|  |
| --- |
| A PROJECT REPORT  on  **PHOTOGRAPHY APPLICATION**    Submitted in partial fullfillment for the award of degree    of  Bachelor of Science (Computer Science)      Submitted By:  **DEVANE ANIRUDH SHIVAKANT**  **DESHMUKH GAJANAN DEVRAO**  **DESHMUKH ANIRUDH GAJANAN**    Under the guidance of  **Ms. Kolhe Sukanya Subhash**      **MGM's**    **COLLEGE OF COMPUTER SCIENCE & IT**  **Department Of Computer Science & IT,**  (Institute established in 2000 and affiliated to S.R.T.M. University, Nanded)  MGM Campus, Near Airport, Nanded - 431605 (M.S.) India  Affiliated to    Swami Ramanand Teerth Marathwada University,  Nanded-431606  2024-25 |

|  |
| --- |
| **MGM's**  **COLLEGE OF COMPUTER SCIENCE & IT**  (Institute established in 2000 and affiliated to S.R.T.M. University, Nanded)  MGM Campus, Near Airport, Nanded - 431605 (M.S.) India      Certificate      This is to certify that the project entitled **"Photography Application"** by **DEVANE**  **ANIRUDH SHIVAKANT ( ), DESHMUKH GAJANAN DEVRAO (**  **), DESHMUKH ANIRUDH GAJANAN ( )** submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Science (Computer Science)** by **Swami Ramanand Teerth Marathwada University, Nanded – 431606** during the academic year **2024-25** is a bonafide record of work carried out under my guidance and supervision.    **Ms. Kolhe Sukanya Subhash**  **Guide**    **Dr.Mrs. Kanchan A. Nandedkar Prof. Shirish L. Kotgire**  **Head of Department Principal**      Examinar-1 Examinar-2 |

|  |
| --- |
| Declaration    We hereby declare that **project** entitled **"Photography Application"** has been completed in the Department of Computer Science & IT, MGM’s College of  Computer Science & IT, Nanded and Submitted to Swami Ramanand Teerth  Marathwada University Nanded, under the guidance of **Ms. Kolhe Sukanya Subhash** for the award of degree of **Bachelor of Science (Computer Science).** This report comprises only our original work and has not been submitted for award of any other degree to any university. Due acknowledgement has been made in the text to all other material used.    DEVANE ANIRUDH SHIVAKANT\_\_\_\_\_\_\_\_  DESHMUKH GAJANAN DEVRAO\_\_\_\_\_\_\_\_  DESHMUKH ANIRUDH GAJANAN\_\_\_\_\_\_\_\_    Date:27/3/2025  Place:Nanded |

**INTRODUCTION**

The

*Camera Project*

is a comprehensive, web

-

based application meticulously crafted to streamline and

simplify camera rental and sales operations. As the demand for high

-

quality photography and videography

equipment grows, this platform serves as a dynamic marketplace caterin

g to professionals, enthusiasts, and

casual users alike.

Designed with an intuitive and user

-

friendly interface, the system allows customers to effortlessly browse an

extensive catalog of cameras, lenses, drones, and accessories, each accompanied by detailed specifications

and pricing. Users can register on the

platform, authenticate securely, and conveniently place orders for

rentals or purchases. The project incorporates a robust order management system that efficiently handles

booking requests, inventory control, and order processing, ensuring a seamless exper

ience from browsing to

checkout.

The application prioritizes data security through advanced encryption protocols, safeguarding sensitive

customer information and secure payment processing. The platform also features a dedicated

*admin panel*

that empowers administrators to manage inventory, update product listings, track orders, and analyze

customer data effectively. This centralized control enhances operational efficiency and supports informed

decision

-

making.

Built using modern web development technologies and integrated with a scalable database, the Camera

Project offers reliable performance and the flexibility to expand as business needs grow. Its responsive

design ensures compatibility across devices, allowi

ng users to access the platform conveniently from

desktops, tablets, and smartphones.

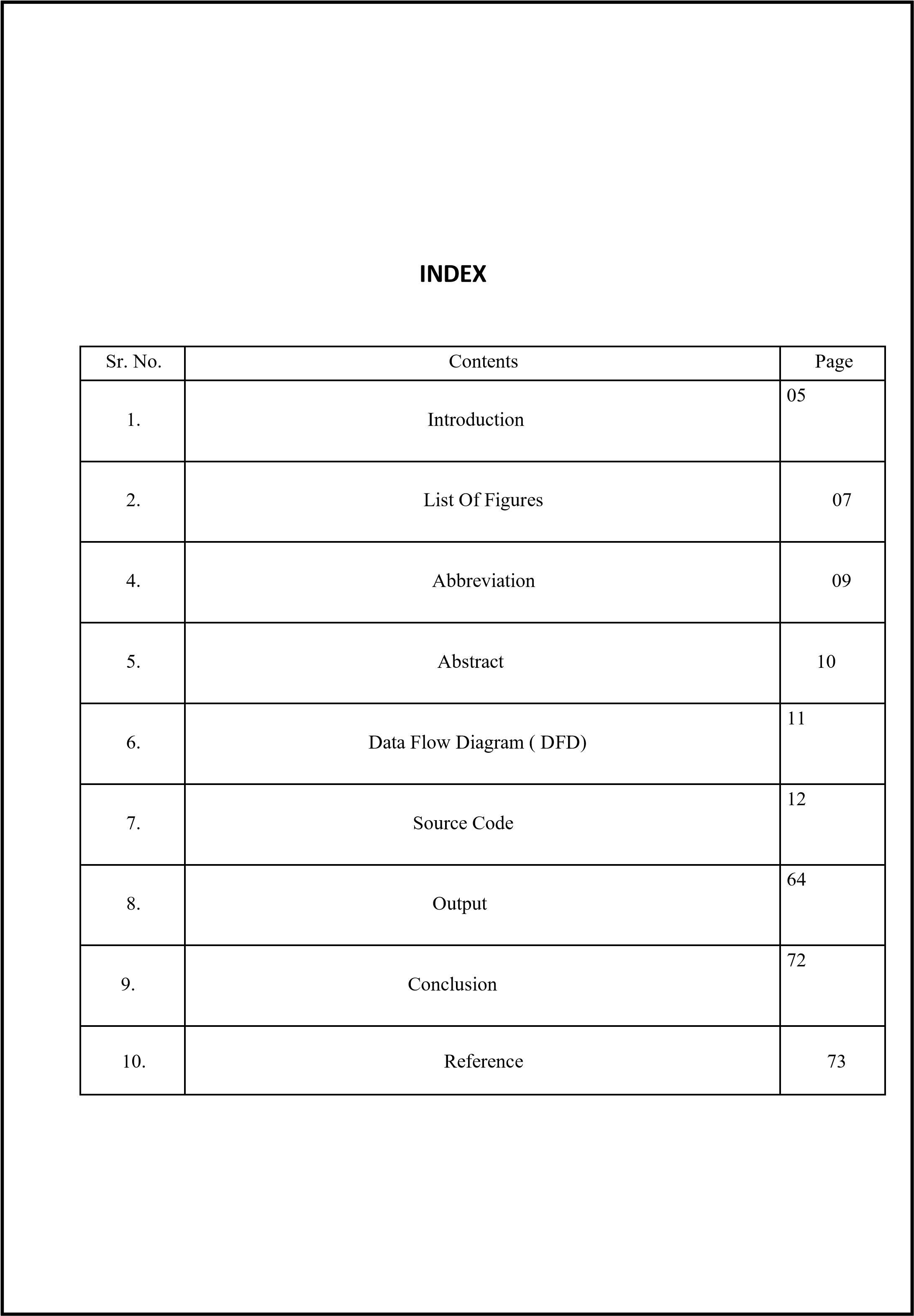
Ultimately, the objective of the Camera Project is to transform the experience of renting or purchasing

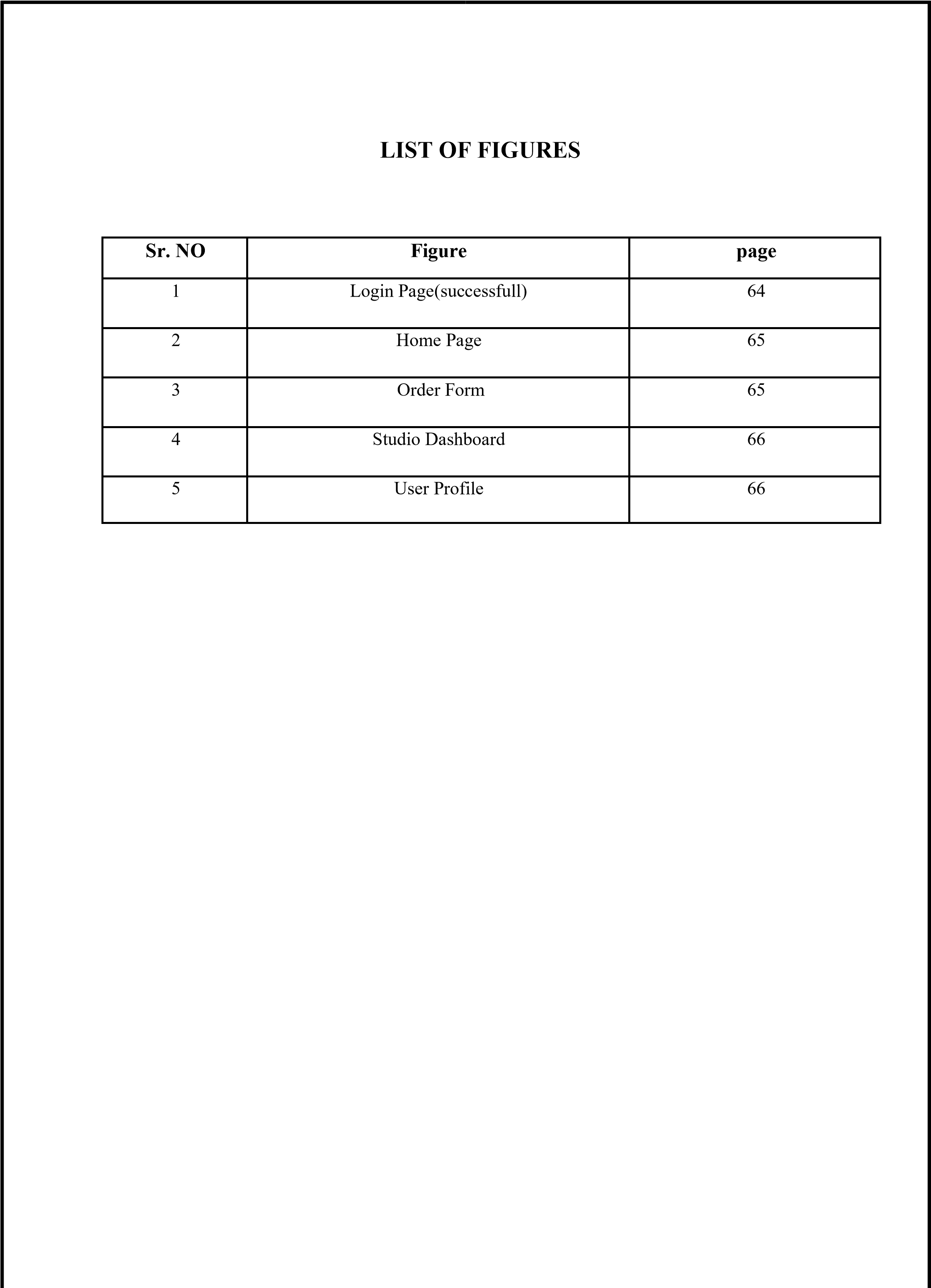
camera equipment, making it more accessible, efficient, and enjoyable. By bridging the gap between

suppliers and customers, the platform aims to be a tr

usted solution for all photography and videography

needs.





