COURIER MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

TIYASA KUNDU [RA2011003010648]
PALAK RANI [RA2011003010661]
PALAK PATEL[RA2011003010666]

Under the guidance of

Mrs. D. Viji

(Assistant Professor, Computing Technologies)

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING
With specialization in Computing Technologies

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

APRIL 2023

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this project report **"Courier Management System"** is the bonafide work of **"Tiyasa Kundu (RA2011003010648), Palak Rani (RA2011003010661) and Palak Patel (RA2011003010666)"** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC303J- Database Management systems in SRM Institute of Science and Technology during the academic year 2022-2023(Even sem).

SIGNATURE	**SIGNATURE**
Mrs. D. Viji Assistant Professor Computing Technology	Dr.M. PUSHPALATHA HEAD OF THE DEPARTMENT Department of computing Technologies

ABSTRACT

This Courier Management System Project will have different modules. The login section will have login facility for the admin and for the user who will operate this system. While taking orders from its customers, it will take all the details of its customers who is placing the orders and all the details for the recipient such as its address, name, mobile number. Through the tracking id, customers or its recipient will able to track their products from any location using internet. It will provide status of the product after placing orders within 1 minute. The admin can manipulate the data through admin login page and add any new consignment if required. The profile section shows the data of the user and the pricing section of the project shows the price that will be charged for the consignment according to the weight categories.

Using the courier service person can easily send his/her parcel to other person in the particular destination within the time.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AES	Advanced Encryption Standard
CSS	Cascading Style Sheet
DBMS	Database Management System
ER	Entity Relationship
HTML	Hypertext Markup Language
MySQL	My Structured Query Language
SQL	Structured Query Language
PHP	Hypertext Preprocessor
XAMPP	Cross Platform Apache MySQL PHP Perl
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

A courier management system is a software solution designed to streamline and optimize the process of managing courier and delivery services. It is an end-to-end solution that enables businesses to manage their entire delivery operations from a single platform, from accepting orders and dispatching couriers to tracking packages and handling payments.

A courier management system typically includes features such as order management, dispatching, real-time package tracking, customer notifications, driver management, billing and invoicing, and reporting and analytics. It may also integrate with third-party logistics providers and transportation companies to extend the reach and capacity of the courier service.

By using a courier management system, businesses can increase their operational efficiency, reduce errors, and improve customer satisfaction. It can help them handle a high volume of orders and deliveries, and provide visibility and transparency to customers throughout the delivery process. Overall, a courier management system can help businesses stay competitive and grow their operations.

1.2 PROBLEM STATEMENT

PROBLEM DEFINITION:

The Current software courier management system provide Transition of Goods across States or Nation, and Nowadays People Face issue to Transit Goods across City Limits Due to Busy Day to Day Schedule.

Our client is not using its own online system for courier management. They are using some third-party tool to manage the consignment. There was no customisation for the client.

It was a very basic tool just to update the status of consignment, the client was maintaining billing in excels, shipment in excel too. There was no proper management of offers. All these works are a management headache without a proper system to manage.

PROBLEM SOLUTION:

As per above mentioned problem we introduced a courier management with the concept for small distance. With low cost and fast delivery. Also provided same Tracking system which we provided through out world and across the cities.

The system will be used for day-to-day activities like out return, company details, hub rates, booking, non-delivery and pickup centres. It is not easy to do this process manually because it would become very hectic.

Hence it is suggested to automate the process by developing the relevant software as the world is moving from manual working to information and technology era where automation becomes important in all part of life.

The main purpose of this system is to connect all branches to centre database so the everywhere information is same. This system increases the efficiency and increases the customer satisfaction level.

1.3 OBJECTIVES

The software aims at making the work of the entire staff of the Courier Department very easy, to save their time. To maintain their documents, to maintain their tasks performed in timely manner. The system saves your time and gives out the accurate documentation.

The proposed system has been designed in such a way that can be operated by various kinds of users. The working of software is very easy to understand it is very graphical and it is user friendly.

This new system is built with the following objective:

1. Information retrieval will become easy.
2. Maintenance of database as well as overall project will become easy.
3. Security measure will be adopted, by maintaining the login of username and the password.
4. Data redundancy will be greatly reduced.

1.4 SCOPE AND APPLICATIONS

It may help collect perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Courier Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

Our project aims at Business process automation, i.e., we have tried to computerize various processes of Courier Management System.

o In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.

o In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.

o To assist the staff in capturing the effort spent on their respective working areas.

o To utilize resources in an efficient manner by increasing their productivity through automation.

o The system types of information that can be used for various purposes.

o It satisfies the user requirement.

o Be easy to understand by the user and operator.

o Be easy to operate.

o Have a good user interface.

o Be expandable.

o Delivered on schedule within the budget.

1.5 GENERAL AND UNIQUE SERVICES IN THE DATABASE

5.1. LIST OF GENERAL SERVICES:

1. User Register:

If the user is a new user, they will need to register by providing their details such as name, email address, phone number, etc. Once registered, they can use the courier management service.

2. Users Login:

The user will login to the system using their registered email ID and password. If they are a new user, they will need to create a new account by providing their details such as name, email address, phone number, etc.

3. Price Catalogue:

After logging in, the user will be presented with a list of available prices to select from. The price catalogue module will provide information about the price of delivering the courier according to the weigt of the parcel.

4. Courier Catalogue:

Once the user has selected the price, they can place the order from the courier catalogue model according to the weight of their parcel by uploading the sender's and receiver's details.

5. Track Catalogue:

Once the user has placed the order, the user can track the status of delivery of parcel. They can view whether the parcel has been delivered or yet to be delivered.

6. Admin Login:

The administrator can login into the system using his registered mail id and password. They have all the sender's and receiver's address details for a particular parcel.

5.2 LIST OF UNIQUE SERVICES:

1. Details Storage:

The database stores all the details of the orders.The administrator have all the sender's and receiver's address details for a particular parcel. They can modify or delete the details of the users.

2. Information Storage:

The administrator has the information about the users. They can modify the details of the users.

3. Modification of Orders:

The administrator has the authority to modify the details of the parcel that is to be delivered.

4. Review Catalogue:

The review catalogue helps the users to give their valuable feedback of their experience of the functionality of the entier courier management system.

1.6 SOFTWARE REQUIREMENT SPECIFICATION

1. HTML:

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and applications. Each page contains a series of connections to other pages called hyperlinks. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are delineated by tags, written using angle brackets. Tags such as <img> and <input/> introduce content into the page directly. Others such as <p>....</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance.

2. CSS:

CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in HTML (Hypertext Markup Language). It is used to style and layout web pages, including those that interact with a database management system (DBMS). Here are some ways CSS can be used in building web applications that interact with a DBMS:

Styling User Interface Components: CSS can be used to style various user interface components in web applications that interact with a DBMS, such as forms, tables, and charts. By using CSS, developers can customize the look and feel of these components to match the branding and visual design of the application.

3. Responsive Design:

CSS can be used to create responsive designs that adapt to different screen sizes and device types. This is especially important in web applications that interact with a DBMS, as users may need to access the application from different devices with varying screen sizes.

4. Animations and Transitions:

CSS can be used to add animations and transitions to user interface components in web applications that interact with a DBMS. This can improve the user experience by providing visual feedback to users when they interact with the application.

5. Accessibility:

CSS can be used to improve the accessibility of web applications that interact with a DBMS by providing high contrast, appropriate font sizes, and other design elements that make the application more accessible to users with disabilities.

6. Theming:

CSS can be used to create different themes for web applications that interact with a DBMS. By using CSS variables, developers can create different color

schemes, font sizes, and other design elements that can be switched dynamically based on user preferences or other application settings.

Overall, CSS plays an important role in building web applications that interact with a DBMS by providing design and layout elements that improve the user experience and make the application more accessible to users with different needs.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

The existing system for courier management system can vary depending on the specific company or organization that uses it. However, in general, the existing system may involve manual processes for managing courier orders, deliveries, and tracking. This may include the use of paper-based records, manual data entry, and phone or email communication with customers.

