

**“ONLINE EXAMINATION AND RESULT SYSTEM”**

A Report submitted under Project-Based Learning

In Partial Fulfillment of the Course Requirements for  
“Web Technologies (22IT104001)”

Submitted By

Pavan Sai Reddy

22102A040238

Under the Guidance of

M.Surya

Department of CSE

**Department of Computer Science and Engineering**

**School of Computing**

**MOHAN BABU UNIVERSITY**

Sree Sainath Nagar, Tirupati – 517 102

**2024-2025**



**MBU MOHAN BABU UNIVERSITY**

### **Vision**

To be a globally respected institution with an innovative and entrepreneurial culture that offers transformative education to advance sustainability and societal good.

### **Mission**

- ❖ Develop industry-focused professionals with a global perspective.
- ❖ Offer academic programs that provide transformative learning experience founded on the spirit of curiosity, innovation, and integrity.
- ❖ Create confluence of research, innovation, and ideation to bring about sustainable and socially relevant enterprises.
- ❖ Uphold high standards of professional ethics leading to harmonious relationship with environment and society.

## **SCHOOL OF COMPUTING**

### **Vision**

To lead the advancement of computer science research and education that has real-world impact and to push the frontiers of innovation in the field.

### **Mission**

- ❖ Instil within our students fundamental computing knowledge, a broad set of skills, and an inquisitive attitude to create innovative solutions to serve industry and community.
- ❖ Provide an experience par excellence with our state-of-the-art research, innovation, and incubation ecosystem to realise our learners' fullest potential.
- ❖ Impart continued education and research support to working professionals in the computing domain to enhance their expertise in the cutting-edge technologies.
- ❖ Inculcate among the computing engineers of tomorrow with a spirit to solve societal challenges.

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **Vision**

To become a Centre of Excellence in Computer Science and its emerging areas by imparting high quality education through teaching, training and research.

### **Mission**

- Imparting quality education in Computer Science and Engineering and emerging areas of IT industry by disseminating knowledge through contemporary curriculum, competent faculty and effective teaching-learning methodologies.
- Nurture research, innovation and entrepreneurial skills among faculty and students to contribute to the needs of industry and society.
- Inculcate professional attitude, ethical and social responsibilities for prospective and promising engineering profession.
- Encourage students to engage in life-long learning by creating awareness of the contemporary developments in Computer Science and Engineering and its emerging areas.

## **B.Tech. Computer Science and Engineering**

### **PROGRAM EDUCATIONAL OBJECTIVES**

After few years of graduation, the graduates of B.Tech. CSE will be:

- PEO1.** Pursuing higher studies in core, specialized or allied areas of Computer Science, or Management.
- PEO2.** Employed in reputed Computer and I.T organizations or Government to have a globally competent professional career in Computer Science and Engineering domain or be successful Entrepreneurs.
- PEO3.** Able to demonstrate effective communication, engage in teamwork, exhibit leadership skills and ethical attitude, and achieve professional advancement through continuing education.

### **PROGRAM OUTCOMES**

On successful completion of the Program, the graduates of B.Tech. CSE Program will be able to:

- PO1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- PO2. Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- PO3. Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- PO4. Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- PO5. Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- PO6. The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- PO7. Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

- PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- PO9. Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- PO11. Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO12. Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

### **PROGRAM SPECIFIC OUTCOMES**

On successful completion of the Program, the graduates of B. Tech. (CSE) program will be able to:

- PSO1.** Apply knowledge of computer science engineering, Use modern tools, techniques and technologies for efficient design and development of computer-based systems for complex engineering problems.
- PSO2.** Design and deploy networked systems using standards and principles, evaluate security measures for complex networks, apply procedures and tools to solve networking issues.
- PSO3.** Develop intelligent systems by applying adaptive algorithms and methodologies for solving problems from inter-disciplinary domains.
- PSO4.** Apply suitable models, tools and techniques to perform data analytics for effective decision making.

Course Code	Course Title	L	T	P	S	C
22IT104001	WEB TECHNOLOGIES	3	-	2	4	5

**COURSE OUTCOMES:** *After successful completion of this course, the students will be able to:*

- CO1.** Demonstrate knowledge on web page design elements, dynamic content and database connection.
- CO2.** Analyze user requirements to develop web applications.
- CO3.** Design client-server applications using web technologies.
- CO4.** Demonstrate problem solving skills to develop enterprise web applications.
- CO5.** Apply HTML, CSS, JavaScript, JQuery, Bootstrap and PHP technologies for device independent web application development.
- CO6.** Apply web technologies to develop interactive, dynamic and scalable web applications for societal needs.

**CO-PO-PSO Mapping Table:**

Course Outcomes	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
<b>CO1</b>	3	3	2	-	-	-	-	-	-	-	-	-	3	2	3	-
<b>CO2</b>	3	3	3	2	2	-	-	-	-	-	-	-	3	2	3	-
<b>CO3</b>	3	3	3	2	2	-	-	-	-	-	-	-	3	2	3	-
<b>CO4</b>	3	3	3	2	2	-	-	-	-	-	-	-	2	2	3	-
<b>CO5</b>	3	2	2	2	2	3	-	-	-	-	-	-	2	2	3	-
<b>CO6</b>				2					3	3						
<b>Course Correlation Mapping</b>	3	3	3	2	2				3	3			3	2	3	

*Correlation Levels: 3: High; 2: Medium;*

*1: Low*



**MBU**  
MOHAN BABU  
UNIVERSITY

**MOHAN BABU UNIVERSITY**

Sree Sainath Nagar, Tirupati 517 102

---

**Department of Computer Science and Engineering**

This is to certify that the Project Entitled

**“ONLINE EXAMINATION AND RESULT SYSTEM”**

Submitted By

Pavan

22102A040238

is the work submitted under Project-Based Learning in Partial Fulfillment of the Course Requirements for “Web Technologies (22IT104001)” during 2024-2025.

**Supervisor:**

M. Surya  
Department of CSE  
School of Computing  
Mohan Babu University  
Tirupati.

**Head:**

Dr. G. Sunitha  
**Professor & Head**  
Department of CSE  
School of Computing  
Mohan Babu University  
Tirupati.

