

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018.



## A MINI PROJECT REPORT

On

## “COURIER MANAGEMENT SYSTEM”

*Submitted in partial fulfillment of the requirement for the curriculum of the 5<sup>th</sup> Semester*

## Bachelor of Engineering

In

## Computer Science & Engineering

*Submitted by*

**G LIKHITH KUMAR REDDY**

**1VI21CS025**

**M RAJESH KUMAR REDDY**

**1VI21CS058**

*Under the supervision of*

**Prof. Naveen H S**

Assistant Professor

**Prof. Shilpa Reddy K**

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VEMANA INSTITUTE OF TECHNOLOGY**

**Bengaluru – 560 034**

**2023-2024**

Karnataka Reddy Jana Sangha®  
**VEMANA INSTITUTE OF TECHNOLOGY.**  
Koramangala, Bengaluru-34.  
(Affiliated to Visvesvaraya Technological University, Belagavi)



**Department of Computer Science & Engineering**

**Certificate**

This is certified that the mini-project work entitled “**COURIER MANAGEMENT SYSTEM**” carried out by **G LIKHITH KUMAR REDDY (1VI21CS025)** and **M RAJESH KUMAR REDDY (1VI21CS058)** are Bonafide students of **Vemana Institute of Technology** in partial fulfillment for the completion of **Database Management System laboratory of Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The mini-project report has been approved as it satisfies the academic requirements in respect of the mini-project work prescribed for the said degree.

**Guide-I**

Mr. Naveen H S

**Guide-II**

Ms. Shilpa Reddy K

**HOD**

Dr. M Ramakrishna

**External Viva**

Name of the Examiners

Signature with date

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

## ACKNOWLEDGEMENT

We sincerely thank **Visvesvaraya Technological University** for providing a platform to do a mini-project.

Firstly, we would like to express our deep sense of gratitude to our institute “**Vemana Institute of Technology**” that provided us an opportunity to do a mini-project entitled “**COURIER MANAGEMENT SYSTEM**”.

We thank **Dr. Vijayasimha Reddy. B. G**, Principal, Vemana Institute of Technology, Bengaluru for providing the necessary support.

We would like to place on record our regards to **Dr. M. Ramakrishna**, Professor & Head of the Department, Computer Science and Engineering for his continued support.

We would like to thank our mini-project guides **Prof. Naveen H S** Assistant Professor and **Prof. Shilpa Reddy K** Assistant Professor, Dept. of CSE for their continuous support and valuable guidance towards successful completion of the mini-project.

We would be failing in our duty if we do not thank the faculty members, batch mate, lab staff, technicians and family members for their constant support and guidance.

**Date:**

**G LIKHITH KUMAR REDDY(1VI21CS025)**

**Place:** Bengaluru

**M RAJESH KUMAR REDDY(1VI21CS058)**

## **TABLE OF CONTENTS**

<b>Chapter No</b>	<b>Title</b>	<b>Page No</b>
	<b>Abstract</b>	<b>i</b>
	<b>List of Figures</b>	<b>ii</b>
	<b>List of Abbreviations</b>	<b>iii</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1-3</b>
	1.1 Overview	2
	1.2 Features of Purposed system	3
<b>2</b>	<b>SYSTEM REQUIREMENTS</b>	<b>4-6</b>
	2.1 Hardware Requirements	4
	2.2 Software Requirements	4
	2.3 Functional Requirement	4-5
	2.4 Non-Functional Requirements	5-6
<b>3</b>	<b>DESIGN</b>	<b>7-9</b>
	3.1 Schema Diagram	7-8
	3.2 ER Diagram	8-9
<b>4</b>	<b>IMPLEMENTATION</b>	<b>10-12</b>
	4.1 Steps of Implementation	10-11

4.2 Codes Snippet

11-12

**5**

**RESULTS**

**13-15**

**CONCLUSION**

**16**

**REFERENCES**

**17**

**BIBLIOGRAPHY**

**18**

## **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

## LIST OF FIGURES

<b>Fig. No</b>	<b>Title</b>	<b>Pag No</b>
3.1	ER Schema diagram	8
3.2	ER Diagram	9
5.1	Home page of CMS	13
5.2	signup page in CMS	13
5.3	Pricing of courier	14
5.4	Courier sending page	14
5.5	Courier tracking page	15
5.6	Xampp control Pannel	15

## **LIST OF ABBREVIATIONS**

CMS	: Courier Management System
SQL	: Structured Query Language
HTTP	: Hyper Text Transfer Protocol
HMTL	: Hypertext Markup Language
EHR	: Electronic Health Records
ORM	: Object-Relational Mapping
WSGI	: Web Server Gateway Interface
DFD	: Data Flow Diagram
ER	: Entity-Relationship
API	: Application Program Interface



## CHAPTER 1

### INTRODUCTION

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. During billing process system will generate a tracking id for their products. Through this 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 courier service is one of the solutions of these problems. It is used to send some things to any person in the world within time. The courier company has number of branches, which are spread over the country or the world. So that when person wants to send things then he has to contact at nearest courier service branch. The courier company creates the schedule & gives internal/external services. The courier service work as destination office or source office.