Some companies may use basic software tools such as spreadsheets or basic database systems to manage their courier operations. However, these tools may lack advanced features such as real-time tracking, automatic routing optimization, and integration with other systems such as GPS, inventory management, or billing systems.

Other organizations may use more sophisticated courier management systems that are specifically designed for the courier industry. These systems may include features such as order management, dispatching, route planning, delivery tracking, proof of delivery, and customer communication.

Overall, the existing system for courier management may vary widely depending on the size and complexity of the organization and the specific needs of the courier industry they serve.

2.2 COMPARISON OF EXISTING VS PROPOSED SYSTEM

To compare the existing system and proposed system for a courier management system, we need to consider the differences in features, functionality, and efficiency. Here are some potential differences that could exist between the two systems:

Features:

The proposed system for courier management system is likely to have more features than the existing system. For example, it could include real-time tracking, automatic routing optimization, integration with GPS and inventory management systems, proof of delivery, customer communication, and billing integration.

Functionality:

The proposed system is likely to be more functional than the existing system. For example, it could automate many of the manual processes currently involved in courier management, such as order processing, dispatching, and delivery tracking. The system could also provide more accurate and up-to-date information on courier orders, delivery status, and customer communication.

Efficiency:

The proposed system is likely to be more efficient than the existing system, which may involve manual processes and data entry. The proposed system could reduce errors and delays in courier operations, which could lead to better customer satisfaction, more efficient delivery routes, and cost savings.

Scalability:

The proposed system could be more scalable than the existing system, allowing for growth and expansion of the courier business. The system could handle a higher volume of courier orders and deliveries, while still maintaining high levels of efficiency and accuracy.

In conclusion, the proposed courier management system is likely to offer more features, functionality, efficiency, scalability, and cost-effectiveness than the existing system. By automating and streamlining many of the manual processes involved in courier management, the proposed system could improve delivery operations, reduce errors, and enhance the customer experience.

CHAPTER 3

ARCHITECTURE AND DESIGN SYSTEM

3.1 ARCHITECTURE DIAGRAM

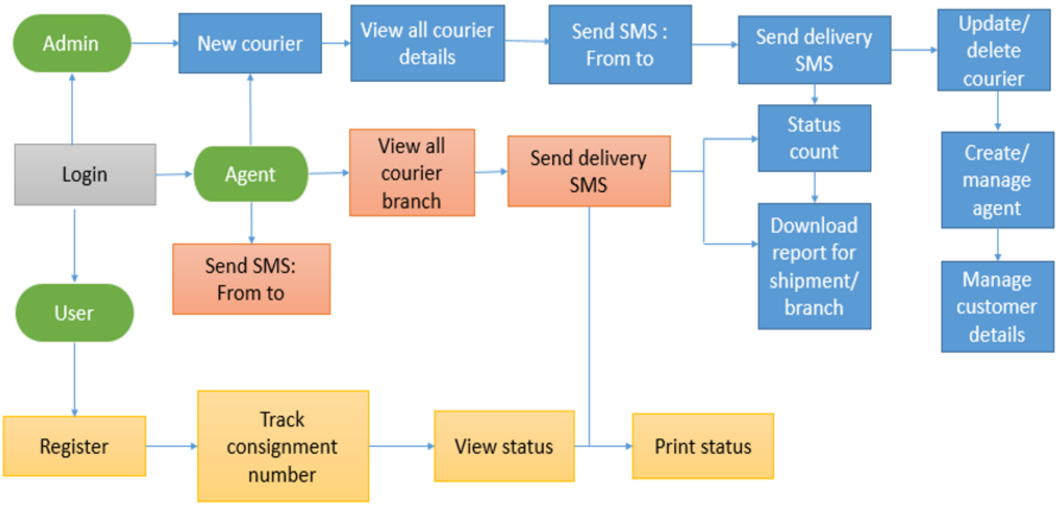


Fig 3.1

Fig. 3.1 represents the architecture diagram of the courier management system.

A courier management system typically consists of several components that work together to facilitate the delivery of packages. These components can include:

User Interface: The user interface is the component that allows users to interact with the system. It could be a web-based interface, a mobile app, or a desktop application.

Customer Management: This component manages customer data, including contact information, delivery addresses, and billing information.

Order Management: This component handles the creation and tracking of orders, including package details, delivery addresses, and delivery dates.

Courier Management: This component is responsible for managing courier details, such as courier profiles, availability, and schedules.

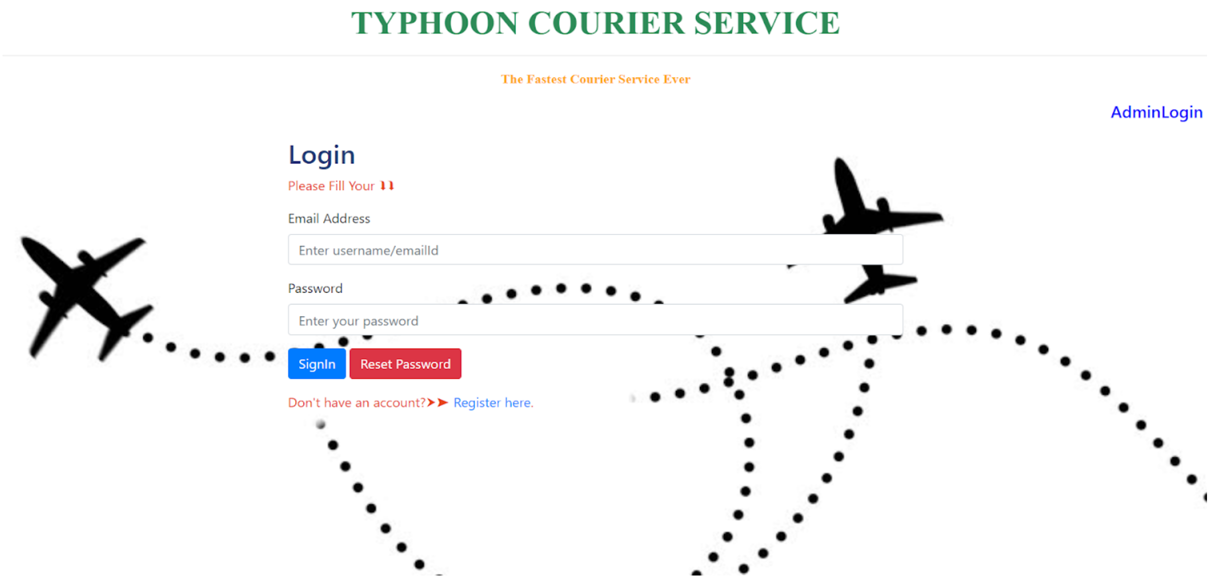
Inventory Management: This component tracks inventory levels and manages stock in the warehouse, including incoming and outgoing shipments.

Payment Gateway: This component manages the payment transactions between customers, couriers, and the company.

Reporting and Analytics: This component provides reports and analytics on various aspects of the system, including delivery times, courier performance, and inventory levels.