## ACKNOWLEDGEMENTS

First and foremost, I extend my sincere thanks to **Dr. M. Mohan Babu, Chancellor**, for his unwavering support and vision that fosters academic excellence within the institution.

My gratitude also goes to **Mr. Manchu Vishnu, Pro-Chancellor**, for creating an environment that promotes creativity and for his encouragement and commitment to student success.

I am deeply appreciative of **Prof. Nagaraj Ramrao, Vice Chancellor**, whose leadership has created an environment conducive to learning and innovation.

I would like to thank **Dr. K. Saradhi**, Registrar, for his support in creating an environment conducive to academic success.

My sincere thanks to **Dr. B.M. Satish**, Dean of the School of Engineering and Computing, for his valuable support and guidance in all academic matters.

I am also grateful to **Dr. G. Sunitha**, Head of the Department of Computer Science and Engineering, for her valuable insights and support.

Finally, I would like to express my deepest appreciation to my project supervisor, M.Surya, Department of Computer Science and Engineering for continuous guidance, encouragement, and expertise throughout this project.

Thank you all for your support and encouragement.



## Table of Contents

Chapter No.	Title	Page No.
	<b>Abstract</b>	1
<b>1</b>	<b>Introduction</b>	2
	1.1 Problem Statement	2
	1.2 Importance of the Problem	2
	1.3 Objectives	2
	1.4 Scope of the Project	3
<b>2</b>	<b>System Design</b>	4
	2.1 Architecture Diagram	4
	2.2 Module Descriptions	4
	2.3 Database Design	6
<b>3</b>	<b>Implementation</b>	9
	3.1 Tools and Technologies Used	9
	3.2 Front-End Development	9
	3.3 Back-End Development	11
	3.4 Integration	12
<b>4</b>	<b>Testing, Results and Discussion</b>	14
	4.1 Test Cases	14
	4.2 Testing Methods	14
	4.3 Output Screenshots	17
	4.4 Analysis of Results	27
<b>5</b>	<b>Conclusion</b>	28
	5.1 Summary of Findings	28
	5.2 Future Enhancements	28
<b>6</b>	<b>Appendix</b>	29
	6.1 Code Snippets	29

# **ABSTRACT**

The "Courier Management System" is a web-based application designed to transform the traditional parcel delivery process, aiming to enhance efficiency and streamline logistics operations. By transitioning from manual tracking to a digital platform, this system not only modernizes the delivery method but also reduces the administrative burden on logistics personnel.

This comprehensive system addresses the diverse needs of courier services by facilitating the seamless management of parcels from dispatch to delivery. Through the platform, users gain the convenience of tracking shipments in real-time, allowing for accurate updates on parcel locations and delivery status without the constraints of physical presence at a logistics center. This not only improves accessibility but also enhances flexibility in managing deliveries according to individual or business preferences.

One of the key features of the system is its real-time tracking capability. Customers can instantly access their shipment status, eliminating the uncertainty and waiting period typically associated with manual tracking processes. This immediate feedback mechanism improves user experience by providing timely insights into delivery progress and estimated arrival times.

Tasks such as order placement, route optimization, delivery scheduling, and customer feedback management are streamlined, allowing logistics personnel to focus more on operational excellence rather than administrative tasks.

**Keywords:** Courier Management, Parcel Tracking, Web-based Platform, Logistics Optimization, Administrative Workload, User Convenience, Real-time Tracking, User Experience, Security, Scalability, Performance, Digital Transformation, Logistics Technology.

# **1.Introduction**

## **1.1 Problem Statement:**

Courier Management System is a web-based platform that facilitates the management of parcel deliveries online, either through the internet or an intranet using computer systems. The main goal of this Courier Management System is to effectively streamline the delivery process through a fully automated system that not only reduces the time required for handling logistics but also provides fast and accurate tracking and management of shipments. This system aims to enhance operational efficiency, improve user experience, and enable real-time visibility into parcel status, ultimately transforming the way courier services are delivered and managed.

## **1.2 Importance of the Problem:**

The Courier Management System is an electronic application designed to streamline the logistics and delivery processes for courier services. This framework helps organizations efficiently manage the dispatch and tracking of parcels by providing various functionalities, including shipment creation, real-time tracking, and delivery status updates. In recent years, the Courier Management System has gained significant traction due to its speed, accuracy, and the increasing demand for contactless delivery methods, especially during the COVID-19 pandemic.

Almost all courier companies today are adopting online systems to manage their deliveries, which significantly reduces the time and resources required to handle logistics. This system minimizes the need for manual intervention, allowing for quicker processing of shipments and faster updates on delivery statuses. Additionally, it helps to reduce paper usage by digitizing processes that would traditionally require paperwork.

The Courier Management System is crucial for courier businesses and logistics providers as it optimizes their operations and enhances customer experience by providing real-time visibility into parcel statuses. The system aids in monitoring deliveries, ensuring timely updates, and improving overall operational efficiency. Its implementation is especially beneficial for courier services, as it simplifies the management of shipments and allows for quicker, more accurate results in logistics tracking and delivery confirmations.

## **1.3 Objectives:**

1. **\*\*Streamline the Delivery Process\*\***: An online courier management system will minimize the time-consuming task of manually tracking and processing delivery requests.
2. **\*\*Reduce Paperwork\*\***: As an integrated courier management system, it will significantly decrease the reliance on paper documentation, leading to more efficient operations.
3. **\*\*Enable Staff to Manage Shipments and Tracking\*\***: The system will allow staff to create and manage shipment orders and tracking information seamlessly.

4. **\*\*Facilitate Automatic and Manual Processing\*\***: The system will support automatic updates and manual adjustments, which can be recorded for each shipment.

5. **\*\*Automate Tracking and Notifications\*\***: Responses and tracking updates will be automatically generated, ensuring timely communication with customers.

6. **\*\*Improve Time Efficiency\*\***: The courier management system will significantly reduce the time spent on logistics and administrative tasks, allowing for faster deliveries and better customer service.

