PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering
B.TECH. (ECE) VI SEMESTER
UE20CS301 – Database Management Systems (Minors)

MINI PROJECT REPORT ON
“Courier Management System.”

Faculty in charge

ASSISTANT PROF Bhargavi Mokashi

SUBMITTED BY

Name	SRN
Neelan.S.Patil	PES1UG20EC120
Naman C R	PES1UG20EC119
Nehal M B	PES1UG20EC122
Nirmiti N	PES1UG20EC128

January – May 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BENGALURU – 560100, KARNATAKA, INDIA

CONTENTS :

Introduction::

The report aims to provide an overview and analysis of the Courier Management System (CMS), a software application designed to streamline and automate various processes involved in managing and tracking courier services. The CMS plays a crucial role in enhancing the efficiency, accuracy, and reliability of courier operations, enabling organizations to deliver packages and parcels in a timely manner.

This report will dive into the key functionalities, benefits, and challenges associated with the Courier Management System. It will explore the system's core components, such as customer management, parcel tracking, delivery person management, and administrative features. Additionally, the report will highlight the significance of CMS in improving customer satisfaction, optimizing delivery routes, and enhancing overall operational efficiency.

Furthermore, the report will examine the integration of the CMS with other essential systems, such as inventory management, billing, and reporting. It will explore how these integrations facilitate seamless coordination between different departments, leading to better resource utilization and cost-effective operations.

The conclusion of the report will provide a summary of the key findings and recommendations for organizations considering the implementation or enhancement of a Courier Management System. It will emphasize the importance of thorough planning, stakeholder engagement, and continuous system evaluation to maximize the benefits of CMS implementation.

Overall, this report aims to provide valuable insights into the Courier Management System, its impact on courier operations, and the potential for future advancements in the industry. It serves as a comprehensive guide for decision-makers, IT professionals, and stakeholders involved in courier services seeking to optimize their operations and deliver exceptional customer experiences.

Objective:

- To create database for Courier Management System(CMS).
- To perform operations on the Database created.
- To connect the backend Database of Courier Management System(CMS) to the frontend using Streamlit.

Methodology:

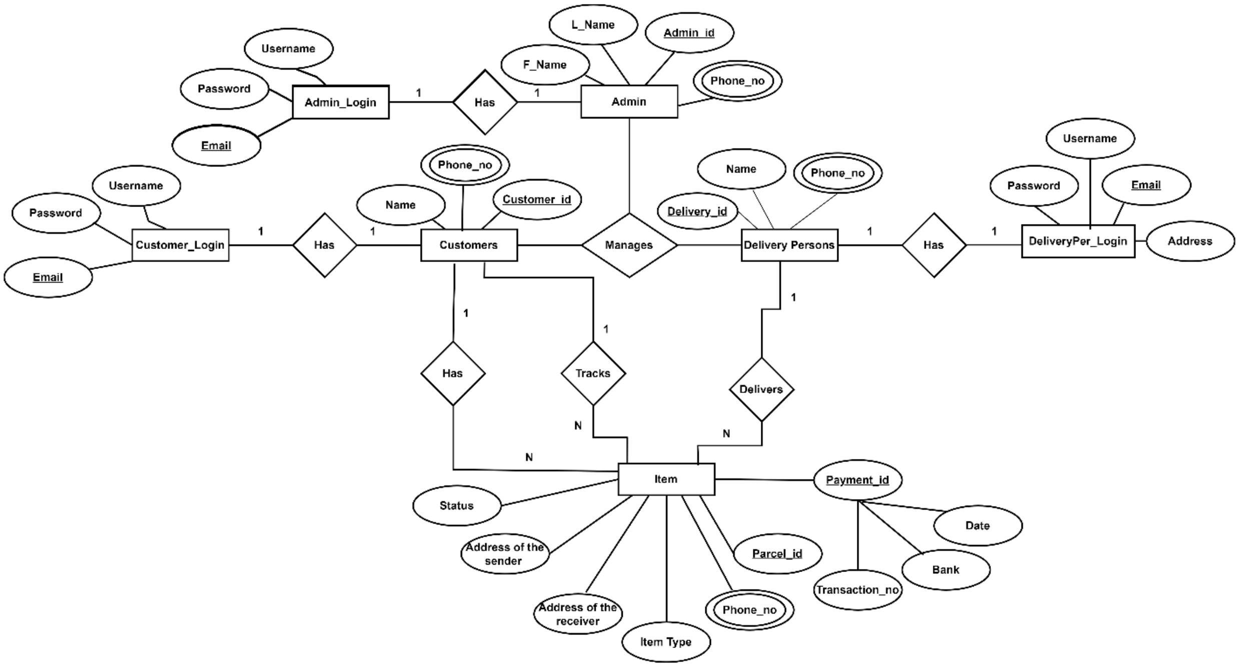
- Create a ER Diagram .
- Convert the created ER Diagram into Relationship Schema.
- Using the Relationship schema, First create the database.
- Create a Database with the Tables taking reference from the Relationship schema.
- Once the Tables are created with the required attributes now populate it with some sample data.
- Perform the required operations to obtain the correct result and operations to retrieve data from the Tables.
- Once the Database gives proper result for the queries now integrate it with the Frontend.
- Connect the database created with the Frontend.
- Test the Frontend and Backend thoroughly .

Tools Used:

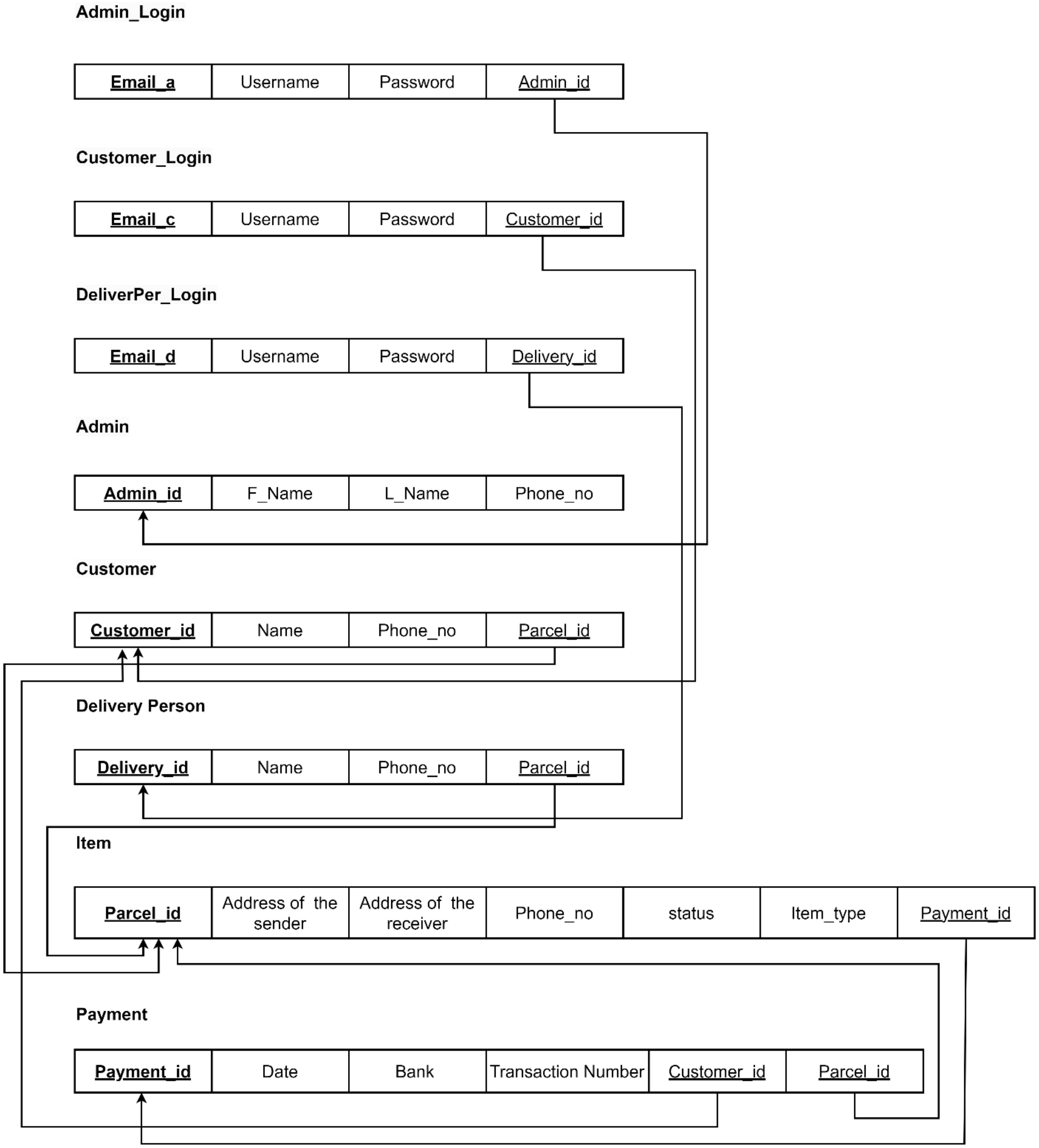
- For ER Diagram and Relationship Schema - Diagrams.net
- For Database Creation and to perform Operations – MySQL
- For Connecting Backend to Frontend – Streamlit Library
- Python IDE – PyCharm Community Edition

Implementation:

- ER diagram:



- Relationship schema :



Admin_Login

Email_a	Username	Password	Admin_id

Customer_Login

Email_c	Username	Password	Customer_id

DeliverPer_Login

Email_d	Username	Password	Delivery_id

Admin

Admin_id	F_Name	L_Name	Phone_no

Customer

Customer_id	Name	Phone_no	Parcel_id

Delivery Person

Delivery_id	Name	Phone_no	Parcel_id

Item

Parcel_id	Address of the sender	Address of the receiver	Phone_no	status	Item_type	Payment_id

Payment

Payment_id	Date	Bank	Transaction Number	Customer_id	Parcel_id

- Database Creation and Tables:

  SQL Queries for Database creation:

```
CREATE DATABASE cms;
```

  SQL Queries for Tables creation:

```
CREATE TABLE Admin ( F_Name VARCHAR(255) NOT NULL ,L_Name
VARCHAR(255) NOT NULL, Phone_no VARCHAR(10) NOT NULL
,Admin_id INT NOT NULL,PRIMARY KEY(Admin_id));
```

```
CREATE TABLE Item (Parcel_id INT PRIMARY KEY, Sender_Address
VARCHAR(255) NOT NULL,Receiver_Address VARCHAR(255) NOT NULL,
Phone_no VARCHAR(20) NOT NULL, Status VARCHAR(255), Item_type
VARCHAR(255),Payment_id INT,FOREIGN KEY
(Payment_id) REFERENCES Payment(Payment_id));
```

```
CREATE TABLE Customer (Name VARCHAR(255) NOT NULL, Phone_no
VARCHAR(10) NOT NULL,Customer_id INT NOT NULL,Parcel_id INT
,PRIMARY KEY(Customer_id),FOREIGN KEY (Parcel_id)
REFERENCES item(Parcel_id));
```

```
CREATE TABLE Payment (Payment_id INT NOT NULL Date DATE NOT
NULL,Bank VARCHAR(255), Transaction_Number VARCHAR(255),
Customer_id INT, Parcel_id INT, PRIMARY KEY(Payment_id), FOREIGN
KEY (Customer_id) REFERENCES Customer(Customer_id),
FOREIGN KEY (Parcel_id) REFERENCES Item(Parcel_id));
```

```
CREATE TABLE Delivery_Person (Name VARCHAR(255) NOT NULL, Phone_no
VARCHAR(10) NOT NULL, Delivery_id INT NOT NULL,
Parcel_id INT,PRIMARY KEY(Delivery_id), FOREIGN KEY (Parcel_id)
REFERENCES item(Parcel_id));
```

```
CREATE TABLE Admin_Login (Email_a VARCHAR(255) NOT NULL,Username
VARCHAR(255) NOT NULL,Password VARCHAR(255) NOT NULL,PRIMARY
KEY(Email_a),Admin_id INT NOT NULL,FOREIGN KEY (Admin_id)
REFERENCES admin(Admin_id));
```

```
CREATE TABLE Customer_Login (Email_c VARCHAR(255) NOT
NULL,Username VARCHAR(255) NOT NULL,Password VARCHAR(255) NOT
NULL,PRIMARY KEY(Email_c),Customer_id INT NOT NULL,FOREIGN KEY
(Customer_id) REFERENCES customer(Customer_id));
```

```
CREATE TABLE DeliverPer_Login (Email_d VARCHAR(255) NOT NULL,
Username VARCHAR(255) NOT NULL, Password VARCHAR(255) NOT NULL,
PRIMARY KEY(Email_d),Delivery_id INT NOT NULL,FOREIGN KEY
(Delivery_id) REFERENCES delivery_person(Delivery_id));
```