**ABBREVIATIONS**

1.

HTML: Hypertext Markup Language

2.

CSS : Cascading Style Sheets

3.

JS : JavaScript

4.

PHP : Hypertext Preprocessor

5.

SQL : Structured Query Language

6.

ER Diagram : Entity

-

Relationship

Diagram

7.

DFD : Data Flow Diagram

8.

DBMS : Database Management System

9.

API : Application Programming Interface

10.

GUI : Graphical User Interface

11.

URL : Uniform Resource Locator

12.

HTTP : Hypertext Transfer Protocol

13.

CRUD : Create, Read, Update, Delete

14.

JSON : JavaScript Object Notation

15.

IDE : Integrated Development Environment

16.

PG : Paying Guest

17.

UI : User Interface

18.

UX : User Experience

19.

DB : Database

20.

SSL : Secure Sockets Layer

**ABSTRACT**

The

**Camera Project**

is a web

-

based platform designed for managing camera rentals

and purchases. It allows users to browse, select, and book cameras with ease. The

system provides essential features such as

**user authentication, product catalog**

**management, and order processing.**

The project is developed using modern web technologies to ensure a smooth and

responsive user experience. It incorporates a

**secure payment gateway**

for hassle

-

free

transactions. The

**admin panel**

enables administrators to manage inventory, update

listings, and track user orders efficiently.

A structured database is implemented to store user details, orders, and camera

information securely. The project follows the

**MVC architecture**

to separate logic and

enhance maintainability.

The main goal of this project is to create a

**user**

**-**

**friendly and scalable**

solution for

individuals and businesses looking to rent or purchase cameras online. By offering an

efficient and secure platform, the system improves accessibility to high

-

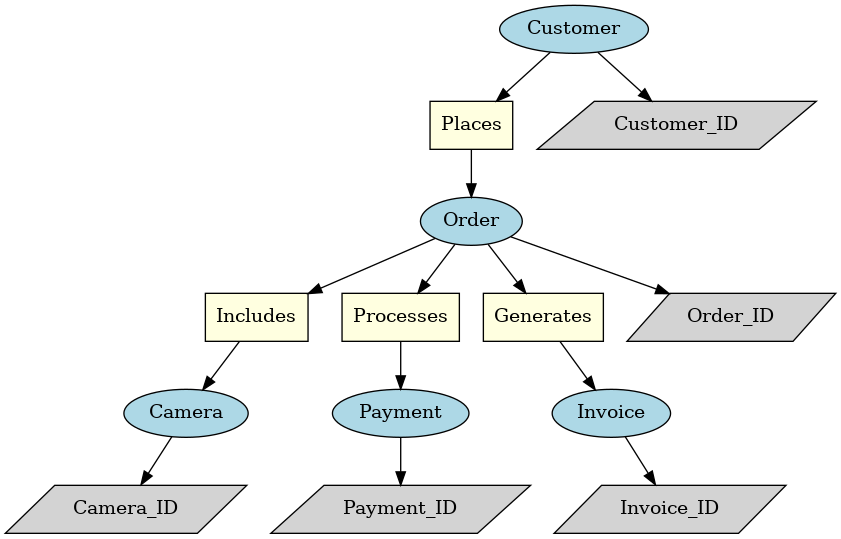
quality camera

equipment.

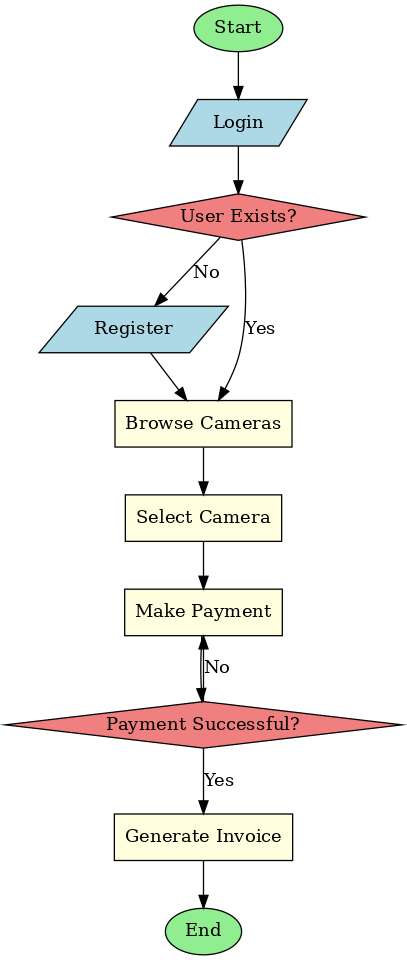
**E**

**-**

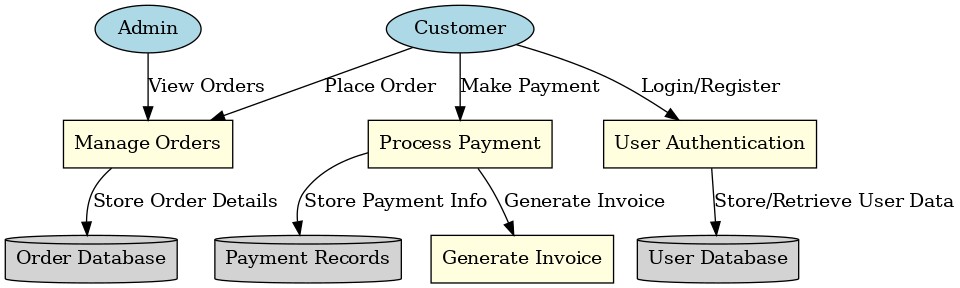
**R Diagram**



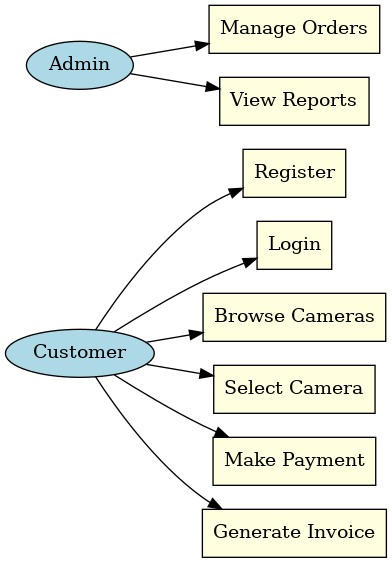
**Flowchart**



**Dataflow Diagram**



**Use case Diagram**



**Tools and Technologi**

**es**

1.

**Python**

–

A powerful, high

-

level programming language used for backend development.

2.

**Flask**

–

A lightweight Python web framework used for building the web application.

3.

**HTML (HyperText Markup Language)**

–

The standard language for structuring web

pages.

4.

**CSS (Cascading Style Sheets)**

–

Used to style and design the web pages for a better user

experience.

5.

**JavaScript**

–

A scripting language that enhances interactivity and functionality on the

frontend.

6.

**SQL (Structured Query Language)**

–

Used for managing and querying the project’s

database.

7.

**SQLite**

–

A lightweight relational database used to store user and camera

-

related data.

8.

**Jinja2**

–

A templating engine for Flask that dynamically generates HTML content.

9.

**Bootstrap**

–

A frontend framework used to design responsive and mobile

-

friendly web

pages.

10.

**WTForms**

–

A Flask extension used for handling and validating user input forms.

11.

**REST APIs**

–

Used for communication between the frontend and backend, ensuring smooth

data exchange.

12.

**JSON (JavaScript Object Notation)**

–

A lightweight data format used for data transmission

between the server and client.

**SOURCE CODE**

**1**

**APP.PY FILE**

from flask import Flask, render\_template, request, redirect, url\_for, session, flash

from flask\_dance.contrib.google import make\_google\_blueprint, google

from flask\_login import LoginManager, UserMixin,

login\_user, logout\_user,

login\_required

from flask\_mail import Mail, Message

from datetime import datetime

import mysql.connector

import os

from openai import OpenAI

from fpdf import FPDF

from flask\_session import Session

import stripe

app = Flask(\_\_name\_\_)

#

# Setting up SQL database URI

db\_host='gajanan.mysql.pythonanywhere

-

services.com'

db\_user='gajanan'

db\_password='gaju1234?'

db\_name='gajanan$default'

#

create table register

# Function to connect to the MySQL database

def get\_db\_connection():

conn = mysql.connector.connect(

host=db\_host,

user=db\_user,

password=db\_password,

database=db\_name

)

return conn

# Create the database table if it doesn't exist

def create\_table():

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS register(

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(20) NOT NULL,

email VARCHAR(30) NOT NULL,

password VARCHAR(255) NOT NULL,

number VARCHAR(12) NOT NULL,

profile\_photo VARCHAR(150),

google\_id VARCHAR(50) DEFAULT NULL

--

New column for Google ID

)

;

''')

cursor.execute('''

CREATE TABLE IF NOT EXISTS orders (

id

INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

number VARCHAR(20),

email VARCHAR(100),

order\_date DATE,

return\_date DATE,

address TEXT,

price\_per\_day FLOAT,

total\_amount FLOAT,

total\_days int,

item\_name VARCHAR(100),

payment\_method VARCHAR(20),

user\_id INT,

FOREIGN KEY (user\_id) REFERENCES register(id)

)

;

''')

conn.commit()

conn.close()

# Configuring Flask

-

M

ail

app.config['MAIL\_SERVER'] = 'smtp.gmail.com'

app.config['MAIL\_PORT'] = 465

app.config['MAIL\_USERNAME'] = 'gajanan19022000@gmail.com'

# Your Gmail

address

app.config['MAIL\_PASSWORD'] = 'jost higa wtmc wzlr'

# Your App Password

app.config['MAIL\_USE\_TLS'] = False

app.config['MAIL\_USE\_SSL'] = True

#

Google Login Route

@app.route("/google")

def google\_login():

if not google.authorized:

return redirect(url\_for("google.login"))

resp = google.get("/oauth2/v2/userinfo")

user\_info = resp.json()

email = user\_info.get("email")

name = user\_info.get("name")

profile\_photo\_url = user\_info.get("picture")

# Google profile photo URL

google\_id = user\_info.get("id")

# Unique Google ID

if not email:

flash("Google login failed: No email provided", "error")

return redirect("/login")

conn = get\_db\_connection()

cursor = conn.cursor(buffered=True)

try:

# Check if the user exists by email

cursor.execute("SELECT id, profile\_photo FROM register WHERE email = %s",

(

email,

))

existing\_user = cursor.fetchone()

if existing\_user:

user\_id = existing\_user[0]

print(" User already exists. Logging in.")

else:

#

New user: Insert into

`register` table

cursor.execute("""

INSERT INTO register (username, email, password, number,

profile\_photo,google\_id)

VALUES (%s, %s, %s, %s, %s, %s)

""", (name, email, "google\_auth", "N/A", None, google\_id))

# Profile photo will

be updated later

conn.commit()

user\_id = cursor.lastrowid

# Get newly inserted user ID

print(" New Google user inserted.")

except mysql.connector.Error as err:

print("

MySQL Error:", err)

flash("Database error occurred. Try again later.", "error")

return redirect("/login")

finally:

cursor.close()

conn.close()

return('/')

@app.route('/')

def index():

return render\_template("order1.html")

@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()

# Check if the username and password match any record in the register table

cursor.execute("SELECT \* FROM register WHERE username = %s AND password

=

%s", (username, password

))

user = cursor.fetchone()

# Fetch one record

conn.close()

if user:

# If a

user is found

session['user\_id'] = user[0]

# Store the user ID in session

return redirect('/')

# Assuming you have a home route

else:

flash("User not found. Please register first!", 'error')

return redirect('/login')

return render\_template("login.html")

@app.route('/check\_user')

def check\_user():

# Check if user is logged in by checking session

if 'user\_id

' in session:

# If user\_id exists in session, they are logged in

return jsonify({'logged\_in': True})

else:

return jsonify({'logged\_in': False})