### **1.4 Scope of the Project:**

"Courier Management System" project is to develop a web-based platform for managing courier operations efficiently. It allows customers to register, book shipments, and track their packages in real-time. Administrators can create and manage delivery options, monitor shipment progress, and handle customer inquiries. The system automates tracking updates and notification alerts to reduce operational workload and stores data securely in a MySQL database. It ensures secure access for users and supports scalability for future expansion. The platform aims to streamline courier processes, making them accessible, fast, and reliable.

## **Features**

### **User Side**

- **\*\*New User Registration\*\*** Users can register for the Courier Management System online.
- **\*\*User Login\*\*** Users can log in to the system using their Email ID and Password.
- **\*\*Profile Management\*\*** Users can view and update their profile details.
- **\*\*Password Management\*\*** Users can change their password securely.
- **\*\*Available Courier Services\*\*** Users can view a list of available courier services.
- **\*\*Create Shipment\*\*** Users can initiate a shipment by selecting the desired courier service from the available list.
- **\*\*Track Shipment\*\*** Users can track the status of their shipments in real-time.
- **\*\*View Shipment History\*\*** Users can view their past shipment history and details.
- **\*\*Analyze Shipment Data\*\*** Users can analyze their shipment data for better decision-making (e.g., cost estimates, delivery times).

### **Admin Side**

- \*\*Admin Can View All Courier Services**
- \*\*Admin can access and view a comprehensive list of all courier services available in the system.**
- \*\*Admin Can View All User Data\*\***

## 2 System design

### 21. Architecture Diagram

#### Data Flow Diagrams (DFD)

DFD, or Data Flow Diagrams, are used to graphically represent the flow of data within a courier management system. These diagrams illustrate the functions and processes involved in capturing, manipulating, storing, and distributing data between the system and its environment, as well as among the various components of the system. The visual representation of DFDs serves as an effective communication tool between users and system designers.

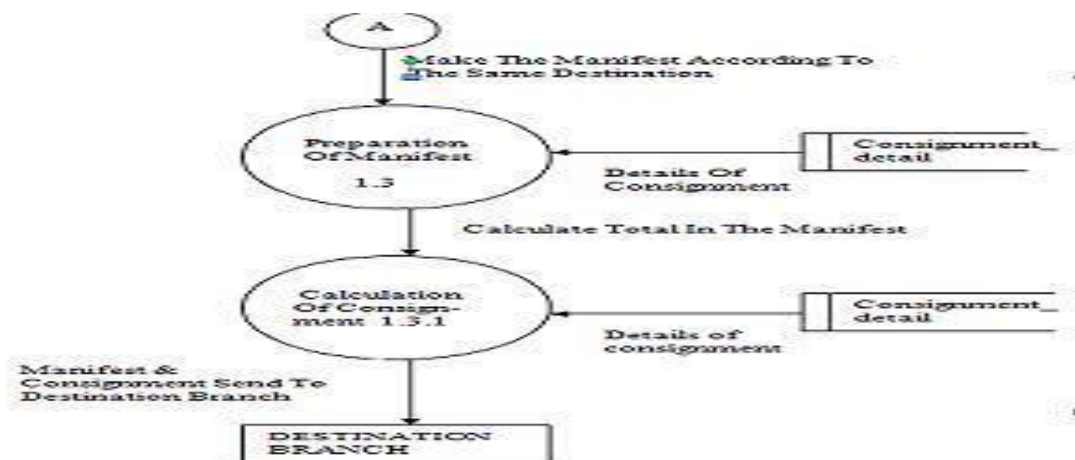
The structure of a DFD enables the depiction of a broad overview of the courier management system, which can then be expanded into a hierarchy of detailed diagrams. In software engineering, DFDs can be created to represent the system at various levels of abstraction, outlining both information and functional elements..

### 2.2 Module Descriptions:

#### Levels in Data Flow Diagrams (DFD)

##### 0-level DFD:

It is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.