- **Populate the Tables with the sample data:**

```
INSERT INTO admin (F_Name, L_Name, Phone_no, Admin_id)
VALUES  ('John', 'Doe', '1234567890', 10001),
('Jane', 'Smith', '2345678901', 10002),
('Michael', 'Johnson', '3456789012', 10003);

INSERT INTO admin_Login (Email_a, Username, Password, Admin_id)
VALUES ('admin1@example.com', 'admin1', 'password1', '10001'),
('admin2@example.com', 'admin2', 'password2', '10002 '),
('admin3@example.com', 'admin3', 'password3', '10003');

INSERT INTO delivery_person (Name, Phone_no, Delivery_id, Parcel_id)
VALUES
('Delivery Person 1', '1234567890', 50001, 1001),
('Delivery Person 2', '2345678901', 50002, 1002),
('Delivery Person 3', '3456789012', 50003, 1003),
('Delivery Person 4', '4567890123', 50004, 1004),
('Delivery Person 5', '5678901234', 50005, 1005),
('Delivery Person 6', '6789012345', 50006, 1006),
('Delivery Person 7', '7890123456', 50007, 1007),
('Delivery Person 8', '8901234567', 50008, 1008),
('Delivery Person 9', '9012345678', 50009, 1009),
('Delivery Person 10', '0123456789', 50010, 1010),
('Delivery Person 11', '1122334455', 50011, 1011),
('Delivery Person 12', '2233445566', 50012, 1012),
('Delivery Person 13', '3344556677', 50013, 1013),
('Delivery Person 14', '4455667788', 50014, 1014),
('Delivery Person 15', '5566778899', 50015, 1015);

INSERT INTO customer_login (Email_c, Username, Password, Customer_id)
VALUES
('customer1@example.com', 'customer1', 'password1', 10001),
('customer2@example.com', 'customer2', 'password2', 10002),
('customer3@example.com', 'customer3', 'password3', 10003),
('customer4@example.com', 'customer4', 'password4', 10004),
('customer5@example.com', 'customer5', 'password5', 10005),
('customer6@example.com', 'customer6', 'password6', 10006),
('customer7@example.com', 'customer7', 'password7', 10007),
('customer8@example.com', 'customer8', 'password8', 10008),
('customer9@example.com', 'customer9', 'password9', 10009),
('customer10@example.com', 'customer10', 'password10', 10010),
('customer11@example.com', 'customer11', 'password11', 10011),
('customer12@example.com', 'customer12', 'password12', 10012),
('customer13@example.com', 'customer13', 'password13', 10013),
('customer14@example.com', 'customer14', 'password14', 10014),
        ('customer15@example.com', 'customer15', 'password15', 10015);

INSERT INTO item (Parcel_id, Sender_Address, Receiver_Address, Phone_no, Status,
Item_type) VALUES
(1001, '123 Elm St', '456 Main St', '5678901234', 'In Transit', 'Electronics'),
(1002, '456 Main St', '789 Oak St', '5678901234', 'Delivered', 'Clothing'),
(1003, '789 Oak St', '123 Elm St', '5678901234', 'In Transit', 'Books'),
(1004, '123 Elm St', '456 Main St', '5678901234', 'In Transit', 'Furniture'),
(1005, '456 Main St', '789 Oak St', '5678901234', 'Delivered', 'Electronics'),
(1006, '789 Oak St', '123 Elm St', '5678901234', 'In Transit', 'Clothing'),
(1007, '123 Elm St', '456 Main St', '5678901234', 'In Transit', 'Books'),
(1008, '456 Main St', '789 Oak St', '5678901234', 'Delivered', 'Furniture'),
(1009, '789 Oak St', '123 Elm St', '5678901234', 'In Transit', 'Electronics'),
(1010, '123 Elm St', '456 Main St', '5678901234', 'In Transit', 'Clothing'),
(1011, '456 Main St', '789 Oak St', '5678901234', 'Delivered', 'Books'),
(1012, '789 Oak St', '123 Elm St', '5678901234', 'In Transit', 'Furniture'),
(1013, '123 Elm St', '456 Main St', '5678901234', 'In Transit', 'Electronics'),
(1014, '456 Main St', '789 Oak St', '5678901234', 'Delivered', 'Clothing'),
(1015, '789 Oak St', '123 Elm St', '5678901234', 'In Transit', 'Books');

INSERT INTO customer (Name, Phone_no, Customer_id, Parcel_id)
VALUES
('John Doe', '1234567890', 10001, 1001),
('Jane Smith', '2345678901', 10002, 1002),
('Michael Johnson', '3456789012', 10003, 1003),
('Emily Davis', '4567890123', 10004, 1004),
('David Anderson', '5678901234', 10005, 1005),
('John Doe', '6789012345', 10006, 1006),
('Jane Smith', '7890123456', 10007, 1007),
('Michael Johnson', '8901234567', 10008, 1008),
('Emily Davis', '9012345678', 10009, 1009),
('David Anderson', '0123456789', 10010, 1010),
('John Doe', '1234567890', 10011, 1011),
('Jane Smith', '2345678901', 10012, 1012),
('Michael Johnson', '3456789012', 10013, 1013),
('Emily Davis', '4567890123', 10014, 1014),
('David Anderson', '5678901234', 10015, 1015);

INSERT INTO payment (Payment_id, Date, Bank, Transaction_Number, Customer_id,
Parcel_id) VALUES
(10001, '2023-05-01', 'Bank A', '1234567890', 10001, 1001),
(10002, '2023-05-02', 'Bank B', '2345678901', 10002, 1002),
(10003, '2023-05-03', 'Bank C', '3456789012', 10003, 1003),
(10004, '2023-05-04', 'Bank D', '4567890123', 10004, 1004),
(10005, '2023-05-05', 'Bank E', '5678901234', 10005, 1005),
(10006, '2023-05-06', 'Bank F', '6789012345', 10006, 1006),
(10007, '2023-05-07', 'Bank G', '7890123456', 10007, 1007),
(10008, '2023-05-08', 'Bank H', '8901234567', 10008, 1008),
(10009, '2023-05-09', 'Bank I', '9012345678', 10009, 1009),
(10010, '2023-05-10', 'Bank J', '0123456789', 10010, 1010),
(10011, '2023-05-11', 'Bank K', '1234567890', 10011, 1011),
(10012, '2023-05-12', 'Bank L', '2345678901', 10012, 1012),
(10013, '2023-05-13', 'Bank M', '3456789012', 10013, 1013),
(10014, '2023-05-14', 'Bank N', '4567890123', 10014, 1014),
(10015, '2023-05-15', 'Bank O', '5678901234', 10015, 1015);

INSERT INTO customer_login (Email_c, Username, Password, Customer_id)
VALUES
('customer1@example.com', 'customer1', 'password1', 10001),
('customer2@example.com', 'customer2', 'password2', 10002),
('customer3@example.com', 'customer3', 'password3', 10003),
('customer4@example.com', 'customer4', 'password4', 10004),
('customer5@example.com', 'customer5', 'password5', 10005),
('customer6@example.com', 'customer6', 'password6', 10006),
('customer7@example.com', 'customer7', 'password7', 10007),
('customer8@example.com', 'customer8', 'password8', 10008),
('customer9@example.com', 'customer9', 'password9', 10009),
('customer10@example.com', 'customer10', 'password10', 10010),
('customer11@example.com', 'customer11', 'password11', 10011),
('customer12@example.com', 'customer12', 'password12', 10012),
('customer13@example.com', 'customer13', 'password13', 10013),
('customer14@example.com', 'customer14', 'password14', 10014),
('customer15@example.com', 'customer15', 'password15', 10015);

INSERT INTO delivery_person (Name, Phone_no, Delivery_id, Parcel_id) VALUES
('Delivery Person 1', '1234567890', 50001, 1001),
('Delivery Person 2', '2345678901', 50002, 1002),
('Delivery Person 3', '3456789012', 50003, 1003),
('Delivery Person 4', '4567890123', 50004, 1004),
('Delivery Person 5', '5678901234', 50005, 1005),
('Delivery Person 6', '6789012345', 50006, 1006),
('Delivery Person 7', '7890123456', 50007, 1007),
('Delivery Person 8', '8901234567', 50008, 1008),
('Delivery Person 9', '9012345678', 50009, 1009),
('Delivery Person 10', '0123456789', 50010, 1010),
('Delivery Person 11', '1122334455', 50011, 1011),
('Delivery Person 12', '2233445566', 50012, 1012),
('Delivery Person 13', '3344556677', 50013, 1013),
('Delivery Person 14', '4455667788', 50014, 1014),
('Delivery Person 15', '5566778899', 50015, 1015);

INSERT INTO deliverper_login (Email_d, Username, Password, Delivery_id)
```

```
VALUES
('delivery1@example.com', 'delivery1', 'password1', 50001),
('delivery2@example.com', 'delivery2', 'password2', 50002),
('delivery3@example.com', 'delivery3', 'password3', 50003),
('delivery4@example.com', 'delivery4', 'password4', 50004),
('delivery5@example.com', 'delivery5', 'password5', 50005),
('delivery6@example.com', 'delivery6', 'password6', 50006),
('delivery7@example.com', 'delivery7', 'password7', 50007),
('delivery8@example.com', 'delivery8', 'password8', 50008),
('delivery9@example.com', 'delivery9', 'password9', 50009),
('delivery10@example.com', 'delivery10', 'password10', 50010),
('delivery11@example.com', 'delivery11', 'password11', 50011),
('delivery12@example.com', 'delivery12', 'password12', 50012),
('delivery13@example.com', 'delivery13', 'password13', 50013),
('delivery14@example.com', 'delivery14', 'password14', 50014),
('delivery15@example.com', 'delivery15', 'password15', 50015);
```

- Retrieving required details and performing required operations:

```
ALTER TABLE item
ADD Delivery_id INT ,
ADD FOREIGN KEY (Delivery_id) REFERENCES delivery_person(Delivery_id);

UPDATE item AS i
 JOIN delivery_person AS DP ON i.Parcel_id = DP.Parcel_id
 SET i.Delivery_id = DP.Delivery_id;
```

Retrieve parcel_id and customer_id of all the parcel's that got DELIVERED.

```
SELECT I.Parcel_id, C.Customer_id
FROM Item AS I
JOIN Customer AS C ON I.Parcel_id = C.Parcel_id
WHERE I.Status = 'DELIVERED';
```

Retervie Customer details of the parcel's which are "IN transit"

```
SELECT C.*
FROM Customer AS C
JOIN Item AS I ON I.Parcel_id = C.Parcel_id
WHERE I.Status = 'IN TRANSIT';
```

