VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Fall Semester 2021-2022

Course Code : CSE3002 Program : B.Tech

Course Name : Internet and Web Programing

Faculty : Dr. JANANI T

Review 3 Report

PhotoSoc

(Social Media WebApp)

Team Members:

Rupin Patel – 19BCB0030

DECLARATION BY THE CANDIDATE

I hereby declare that the Project report entitled “PhotoSoc (Social media app)” submitted by me to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of B.Tech in Computer Science and Engineering with Specialisation in Bioinformatics is a record of bonafide course Internet and Web Programing undertaken by me under the supervision of Dr Janani T. I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Rupin Patel
19BCB0030
5-12-2021

Table of Content:

1.Abstract

Social Networking becoming more and more popular with time and youth. It is famous because the ease of access , popularity, marketing, flexing skills and grab other people attention. But somehow some people use social media for surfing other people work, its good but there is mean to learn that particular skill for that work, So ultimately the time is wasted and no productive knowledge is acquired. PhotoSoc is a simple Social media app where you can see exif image meta-data. App specially for flexing photography skills and where new user get help for gaining knowledge about photography and different techniques of Photography.

2.Introduction

PhotoSoc app allows users to post their photographs no matter what they are capturing but it should be captured with camera (phone, DSLR, Adobe Edit).The Image in the post get processed at backend and create a metadata containing all the details, settings, orientations, device Name, device specifications. The whole post with image, metadata, caption will be stored and get displayed on posts screen. The Image metadata will help other users to know what things and procedure they must follow to capture this particular photo. The user also can create Events to conduct workshops, competitions to spread awareness and knowledge. Overall photosoc help to gain knowledge in photography and also for spreading awareness.

3.Software and tool used

1. Python and Flask Frameworks for Backend and Image Processing

2. PHP framework for event management system

3. MySQL for Database

4. Firebase for user documents, authentication, and file storage

5. HTML, CSS, Bootstrap for frontend and Responsiveness

6. VS Code and Pycharm for Code Editing and project creation

7. Python Modules for Image Processing

8. XAMPP for PHP and MySQL

9. Google Cloud API

10. EXIF metdata generator

4. Workflow

Sequence Diagram for PhotoSoc:

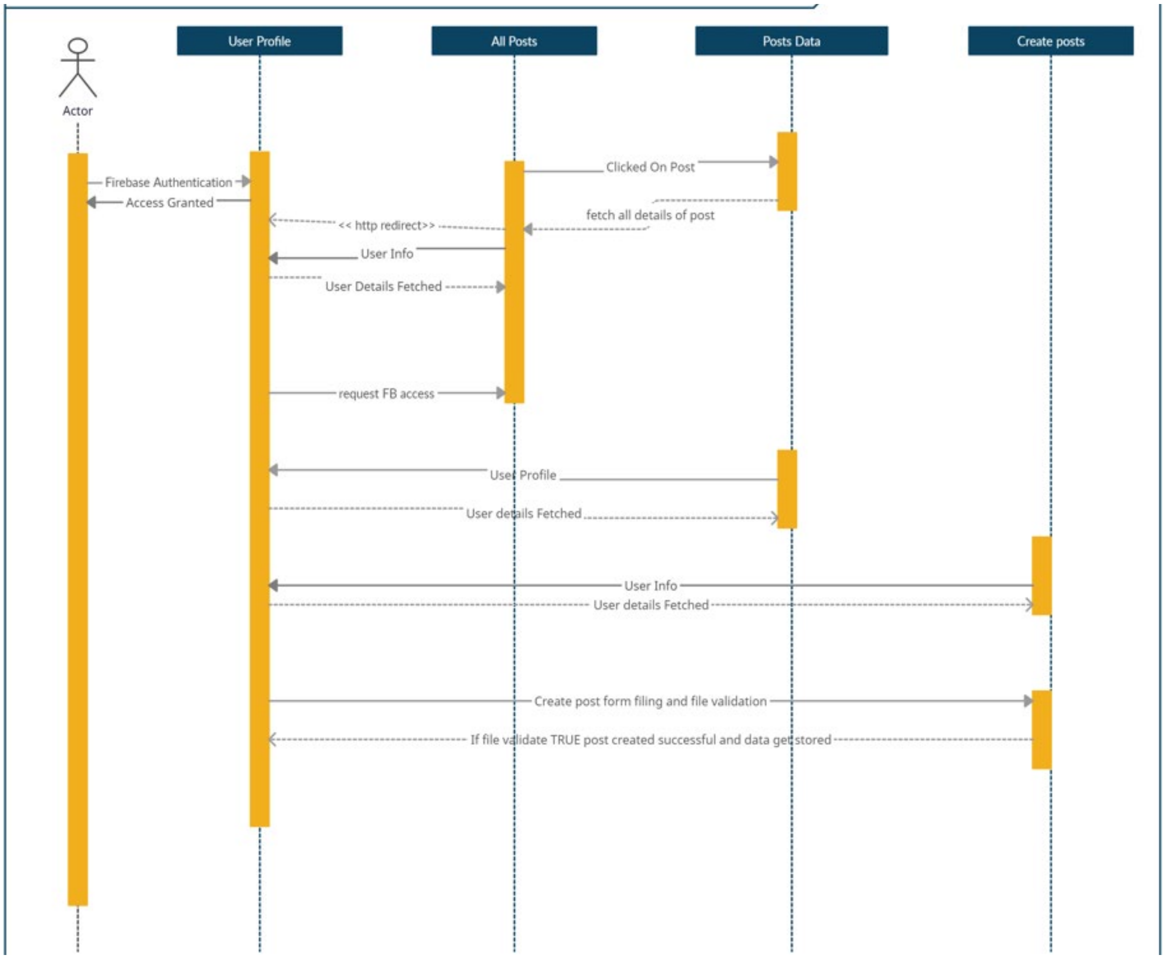


Figure 1

Sequence Diagram for Event Management:

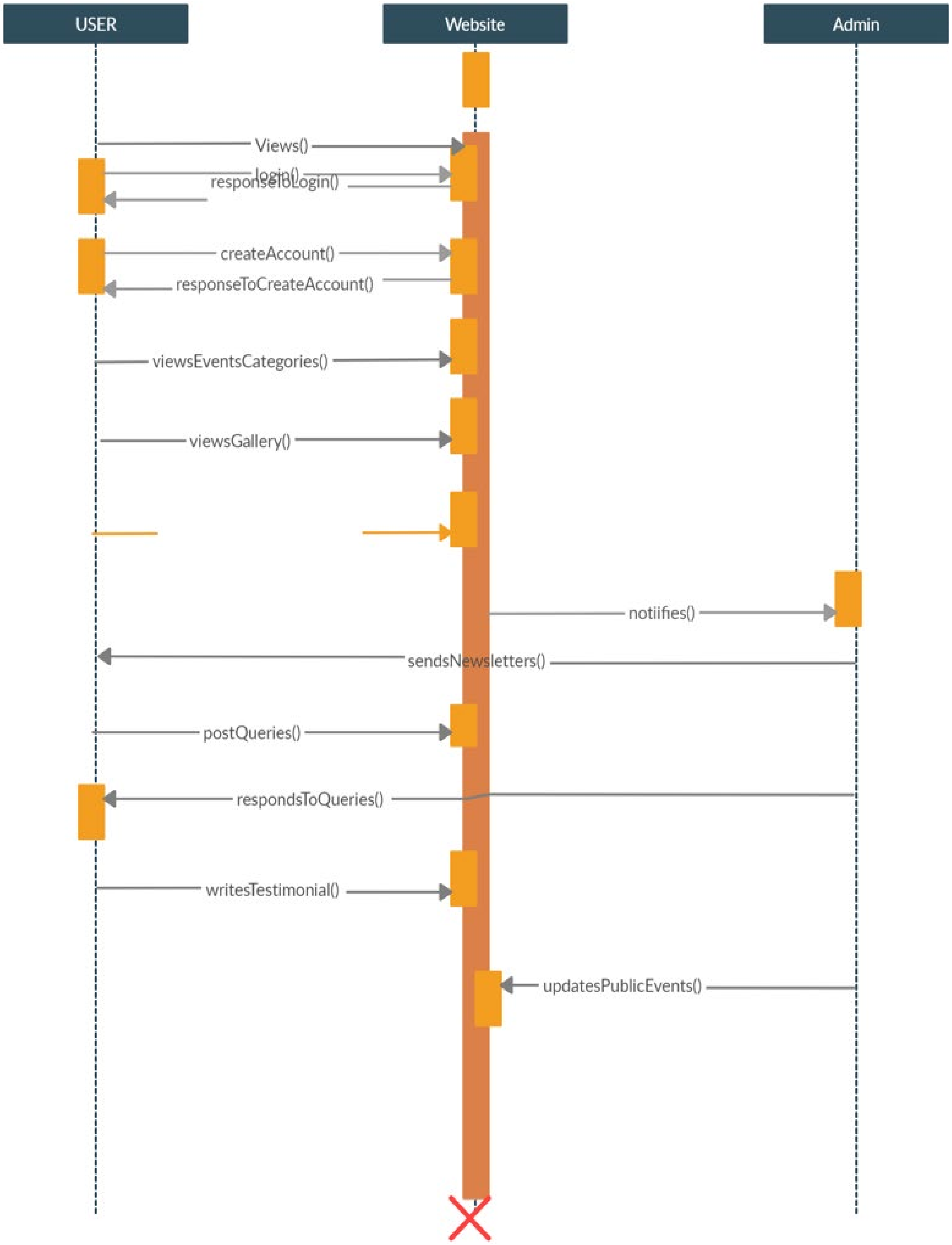


Figure 2

5. Modules

1. Register to be member module:

This module provide functionalities for those people who wants to open account. Applicants can post their views with personal and professional details. They can also update profile as frequent as required. The member can also browse through the post details. Members can also see metadata and user profile.

2.Profile Module:

The module provides functionalities related to members profile. Logged users can see their details and number posts and if they wish they can delete their posts.

3.Admin Module:

The module provide administrator related functionalities. Administrator manages entire application and maintain the profiles of all users and activities and also wireframes.

4.Create Post Module:

The module allow to fill form contains all options required to create post such as caption characteristic and image choose file button. And all data and file get stored in firebase.

5.Image Processing Module:

The module fetch the image metadata and get all information of image and display as post details, with the help of python modules such as OpenCV, EXIF and Pillow.

Event Management System Module
1.Admin Module

2.User Module

Admin Module

1. Dashboard: In this section, admin can see all detail in brief like listed categories, Sponsors, Total Events, Total Registered Users, Total Booking, Total New Booking, Total Confirmed Booking and Total Cancelled Booking.
2. Category: In this section, admin manage event category (add and update).
3. Manage Sponsors: In this section, admin can add sponsors and manage sponsors details (Add/Update/Delete).
4. Events: In this section, admin manage events (add and update).
5. Manage Users: In this section, admin can update details of registered users and also block them.
6. Manage Booking: In this section, admin can manage booking by cancel and confirm it.
7. News: In this section admin manage news (add and delete).
8. Website Setting: In this section, admin can update about us and another general website setting.