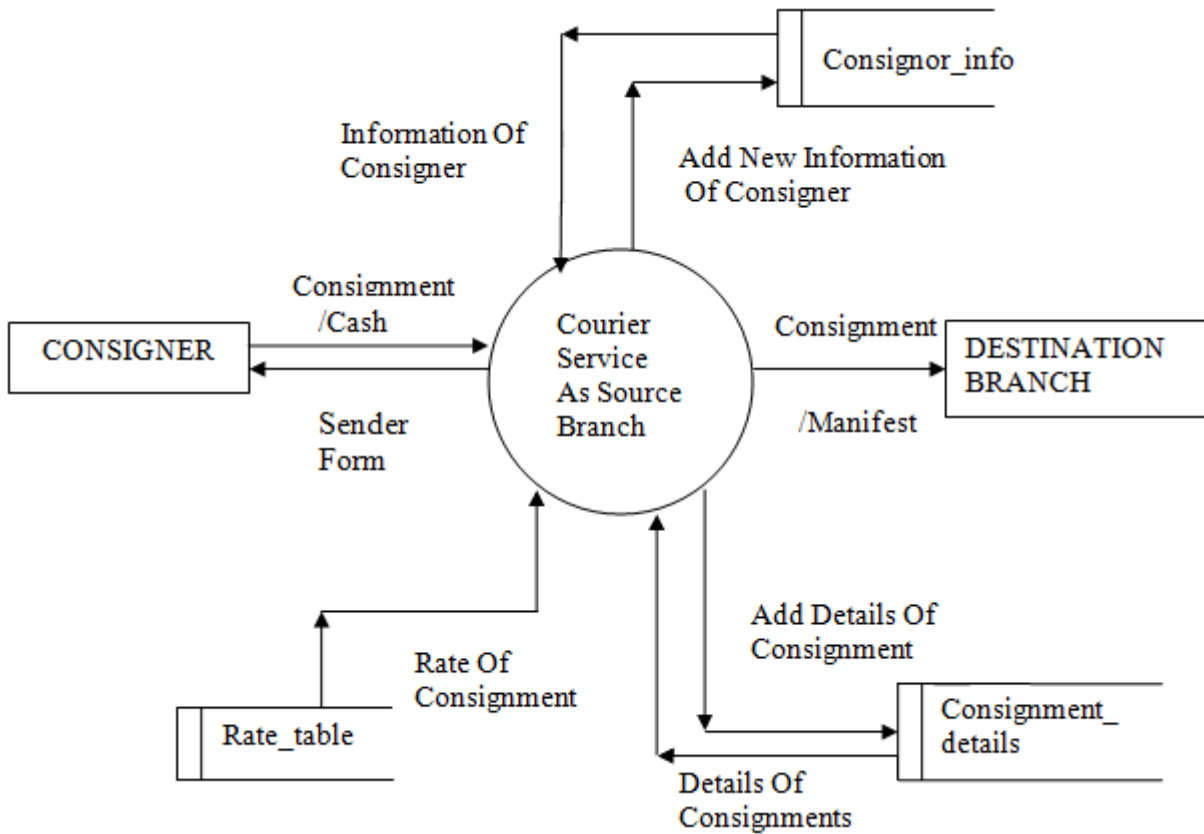


**courier management system**

**Fig**

## 1-level DFD:

In a 1-level Data Flow Diagram (DFD) for a Courier Management System, the high-level process from the context diagram is decomposed into multiple subprocesses. Each subprocess represents a specific function of the system, highlighting the interactions and data flows to and from external entities.



Fig

This DFD illustrates how user data flows through an online examination system and transitions into a courier management process for result delivery. It helps visualize the interaction between both systems and ensures clarity regarding how data is processed and managed across different functions.

## ER Diagram

The ER diagram for a Courier Management System provides a structured framework for understanding how different entities within the system are related. It outlines the essential components and relationships necessary for database design, serving as a guide for implementing a relational database to support the functionality of the courier management system.

### Components of the ER Diagram for a Courier Management System

This model is based on three basic concepts: Entities, Attributes, Relationships.

#### Entities:

Entities represent real-world objects or concepts relevant to the Courier Management System, such as individuals, requests, and packages. Each entity is depicted as a rectangle in the ER diagram, with its name written inside. Key entities in this system may include:

**Customer:** Represents individuals or businesses requesting courier services.

**Courier:** Represents individuals or services responsible for delivering packages.

**DeliveryRequest:** Represents requests made by customers for the delivery of packages.

**Package:** Represents items being picked up and delivered.

**Tracking:** Represents the various status updates associated with a delivery request.

### (ER) MODEL FOR A COURIER MANAGEMENT SYSTEM

The Entity-Relationship (ER) model is a high-level conceptual data model diagram that is especially useful in the context of a Courier Management System. ER modeling aids in systematically analyzing data requirements to produce a well-designed database that effectively manages the complex interactions between various entities involved in courier services.

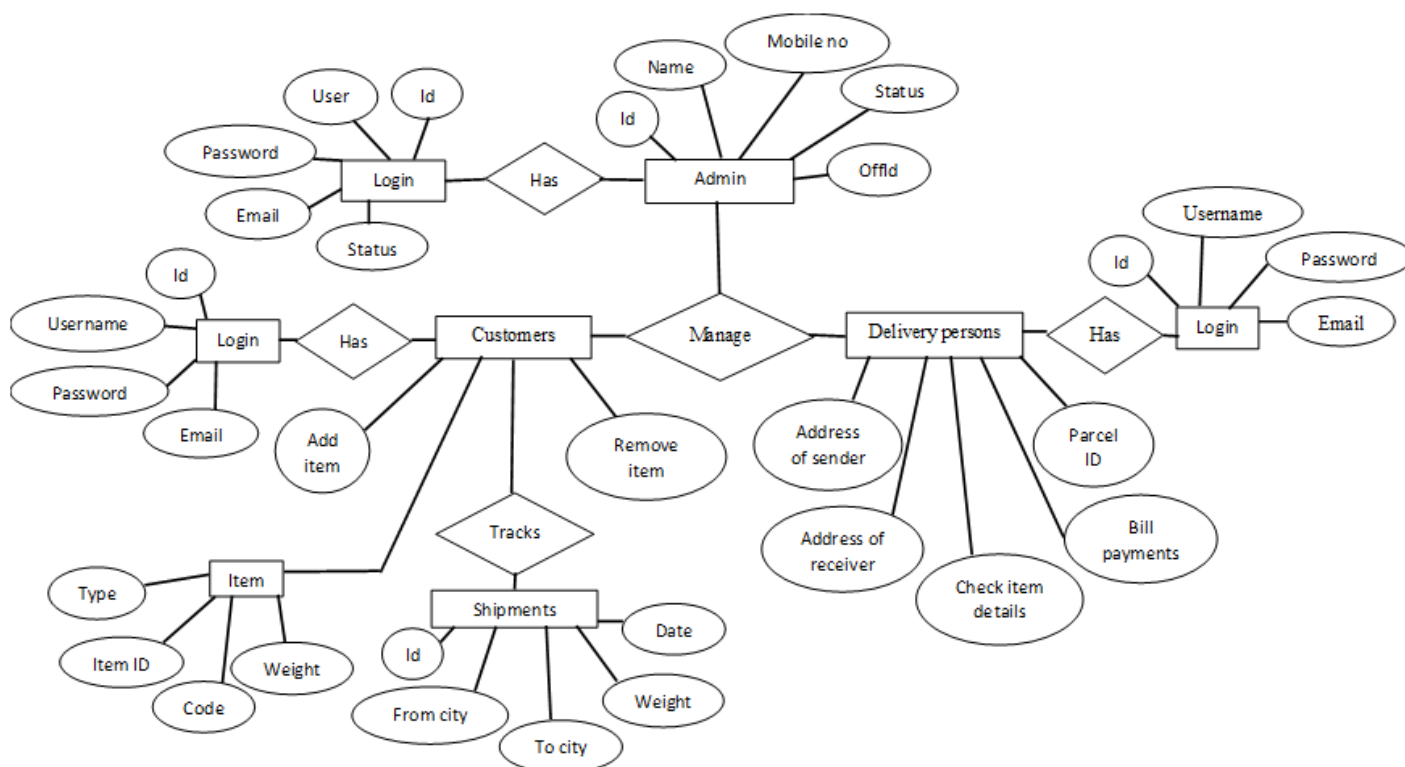
The ER model represents real-world entities—such as customers, couriers, delivery requests, and packages—and the relationships between these entities. Creating an ER diagram before the implementation of the database is considered a best practice to ensure that the structure is logical and efficient.

