

AN INDUSTRIAL ORIENTED MINI PROJECT REPORT ON  
**COURIER MANAGEMENT SYSTEM**  
(A WEB-BASED APPLICATION TO TRACK AND MANAGE COURIERS)

Submitted in partial fulfillment of the requirement for the award of the degree of

**Bachelor of Technology**

**In**

**Information Technology**

by

SHAIK LATEEF	21W91A1251
VADLA PRAVALEKA	21W91A1260
SHIVUKUMAR	21W91A1254
ESHA HARIDAS	21W91A1225
SAI SIDHARTHA ALETI	21W91A1202

**Under the Guidance of**

**MR. K. MAHENDAR**

Assistant professor

**Department of Information Technology**

**MRIET (Autonomous, UGC Govt. of India)**



**DEPARTMENT OF IT**

**MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(UGC AUTONOMOUS)**

(Sponsored by Malla Reddy Educational society)

(Affiliated to JNTU, Hyderabad)

Maisammaguda, Dhulapally post, Secunderabad-500014.

**2023-2024**

## DECLARATION

We hereby declare that the project entitled “ **Courier Management System: A Web-Based Application to Track and Manage Couriers** ” submitted to Malla Reddy Institute of Engineering and Technology, UGC Autonomous Institution, affiliated to Jawaharlal Nehru Technological University Hyderabad(JNTUH) as part of III Year B. Tech-II Semester and for the partial fulfillment of the requirement for the award of Bachelor of Technology in Information Technology is a result of original research work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

**SHAIK LATEEF                      21W91A1251**

**VADLA PRAVALEKA            21W91A1260**

**SHIVUKUMAR                   21W91A1254**

**ESHA HARIDAS                 21W91A1225**

**SAI SIDHARTHA ALETI        21W91A1202**



**MALLA REDDY**  
**INSTITUTE OF ENGINEERING & TECHNOLOGY**  
(Autonomous Institution - UGC, Govt. of India )  
(Sponsored by Malla Reddy Educational Society)  
Approved by AICTE, New Delhi, Recognized Under 2(f) & 12(B)  
Affiliated to JNTU, Hyderabad, Accredited by NBA & NAAC with 'A' Grade



## CERTIFICATE

This is to certify that this is the Bonafide certificate of an Mini Project report titled **“Courier Management System: A Web-Based Application to Track and Manage Couriers”** is submitted by **SHAIK LATEEF (21W91A1251), VADLA PRAVALEKA (21W91A1260), SHIVUKUMAR (21W91A1254), ESHA HARIDAS (21W91A1225), SAI SIDHARTHA ALETI (21W91A1202)** of B. Tech in the partial fulfillment of the requirements for the degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY**, Department of Information Technology and this has not been submitted for the award of any other institute.

**Project Guide**

**MR. K. MAHENDAR**

**Head of the Department**

**Dr. MR. V. RAMAKRISHNAN**

**EXTERNAL EXAMINER**

**Date of Viva-Voice Examination held on: \_\_\_\_\_**

## ACKNOWLEDGEMENT

First and foremost, we are grateful to the Principal **Dr. P. SRINIVAS**, for providing us with all the resources in the college to make our project a success. We thank him for his valuable suggestions at the time of seminars which encouraged us to give our best in the project.

We would like to express our gratitude to **Dr. V. RAMAKRISHNAN**, Head of the Department, Department of Information Technology for his support and valuable suggestions during the dissertation work.

We offer our sincere gratitude to our project -coordinator **Dr. V. RAMAKRISHNAN** and Project guide **MR. K. MAHENDAR** Assistant Professor of Information Technology department who has supported us throughout this project with their patience and valuable suggestions.

We would also like to thank all the supporting staff of the Department of Information Technology and all other departments who have been helpful directly or indirectly in making the project a success.

We are extremely grateful to our parents for their blessings and prayers for our completion of project that gave us strength to do our project.

**SHAIK LATEEF**                      **21W91A1251**

**VADLA PRAVALEKA**            **21W91A1260**

**SHIVUKUMAR**                   **21W91A1254**

**ESHA HARIDAS**                 **21W91A1225**

**SAI SIDHARTHA ALETI**   **21W91A1202**

## **ABSTRACT**

The Courier Management System: A Web-Based Application to Track and Manage Couriers is a robust, web-based platform designed to streamline courier tracking and management processes for both administrators and customers. The system offers real-time tracking of shipments, enabling administrators to monitor the status of deliveries, manage feedback, and ensure efficient coordination. For customers, the system provides an intuitive interface to track their couriers using unique tracking IDs, ensuring timely updates and a seamless user experience. The platform's main goal is to simplify courier management tasks while enhancing operational transparency and customer satisfaction. Developed using Python and Django for the backend and HTML, CSS, and JavaScript for the frontend, the Courier Management System prioritizes both functionality and user experience. The responsive design ensures compatibility across devices, providing accessibility for users at any time. With its focus on accuracy and ease of use, the CMS is a reliable tool for logistics and courier companies, offering a comprehensive solution for tracking and managing deliveries while catering to the growing demand for efficient, real-time courier services.

## TABLE OF CONTENTS

Chapter No.		Title	Page No.
1		Introduction	1-6
	1.1	Introduction	1-3
	1.2	Motivation	3-4
	1.3	Literature Survey	4-5
	1.4	Objective of the Project	6
2		System Analysis	7-11
	2.1	Existing System and Proposed System	7-10
	2.2	Functional Requirements	11
3		Software Environment for python	12-16
	3.1	Python	12
	3.2	History of Python	12-15
	3.3	Python Features	15-16
4		System Study	17-18
	4.1	Feasibility Study	17-18
5		System Design and Architecture	19-21
	5.1	System Architecture Diagram	19
	5.2	Use Case Diagrams	20
	5.3	Class Diagrams	20-21
	5.4	Sequence Diagrams	21

6		Software Development Life Cycle	22-26
	6.1	Phases of SDLC	22-26
7		Implementation	27-28
	7.1	Sample code	27-28
8		Testing	29-35
	8.1	Testing Methodologies	29-30
	8.2	User Training	30-32
	8.3	Maintenance	32-33
	8.4	Testing Strategy	33-35
9		Output Screens	36-50
	9.1	Screenshots	36-50
10		Conclusion and Future Scope	51-54
	10.1	Conclusion	51
	10.2	Future Scope	52-54
11		References	55-57
	11.1	Publications	55-56
	11.2	Books	56-57
	11.3	Websites	57

## LIST OF FIGURES

<b>Fig. No</b>	<b>Figure Title</b>	<b>Page No</b>
5.1	System Architecture Diagram	19
5.2	Use Case Diagram	20
5.3	Class Diagram	21
5.4	Sequential Diagram	21



## LIST OF SCREENSHOTS

<b>Screenshot No.</b>	<b>Name of the Screenshot</b>	<b>Page No</b>
9.1	Terminal Window	36
9.2	Home Screen	37
9.3	Admin Login Screen	37
9.4	Admin Home Screen	38
9.5	Add New Employee Screen	38
9.6	Employee Details Added	39
9.7	List of Employees	39
9.8	List of all Booked Couriers	40
9.9	Employee Login Screen	40
9.10	Employee Home Screen	41
9.11	Add New Courier Screen	41
9.12	Courier Details Added and Assigned Id	42
9.13	Update Courier Status Screen	43
9.14	Updating Courier Status	43
9.15	Employee View Courier Status Screen	44
9.16	Employee Courier Tracking Screen	45
9.17	Courier Track Screen View Map	45
9.18	User Courier Track Screen	46
9.19	User Courier Tracking in Detail	46
9.20	User Feedback Screen	47
9.21	Admin View Feedback Screen	48

## LIST OF TABLES

Table No.	Name of the Table	Page No
9.22	Employee Table	48
9.23	Parcel Table	49
9.24	Update Status Table	49
9.25	Feedback Table	50

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

In the fast-evolving world of logistics and courier services, it is crucial for businesses to adopt systems that streamline the process of managing and tracking shipments. Traditional manual tracking methods are increasingly becoming inefficient and prone to human errors, which can negatively impact customer satisfaction. The Courier Management System: A Web-Based Application to Track and Manage Couriers is designed to address these challenges by providing a centralized, automated platform for tracking and managing couriers. This system aims to improve operational efficiency, enhance transparency, and offer a seamless user experience for both customers and courier service providers.

The Courier Management System allows customers to book shipments, track their couriers in real-time, and receive timely updates about their parcels' journey. When a customer wishes to send a courier, they visit a logistics shop or courier center that utilizes this system. The employee collects the courier from the customer and enters the shipment details, including sender and recipient information, package dimensions, and weight, into the system. Upon submission, the system generates a unique tracking ID, which is then provided to the customer.

This tracking ID allows customers to monitor the status of their courier as it moves through various stages of the delivery process. The customer can use the unique ID to track the courier's location at any time by entering the ID in the system's user interface. The system, through an embedded map feature, displays the real-time location of the courier on a map using an iframe tag to integrate the map view. This integration provides an intuitive way for customers to visually track their shipments' journey, offering greater transparency and control over their deliveries.

Each time the courier reaches a new checkpoint, the delivery staff manually updates the courier's status in the system. This includes recording information such as when the courier is dispatched, when it reaches a warehouse, or when it is out for delivery. These manual updates ensure that the status is as accurate as possible, despite not relying on automated tracking systems, which are often not available for every shipment. The manual entry of information by

delivery staff at key stages in the process ensures that customers receive the most accurate and timely updates on their shipments.

The Courier Management System is built using Python and Django, which offer a robust and secure backend infrastructure for handling customer data and managing shipments efficiently. The front end is developed using HTML, CSS, and JavaScript, ensuring that the system is responsive and easy to navigate on various devices, such as desktop computers, tablets, and smartphones.

One of the key features of the system is the integration of the iframe tag to display the courier's location on a map. The iframe provides a simple, yet effective, way to embed a live map directly into the user interface, allowing customers to view the location of their courier in real-time. By entering their unique tracking ID into the application, customers can see the updated location of their courier on a map, which adds a layer of convenience and enhances the user experience.

While the system does not currently include automated notifications or alerts, customers are able to track their couriers manually by entering the tracking ID in the system. Future versions of the system may include features such as email or SMS notifications to alert customers when their courier reaches certain milestones, such as dispatch, arrival at a distribution center, or out-for-delivery status.

Additionally, the system provides an admin panel for delivery staff and management. This panel allows employees to update the status of shipments, access customer details securely, and generate performance reports. These features not only streamline the management of couriers but also enable businesses to make data-driven decisions to enhance service quality and operational efficiency.

By automating the courier booking process and providing a simple, yet powerful, tracking system, the Courier Management System enhances the efficiency of courier services, reduces human error, and improves customer satisfaction. Through real-time tracking and clear updates, customers are given the information they need to confidently monitor their shipments from start to finish. The integration of a map display feature further enhances the user experience, providing a visual representation of the courier's journey, thereby increasing customer trust and satisfaction with the service.

The Courier Management System is a scalable and adaptable solution for businesses looking to modernize their courier operations. The system's flexible design allows it to be customized to meet the specific needs of different logistics companies, ensuring that it can handle various types of shipments, delivery processes, and tracking requirements.

### **1.2 MOTIVATION**

The courier and logistics industry has seen significant growth and transformation with the rise of e-commerce and the increasing demand for fast, reliable delivery services. In this competitive landscape, customers expect high levels of transparency, including the ability to track shipments in real time and receive consistent updates at each stage of the delivery process. Traditional courier systems, however, often rely on manual processes, limiting their ability to meet these evolving customer expectations. The Courier Management System: A Web-Based Application to Track and Manage Couriers was developed to address these needs, offering a streamlined, digital platform for enhanced customer experience and operational efficiency. By providing real-time location tracking in an easy-to-use interface, this system allows customers to access accurate, up-to-date information on their couriers, thereby improving transparency and satisfaction.

The primary motivations for implementing the Courier Management System include:

- **Enhanced Customer Experience:** Customers can track their couriers in real time, providing them with greater visibility into the status of their shipments.
- **Operational Efficiency:** Automation of routine booking and tracking tasks reduces manual intervention, saving time and minimizing the risk of errors.
- **Cost Reduction:** Reduced reliance on manual updates lowers operational costs and allows resources to be used more effectively.
- **Data-Driven Decision-Making:** The system's admin panel provides insights and analytics, enabling courier companies to monitor trends, optimize operations, and improve service quality.
- **Flexibility and Scalability:** The system is designed to adapt to various business needs, whether for small courier services or large-scale logistics operations, making it suitable for long-term growth and expansion.

This web-based solution not only aligns with industry trends toward automation and digital integration but also positions courier companies to remain competitive in a rapidly evolving market. By leveraging modern technology for tracking and reporting, the Courier Management System helps businesses meet current demands and prepares them for future innovations, ensuring sustainable growth in the logistics sector.

### 1.3 Literature Review

The field of Courier Management Systems (CMS) has attracted significant research interest due to its potential to enhance logistics efficiency, improve customer satisfaction, and incorporate emerging technologies within the supply chain. Over the years, various studies have explored different aspects of CMS, from traditional manual tracking methods to more advanced solutions that integrate real-time GPS tracking, heuristic optimization algorithms, and customer relationship management (CRM) systems. This literature review synthesizes key contributions in the domain, focusing on advancements such as real-time location tracking for increased transparency, algorithmic route optimization to improve operational efficiency, and the role of IoT and AI technologies in predictive analytics. In addition, the review identifies existing gaps in the research, particularly in the areas of security and scalability, and underscores the potential benefits of incorporating blockchain technology to address these issues. These insights highlight the need for a modernized, multi-technology approach to meet evolving customer expectations and operational demands within the courier logistics sector.

The following studies contribute significantly to the understanding and development of CMS, providing a foundation for the advancement of this field:

1. **Title:** Optimization of Courier Management System Using Real-Time GPS Tracking  
**Author:** Dr. JohnA.Smith  
**Description:** Dr. Smith's research investigates the impact of real-time GPS tracking on courier service efficiency. The study demonstrates how the integration of GPS technology into CMS can optimize route planning, reduce delivery times, and enhance operational transparency. His work is pivotal in showcasing the technological advancements that have revolutionized courier logistics.
2. **Title:** Heuristic Approaches for Route Optimization in Courier Services  
**Author:** Dr. EmilyK.Johnson  
**Description:** Dr. Johnson's research explores various heuristic methods for route

optimization within courier management. By comparing algorithms such as genetic algorithms and simulated annealing, the study provides valuable insights into the most effective strategies for minimizing delivery costs and improving service reliability. Her contributions are essential in developing efficient routing solutions within CMS.

3. **Title:** Customer Relationship Management in Courier Management Systems

**Author:** Dr. Michael R. Lee

**Description:** Dr. Lee examines the integration of customer relationship management (CRM) systems within CMS. The study highlights how CRM practices, such as timely communication, real-time tracking updates, and feedback mechanisms, can significantly improve customer satisfaction and retention. This research provides a comprehensive overview of CRM best practices in the courier industry.

4. **Title:** The Role of IoT and AI in Enhancing Courier Management Systems

**Author:** Dr. Sarah L. Green

**Description:** Dr. Green's research investigates the role of Internet of Things (IoT) and Artificial Intelligence (AI) technologies in CMS. The study demonstrates how these emerging technologies can enhance predictive analytics, automate processes, and improve data-driven decision-making. Her work is essential in understanding future trends and innovations in courier management systems.

5. **Title:** Blockchain for Secure and Transparent Courier Management

**Author:** Dr. David P. White

**Description:** Dr. White's research explores the application of blockchain technology in CMS, focusing on its potential to enhance data security and operational transparency. The study discusses how blockchain can provide a tamper-proof record of transactions, improve accountability, and foster trust among stakeholders. This research is crucial for addressing the security challenges in modern CMS.

