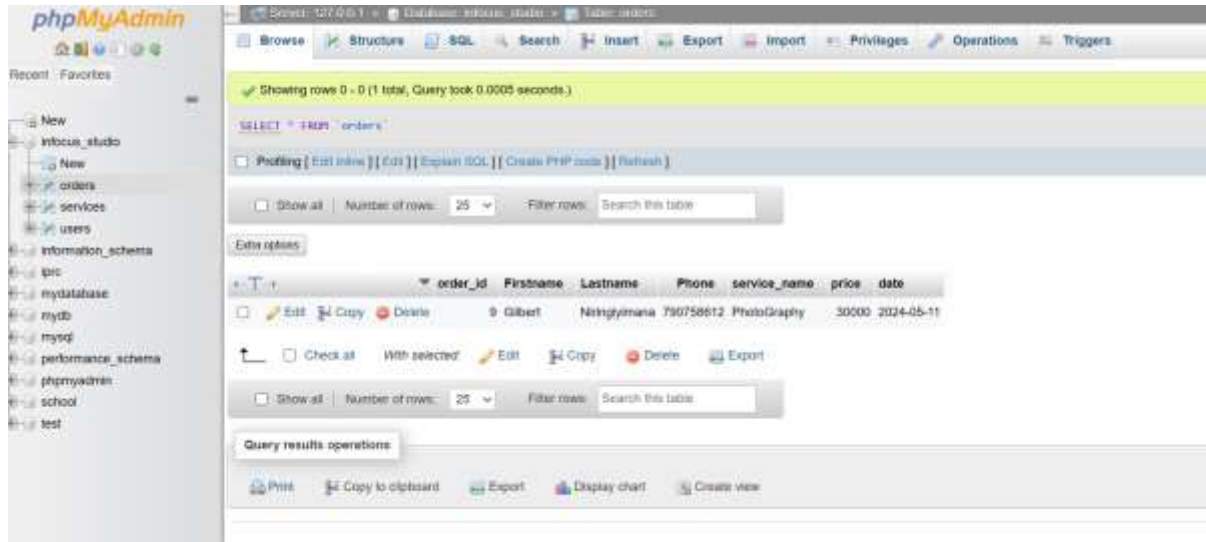


Project name: **Infocus Studio**

Infocus Studio Ltd provides easy access in photography and videography service:

Database structure:



Implementation project codes:

```
from flask import Flask, render_template, request, redirect, url_for, session, flash
import mysql.connector

app = Flask(__name__)
app.secret_key = "your_secret_key" # Change this to a secure random key in production

# Function to connect to the MySQL database
def get_db_connection():
    return mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='infocus_studio'
    )

# Register user route
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        firstname = request.form['firstName']
        lastname = request.form['lastName']
```

```

        phone = request.form['phone']
        username = request.form['username']
        password = request.form['password']

        conn = get_db_connection()
        cursor = conn.cursor()
        cursor.execute('INSERT INTO users (Firstname, Lastname, phone,
Username, Password) VALUES (%s, %s, %s, %s, %s)',
                        (firstname, lastname, phone, username, password))
        conn.commit()
        cursor.close()
        conn.close()
        return redirect(url_for('login'))
    return render_template('index.html')

@app.route('/')
def index():
    return render_template('index.html')

# Login route
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        conn = get_db_connection()
        cursor = conn.cursor(dictionary=True)
        cursor.execute('SELECT * FROM users WHERE Username = %s AND Password =
%s', (username, password))
        user = cursor.fetchone()
        cursor.close()
        conn.close()

        if user:
            session['user_id'] = user['user_id']
            return redirect(url_for('dashboard'))
        else:
            return render_template('login.html', error='Invalid username or
password')
    return render_template('login.html')

# Dashboard route
@app.route('/dashboard')
def dashboard():
    if 'user_id' in session:
        conn = get_db_connection()
        cursor = conn.cursor(dictionary=True)

```

```

        cursor.execute('SELECT * FROM users WHERE user_id = %s',
(session['user_id'],))
        user = cursor.fetchone()
        cursor.close()
        conn.close()

        if user:
            return render_template('dashboard.html',
username=user['Username'])
            return redirect(url_for('login'))

@app.route('/order_service', methods=['GET', 'POST'])
def order_service():
    # Fetch services from the database
    services = fetch_services_from_database()

    if request.method == 'POST':
        # Retrieve form data
        firstName = request.form['firstName']
        lastName = request.form['lastName']
        phone = request.form['phone']
        service_name = request.form['service_name']
        service_price = request.form['service_price']
        date = request.form['date']

        # Fetch user data from the database based on the session
        user = fetch_user_from_session()

        if user:
            # Insert order into database
            insert_order_into_database(firstName, lastName, phone,
service_name, service_price, date)

            # Redirect or render a thank you page
            flash('Order sent successfully!', 'success')

            return redirect(url_for('order_service'))
        else:
            # Handle the case where the user session is invalid
            return redirect(url_for('login'))

    # Fetch user data from the session
    user = fetch_user_from_session()

    return render_template('order_service.html', services=services, user=user)

def fetch_user_from_session():
    # Fetch user data from the session based on the user_id