Admin can also update his profile, change password and recover password.

User Module

In this module there is two types of user guest user and registered user.

Guest User: In this guest user can see only general information like about us, event details, contact details and news about events.

Registered users can do the following activity

1.Books the events.

2.Update his/her own profile.

3.Change Password.

4.Users can also cancel booking which is not confirmed.

5. Registered user can also recover his/her own password.

6.Frontend

1. Login Page allow user to get into system with username and password.

2. SignUp page allow new user to create and login into system.

3. Profile page allow users to see his/her own details and posts with deleting and uploading profile photo privileges.

4. Create Post page allow users to upload image and entered required forms details for further process.

5. Posts Page allow users to see all posts of different users and also allow user to filter result with search posts options.

6. Post details page where user can see the image metadata and all other details such as camera

name, settings, and other required details.

9. Events page will redirect users directly to event management system.

10. Venue Page where user can see all available venues.

11. Manage Page where user have to type admins credentials and have permission to manage.

12. Users where admin can see all users an there details

13. And all pages such as Event audience, Audience Report and Venues where admin have more privileges.

14. Registration page where user can type required information in form get registered himself.

7. Backend

In backend flask framework, php admin, MySQL, Firebase and Python is used.
Flask Framework is used as a main structure of whole system where different pages are get routes with different function and returned required values and tuples on frontend.

Authentication is done by simple request and check credentials with firebase authentication facilities.
Signup page is also done with flask and firebase and the details of users are stored as document in firestore with the help of some NoSQL queries.

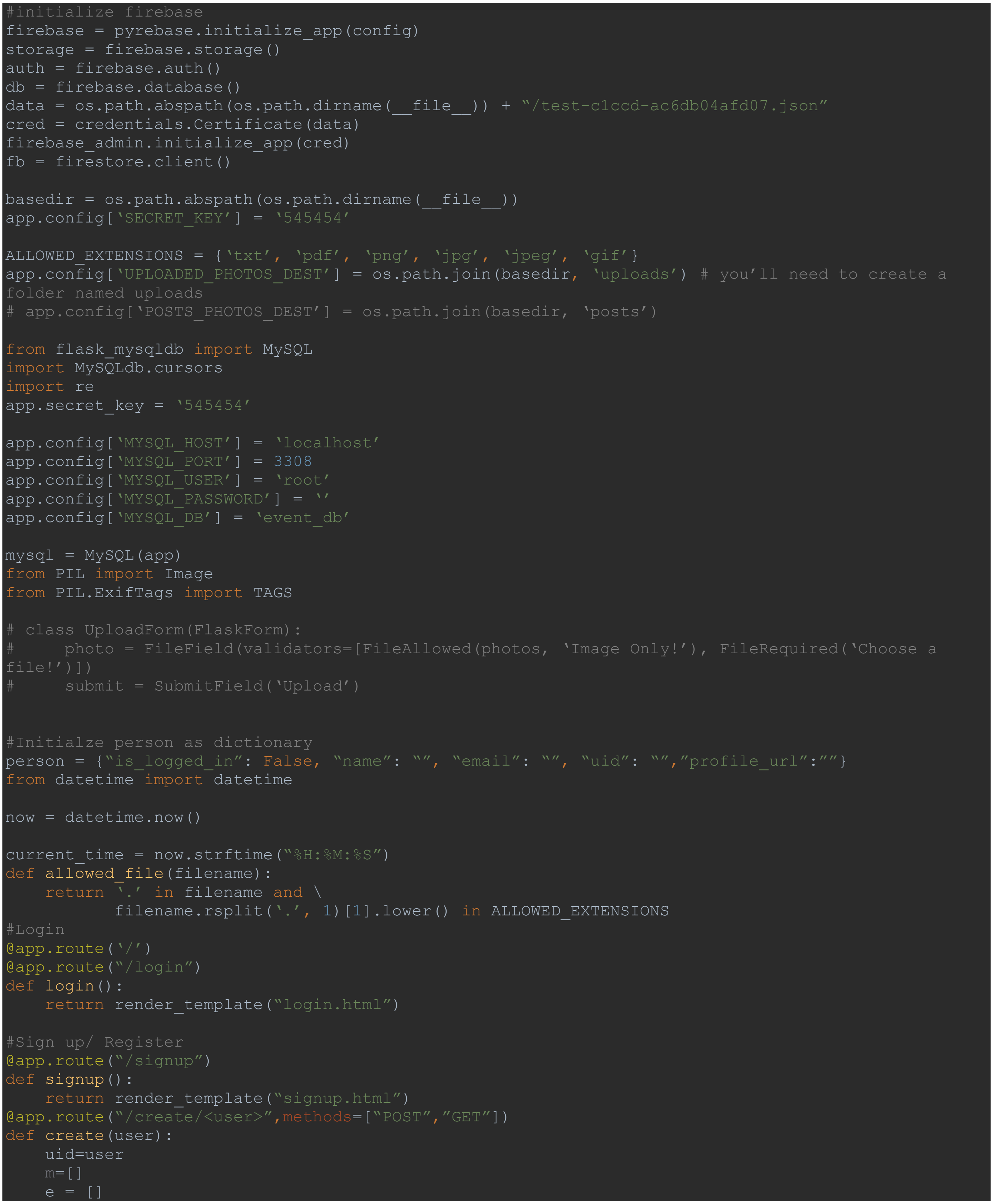
Overall fetching values and storing them is done by flask and firebase integration using python as a base language in backend. Authentication Firebase, Real-time database, firestore and fire storage bucket is used ,

Image Processing also done in backend part by using various python libraries.
Searching and filtering of posts is done by JavaScript and Jquery.