In modern age, as time increase, needs & requirements of the person are also increased. They want more facility & try to do their task quickly & within time. But they can not get all the things at nearest market or area, so they have to import the things from any place in the world. Within the country, the things can be imported through post service. But it consumes the time & sometimes problem of damage or missing occur in the database.

The Courier Management System is a comprehensive solution for managing the courier services within an organization or across multiple branches. It provides a streamlined process for tracking and managing the delivery of packages from the point of pickup to the final destination. The system is designed to enhance the efficiency, accuracy, and transparency of courier operations, ultimately leading to improved customer satisfaction.

### 1.1 Overview

A Courier Management System (CMS) is a comprehensive software solution designed to streamline and optimize the process of managing courier services. It is used by courier companies to track, manage, and deliver packages efficiently, ensuring timely delivery and customer satisfaction. The system automates various aspects of courier operations, such as package tracking, route optimization, and customer information management, making the entire process more efficient and cost-effective..

One of the key features of a Courier Management System is package tracking. With the CMS, customers can track the status and location of their packages in real-time, providing them with peace of mind and a sense of control over their deliveries. This feature also allows courier companies to monitor the progress of packages and identify any potential delays or issues that may arise during transit.

Another important feature of a CMS is customer information management. The system stores and manages customer data, including contact details, delivery preferences, and billing information. This allows courier companies to provide personalized services to their customers and ensures that deliveries are made according to the customer's preferences.

In addition to package tracking and customer information management, a CMS also includes features such as delivery scheduling and route optimization. These features help courier companies plan and manage their delivery routes more effectively, reducing the time and resources required to complete deliveries. By optimizing delivery routes, courier companies can minimize fuel consumption, reduce vehicle wear and tear, and improve overall efficiency.

Overall, a Courier Management System is a valuable tool for courier companies looking to improve their operations and provide better service to their customers. By automating various aspects of courier operations, the system helps companies save time and money, improve delivery times, and enhance customer satisfaction.

### 1.2 Features of Purposed System:

The proposed Courier Management System (CMS) aims to streamline and enhance the operations of courier companies. The system will offer a range of features designed to improve efficiency, accuracy, and customer satisfaction.

These are the important features of the project Courier Management System:

- In computer system of the courier service computation of the rate is easily & quickly done.
- Computer system of the courier service provide fast access.
- Using this computerized system, bill issued procedure becomes fast.
- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the Manifest but we can directly print it, which saves our time.
- It contain better storage capacity.
- Accuracy in work.
- Easy & fast retrieval of information.
- Well designed reports.
- Decrease the load of the person involve in existing manual system.
- Access of any information individually.
- Work becomes very speedy.
- Easy to update information

The system aims to automate key aspects of courier services, such as package tracking, delivery scheduling, and route optimization, to improve the overall delivery process. By providing real-time tracking and management of packages, the CMS enhances transparency and accountability, allowing customers to track their deliveries and receive timely updates. Additionally, the system aims to improve resource utilization and cost-effectiveness by optimizing delivery routes and reducing manual errors. Overall, the main objective of the CMS is to provide a comprehensive solution for managing courier services, ultimately leading to improved customer service and operational excellence.

## CHAPTER 2

### SYSTEM REQUIREMENTS AND SPECIFICATIONS

The Courier Management System (CMS) requires a robust infrastructure to handle its various components and functionalities efficiently. The system should be designed to accommodate a large number of users, manage extensive data sets, and ensure high performance and reliability.

#### 2.1 Hardware Requirements

- Processor Pentium IV or higher version.
- Ram 128 MB or above
- Hard Disk 150 MB or above

#### 2.2 Software Requirements

- Apache Server 2.0
- PHP Version 5.3 or above
- MySQL Version 5.5 or above
- Latest browser: Chrome, Firefox, Safari etc
- Operating System: Any (Linux, Windows, Mac etc)

#### 2.3 Functional Requirements

A Courier Management System (CMS) is a software application designed to streamline and automate the process of managing courier services. The functional requirements for a CMS are essential to ensure that the system meets the needs of courier companies and their customers. These requirements encompass a wide range of features and capabilities that enable the system to effectively manage various aspects of courier operations.