Furthermore, the Entity-Relationship model supports hierarchical organization and

normalization principles, enabling the representation of complex relationships that may arise in the courier domain. It accommodates extended notations for intricate scenarios, such as multiple package deliveries or tracking statuses, and ensures the semantic modeling of concepts relevant to the real world.

Overall, the ER model provides a versatile and intuitive framework for conceptualizing the data architecture of a Courier Management System. It lays the groundwork for effective database design and management, ensuring that all data elements and their relationships are thoroughly defined and understood.

### ER Diagram



**Fig : Online Examination and Result System ER Diagram**



## 3 Implementation

### 3.3 Tools and Technologies Used

The main objective of the Courier Management System is to facilitate the efficient management of parcel deliveries and logistics for both businesses and customers. This system enables organizations to handle a large volume of shipments seamlessly and in an automated manner, significantly reducing the time and labor involved in traditional courier management..

#### Software Requirements

 Front End Tools

3.3.1.1 HTML

3.3.1.2 CSS

3.3.1.3 JAVA SCRIPT

3.3.1.4 PHP

 Back End Tools

3.3.2.1 MySQL

### 3.4 Front-End Development

The frontend design for the online examination and result system utilizes a combination of HTML, CSS, JavaScript, and PHP integrated development environment to create a robust and user-friendly interface.

#### HTML (Hypertext Markup Language):

HTML serves as the standard markup language for creating the structure and content of web pages and applications within the online examination system. It defines elements such as headings, paragraphs, lists, links, images, forms, and other essential components necessary for presenting exam questions, results, and navigation.

#### CSS (Cascading Style Sheets):

CSS plays a crucial role in styling HTML elements and defining the layout of web pages within the online examination system. It allows developers to control the appearance of exam interfaces by specifying properties such as colors, fonts, margins, padding, and positioning. CSS ensures consistency in design and enhances the visual presentation of exam questions, results, and user interfaces.

### **JavaScript:**

JavaScript enables interactive and dynamic behavior on exam pages, enhancing user engagement and functionality within the online examination system. It facilitates features such as form validation, real-time updates, interactive elements, and dynamic content loading. JavaScript ensures a seamless and interactive user experience during exam taking, result viewing, and navigation.

### **PHP (Hypertext Preprocessor):**

PHP serves as the server-side scripting language used for backend processing, database interactions, and dynamic content generation within the online examination system. It enables functionalities such as user authentication, exam creation, result processing, and database management. PHP integrates seamlessly with HTML, allowing developers to embed dynamic content, execute server-side scripts, and generate personalized exam interfaces based on user interactions.

### **Integration:**

The integration of HTML, CSS, JavaScript, and PHP within the online examination system provides a comprehensive toolkit for frontend development. HTML defines the structure and content of exam interfaces, CSS styles the presentation and layout, JavaScript adds interactivity and dynamic behavior, while PHP handles server-side processing and dynamic content generation. This integrated approach ensures the creation of dynamic, responsive, and user-friendly exam interfaces that enhance the overall user experience.


### **Benefits:**

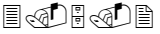
By leveraging a combination of HTML, CSS, JavaScript, and PHP, the online examination system offers a robust and versatile frontend design. HTML provides the foundational structure and content, CSS enhances the visual presentation and layout, JavaScript adds interactivity and dynamic behavior, and PHP enables server-side processing and dynamic content generation. This integrated approach ensures the development of a feature-rich, responsive, and user-friendly online examination and result system

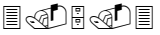
### **3.5 Back-End Development:**

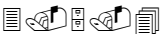
The back-end of the online examination and result system will be developed and managed using MySQL, a powerful open-source relational database management system. Here's a detailed breakdown of MySQL's features and its role within the system:

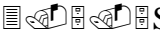
#### **MySQL (Relational Database Management System):**

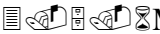
 MySQL is utilized for the permanent storage of records within the system, ensuring data integrity, reliability, and scalability.

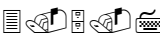
 It is the most popular open-source SQL database management system, developed, distributed, and supported by Oracle Corporation.

 MySQL databases are relational, storing data in separate tables organized into physical files optimized for speed.

 The logical model includes objects such as databases, tables, views, rows, and columns, offering a flexible programming environment for managing relationships between different data fields.





 SQL (Structured Query Language) is used to interact with MySQL databases, providing a common standardized language for accessing and manipulating data.





 MySQL software is open-source, allowing anyone to download, use, and modify the software without any cost.

 MySQL Server works in client/server or embedded systems, providing a multithreaded SQL server that supports various client programs, libraries, administrative tools, and application programming interfaces (APIs).





## Integration





### **courier\_management\_system\_data:**

    This database contains tables for storing shipment details and tracking information.





    Records in this database represent individual shipments, with details associated with each shipment, including sender and recipient information.





### **courier\_user\_authentication\_data:**

    This database manages authentication and login data for system users, including administrators and regular customers.

    Tables within this database store records for admin login credentials and customer login credentials.





### **courier\_tracking\_and\_delivery\_results:**





    This database stores the results of deliveries made through the system, including status updates and delivery confirmations.





    Tables within this database store records for delivery results, including user IDs, shipment IDs, and delivery statuses.

## 3.6 Integration





### **PHP Integration:**

    PHP scripts serve as the intermediary between the frontend and MySQL databases, executing SQL queries to interact with the data.

    PHP scripts can retrieve data from MySQL databases to dynamically generate HTML content, such as displaying exam questions or user results.

    Additionally, PHP scripts handle form submissions from HTML pages, processing user input and storing data into MySQL databases.

### **JavaScript Integration:**

    JavaScript enhances the user experience by providing interactive features and dynamic content manipulation.

📄👉⌚👉📄JavaScript can make asynchronous HTTP requests (AJAX) to PHP scripts, which then

interact with MySQL databases to fetch data without reloading the entire webpage.



This allows for real-time updates, dynamic content loading, and seamless user interactions, enhancing the responsiveness and usability of the system.