These five studies collectively provide a comprehensive overview of the key areas of development in Courier Management Systems, such as GPS tracking, route optimization, CRM integration, IoT and AI applications, and blockchain technology. They highlight the technological advancements that are shaping the future of courier logistics and underscore the importance of adopting integrated solutions to address both current challenges and future demands in the industry.

## 1.4 OBJECTIVE OF THE PROJECT

The primary objective of the project "**Courier Management System: A Web-Based Application to Track and Manage Couriers**" is to create an efficient, scalable, and user-friendly web application that enhances the tracking and management of courier deliveries. The project aims to improve operational efficiency and customer satisfaction by leveraging modern web technologies, addressing the limitations of traditional methods, and ensuring a robust, seamless user experience.

### **Specific Objectives:**

#### **1. Streamlining Courier Tracking:**

- To develop a real-time tracking system that allows customers and administrators to monitor the status and location of their parcels throughout the delivery process.

#### **2. Automating Courier Management:**

- To implement features to efficiently manage courier assignments, orders, and schedules, ensuring effective utilization of resources and better service delivery.

#### **3. Improving Customer Interaction:**

- To design a user-friendly interface for customers to easily request services, track couriers, and provide feedback, improving the overall experience.

#### **4. Enhancing Data Transparency:**

- To provide detailed logs of courier activities, such as pickup and delivery times, to ensure transparency and accountability for both customers and management.

#### **5. Security and Privacy:**

- To incorporate secure user authentication and data encryption to protect sensitive customer information and ensure compliance with data protection regulations.

#### **6. Scalability and Flexibility:**

- To build a system that is scalable and capable of handling large amounts of data, allowing for future expansion as the business grows.



## CHAPTER 2

### SYSTEM ANALYSIS

#### 2.1 EXISTING SYSTEM AND PROPOSED SYSTEM

##### Existing System:

The existing systems in the courier management industry predominantly operate on a mix of manual processes and semi-automated solutions. Traditional courier operations often rely on paper-based methods for order processing, dispatching, and tracking, leading to inefficiencies and delays. Semi-automated systems, while incorporating some level of digitalization, usually lack comprehensive integration, resulting in fragmented workflows and data silos. These systems typically use basic GPS tracking and simple route planning algorithms, which may not fully optimize delivery routes or provide real-time updates. Customer interactions are often managed through phone calls and emails, with limited automated notifications and tracking capabilities. Additionally, reporting and analytics are generally underdeveloped, hindering the ability to make data-driven decisions and improve operational performance. Overall, the existing systems struggle to meet the demands of modern logistics, where speed, accuracy, and transparency are paramount. The need for a more advanced, fully integrated Courier Management System that leverages cutting-edge technologies is evident to address these shortcomings and enhance overall efficiency and customer satisfaction.

##### Disadvantages

1. **Inefficiency and Time-Consumption:** Manual processes such as paper-based order processing and tracking are time-consuming and prone to human error. This inefficiency leads to delays in order handling and delivery.
2. **Fragmented Workflows:** Semi-automated systems usually lack full integration across different operational areas, resulting in fragmented workflows. This fragmentation creates data silos, complicates communication between departments, and hinders overall operational cohesion.
3. **Limited Route Optimization:** Basic GPS tracking and simple route planning algorithms used in existing systems do not fully optimize delivery routes. This

limitation results in longer delivery times, increased fuel consumption, and higher operational costs.

4. **Poor Real-Time Tracking and Updates:** Many existing systems provide inadequate real-time tracking and status updates. This lack of transparency can lead to customer dissatisfaction due to uncertainty about delivery status and potential delays.
5. **Manual Customer Interaction:** Customer interactions are often managed manually through phone calls and emails, leading to inefficiencies and delays in communication. Limited automated notifications and tracking capabilities fail to meet modern customer expectations for instant information.
6. **Inadequate Reporting and Analytics:** Existing systems often have underdeveloped reporting and analytics features. This deficiency hampers the ability to generate actionable insights, make data-driven decisions, and identify areas for improvement.
7. **Error-Prone Operations:** Manual data entry and handling are susceptible to errors, which can result in incorrect shipments, misplaced packages, and other logistical issues. These errors can significantly impact customer satisfaction and operational efficiency.
8. **Scalability Issues:** Many traditional systems are not designed to scale effectively with growing business demands. As the volume of deliveries increases, the limitations of these systems become more pronounced, leading to further inefficiencies and challenges in managing larger operations.
9. **Security Concerns:** Semi-automated systems may not provide robust security measures, making them vulnerable to data breaches and unauthorized access. This lack of security can compromise sensitive customer information and operational data.
10. **High Operational Costs:** Inefficiencies and lack of optimization contribute to higher operational costs, including increased labor, fuel, and administrative expenses. These costs can erode profit margins and affect overall business sustainability.

## Proposed System

The proposed Courier Management System (CMS) aims to revolutionize the courier and delivery industry by integrating advanced technologies and comprehensive features to address the limitations of existing systems. This system will fully automate order processing, shipment tracking, and delivery scheduling, ensuring a seamless and efficient workflow. By incorporating real-time GPS tracking, the proposed CMS will provide precise and dynamic tracking, significantly reducing delivery times and operational costs. Customers will benefit from automated notifications and real-time tracking updates, enhancing transparency and satisfaction. Additionally, electronic proof of delivery will ensure accuracy and accountability. The proposed CMS will leverage IoT and AI technologies to predict delivery times, manage fleet maintenance proactively, and optimize resource allocation. Enhanced security features, such as blockchain, will ensure data integrity and protect against unauthorized access. Comprehensive reporting and analytics tools will enable businesses to monitor performance, identify trends, and make informed decisions. Scalable and flexible, the proposed system will be capable of growing with business demands, maintaining high efficiency and service quality even as the volume of deliveries increases. This holistic approach will transform courier operations, driving operational excellence and superior customer service.

## Advantages

1. **Enhanced Efficiency and Automation:** The proposed system automates critical processes such as order processing, shipment tracking, and delivery scheduling, significantly reducing manual effort and minimizing human error.
2. **Improved Customer Experience:** Automated notifications and real-time tracking updates provide customers with timely and accurate information about their shipments, increasing transparency and satisfaction.
3. **Accurate and Accountable Deliveries:** Electronic proof of delivery ensures that every package is tracked and confirmed upon receipt, reducing disputes and enhancing accountability.
4. **Predictive Analytics:** The integration of IoT and AI technologies allows for predictive maintenance of the fleet, forecasting potential issues before they arise, and ensuring optimal resource allocation.

5. **Enhanced Security:** Utilizing blockchain technology, the system ensures the integrity and security of data, protecting against unauthorized access and tampering.
6. **Comprehensive Reporting and Analytics:** Robust analytics tools provide insights into performance metrics, enabling businesses to monitor operations, identify trends, and make data-driven decisions to improve efficiency.
7. **Scalability and Flexibility:** The proposed system is designed to scale with business growth, maintaining high efficiency and service quality even as the volume of deliveries increases.
8. **Cost Reduction:** By automating processes and reducing errors, the proposed CMS can significantly lower operational costs, including labor, fuel, and administrative expenses.
9. **Enhanced Customer Relationship Management (CRM):** The system's CRM capabilities facilitate better communication with customers, timely feedback collection, and personalized services, leading to higher customer retention and loyalty.
10. **Real-Time Decision Making:** With access to real-time data, managers can make informed decisions quickly, improving responsiveness to changing conditions and customer needs.
11. **Environmental Benefits:** Optimized processes and efficient resource allocation reduce fuel consumption and carbon emissions, contributing to a more sustainable operation.

## **2.2 FUNCTIONAL REQUIREMENTS (HARDWARE AND SOFTWARE)**

### **HARDWARE REQUIREMENT SPECIFICATIONS:**

Processor	: Intel i3 or above (2.4 GHz or higher).
RAM	: 4 GB (minimum).
Hard Disk	: 40 GB (minimum).
Keyboard	: Standard Windows Keyboard.
Mouse	: Two or Three Button Mouse.
Monitor	: SVGA or higher resolution.

### **SOFTWARE REQUIREMENT SPECIFICATIONS:**

Operating System	: Windows 8 or above.
Coding Language	: Python.
Framework	: Django.
Designing Tools	: HTML, CSS, JavaScript.
Database	: MySQL.

## CHAPTER 3

### SOFTWARE ENVIRONMENT FOR PYTHON

#### 3.1 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level.

#### 3.2 History of Python:

Python's development history reflects its evolution from a simple scripting language to one of the most widely used programming languages worldwide, especially popular for web development, data science, machine learning, and automation. Here's an overview of its journey:

##### 1. Early Beginnings (1980s)

- **Conception:** Python was conceived in the late 1980s by Guido van Rossum, a Dutch programmer working at Centrum Wiskunde & Informatica (CWI) in the Netherlands.
- **Inspiration:** Van Rossum aimed to create a new scripting language to address gaps he observed in existing languages like ABC, a language developed at CWI that was easy to use but lacked extensibility. Python was inspired by ABC's simplicity but was designed to be more powerful and extensible.
- **Initial Release:** In December 1989, van Rossum began working on Python as a hobby project over Christmas. The name "Python" was inspired by the British comedy group Monty Python, as van Rossum was a fan of their show, Monty Python's Flying Circus.

## 2. Python 1.x Series (1991–2000)

- **Python 1.0 (1991):** Python 1.0 was officially released in February 1991, with features like exception handling, functions, and the core data types of str, list, dict, and tuple. The first version already emphasized readability and simplicity.
- **Core Concepts:** From the start, Python included key principles such as readability, easy syntax, and extensibility with modules. These were central to the language's philosophy.
- **Growth and Adoption:** Python 1.x introduced important updates, like the addition of modules, libraries, and basic built-in functions. Its simplicity and versatility attracted a small but dedicated user base.

## 3. Python 2.x Series (2000–2010)

- **Python 2.0 (2000):** Released in October 2000, Python 2.0 marked a significant step forward. It introduced features like list comprehensions, garbage collection, and improved support for Unicode, making Python more powerful and efficient.
- **Community Involvement:** Python 2.0 was the first release managed by the newly created Python Software Foundation (PSF), marking a transition to open-source, community-driven development.
- **Significant Additions:** Throughout the Python 2.x series, numerous features were added, including new-style classes (Python 2.2), iterators, and generators. By the end of the 2.x series, Python had grown substantially in both capability and popularity, particularly in fields like web development.

#### 4. Transition to Python 3.x (2008-Present)

- **Python 3.0 (2008):** Released in December 2008, Python 3.0, also known as "Python 3000" or "Py3k," was a major overhaul of the language, intended to fix design flaws and remove redundancies from Python 2.
- **Key Changes:** Python 3 introduced significant changes, such as the `print()` function (instead of the `print` statement), `str` objects becoming Unicode by default, improved integer division, and better handling of text and binary data.
- **Migration Period:** Due to compatibility issues, adoption of Python 3 was initially slow, and Python 2.7 (released in 2010) continued to receive support. A long transition period ensued, with Python 2 support only officially ending on January 1, 2020.

#### 5. Python 3.x Series and Modern Development

- **Python 3.x Advancements:** Since Python 3.4, new releases have been issued consistently, with each version adding modern features to enhance performance, ease of use, and functionality. Some notable features include:
  - **Python 3.4 (2014):** Introduced `asyncio` for asynchronous programming.
  - **Python 3.5 (2015):** Added `async/await` syntax for easier asynchronous programming.
  - **Python 3.6 (2016):** Brought f-strings, providing a more convenient way to format strings.
  - **Python 3.7 (2018):** Introduced data classes, which simplify class definition for storing data.
  - **Python 3.8 (2019):** Added the "walrus operator" (`:=`) for assignment expressions.
  - **Python 3.9 (2020):** Improved the parser, added new string functions, and dictionary merge operators.
  - **Python 3.10 (2021):** Introduced structural pattern matching and improved error messages.



- **Python 3.11 (2022):** Focused on performance enhancements and introduced more efficient error handling.
- **Widespread Adoption:** With each new release, Python 3's features have driven adoption in diverse areas such as web development, data science, machine learning, artificial intelligence, automation, and more.

## 6. Python's Impact and Future

- **Popularity and Community:** Python's simplicity, versatility, and supportive community have propelled it to be one of the most popular programming languages worldwide. It's consistently rated highly in developer surveys and is known as a beginner-friendly language with extensive libraries for varied applications.
- **Ongoing Development:** Python continues to evolve under the guidance of the Python Software Foundation (PSF) and a large open-source community. Emphasis is on improving speed, expanding library support, and refining language features to maintain Python's relevance in a rapidly changing tech landscape.
- **Legacy:** Python's influence is seen in many fields, especially data science and AI, where libraries like Pandas, NumPy, TensorFlow, and PyTorch are foundational. Its impact on education is also notable, as it's frequently used as a teaching language in programming courses due to its accessibility.

Python's history reflects its transformation from a small project into a foundational tool for developers and data scientists globally. With its user-friendly syntax, a strong open-source community, and continual improvements, Python is positioned to remain relevant and influential well into the future.

## 3.3 Python Features

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax.
- This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

- A broad standard library: Python's bulk of the library is very portable and cross-platform
- compatible on UNIX, Windows, and Macintosh.
- Interactive Mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases: Python provides interfaces to all major commercial databases.
- GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable: Python provides a better structure and support for large programs than shell
- scripting.