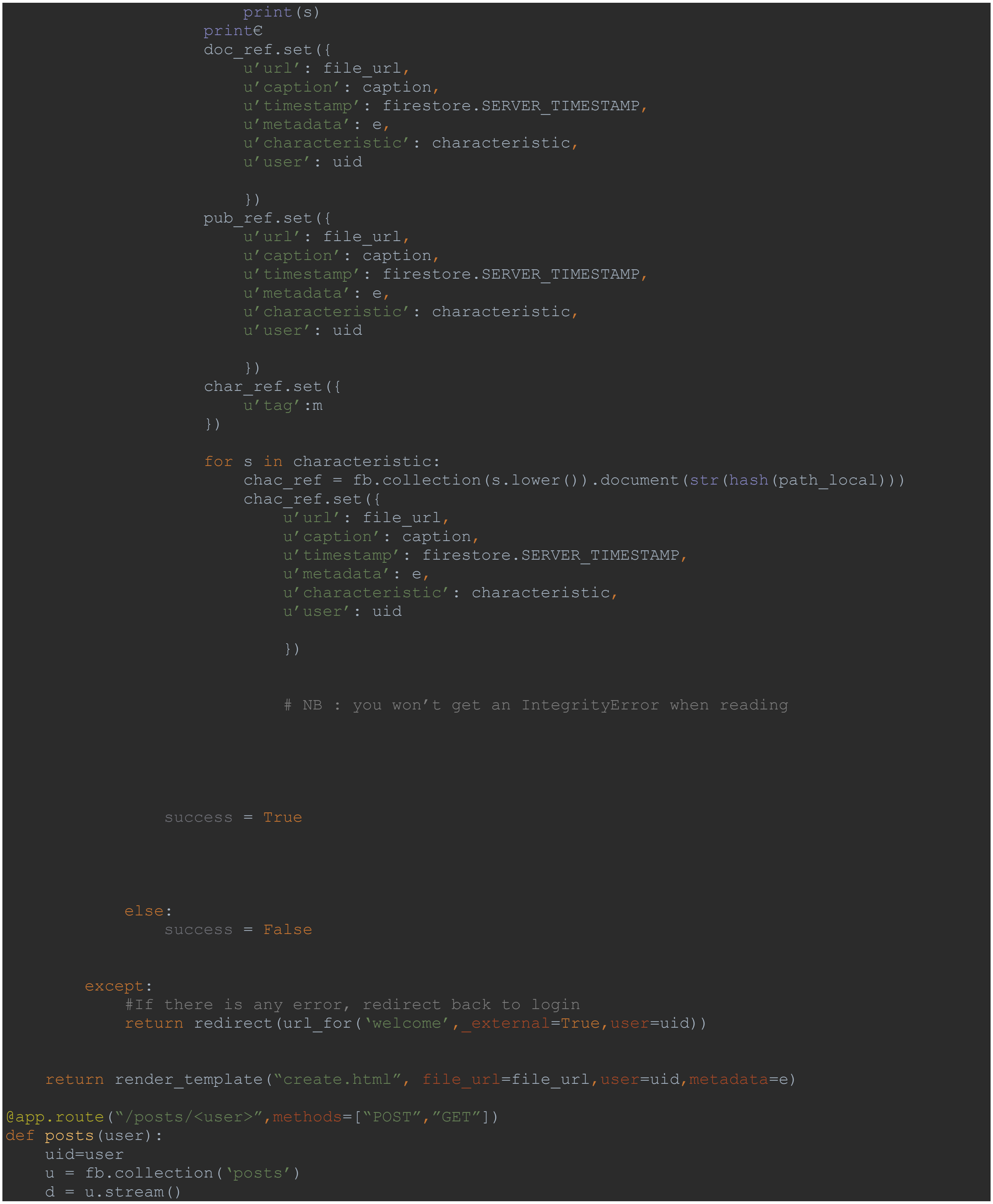
PHP and Mysql is used for event management part where dbConnect.php file is responsible for database connection and with various features and MySQL queries the whole system is been created with no redundancy and better admin privileges.

CODE:

```
import pyrebase
import os

import logging

import MySQLdb.cursors
import re

from flask import Flask, flash, redirect, render_template, request, session,make_response, abort,
url_for

from werkzeug.utils import secure_filename
basedir = os.path.abspath(os.path.dirname(__file__))
import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore
# from firebase_admin import auth
app = Flask(__name__)        #Initialze flask constructor
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

EXIFTOOL_PATH =”sudo chmod a+x”+os.path.abspath(os.path.dirname(__file__)) + “/exiftool.exe”

ELEMENTS_PER_PAGE = 50

#Add your own details
config = {


    “apiKey”: “AizaSyDtqr58_Sfj_KBkUxlVtCiDHQATf73yzNw”,
    “authDomain”: “test-c1ccd.firebaseapp.com”,
    “databaseURL”: “https://test-c1ccd-default-rtdb.firebaseio.com”,
    “projectId”: “test-c1ccd”,
    “storageBucket”: “test-c1ccd.appspot.com”,
    “messagingSenderId”: “638404195953”,
    “appId”: “1:638404195953:web:ff87b33a0637940e9b0dd0”
}
```