## HTML Forms and Data Submission:



HTML forms are used to collect user input, such as exam answers or login credentials.



When a user submits a form, the data is sent to PHP scripts for processing.



PHP scripts execute SQL queries to insert, update, or delete data in MySQL databases based on the form submission.

## Server-side Data Processing:



MySQL databases store and manage the system's data, including shipment records, user credentials, and delivery results.



PHP scripts execute SQL queries to retrieve relevant data from MySQL databases based on user interactions and system requirements (e.g., retrieving all shipments for a user or checking the status of a specific shipment).



Data retrieved from MySQL databases is processed, formatted, and sent back to the frontend for display or further interaction, ensuring users can easily track shipments and manage their accounts.

## 4 Testing, Results and Discussion

### 4.3 Test cases

#### 4.4


4.5 Unit testing is a crucial phase in the development of a courier management system, where individual units or components of the software are rigorously tested to ensure they perform as intended. In this context, a unit can be defined as the smallest testable part of the courier management software, which typically takes one or a few inputs and produces a single output.


#### 4.6

4.7 In the realm of procedural programming, a unit in the courier management system might refer to specific functions or procedures, such as those managing shipment tracking, user authentication, or delivery status updates. In object-oriented programming, the smallest unit can be identified as methods within classes — for instance, a method responsible for calculating shipping costs within a shipment class or a method for validating user credentials within a user authentication class.

### 4.8 Testing Methods:

 **Black Box Testing** - Using which the user interface, input and output are tested.

 **White Box Testing** - used to test each one of those functions behaviors is tested.

 **Gray Box Testing** - Used to execute tests, risks and assessment methods.

### Integration Testing


Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. It is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between

these software modules when they are Integrated Integration Testing focuses on checking data communication amongst these modules.

## Approaches, Strategies, Methodologies of Integration Testing

Software Engineering defines variety of strategies to execute Integration testing, viz.

 Big Bang Approach

 Incremental Approach: which is further divided into the following

- Top-Down Approach
- Bottom-Up Approach
- Sandwich Approach - Combination of Top Down and Bottom Up

## System Testing

**System Testing** is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computerbased system.

**System Testing** is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and nonfunctional



testing. System Testing is a black-box testing. It is performed after the integration testing and before the acceptance testing.

## **Types of System Testing**

**Performance Testing:** Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.

**Load Testing:** Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.

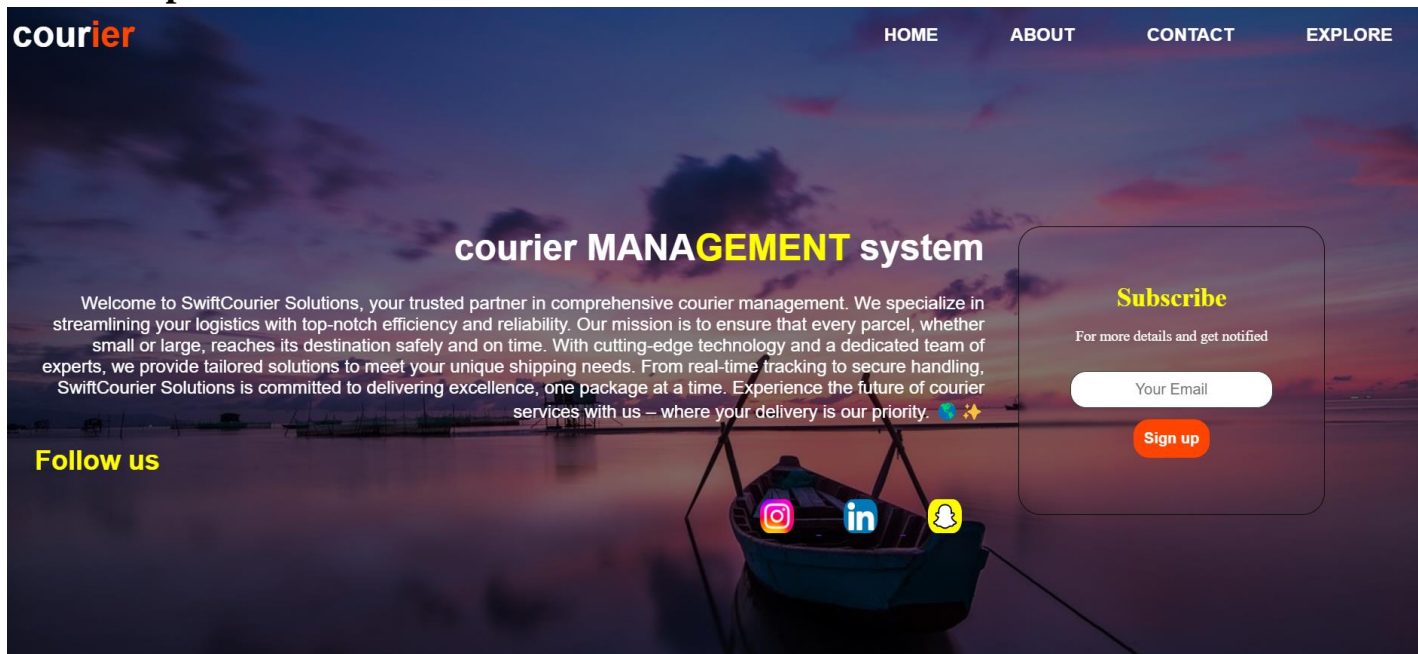
**Stress Testing:** Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.

**Scalability Testing:** Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.

## **Secure Authentication and Access Control System for Student and Admin Accounts**

In our project we are testing whether the student is authorized or not. When student want to register themselves, it will check whether the user name entered is the same as in the database. If authorized student has login, they can register themselves. If not authorized then, back to main page. We are also checking the admin login whether it is authorized or not. Only authorized admin can login. No one can register themselves as admin. Admin can change username and password in admin page only. Member login is also verified from the database values.