Python has a big list of good features:

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

## **CHAPTER 4**

### **SYSTEM STUDY**

#### **4.1 FEASIBILITY STUDY**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

#### **ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

#### **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

## **SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

SYSTEM DESIGN AND ARCHITECTURE

5.1 SYSTEM ARCHITECTURE DIAGRAM

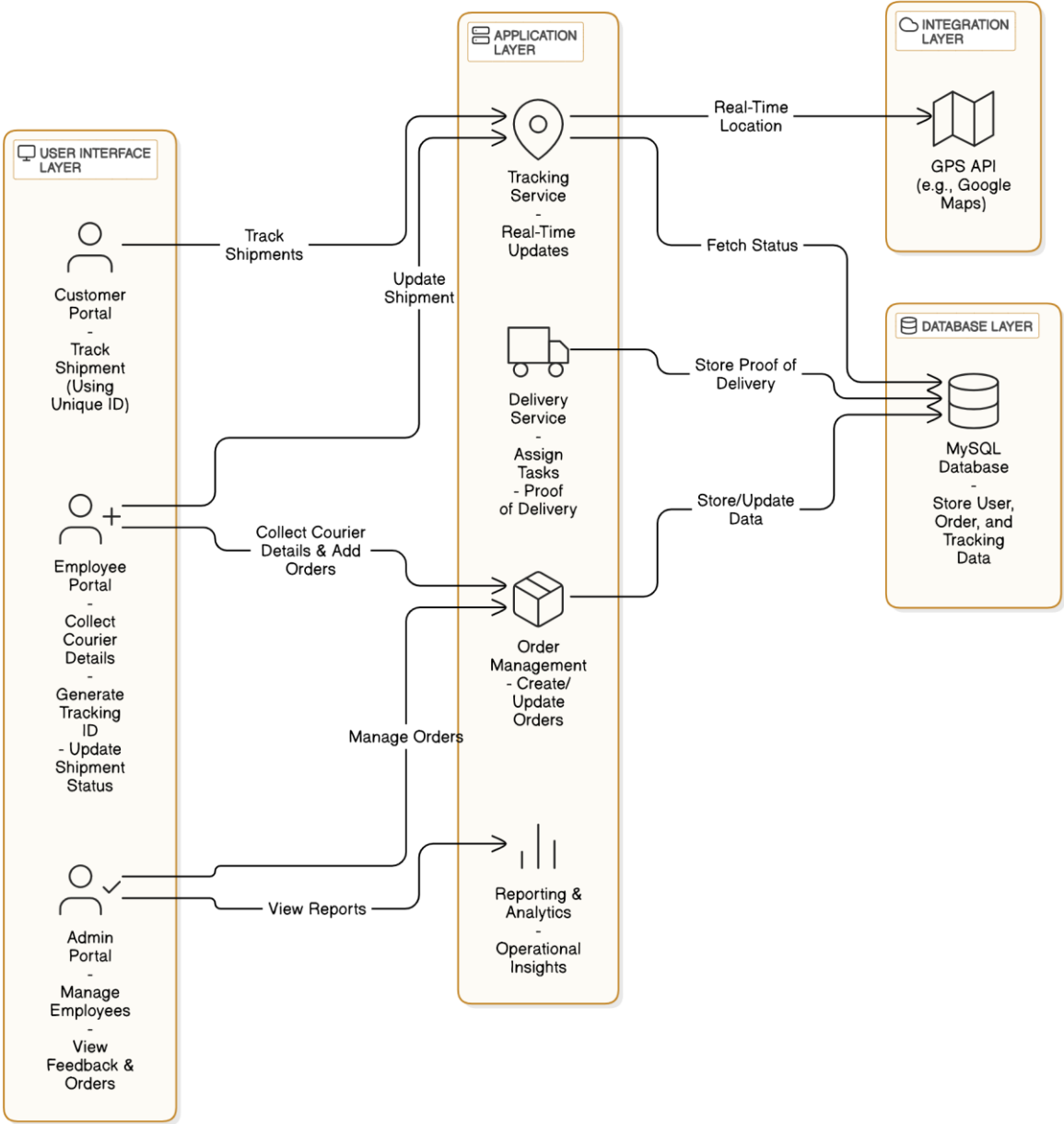


Fig. 5.1 System Architecture Diagram

## 5.2 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

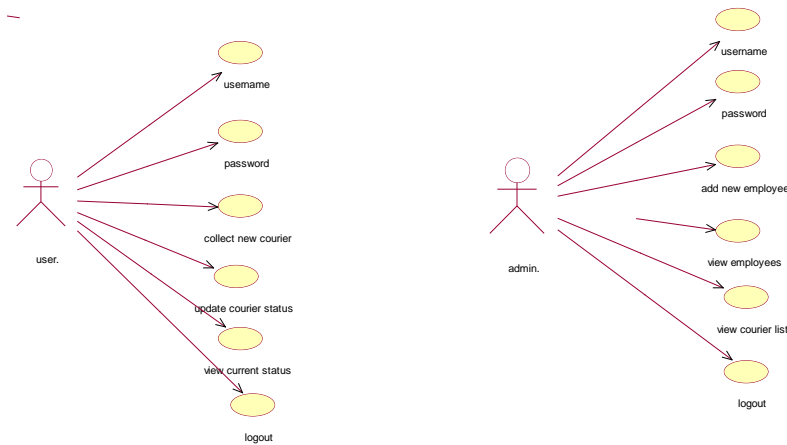


Fig. 5.2 Use Case Diagram

## 5.3 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

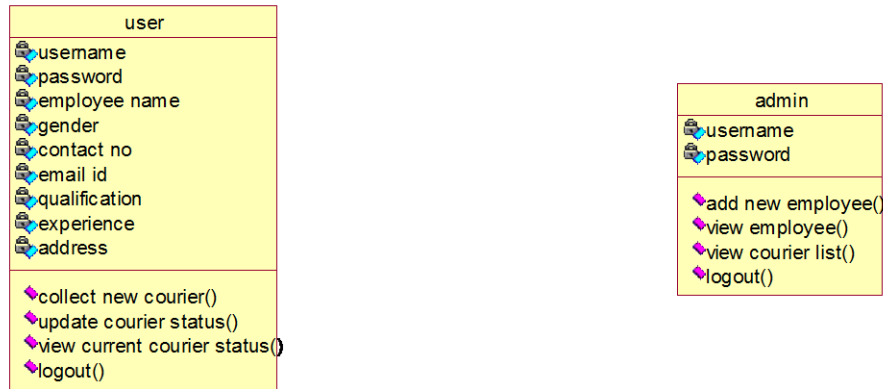


Fig. 5.3 Class Diagram

## 5.4 SEQUENCE DIAGRAM:

A diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

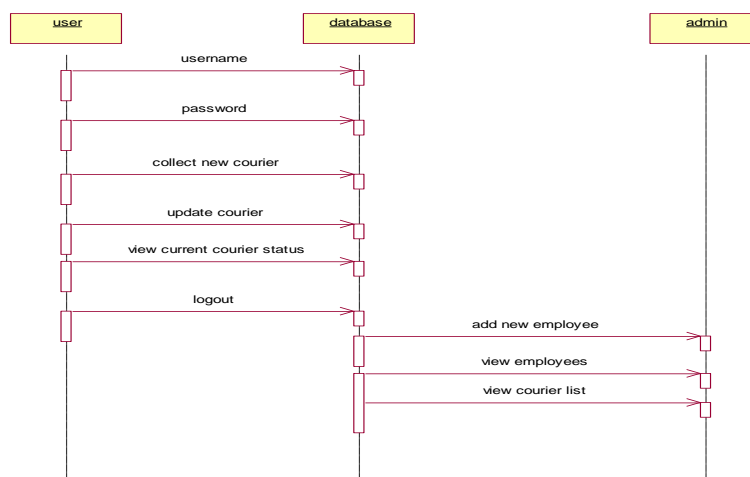


Fig. 5.4 Sequence Diagram

## CHAPTER 6

# SOFTWARE DEVELOPMENT LIFE CYCLE

### 6.1 PHASES OF SDLC

#### 1. Planning and Requirement Analysis

- **Objective:** Define the project goals, determine deliverables, and gather detailed requirements for the Courier Management System.
- **Activities:**
  - Identify stakeholders: Logistics team, delivery personnel, customers, and admin users.
  - Gather and define functional requirements:
    - User registration, authentication, and authorization.
    - Courier assignment, tracking, and status updates.
    - Admin dashboard for managing couriers, viewing customer feedback, and tracking system performance.
    - Customers can only track couriers using an ID (without accessing other sensitive features).
  - Gather and define non-functional requirements:
    - System scalability to handle an increasing number of users.
    - Real-time data updates for package tracking and status changes.
    - Security and data privacy measures for user information.
  - Establish technical feasibility:
    - Evaluate backend and frontend technologies (Python/Django, MySQL, HTML/CSS/JavaScript).



## 2. System Design

- **Objective:** Design the overall system architecture and structure to meet functional and non-functional requirements.
- **Activities:**
  - Design the database schema:
    - Tables for users, couriers, orders, transactions, tracking information, and customer feedback.
  - Design the system architecture:
    - Web-based architecture with Django backend, MySQL database, and frontend using HTML, CSS, JavaScript.
    - User authentication using Django's built-in authentication system.
    - Real-time tracking system for updating courier statuses and locations.
    - Admin dashboard to manage couriers, view customer feedback, and track system performance.
  - Plan the flow of data in the system:
    - User registration → Courier order creation → Courier assignment → Tracking and status updates → Feedback submission → Admin review of feedback.
  - Design user interfaces (UI):
    - Admin dashboard for managing couriers, viewing feedback, and tracking system performance.
    - Customer-facing interface for tracking couriers using an ID.

## 3. Implementation

- **Objective:** Build the Courier Management System based on the design and requirements.
- **Activities:**
  - **Backend Development:**
    - Set up Django environment and configure MySQL database.
    - Implement user registration, login, and profile management.

## Courier Management System: A Web-Based Application to Track and Manage Couriers

- Develop the courier assignment system, where couriers are assigned to specific orders.
- Implement courier tracking system using real-time status updates.
- Implement feedback collection from customers after order delivery.
- **Frontend Development:**
  - Create responsive web pages using HTML, CSS, and JavaScript (using frameworks like Bootstrap).
  - Implement user-friendly interfaces for tracking orders by ID, submitting feedback, and viewing feedback on the admin dashboard.
- **API Development:**
  - Develop APIs for real-time updates, courier status changes, feedback submission, and order management.
- **Testing and Debugging:**
  - Begin unit testing individual modules like user authentication, order processing, courier tracking, and feedback submission.

### 4. Testing

• **Objective:** Ensure the system is functional, reliable, and meets the specified requirements.

• **Activities:**

- Perform **Unit Testing** on each component of the system, including:
  - User registration and login.
  - Courier tracking and status updates.
  - Feedback submission and retrieval.
- Perform **Integration Testing** to ensure the backend and frontend communicate effectively:
  - Test the full order lifecycle from creation to delivery tracking and feedback submission.

## Courier Management System: A Web-Based Application to Track and Manage Couriers

- Perform **System Testing** to evaluate overall system performance under various conditions:
  - Test system scalability, security, and reliability with multiple simultaneous users.
  - Verify that the system can handle a high volume of tracking data.
- Perform **User Acceptance Testing (UAT)**:
  - Test the system with actual users (logistics personnel, customers, and admins) to ensure ease of use and functionality.
  - Collect feedback and refine the system based on user input.

### 5. Deployment

• **Objective:** Deploy the system to a live environment for end-users.

• **Activities:**

- Set up cloud hosting or web servers (e.g., AWS, Azure, or GCP) for the system to handle real-time traffic and user requests.
- Deploy the Django application and ensure proper configuration for live operation.
- Configure the system for secure data management, including user information and courier details.
- Make the application live for customers and logistics teams to use, with real-time updates for courier tracking and order management.
- Set up the admin dashboard for viewing customer feedback, managing couriers, and tracking system performance.
- Set up monitoring tools to track system performance and user activity.

## **6.Maintenance**

• **Objective:** Ensure the system's ongoing performance, reliability, and adaptation to future requirements.

• **Activities:**

- Monitor system performance and address issues such as downtime, slow responses, or bugs.
- Update the system with new features based on user feedback (e.g., adding route optimization or alerts).
- Regularly update the database with new courier, order, and transaction information.
- Maintain security patches and ensure data privacy through regular updates.

## CHAPTER 7

### IMPLEMENTATION

#### 7.1 SAMPLECODE

```
from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponseRedirect

import os

import pickle

import pymysql

import os

from django.core.files.storage import FileSystemStorage

from datetime import date

import matplotlib.pyplot as plt

import io

import base64

import numpy as np

global uname

def ViewFeedback(request):

    if request.method == 'GET':

        output = '<table border=1><tr>'

        output+='<td><font size="" color="black">Courier&nbsp;ID</td>'

        output+='<td><font size="" color="black">Feedback</td>'

        output+='<td><font size="" color="black">Feedback Date</td>'
```

## Courier Management System: A Web-Based Application to Track and Manage Couriers

```
output+=''<td><font size="" color="black">Feedback Rank</td></tr>'
```

```
rank = []
```

```
con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password =  
'LaDB43log', database = 'courier',charset='utf8')
```

```
with con:
```

```
    cur = con.cursor()
```

```
    cur.execute("select * FROM feedback")
```

```
    rows = cur.fetchall()
```

```
    for row in rows:
```

```
        output+=''<tr><td><font size="" color="black">'+str(row[0])+''</td>'
```

```
        output+=''<td><font size="" color="black">'+str(row[1])+''</td>'
```

```
        output+=''<td><font size="" color="black">'+str(row[2])+''</td>'
```

```
        output+=''<td><font size="" color="black">'+str(row[3])+''</td></tr>'
```

```
        rank.append(row[3])
```

```
output += "</table><br/>"
```

```
unique, count = np.unique(np.asarray(rank), return_counts=True)
```

```
mplt.pie(count,labels=unique,autopct='%1.1f%%')
```

```
mplt.title('Feedback Ranking Graph')
```

```
mplt.axis('equal')
```

```
buf = io.BytesIO()
```

```
mplt.savefig(buf, format='png', bbox_inches='tight')
```

```
mplt.close()
```

```
img_b64 = base64.b64encode(buf.getvalue()).decode()
```

```
context= {'data':output, 'img': img_b64}
```

```
return render(request, 'ViewFeedback.html', context)
```

## **CHAPTER 8**

### **TESTING**

#### **8.1 TESTING METHODOLOGIES**

Testing is a process of executing a program with the aim of finding errors. To make our software perform well it should be error free. If testing is done successfully, it will remove all the errors from the software.

##### **Types of Testing**

1. White Box Testing
2. Unit testing
3. Integration Testing
4. Alpha Testing
5. Beta Testing
6. Performance Testing and so on

##### **White Box Testing**

Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performed by software developers

##### **Black Box Testing**

A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality.

##### **Unit Testing**

Software verification and validation method in which a programmer tests if individual units of source code are fit for use. It is usually conducted by the development team.

### **Integration Testing**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

### **Alpha Testing**

Type of testing a software product or system conducted at the developer's site. Usually, it is performed by the end users.

### **Beta Testing**

Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

### **Performance Testing**

Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.

## **8.2 USER TRAINING**

Effective user training is essential to ensure that the users can efficiently utilize the Courier Management System: A Web-Based Application to Track and Manage Couriers for tracking and managing couriers. The training plan includes the following components:

### **1. Introduction to the System**

- **Overview:** Explain the purpose of the system—tracking and managing couriers—and how it facilitates seamless courier monitoring and feedback management.
- **User Scenarios:** Discuss scenarios like tracking couriers, receiving feedback from customers, and the roles of administrators in managing these couriers.



## 2. System Interface Training

- **Dashboard Navigation:** Guide users through the admin dashboard, showing how to manage feedback, track couriers using tracking IDs, and view courier statuses.
- **Key Features:** Demonstrate how users (customers) can input tracking IDs and track their couriers, and how the admin can view customer feedback and courier statuses.
- **Data Views:** Show how to interpret the status data, including delivered, in-transit, and pending couriers.

## 3. Tracking and Feedback Interpretation

- **Tracking Data:** Train users on how to use courier tracking IDs to access the real-time status of deliveries.
- **Feedback Management:** Show how to view and respond to customer feedback via the admin dashboard.

## 4. Using the System for Decision-Making

- **Tracking for Delivery Efficiency:** Demonstrate how to use the system for monitoring delivery times, improving customer satisfaction by identifying delivery bottlenecks.
- **Feedback-Based Improvement:** Train users (admins) to act on customer feedback, optimizing delivery routes or improving customer service based on the insights provided.

## 5. Troubleshooting and Support

- **Common Issues:** Provide guidance on resolving common user problems such as invalid tracking ID entries or issues viewing courier statuses.
- **Help Resources:** Direct users to available resources like user manuals, FAQs, or helpdesk contacts for support.

## 6. Feedback Mechanism

- **User Feedback Collection:** Teach users how to provide feedback on the system, highlighting areas for improvement or reporting issues to help refine future versions.

## 8.3 MAINTENANCE

Maintenance ensures the system continues to run smoothly, offering efficient courier management and feedback monitoring over time. Here's a maintenance plan for the Courier Management System: A Web-Based Application to Track and Manage Couriers:

### 1. Data Maintenance

- **Tracking Data Integrity:** Regular checks for missing or outdated tracking information to prevent incorrect delivery statuses.
- **Database Management:** Ensure database optimization for smooth data retrieval, especially during high traffic (e.g., during festive seasons or sales).
- **Data Archiving:** Archive old courier data periodically to maintain system performance while retaining important information.