One of the key functional requirements for a CMS is package tracking. The system should allow customers to track the status and location of their packages in real-time. This feature is essential for providing customers with visibility into the delivery process and ensuring that packages are delivered on time.

Customer management is another important functional requirement for a CMS. The system should support the management of customer information, including contact details, delivery preferences, and billing information. This feature allows courier companies to provide personalized services to their customers and ensures that deliveries are made according to the customer's preferences.

Delivery scheduling is a critical functional requirement for a CMS. The system should allow customers to schedule deliveries at their convenience. This feature helps optimize delivery routes and ensures that packages are delivered in a timely manner. The CMS should also provide users with the ability to select a delivery date and time and specify any special instructions.

Route optimization is another essential functional requirement for a CMS. The system should optimize delivery routes based on factors such as package size, weight, and delivery location. This feature helps minimize delivery times and reduce fuel consumption, ultimately leading to cost savings for courier companies.

## 2.4 Non-Functional Requirements

Non-functional requirements for a Courier Management System (CMS) typically include:

### 1. Performance:

- The system should handle a large number of transactions concurrently.
- Response times for critical operations should be within acceptable limits, even under heavy loads.

### 2. Reliability:

- The system should be reliable, ensuring data integrity and consistency.
- It should have a backup and recovery mechanism to prevent data loss.

### **3. Scalability:**

- The system should be scalable to accommodate increasing data and user loads.
- It should support easy integration with other systems or modules.

### **4. Security:**

- The system should ensure the confidentiality, integrity, and availability of data.
- It should have authentication and authorization mechanisms to control access to the system.

### **5. Usability:**

- The system should be user-friendly, with an intuitive interface.
- It should support multiple languages and be accessible to users with disabilities.

### **6. Compatibility:**

- The system should be compatible with different operating systems and browsers.
- It should also be compatible with different devices, such as desktops, laptops, and mobile devices.

### **7. Maintainability:**

- The system should be easy to maintain and upgrade.
- It should have clear documentation and well-structured code.

### **8. Availability:**

- The system should be available 24/7, with minimal downtime for maintenance or upgrades.
- It should have mechanisms in place for monitoring and alerting in case of failures.

### **9. Legal and Regulatory Requirements:**

- The system should comply with relevant laws and regulations, such as data protection laws.

These requirements are crucial for ensuring the overall success and effectiveness of a Courier Management System.

## CHAPTER 3

### DESIGN

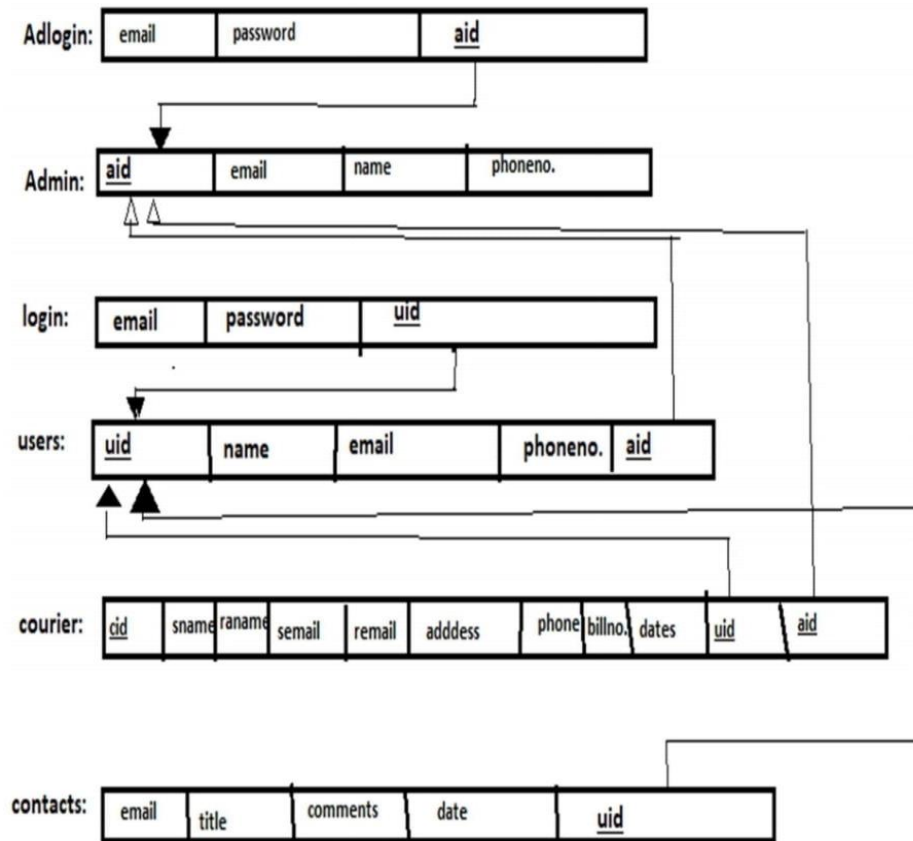
A Courier Management System (CMS) is a comprehensive software solution designed to streamline and optimize the process of managing courier services. At its core, the CMS facilitates the efficient handling of parcels and packages from the point of acceptance to final delivery. The system typically includes modules for order entry, dispatching, tracking, and reporting, providing users with real-time visibility into the status of shipments. Key features of a CMS may include automated routing and scheduling, barcode scanning for tracking, electronic proof of delivery (ePOD), and integration with other systems such as accounting or CRM software. By centralizing and automating these processes, a CMS can help courier companies improve operational efficiency, reduce errors, and enhance customer satisfaction.