## 4.9 Output Screens:



Fig

Fig : courier management

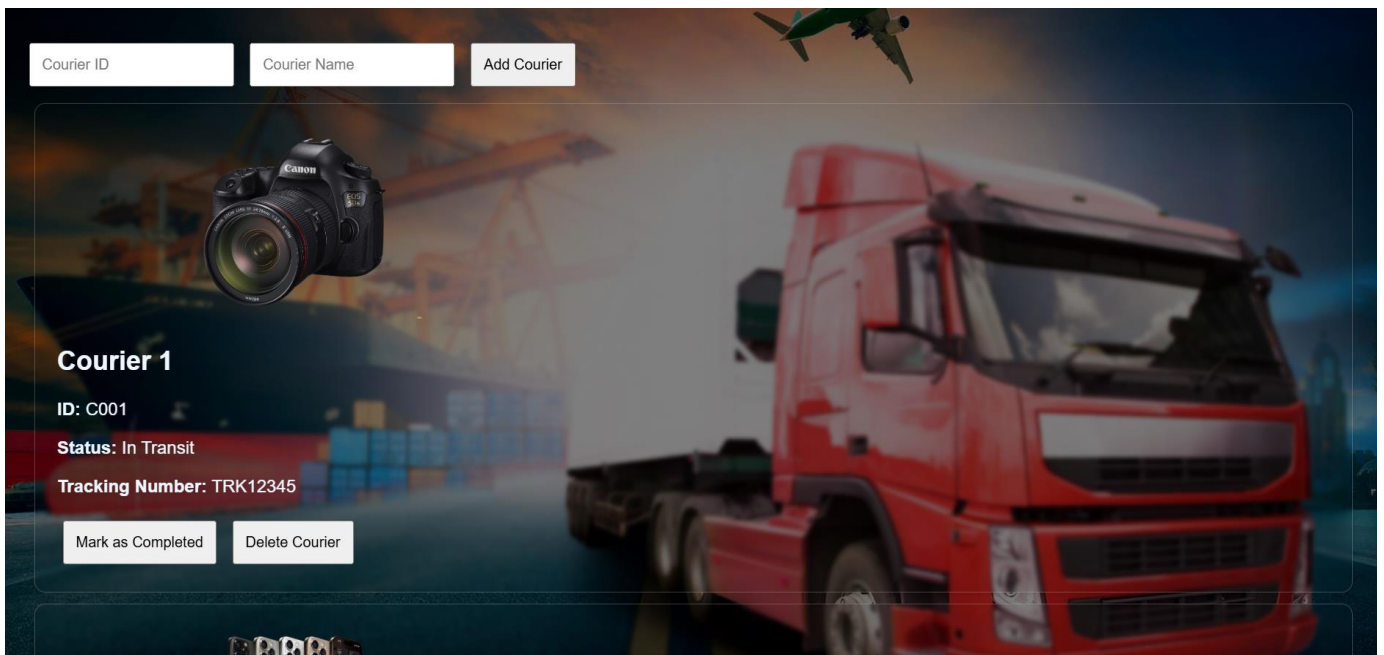


Fig : Add courier

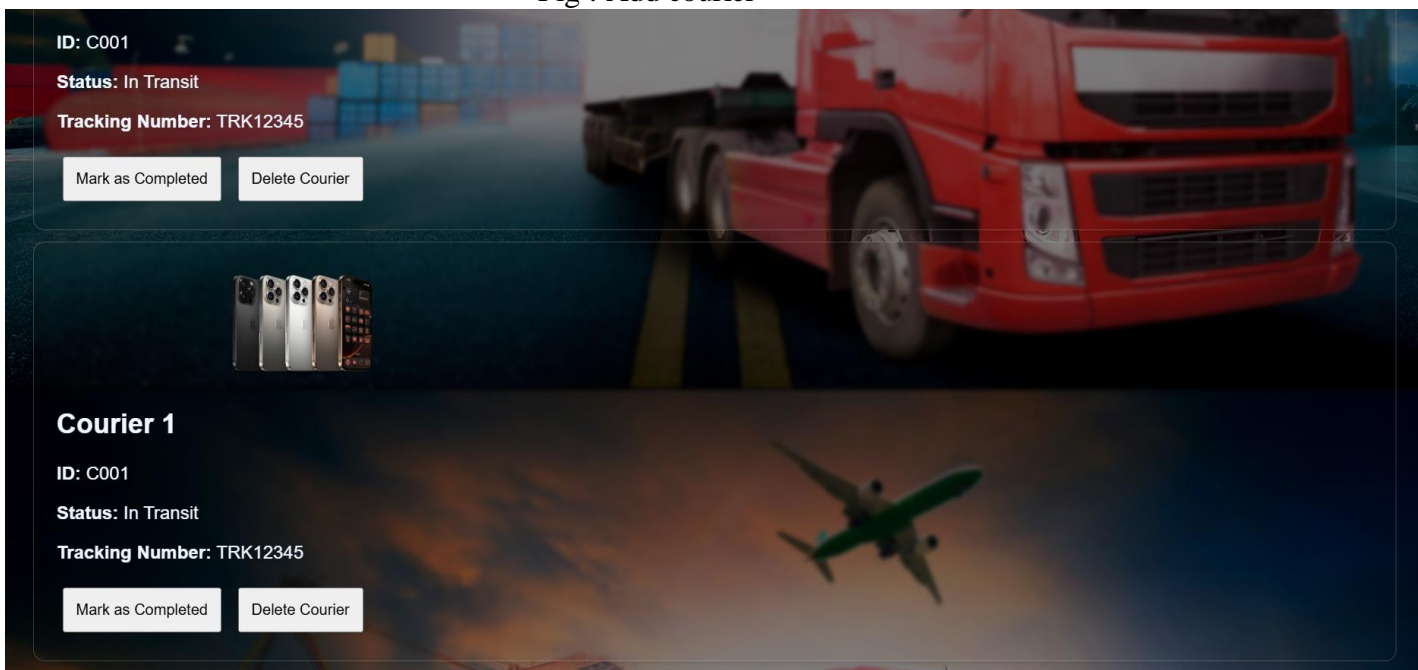


Fig : delete courier

#### **4.10 analysis of results:**

The “Courier Management System” provides both customers and administrators with immediate insights into shipment performance. Upon completion of a delivery, customers receive instant notifications with tracking information and delivery status, allowing them to understand the current location of their parcels and any potential delays. This feature enhances customer satisfaction by offering timely updates and transparency.