### 2. System and Code Maintenance

- **Code Bug Fixing:** Address system errors, bugs, and inconsistencies promptly.
- **Performance Monitoring:** Regular monitoring of system performance to ensure it runs efficiently under high user load.
- **Security Updates:** Apply regular security patches to safeguard user and system data.

### 3. User Interface Updates

- **UI Enhancements:** Based on user feedback, improve the admin dashboard interface for better feedback management and user experience.
- **User Access Control:** Review and update user access levels to ensure appropriate privileges for admins and customers.

#### 4. Infrastructure Maintenance

- **Server and Cloud Monitoring:** Monitor server uptime, performance, and resource allocation to handle peak loads.
- **Backup and Disaster Recovery:** Implement regular system backups and disaster recovery procedures to ensure minimal downtime in case of failures.
- **Latency Optimization:** Monitor and reduce response times for smoother customer and admin interactions.

#### 5. Bug Tracking and Issue Resolution

- **Bug Tracking System:** Use a bug tracking tool to document, prioritize, and resolve bugs.
- **User Feedback:** Incorporate feedback into the development process for system improvements and new feature integrations.

#### 6. Documentation Updates

- **Updating Technical Documentation:** Ensure system architecture and implementation documents are updated to reflect any code or system changes.
- **User Manual Updates:** Periodically refresh the user manual to incorporate new system features or updates.

## 8.4 TESTING STRATEGY

To ensure your Courier Management System functions smoothly and meets user expectations, a thorough testing strategy is essential. Here's a structured approach for testing:

#### 1. Unit Testing

- **Tracking Module:** Test the core functionality of tracking couriers using the tracking ID.
- **Admin Dashboard:** Verify the admin features, including feedback management and status monitoring.
- **Error Handling:** Ensure that any incorrect inputs (invalid tracking IDs, system errors) are handled gracefully.

## 2. Integration Testing

- **Frontend-Backend Integration:** Test the connection between the frontend (customer-facing) and backend (admin panel) for smooth data flow.
- **Database Integration:** Ensure that updates in the tracking database reflect in real-time on the user interface.
- **User Input Validation:** Ensure proper handling of inputs from users, especially during feedback submissions and tracking queries.

## 3. System Testing

- **End-to-End Testing:** Simulate real-world scenarios where users (customers and admins) interact with the system from tracking couriers to giving and viewing feedback.
- **Data Consistency:** Check that the courier status and feedback remain consistent across multiple sessions.
- **Load Testing:** Simulate multiple users accessing the system simultaneously to evaluate performance under stress (e.g., tracking multiple couriers at once).

## 4. Performance Testing

- **Latency Testing:** Measure the system's response time for tracking couriers and updating feedback.
- **Throughput Testing:** Check the system's ability to handle a high volume of tracking queries or admin actions.
- **Scalability Testing:** Ensure the system can scale when more couriers are added or more users access it simultaneously.

## 5. User Acceptance Testing (UAT)

- **Scenario-Based Testing:** Develop scenarios based on actual user tasks, such as tracking couriers or submitting feedback.
- **Usability Testing:** Ensure the system is easy to use, and the interface is intuitive for both customers and admins.

- **Feedback Collection:** Gather real user feedback to further enhance the system's functionality and user interface.

## 6. Security and Compliance Testing

- **Data Privacy Testing:** Verify the system adheres to data privacy regulations (e.g., GDPR) and ensures secure handling of user data.
- **Access Control Testing:** Ensure that different users (admins and customers) have the appropriate permissions to access certain features.

## 7. Regression Testing

- **Feature Updates Testing:** Whenever new features or updates are added, test for compatibility with existing functionalities to avoid breaking any parts of the system.
- **Automated Regression Tests:** Use automated tools to continuously test the system after every update.

## 8. Continuous Monitoring in Production

- **Real-Time Monitoring:** Set up monitoring tools to track system errors and performance in production.
- **Feedback Loop:** Keep a continuous loop for collecting feedback from both customers and administrators to keep refining the system.

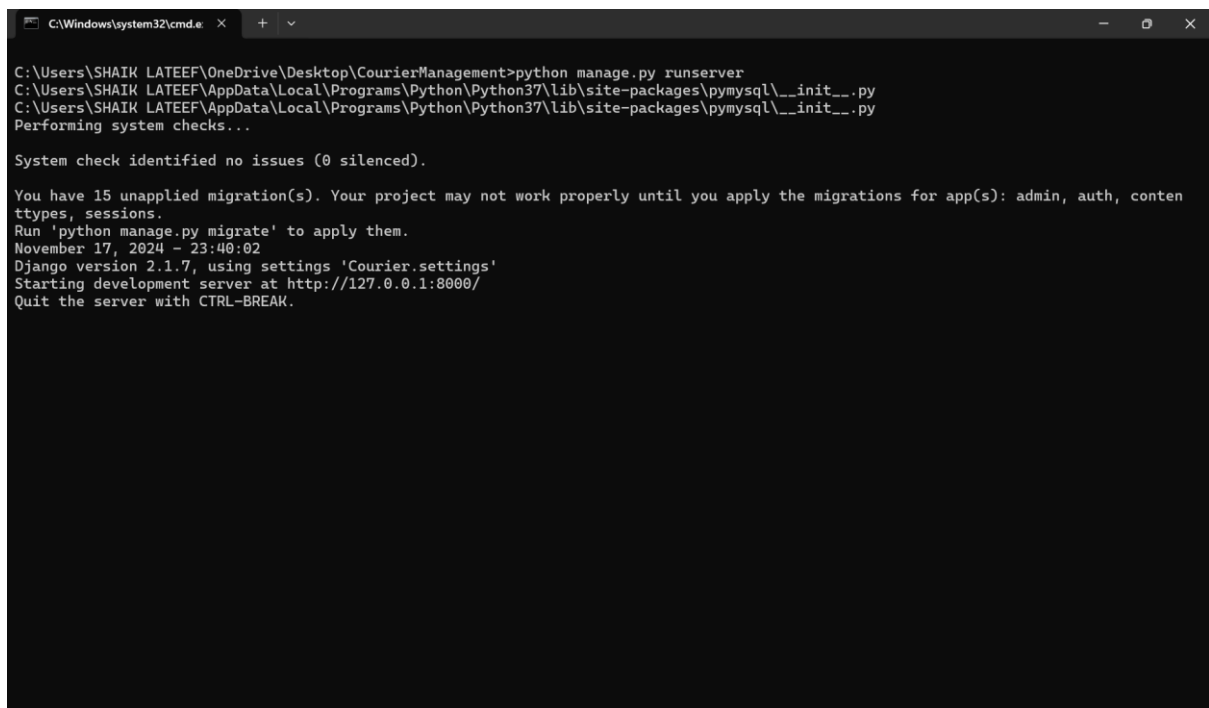
## CHAPTER 9

### OUTPUT SCREEN

#### 9.1 SCREEN SHOTS

To run project, install MYSQL and then open MYSQL console and then copy content from 'Database.txt' file and paste in MYSQL console to create database.