3.1.1 FRONT END (UI) DESIGN

3.1.1.1 USER LOGIN PAGE



TYPHOON COURIER SERVICE

The Fastest Courier Service Ever

Login

Please Fill Your ⬇⬇

Email Address: Enter username/emailId

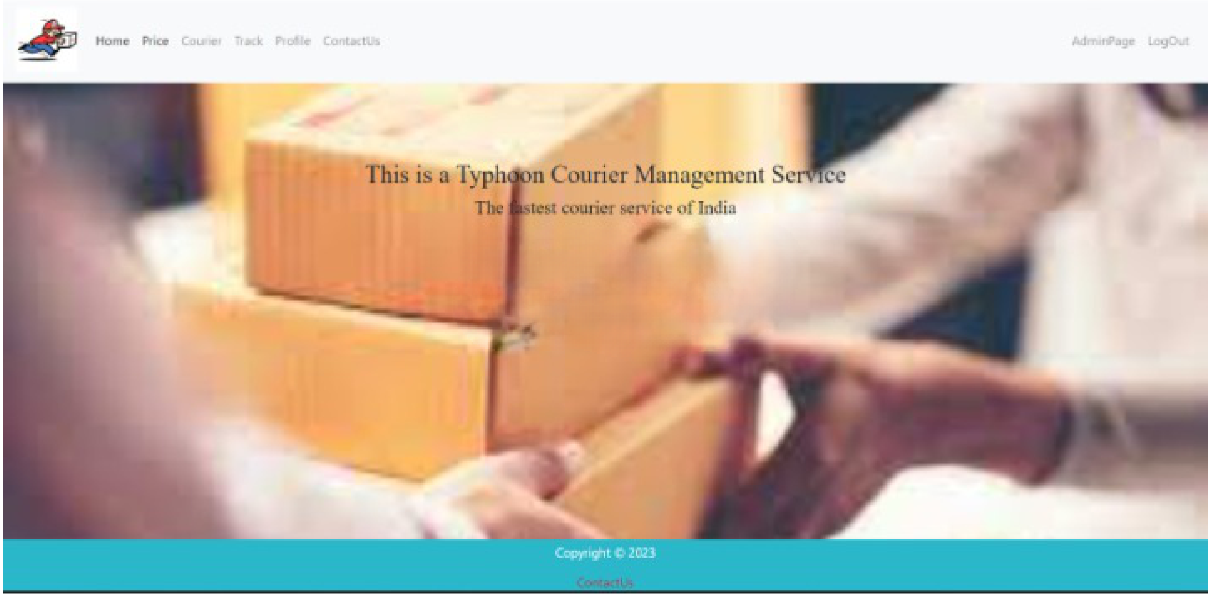
Password: Enter your password

Don't have an account?➤➤ Register here.

Fig 3.2

The fig 3.2 shows the user login interface of the website/portal. An existing user can login into the website through this page. If a user is new they can register through the register option given. An administrator can also login into their account by clicking on the admin login button.

3.1.1.2 HOME PAGE



This is a Typhoon Courier Management Service

The fastest courier service of India

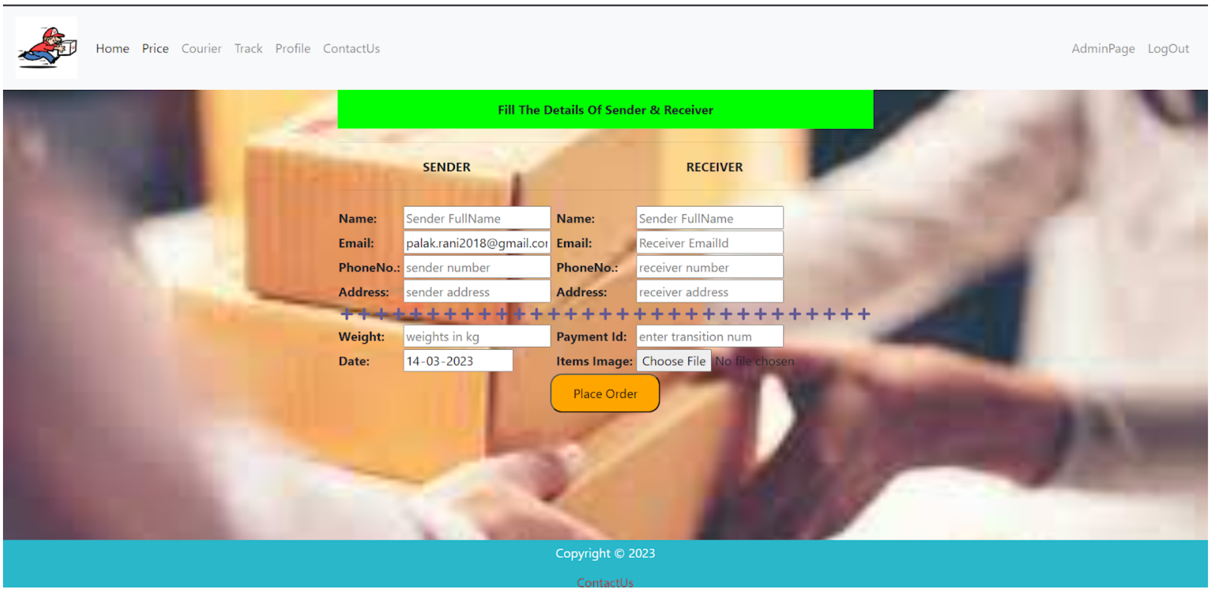
Copyright © 2023

ContactUs

Fig 3.3

Fig 3.3 is the home page of the website/portal that redirects to other pages. It contains a simple header and the menu selector to go to the other connected pages of the website. It can also redirect to the admin page through the AdminPage selector.

3.1.1.3 COURIER SENDING PAGE



Fill The Details Of Sender & Receiver

SENDER		RECEIVER	
Name:	Sender FullName	Name:	Sender FullName
Email:	palak.rani2018@gmail.cor	Email:	Receiver EmailId
PhoneNo.:	sender number	PhoneNo.:	receiver number
Address:	sender address	Address:	receiver address
Weight:	weights in kg	Payment Id:	enter transition num
Date:	14-03-2023	Items Image:	Choose File No file chosen

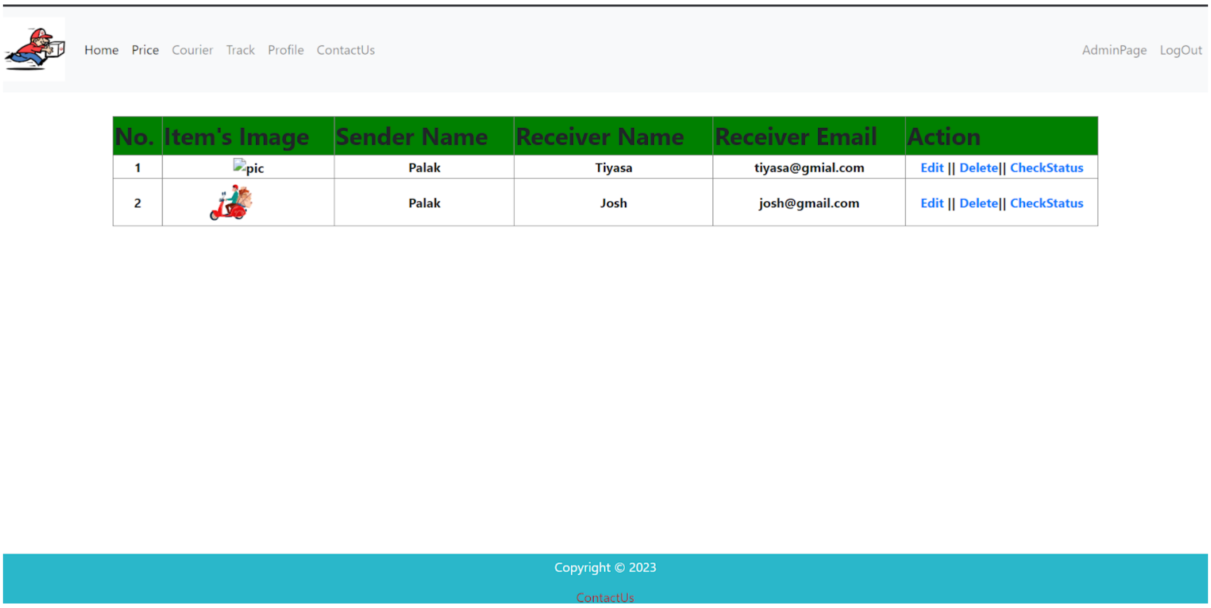
Copyright © 2023

ContactUs

Fig 3.4

Fig 3.4 is the page that pops up when the courier selector icon is clicked. It asks the user to fill in all the details of the sender sending the courier and of the receiver receiving the courier. It also asks the user to fill in the details of the parcel and asks the user to choose a payment method.

3.1.1.4 TRACK CONSIGNMENT PAGE



No.	Item's Image	Sender Name	Receiver Name	Receiver Email	Action
1	pic	Palak	Tiyasa	tiyasa@gmial.com	Edit || Delete|| CheckStatus
2		Palak	Josh	josh@gmail.com	Edit || Delete|| CheckStatus

Fig 3.5

Fig 3.5 is the page that pops up when the track selector icon is clicked. It shows the user the status of their parcel. It tells the user that if their parcel is delivered or is yet to be delivered.