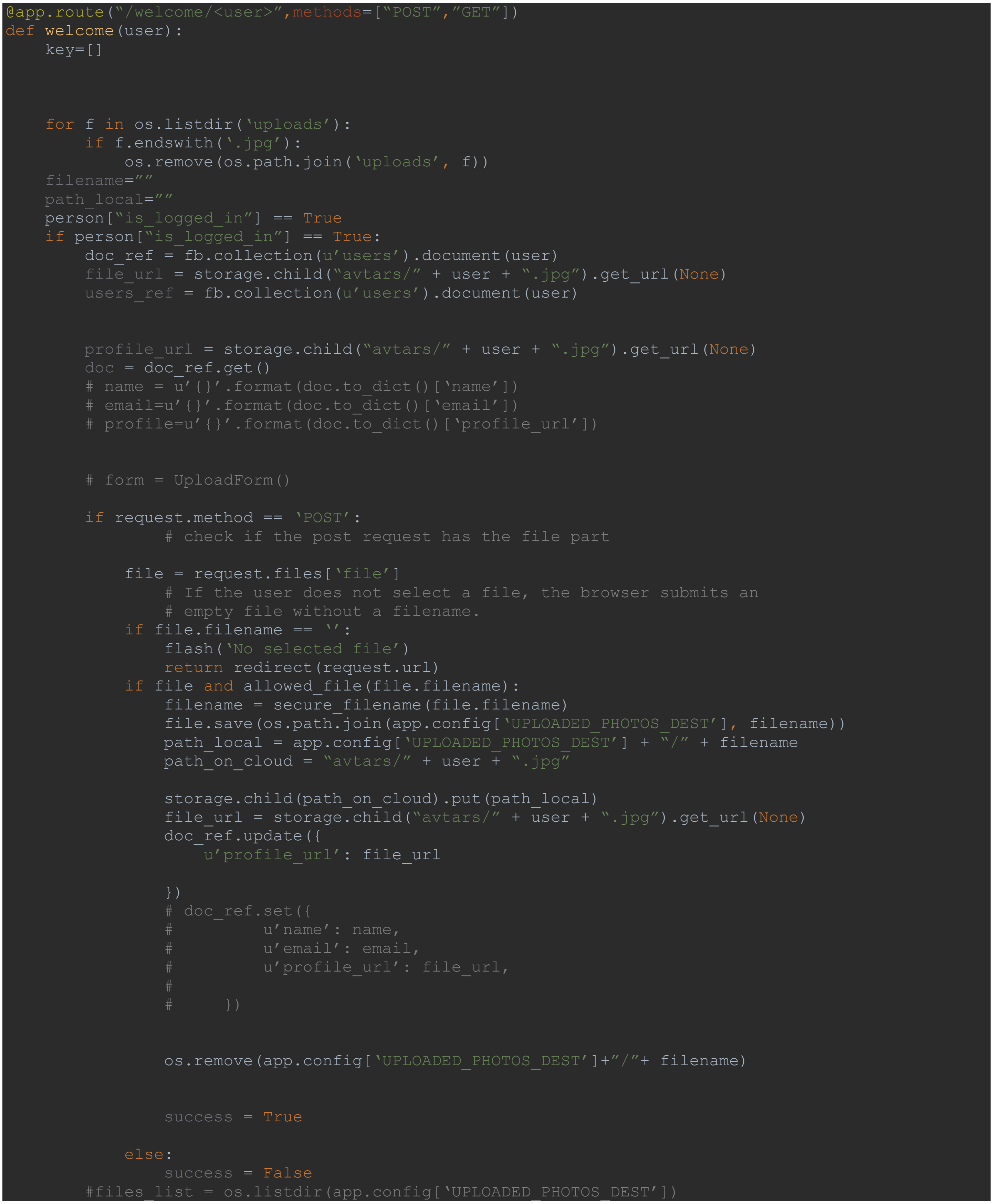
```
#initialize firebase
firebase = pyrebase.initialize_app(config)
storage = firebase.storage()
auth = firebase.auth()
db = firebase.database()
data = os.path.abspath(os.path.dirname(__file__)) + “/test-c1ccd-ac6db04afd07.json”
cred = credentials.Certificate(data)
firebase_admin.initialize_app(cred)
fb = firestore.client()

basedir = os.path.abspath(os.path.dirname(__file__))
app.config[‘SECRET_KEY’] = ‘545454’

ALLOWED_EXTENSIONS = {‘txt’, ‘pdf’, ‘png’, ‘jpg’, ‘jpeg’, ‘gif’}
app.config[‘UPLOADED_PHOTOS_DEST’] = os.path.join(basedir, ‘uploads’) # you’ll need to create a
folder named uploads
# app.config[‘POSTS_PHOTOS_DEST’] = os.path.join(basedir, ‘posts’)

from flask_mysqldb import MySQL
import MySQLdb.cursors
import re
app.secret_key = ‘545454’

app.config[‘MYSQL_HOST’] = ‘localhost’
app.config[‘MYSQL_PORT’] = 3308
app.config[‘MYSQL_USER’] = ‘root’
app.config[‘MYSQL_PASSWORD’] = ‘’
app.config[‘MYSQL_DB’] = ‘event_db’

mysql = MySQL(app)
from PIL import Image
from PIL.ExifTags import TAGS

# class UploadForm(FlaskForm):
#     photo = FileField(validators=[FileAllowed(photos, ‘Image Only!’), FileRequired(‘Choose a
file!’)])
#     submit = SubmitField(‘Upload’)


#Initialze person as dictionary
person = {“is_logged_in”: False, “name”: “”, “email”: “”, “uid”: “”,”profile_url”:””}
from datetime import datetime

now = datetime.now()

current_time = now.strftime(“%H:%M:%S”)
def allowed_file(filename):
    return ‘.’ in filename and \
           filename.rsplit(‘.’, 1)[1].lower() in ALLOWED_EXTENSIONS
#Login
@app.route(‘/’)
@app.route(“/login”)
def login():
    return render_template(“login.html”)

#Sign up/ Register
@app.route(“/signup”)
def signup():
    return render_template(“signup.html”)
@app.route(“/create/<user>”,methods=[“POST”,”GET”])
def create(user):
    uid=user
    m=[]
    e = []
    char_ref = fb.collection(‘search’).document(‘chara’)
    chac = char_ref.get()
    m=(chac.to_dict()[‘tag’])
    print(m)
    #timestamp = datetime.datetime(2017, 12, 1, 0, 0).timestamp()

    file_url=””

    if request.method == “POST”:  # Only listen to POST
        result = request.form  # Get the data submitted
        caption = result[“caption”]
        characteristic = result[“characteristic”]
        characteristic=characteristic.split(“,”)