Now install python 3.7 and then install all packages given in 'requirements.txt' file and then double click on 'run.bat' to start python web server and get below page



```
C:\Windows\system32\cmd.e  x  +  v
C:\Users\SHAIK LATEEF\OneDrive\Desktop\CourierManagement>python manage.py runserver
C:\Users\SHAIK LATEEF\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
C:\Users\SHAIK LATEEF\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, conten
ttypes, sessions.
Run 'python manage.py migrate' to apply them.
November 17, 2024 - 23:40:02
Django version 2.1.7, using settings 'Courier.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig. 9.1 Terminal Window

In the above screen, the Python server has started. Open a browser and enter the URL <http://127.0.0.1:8000/index.html>, then press the Enter key to navigate to the page shown below.

## Courier Management System: A Web-Based Application to Track and Manage Couriers

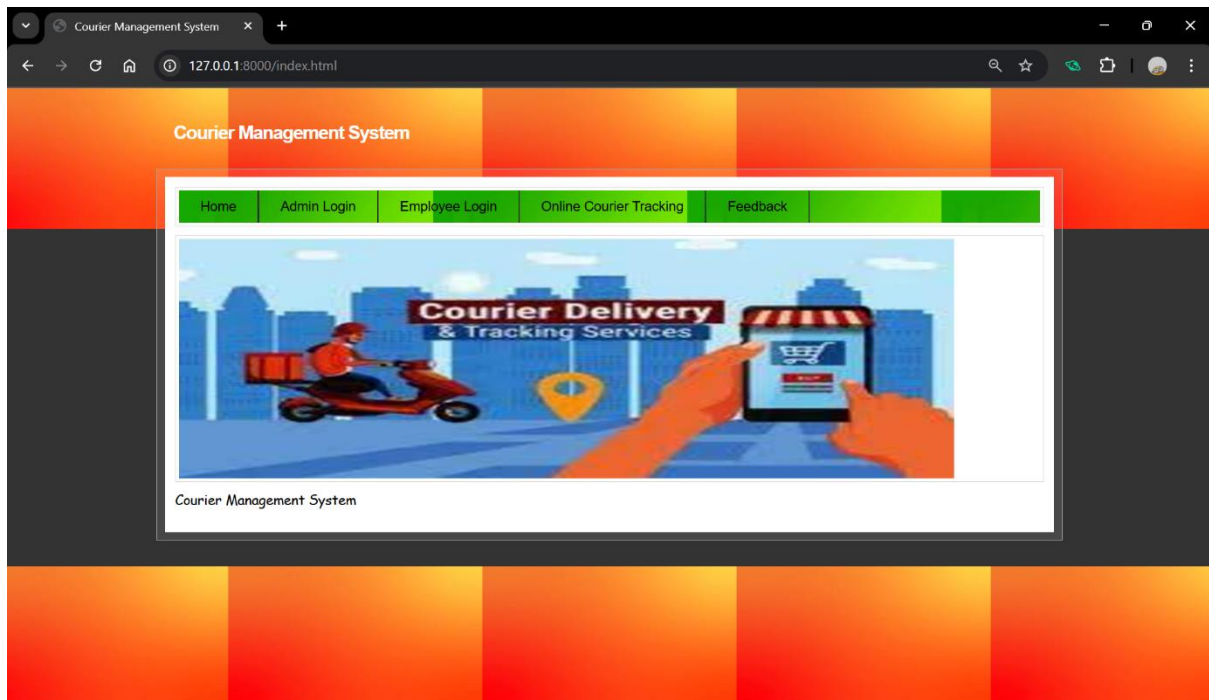


Fig. 9.2 Home Screen

In the above screen, click on the 'Admin Login' link to navigate to the login page displayed below.

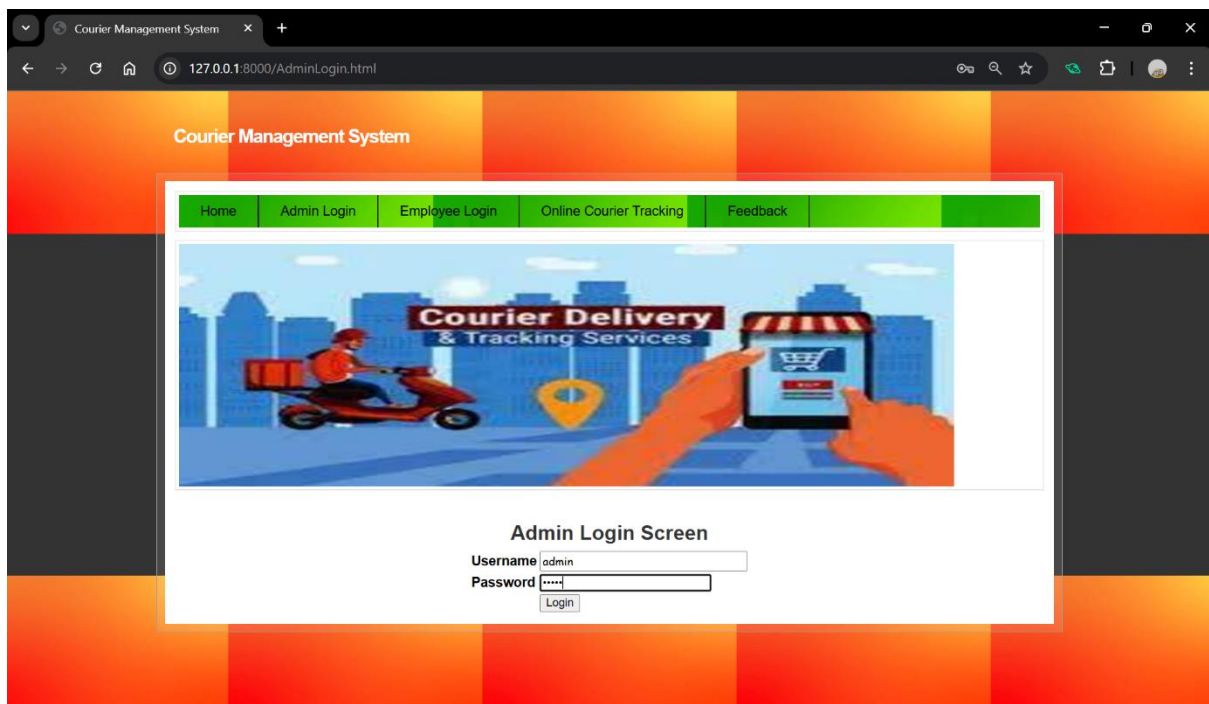


Fig. 9.3 Admin Login Screen

In the above screen, the admin is logged in. After logging in, the page shown below will be displayed.

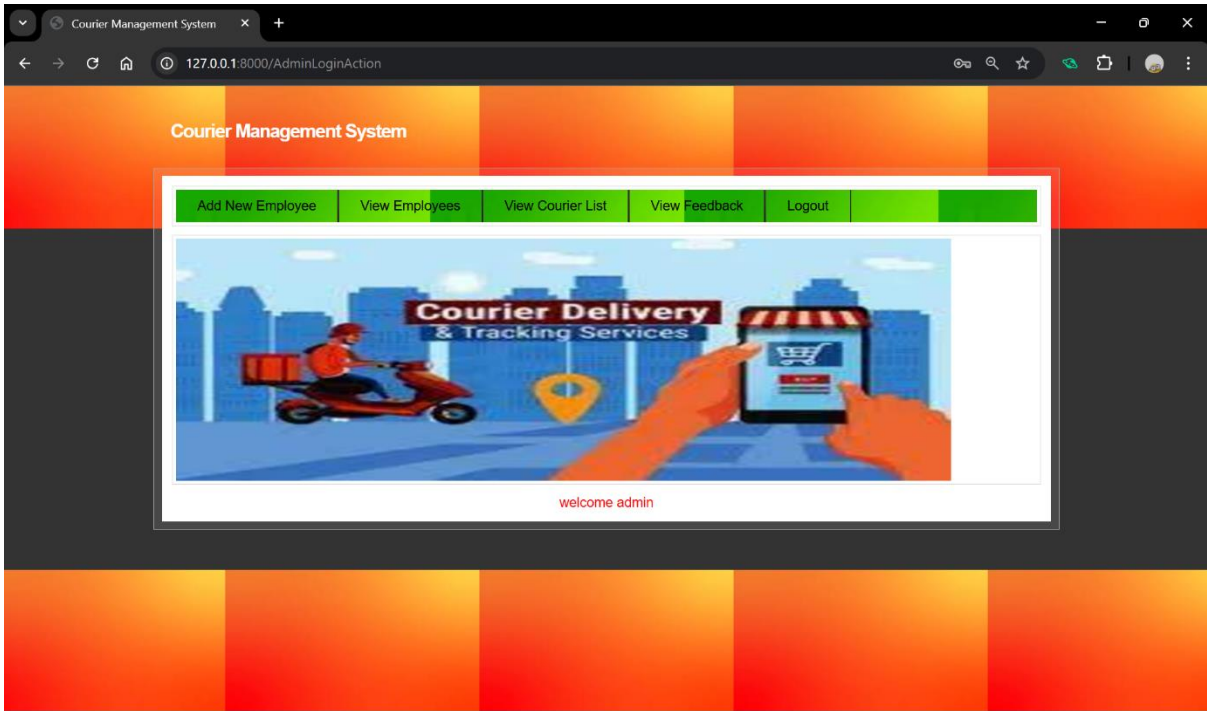


Fig. 9.4 Admin Home Screen

In the above screen, the admin can click on the 'Add New Employee' link to enter employee details, which will redirect to the page shown below.

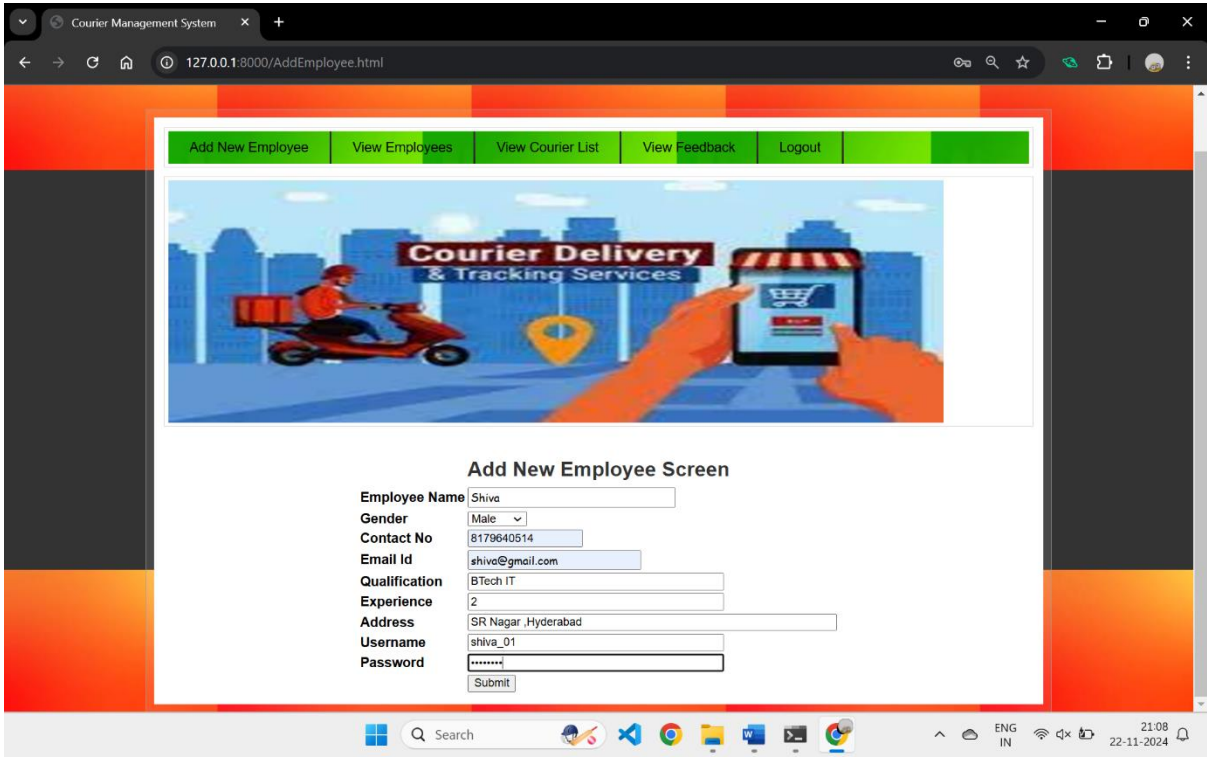


Fig. 9.5 Add New Employee Screen



# Courier Management System: A Web-Based Application to Track and Manage Couriers

In the above screen, the admin will enter the employee details and then press the button to save the information.

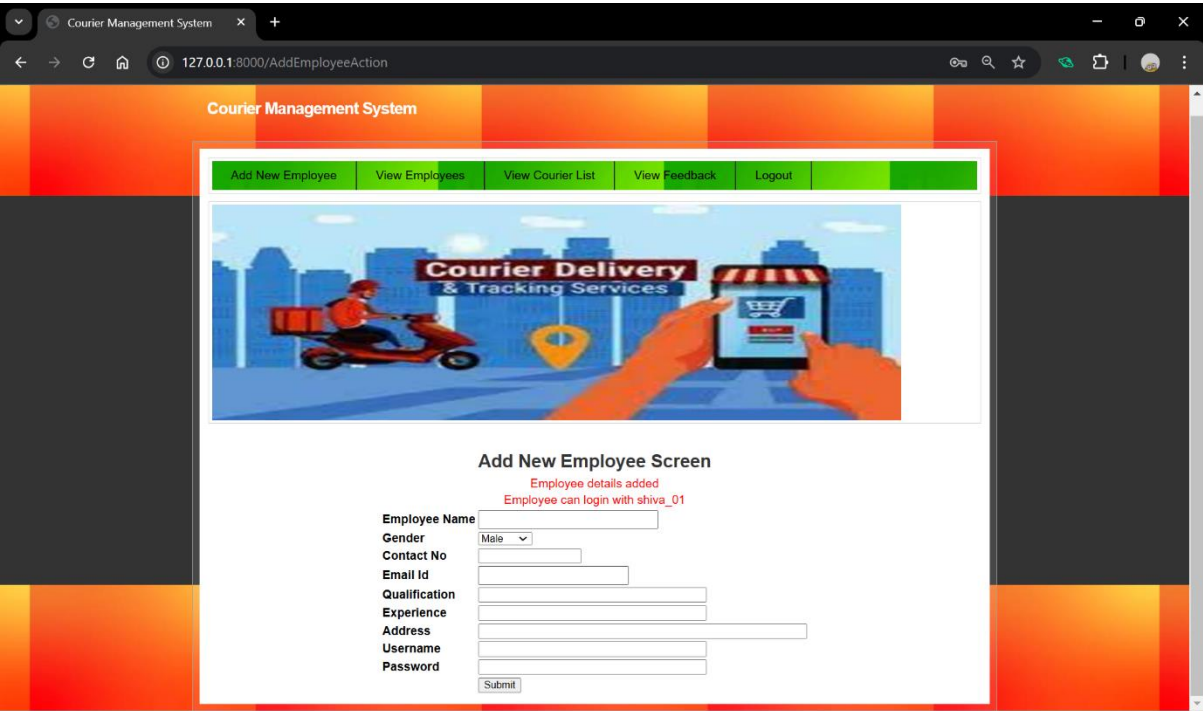


Fig. 9.6 Employee Details Added

In the above screen, the employee details are added to the database. Using these login credentials, the employee can log in to the application. Now, click on the 'View Employee' link to view the list of all employees.

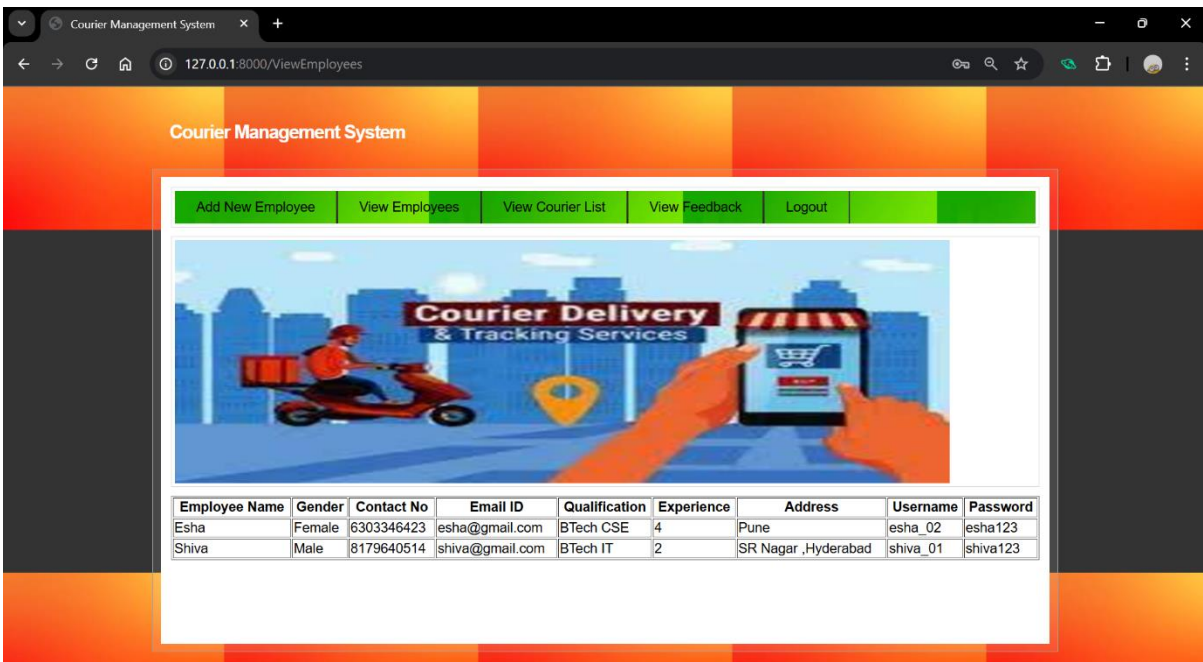


Fig. 9.7 List of Employees

In the above screen, the admin can view details of all employees. Now, click on the 'View Courier List' to see the details of all booked couriers.

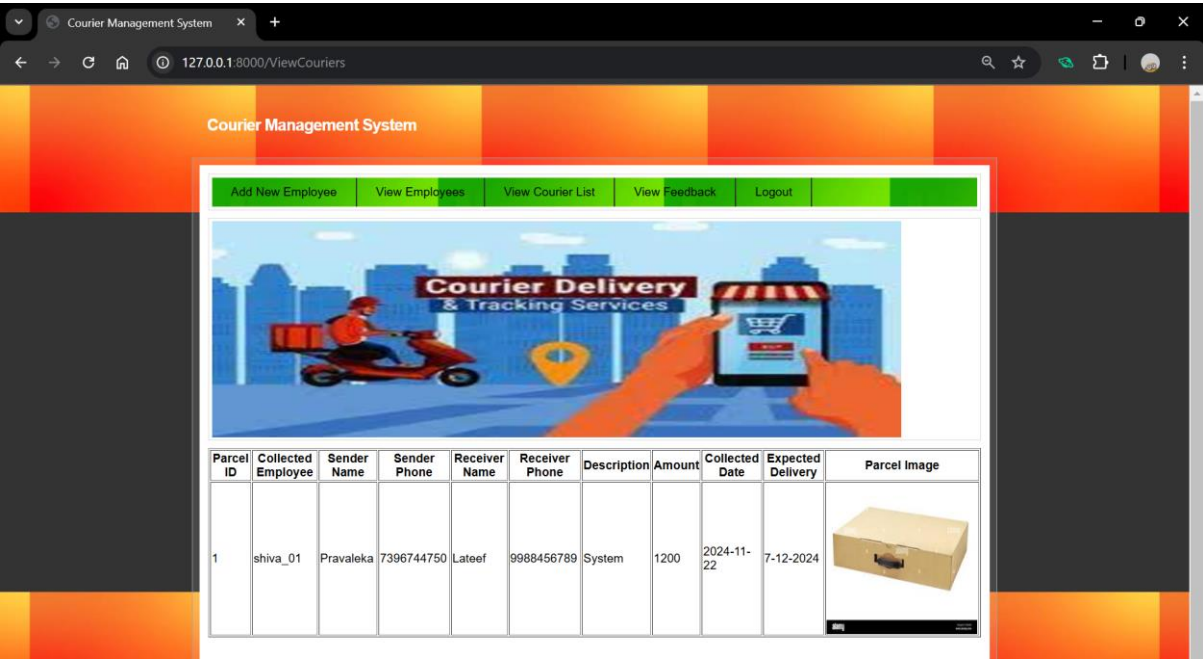


Fig. 9.8 List of all Booked Couriers

In the above screen, the admin can view a list of all booked couriers. Now, log out and log in as an employee.

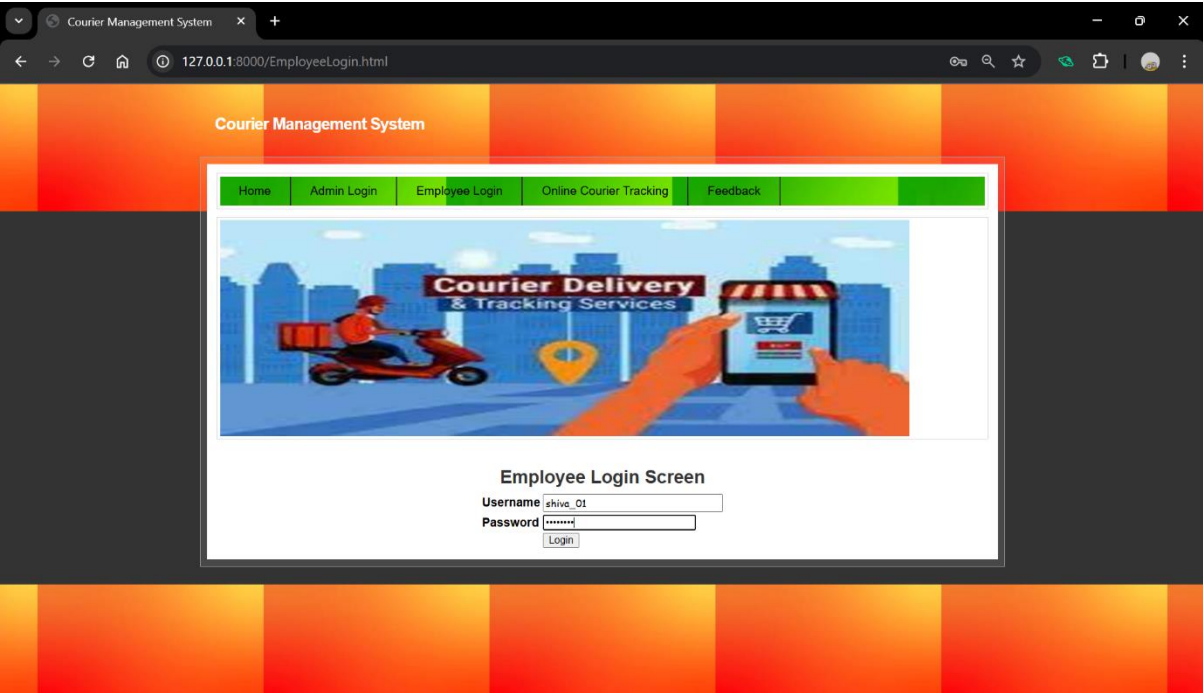


Fig. 9.9 Employee Login Screen

# Courier Management System: A Web-Based Application to Track and Manage Couriers

In the above screen, the employee is logged in. After logging in, the page shown below will be displayed.

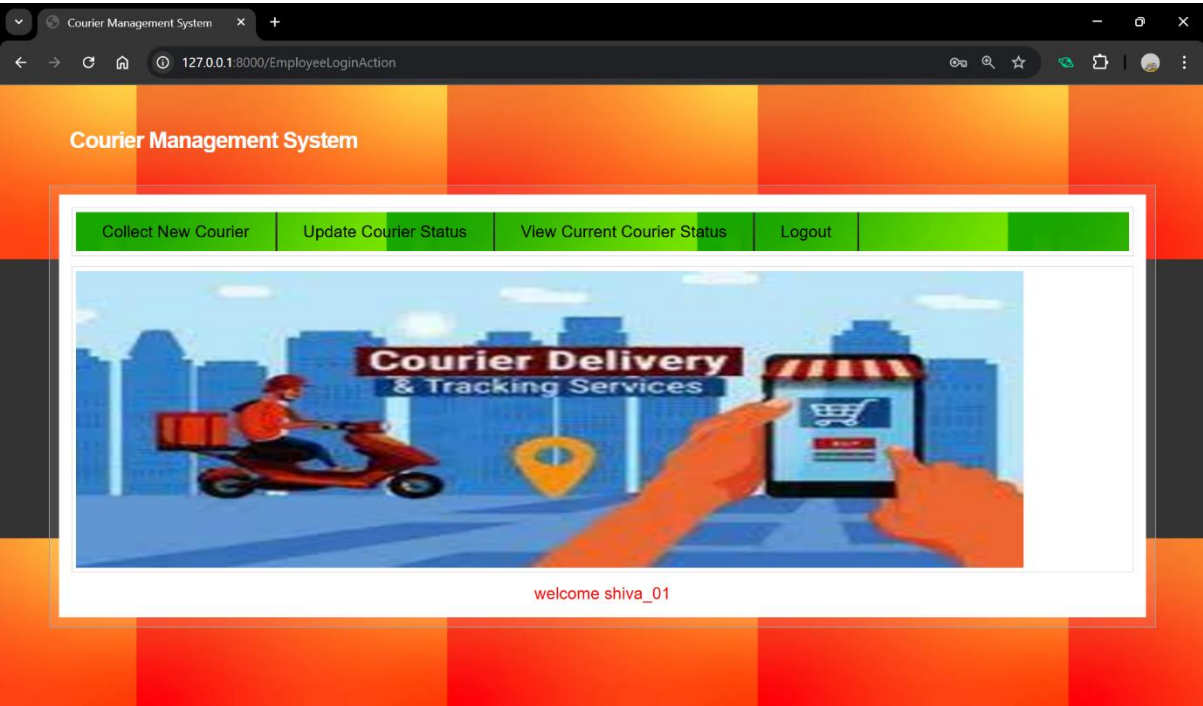


Fig. 9.10 Employee Home Screen

In the above screen, the employee will click on the 'Collect New Courier' link to add new courier details

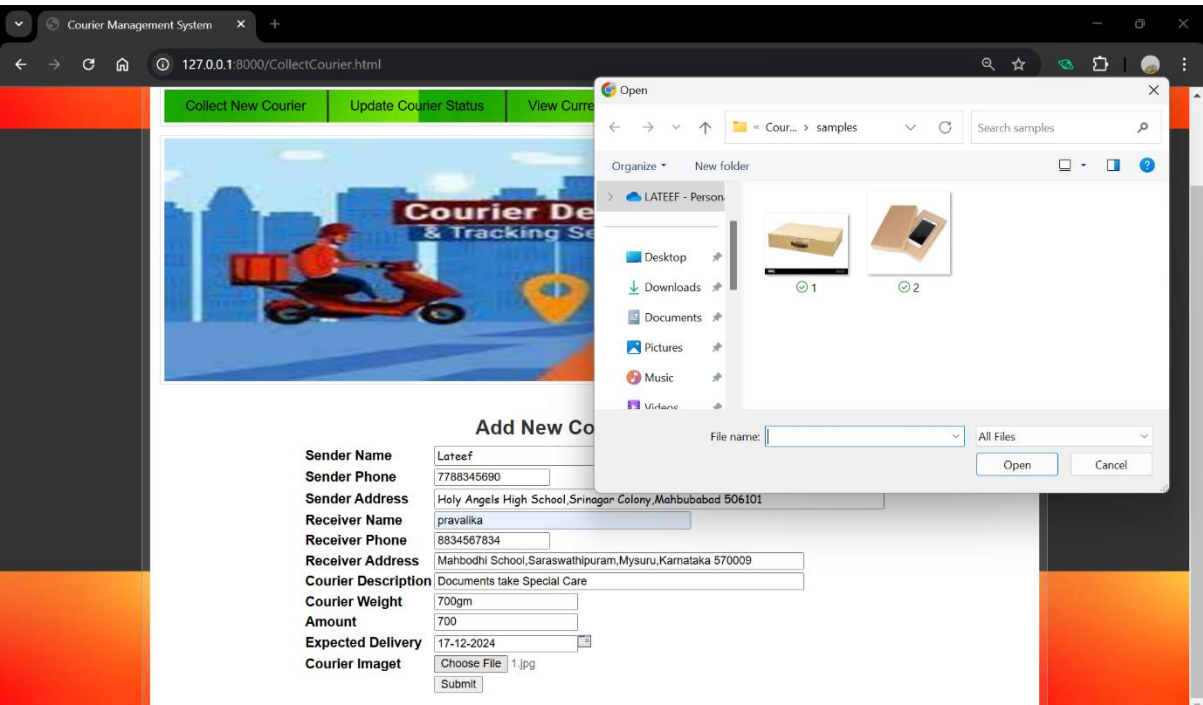
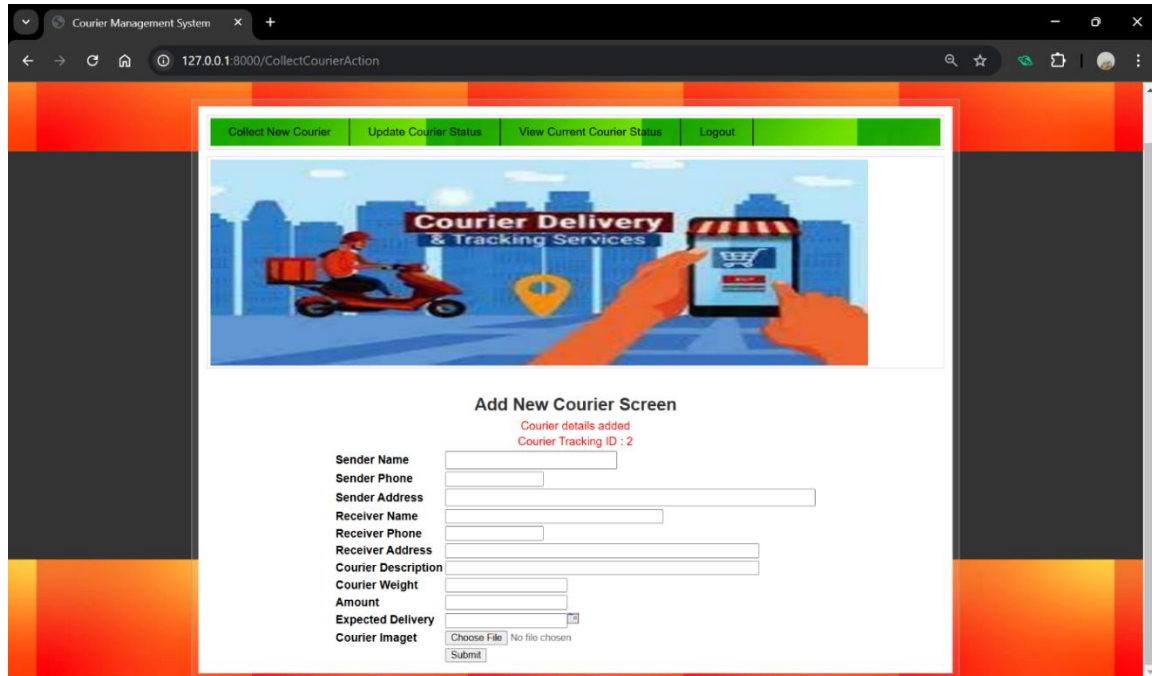


Fig. 9.11 Add New Courier Screen

## Courier Management System: A Web-Based Application to Track and Manage Couriers

In the above screen, the employee will collect all courier details, upload the courier item image, and then press the 'Submit' button to add the new courier details. This will redirect to the page shown below.



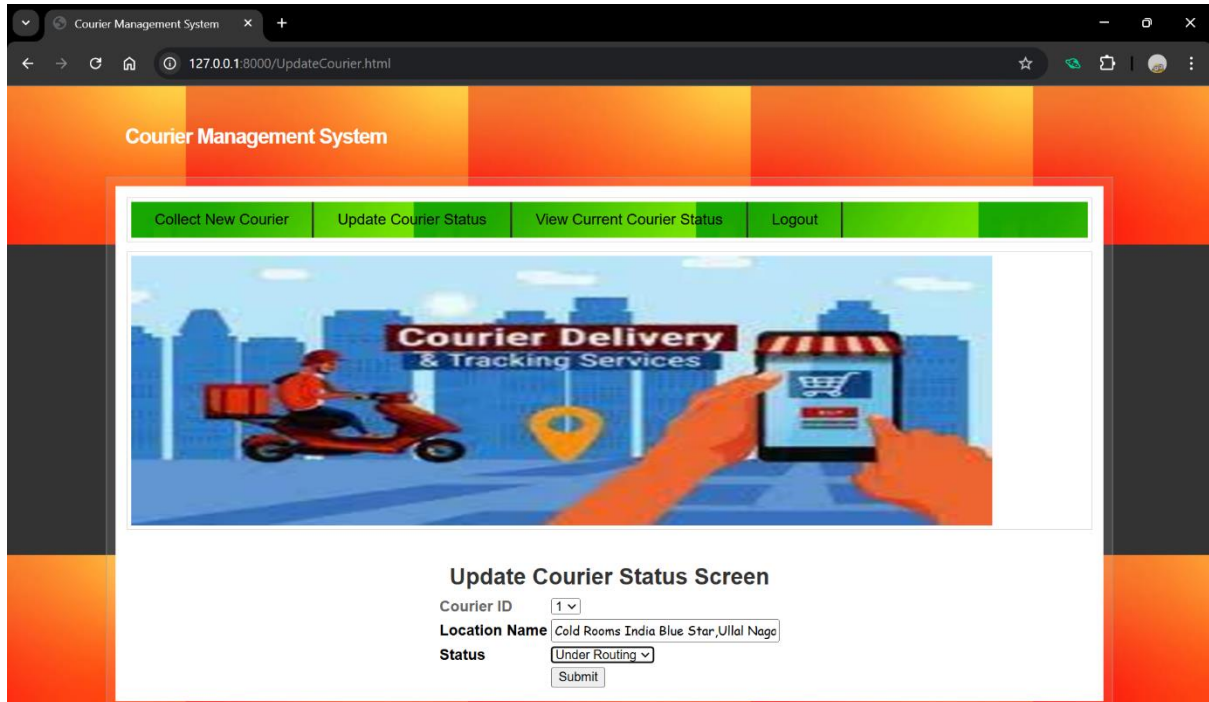
The screenshot displays a web browser window with the title 'Courier Management System'. The address bar shows the URL '127.0.0.1:8000/CollectCourierAction'. The page features a navigation bar with four links: 'Collect New Courier', 'Update Courier Status', 'View Current Courier Status', and 'Logout'. Below the navigation bar is a banner image with the text 'Courier Delivery & Tracking Services' and an illustration of a courier on a scooter. The main content area is titled 'Add New Courier Screen' and contains a form with the following fields: Sender Name, Sender Phone, Sender Address, Receiver Name, Receiver Phone, Receiver Address, Courier Description, Courier Weight, Amount, Expected Delivery, and Courier Image. A red message above the form states 'Courier details added' and 'Courier Tracking ID : 2'. The 'Courier Image' field includes a 'Choose File' button and a 'Submit' button.

Fig. 9.12 Courier Details Added and assigned Id

In the above screen, courier details are added, and the courier can be tracked using ID 2. As the courier progresses, the employee can click on the 'Update Courier Status' link to update the location, which will display the page shown below.



## Courier Management System: A Web-Based Application to Track and Manage Couriers



Courier Management System

Collect New Courier | Update Courier Status | View Current Courier Status | Logout

**Courier Delivery & Tracking Services**

**Update Courier Status Screen**

Courier ID: 1

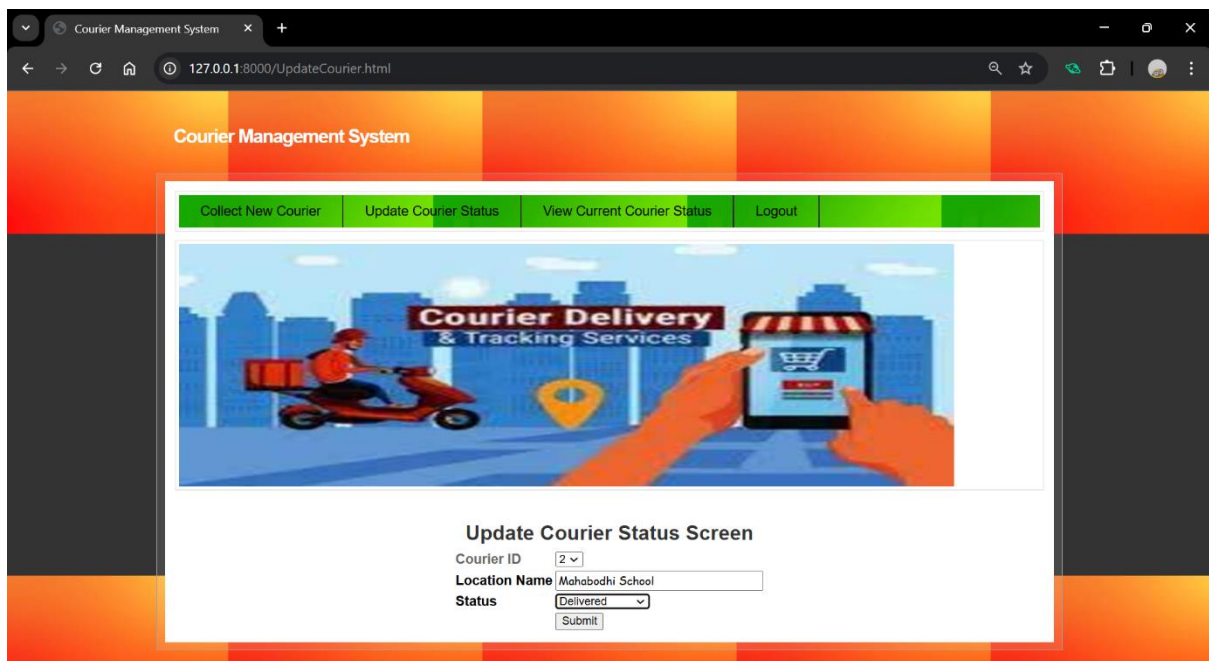
Location Name: Cold Rooms India Blue Star,Ullal Nage

Status: Under Routing

Submit

Fig. 9.13 Update Courier Status Screen