SOFTWARE USED

1. HTML : Hypertext Markup Language (HTML) is the standard markup language for creating web pages and applications. Each page contains a series of connections to other pages called hyperlinks. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are delineated by tags, written using angle brackets. Tags such as <img> and <input/> introduce content into the page directly. Others such as <p>….</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML code ensures the proper formatting of text and images so that your Internet browser may display them as

they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance.

2. CSS: CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in HTML (Hypertext Markup Language). It is used to style and layout web pages, including those that interact with a database management system (DBMS). Here are some ways CSS can be used in building web applications that interact with a DBMS:

Styling User Interface Components: CSS can be used to style various user interface components in web applications that interact with a DBMS, such as forms, tables, and charts. By using CSS, developers can customize the look and feel of these components to match the branding and visual design of the application.

Responsive Design: CSS can be used to create responsive designs that adapt to different screen sizes and device types. This is especially important in web applications that interact with a DBMS, as users may need to access the application from different devices with varying screen sizes.

Animations and Transitions: CSS can be used to add animations and transitions to user interface components in web applications that interact with a DBMS. This can improve the user experience by providing visual feedback to users when they interact with the application.

Accessibility: CSS can be used to improve the accessibility of web applications that interact with a DBMS by providing high contrast, appropriate font sizes, and other design elements that make the application more accessible to users with disabilities.

Theming: CSS can be used to create different themes for web applications that interact with a DBMS. By using CSS variables, developers can create different color schemes, font sizes, and other design elements that can be switched dynamically based on user preferences or other application settings.

Overall, CSS plays an important role in building web applications that interact with a DBMS by providing design and layout elements that improve the user experience and make the application more accessible to users with different needs.

3.1.2 BACK END (DATABASE) DESIGN

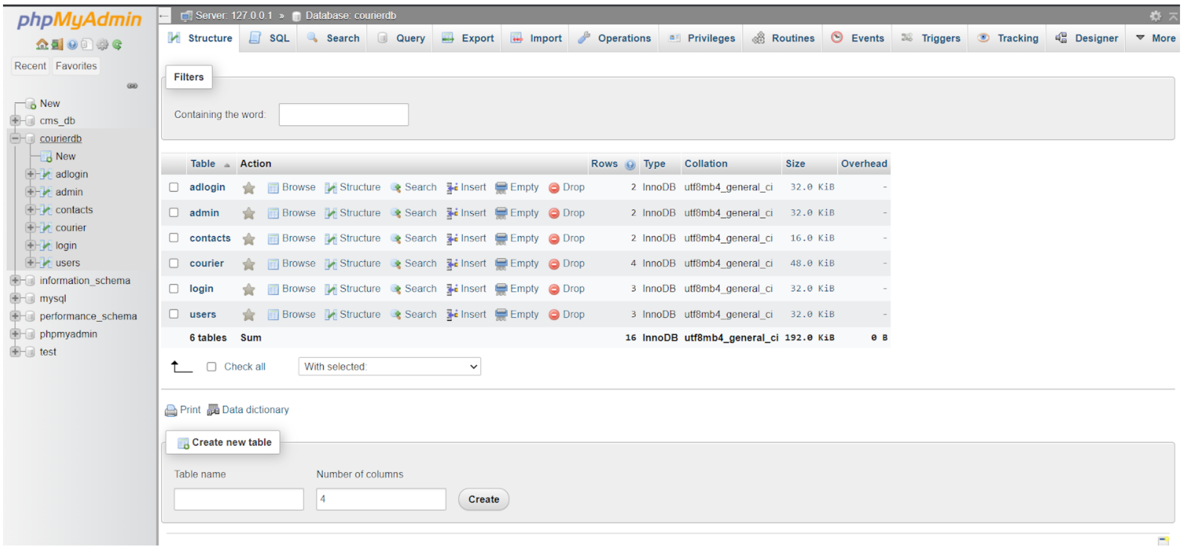


	Table	Action	Rows	Type	Collation	Size	Overhead
□	adlogin	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□	admin	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□	contacts	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	courier	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□	login	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□	users	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
	6 tables	Sum	16	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B

Fig 3.6

Fig 3.6 is the image that represents the backend connectivity to MySQL database wherein the user’s information is stored for future reference.From the track gateway page the Backend gets connected where we use MySQL and PHP to store the track history and all details about the users.

1. Storing customer information:

MySQL can be used to store customer information, such as name, sender’s address, and receiver’s address. This information can be encrypted and secured using MySQL's security features to protect customer data from unauthorized access.

2. Recording tracking details:

MySQL can be used to record tracking details such as checking whether the parcel is delivered or is yet to be delivered. This data can be used for auditing and reporting purposes.

3. PHP:

PHP is used for the backend and can connect to the database very easily. This is because PHP is designed with specific capabilities to facilitate these connections in an efficient way. It's no surprise the PHP is part of the widely popular LAMP stack.

4. Xampp-SQL Server--Apache Server:

XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. This collection of software contains a web server named Apache, a database management system named MariaDB and scripting/ programming languages such as PHP and Perl. X denotes Crossplatform, which means that it can work on different platforms such as Windows, Linux, and macOS.