#### 3.1 ER Schema Diagram

The schema diagram for a Courier Management System (CMS) represents the structure of the database that stores information related to the management of courier services. The diagram consists of several entities, each representing a different aspect of the system.

The relationships between these entities are crucial for the functioning of the CMS. For example, a customer can have multiple deliveries with different couriers, and each delivery can consist of multiple parcels. Additionally, each parcel is associated with a single delivery, and each delivery occurs between two locations (pickup and delivery).

This schema diagram serves as a blueprint for the database design of the CMS, ensuring that the system can efficiently store and manage the data necessary for the effective operation of courier services.

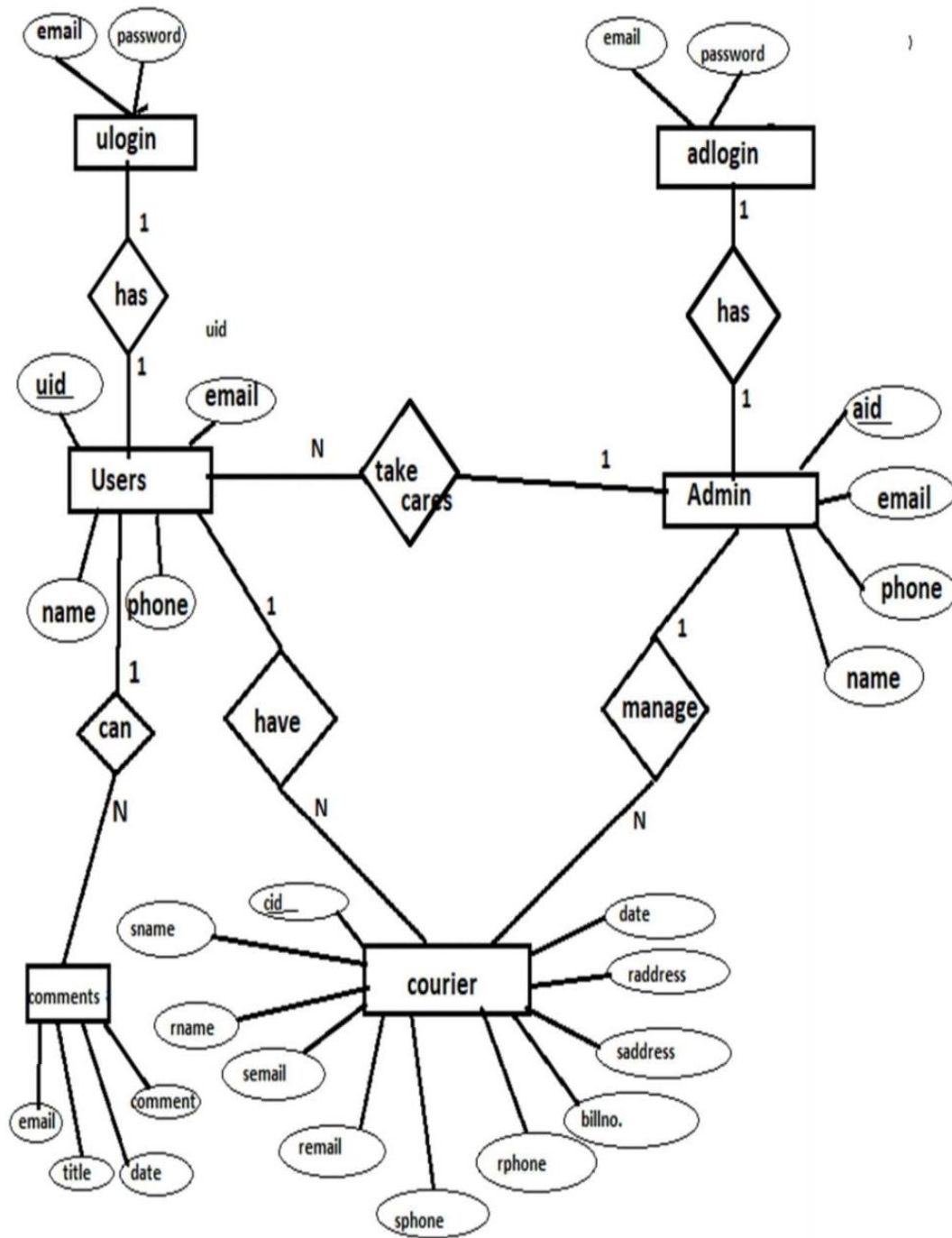


**Fig.3.1: Describes the ER Diagram Of The Courier Management System**

## 3.2 ER DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.





**Fig 3.2: Describes The Er Diagram of Courier Management System**

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 Steps for Implementation:

**Requirements Gathering:** The first step is to gather requirements from stakeholders, including courier companies, customers, and administrators. This helps define the scope and functionalities of the CMS.

**System Design:** Based on the requirements, a detailed system architecture is designed. This includes database design, user interface design, and integration points with other systems. The design phase also involves creating data models, ER diagrams, and use cases to guide implementation.

**Database Design:** A database schema is designed to store information such as customer details, parcel information, and delivery statuses. Tables, relationships, and constraints are defined using a relational database management system.

**User Interface Design:** User-friendly interfaces are designed for different modules and user roles. Wireframes or mockups are created to visualize the user interface.

**Backend Development:** The backend logic is implemented using a suitable programming language. Modules for order entry, dispatching, tracking, and reporting are developed to ensure core functionalities are met.

**Database Implementation:** The database tables are created and populated with sample data for testing. Database operations, such as Create, Read, Update, and Delete (CRUD) operations, are implemented in the backend code.