In the above screen, the employee will select the courier ID and update the current location. This process continues until the courier is delivered. Once delivered, the status will be updated as 'Delivered'.



Courier Management System

Collect New Courier | Update Courier Status | View Current Courier Status | Logout

**Courier Delivery & Tracking Services**

**Update Courier Status Screen**

Courier ID: 2

Location Name: Mahabodhi School

Status: Delivered

Submit

Fig. 9.14 Updating Courier Status

## Courier Management System: A Web-Based Application to Track and Manage Couriers

In the above screen, once a courier is delivered, the employee will select 'Delivered' as the status and update it. Then, they can click on the 'View Courier Current Status' link to view the page shown below.

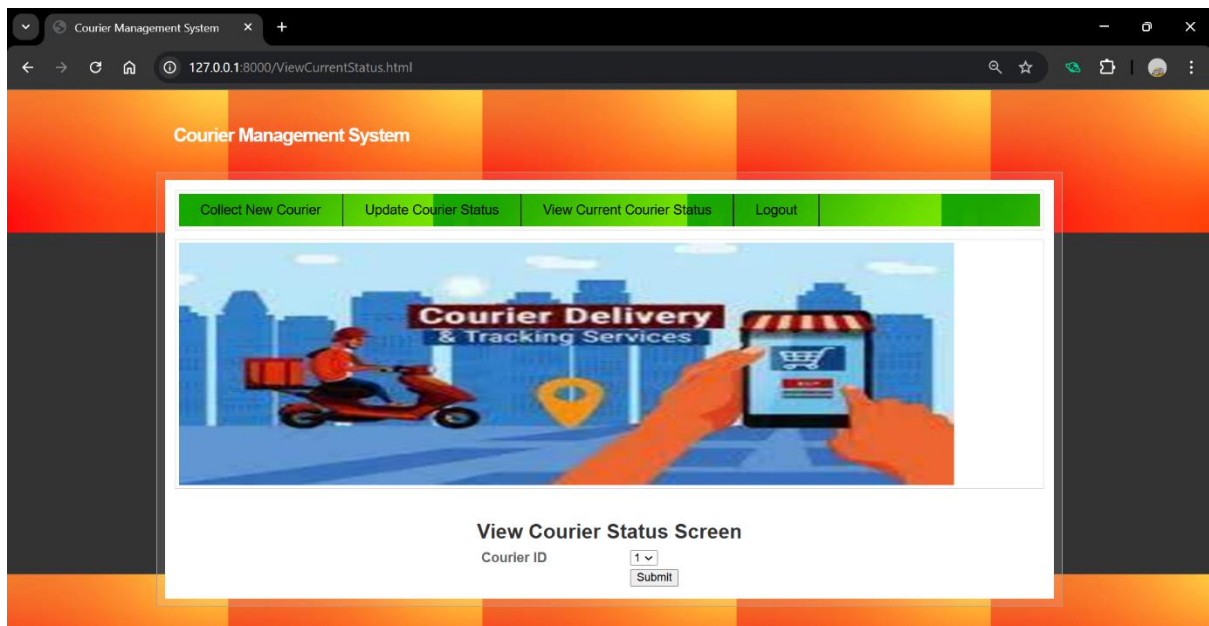


Fig. 9.15 Employee View Courier Status Screen

In the above screen, select the courier ID and then click on the 'Submit' button to view the page displayed below.

Courier Management System: A Web-Based Application to Track and Manage Couriers

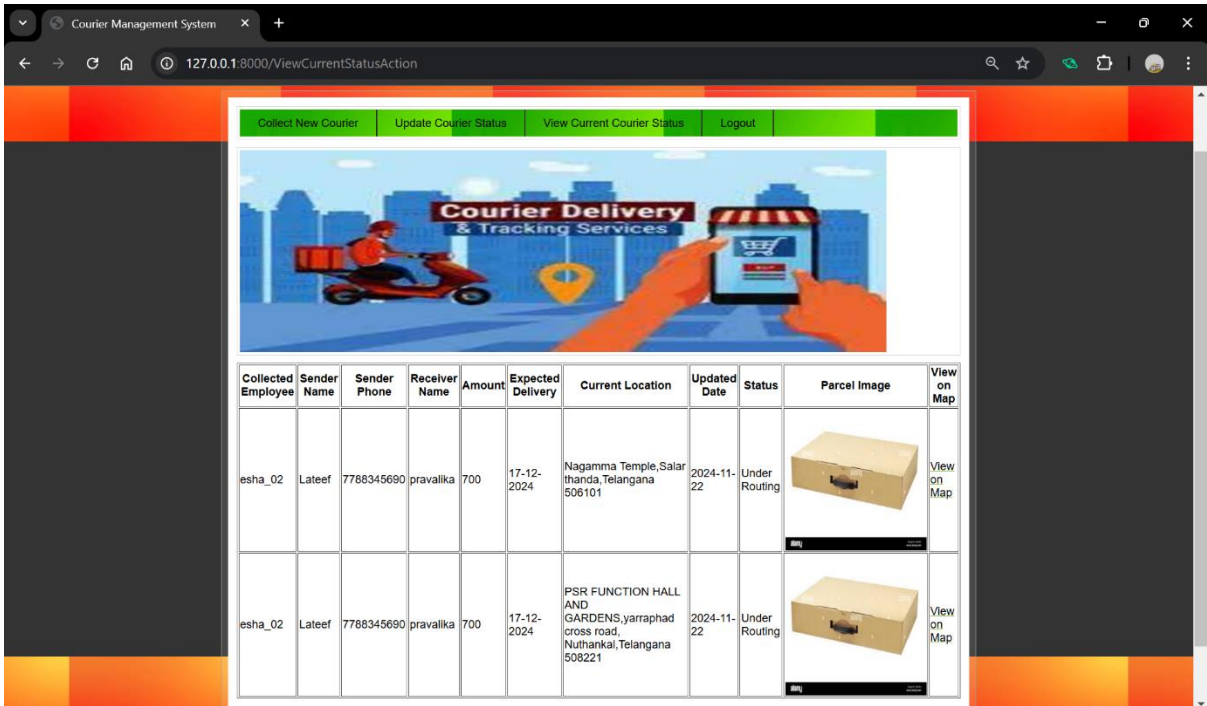


Fig. 9.16 Employee Courier Tracking Screen

In the above screen, for the selected courier ID, the employee can view the current location. Once delivered, the status will be shown as 'Delivered.' The employee can then click on the 'View on Map' link to see the page displayed below.

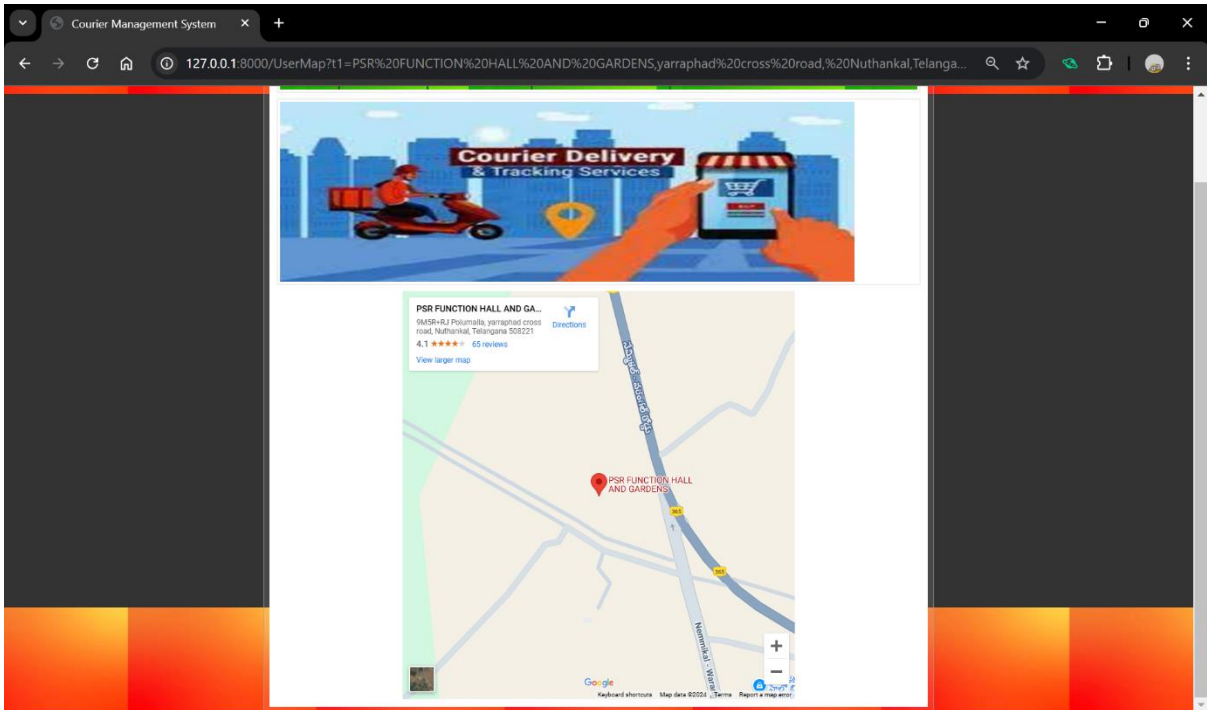


Fig. 9.17 Courier Track Screen View Map

Courier Management System: A Web-Based Application to Track and Manage Couriers

In the above screen, the employee can see that the 'courier current location is PSR Function Hall, Nuthankal.' Now, log out and track the same courier as a user.

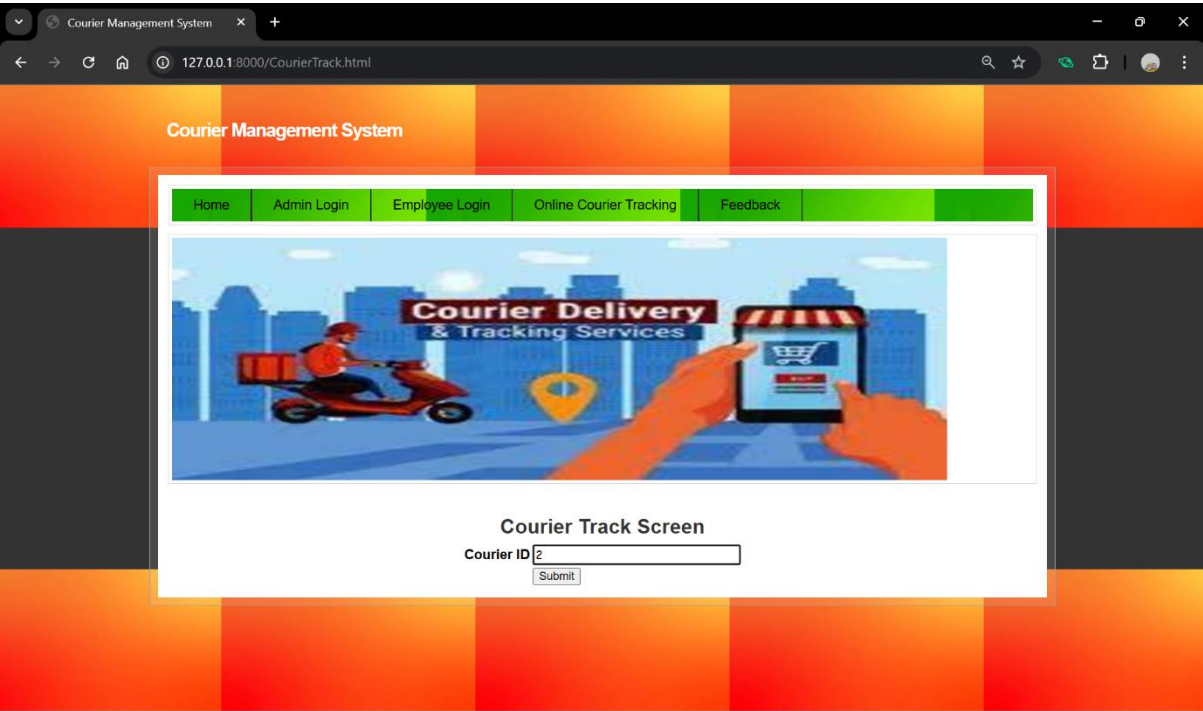


Fig. 9.18 User Courier Track Screen

In the above screen, the user can track their courier by simply entering the ID and pressing the button, which will redirect them to the page shown below

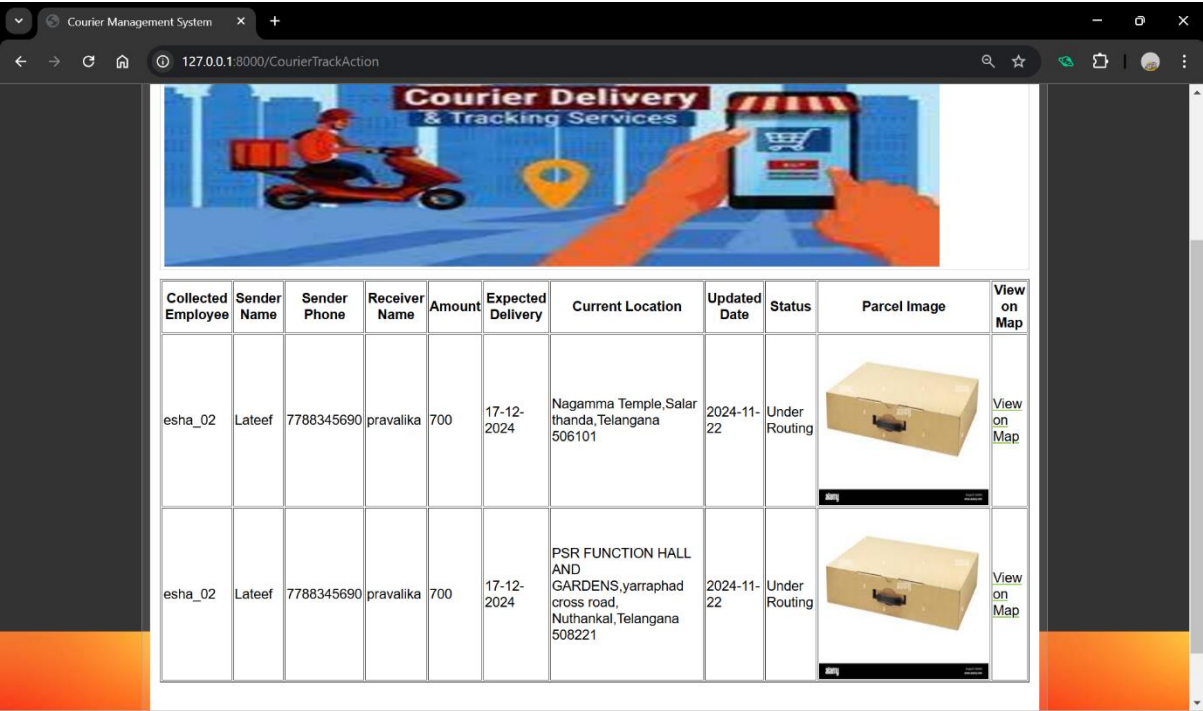


Fig. 9.19 User Courier Tracking in Detail



## Courier Management System: A Web-Based Application to Track and Manage Couriers

In above screen, user can also see all possible location of his courier. Similarly, by following above screens you can manage and run all courier system.

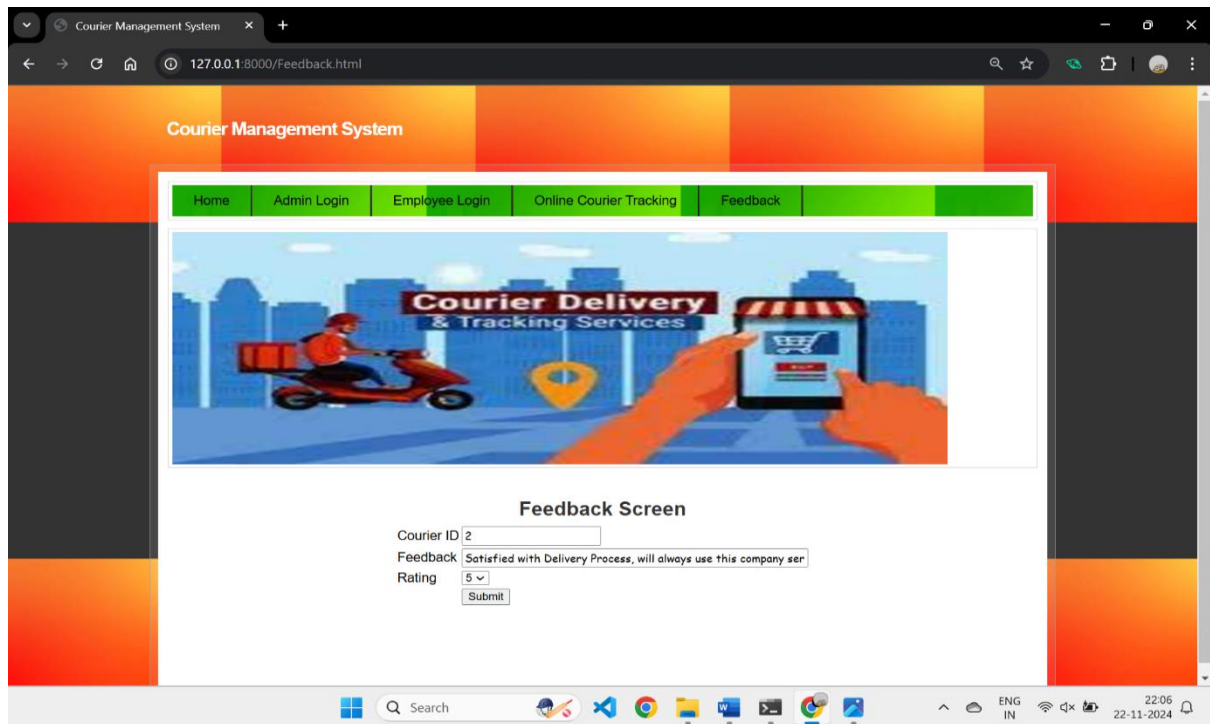


Fig. 9.20 User Feedback Screen

In the above screen, the user can provide their feedback and rate their overall experience on a scale of 1 to 5.

## Courier Management System: A Web-Based Application to Track and Manage Couriers

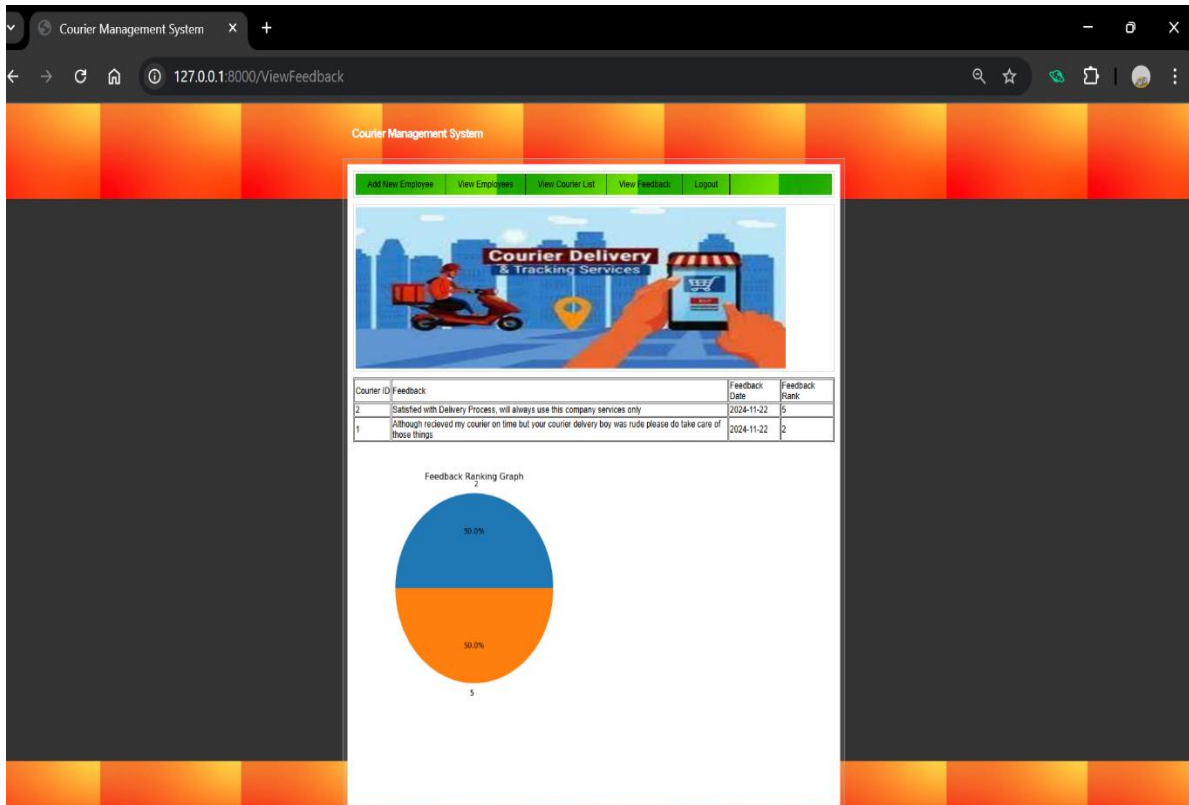


Fig. 9.21 Admin View Feedback Screen

In the above screen, the admin can view all the feedback provided by users. Additionally, a pie chart is generated based on the feedback ratings given by the users, offering a visual representation of the overall feedback distribution.

Employee Name	Gender	Contact No.	Email	Qualification	Experience	Address	Username	Password
Esha	Female	6303346423	esha.johnson@mail.com	BTech CSE	4 years	Pune	esha_02	esha123
Shiva	Male	8179640514	shiva.rajana@mail.com	BTech IT	2 years	SR Nagar, Hyderabad	shiva_01	shiva123

Table No. 9.22 Employee Table

The above table represents how employee data is stored in the system, including personal information, qualifications, and login credentials.

## Courier Management System: A Web-Based Application to Track and Manage Couriers

Parcel ID	Collected Employee	Sender Name	Sender Phone	Sender Address	Receiver Name	Receiver Phone	Receiver Address	Description	Parcel Weight
1	shiva_01	Pravaleka	7396744750	Belaku Study Center, Saraswathipuram, Mysuru, Karnataka 570009	Lateef	9988456789	Holy Angels High School, Mahabubabad	System	1.2kg
2	esha_02	Lateef	7788345690	Holy Angels High School, Mahabubabad 506101	Pravalika	8834567834	Mahbodhi School, Saraswathipuram, Mysuru	Documents take Special Care	700gm

Table No. 9.23 Parcel Table

The above table contains details about parcels collected by employees. It includes sender and receiver information, parcel description, weight, and delivery schedules.

Parcel ID	Employee Reporting	Current Location	Update Date	Status
1	esha_02	Cold Rooms India Blue Star, Ullal Nagar, Manganahalli, Bengaluru, Karnataka 560060	2024-11-22	Under Routing
2	esha_02	Nagamma Temple, Salar Thanda, Telangana 506101	2024-11-22	Under Routing
2	esha_02	PSR FUNCTION HALL AND GARDENS, Yarraphad Cross Road, Nuthankal, Telangana 508221	2024-11-22	Under Routing
2	esha_02	Mahabodhi School	2024-11-22	Delivered
1	shiva_01	Holy Angels High School, Mahabubabad	2024-11-22	Delivered

Table No. 9.24 Update Status Table

## Courier Management System: A Web-Based Application to Track and Manage Couriers

The above table tracks the status of parcels, including the employee reporting, current location, update date, and status. This ensures transparency in the delivery process.

Username	Feedback	Feedback Date	Feedback Rank
2	Satisfied with Delivery Process, will always use this company services only	2024-11-22	5