        characteristic = [each_string.lower() for each_string in characteristic]
        m=m+characteristic
        l=set(m)
        m=l
        try:
            if request.method == ‘POST’:
                # check if the post request has the file part

                file = request.files[‘file’]
                # If the user does not select a file, the browser submits an
                # empty file without a filename.
                if file.filename == ‘’:
                    flash(‘No selected file’)
                    return redirect(request.url)
                if file and allowed_file(file.filename):
                    filename = secure_filename(file.filename)
                    file.save(os.path.join(app.config[‘UPLOADED_PHOTOS_DEST’], filename))
                    path_local = app.config[‘UPLOADED_PHOTOS_DEST’] + “/” + filename
                    path_on_cloud = “photos/” + uid + “/” + str(hash(path_local)) + “.jpg”

                    storage.child(path_on_cloud).put(path_local)
                    file_url = storage.child(“photos/” + uid + “/” + str(hash(path_local)) +
“.jpg”).get_url(
                        None)
                    doc_ref = fb.collection(uid).document(str(hash(path_local)))
                    pub_ref = fb.collection(‘posts’).document(str(hash(path_local)))

                    print(m)
                    image = Image.open(path_local)

                    q = []

                    exifdata = image.getexif()
                    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
                    cursor.execute(‘INSERT INTO posts VALUES (NULL, % s, % s, % s)’, (file_url,
caption, uid))
                    mysql.connection.commit()


                    # looping through all the tags present in exifdata
                    for tagid in exifdata:
                        # getting the tag name instead of tag id
                        tagname = TAGS.get(tagid, tagid)

                        # passing the tagid to get its respective value
                        value = exifdata.get(tagid)

                        # printing the final result
                        s = f”{tagname:24}:{value}”
                        if (tagname != “MakerNote” and tagname != “UserComment”):
                            e.append(s)
```