@app.route('/register', methods=['GET', 'POST'])

def register():

if request.method == 'POST':

username = request.form['username']

password = request.form['password']

email = request.form['email']

number = request.form['number']

conn = get\_db\_connection()

cursor = conn.cursor()

# C

heck if the username already exists

cursor.execute("SELECT \* FROM register WHERE username = %s", (username,))

existing\_user = cursor.fetchone()

# Fetch one result

if existing\_user:

conn.close()

return "Username already exists!"

cursor.execute("INSERT INTO register (username, password, email, number)

VALUES (%s, %s, %s, %s)",

(

username, password, email, number

))

conn.commit()

conn.close()

return redirect('/log

in')

return render\_template('login.html')

@app.route('/profile')

def profile():

if 'user\_id' not in session:

return redirect('/login')

# Get the user data from the database using the user\_id from the session

conn = get\_db\_connection()

cursor = conn.cursor()

# Fetch user details from the 'register' table using the session's user\_id

cursor.execute("SELECT \* FROM register WHERE id = %s", (session['user\_id'],))

user = cursor.fetchone()

if user is None:

return redirect('/login')

# If no user found, redirect to login

# Fetch user's orders from the 'orders' table

cursor.execute("SELECT \* FROM orders WHERE user\_id = %s",

(

session['user\_id'],

))

user\_orders = cursor.fetchall()

# Fetch all orders for the user

conn.close()

# Map user tuple to field names for easy access in the template

user\_info = {

'id': user[0],

'username': user[1],

'email': user[2],

'password': user[3],

# Adjust this

index if the password field is in a different place

'number': user[4],

'profile\_photo': user[5]

}

# Render the profile page and pass user info and orders to the template

return render\_template('profile.html')

# upload\_photo file has an allowed extension

@app.route('/upload\_photo', methods=['POST'])

def upload\_photo():

if 'user\_id' not in session:

return redirect('/login')

if 'file' not in request.files:

flash("No file uploaded", "error")

retu

rn redirect('/profile')

file = request.files['file']

if file.filename == '':

flash("No selected file", "error")

return redirect('/profile')

if file:

filename = secure\_filename(file.filename)

file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

file.save(file\_path)

#

Save file in `static/uploads/`

#

Store the filename (not full path) in the database

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute("UPDATE register SET profile\_photo = %s WHERE id = %s",

(

filename, session['user\_id'

]))

conn.commit()

conn.close()

return redirect('/profile')

@app.route('/logout')

def logout():

session.pop('user\_id', None)

return redirect('/login')

stripe.api\_key =

"sk\_test\_51QreFNJRU2FVlSQlyKuQG297bHhNDfspewEbo0p7KQJ3ehfEfK1mMsh05

MfokPtl0wPSznQ5ecFrIti7pEou5XjQ00vLewlHRG"

@app.route('/place\_order', methods=["POST"])

def place\_order():

if 'user\_id' not in session:

return redirect('/login')

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute("SELECT \* FROM register WHERE id = %s", (session['user\_id'],))

use

r = cursor.fetchone()

if user is None:

conn.close()

return redirect('/login')

name = request.form['name']

number = request.form['number']

email = request.form['email']

order\_date\_str = request.form['order\_date']

return\_date\_str = request.form['return\_date']

address = request.form['address']

payment\_method = request.form['payment\_method']

price\_per\_day = float(request.form['price\_per\_day'])

item\_name = request.form['item\_name']

order\_date = datetime.strptime(order\_date\_str, '%Y

-

%m

-

%d').date()

return\_date = datetime.strptime(return\_date\_str, '%Y

-

%m

-

%d').date()

total\_days = max((return\_date

-

order\_date).days, 1)

total\_amount = total\_days \* price\_per\_day

if payment\_method == "Cards":

session['order\_data'] = {

"name": name,

"number": number,

"email": email,

"order\_date": order\_date\_str,

"return\_date": return\_date\_str,

"address": address,

"price\_per\_day": price\_per\_day,

"total\_days": total\_days,

"total\_amount": total\_amount,

"item\_name": item\_name,

"user\_id": user[0]

}

return redirect(url\_for('stripe\_payment'))

cursor.execute('''

INSERT INTO orders (name, num

ber, email, order\_date, return\_date, address,

price\_per\_day, total\_days, total\_amount, item\_name, payment\_method, user\_id)

VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)

''', (name, number, email, order\_date, return\_date, address, price\_per\_day, total\_days,

total\_amount, item\_name, payment\_method, user[0]))

conn.commit()

conn.close()

#

Prepare order confirmation email

from pdf import hi

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute("""

SELECT id,name, number, item\_name, total\_amount, order\_date, return\_date,

payment\_method, address

FROM orders

ORDER BY id DESC LIMIT 1

""")

last\_order = cursor.fetchone()

# Fetch the last order r

pdf\_file = hi(last\_order

[0],(

session['user\_id']) )# Path to the generated PDF file

subject = "Order Confirmation"

message\_body = f"""Hello {name},

ORDER SUCCESSFUL

msg = Message(subject=subject, sender=app.config['MAIL\_USERNAME'],

recipients=[email])

msg.body = message\_body

# Attach the PDF file to the email

msg.attach(filename="order\_invoice.pdf", content\_type="application/pdf",

data=pdf\_content)

# Send the email

mail.send(msg)

return """

<

script

>

alert("Order Succes

sful! Your order has been placed. Check your email for

details.");

window.location.href = "/";

<

/script

>

"""

@app.route('/send\_contact', methods=["POST"])

def send\_email():

name = request.form['name']

email = request.form['email']

subject = request.form['subject']

message = request.form['message']

phone = request.form['phone']

# Create email message

msg = Message(subject=f"New Contact Messge from {name}: {subject}",

# Include

name in subject

sender=app.config['MAIL\_USERNAME'],

# Must be your verified

sender email

recipients=['gajanan19022000@gmail.com'])

# Use your email as

recipient

# Include user details in the body of the email

msg.body = f"Message from :

-

{

name

}

\

n

\

n Email:

-

{

email

}

\

n

\

Requirment:

-

{

subject

}

\

n

\

nmessage:

-

{

message

}

\

n

\

nNumber:

-

{

phone}"

# Optionally set the reply

-

to header to the user's email

msg.reply\_to = email

mail.send(msg)

return redirect('/')

@app.route('/blog')

def blog():

return render\_template("blog.html")

@app.route('/garelly')

def garelly():

return render\_template("garelly.html")

@app.route("/contact", methods=["GET"])

def contact():

conn = get\_db\_connection()

cursor = conn.cursor()

# Retrieve all ratings from the database

cursor.execute("SELECT \* FROM ratings")

ratings = cursor.fetchall()

# Retrieve all feedback from the database

cursor.execute("SELECT \* FROM feedback")

feedback =

cursor.fetchall()

# Close the database connection

cursor.close()

conn.close()

# Pass ratings to the template (even if empty)

return render\_template('contact.html', ratings=ratings )

@app.route('/blogs/<blog\_id>')

def get\_blog(blog\_id):

# Renders the combined blogs file and passes the blog\_id

return render\_template('blogs.html', blog\_id=blog\_id)

client = OpenAI(api\_key="sk

-

proj

-

7

UjyhPi6vSUk7Xkjtg7wEBO7xYu\_lZaR3tRmYzCk

-

Ja

-

0

JTAgYSo8f5SoBDEkOzD0VRTRrDKBnT3BlbkFJj

5

-

wO9YR9hQIzBQyJ1GLdVLTeHzvUG3B\_Ss2CKyGmMuMH32YP9ES44I9sgS6H5xO

HQ864jAbgA")

# Route to save user message & AI response

@app.route("/chat", methods=["POST"])

def chat():

data = request.json

user\_message = data.ge

t("message", "")

if not user\_message:

return jsonify({"error": "Message is required"}), 400

try:

# Get AI response

completion = client.chat.completions.create(

model="gpt

-

4

o

-

mini",

messages=[{"role": "user", "content": user\_message}]

)

ai\_response = completion.choices[0].message.content

# Save to MySQL

conn = get\_db\_connection()

cursor = conn.cursor()

query = "INSERT INTO chat\_history (user\_message,

ai\_response) VALUES (%s,

%s)"

cursor.execute(query, (user\_message, ai\_response))

conn.commit()

return jsonify({"response": ai\_response})

except Exception as e:

return jsonify({"error": str(e)}), 500

if \_\_name\_\_ == '\_\_main\_\_':

create\_table()

**1**

**Login**

**.html**

<!

DOCTYPE html

>

<

html

>

<

head

>

<

title>Login/Register Form</title

>

<

link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/font

-

awesome/6.0.0

-

beta3/css/all.min.css">

<!

--

Bootstrap CSS

--

>

<

link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0

-

alpha3/dist/css/bootstrap.min.css

"

rel="stylesheet">

<!

--

FontAwesome CSS

--

>

<

link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font

-

awesome/6.0.0

-

beta3/css/all.min.css">

<

link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font

-

awesome/6.0.0

-

beta3/css/all.min.css">

<

style

>

/\* Full

-

screen canvas for the fireworks background \*/

canvas {

position: fixed;

z

-

index:

-

1

;

}

body {

background: url('https://wallpapercave.com/wp/1C6G3g0.jpg') ;

background

-

color: #f3e0e2;

display: flex;

justify

-

content: center;

align

-

items: center;

height: 100vh;

font

-

family: 'Montserrat', sans

-

serif;

margin:

-

20

px 0 50px;

}

.form

-

container {

background

-

color: #fff;

box

-

shadow: 0 10px 28px black, 0 10px 10px yellow;

border

-

radius: 15px;

display: flex;

width: 768px;

height: 480px;

}

.form

-

left, .form

-

right {

width: 50%;

padding: 30px;

display: flex;

flex

-

direction: column;

justify

-

content: center;

}

.form

-

right {

background: linear

-

gradient(to right, #FF4B2B, #FF416C);

color: #fff;

text

-

align: center;

border

-

radius: 0 15px 15px 0;

position: relative;

}

.social

-

icons a {

border

-

radius: 50%;

display: inline

-

flex;

justify

-

content: center;

align

-

items: center;

margin: 5px;

height: 45px;

width: 45px;

text

-

decoration: none;

background: linear

-

gradient(to right, #DB4437 25%, #F4B400 25%, #F4B400 50%,

#0F9D58 50%, #0F9D58 75%, #4285F4 75%);

background

-

clip: text;

-

webkit

-

background

-

clip: text;

-

webkit

-

text

-

fill

-

color: transparent;

box

-

shadow: 0 10px 28px black, 0 10px 10px yellow;

}

.form

-

title {

margin

-

bottom: 20px;