1	My package was delivered late, and the packaging was torn. Not satisfied with the service.	2024-11-22	2

Table No. 9.25 Feedback Table

The above table records feedback from customers, providing comments, feedback date, and ranking to improve services.

## **CHAPTER 10**

### **CONCLUSION AND FUTURE SCOPE**

#### **10.1 CONCLUSION**

The Courier Management System: A Web-Based Application to Track and Manage Couriers, developed using Python, Django, HTML, CSS, and JavaScript, successfully addresses the core needs of managing and tracking couriers in a streamlined and user-friendly manner. The system enables administrators to monitor feedback through an intuitive dashboard, while customers can easily track their couriers via a unique tracking ID. Although the current version lacks certain features like real-time notifications, payment gateway integration, and route optimization, these have been recognized as potential areas for improvement in future releases. The system is designed to be easily maintainable and adaptable for future expansions, ensuring continued relevance and efficiency.

## 10.2 Future Scope

The **Courier Management System: A Web-Based Application to Track and Manage Couriers** has significant potential for expansion and improvement, driven by rapid technological advancements and changing market needs. As new innovations emerge, the system will evolve to integrate cutting-edge technologies that enhance its functionality, efficiency, and user experience. The future scope includes several promising directions:

### 1. **Integration of IoT for Enhanced Tracking and Predictive Maintenance**

As the Internet of Things (IoT) continues to expand, the CMS can integrate advanced sensors and devices for more precise tracking of couriers and parcels. These IoT-enabled systems will provide real-time data, improving tracking accuracy and enabling predictive maintenance for couriers and vehicles.

### 2. **Machine Learning for Improved Predictions and Forecasting**

The system will incorporate machine learning algorithms to enhance demand forecasting and optimize delivery routes. By analysing historical data, weather conditions, and real-time traffic information, the system can predict delivery times more accurately and dynamically adjust routes to reduce delivery costs and improve efficiency.

### 3. **AI-Powered Chatbot for Enhanced Customer Interaction**

An AI-driven chatbot will be introduced to provide instant assistance to customers. This chatbot will handle real-time queries, update delivery statuses, and automate frequently asked questions, thus enhancing customer engagement and offering a seamless user experience.

### 4. **Blockchain Technology for Enhanced Data Security and Transparency**

Blockchain technology will play a critical role in enhancing data security, transparency, and the overall integrity of shipments. It will ensure secure and tamper-proof transaction records, which is particularly important for handling sensitive shipments and providing customers with trusted tracking information.

## **5. Multi-Modal Logistics and Transport Integration**

The CMS will expand to support multi-modal logistics, integrating various transport methods such as air, sea, and ground shipping. This feature will optimize the selection of transport modes based on cost, delivery speed, and geographic location, making the system more adaptable to diverse logistics requirements.

## **6. Integration of Autonomous Delivery Vehicles and Drones**

The rise of autonomous delivery vehicles and drones presents new opportunities for the CMS. The system will evolve to coordinate and optimize these emerging delivery methods, enabling faster, more cost-effective deliveries, particularly for last-mile logistics.

## **7. Advanced Data Analytics and Business Intelligence Tools**

The future CMS will leverage advanced data analytics and business intelligence tools to provide administrators with deeper insights into delivery patterns, customer feedback, and operational performance. These tools will facilitate data-driven decision-making and continuous improvement in service quality.

## **8. Real-Time Notifications and Alerts System**

A robust notification and alert system will be developed to provide real-time updates to customers and administrators regarding key events such as package status changes, delivery delays, and issues. This feature will enhance communication, ensuring that all stakeholders remain informed and engaged throughout the delivery process.

## **9. Payment Gateway Integration**

Future releases will integrate a payment gateway, allowing customers to make seamless online payments for delivery services. This integration will streamline the payment process and improve the overall user experience.

## **10. Mobile Application for Greater Accessibility**

To increase accessibility and convenience, a mobile application version of the system will be developed. Customers and administrators will be able to manage couriers, receive real-time updates, and track shipments directly from their smartphones, allowing for more flexibility in managing deliveries.

