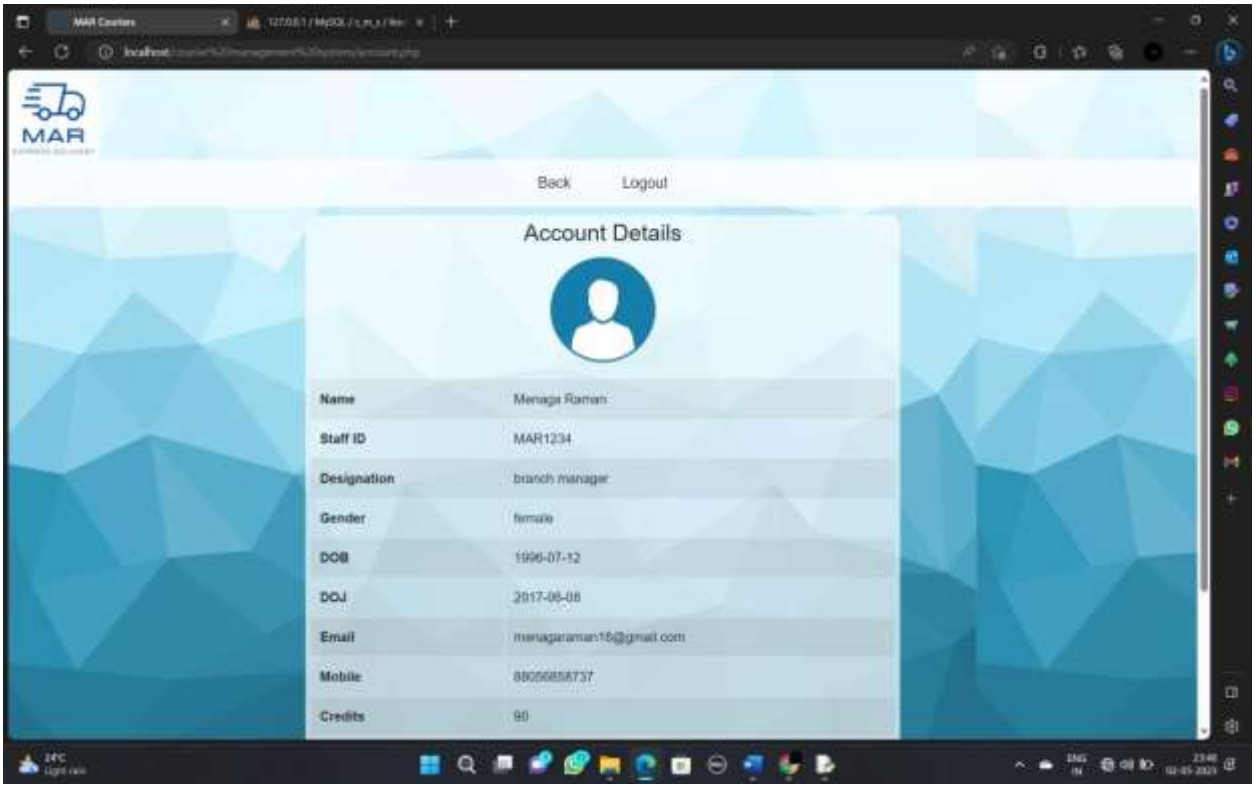


Fig 44 : Page displaying the fetched staff details from the staff table

LOGOUT :

logout.php

```php
<?php
    session_start();
    session_destroy();
    header("Location: login.php");
?>
```

PHP code snippet to logout the current staff by destroying the session storing thecredentials of the logged in staff id. On successful logout, it redirects to the login page.

Fig 45 : The logout option is available in the dropdown near the logged in staff which redirectsto the login page.

Fig 45 : in the footer tag have the some other social media platform of our MAR courier

14. CONCLUSION:

From understanding the user and admin requirements to system design and finally consolidation of everything, each step requires in-depth understanding and commitment toward achieving the objectives of this project. CC Couriers is a customer-friendly delivery service with flexible policies and proper management system.

Although the Courier Management System developed in this project is not fully integrated to the real world needs of a system, the prototype and implementation demonstrates easy navigation in the system and how data are stored in a systematic view. The study of how a relational database will be developed and maintained for industry purposes was done and been implemented. Overall, the main motive of this project is to gain more knowledge about the usage of database systems in the industrial view which was done efficiently. Also it is indeed a great learningexperience.

15. REFERENCES

- ✓ https://www.w3schools.com/
- ✓ https://www.javatpoint.com/dbms-tutorial
- ✓ https://www.guru99.com
- ✓ www.stackoverflow.com

****