Retrieve the payment and customer details of the parcel's whose payment is done between date 2023-05-01 to 2023-05-05

```
SELECT P.*, C.*
FROM Payment AS P
JOIN Customer AS C ON P.Customer_id = C.Customer_id
WHERE P.Date BETWEEN '2023-05-01' AND '2023-05-05';
```

Retrieve the name and phone_no the delivery person who delivered the parcels

```
SELECT DP.Name, DP.Phone_no, I.Status
FROM item AS I
NATURAL JOIN delivery_person AS DP
WHERE I.Status = 'DELIVERED' AND I.Parcel_id = DP.Parcel_id;
```

Retrieve the name and the Phone_no of the deliveryperson who delivered the parcel with parcel_id = 1002

```
SELECT DP.Name, DP.Phone_no
FROM item AS I
INNER JOIN delivery_person AS DP ON I.Parcel_id = DP.Parcel_id
WHERE I.Parcel_id = 1002
  AND I.Status = 'DELIVERED';
```

Retrive all delivery persons list the Name, Phone_no and also the Parcel_id if he has made any delivery

```
SELECT DP.Name, DP.Phone_no, I.Parcel_id
FROM delivery_person AS DP
LEFT JOIN item AS I ON DP.Parcel_id = I.Parcel_id
WHERE I.Status = 'DELIVERED' OR I.Status IS NULL;
```

Retrieve the details of parcel which is having maximum price.

```
ALTER TABLE payment
ADD price DECIMAL(10,2) NOT NULL;
UPDATE payment SET price = ROUND(RAND() * (5000 - 100) + 100, 2)
LIMIT 15;
SELECT I.Parcel_id, I.Sender_Address, I.Receiver_Address, I.Phone_no, I.Status,
I.Item_type, P.Payment_id, P.Date, P.Bank, P.Transaction_Number, P.Customer_id,
P.price
FROM item AS I
INNER JOIN payment AS P ON I.Parcel_id = P.Parcel_id
WHERE P.price = (SELECT MAX(price) FROM Payment);
```

Retrieve parcel and customer details whose parcel is not yet delivered.

```
SELECT I.Parcel_id, I.Sender_Address, I.Receiver_Address, I.Phone_no, I.Status, I.Item_type, C.Name AS Customer_Name, C.Phone_no AS Customer_Phone_no
FROM item AS I
INNER JOIN customer AS C ON I.Parcel_id = C.Parcel_id
WHERE I.Status <> 'DELIVERED';
```

Find the average price in the payments table

```
SELECT AVG(price) AS Average_Price
FROM payment;
```

Retrieve the count of the parcels which are delivered and which are yet to be delivered.

```
SELECT
   COUNT(CASE WHEN Status = 'DELIVERED' THEN Parcel_id END) AS Delivered_Count,
   COUNT(CASE WHEN Status <> 'DELIVERED' THEN Parcel_id END) AS Undelivered_Count
FROM item;
```

Retrieve the details of parcel which is having minimum price.

```
SELECT I.Parcel_id, I.Sender_Address, I.Receiver_Address, I.Phone_no, I.Status, I.Item_type, P.Payment_id, P.Date, P.Bank, P.Transaction_Number, P.Customer_id, P.price
FROM item AS I
INNER JOIN payment AS P ON I.Parcel_id = P.Parcel_id
WHERE P.price = (SELECT MIN(price) FROM Payment);
```

Find customers details who has the same item type ‘Electronics’ and ‘Clothing’

```
SELECT C.Name, C.Phone_no
FROM customer AS C
INNER JOIN item AS I ON C.Parcel_id = I.Parcel_id
WHERE I.Item_type = 'Electronics'
UNION
SELECT C.Name, C.Phone_no
FROM customer AS C
INNER JOIN item AS I ON C.Parcel_id = I.Parcel_id
WHERE I.Item_type = 'Clothing';
```

Find customer details whose name is other than John Doe and status is "In transit"

```
SELECT C.Name, C.Phone_no
FROM customer AS C
INNER JOIN item AS I ON C.Parcel_id = I.Parcel_id
WHERE C.Name <> 'John Doe'
INTERSECT
SELECT C.Name, C.Phone_no
FROM customer AS C
INNER JOIN item AS I ON C.Parcel_id = I.Parcel_id
WHERE I.Status = 'In transit';
```

Create a function to know the customer details of whose payment date is after 10-05-2023 and before 15-05-2023

```
DELIMITER $$
CREATE PROCEDURE GetCustomerDetailsByPaymentDate()
BEGIN
   SELECT C.Name, C.Phone_no
   FROM customer AS C
   INNER JOIN payment AS P ON C.Customer_id = P.Customer_id
   WHERE P.Date > '2023-05-10' AND P.Date < '2023-05-15';
END ;$$
DELIMITER ;
CALL GetCustomerDetailsByPaymentDate();
```

Write a procedure to retrive the details of the Customer and the parcel. Also display the details of the delivery person who delivered the parcel .

```
DELIMITER $$
CREATE PROCEDURE GetCustomerParcelDeliveryDetails()
BEGIN
   SELECT C.Name AS CustomerName, C.Phone_no AS CustomerPhone,
         I.Parcel_id, I.Sender_Address, I.Receiver_Address,
         DP.Name AS DeliveryPersonName, DP.Phone_no AS DeliveryPersonPhone
   FROM customer AS C
   INNER JOIN item AS I ON C.Parcel_id = I.Parcel_id
   INNER JOIN delivery_person AS DP ON I.Parcel_id = DP.Parcel_id;
END ;$$
DELIMITER ;
CALL GetCustomerParcelDeliveryDetails();
```

Create a trigger to stop updating the admin table and to show an error message and if the number of admins is greater than 3

```
DELIMITER $$
CREATE TRIGGER prevent_insert_admin
BEFORE INSERT ON admin
FOR EACH ROW
BEGIN
   DECLARE admin_count INT;
   SELECT COUNT(*) INTO admin_count FROM admin;
   IF admin_count >= 3 THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Cannot insert into Admin table. Maximum admin count reached.';
   END IF;
END ;$$
DELIMITER ;
```

```
INSERT INTO admin (F_Name, L_Name, Phone_no, Admin_id)
VALUES ('John', 'Seena', '9988776665', 10004);
```

Create a trigger to add payment information to the backup payment table when we try to delete some information from the payment table

```
CREATE TABLE backup_payment (Payment_id INT NOT NULL,Date DATE NOT NULL,Bank VARCHAR(255),Transaction_Number VARCHAR(255),Customer_id INT,
Parcel_id INT,PRIMARY KEY (Payment_id));

DELIMITER $$
CREATE TRIGGER backup_payment_delete_trigger
AFTER DELETE ON Payment
FOR EACH ROW
BEGIN
   INSERT INTO backup_payment (Payment_id, Date, Bank, Transaction_Number, Customer_id, Parcel_id)
   VALUES (OLD.Payment_id, OLD.Date, OLD.Bank, OLD.Transaction_Number, OLD.Customer_id, OLD.Parcel_id);
END ;$$
DELIMITER ;
```

```
delete from payment where Customer_id = 10001;
```

- **Connecting the backend to the Front end using Streamlit:**

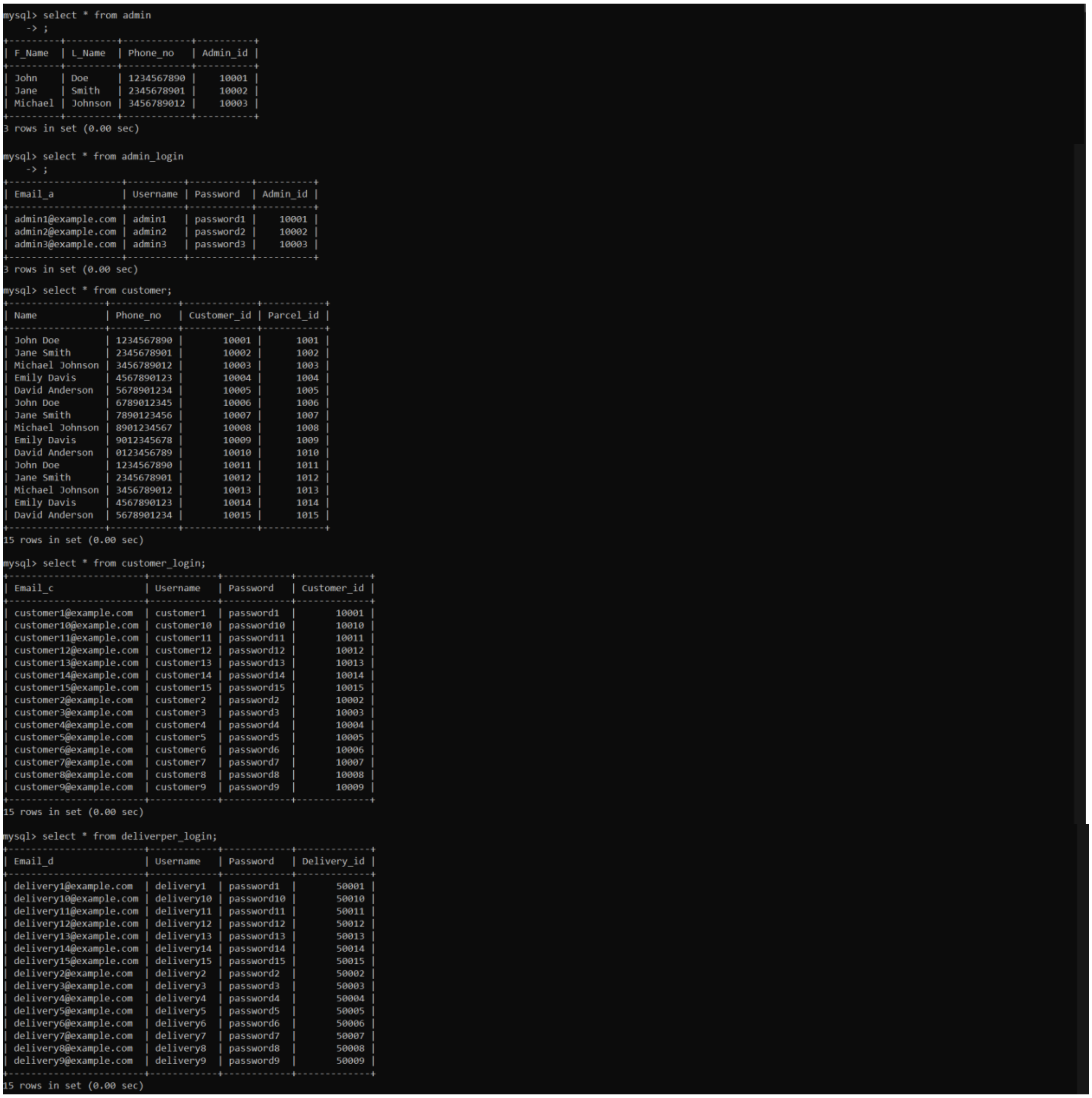
Python Code :

```python
import streamlit as st
import mysql.connector
from random import randint
from pandas import DataFrame

# Connect to the MySQL database
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="12345678",
    database="cms",
    auth_plugin='mysql_native_password'
)


# Function to authenticate admin login
def authenticate_admin(username, password):
    query = "SELECT * FROM admin_login WHERE Username = %s AND Password = %s"
    values = (username, password)
    cursor = db.cursor()
    cursor.execute(query, values)
    result = cursor.fetchone()
    cursor.close()
    return result


# Function to authenticate customer login
def authenticate_customer(username, password):
    query = "SELECT * FROM customer_login WHERE Username = %s AND Password =
%s"
    values = (username, password)
    cursor = db.cursor()
    cursor.execute(query, values)
    result = cursor.fetchone()
    cursor.close()
    return result


# Function to authenticate delivery person login
def authenticate_delivery_person(username, password):
    query = "SELECT * FROM deliverper_login WHERE Username = %s AND Password
= %s"
    values = (username, password)
    cursor = db.cursor()
    cursor.execute(query, values)
    result = cursor.fetchone()
    cursor.close()
    return result


# Admin Login Page
# Admin Login Page
def admin_login():
    st.title("Admin Login")
    username = st.text_input("Username")
    password = st.text_input("Password", type="password")
    if st.button("Login"):
        result = authenticate_admin(username, password)
        if result:
            st.success("Login Successful")
            admin_dashboard(result)  # Pass the result (admin credentials) to
the dashboard
        else:
            st.error("Invalid Username or Password")



# Customer Login Page
def customer_login():
    st.title("Customer Login")
    username = st.text_input("Username")
    password = st.text_input("Password", type="password")
    customer_id = st.text_input("Customer ID")
    if st.button("Login"):
        result = authenticate_customer(username, password)
        if result:
            st.success("Login Successful")
            customer_dashboard(customer_id)
        else:
            st.error("Invalid Username or Password")


# Delivery Person Login Page
def delivery_person_login():
    st.title("Delivery Person Login")
    username = st.text_input("Username")
    password = st.text_input("Password", type="password")
    delivery_person_id = st.text_input("Delivery_id")
    if st.button("Login"):
        result = authenticate_delivery_person(username, password)
        if result:
            st.success("Login Successful")
            delivery_person_dashboard(delivery_person_id)
            # Redirect to delivery person dashboard or perform other actions
        else:
            st.error("Invalid Username or Password")

# Function to get assigned parcel details for a delivery person
def get_assigned_parcel_details(delivery_person_id):
    query = f"SELECT * FROM item WHERE Delivery_id = {delivery_person_id}"
    cursor = db.cursor()
    cursor.execute(query)
    df = DataFrame(cursor.fetchall(), columns=['Parcel_id', 'Sender_Address',
'Receiver_Address', 'Phone_no', 'Status', 'Item_type',
                                            'Delivery_id'])
    cursor.close()
    return df
    cursor.close()
```