## **11. Internationalization and Localization**

The system will support multi-language and multi-currency features, making it adaptable for use in global markets. This will expand the system's reach, catering to international customers and facilitating cross-border shipments.

With these advancements, the Courier Management System: A Web-Based Application to Track and Manage Couriers will not only improve logistics and delivery but will also proactively adapt to emerging trends and technologies. The incorporation of AI, machine learning, IoT, blockchain, and other cutting-edge technologies will ensure that the system remains competitive, efficient, and responsive to the evolving needs of the courier and logistics industry.



## CHAPTER 11

### REFERENCES

#### 11.1 Publications

1. **Name:** "Courier Management Systems: Challenges and Opportunities"  
**Published In:** International Journal of Engineering Research & Technology  
**Description:** Provides insights into the challenges faced by courier management systems, relevant for identifying areas for improvement.
2. **Name:** "Design and Implementation of Courier Management System"  
**Published In:** International Journal of Scientific & Engineering Research  
**Description:** Offers case studies and implementation examples that can guide your system design.
3. **Name:** "Optimizing Courier Management with Web Applications"  
**Author:** Johnson, Emily K.  
**Published In:** Journal of Supply Chain Management, 2021  
**Description:** Explores how web applications are used to optimize courier management systems.
4. **Name:** "Enhancing Customer Satisfaction through Courier Tracking Platforms"  
**Author:** Lee, Michael R.  
**Published In:** Journal of Business and Logistics, 2020  
**Description:** Discusses how tracking systems enhance customer satisfaction in courier services.
5. **Name:** "The Role of IoT in Logistics Management"  
**Author:** Green, Sarah L.  
**Published In:** Journal of Emerging Technologies in Logistics, 2023  
**Description:** Explores the use of IoT in courier logistics.

6. **Name:** "Blockchain for Secure Courier Management"  
**Author:** White, David P.  
**Published In:** Journal of Digital Innovations in Logistics, 2021  
**Description:** Examines how blockchain technology can enhance security in courier management systems.
7. **Name:** "Supply Chain Analytics and Management in Courier Systems"  
**Author:** Manners-Bell, John  
**Published In:** Global Supply Chain Review, 2023  
**Description:** Discusses the role of analytics in optimizing courier system management.
8. **Name:** "Supply Chain Logistics Management"  
**Authors:** Bowersox, Donald J., Closs, David J., & Cooper, M. Bixby  
**Published By:** McGraw-Hill, 2013  
**Description:** A detailed resource for understanding the logistics operations critical to courier systems.
9. **Name:** "Logistics Management and Strategy: Competing Through the Supply Chain"  
**Authors:** Harrison, Alan, and van Hoek, Remko  
**Published By:** Pearson Education, 2014  
**Description:** Focuses on strategies for improving operational efficiency in logistics and courier systems.

## 11.2 Books

1. Logistics and Supply Chain Management by Martin Christopher – A fundamental book to understand the broader context of logistics and its integration with courier management systems.
2. Transportation and Distribution Management by Satish K. Kapoor – Offers valuable insights into the operations of transportation and distribution, key to your courier system.
3. Supply Chain Management: Strategy, Planning, and Operation by Sunil Chopra and Peter Meindl – A comprehensive book covering strategies that could improve the effectiveness of your CMS by aligning it with logistics operations.

4. Operations and Supply Chain Management by F. Robert Jacobs and Richard B. Chase – Discusses integration of supply chain operations, helping to improve the performance of courier systems.
5. E-Commerce Logistics and Fulfilment: Delivering the Goods by Deborah L. Bayles – A great resource for understanding the e-commerce aspect of courier management and its integration with fulfilment processes.
6. Last Mile Logistics and E-Commerce Delivery: Creating Differentiated Customer Experiences by John Manners-Bell – Offers insights into solving last-mile delivery challenges and enhancing the customer experience in courier systems.
7. Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse by Gwynne Richards – Crucial for integrating warehouse management and logistics functions in a CMS.

### 11.3 Websites

1. Django Documentation – <https://docs.djangoproject.com/>
2. MySQL Documentation – <https://dev.mysql.com/doc/>
3. MDN Web Docs – HTML & CSS – <https://developer.mozilla.org/>
4. Google Maps API for Real-Time Tracking – <https://developers.google.com/maps/>
5. Python Programming Tutorials – Real Python – <https://realpython.com/>