}

.form

-

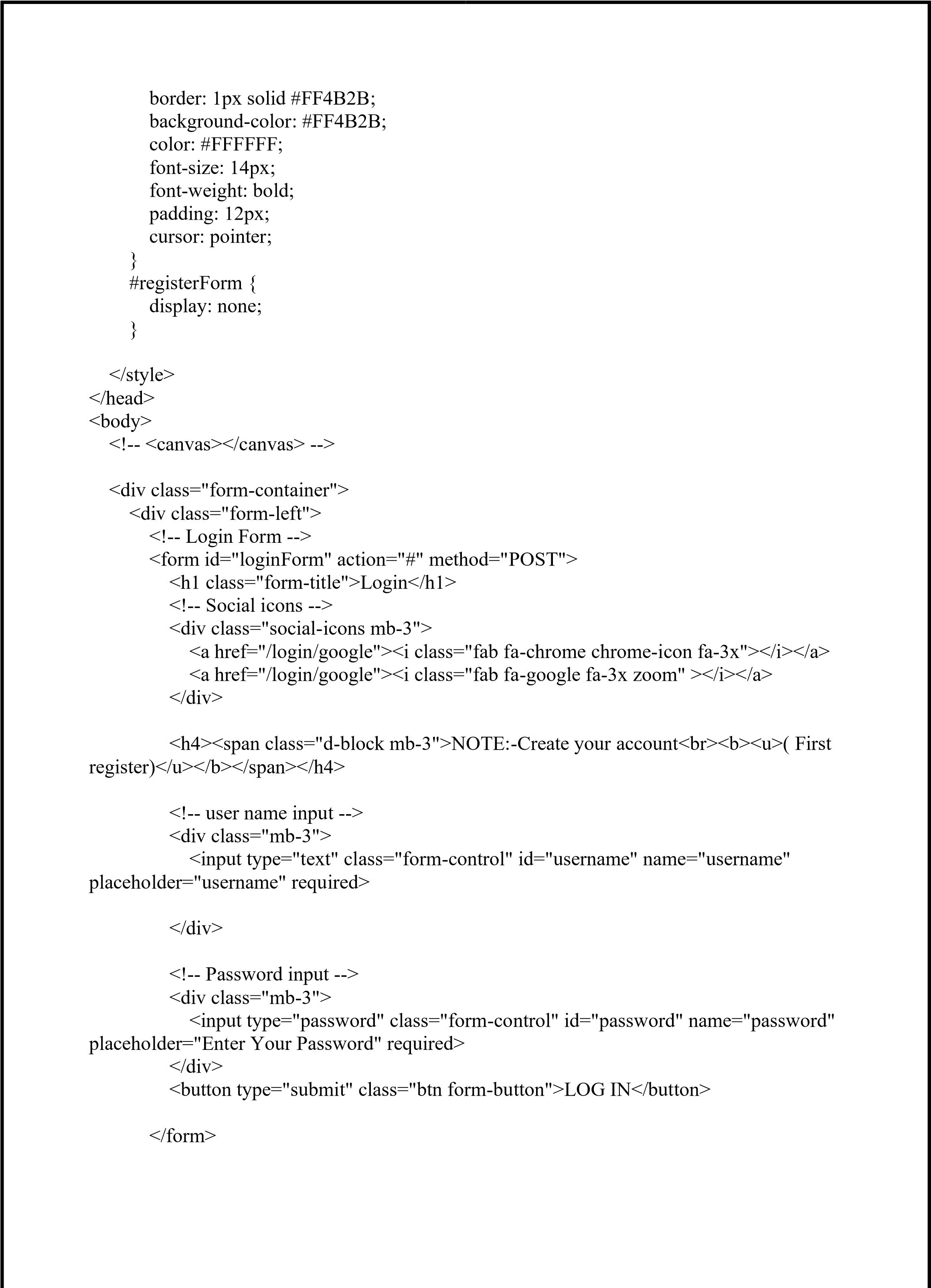
button {

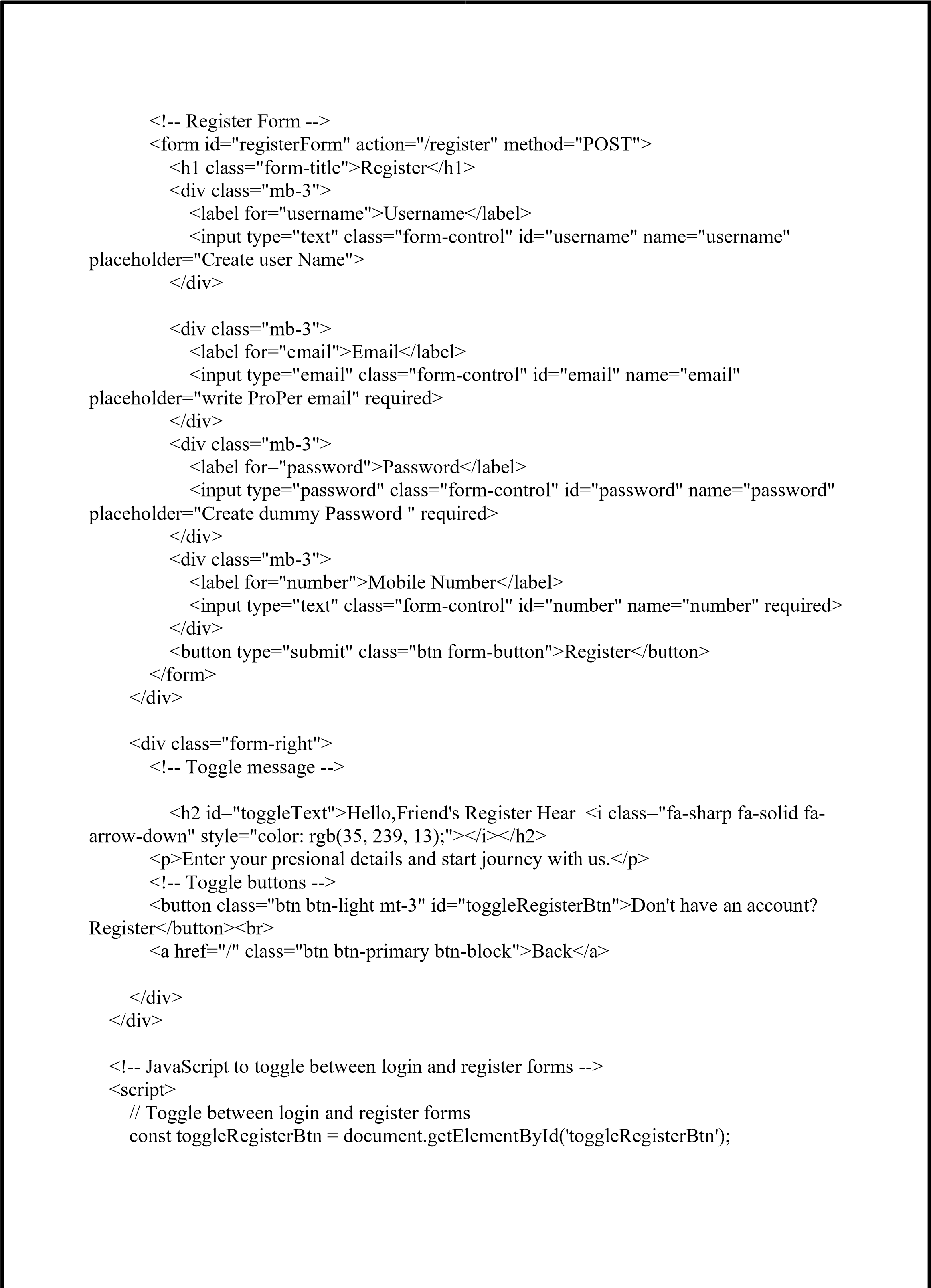
width: 100%;

border

-

radius: 20px;





const loginForm = document.getElementById('loginForm');

const registerForm = document.getElementById('registerForm');

const toggleText = document.getElementById('toggleText');

// Ensure register form is hidden on load

registerForm.style.display = 'none';

toggleRegisterBtn.addEventListener('click', () => {

if (registerForm.style.display === 'none') {

registerForm.style.display = 'block';

loginForm.style.display = 'none';

toggleText.innerText = 'Register Form';

toggleRegisterBtn.innerText

=

'Already have an account? Login';

}

else

{

registerForm.style.display = 'none';

loginForm.style.display = 'block';

toggleText.innerText = 'Login Form';

toggleRegisterBtn.innerText = "Don't have an account? Register";

}

})

;

// Flash message handling

const flashMessages = JSON.parse('{{

get\_flashed\_messages(with\_categories=true)|tojson|safe }}');

if (Array.isArray(flashMessages) && flashMessages.length > 0) {

flashMessages.forEach(message => {

alert(message[1]); // Display each flash message using alert()

})

;

}

<

/script

>

<!

--

Fireworks Animation Script

--

>

<

script

>

var t = 0;

var c = document.querySelector("canvas");

var $ = c.getContext('2d');

c.width = window.innerWidth;

c.height = window.innerHeight;

$.fillStyle = 'hsla(0,0%,0%,1)';

window.addEventListener('resize', function() {

c.width = window.innerWidth;

c.height = window.innerHeight;

}

, false);

function draw() {

$.globalCompositeOperation = 'source

-

over';

$.fillStyle = 'hsla(0,0%,0%,.05)';

$.fillRect(0, 0, c.width, c.height);

var foo, i, j, r;

foo = Math.sin(t) \* 0.8 \* Math.PI;

for (i = 0; i < 500; ++i) {

r = 1100 \* Math.sin(i \* foo);

$.globalCompositeOperation = '';

$.fillStyle = 'hsla(' + i + 12 + ',100%, 60%,1)';

$.beginPath();

$.arc(Math.sin(i) \* r + (c.width / 2),

Math.cos(i) \* r + (c.height / 2),

1.5

, 0, Math.PI \* 2);

$.fill();

}

t += 0.000005;

return t %= 2 \* Math.PI;

}

function run() {

window.requestAnimationFrame(run);

draw();

}

run();

<

/script

>

<!

--

Bootstrap JS (Optional for modal, dropdowns, etc.)

--

>

<

script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0

-

alpha3/dist/js/bootstrap.bundle.min.js"></script

>

<

/body

>

<

/html

>

**2**

**Home.html**

<!

DOCTYPE html

>

<

html lang="en"

>

<

head> <meta name="viewport" content="width=device

-

width, initial

-

scale=1.0">

<

link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css

"

rel="stylesheet">

<

script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">

<

/script

>

<!

--

Bootstrap Icons CSS

--

>

<

link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css

"

rel="stylesheet" integrity="sha384

-

QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+A

LEwIH" crossorigin="anonymous">

<!

--

Font Awesome CSS

--

>

<

link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font

-

awesome/6.0.0

-

beta3/css/all.min.css">

<

title>amera</title

>

<

style

>

.section {

animation: slideDown 1.6s ease

-

out;

}

@keyframes slideDown {

from {

transform: translateY(

-

100%)

;

opacity: 0;

}

to {

transform: translateY(0);

opacity: 1;

}

}

.navbar {

animation: slideDown 1.1s ease

-

out;

}

.navbar

-

nav .nav

-

link {

position: relative;

font

-

size: 18px;

}

.navbar

-

nav .nav

-

link::after {

content: "";

position: absolute;

background: #295a8d;

height: 3px;

width: 0;

left: 0;

bottom:

-

10

px;

transition: 0.3s;

}

.navbar

-

nav .nav

-

link:hover::after {

width: 100%;

}

@keyframes fadeInUp {

from {

transform: translateY(17px);

opacity: 0;

}

to{

transform: translateY(0);

opacity: 1;

}

}

#home {

animation: fadeInUp 1s ease

-

out;

}

.custom

-

btn {

width: 180px; /\* Adjust width as needed \*/

padding: 15px 30px;

font

-

size: 14px;

letter

-

spacing: 2px;

border

-

radius: 50px;

}

.card

-

img

-

container {

overflow: hidden;

}

.card

-

img

-

top.zoom {

transition: transform 0.2s;

}

.card

-

img

-

top.zoom:hover {

transform: scale(1.1);

z

-

index: 10;

position: relative;

}

@keyframes fadeInUp {

from {

transform: translateY(50px);

opacity: 0;

}

to {

transform: translateY(0);

opacity: 1;

}

}

@keyframes fadeOutDown {

from {

transform: translateY(0);

opacity: 1;

}

to {

transform: translateY(50px);

opacity: 0;

}

}

.card {

opacity: 0;

transform: translateY(20px);

transition: transform 1s ease

-

out, opacity 1s ease

-

out;

}

.card.show {

animation: fadeInUp 2s ease

-

out forwards;

}

.card.hide {

animation: fadeOutDown 2s ease

-

out forwards;

}

.map

-

responsive {

overflow: hidden;

padding

-

bottom: 56.24%;

position: relative;

height: 0;

}

.map

-

responsive iframe {

left: 0;

top: 0;

height: 200px;

width: 100%;

position: absolute;

}

@keyframes slideInRight {

from {

transform: translateX(100%);

opacity: 0;

}

to {

transform: translateX(0);

opacity: 1;

}

}

/\* Floating Button \*/

#chatButton {

position: fixed;

bottom: 40px;

right: 40px;

background

-

color: blueviolet;

color: white;

border: none;

border

-

radius: 50%;

width: 45px;

height: 45px;

font

-

size: 20px;

cursor: pointer;

box

-

shadow: 0 2px 10px rgba(0, 0, 0, 0.2);

z

-

index: 1000;

}

/\* Chat Container \*/

#chatContainer {

width: 290px;

position: fixed;

bottom: 120px;

right: 25px;

display: none;

z

-

index: 1000;

background

-

color: aqua;

}

#chatBox {

max

-

height: 400px;

overflow

-

y: auto;

}

<

/style

>

<

/head

>

<

body

>

<!