```

```

user_id = session.get('user_id')
if user_id:
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute("SELECT Firstname, Lastname, phone FROM users WHERE
user_id = %s", (user_id,))
    user = cursor.fetchone()
    cursor.close()
    connection.close()
    return user
else:
    return None

def fetch_services_from_database():
    # Fetch services data from the database (replace this with your actual
    database query)
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute("SELECT service_id, service_name, price FROM services")
    services = cursor.fetchall()
    cursor.close()
    connection.close()
    return services

def get_service_price_from_database(service_id):
    # Fetch service price from the database (replace this with your actual
    database query)
    connection = get_db_connection()
    cursor = connection.cursor()
    cursor.execute("SELECT price FROM services WHERE service_id = %s",
(service_id,))
    result = cursor.fetchone()
    cursor.close()
    connection.close()
    if result:
        return result[0]
    else:
        return None

def insert_order_into_database(firstName, lastName, phone, service_name,
service_price, date):
    # Insert order into the database (replace this with your actual database
    insert statement)
    connection = get_db_connection()
    cursor = connection.cursor()
    cursor.execute("INSERT INTO orders (Firstname, Lastname, phone,
service_name, price, date) VALUES (%s, %s, %s, %s, %s, %s)",

```

```

        (firstName, lastName, phone, service_name, service_price,
date))
    connection.commit()
    cursor.close()
    connection.close()

# Logout route
@app.route('/logout')
def logout():
    session.pop('user_id', None)
    return redirect(url_for('login'))

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

```

if we run our code:

The screenshot shows a code editor with Python code for a Flask application. The code includes routes for index, login, and logout, and a main block to run the application. Below the code, the terminal output shows the application running successfully on http://127.0.0.1:5000.

```

34:         return render_template('index.html')
35:
36: @app.route('/')
37: def index():
38:     return render_template('index.html')
39:
40: # Login route
41: @app.route('/login', methods=['GET', 'POST'])
42: def login():
43:     if request.method == 'POST':
44:         username = request.form['username']
45:         password = request.form['password']
46:
47:         conn = get_db_connection()
48:         cursor = conn.cursor(dictionary=True)
49:         cursor.execute('SELECT * FROM users WHERE Username = %s AND Password = %s', (username, password))
50:         user = cursor.fetchone()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```

[Running] python -u "e:\InfocusStudioSystem\InfocusStudioSystem\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 328-849-483

```

1. Register the clients

Welcome to Infocus Studio Registration Page

First Name
Gilbert

Last Name
Niringiyimana

Phone
0790758612

Username
gilbert

Password

[Register](#)

Already have an account? [Login](#)

After register please login:

Login to In focus Studio

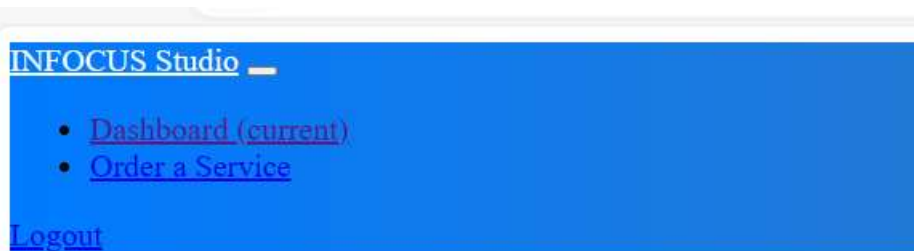
Username
gilbert

Password

[Login](#)

If you don't have an account, [Register](#)

- After login you will go to the dashboard please select one of 2 services
2. Request and order the photography day
 3. Request and order the videography day



Welcome, User: gilbert

Welcome to Infocus Studio System

Our Services

1. Photography
2. Videography

4. Pay for the services

INFOCUS Studio

- [Dashboard](#)
- [Order a Service](#)

[Logout](#)

Order a Service

First Name

Last Name

Phone

Select Service

Note: Remember Price 1 is For Photography*
Price 2 is For Videography

Select Price

Select Date

5. Receive the confirmation message

6. Accessing their pictures easily
Logout in the system

[INFOCUS Studio](#) 

- [Dashboard](#)
- [Order a Service](#)

[Logout](#)

Order sent successfully!

Order a Service

First Name
Last Name
Phone
Select Service

Note: Remember Price 1 is For Photography*
Price 2 is For Videography

Select Price
Select Date 