**Integration:** The CMS is integrated with other systems or services, such as payment gateways or geolocation services, to ensure seamless data flow between different modules

**Testing:** Unit testing, integration testing, and system testing are conducted to ensure the functionality and reliability of the CMS. Test cases are used to verify that the system meets the requirements.

**Deployment:** The CMS is deployed to a production environment, such as a web server or cloud platform. The system is configured for scalability, security, and performance.

**Maintenance and Support:** Ongoing maintenance and support are provided for the CMS. The system is monitored for issues, and updates and enhancements are implemented as needed.

### 4.2 CODE SNIPPET:

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///courier_management.db'
db = SQLAlchemy(app)
```

```
class Customer(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50), nullable=False)
    address = db.Column(db.String(100), nullable=False)
    contact_number = db.Column(db.String(15), nullable=False):
```

```
class Parcel(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    tracking_number = db.Column(db.String(20), nullable=False)
    weight = db.Column(db.Float, nullable=False)
    customer_id = db.Column(db.Integer, db.ForeignKey('customer.id'), nullable=False)
```

```
customer = db.relationship('Customer', backref=db.backref('parcels', lazy=True))

class Delivery(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    status = db.Column(db.String(20), nullable=False, default='Pending')
    courier = db.Column(db.String(50), nullable=False)
    parcel_id = db.Column(db.Integer, db.ForeignKey('parcel.id'), nullable=False)
    parcel = db.relationship('Parcel', backref=db.backref('deliveries', lazy=True))

@app.route('/customers', methods=['POST'])
def add_customer():
    data = request.json
    customer = Customer(name=data['name'], address=data['address'],
contact_number=data['contact_number'])
    db.session.add(customer)
    db.session.commit()
    return jsonify({'message': 'Customer added successfully'}), 201

@app.route('/parcels', methods=['POST'])
def add_parcel():
    data = request.json
    parcel = Parcel(tracking_number=data['tracking_number'], weight=data['weight'],
customer_id=data['customer_id'])
    db.session.add(parcel)
    db.session.commit()
    return jsonify({'message': 'Parcel added successfully'}), 201