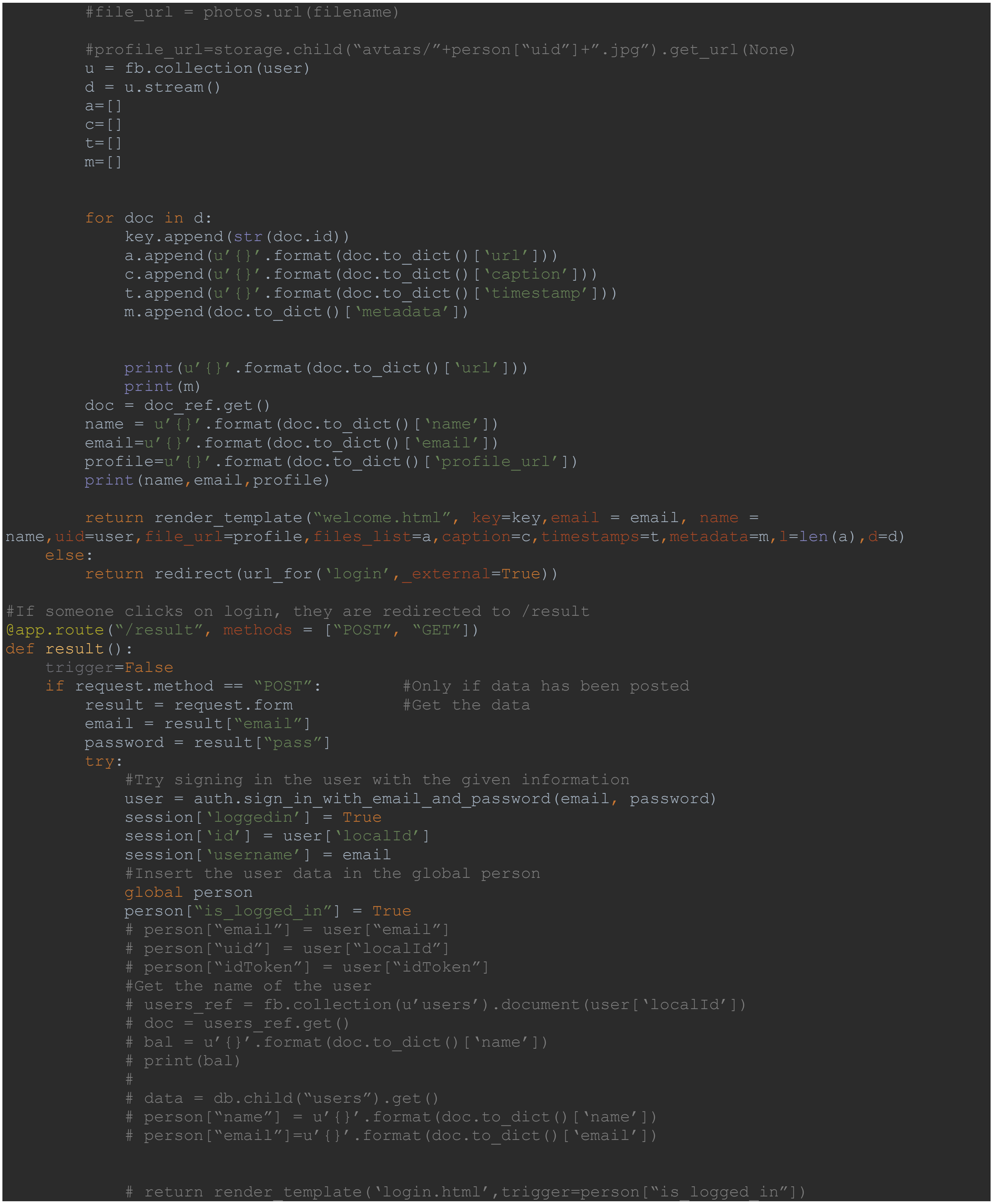
```
                        print(s)
                    print€
                    doc_ref.set({
                        u’url’: file_url,
                        u’caption’: caption,
                        u’timestamp’: firestore.SERVER_TIMESTAMP,
                        u’metadata’: e,
                        u’characteristic’: characteristic,
                        u’user’: uid

                        })
                    pub_ref.set({
                        u’url’: file_url,
                        u’caption’: caption,
                        u’timestamp’: firestore.SERVER_TIMESTAMP,
                        u’metadata’: e,
                        u’characteristic’: characteristic,
                        u’user’: uid

                        })
                    char_ref.set({
                        u’tag’:m
                    })

                    for s in characteristic:
                        chac_ref = fb.collection(s.lower()).document(str(hash(path_local)))
                        chac_ref.set({
                            u’url’: file_url,
                            u’caption’: caption,
                            u’timestamp’: firestore.SERVER_TIMESTAMP,
                            u’metadata’: e,
                            u’characteristic’: characteristic,
                            u’user’: uid

                            })


                            # NB : you won’t get an IntegrityError when reading





                success = True




            else:
                success = False


        except:
            #If there is any error, redirect back to login
            return redirect(url_for(‘welcome’,_external=True,user=uid))


    return render_template(“create.html”, file_url=file_url,user=uid,metadata=e)

@app.route(“/posts/<user>”,methods=[“POST”,”GET”])
def posts(user):
    uid=user
    u = fb.collection(‘posts’)
    d = u.stream()
```

```
a = []
c = []
t = []
m = []
key=[]
usernames=[]
user_emails=[]
user_profiles=[]

char_ref = fb.collection(‘search’).document(‘chara’)
chac = char_ref.get()
chac=(chac.to_dict()[‘tag’])
print(chac)


for doc in d:
    key.append(str(doc.id))
    print(doc.id)
    a.append(u’{}’.format(doc.to_dict()[‘url’]))
    c.append(u’{}’.format(doc.to_dict()[‘caption’]))
    t.append(u’{}’.format(doc.to_dict()[‘timestamp’]))

    m.append(doc.to_dict()[‘metadata’])
    id = u’{}’.format(doc.to_dict()[u’user’])
    users_ref = fb.collection(u’users’).document(id)
    user_doc = users_ref.get()
    name = u’{}’.format(user_doc.to_dict()[‘name’])
    usernames.append(name)
    email = u’{}’.format(user_doc.to_dict()[‘email’])
    user_emails.append(email)
    profile = u’{}’.format(user_doc.to_dict()[‘profile_url’])
    user_profiles.append(profile)
    print(u’{}’.format(doc.to_dict()[‘url’]))
    print(m)
if request.method == “POST”:
    a = []
    c = []
    t = []
    m = []
    usernames = []
    user_emails = []
    user_profiles = []
    # Only listen to POST

    result = request.form  # Get the data submitted
    tag = result[“tag”]
    print(tag)
    tagname= fb.collection(tag)
    k= tagname.stream()
    for doc in k:
        key.append(doc.id)

        a.append(u’{}’.format(doc.to_dict()[‘url’]))
        c.append(u’{}’.format(doc.to_dict()[‘caption’]))
        t.append(u’{}’.format(doc.to_dict()[‘timestamp’]))
        m.append(doc.to_dict()[‘metadata’])
        id=u’{}’.format(doc.to_dict()[u’user’])
        users_ref = fb.collection(u’users’).document(id)
        user_doc = users_ref.get()
        name = u’{}’.format(user_doc.to_dict()[‘name’])
        usernames.append(name)
        email = u’{}’.format(user_doc.to_dict()[‘email’])
        user_emails.append(email)
        profile = u’{}’.format(user_doc.to_dict()[‘profile_url’])
```