3.2 ER DIAGRAM AND USE CASE DIAGRAM

3.2.1 ER DIAGRAM

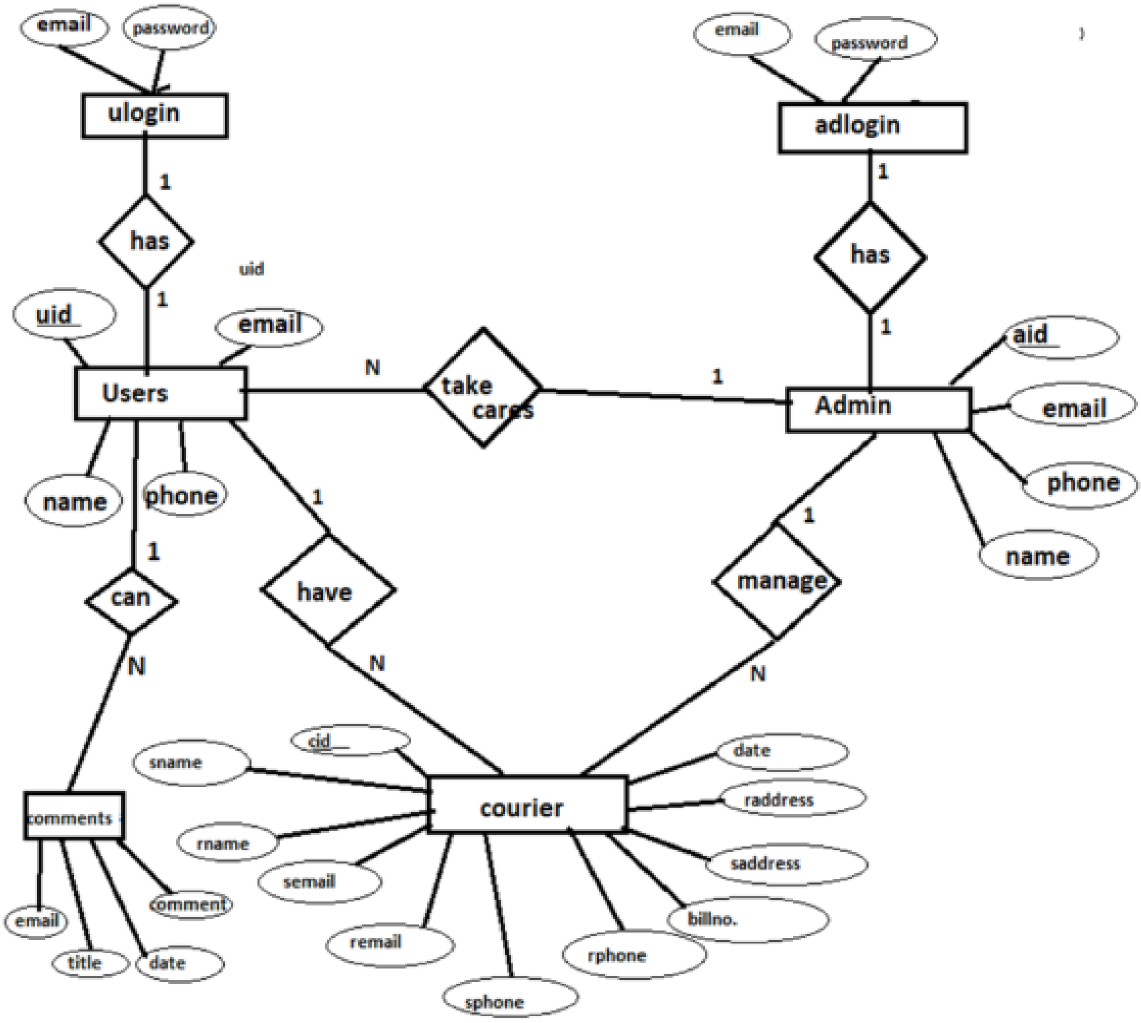


Fig 3.7

Fig 3.7 shows the ER (Entity Relationship) Diagram represents the model of Courier Management System Entity. The entity-relationship diagram of Courier Management System shows all the visual instruments of database tables and the relations between the Customer, Admin, Courier, Delivery, etc. It used structure data and to define the relationships between structured data groups of Courier Management System functionalities.

In this ER diagram, we have the following entities:

User: This entity represents the person or organization that sends the courier.

Courier: This entity represents the package or item that is being sent.

Admin: This entity represents the admin of the company.

The relationships between these entities are as follows:

A customer can send one or many couriers.

A courier can be sent by only one customer.

A courier can be delivered by only one employee.

A courier can be sent from one branch and delivered to another branch.

3.2.2 USE CASE DIAGRAM

There are two actors in the courier management system use case

1. Users
2. Administrator

USER

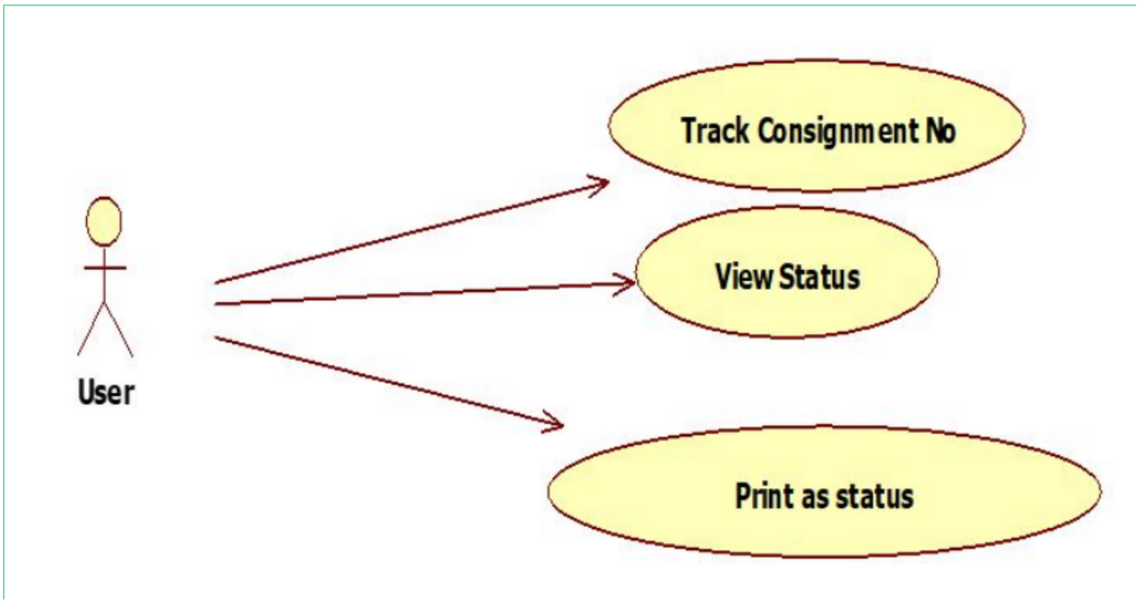


Fig 3.8

Fig 3.8 shows the use case diagram for the user and the actions that can be performed by a user. The user can register and login into the system using the registered mail id and password. The user can make a request for courier delivery by choosing from the various options available and filling the required details. The user can check and print the status of the courier using the track courier option given to the user.

ADMIN

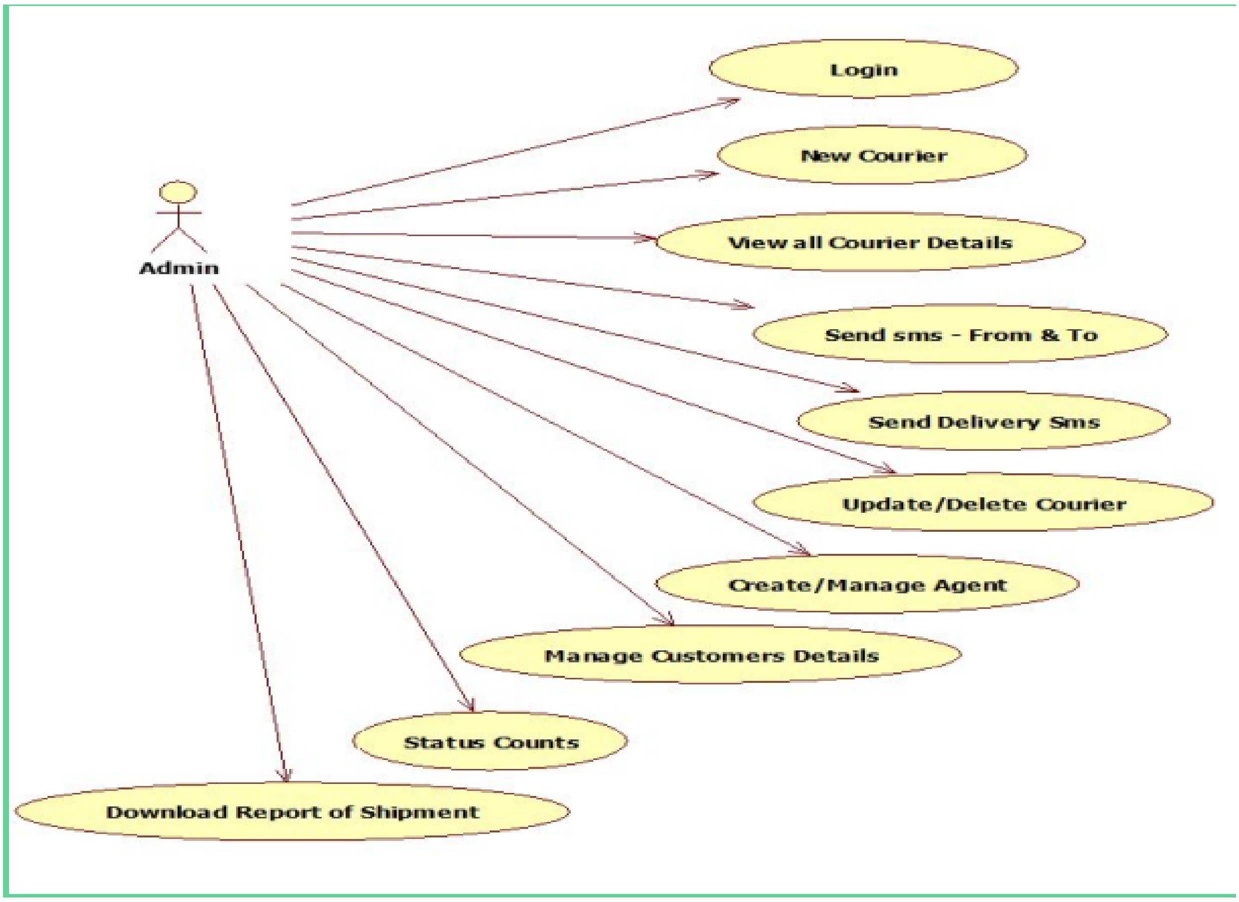


Fig 3.9

Fig 3.9 shows the use case diagram for admin and the actions that can be performed by admin. The admin can login into the system using the username and password. The admin can manage all the details about the user and the couriers. The admin can delete the user and courier details, can track the couriers and check the status of the couriers. The admin can update and manage the shipment reports and can count the status of the courier.