For administrators, the system compiles comprehensive data on delivery performance across different routes and time periods. It enables them to monitor efficiency, identify trends in delivery times, and make informed decisions for operational adjustments or resource allocation. Overall, the results analysis within the Courier Management System enhances operational transparency, encourages continuous improvement, and assists in data-driven logistics planning.

## 5 Conclusion

### 5.3 summary of findings:

The **Courier Management System** significantly enhances the efficiency and effectiveness of shipping and delivery processes by transitioning from outdated manual methods to an automated platform. Key findings indicate that the system streamlines order processing and reduces errors, which results in faster delivery times and improved accuracy in tracking shipments. Customers benefit from real-time tracking updates and notifications, which not only enhance their experience but also foster greater trust in the service. Additionally, the secure database framework ensures data integrity and enables efficient retrieval of shipping records. Overall, the Courier Management System not only improves the user experience for both customers and administrators but also provides a scalable and reliable solution that meets the demands of modern logistics operations.

### Future Enhancements

Future enhancements for the **Courier Management System** could center on expanding features to improve accessibility, security, and adaptability. Firstly, incorporating support for advanced tracking features, such as geo-fencing alerts or enhanced delivery scheduling options, could improve customer engagement and satisfaction. Implementing AI-driven predictive analytics could optimize route planning and resource allocation, ensuring more efficient delivery processes.

Moreover, introducing a mobile app could enable customers and drivers to manage deliveries more conveniently on various devices. This app could include features like push notifications for delivery status updates and an interactive interface for real-time communication. To bolster security, techniques such as blockchain technology could be integrated to provide a tamper-proof record of shipments.

Furthermore, incorporating feedback mechanisms within the system could facilitate continuous improvement based on user insights, leading to enhanced service offerings. Enhanced multi-language support would also increase accessibility for a diverse customer base, promoting inclusivity in various geographical regions. Overall, these future enhancements could significantly improve system functionality and user satisfaction, ensuring the Courier Management System remains aligned with evolving logistical needs.

## 6 Appendix

### 6.3 Code Snippets:

Snippets of codes include are:

1. **Admin and User Login**
  2. **Home Page**
  3. **User Sign Up**
  4. **Reset Password**
  5. **Viewing Courier**
- 

### HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Courier Management</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1 style="color: rgba(255, 250, 244, 0.918);">Courier Management</h1>
  <div id="courier-form">
    <h2>Add Courier</h2>
    <input type="text" id="courier-id" placeholder="Courier ID">
    <input type="text" id="courier-name" placeholder="Courier Name">
    <button onclick="addCourier()">Add Courier</button>
  </div>
  <div class="courier">
    
    <h2>Courier 1</h2>
    <p><strong>ID:</strong> C001</p>
```

```

    <p><strong>Status:</strong> In Transit</p>
    <p><strong>Tracking Number:</strong> TRK12345</p>
    <button onclick="markCompleted()">Mark as Completed</button>
    <button onclick="deleteCourier()">Delete Courier</button>
</div>
<div class="courier">
    
    <h2>Courier 1</h2>
    <p><strong>ID:</strong> C001</p>
    <p><strong>Status:</strong> In Transit</p>
    <p><strong>Tracking Number:</strong> TRK12345</p>
    <button onclick="markCompleted()">Mark as Completed</button>
    <button onclick="deleteCourier()">Delete Courier</button>
</div>
<div id="courier"></div>
<script src="script.js"></script>
</body>
</html>

```

---

## CSS

```
/* Basic styling */
```

```

body {
    font-family: Arial, sans-serif;
    margin: 20px;
    background-image: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),url(download.jpg);
    background-repeat: repeat;
    background-size: 100%;
    filter: brightness(-30px);
}

```

```

input, button {
    padding: 10px;
}

```

```

margin: 5px;

}

@media (max-width: 768px) {
  /* Adjust layout for smaller screens */
  .container {
    grid-template-columns: 1fr;
  }
  /* Other adjustments for smaller screens */
}

.courier {

  border: 1px solid #cccccc46;
  border-radius: 10px;
  color: aliceblue;
  padding: 20px;
  margin: 10px;
  background-color: #ffffff00;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  backdrop-filter: blur(2px);
}

.courier img {
  max-width: 20%;
  height: 20%;
}

h1{
  text-align: center;
  position: relative;
  top:30px;
  font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
  font-size:40px;
}

```



---

## JS

```
function addCourier() {  
    const id = document.getElementById('courier-id').value;  
    const name = document.getElementById('courier-name').value;  
  
    // Create a new courier object  
    const courier = {  
        id: id,  
        name: name,  
        status: 'Pending'  
    };  
  
    // Store the courier in local storage  
    let couriers = JSON.parse(localStorage.getItem('couriers')) || [];  
    couriers.push(courier);  
    localStorage.setItem('couriers', JSON.stringify(couriers));  
  
    // Display the courier in the list  
    const courierList = document.getElementById('courier-list');  
    const li = document.createElement('li');  
    li.textContent = `${id} - ${name} - ${courier.status}`;  
    courierList.appendChild(li);  
  
    // Clear the form fields  
    document.getElementById('courier-id').value = "";  
    document.getElementById('courier-name').value = "";  
}  
  
function deleteCourier() {  
    // Remove the courier from the DOM  
    const courierBox = document.querySelector('.courier');  
    courierBox.remove();  
}
```

```

// Remove the courier from local storage (if used)
const couriers = JSON.parse(localStorage.getItem('couriers')) || [];
const index = couriers.findIndex(courier => courier.id === 'C001'); // Replace 'C001' with the actual ID
if (index !== -1) {
  couriers.splice(index, 1);
  localStorage.setItem('couriers', JSON.stringify(couriers));
}
}

// Load existing couriers from local storage on page load
window.onload = function() {
  const couriers = JSON.parse(localStorage.getItem('couriers')) || [];
  const courierList = document.getElementById('courier-list');
  couriers.forEach(courier => {
    const li = document.createElement('li');
    li.textContent = `${courier.id} - ${courier.name} - ${courier.status}`;
    courierList.appendChild(li);
  });
};

```

---