@app.route('/deliveries', methods=['POST'])
def add_delivery():
    data = request.json
    delivery = Delivery(status=data['status'], courier=data['courier'], parcel_id=data['parcel_id'])
    db.session.add(delivery)
    return jsonify({'message': 'Delivery added successfully'}), 201

if __name__ == '__main__':
    db.create_all( app.run(debug=True))
```

---

## CHAPTER 5

### RESULTS



**Fig 5.1: Describes the home page of Courier management system**

The image displays the login and sign-up interface for the RL Courier Service. At the top, the text "RL COURIER SERVICE" is written in green, followed by the tagline "The Fastest Courier Service Ever" in orange. On the right side, there is a link for "AdminLogin". The main section is titled "Login" in blue. Below this title, a red message says "Please Fill Your !!". There are two input fields: "Email Address" with the placeholder text "Enter username/emailid" and "Password" with the placeholder text "Enter your password". Below these fields are two buttons: a blue "SignIn" button and a red "Reset Password" button. At the bottom, there is a link that says "Don't have an account? >>> Register here."

**Fig 5.2: Sign up and login in the Courier Management System**



The screenshot shows the 'Price' page of the Courier Management System. At the top, there is a navigation bar with a logo and links: Home, Price, Courier, Track, Profile, ContactUs, AdminPage, and LogOut. The main content area features a table with two columns: 'Weight in Kg' and 'Price'. The table lists weight ranges and their corresponding prices. Below the table, a message states: 'As per your courier's weight pay the amount on:'. This is followed by three payment options: 1. UPI: likhith@sbi.com, 2. GooglePay: 9901889587, and 3. PhonePay: 9553734624. At the bottom, a teal footer contains the copyright notice: 'Copyright © 2024 Designed By Likhith & Rajesh' and the contact information: 'ContactUs 9901889587, 9553734624'.

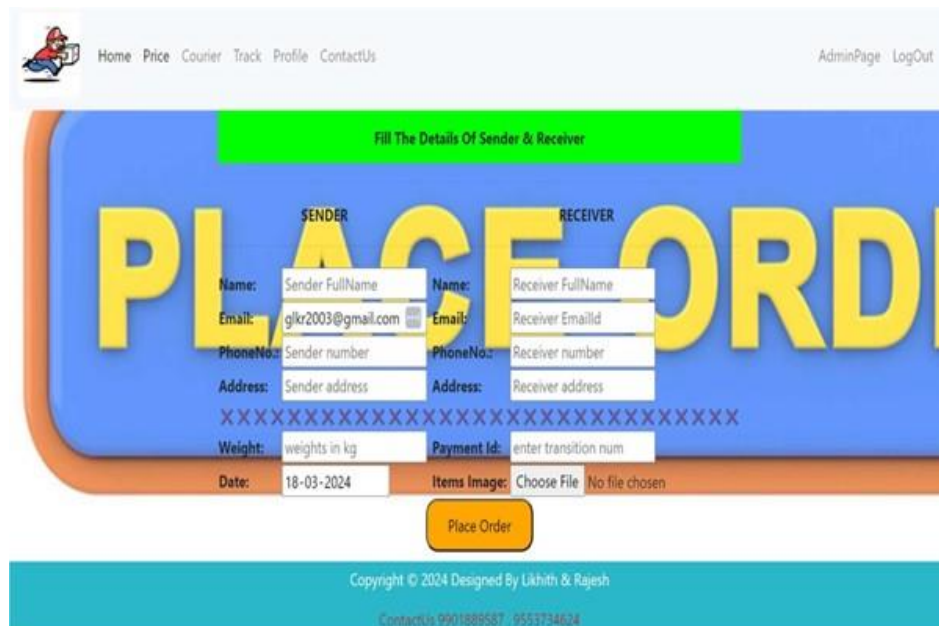
Weight in Kg	Price
0-1	120
1-2	200
2-4	250
4-5	300
5-7	400
7-above	500

As per your courier's weight pay the amount on:

1. UPI: likhith@sbi.com
2. GooglePay: 9901889587
3. PhonePay: 9553734624

Copyright © 2024 Designed By Likhith & Rajesh  
ContactUs 9901889587, 9553734624