CHAPTER 4

MODULES AND FUNCTIONALITIES

4.1 LIST OF MODULES

4.1.1. Owner Profile: System Administration

This refers to the profile of the person or group responsible for managing the overall system. They have access to the highest level of privileges and are responsible for overseeing the entire system and its performance.

4.1.2. User Profile: User Authentication

This refers to the profile of a user who is accessing the system. User authentication is the process of verifying the identity of a user, usually through a username and password, before allowing them access to the system.

4.1.3. Courier Booking: Placing Order

This refers to the process of placing an order for delivery of the parcel by filling in the sender's and receiver's addresses and selecting the payment method.

4.1.4. Price Module: Price Order

This refers to the process of presentation of the various prices for the delivery of the parcel based on the weight of the parcel to be delivered.

4.1.5. Courier Tracking: Tracking Management

This refers to the process of tracking of the parcel that is to be delivered. It enables the user to view the status of their parcel.

4.1.6. Add/update/delete courier Details: Content Management

This refers to the process where the admin has the access to change or delete the details of the users and their parcel’s to be delivered.

4.2 CONNECTIVITY USED FOR DATABASE ACCESS

4.2.1. MySQL

MySQL is a highly popular database management system that can power projects of all sizes. Its ability to handle huge volumes of data without breaking a sweat is one of its biggest selling points. Connect MySQL with PHP code, you can make use of one of three methodologies.

1. Create MySQL Database at the Localhost

1.1. Create Database

1.2. Create a Folder in htdocs

1.3. Create Database Connection File In PHP

1.4. Create new php file to check your database connection

1.5. Run it

2. Create MySQL Database at Cloudways Server

2.1. Create Database Connection

2.2. MySQLi Procedural Query

2.3. Connect MySQL Database with PHP Using PDO

2.4. Check Connection

3. Remote MySQL

4. Top MySQL Management tools

4.1. MySQL Workbench

4.2. Navicat For MySQL

4.3. MySQL Yog

4.4. Cloudways MySQL Database Manager

CHAPTER 5

CODING AND TESTING

Admin Login

```
<!-- admin logIn page and login logic -->
<?php

session_start();
if (isset($_SESSION['uid'])) {

    header('location: dashboard.php');
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Login</title>
</head>

<body bgcolor="#E0E0E0">
         <h5><a href="../index.php" style="float: right; margin-right:50px;
color:#6D6D6D">BackToHome</a></h5><br>
    <h1 align='center' style="color: #6D6D6D;font-size:60px">Admin Login</h1>
                 <h6 align='center' style="color: #212121;font-weight:
bold;font-size:15px">Welcome to our services!</h6>
    <form action="adminlogin.php" method="POST" style="margin: auto;">
        <table align="center">
            <tr>
                <td>Email_ID:</td>
                <td><input type="email" name="uname" require></td>
            </tr>
            <tr><td><br></td></tr>
            <tr>
                <td>Password:</td>
                <td><input type="password" name="pass" require></td>
            </tr>
            <tr>
                <td colspan="2">
                    <hr>
                </td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input type="submit" name="login"
value="Login" style="cursor: pointer;"></td>
            </tr>
        </table>
    </form>
</body>

</html>

<?php

include('../dbconnection.php');
if (isset($_POST['login'])) {
    $ademail = $_POST['uname'];
    $password = $_POST['pass'];
    $qry = "SELECT * FROM `adlogin` WHERE `email`='$ademail' AND
`password`='$password'";
    $run = mysqli_query($dbcon, $qry);
    $row = mysqli_num_rows($run);
    if ($row < 1) {
        ?>
        <script>
            alert("Only admin can login..");
            window.open('adminlogin.php', '_self');
        </script><?php
    }
    else {
        $data = mysqli_fetch_assoc($run);
        $id = $data['a_id'];
        $_SESSION['uid'] = $id;
        header('location:dashboard.php');
    }
}
?>
```

Registration Page

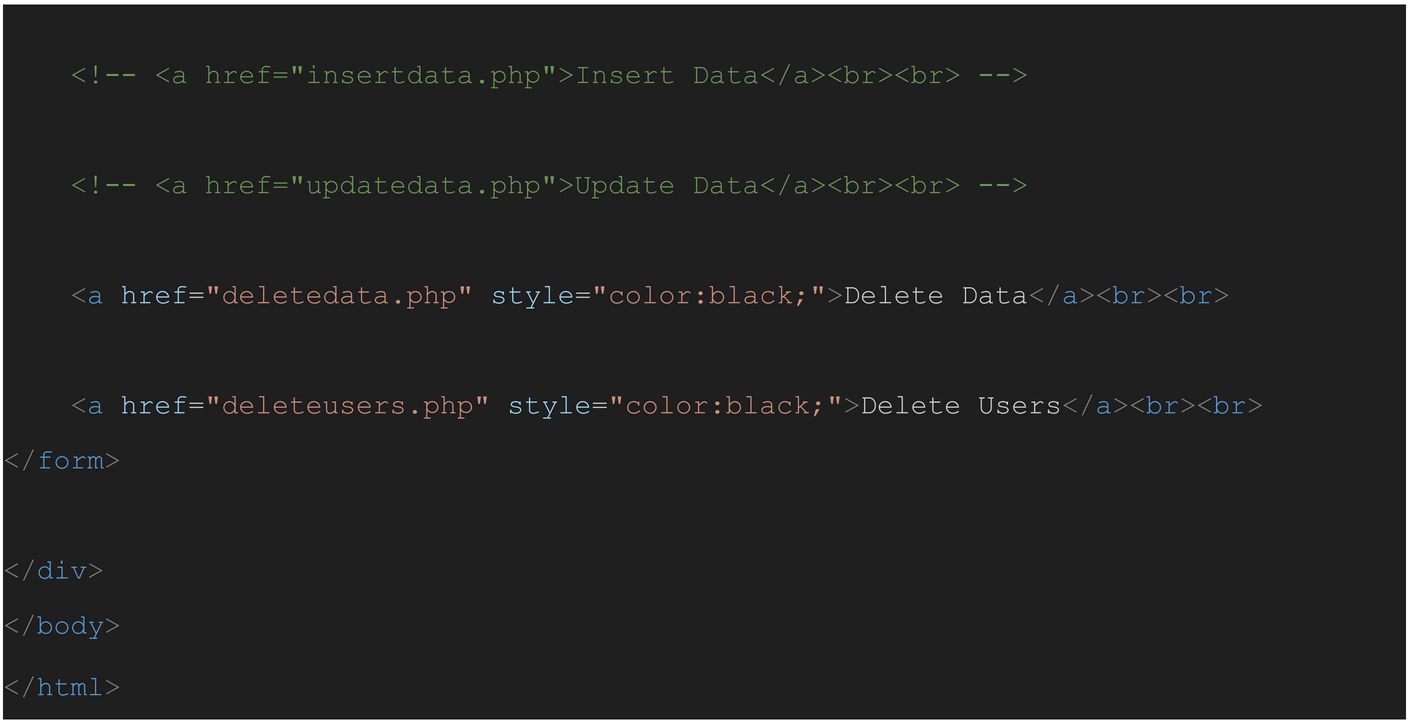
```
<?php
require_once "dbconnection.php";
require_once "session.php";

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {

    $fullname = $_POST['name'];
    $phn = $_POST['ph'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
    if($password==$confirm_password){

      $qry = "INSERT INTO `users` (`email`, `name`, `pnumber`) VALUES ('$email',
'$fullname', '$phn')";
    $run = mysqli_query($dbcon,$qry);

    if($run==true){

             $psqry = "INSERT INTO `login` (`email`, `password`, `u_id`) VALUES
('$email', '$password',LAST_INSERT_ID() )";
        $psrun = mysqli_query($dbcon,$psqry);
        ?>  <script>
            alert('Registration Successfully :)');
            window.open('index.php','_self');
            </script>
        <?php
    }
    }else{
        ?>  <script>
            alert('Password mismatched!!');
            </script>
        <?php
    }
```

```
}
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Sign Up</title>
                                                        <link       rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
        <style>
        body
        {

        background-repeat: no-repeat;
        background-size: cover;
        }
    </style>
    </head>
    <body><br>
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <h2 style="color:green">Register</h2>
                    <p>Please fill this form to create an account.</p>
                    <!-- <?php echo $success; ?>
                    <?php echo $error; ?> -->
                    <form action="" method="post">
                        <div class="form-group">
                            <label>Full Name</label>
                                <input type="text" name="name" class="form-control"
required>
                        </div>
                        <div class="form-group">
                            <label>Phone Num.</label>
```

```
                                <input type="number" name="ph" class="form-control"
required>
                        </div>
                        <div class="form-group">
                            <label>Email Address</label>
                              <input type="email" name="email" class="form-control"
required />
                        </div>
                        <div class="form-group">
                            <label>Password</label>
                                            <input type="password" name="password"
class="form-control" required>
                        </div>
                        <div class="form-group">
                            <label>Confirm Password</label>
                                     <input type="password" name="confirm_password"
class="form-control" required>
                        </div>
                        <div class="form-group">
                                     <input type="submit" name="submit" class="btn
btn-danger" value="Register">
                        </div>
                                  <p>Already have an account? <a href="index.php"
style="color: red;">Login here</a>.</p>
                    </form>
                </div>
            </div>
                <hr><p>Notice: If the email Id is registered before, it will not
respond.</p>
             <p>In this case, reset your password or register with different email
Id....</p>
        </div>
    </body>
</html>
```