--

Floating Button

--

>

<

button id="chatButton" class="btn btn

-

primary">

<!

--

<

i class="fas fa

-

comments"></i>

--

>

<

i class="fa

-

brands fa

-

rocketchat"></i>

<

/button

>

<!

--

Chat Container

--

>

<

div id="chatContainer" class="card"

>

<

div id="chatBox" class="card

-

body"></div>

<

div class="card

-

footer">

<

div class="input

-

group">

<

input type="text" id="userInput" class="form

-

control"

placeholder="Type your message..." />

<

div class="input

-

group

-

append">

<

button id="sendButton" class="btn btn

-

primary">Send</button>

<

/div

>

<

/div

>

</div>

</div>

<!-- login -->

<div class="modal fade" id="loginModal" tabindex="-1" arialabelledby="loginModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="loginModalLabel">Login</h5>

<button type="button" class="btn-close" data-dismiss="modal" arialabel="Close"></button>

</div>

<div class="modal-body">

<form id="loginForm" action="login.php" method="post">

<div class="form-group mb-3">

<label for="email">Email address</label>

<input type="email" class="form-control" placeholder="Enter email" required name="email">

</div>

<div class="form-group mb-3">

<label for="password">Password</label> <input type="password" class="form-control" placeholder="Password" required name="password">

</div>

<button type="submit" class="btn btn-primary">Login</button>

</form>

</div>

</div>

</div>

</div>

<!-- //////////////////////////////////////////////////////////////////////////////////////////////////////// -->

<!-- Registration Modal -->

<div class="modal fade" id="registerModal" tabindex="-1" arialabelledby="registerModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="registerModalLabel">Register</h5> <button type="button" class="btn-close" data-dismiss="modal" arialabel="Close"></button>

</div>

<div class="modal-body">

<form id="registerForm" action="Registration.php" method="post">

<div class="form-group mb-3">

<label for="username">Username</label>

<input type="text" class="form-control" placeholder="Enter username" required name="username">

</div>

<div class="form-group mb-3">

<label for="email">Email address</label>

<input type="email" class="form-control" placeholder="Enter email" required name="email">

</div>

<div class="form-group mb-3">

<label for="password">Password</label> <input type="password" class="form-control"

placeholder="Password" required name="password">

</div>

<div class="form-group mb-3">

<label for="confirmPassword">Confirm Password</label>

<input type="password" class="form-control" placeholder="Confirm Password" required name="conformpassword">

</div>

<button type="submit" class="btn btn-primary btn-block mt3">Register</button>

</form>

</div>

<div class="modal-footer">

<small>Already have an account? <a href="#" data-bs-dismiss="modal" data-bs-toggle="modal" data-bs-target="#loginModal">Sign in</a></small>

</div>

</div>

</div>

</div>

<!-- //////////////////////////////////////////////////////////////////////////////////////////// --> <!-- nav bar -->

<nav class="navbar navbar-expand-lg navbar-light bg-primary sticky-top shadow-sm p-0">

<div class="container-fluid">

<i class="fa fa-camera " aria-hidden="true" style="width: 30px; font-size: xlarge;"></i>

<p class="navbar-brand nav-link text-white">Portrait Photography</p> <button class="navbar-toggler " type="button" data--toggle="collapse" databs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" arialabel="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ms-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link" href="#home">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#nikon">Nikon</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#Canon">Canon</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#contact">Contact</a>

</li>

<li class="nav-item">

<!-- <li class="nav-item"><a href="#" class="nav-link text-white"

id="signInLink" data-bs-toggle="modal" data-bstarget="#loginModal">Login</a></li> -->

<li class="nav-item"><a href="#" class="nav-link " id="signInLink" data-bs-toggle="modal" data-bs-target="#loginModal">Login</a></li>

</li>

<li class="nav-item">

<!-- <a href="#" class="nav-link" id="signUpLink" data-bstoggle="modal" data-bs-target="#registerModal">Registation</a> -->

<li class="nav-item"><a href="#" class="nav-link" id="signUpLink" databs-toggle="modal" data-bs-target="#registerModal">Registration</a></li>

</li>

</ul>

</div>

</div>

</nav>

<!-- ///////////////////////////////////section 1 //////////////////////////////////////////////////////// -->

<section id="home">

<div class="container-fluid position-relative p-0">

<div class="row no-gutters">

<div class="col-12">

<img src="https://cdn.pixabay.com/photo/2016/10/25/17/00/camera1769414\_1280.jpg" alt="img" class="img-fluid w-100">

</div>

</div>

</div>

</section>

<!-- ///////////////////////////// ////////////////// section 2 ////////////////////////////////// -->

<div id="nikon">

<div>

<h1 class="mx-2 pt-2">Nikon</h1>

<hr class="bg-white">

<!-- <!-/////////////////////////////////////////////////////////////////////////- -->

<section id="collection">

<div class="container-fluid">

<div class="row">

<!-- Card 1 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img src="https://1.bp.blogspot.com/-

uS1e\_yoyopw/XqLkiVQBirI/AAAAAAAAEOk/b3ZwgoNNAIYVij6RkOfYQxFZ7DZ719jgCLcBGAsYHQ/w0/drone-camera-4k-

techyleaf.jpeg" class="card-img-top zoom" alt="Clothing Collections">

</div>

<div class="card-body">

<p class="card-text">Clothing Collections</p>

<p>Price per Day: $10</p>

<button id="addDay1" class="btn btn-success">Add Day</button>

<button id="removeDay1" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity1">0</span></p>

<p class="total">Total: $<span id="total1">0</span></p>

<button id="orderNow1" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

<!-- Card 2 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img

src="https://d3ekkp2oigezer.cloudfront.net/business/4902/blogs/4620/images/cama ras-fotograficas-2021-7.jpg" class="card-img-top zoom" alt="Shoes Spring">

</div>

<div class="card-body">

<p class="card-text">Shoes Spring</p>

<p>Price per Day: $10</p>

<button id="addDay2" class="btn btn-success">Add Day</button>

<button id="removeDay2" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity2">0</span></p>

<p class="total">Total: $<span id="total2">0</span></p> <button id="orderNow2" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

<!-- Card 3 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img

src="https://th.bing.com/th/id/OIP.ej4t4vjK13JYgCRfbImHUAHaE7?rs=1&pid=I mgDetMain" class="card-img-top zoom" alt="Shoes Spring">

</div>

<div class="card-body">

<p class="card-text">Shoes Spring</p>

<p>Price per Day: $10</p>

<button id="addDay3" class="btn btn-success">Add Day</button> <button id="removeDay3" class="btn btn-danger">Remove Day</button>

<p>Day: <span id="quantity3">0</span></p>

<p class="total">Total: $<span id="total3">0</span></p> <button id="orderNow3" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

<!-- Card 4 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img src="https://www.imagesredefined.in/wp-

content/uploads/2018/09/wedding-photography-tips.jpg" class="card-img-top zoom" alt="Shoes Spring">

</div>

<div class="card-body">

<p class="card-text">Shoes Spring</p>

<p>Price per Day: $10</p>

<button id="addDay4" class="btn btn-success">Add Day</button>

<button id="removeDay4" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity4">0</span></p>

<p class="total">Total: $<span id="total4">0</span></p> <button id="orderNow4" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- Canon Section -->

<section id="Canon">

<h1 class="mx-5 pt-5">Canon</h1>

<hr class="bg-white">

<div class="containe-fluidr">

<div class="row">

<!-- Card 5 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img src="https://th.bing.com/th/id/OIP.c1IVQowd-

DRX26opjic9KQHaEc?rs=1&pid=ImgDetMain" class="card-img-top zoom" alt="Accessories">

</div>

<div class="card-body">

<p class="card-text">Clothing Collections</p>

<p>Price per Day: $120</p>

<button id="addDay5" class="btn btn-success">Add Day</button> <button id="removeDay5" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity5">0</span></p>

<p class="total">Total: $<span id="total5">0</span></p> <button id="orderNow5" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

<!-- Card 6 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img src="https://i.ytimg.com/vi/UT0zl\_GkFJg/maxresdefault.jpg" class="card-img-top zoom" alt="Shoes Spring">

</div>

<div class="card-body">

<p class="card-text">Shoes Spring</p>

<p>Price per Day: $10</p>

<button id="addDay6" class="btn btn-success">Add Day</button> <button id="removeDay6" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity6">0</span></p>

<p class="total">Total: $<span id="total6">0</span></p> <button id="orderNow6" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

<!-- Card 7 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img src="https://th.bing.com/th/id/OIP.QoDeuRRUUPZ8TF\_lOppWQAAAA?rs=1&pid=ImgDetMain" class="card-img-top zoom" alt="Shoes Spring">

</div>

<div class="card-body">

<p class="card-text">Shoes Spring</p>

<p>Price per Day: $10</p>

<button id="addDay7" class="btn btn-success">Add Day</button> <button id="removeDay7" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity7">0</span></p>

<p class="total">Total: $<span id="total7">0</span></p> <button id="orderNow7" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button>

</div>

</div>

</div>

<!-- Card 8 -->

<div class="col-md-3 container py-4">

<div class="card">

<div class="card-img-container">

<img src="https://th.bing.com/th/id/OIP.HwuA2gygMOv9FY6uKAfKwHaEO?rs=1&pid=ImgDetMain" class="card-img-top zoom" alt="Shoes Spring">

</div>

<div class="card-body">

<p class="card-text">Shoes Spring</p>

<p>Price per Day: $10</p>

<button id="addDay8" class="btn btn-success">Add Day</button>

<button id="removeDay8" class="btn btn-danger">Remove

Day</button>

<p>Day: <span id="quantity8">0</span></p>

<p class="total">Total: $<span id="total8">0</span></p>

<button id="orderNow7" class="btn btn-primary" data-toggle="modal" data-target="#orderModal" type="submit">ORDER NOW</button>

<!-- <button id="orderNow8" class="btn btn-primary" datatoggle="modal" data-target="#orderModal">ORDER NOW</button> -->

</div>

</div>

</div>

</div>

</div>

</section>

<!-- ////////////////////////////////////////////////////////////////////////////////////////// -->

<!-- Modal -->

<div class="modal fade" id="orderModal" tabindex="-1" role="dialog" arialabelledby="orderModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="orderModalLabel">Order Details</h5> <button type="button" class="btn-close" data-dismiss="modal" arialabel="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form id="orderForm" method="POST" action="order.php">

<!-- Form fields -->

<div class="mb-3">

<label for="name" class="form-label">Name</label>

<input type="text" class="form-control" id="name" name="name" required>

</div>

<div class="mb-3">

<label for="mobile" class="form-label">Mobile No</label> <input type="number" class="form-control" id="mobile" name="mobile" required>