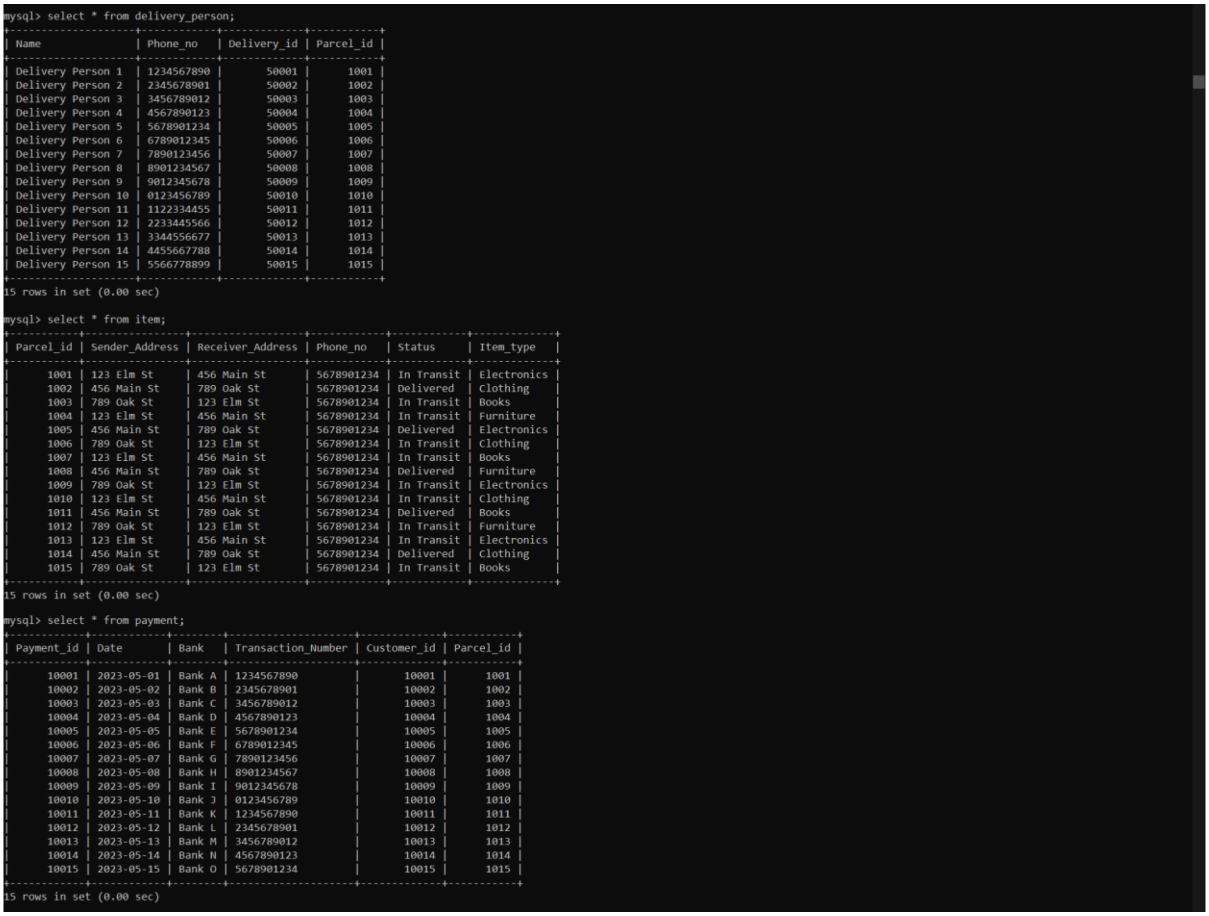
```
# Delivery Person Dashboard
def delivery_person_dashboard(delivery_person_id):
    st.title("Delivery Person Dashboard")

    # Get assigned parcel details
    assigned_parcel_details = get_assigned_parcel_details(delivery_person_id)
    st.header("Assigned Parcel Details")
    st.table(assigned_parcel_details)

    # Add other functionalities or actions specific to the delivery person
dashboard

    # Logout
    if st.button("Logout"):
        logout()
        st.success("Logged out successfully.")
        main()

# Admin Dashboard
def customer_dashboard(customer_id):
    st.title("Customer Dashboard")

    # Get Customer Details
    customer_details = get_customer_details(customer_id)
    st.header("Customer Details")
    st.table(customer_details)

    # Get Assigned Delivery Person
    assigned_delivery_person = get_assigned_delivery_person(customer_id)
    st.header("Assigned Delivery Person")
    st.table(assigned_delivery_person)

    # Add New Parcel
    st.header("Add New Parcel")
    parcel_sender_address = st.text_input("Sender Address")
    parcel_receiver_address = st.text_input("Receiver Address")
    parcel_phone_no = st.text_input("Phone Number")
    parcel_item_type = st.text_input("Item Type")
    if st.button("Add Parcel"):
        assigned_delivery_person_id = assign_delivery_person_randomly()
        parcel_id = randint(1, 10000)
        add_parcel(parcel_id, parcel_sender_address, parcel_receiver_address,
parcel_phone_no, parcel_item_type,
                   assigned_delivery_person_id)
        st.success("Parcel added successfully.")

    # Logout
    if st.button("Logout"):
        st.success("Logged out successfully.")
        main()


# Customer Dashboard
# Admin Dashboard
def admin_dashboard(admin_credentials):
    st.title("Admin Dashboard")

    # Display Customer Details
    st.header("Customer Details")
    customer_details = get_customer_details()
    st.table(customer_details)

    # Add Delivery Person
    st.header("Add Delivery Person")
    delivery_person_name = st.text_input("Name")
    delivery_person_phone = st.text_input("Phone Number")
    delivery_person_Delivery_id = st.text_input("Delivery_id")
    delivery_person_Parcel_id = st.text_input("Parcel_id")
    if st.button("Add"):
        add_delivery_person(delivery_person_name, delivery_person_phone,
delivery_person_Delivery_id, delivery_person_Parcel_id)
        st.success("Delivery Person added successfully.")

    # Delete Delivery Person
    st.header("Delete Delivery Person")
    delivery_person_id = st.number_input("Delivery Person ID")
    if st.button("Delete"):
        delete_delivery_person(delivery_person_id)
        st.success("Delivery Person deleted successfully.")

    # Display Parcel Details
    st.header("Parcel Details")
    parcel_details = get_parcel_details()
    st.table(parcel_details)

    if st.button("Logout"):
        logout()
        st.success("Logged out successfully.")
        main()


# Function to fetch customer details from the database
# Function to fetch all customer details from the database
def logout():
    pass
def get_customer_details():
    query = "SELECT * FROM customer"
    cursor = db.cursor()
    cursor.execute(query)
    df = DataFrame(cursor.fetchall(), columns=['Name', 'Phone_no',
'Customer_id', 'Parcel_id', 'Delivery_id'])
    cursor.close()
    return df


# Function to add a delivery person to the delivery_person table
def add_delivery_person(name, phone_no, parcel_id):
    count = get_delivery_person_count()
    if count >= 15:
        st.warning("Maximum limit of delivery persons reached. Cannot add new
delivery person.")
    else:
        query = "INSERT INTO delivery_person (Name, Phone_no, Parcel_id)
VALUES (%s, %s, %s)"
        values = (name, phone_no, parcel_id)
        cursor = db.cursor()
        cursor.execute(query, values)
        db.commit()
        cursor.close()


# Function to delete a delivery person from the delivery_person table
def delete_delivery_person(delivery_person_id):
    query = "DELETE FROM delivery_person WHERE Delivery_id = %s"
    values = (delivery_person_id,)
    cursor = db.cursor()
    cursor.execute(query, values)
    db.commit()
    cursor.close()


# Function to fetch parcel details from item table
def get_parcel_details():
    query = "SELECT * FROM item"
    cursor = db.cursor()
    cursor.execute(query)
    df = DataFrame(cursor.fetchall(),
                   columns=['Parcel_id', 'Sender_Address',
'Receiver_Address', 'Phone_no', 'Status', 'Item_type',
                            'Delivery_id'])
    cursor.close()
    return df


# Function to get the count of delivery persons
def get_delivery_person_count():
    cursor = db.cursor()
    query = "SELECT COUNT(*) FROM delivery_person"
    cursor.execute(query)
    count = cursor.fetchone()[0]
    cursor.close()
    return count


# Function to get the assigned delivery person for a customer
def get_assigned_delivery_person(customer_id):
    query = f"SELECT dp.Name, dp.Phone_no FROM delivery_person dp INNER JOIN
customer c ON dp.Delivery_id = c.Delivery_id WHERE c.Customer_id =
{customer_id}"
    cursor = db.cursor()
    cursor.execute(query)
    results = cursor.fetchall()
    cursor.close()
    return results


# Function to assign a delivery person randomly to a customer
def assign_delivery_person_randomly():
    cursor = db.cursor()
    query = "SELECT Delivery_id FROM delivery_person"
    cursor.execute(query)
    delivery_person_ids = cursor.fetchall()
    assigned_delivery_person_id = delivery_person_ids[randint(0,
len(delivery_person_ids) - 1)][0]
    cursor.close()
    return assigned_delivery_person_id


# Function to add a new parcel to the item table
def add_parcel(parcel_id, sender_address, receiver_address, phone_no,
item_type, assigned_delivery_person_id):
    query = "INSERT INTO item (Parcel_id, Sender_Address, Receiver_Address,
Phone_no, Status, Item_type, Delivery_id) VALUES (%s, %s, %s, %s, %s, %s,
%s)"
    values = (parcel_id, sender_address, receiver_address, phone_no,
"Pending", item_type, assigned_delivery_person_id)
    cursor = db.cursor()
    cursor.execute(query, values)
    db.commit()
    cursor.close()


# Main function to handle the application flow
def main():
    st.title("Courier Management System")
    option = st.selectbox("Select User Type", ("Admin", "Customer", "Delivery
Person"))

    if option == "Admin":
        admin_login()
    elif option == "Customer":
        customer_login()
    elif option == "Delivery Person":
        delivery_person_login()


# Run the application
if __name__ == "__main__":
    main()
```