**Fig 5.3: Describes the Pricing Of Courier**



The screenshot shows the 'Place Order' page of the Courier Management System. The page has a large blue background with the text 'PLACE ORDER' in yellow. At the top, there is a navigation bar with a logo and links: Home, Price, Courier, Track, Profile, ContactUs, AdminPage, and LogOut. Below the navigation bar, there is a green banner that says 'Fill The Details Of Sender & Receiver'. The form is divided into two columns: 'SENDER' and 'RECEIVER'. Each column has input fields for Name, Email, PhoneNo, Address, Weight, and Date. There is also a 'Payment Id' field and an 'Items Images' section with a 'Choose File' button and a 'No file chosen' message. A large orange 'Place Order' button is at the bottom. At the bottom of the page, a teal footer contains the copyright notice: 'Copyright © 2024 Designed By Likhith & Rajesh' and the contact information: 'ContactUs 9901889587, 9553734624'.

Fill The Details Of Sender & Receiver

**PLACE ORDER**

**SENDER**

Name: Sender FullName  
Email: gikr2003@gmail.com  
PhoneNo: Sender number  
Address: Sender address  
Weight: weights in kg  
Date: 18-03-2024

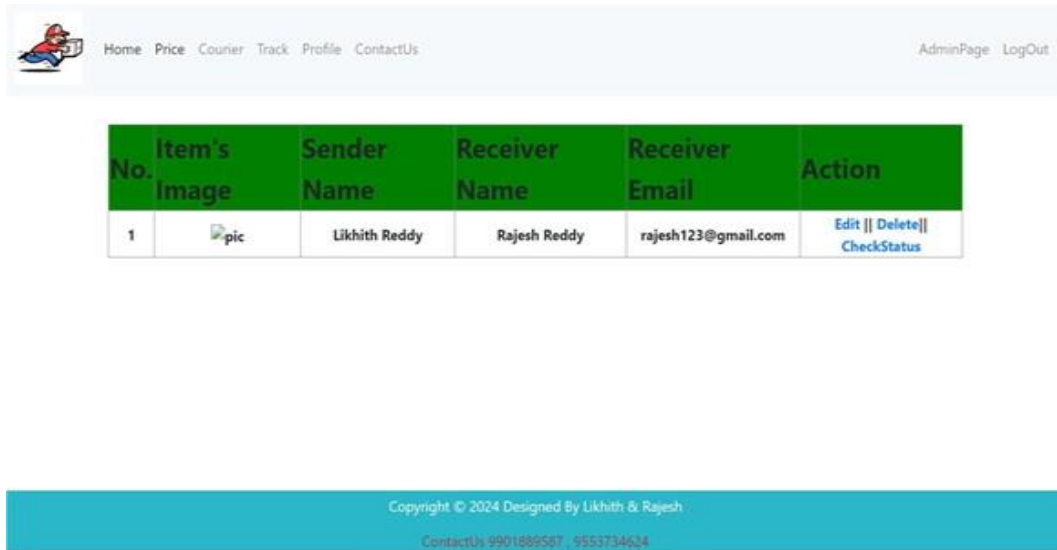
**RECEIVER**

Name: Receiver FullName  
Email: Receiver EmailId  
PhoneNo: Receiver number  
Address: Receiver address  
Payment Id: enter transition num  
Items Images: Choose File No file chosen

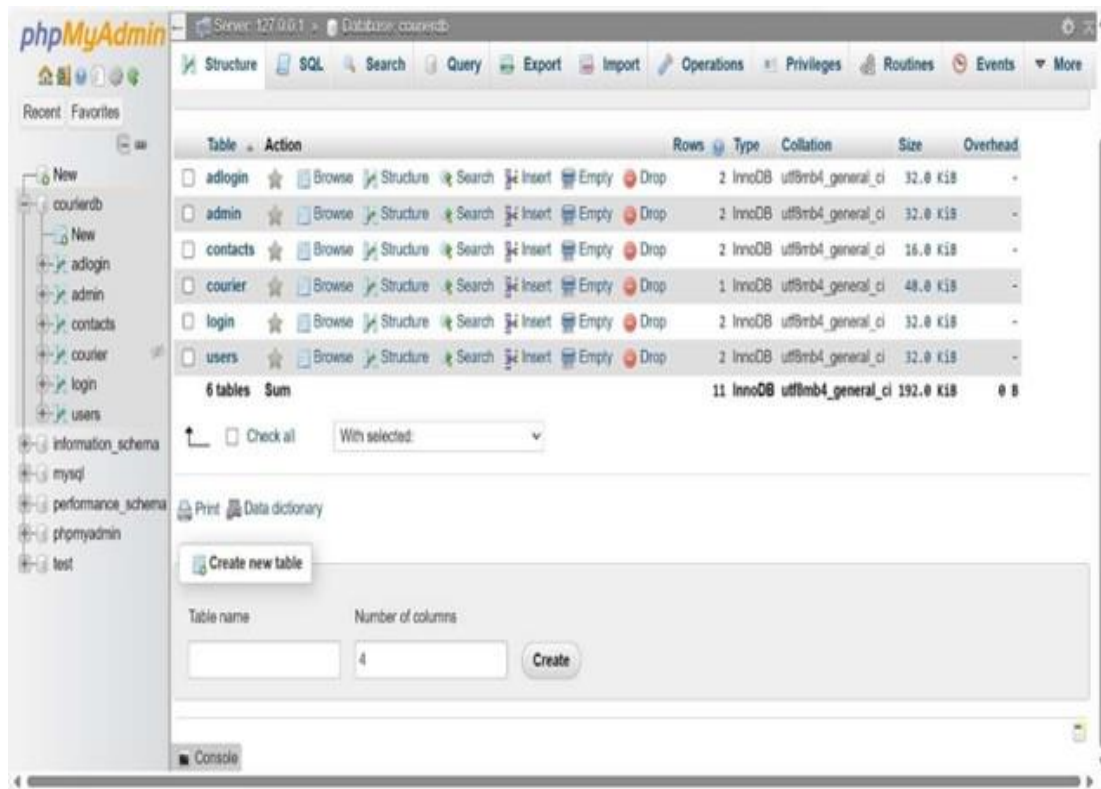
Place Order

Copyright © 2024 Designed By Likhith & Rajesh  
ContactUs 9901889587, 9553734624

**Fig 5.4: Describes The Courier sending page**



**Fig 5.5: Describes The courier track consignment page**



**Fig 5.6 LIST OF ALL THE TABLES IN PROJECT**

## CONCLUSION & FUTURE ENHANCEMENT

### Conclusion

A conclusion for a Courier Management System (CMS) report should summarize the key points discussed in the report and provide insights into the significance of the CMS in the courier industry. Here's a sample conclusion:

In conclusion, the Courier Management System (CMS) is a crucial tool for streamlining and optimizing courier services in the modern digital era. The system offers a wide range of functionalities that facilitate efficient order management, tracking, and delivery processes. Throughout this report, we have discussed the various components and features of a CMS, including customer management, parcel tracking, and delivery scheduling.

One of the key benefits of a CMS is its ability to improve operational efficiency and reduce costs for courier companies. By automating manual processes and providing real-time tracking capabilities, the CMS enables companies to manage their resources more effectively and deliver parcels in a timely manner. Additionally, the system's integration with external services such as mapping and payment gateways enhances its functionality and provides a seamless user experience.