</div>

<div class="mb-3">

<label for="email" class="form-label">Email</label>

<input type="email" class="form-control" id="email" name="email" required>

</div>

<div class="mb-3">

<label for="order" class="form-label">Order Date</label>

<input type="date" class="form-control" id="order" name="order" required>

</div>

<div class="mb-3">

<label for="return" class="form-label">Return Date</label> <input type="date" class="form-control" id="return" name="return" required>

</div>

<div class="mb-3">

<label for="address" class="form-label">Address</label> <textarea class="form-control" id="address" name="address" rows="3" required></textarea>

</div>

<div class="mb-3">

<label for="paymentMethod" class="form-label">Payment

Method</label>

<select class="form-control" id="paymentMethod" name="paymentMethod" required>

<option value="card">Cash On Delivery</option>

<option value="phonepe">PhonePe</option>

</select>

</div>

<div class="mb-3">

<p>Days: <span id="orderDays">0</span></p>

<p>Total: $<span id="orderTotal">0</span></p>

</div>

<button type="submit" class="btn btn-primary">Submit Order</button>

</form>

</div>

</div>

</div>

</div>

<!-- //////////////////////////////////////////////////////////////////////////// -->

<!-- Contact Section -->

<section class="py-5" id="contact">

<div class="container">

<h1 class="text-center mb-4">Contact Us</h1>

<div class="row">

<div class="col-md-6">

<form id="contactForm" method="post" action="contact.php">

<div class="form-group mb-3">

<label for="name">Full Name</label>

<input type="text" class="form-control" placeholder="Enter your name" name="name" required>

</div>

<div class="form-group mb-3">

<label for="email">Email address</label>

<input type="email" class="form-control" placeholder="Enter your email" name="email" required>

</div>

<div class="form-group mb-3">

<label for="message">Message</label>

<textarea class="form-control" rows="5" placeholder="Enter your message" name="message" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

<div class="col-md-6">

<h5>Contact Information</h5>

<p><strong>Address:</strong> 431605, Nanded, Maharashtra,

India</p>

<p><strong>Phone:</strong> +91 935 945 8009</p>

<p><strong>Email:</strong> gajanana19022000@gmail.com</p> <div class="map-responsive">

<iframe

src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3153.835434

509374!2d144.9630579153169!3d-

37.81410797975195!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x6 ad642af0f11fd81%3A0xf577d8b1e4b0b1e!2sFederation%20Square!5e0!3m2!1sen !2sau!4v1611810190848!5m2!1sen!2sau" frameborder="0" style="border:0;" allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>

</div>

</div>

</div>

</div>

</section>

<!-- //////////////////////////////////// -->

<footer class="bg-dark text-white py-4">

<div class="container">

<div class="row">

<div class="col-md-3 container py-4">

<h3 style="color: aliceblue;">Portrait Photography</h3>

<p>The customer is at the heart of our<br>unique business model, which includes<br>design.</p>

<img src="https://i.postimg.cc/Nj9dgJ98/cards.png" alt="cards" class="img-fluid">

</div>

<div class="col-md-3 container py-4">

<h4>SERVIES</h4>

<ul class="list-unstyled">

<li><a href="#sellers" class="text-white">Wedding Shooting</a></li>

<li><a href="#sellers" class="text-white">Video edeating</a></li>

<li><a href="#sellers" class="text-white">Dron</a></li>

<li><a href="#sellers" class="text-white">Live Shotting</a></li>

</ul>

</div>

<div class="col-md-3 container py-4">

<h4 >FOR Custmer</h4>

<ul class="list-unstyled">

<div class="nav-link">

<li><a href="#contact" class="text-white">Contact Us</a></li>

<li><a href="#" class="text-white">Payment Method</a></li>

<li><a href="#" class="text-white">Delivery</a></li>

<li><a href="#" class="text-white">Return and Exchange</a></li> </ul>

</div>

<div class="col-md-3 container py-4">

<h5>Follow Us</h5>

<ul class="list-unstyled d-flex">

<li class="me-3">

<a href="https://www.facebook.com/profile.php?id=100083728363365" class="text-white d-flex justify-content-center align-items-center rounded-circle bg-primary" style="width: 40px; height: 40px;">

<i class="fab fa-facebook"></i>

</a>

</li>

<li class="me-3">

<a href="https://www.instagram.com/gajanandeshmukh5823/" class="text-white d-flex justify-content-center align-items-center rounded-circle bg-danger" style="width: 40px; height: 40px;">

<i class="fab fa-instagram"></i>

</a>

</li>

<li class="me-3">

<a href="https://web.whatsapp.com/" class="text-white d-flex justifycontent-center align-items-center rounded-circle bg-success" style="width: 40px; height: 40px;">

<i class="fab fa-whatsapp"></i>

</a>

</li>

<li> <a

href="https://www.linkedin.com/search/results/all/?fetchDeterministicClustersOnl y=true&heroEntityKey=urn%3Ali%3Afsd\_profile%3AACoAAD322BsBg4j2Wp

DbgZRXjKM5dip-dXM-

6Ho&keywords=gajanan%20deshmukh&origin=RICH\_QUERY\_SUGGESTION

&position=0&searchId=4602c0cd-dba8-4493-8566-

5e2fddf421be&sid=%3B(\*&spellCorrectionEnabled=false" class="text-white dflex justify-content-center align-items-center rounded-circle bg-info" style="width:

40px; height: 40px;">

<i class="fab fa-linkedin"></i>

</a>

</li>

</ul>

<

h5>For refund, please write to<br>

refunds@starair.in</h5

>

<

/div

>

<

div

>

<

hr class="bg

-

white">

<

/div

>

<

div class="text

-

center mt

-

3

"

>

<

p>© 2024 Design By Gajanan Deshmukh</p

>

<

/div

>

<

/div

>

<

/footer

>

<!

--

jQuery and Bootstrap Bundle with Popper

--

>

<

script src="https://code.jquery.com/jquery

-

3.5.1

.slim.min.js"></script

>

<

script

src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js">

<

/script

>

<

/body

>

<

/html

>

Script

**cript.js**

document.addEventListener('DOMContentLoaded', function() {

const cards = document.querySelectorAll('.card');

const observer = new IntersectionObserver(entries => {

entries.forEach(entry => {

if (entry.isIntersecting) {

entry.target.classList.add('show');

entry.target.classList.remove('hide');

}

else

{

entry.target.classList.remove('show');

entry.target.classList.add('hide');

}

})

;

})

;

cards.forEach(card => {

observer.observe(card);

})

;

})

;

//

/////////////////////////////////////////////

document.addEventListener('DOMContentLoaded', function () {

function handleCard(cardNumber, pricePerDay) {

let quantityElement = document.getElementById(`quantity${cardNumber}`);

let totalElement = document.getElementById(`total${cardNumber}`);

let addDayButton = document.getElementById(`addDay${cardNumber}`);

let removeDayButton =

document.getElementById(`removeDay${cardNumber}`);

let orderNowButton =

document.getElementById(`orderNow${cardNumber}`);

let quantity = 0;

addDayButton.addEventListener('click', function () {

quantity++;

updateTotal();

})

;

removeDayButton.addEventListener('click', function () {

if (quantity > 0) {

quantity

--

;

updateTotal();

}

})

;

orderNowButton.addEventListener('click', function () {

document.getElementById('orderDays').textContent = quantity;

document.getElementById('orderTotal').textContent = `$${quantity \*

pricePerDay}`;

})

;

function updateTotal() {

quantityElement.textContent = quantity;

totalElement.textContent = quantity \* pricePerDay;

}

}

handleCard(1, 10);

handleCard(2, 10);

handleCard(3, 10);

handleCard(4, 10);

handleCard(5, 120);

handleCard(6, 10);

handleCard(7, 10); handleCard(8, 10);

});

// /////////////////////////////////////////////////////////

const chatButton = document.getElementById('chatButton'); const chatContainer = document.getElementById('chatContainer'); const chatBox = document.getElementById('chatBox'); const userInput = document.getElementById('userInput'); const sendButton = document.getElementById('sendButton');

// Toggle chat visibility and refresh if already visible chatButton.addEventListener('click', () => {

if (chatContainer.style.display === 'none' || chatContainer.style.display === '')

{

chatContainer.style.display = 'block';

} else {

chatContainer.style.display = 'none';

chatBox.innerHTML = ''; // Clear the chat box

}

});

// Function to append a new message to the chat box function appendMessage(sender, message) { const messageDiv = document.createElement('div');

messageDiv.classList.add('message', sender === 'user' ? 'text-right' : 'textleft');

messageDiv.innerHTML = `<div class="alert ${sender === 'user' ? 'alertprimary' : 'alert-secondary'}" role="alert">${message}</div>`; chatBox.appendChild(messageDiv);

chatBox.scrollTop = chatBox.scrollHeight;

}

// Predefined responses (simple AI logic) function getAIResponse(userMessage) { const responses = {

"hi": "Hello! How can I help you?",

"hellow":"Hellow! How can I assist you today? ",

"how are you": "I'm just a bunch of code, but I'm here to help!",

"what is your name": "I'm your friendly AI assistant!",

"bye": "Goodbye! Have a great day!",

"location": "Here is the LOcation: <a href='https://www.google.co.in/maps' target='\_blank'>Visit Location</a>",

"database":"Hear Is Data base:<a href='database.php' >click hear</a>",

"email": "Here is the link to open your email: <a href='mailto:youremail@example.com'>Send Email</a>",

"phone": "Here is the link to call: <a href='tel:9359458009'>9359458009</a>",

"sms": "Here is the link to send a message: <a href='sms:9359458009'>Send Message to 9359458009</a>"

};

const lowerCaseMessage = userMessage.toLowerCase();

if (lowerCaseMessage === "date") { const today = new Date();

const date = today.getFullYear() + '-' + (today.getMonth() + 1) + '-' + today.getDate();

return "Today's date is " + date;

}

if (lowerCaseMessage === "time") {

const currentTime = new Date().toLocaleTimeString(); return `The current time is ${currentTime}`;

}

return responses[lowerCaseMessage] || "Sorry, I didn't understand that. Can you rephrase?";

}

// Event listener for the Send button sendButton.addEventListener('click', () => { const userMessage = userInput.value.trim(); if (userMessage) {

appendMessage('user', userMessage); const aiResponse = getAIResponse(userMessage); setTimeout(() => appendMessage('ai', aiResponse), 500); userInput.value = '';

}

});

// Allow sending the message with the Enter key userInput.addEventListener('keydown', (event) => {

if (event.key === 'Enter') {

sendButton.click();

}

})

;

document.getElementById('signInLink').addEventListener('mouseover',

function() {

var loginModal = new

bootstrap.Modal(document.getElementById('loginModal'));

loginModal.show();

})

;

document.getElementById('loginForm').addEventListener('submit',

function(event) {

event.preventDefault(); // Prevent the default form submission

// Add the move

-

up class to trigger the animation

document.querySelector('.modal

-

content').classList.add('move

-

up');

// Submit the form data via AJAX

fetch('login.php', {

method: 'POST',

body: new FormData(document.getElementById('loginForm'))

})

.then(response => response.text

())

.then(data => {

// Handle the response data

console.log(data);

// Hide the modal after the animation

setTimeout(function() {

var loginModal = new

bootstrap.Modal(document.getElementById('loginModal'));

loginModal.hide();

// Reset the form

document.getElementById('loginForm').reset();

// Refresh the page

location.reload();

}

, 500); // Match the duration of the animation

})

.catch(error =>

{

console.error('Error:', error);

})

;

})

;

document.getElementById('signUpLink').addEventListener('mouseover',

function() {

var registerModal = new

bootstrap.Modal(document.getElementById('registerModal'));

registerModal.show();

})

;

document.getElementById('registerForm').addEventListener('submit',

function(event) {

event.preventDefault(); // Prevent the default form submission

// Add the move

-

up class to trigger the animation

document.querySelector('#registerModal

.modal

-

content').classList.add('move

-

up');