Result:

- Screenshots after inserting sample data and performing required operations:



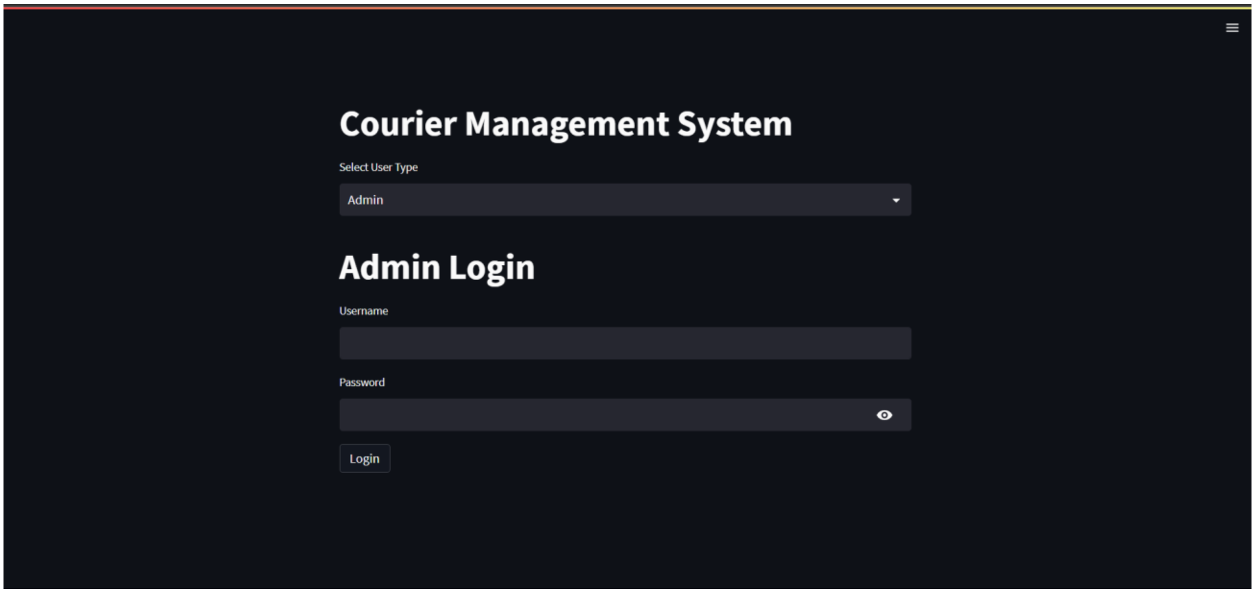
```
mysql> select * from admin
    -> ;
+---------+---------+------------+----------+
| F_Name  | L_Name  | Phone_no   | Admin_id |
+---------+---------+------------+----------+
| John    | Doe     | 1234567890 |    10001 |
| Jane    | Smith   | 2345678901 |    10002 |
| Michael | Johnson | 3456789012 |    10003 |
+---------+---------+------------+----------+
3 rows in set (0.00 sec)

mysql> select * from admin_login
    -> ;
+--------------------+----------+-----------+----------+
| Email_a            | Username | Password  | Admin_id |
+--------------------+----------+-----------+----------+
| admin1@example.com | admin1   | password1 |    10001 |
| admin2@example.com | admin2   | password2 |    10002 |
| admin3@example.com | admin3   | password3 |    10003 |
+--------------------+----------+-----------+----------+
3 rows in set (0.00 sec)

mysql> select * from customer;
+-----------------+------------+-------------+-----------+
| Name            | Phone_no   | Customer_id | Parcel_id |
+-----------------+------------+-------------+-----------+
| John Doe        | 1234567890 |       10001 |      1001 |
| Jane Smith      | 2345678901 |       10002 |      1002 |
| Michael Johnson | 3456789012 |       10003 |      1003 |
| Emily Davis     | 4567890123 |       10004 |      1004 |
| David Anderson  | 5678901234 |       10005 |      1005 |
| John Doe        | 6789012345 |       10006 |      1006 |
| Jane Smith      | 7890123456 |       10007 |      1007 |
| Michael Johnson | 8901234567 |       10008 |      1008 |
| Emily Davis     | 9012345678 |       10009 |      1009 |
| David Anderson  | 0123456789 |       10010 |      1010 |
| John Doe        | 1234567890 |       10011 |      1011 |
| Jane Smith      | 2345678901 |       10012 |      1012 |
| Michael Johnson | 3456789012 |       10013 |      1013 |
| Emily Davis     | 4567890123 |       10014 |      1014 |
| David Anderson  | 5678901234 |       10015 |      1015 |
+-----------------+------------+-------------+-----------+
15 rows in set (0.00 sec)

mysql> select * from customer_login;
+------------------------+------------+------------+-------------+
| Email_c                | Username   | Password   | Customer_id |
+------------------------+------------+------------+-------------+
| customer1@example.com  | customer1  | password1  |       10001 |
| customer10@example.com | customer10 | password10 |       10010 |
| customer11@example.com | customer11 | password11 |       10011 |
| customer12@example.com | customer12 | password12 |       10012 |
| customer13@example.com | customer13 | password13 |       10013 |
| customer14@example.com | customer14 | password14 |       10014 |
| customer15@example.com | customer15 | password15 |       10015 |
| customer2@example.com  | customer2  | password2  |       10002 |
| customer3@example.com  | customer3  | password3  |       10003 |
| customer4@example.com  | customer4  | password4  |       10004 |
| customer5@example.com  | customer5  | password5  |       10005 |
| customer6@example.com  | customer6  | password6  |       10006 |
| customer7@example.com  | customer7  | password7  |       10007 |
| customer8@example.com  | customer8  | password8  |       10008 |
| customer9@example.com  | customer9  | password9  |       10009 |
+------------------------+------------+------------+-------------+
15 rows in set (0.00 sec)

mysql> select * from deliverper_login;
+------------------------+------------+------------+-------------+
| Email_d                | Username   | Password   | Delivery_id |
+------------------------+------------+------------+-------------+
| delivery1@example.com  | delivery1  | password1  |       50001 |
| delivery10@example.com | delivery10 | password10 |       50010 |
| delivery11@example.com | delivery11 | password11 |       50011 |
| delivery12@example.com | delivery12 | password12 |       50012 |
| delivery13@example.com | delivery13 | password13 |       50013 |
| delivery14@example.com | delivery14 | password14 |       50014 |
| delivery15@example.com | delivery15 | password15 |       50015 |
| delivery2@example.com  | delivery2  | password2  |       50002 |
| delivery3@example.com  | delivery3  | password3  |       50003 |
| delivery4@example.com  | delivery4  | password4  |       50004 |
| delivery5@example.com  | delivery5  | password5  |       50005 |
| delivery6@example.com  | delivery6  | password6  |       50006 |
| delivery7@example.com  | delivery7  | password7  |       50007 |
| delivery8@example.com  | delivery8  | password8  |       50008 |
| delivery9@example.com  | delivery9  | password9  |       50009 |
+------------------------+------------+------------+-------------+
15 rows in set (0.00 sec)
```



```
mysql> select * from delivery_person;
+--------------------+------------+-------------+-----------+
| Name               | Phone_no   | Delivery_id | Parcel_id |
+--------------------+------------+-------------+-----------+
| Delivery Person 1  | 1234567890 |       50001 |      1001 |
| Delivery Person 2  | 2345678901 |       50002 |      1002 |
| Delivery Person 3  | 3456789012 |       50003 |      1003 |
| Delivery Person 4  | 4567890123 |       50004 |      1004 |
| Delivery Person 5  | 5678901234 |       50005 |      1005 |
| Delivery Person 6  | 6789012345 |       50006 |      1006 |
| Delivery Person 7  | 7890123456 |       50007 |      1007 |
| Delivery Person 8  | 8901234567 |       50008 |      1008 |
| Delivery Person 9  | 9012345678 |       50009 |      1009 |
| Delivery Person 10 | 0123456789 |       50010 |      1010 |
| Delivery Person 11 | 1122334455 |       50011 |      1011 |
| Delivery Person 12 | 2233445566 |       50012 |      1012 |
| Delivery Person 13 | 3344556677 |       50013 |      1013 |
| Delivery Person 14 | 4455667788 |       50014 |      1014 |
| Delivery Person 15 | 5566778899 |       50015 |      1015 |
+--------------------+------------+-------------+-----------+
15 rows in set (0.00 sec)

mysql> select * from item;
+-----------+----------------+------------------+------------+------------+-------------+
| Parcel_id | Sender_Address | Receiver_Address | Phone_no   | Status     | Item_type   |
+-----------+----------------+------------------+------------+------------+-------------+
|      1001 | 123 Elm St     | 456 Main St      | 5678901234 | In Transit | Electronics |
|      1002 | 456 Main St    | 789 Oak St       | 5678901234 | Delivered  | Clothing    |
|      1003 | 789 Oak St     | 123 Elm St       | 5678901234 | In Transit | Books       |
|      1004 | 123 Elm St     | 456 Main St      | 5678901234 | In Transit | Furniture   |
|      1005 | 456 Main St    | 789 Oak St       | 5678901234 | Delivered  | Electronics |
|      1006 | 789 Oak St     | 123 Elm St       | 5678901234 | In Transit | Clothing    |
|      1007 | 123 Elm St     | 456 Main St      | 5678901234 | In Transit | Books       |
|      1008 | 456 Main St    | 789 Oak St       | 5678901234 | Delivered  | Furniture   |
|      1009 | 789 Oak St     | 123 Elm St       | 5678901234 | In Transit | Electronics |
|      1010 | 123 Elm St     | 456 Main St      | 5678901234 | In Transit | Clothing    |
|      1011 | 456 Main St    | 789 Oak St       | 5678901234 | Delivered  | Books       |
|      1012 | 789 Oak St     | 123 Elm St       | 5678901234 | In Transit | Furniture   |
|      1013 | 123 Elm St     | 456 Main St      | 5678901234 | In Transit | Electronics |
|      1014 | 456 Main St    | 789 Oak St       | 5678901234 | Delivered  | Clothing    |
|      1015 | 789 Oak St     | 123 Elm St       | 5678901234 | In Transit | Books       |
+-----------+----------------+------------------+------------+------------+-------------+
15 rows in set (0.00 sec)

mysql> select * from payment;
+------------+------------+--------+--------------------+-------------+-----------+
| Payment_id | Date       | Bank   | Transaction_Number | Customer_id | Parcel_id |
+------------+------------+--------+--------------------+-------------+-----------+
|      10001 | 2023-05-01 | Bank A | 1234567890         |       10001 |      1001 |
|      10002 | 2023-05-02 | Bank B | 2345678901         |       10002 |      1002 |
|      10003 | 2023-05-03 | Bank C | 3456789012         |       10003 |      1003 |
|      10004 | 2023-05-04 | Bank D | 4567890123         |       10004 |      1004 |
|      10005 | 2023-05-05 | Bank E | 5678901234         |       10005 |      1005 |
|      10006 | 2023-05-06 | Bank F | 6789012345         |       10006 |      1006 |
|      10007 | 2023-05-07 | Bank G | 7890123456         |       10007 |      1007 |
|      10008 | 2023-05-08 | Bank H | 8901234567         |       10008 |      1008 |
|      10009 | 2023-05-09 | Bank I | 9012345678         |       10009 |      1009 |
|      10010 | 2023-05-10 | Bank J | 0123456789         |       10010 |      1010 |
|      10011 | 2023-05-11 | Bank K | 1234567890         |       10011 |      1011 |
|      10012 | 2023-05-12 | Bank L | 2345678901         |       10012 |      1012 |
|      10013 | 2023-05-13 | Bank M | 3456789012         |       10013 |      1013 |
|      10014 | 2023-05-14 | Bank N | 4567890123         |       10014 |      1014 |
|      10015 | 2023-05-15 | Bank O | 5678901234         |       10015 |      1015 |
+------------+------------+--------+--------------------+-------------+-----------+
15 rows in set (0.00 sec)
```