Dashboard page

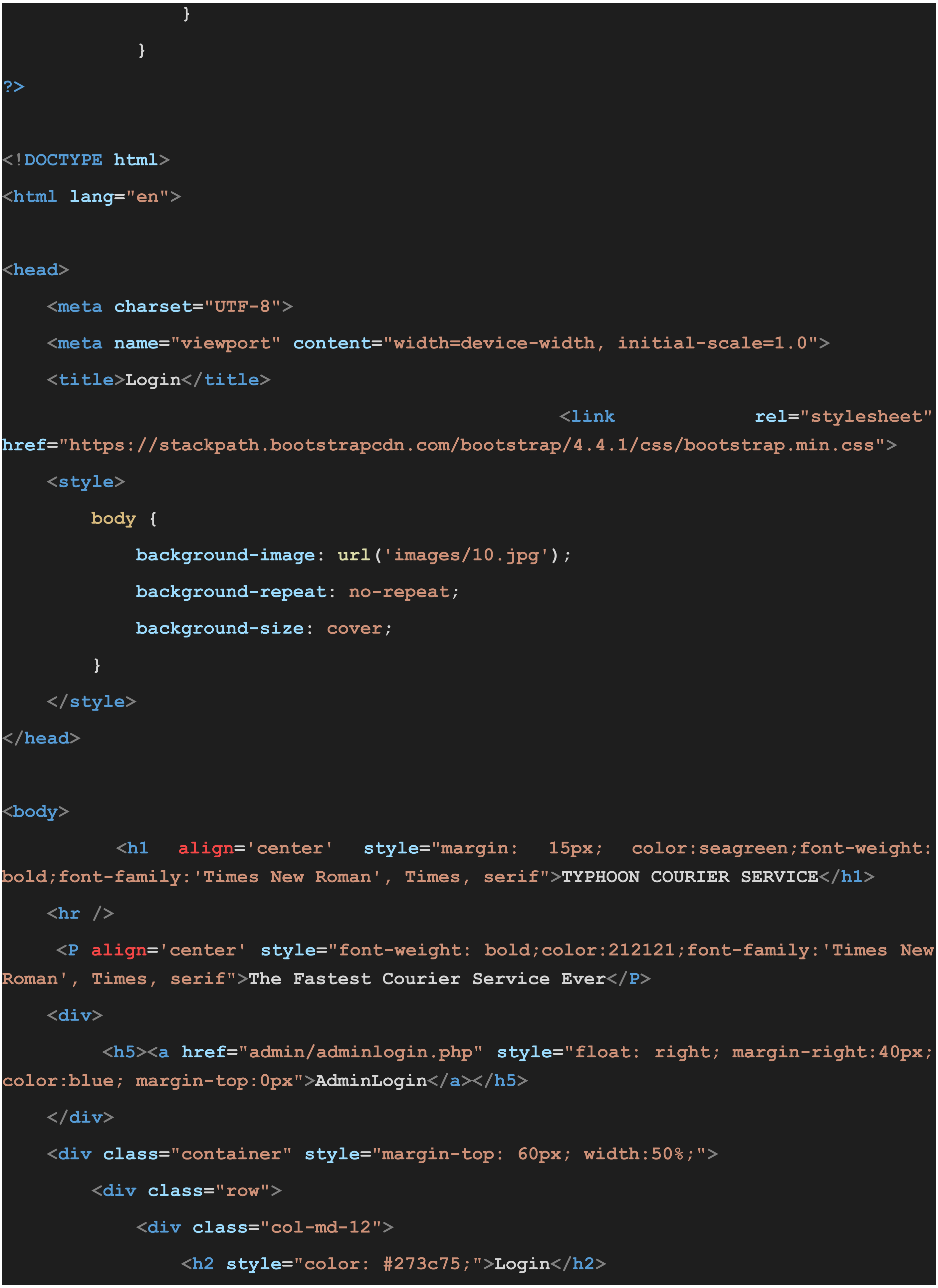
```
<!-- admin dashbord page with options for admin -->


<?php
session_start();
if(isset($_SESSION['uid'])){
    echo "";
}else{
    header('location: ../login.php');
}


?>


<?php
include('head.php');
?>


<style>
    body {


        background-size: cover;
        }
</style>


<div class="admintitle">
    <div>
          <h5 ><a href="../index.php" style="float: left; margin-left:20px; color:aliceblue;">LoginAsUser</a></h5>
          <h5 ><a href="logout.php" style="float: right; margin-right:20px; color:aliceblue;">LogOut</a></h5>
    </div>
     <h1 align='center' style="text-shadow: 0.1em 0.1em 0.15em #f9829b;">Welcome To Admin Dashbord</h1>
</div>
<div align="center" style="margin-top:240px;">
<form style="position: center;color:lightblue;font-weight:bold;font-size:50px">
```



```
    <!-- <a href="insertdata.php">Insert Data</a><br><br> -->

    <!-- <a href="updatedata.php">Update Data</a><br><br> -->

    <a href="deletedata.php" style="color:black;">Delete Data</a><br><br>

    <a href="deleteusers.php" style="color:black;">Delete Users</a><br><br>
</form>

</div>
</body>
</html>
```

Home Page

```
<?php

require_once "dbconnection.php";
require_once "session.php";

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {

    $email = $_POST['email'];
    $password = $_POST['password'];

            $qry = "SELECT * FROM `login` WHERE `email`='$email' AND
`password`='$password'";
    $run = mysqli_query($dbcon, $qry);
    $row = mysqli_num_rows($run);
    if ($row < 1) {
?>
        <script>
            alert("Opps! plz Enter Your Username and Pswd again..");
            window.open('index.php', '_self');
        </script> <?php
                } else {
                    $data = mysqli_fetch_assoc($run);
                    $id = $data['u_id'];    //fetch id value of user
                    $email = $data['email'];
                       $_SESSION['uid'] = $id;    //now we can use it until session
destroy
                    $_SESSION['emm'] = $email;
                    ?>
        <script>
            alert("स्वागत हैं आपका 🙏");
            window.open('home/home.php', '_self');
            // changes made here
        </script> <?php
```



```
                }
            }
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
                                                  <link                rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <style>
        body {
            background-image: url('images/10.jpg');
            background-repeat: no-repeat;
            background-size: cover;
        }
    </style>
</head>

<body>
          <h1   align='center'   style="margin:   15px;   color:seagreen;font-weight:
bold;font-family:'Times New Roman', Times, serif">TYPHOON COURIER SERVICE</h1>
    <hr />
     <P align='center' style="font-weight: bold;color:212121;font-family:'Times New
Roman', Times, serif">The Fastest Courier Service Ever</P>
    <div>
         <h5><a href="admin/adminlogin.php" style="float: right; margin-right:40px;
color:blue; margin-top:0px">AdminLogin</a></h5>
    </div>
    <div class="container" style="margin-top: 60px; width:50%;">
        <div class="row">
            <div class="col-md-12">
                <h2 style="color: #273c75;">Login</h2>
                <p style="color:#e84118;">Please Fill Your □□</p>
                <!-- <?php echo $error; ?> -->
                <form action="" method="post">
                    <div class="form-group">
                        <label>Email Address</label>
                              <input type="email" name="email" class="form-control"
placeholder="Enter username/emailId" required />
                    </div>
                    <div class="form-group">
                        <label>Password</label>
                        <input type="password" name="password" class="form-control"
placeholder="Enter your password" required>
                    </div>
                    <div class="form-group">
                         <input type="submit" name="submit" class="btn btn-primary"
value="SignIn" />
                             <button onclick="window.location.href='resetpswd.php'"
class="btn btn-danger" style="cursor:pointer">Reset Password</button>
                    </div>
                            <p style="color: #e84118;">Don't have an account?□➤ <a
href="register.php">Register here</a>.</p>

                </form>
            </div>
        </div>
    </div>
</body>

</html>
```