// Submit the form data via AJAX

fetch('Registration.php', {

method: 'POST',

body: new FormData(document.getElementById('registerForm'))

})

.then(response => response.text

())

.then(data => {

// Handle the response data

console.log(data);

// Hide the modal after the animation

setTimeout(function() {

var registerModal = new

bootstrap.Modal(document.getElementById('registerModal'));

registerModal.hide();

// Reset the form

document.getElementById('registerForm').reset();

// Refresh the page

location.reload();

}

, 500); // Match the duration of the animation

})

.catch(error =>

{

console.error('Error:', error);

})

;

})

;

document.addEventListener('DOMContentLoaded', function () {

document.getElementById('contactForm').addEventListener('submit', function

(

event)

{

event.preventDefault(); // Prevent the default form submission

const formData = new FormData(document.getElementById('contactForm'));

// Log form data for debugging

for (let [key, value] of formData.entries()) {

console.log(`${key}: ${value}`);

}

fetch('contact.php', {

method: 'POST',

body: formData

})

.then(response => response.text

())

.then(data => {

// Handle the response data

console.log(data);

// Reset the form

document.getElementById('contactForm').reset();

// Optionally, show a success message or refresh the page

alert('Your message has been sent successfully!');

})

.catch(error =>

{

console.error('Error:', error);

})

;

})

;

})

;

document.addEventListener('DOMContentLoaded', function () {

// Submit the form data via AJAX

document.getElementById('orderForm').addEventListener('submit', function

(

event)

{

event.preventDefault(); // Prevent the default form submission

fetch('order.php', {

method: 'POST',

body: new FormData(document.getElementById('orderForm'))

})

.then(response => response.text

())

.then(data => {

// Handle the response data

console.log(data);

// Hide the modal after the animation

setTimeout(function() {

var orderModal = new

bootstrap.Modal(document.getElementById('orderModal'));

orderModal.hide();

// Reset the form

document.getElementById('orderForm').reset();

// Refresh the page

location.reload();

}

, 500); // Match the duration of the animation

})

.catch(error =>

{

console.error('Error:', error);

})

;

})

;

// Remove the animation class when the modal is shown again

document.getElementById('orderModal').addEventListener('shown.bs.modal',

function () {

document.querySelector('#orderModal .modal

-

content').classList.remove('move

-

up');

})

;

})

;

**4**

**.**

**DATABASE QUERIES**

CREATE TABLE users ( id INT NOT

NULL

AUTO\_INCREMENT,

email

VARCHAR(100) NOT NULL, password

VARCHAR(100) NOT NULL, full\_name

VARCHAR(100) NOT NULL, phone

VARCHAR(20) NOT NULL, gender

ENUM('male', 'female') NOT NULL,

college\_name VARCHAR(100) NOT

NULL, PRIMARY KEY(id)

)

;

CREATE TABLE cities ( id

INT

NOT

NULL

name

AUTO\_INCREMENT,

VARCHAR(100) NOT NULL,

PRIMARY KEY(id)

)

;

CREATE TABLE properties (

id

INT

NOT

NULL

AUTO\_INCREMENT, city\_id

INT NOT NULL, name

VARCHAR(100) NOT NULL,

address VARCHAR(255) NOT

NULL,

description

LONGTEXT NOT NULL,

gender ENUM('male', 'female', 'unisex') NOT

NULL, rent INT NOT NULL, rating\_clean

FLOAT(2,1)

NOT

NULL,

rating\_food

FLOAT(2,1) NOT NULL, rating\_safety

FLOAT(2,1) NOT NULL,

PRIMARY KEY(id),

FOREIGN KEY(city\_id) REFERENCES cities(id)

)

;

CREATE TABLE amenities ( id

INT

NOT

NULL

name

AUTO\_INCREMENT,

VARCHAR(100) NOT NULL,

type VARCHAR(100) NOT

NULL, icon VARCHAR(30)

NOT

NULL,

PRIMARY

KEY(id)

)

;

CREATE

TABLE

properties\_amenities

(

id

INT

NOT

NULL

AUTO\_INCREMENT,

property\_id INT NOT NULL,

amenity\_id INT NOT NULL,

PRIMARY KEY(id),

FOREIGN KEY(property\_id) REFERENCES properties(id),

FOREIGN KEY(amenity\_id) REFERENCES amenities(id)

)

;

CREATE TABLE testimonials ( id INT NOT NULL AUTO\_INCREMENT, property\_id

INT NOT NULL,

(9

, 5, 'Rajkumar Hirani', 'You just have to arrive at the place, it

\

's

fully furnished and stocked with all basic amenities and services and even

your friends are welcome.'),

(10

, 5, 'Sanjay Leela Bhansali', 'You just have to arrive at the place,

it

\

's fully furnished and stocked with all basic amenities and services

and even your friends are welcome.');

INSERT INTO

`users`

(

`id`, `email`, `password`, `full\_name`, `phone`, `gender`,

`college\_name`)

VALUES

'

(1

,

anuj.kalbalia@gmail.com

',

'b1b3773a05c0ed0176787a4f1574ff0075f7521e',

'Anuj

Kalbalia',

'9535100112', 'male', 'NITD'),

(2

, '

shadab@gmail.com

', 'b1b3773a05c0ed0176787a4f1574ff0075f7521e',

'Shadab Alam', '9876543210', 'male', 'NITJ'),

(3

, '

aditya@gmail.com

', 'b1b3773a05c0ed0176787a4f1574ff0075f7521e',

'Aditya

Sood', '9876543210', 'male', 'Chandigarh University'),

(4

, '

radhika@gmail.com

', 'b1b3773a05c0ed0176787a4f1574ff0075f7521e',

'Radhika Bhatia', '9876543210', 'female', 'Delhi University');

INSERT INTO

`interested\_users\_properties`

(

`id`, `user\_id`, `property\_id`)

VALUES

(1

, 1, 1),

(2

, 1, 2),

(3

, 1, 5),

(4

, 2, 1),

(5

, 2, 5),

(6

, 3, 1),

(7

, 3, 2),

(8

, 3, 5),

(9

, 4, 2),

(10

, 4, 3),

(11

,

4

, 4);

INSERT INTO

`amenities`

(

`id`, `name`, `type`, `icon`)

VALUES

(1

, 'Wifi', 'Common Area', 'wifi'),

(2

, 'Power Backup', 'Building', 'powerbackup'),

(3

, 'Fire Extinguisher', 'Building', 'fireext'),

(4

, 'TV', 'Common Area', 'tv'),

(5

, 'Bed with Mattress', 'Bedroom', 'bed'),

(6

, 'Parking', 'Building', 'parking'),

(7

, 'Water Purifier', 'Common Area', 'rowater'),

(8

, 'Dining', 'Common Area', 'dining'),

(9

, 'Air Conditioner', 'Bedroom', 'ac'),

(10

, 'Washing Machine', 'Common Area', 'washingmachine'),

(11

, 'Lift', 'Building', 'lift'),

(12

, 'CCTV', 'Building', 'cctv'),

(13

, 'Geyser', 'Washroom', 'geyser');