- Screenshots after connecting Database to the Frontend:

Landing Page : Admin Login Page



Courier Management System

Select User Type

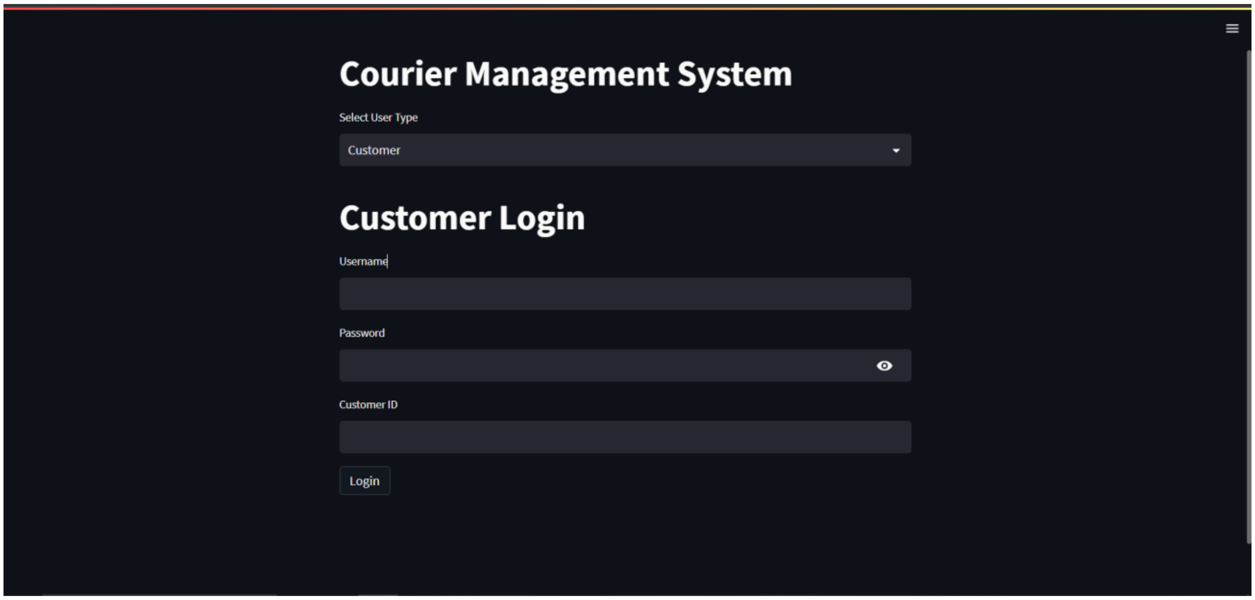
Admin

Admin Login

Username

Password

Customer Login Page :



Courier Management System

Select User Type

Customer

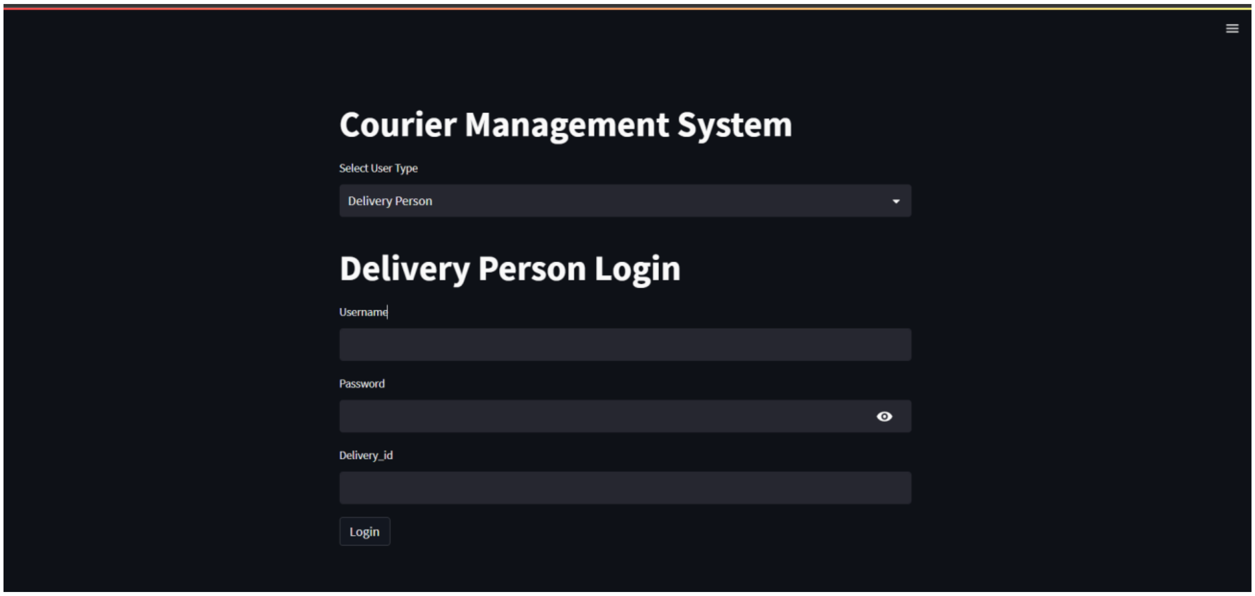
Customer Login

Username

Password

Customer ID

Delivery Person Login Page :



Courier Management System

Select User Type

Delivery Person

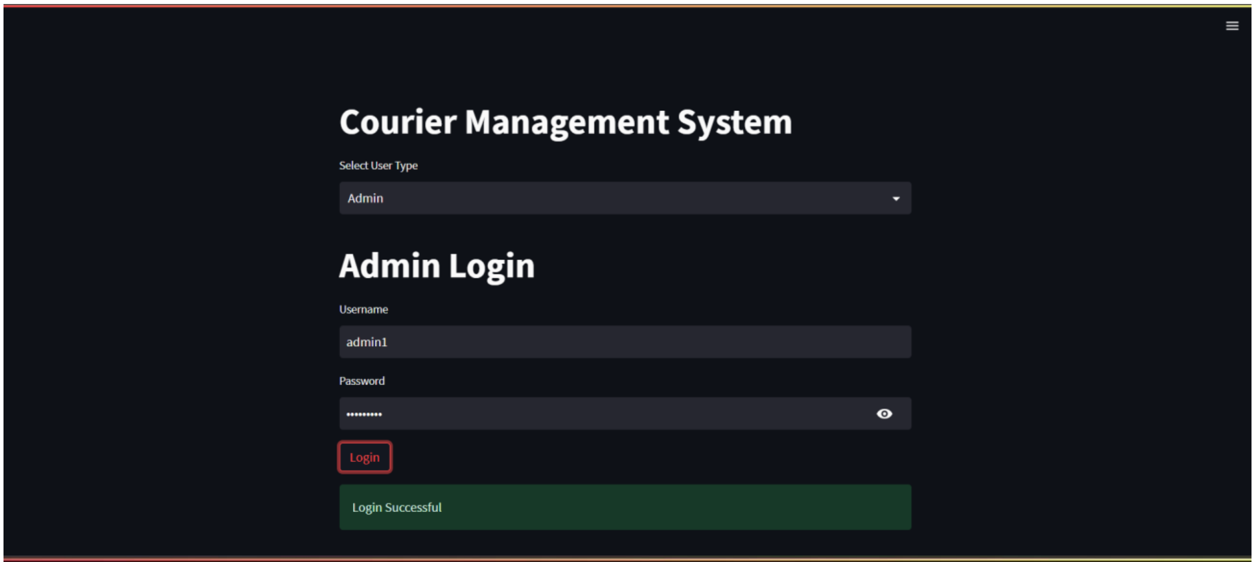
Delivery Person Login

Username

Password

Delivery_id

Admin Dashboard after successful login:



Courier Management System

Select User Type

Admin

Admin Login

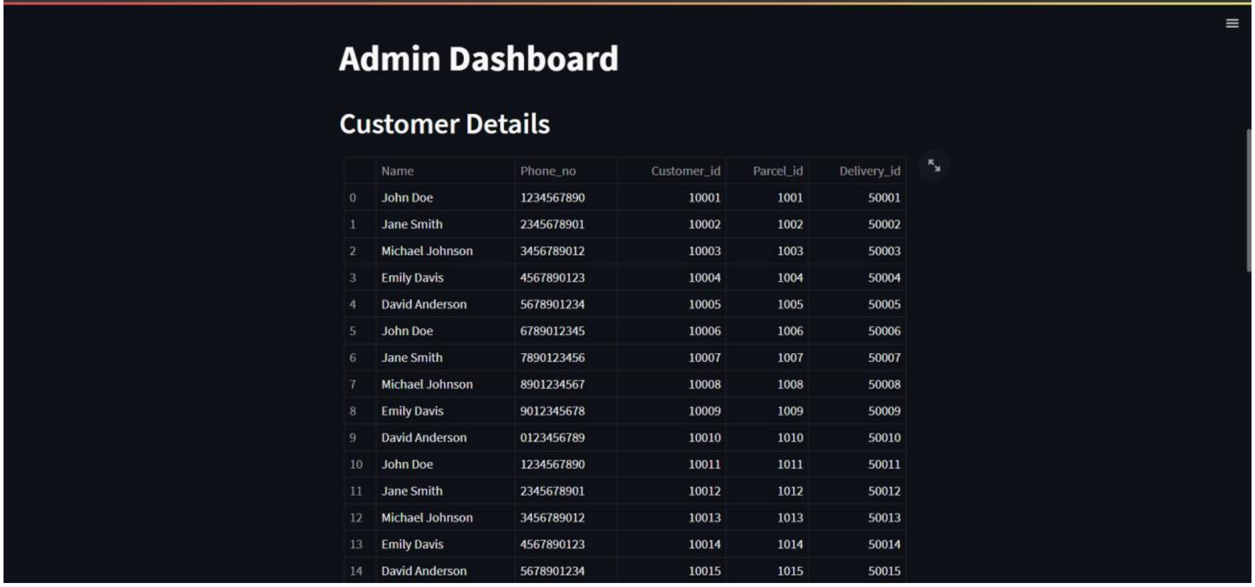
Username

admin1

Password

•••••••••

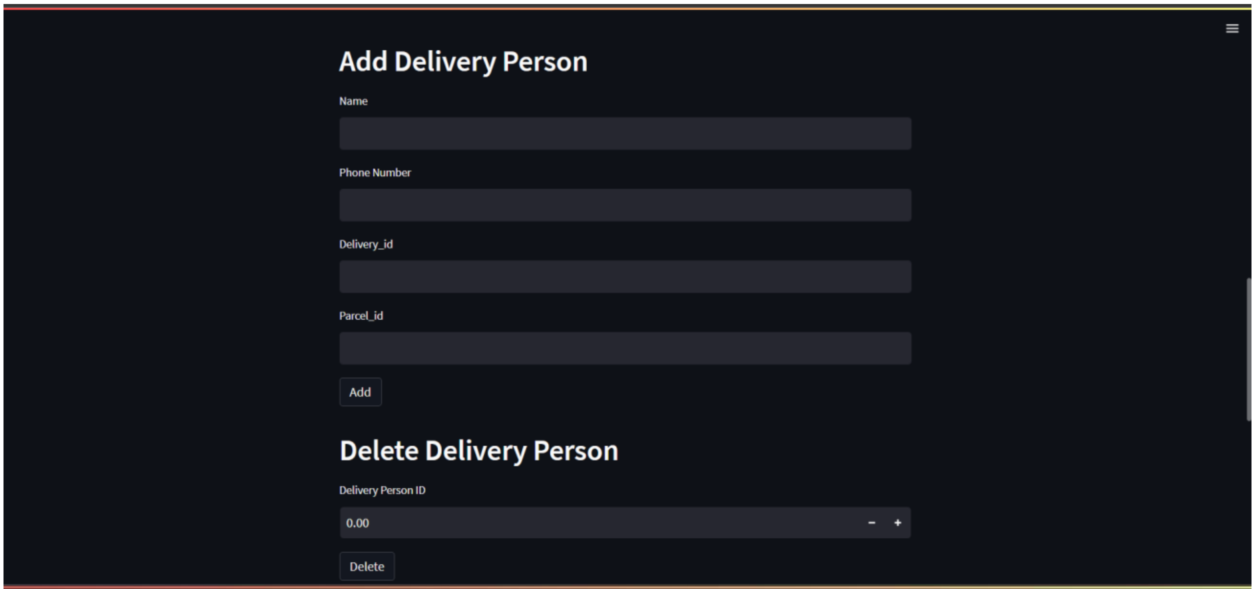
Login Successful



Admin Dashboard

Customer Details

	Name	Phone_no	Customer_id	Parcel_id	Delivery_id
0	John Doe	1234567890	10001	1001	50001
1	Jane Smith	2345678901	10002	1002	50002
2	Michael Johnson	3456789012	10003	1003	50003
3	Emily Davis	4567890123	10004	1004	50004
4	David Anderson	5678901234	10005	1005	50005
5	John Doe	6789012345	10006	1006	50006
6	Jane Smith	7890123456	10007	1007	50007
7	Michael Johnson	8901234567	10008	1008	50008
8	Emily Davis	9012345678	10009	1009	50009
9	David Anderson	0123456789	10010	1010	50010
10	John Doe	1234567890	10011	1011	50011
11	Jane Smith	2345678901	10012	1012	50012
12	Michael Johnson	3456789012	10013	1013	50013
13	Emily Davis	4567890123	10014	1014	50014
14	David Anderson	5678901234	10015	1015	50015



Add Delivery Person

Name

Phone Number

Delivery_id

Parcel_id

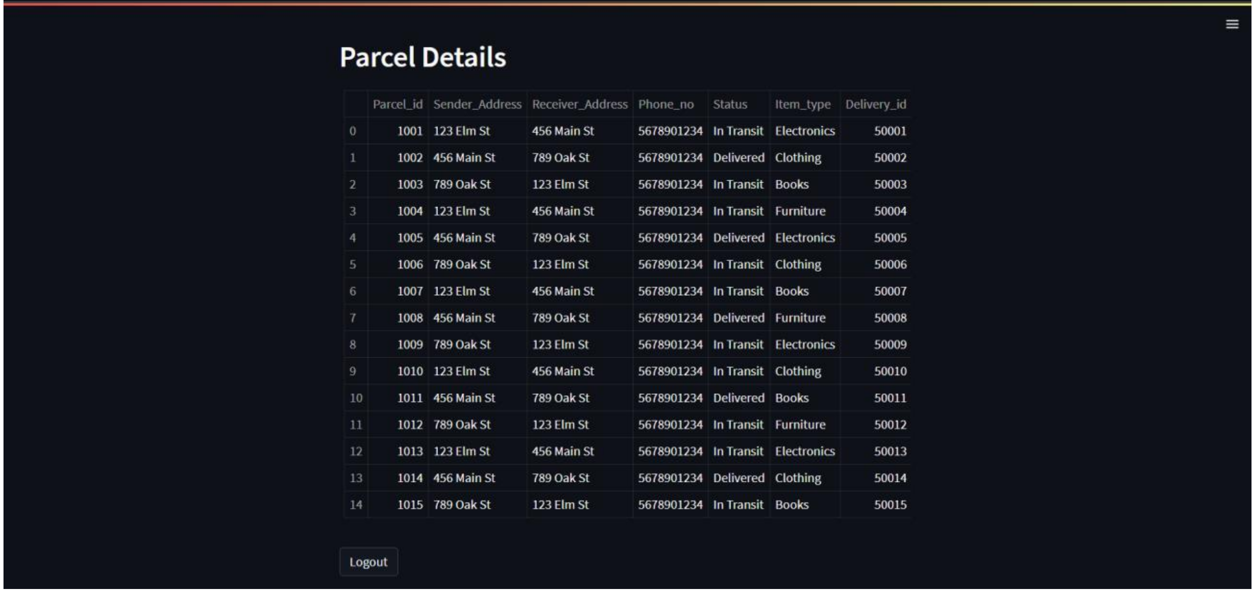
Add

Delete Delivery Person

Delivery Person ID

0.00

Delete

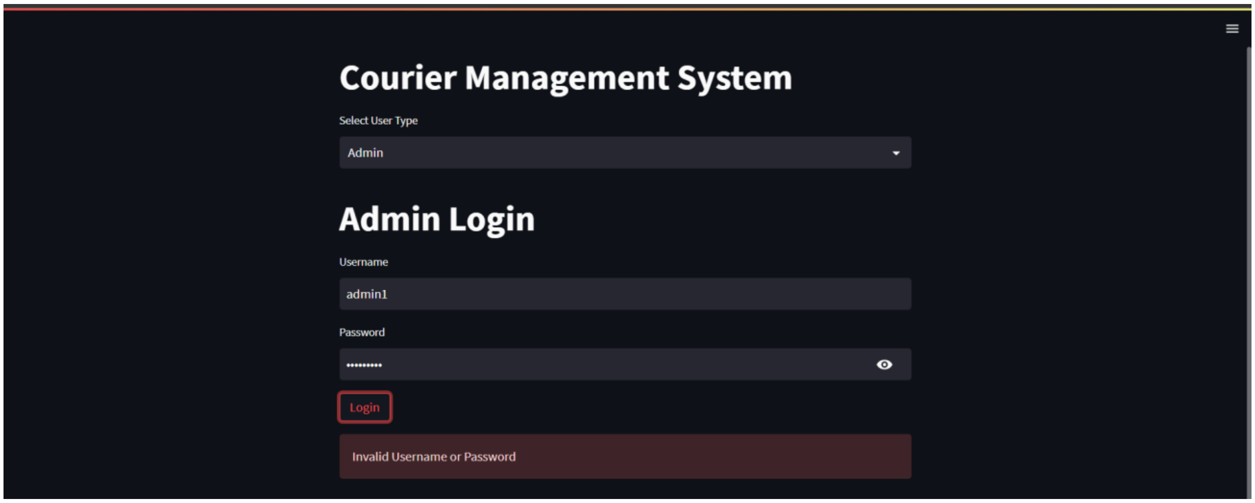


Parcel Details

	Parcel_id	Sender_Address	Receiver_Address	Phone_no	Status	Item_type	Delivery_id
0	1001	123 Elm St	456 Main St	5678901234	In Transit	Electronics	50001
1	1002	456 Main St	789 Oak St	5678901234	Delivered	Clothing	50002
2	1003	789 Oak St	123 Elm St	5678901234	In Transit	Books	50003
3	1004	123 Elm St	456 Main St	5678901234	In Transit	Furniture	50004
4	1005	456 Main St	789 Oak St	5678901234	Delivered	Electronics	50005
5	1006	789 Oak St	123 Elm St	5678901234	In Transit	Clothing	50006
6	1007	123 Elm St	456 Main St	5678901234	In Transit	Books	50007
7	1008	456 Main St	789 Oak St	5678901234	Delivered	Furniture	50008
8	1009	789 Oak St	123 Elm St	5678901234	In Transit	Electronics	50009
9	1010	123 Elm St	456 Main St	5678901234	In Transit	Clothing	50010
10	1011	456 Main St	789 Oak St	5678901234	Delivered	Books	50011
11	1012	789 Oak St	123 Elm St	5678901234	In Transit	Furniture	50012
12	1013	123 Elm St	456 Main St	5678901234	In Transit	Electronics	50013
13	1014	456 Main St	789 Oak St	5678901234	Delivered	Clothing	50014
14	1015	789 Oak St	123 Elm St	5678901234	In Transit	Books	50015

Logout

Unsuccessful login:



Courier Management System

Select User Type

Admin

Admin Login

Username

admin1

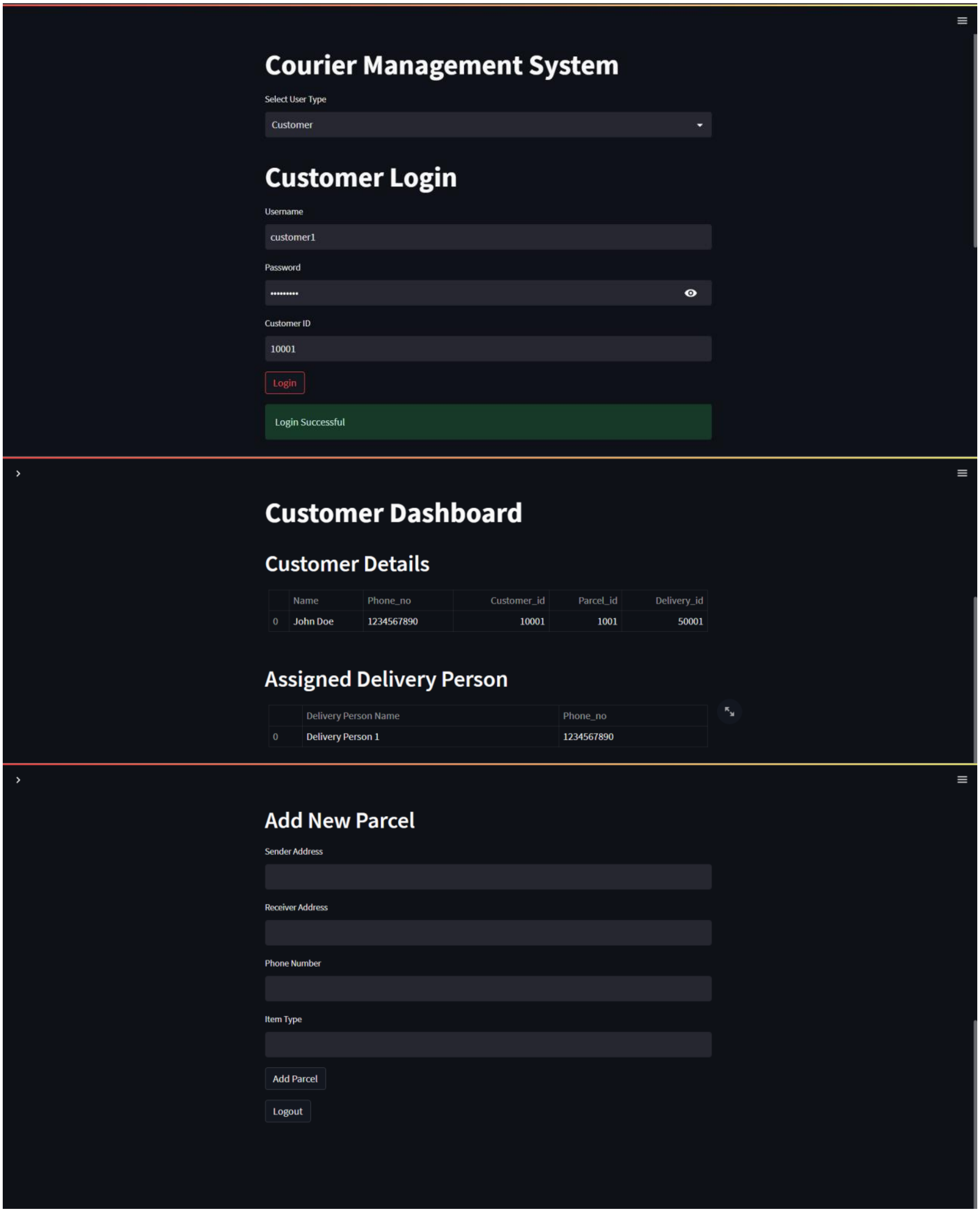
Password

•••••••••

Login

Invalid Username or Password

Customer Dashboard after successful login:



Courier Management System

Select User Type: Customer

Customer Login

Username: customer1

Password: •••••••••

Customer ID: 10001

Login Successful

Customer Dashboard

Customer Details

	Name	Phone_no	Customer_id	Parcel_id	Delivery_id
0	John Doe	1234567890	10001	1001	50001

Assigned Delivery Person

	Delivery Person Name	Phone_no
0	Delivery Person 1	1234567890

Add New Parcel

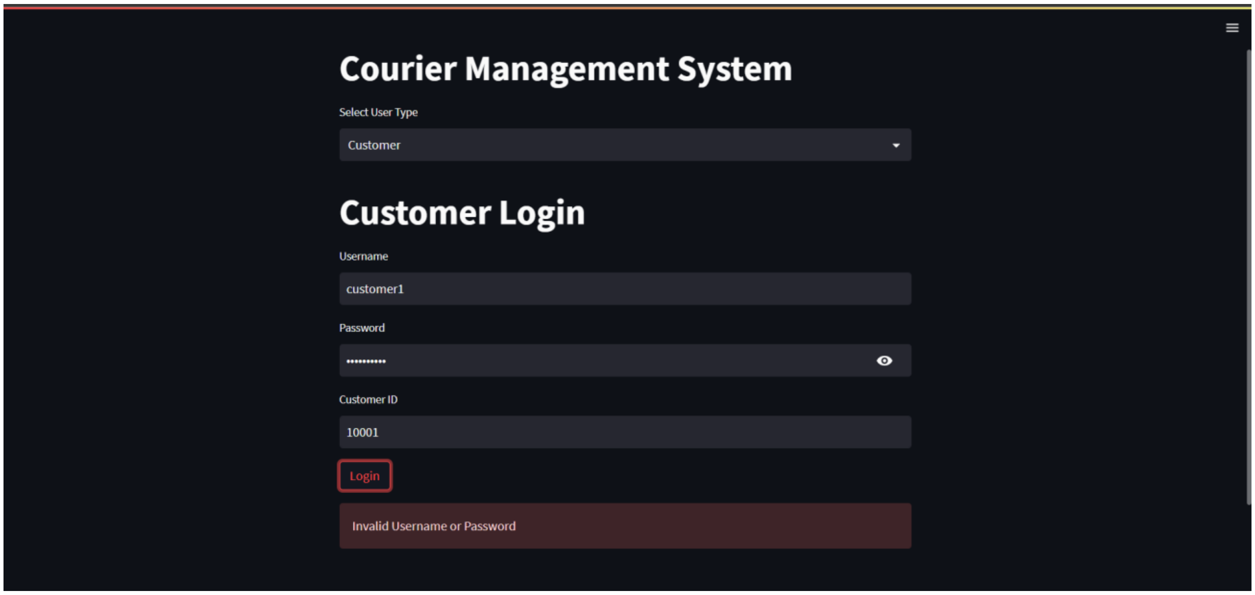
Sender Address

Receiver Address

Phone Number

Item Type

Unsuccessful login:



Courier Management System

Select User Type: Customer

Customer Login

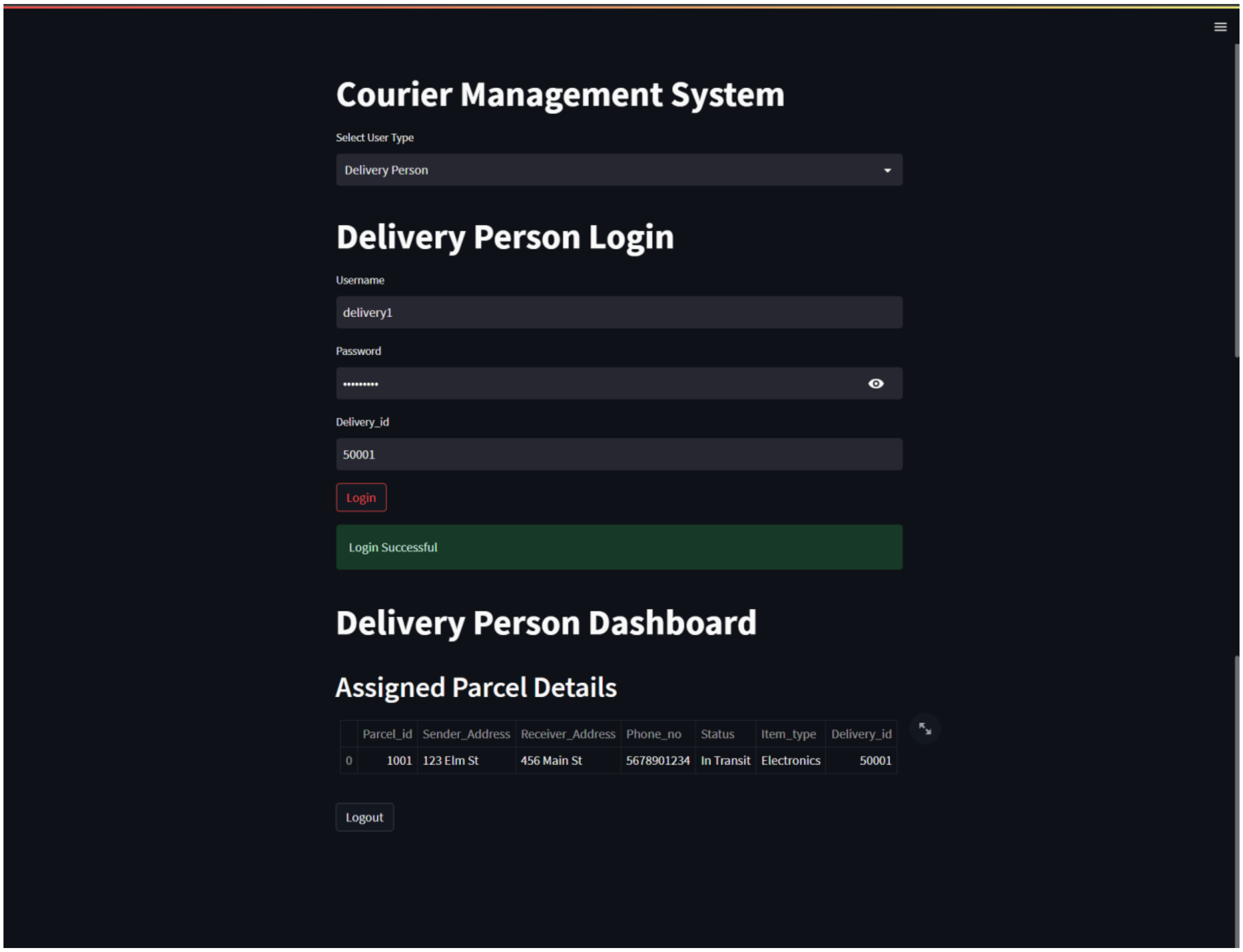
Username: customer1

Password: ••••••••••

Customer ID: 10001

Invalid Username or Password

Delivery Person Dashboard after successful login:



Courier Management System

Select User Type: Delivery Person

Delivery Person Login

Username: delivery1

Password: •••••••••

Delivery_id: 50001

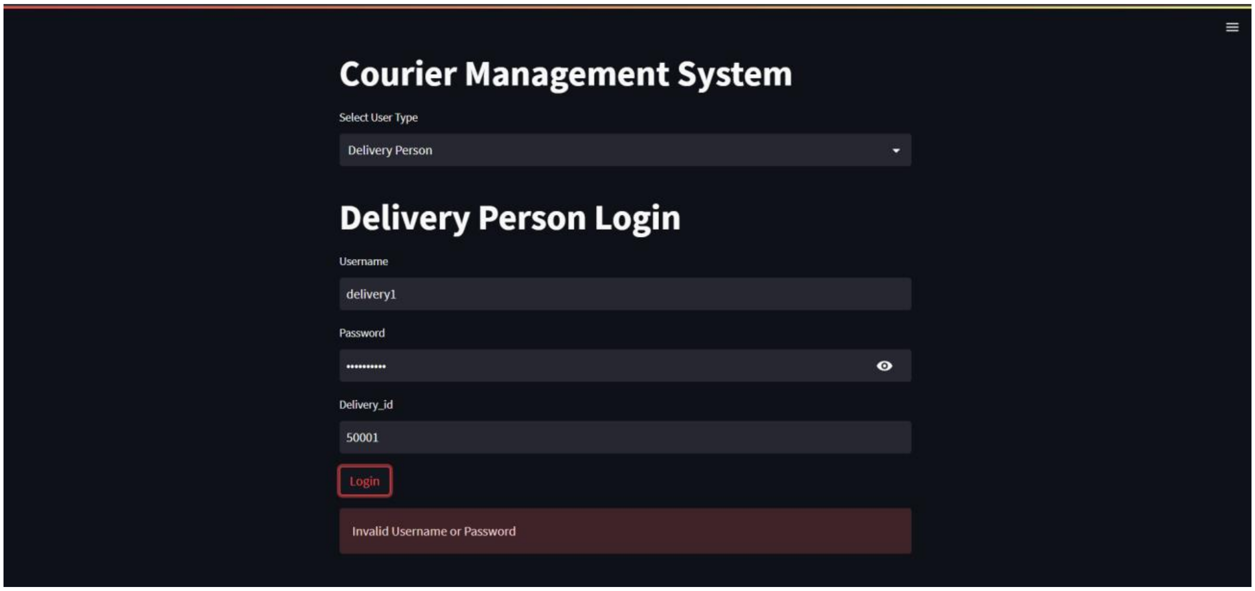
Login Successful

Delivery Person Dashboard

Assigned Parcel Details

	Parcel_id	Sender_Address	Receiver_Address	Phone_no	Status	Item_type	Delivery_id
0	1001	123 Elm St	456 Main St	5678901234	In Transit	Electronics	50001

Unsuccessful login:



Courier Management System

Select User Type

Delivery Person

Delivery Person Login

Username

delivery1

Password

••••••••••

Delivery_id

50001

Invalid Username or Password

Conclusions:

In conclusion, the mini project on courier management system database has successfully addressed the challenges faced in managing the courier operations efficiently. The project focused on designing a well-structured database schema that captures relevant information such as customers, delivery persons, parcels, and their relationships. By implementing this database, various tasks such as tracking parcels, assigning delivery persons, and managing customer details have been streamlined.

The database provides a reliable and scalable solution for storing and retrieving courier-related data. It offers flexibility in managing customer information, allowing efficient tracking and monitoring of parcel status. The integration of delivery person details enables proper assignment of parcels, optimizing the delivery process. The system also incorporates security measures, ensuring authorized access and protecting sensitive information.

Overall, the mini project on courier management system database has proved its effectiveness in enhancing the courier management process. By automating key tasks, it reduces manual efforts, minimizes errors, and improves overall operational efficiency. The project serves as a foundation for further enhancements and integration with other systems to create a comprehensive courier management solution.

Refrences:

- **UE20CS301 – Database Management Systems (Minors) Lab Assignments**
- YouTube

THANK YOU