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 RESULTS

- The Courier Management System project successfully allows users to log in with their username and password, send and receive courier online without any hustle, and track the delivery status from their own devices.
- The admin can manage the database that consists of the details of the users sending and receiving couriers, tracking ids, sending and receiving addresses, etc.
- The admin can keep a track of all the orders and their details and can also delete the users who no longer use the service.
- The system provides secure connectivity using PHP, and MySQL database.
- The project is built using Frontend and Backend technologies which include HTML, CSS, JS, MySQL, and PHP technologies.
- The special feature in the project is to allow users to track their courier online.

6.2 DISCUSSIONS

At the beginning of the project I set out certain objectives which ought to be accomplished by the end of the system creation, at the end I was able to achieve most of the objectives set out.

Developed a user friendly interface

The system incorporates a user-friendly interface whereby customers can easily access different elements at ease and have the option to go back to the top of the Login Page at ease. The user can access home, menus, reservations, services, contact, Staff Homepages at ease and manipulate the components such as making orders, sending messages and checking out.

Developed a database for storing all the information

This system incorporates a database by the name multi_login. The database stores information of the users in user table, it stores system settings in system setting table, the parcel information in parcel table, parcel tracks in parcel track table, mpesa verification in pa table and branches in branch table.

Incorporated Fast payment of the courier

The system does not yet incorporate the m-pesa API however, I incorporated payment verification that verifies whether the payment has been received in the account to ease in lying by unscrupulous merchants.

Developed a back-end system only accessible to the administrator

This system incorporates a database by the name multi_login. The database stores information of the users in the user table, and the changes that the admin makes at any given time.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

To summarize, the world is rapidly evolving and heading toward technical expertise. Technology is not a static or stagnant field, but rather one that is constantly changing as new trends arise. As patterns change and improve, it's past time for us to change with them. The use of online courier management systems is important for getting accountability and making goods get delivered quickly and making the work easier. As a result, this system would make it easier for Typhoon courier companies to be accountable for all their services.

7.2 FUTURE ENHANCEMENT

- Real-time package tracking: Integrating GPS technology into the courier management system would allow customers to track their packages in real-time, providing them with up-to-date information on the package's location and estimated delivery time.
- Automated scheduling: Developing an algorithm that automatically assigns delivery routes to couriers based on factors such as location, traffic, and delivery time would optimize delivery schedules and minimize delays.
- Predictive analytics: Using machine learning algorithms to analyze data on delivery times, traffic patterns, and other factors could help predict future delivery times and optimize delivery routes.
- Mobile optimization: Creating a mobile application for the courier management system would allow couriers to easily access delivery information on-the-go, update delivery statuses, and communicate with customers in real-time.
- Customer self-service portal: Providing customers with a self-service portal to track their packages, schedule deliveries, and update delivery preferences would improve customer satisfaction and reduce the workload on customer service representatives.

- Integration with eCommerce platforms: Integrating with popular eCommerce platforms such as Amazon and Shopify would allow merchants to seamlessly schedule deliveries through the courier management system, streamlining the delivery process and improving efficiency.
- Digital signature capture: Integrating digital signature capture technology into the courier management system would allow customers to sign for packages electronically, reducing the need for physical contact during the delivery process.

REFERENCES

[1] "Design and Implementation of a Courier Management System" by M. O. Adeoye, K. S. Olabode, and O. T. Oyediran. This paper describes the design and implementation of a courier management system using PHP and MySQL. The system includes features such as order management, dispatching, package tracking, and billing.

[2] "Design and Development of Courier Management System using Cloud Computing" by R. Vijaya Kumar and V. Nirmala Devi. This paper proposes a cloud-based courier management system that utilizes Amazon Web Services (AWS) to improve scalability and availability. The system includes features such as order management, dispatching, package tracking, and invoicing.

[3] "A Review of Courier Management System" by M. M. Kamath, M. B. Rao, and K. Rajasekhara. This paper provides an overview of courier management systems and their features, including order management, dispatching, package tracking, and billing. It also discusses the benefits of using a courier management system, such as increased efficiency and customer satisfaction.

[4] "Development of a Real-Time Courier Management System for E-commerce Industry" by V. M. Choudhary and S. P. Muley. This paper describes the development of a real-time courier management system for the e-commerce industry using Java and MySQL. The system includes features such as order management, dispatching, package tracking, and billing.

[5] "A Comparative Study of Courier Management Systems" by S. T. Shrivastava and N. K. Jain. This paper compares several courier management systems based on their features, performance, and usability. The systems evaluated include Shiprocket, Shipway, and Trackon Courier.

[6] "Courier Management System: A Case Study of Pakistan Post," by A. Naeem et al., in Journal of Information Systems Engineering & Management, 2020. This paper provides a case study of a courier management system implemented by Pakistan Post, including its features, benefits, and challenges.

[7] "Design and Implementation of a Courier Management System Using RFID and GPS Technologies," by A. Adeyemo and A. Adeyemi, in International Journal of Advanced Computer

Science and Applications, 2018. This paper presents the design and implementation of a courier management system using RFID and GPS technologies.

[8] "Design and Development of a Web-Based Courier Management System," by M. Hussain and M. M. Ali, in International Journal of Scientific & Engineering Research, 2014. This paper describes the design and development of a web-based courier management system, including its architecture, features, and functionalities.

[9] "Courier Management System Using Mobile Devices and Web Services," by M. T. Islam et al., in International Journal of Computer Applications, 2013. This paper proposes a courier management system that uses mobile devices and web services for tracking packages, managing couriers, and generating reports.

[10] "A Comparative Study of Courier Management Systems," by P. Sharma and P. Sharma, in International Journal of Computer Science and Mobile Computing, 2013. This paper provides a comparative study of various courier management systems, including their features, advantages, and disadvantages.

[11] "Courier Management System" by Ashish Kumar, published in the International Journal of Computer Science and Mobile Computing. This paper describes the design and development of a courier management system using ASP.NET and C# programming languages.

[12] "Development of Courier Management System" by S.S. Sood, published in the Journal of Information Engineering and Applications. This paper presents the design and implementation of a courier management system using PHP and MySQL.

[13] "An Intelligent Courier Management System using GPS and Android Application" by Rajesh Kumar Singh and Gyanendra Kumar, published in the International Journal of Computer Applications. This paper describes the development of a courier management system that utilizes GPS tracking and an Android application to manage the delivery of packages.

[14] "Design and Development of a Courier Management System for a Large-Scale Logistics Company" by Hao Zhang, published in the International Journal of Computer Science and Network Security. This paper presents the design and development of a courier management system for a large-scale logistics company using Java and Oracle database.

[15] "A Comparative Analysis of Courier Management Systems" by Nikunj Patel and Niraj Patel, published in the International Journal of Advanced Research in Computer Science and Software Engineering. This paper compares and analyzes different courier management systems and their features.