INSERT INTO

`properties\_amenities`

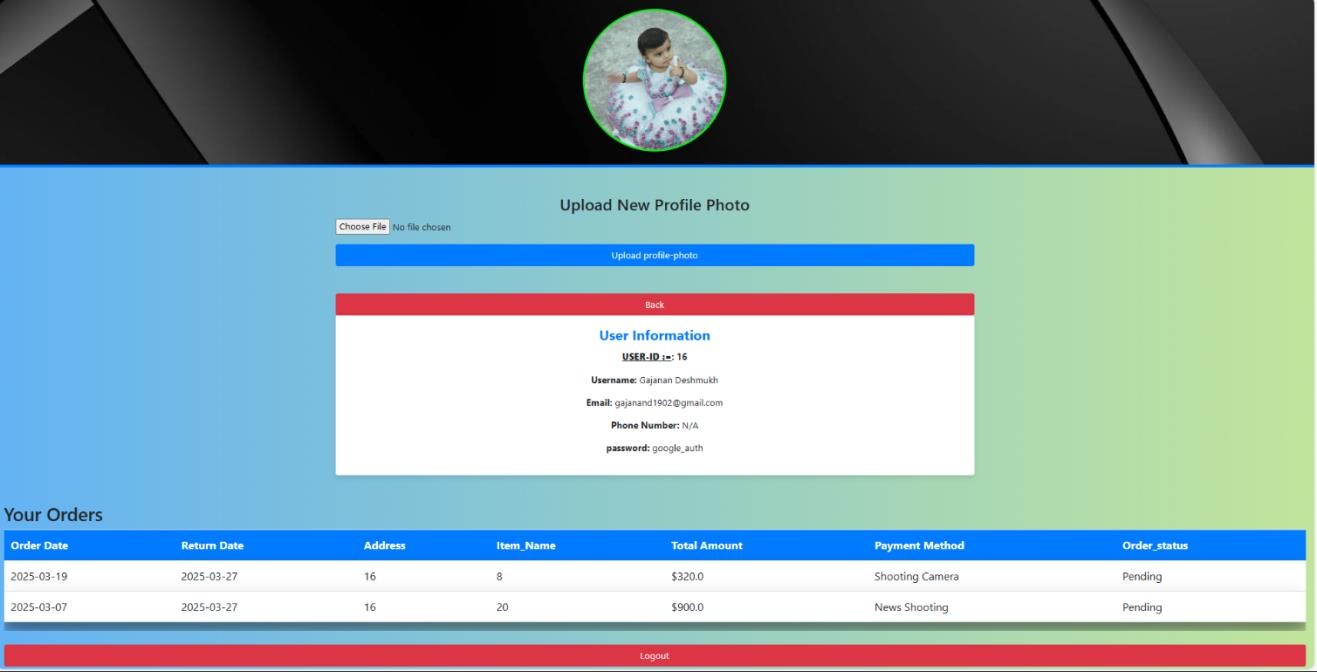
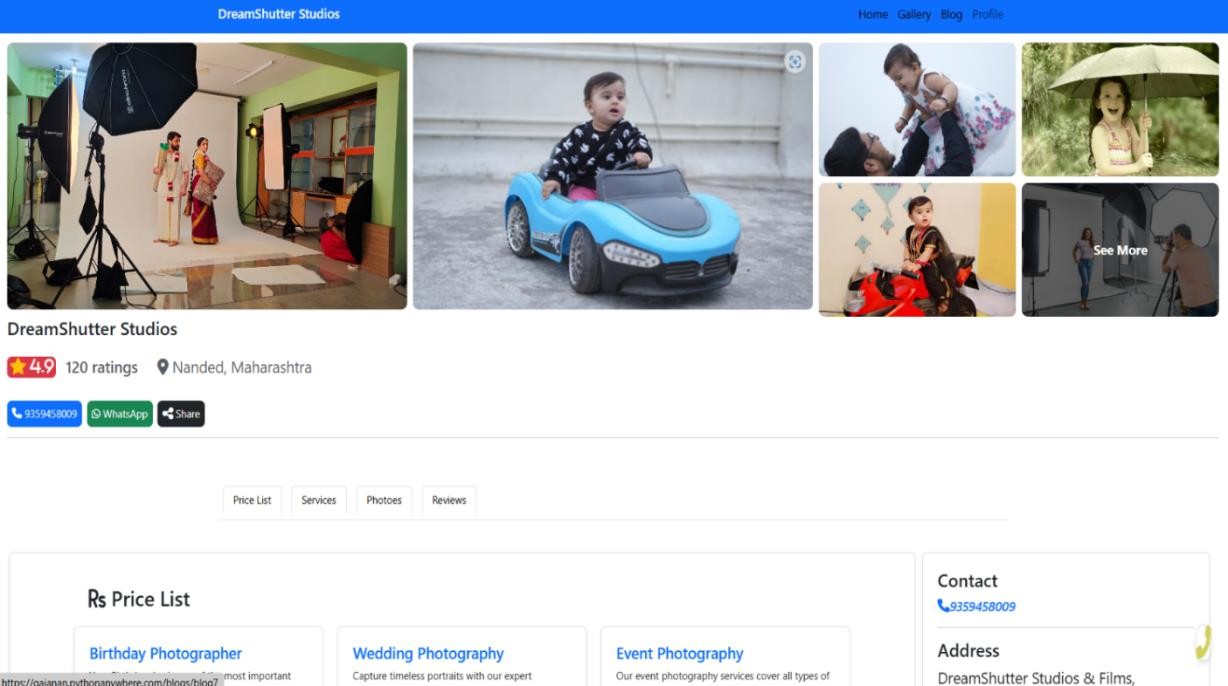
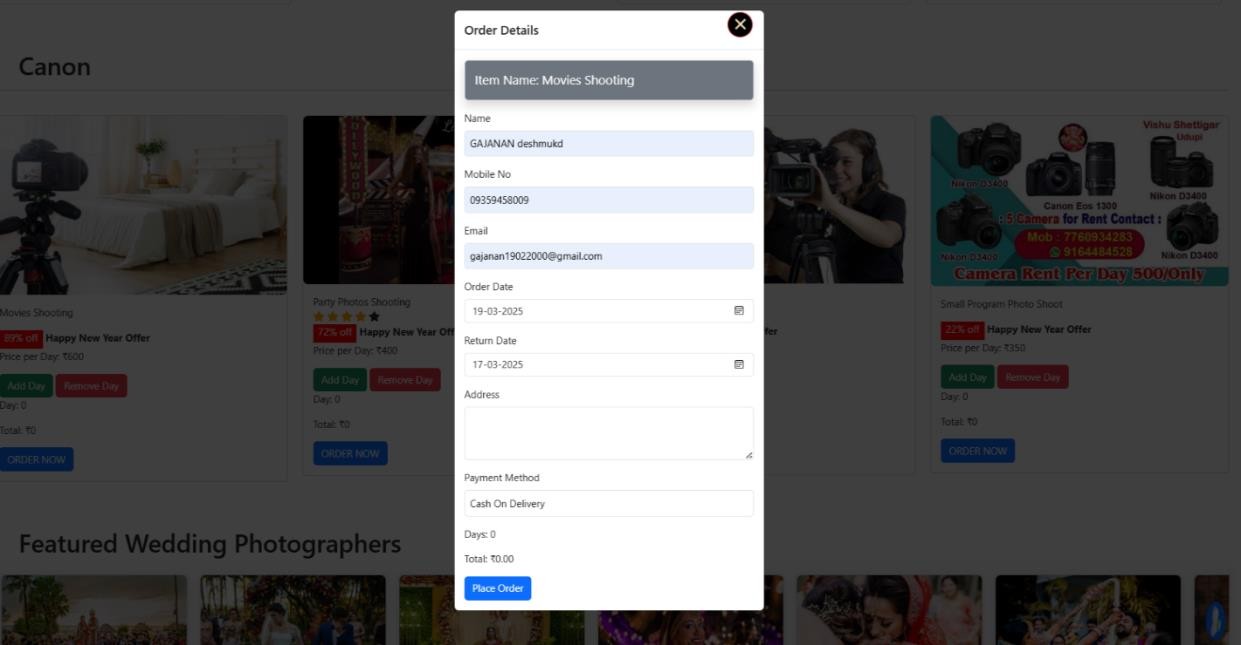
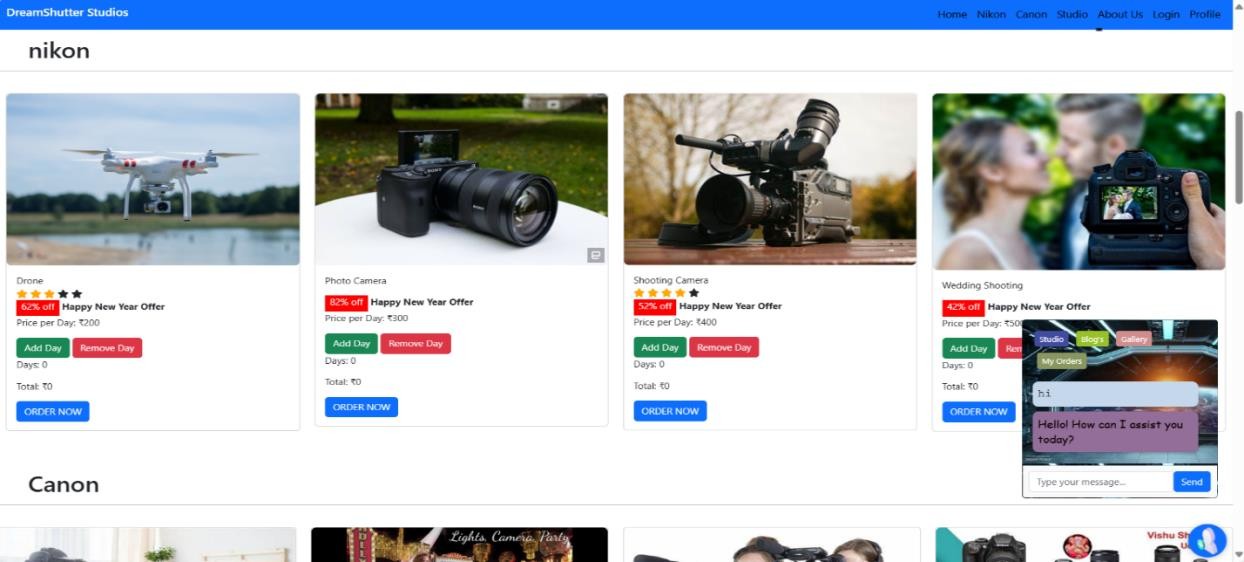
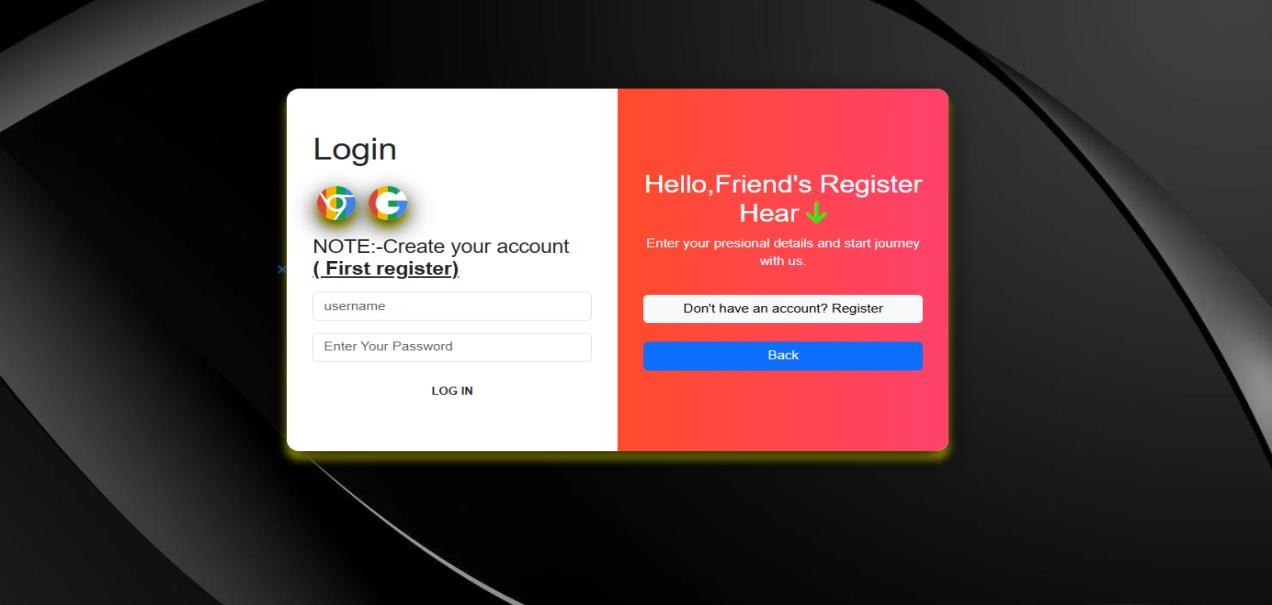
(

`id`, `property\_id

`, `amenity\_id`)

VALUES

**OUTPUT**



**Conclusion**

The

**Camera Project**

successfully provides a user

-

friendly and efficient platform for

managing camera rentals and purchases. By integrating modern web technologies, the

system ensures a seamless experience for both users and administrators. The project

features

**secure authentication, a well**

**-**

**structured product catalog, an intuitive**

**booking system, and a reliable payment gateway.**

The use of

**Flask, SQLite, and REST APIs**

ensures a robust backend, while

**HTML,**

**CSS, JavaScript, and Bootstrap**

enhance the frontend experience. The structured

database design ensures data security and efficient retrieval of information.

This project demonstrates the effective implementation of

**web development principles,**

**database management, and**

**user**

**-**

**centric design.**

It can be further enhanced by

integrating advanced features like AI

-

based recommendations, a review system, and

cloud

-

based storage for improved scalability.

Overall, the Camera Project serves as a

**scalable, secure, and well**

**-**

**organized solution**

for users looking for camera rental and purchasing services.

**References**

1.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020).

*Database System Concepts*

(7

th ed.).

McGraw

-

Hill Education.

2.

Sommerville, I. (2019).

*Software Engineering*

(10

th ed.). Pearson.

3.

W3Schools.

(

n.d.).

HTML,

CSS,

and

JavaScript

Tutorials.

Retrieved

from

[https://www.w3schools.co](https://www.w3schools.com/)

[m](https://www.w3schools.com/)

4.

Mozilla Developer Network (MDN). (n.d.). Web Development Documentation. Retrieved

from

[https://developer.mozilla.or](https://developer.mozilla.org/)

[g](https://developer.mozilla.org/)

5.

React Official Documentation. (n.d.). Learn React. Retrieved from

[h](https://react.dev/)

[ttps://react.de](https://react.dev/)

[v](https://react.dev/)

6.

PHP

Official

Documentation.

(

n.d.).

PHP

Manual.

Retrieved

from

[https://www.php.net/docs.ph](https://www.php.net/docs.php)

[p](https://www.php.net/docs.php)

7.

MySQL Official Documentation. (n.d.). Database Management Guide. Retrieved from

[https://dev.mysql.com/doc](https://dev.mysql.com/doc/)

[/](https://dev.mysql.com/doc/)

8.

Bootstrap.

(

n.d.).

Responsive

Web

Design

Framework.

Retrieved

from

[https://getbootstrap.co](https://getbootstrap.com/)

[m](https://getbootstrap.com/)

9.

GitHub. (n.d.). Version Control and Collaboration. Retrieved from

[h](https://github.com/)

[ttps://github.co](https://github.com/)

[m](https://github.com/)

10.

Visual

Studio

Code.

(

n.d.).

Code

Editor

for

Developers.

Retrieved

from

[https://code.visualstudio.co](https://code.visualstudio.com/)

[m](https://code.visualstudio.com/)

11.

GeeksforGeeks. (n.d.

)

. Programming and Web Development Tutorials. Retrieved from

[https://www.geeksforgeeks.or](https://www.geeksforgeeks.org/)

[g](https://www.geeksforgeeks.org/)

12.

Stack

Overflow.

(

n.d.).

Developer

Community

Forum.

Retrieved

from

[https://stackoverflow.co](https://stackoverflow.com/)

[m](https://stackoverflow.com/)

13.

XAMPP Documentation. (n.d.). PHP and MySQL Server Management. Retrieved from

[https://www.apachefriends.or](https://www.apachefriends.org/)

[g](https://www.apachefriends.org/)