Furthermore, the CMS plays a significant role in enhancing customer satisfaction. With features like online booking, parcel tracking, and electronic proof of delivery (ePOD), customers can easily track their parcels and stay informed about their delivery status. This transparency and convenience contribute to a positive customer experience and help build trust and loyalty towards the courier company. In conclusion, the Courier Management System is a fundamental tool for modern courier companies looking to improve their operational efficiency, reduce costs, and enhance customer satisfaction. With its wide range of functionalities and potential for future advancements, the CMS is set to play an increasingly important role in the courier industry in the years to come.



## REFERENCES

### 1. Role Based Access Control (RBAC):

NIST Special Publication 80053: Security and Privacy Controls for Federal Information Systems and Organizations Provides detailed guidance on implementing RBAC in information systems. [Link](<https://csrc.nist.gov/publications/detail/sp/80053/rev5/final>)

### 2. Reporting and Data Analysis:

"Python Data Analysis" by Ivan Idris Offers a comprehensive guide to data analysis using Python, including techniques for generating reports and visualizations.

### 3. Mobile Application Development:

"Building Mobile Applications with Python" by Michael Lanham Explores various frameworks and tools for building mobile apps using Python, including Flask extensions for mobile development.

### 4. Learning Management Systems (LMS):

"Learning Management Systems and Instructional Design: Best Practices in Online Education" by Yefim Kats Discusses best practices for integrating LMS platforms with external systems, including student management systems.

### 5. Machine Learning for Predictive Analysis:

"HandsOn Machine Learning with ScikitLearn, Keras, and TensorFlow" by Aurélien Géron Offers practical examples and tutorials on implementing machine learning algorithms for predictive analysis.

## BIBLIOGRAPHY

### REFERENCE BOOKS:

1. "Flask Web Development: Developing Web Applications with Python" by Miguel Grinberg
2. "Flask by Example" by Gareth Dwyer
3. "Flask Framework Cookbook" by Shalabh Aggarwal
4. "Essential SQLAlchemy" by Jason Myers and Rick Copeland
5. "Flask Blueprints" by Joel Perrais
6. "Flask Web Development with Python Cookbook" by Talha Khan

### WEBSITES REFERRED:

- <http://www.bluedart.com/>
- <http://www.xamppserver.com/en/>
- <http://www.php.net/> <http://youtube.com/>
- <http://www.tutorialspoint.com/mysql/> <https://apache.org/docs/2.0/misc/tutorials.htm>