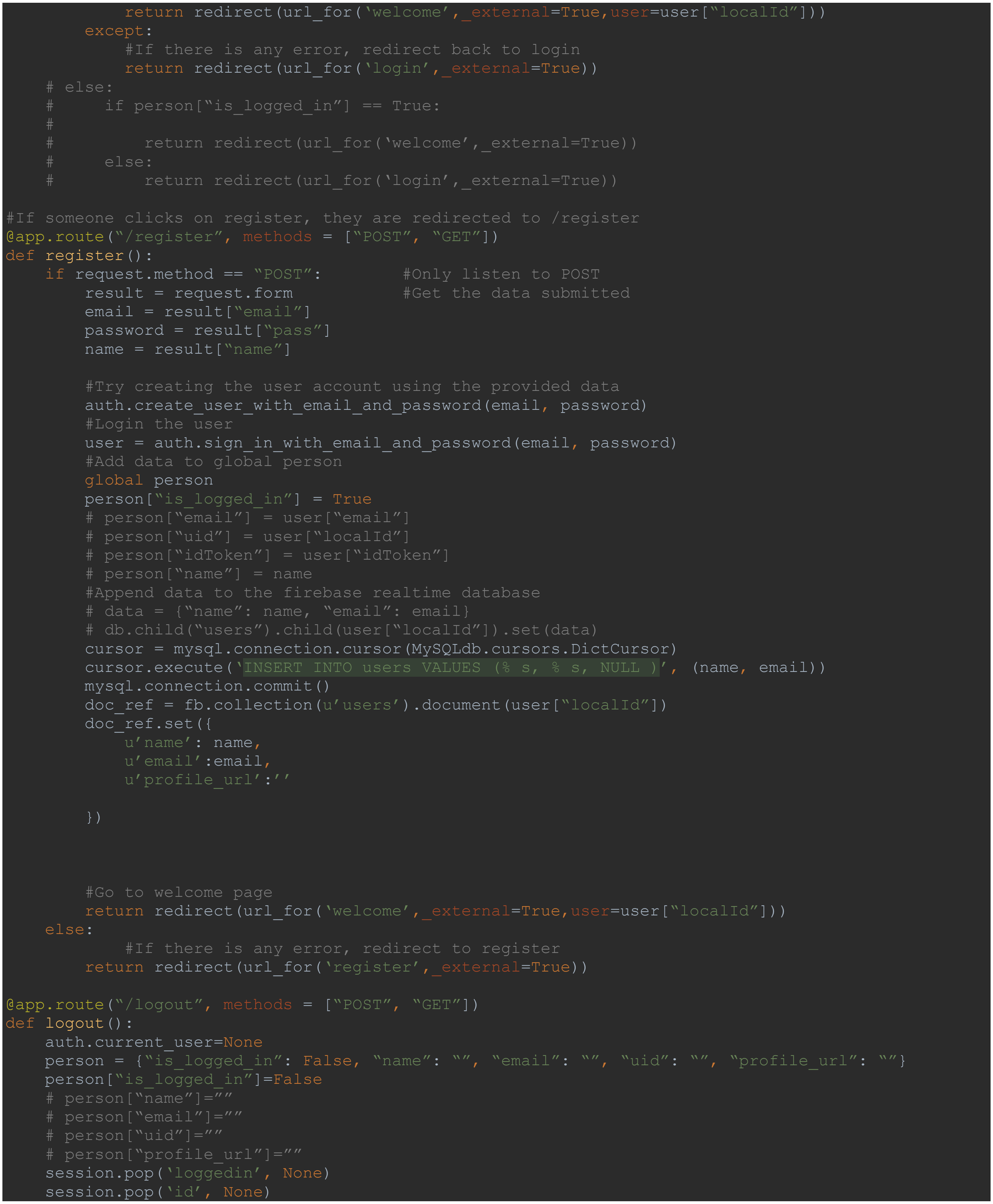
```
            user_profiles.append(profile)
            print(u’{}’.format(doc.to_dict()[‘url’]))
            print(m)








    return
render_template(“posts.html”,files_list=a,caption=c,timestamps=t,metadata=m,l=len(a),user=uid,usern
ames=usernames,user_emails=user_emails,user_profiles=user_profiles,tags=chac,key=key)
@app.route(“/details/<user>/<key>”,methods=[“POST”,”GET”])
def details(key,user):

    m=[]
    uid=user
    doc_ref = fb.collection(u’posts’).document(key)
    doc = doc_ref.get()
    a=u’{}’.format(doc.to_dict()[‘url’])
    c=u’{}’.format(doc.to_dict()[‘caption’])
    t=u’{}’.format(doc.to_dict()[‘timestamp’])
    image_location =a
    m.append(doc.to_dict()[‘metadata’])
    id = u’{}’.format(doc.to_dict()[u’user’])
    users_ref = fb.collection(u’users’).document(id)
    user_doc = users_ref.get()
    name = u’{}’.format(user_doc.to_dict()[‘name’])

    email = u’{}’.format(user_doc.to_dict()[‘email’])

    profile = u’{}’.format(user_doc.to_dict()[‘profile_url’])

    print(u’{}’.format(doc.to_dict()[‘url’]))
    print(m)



    return render_template(“details.html”, file=a, caption=c, timestamps=t, metadata=m, l=len(a),
user=uid,
                           username=name, user_email=email, user_profile=profile,
                           key=key)
@app.route(“/delete/<user>/<key>”,methods=[“POST”,”GET”])
def delete(user,key):
    uid=user
    doc_ref = fb.collection(u’posts’).document(key)
    doc = doc_ref.get()
    a = u’{}’.format(doc.to_dict()[‘url’])
    print(a)
    doc_ref_1=fb.collection(uid).document(key)
    c= doc_ref.get().to_dict()[‘characteristic’]
    print(c)
    # cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    # cursor.execute(‘DELETE FROM posts WHERE url=”%s” ‘ % (str(a)))
    # mysql.connection.commit()
    fb.collection(u’posts’).document(key).delete()
    fb.collection(uid).document(key).delete()
    for € in c:
        fb.collection(i).document(key).delete()
    return redirect(url_for(‘welcome’,_external=True,user=uid))


#Welcome page
```



```
@app.route(“/welcome/<user>”,methods=[“POST”,”GET”])
def welcome(user):
    key=[]



    for f in os.listdir(‘uploads’):
        if f.endswith(‘.jpg’):
            os.remove(os.path.join(‘uploads’, f))
    filename=””
    path_local=””
    person[“is_logged_in”] == True
    if person[“is_logged_in”] == True:
        doc_ref = fb.collection(u’users’).document(user)
        file_url = storage.child(“avtars/” + user + “.jpg”).get_url(None)
        users_ref = fb.collection(u’users’).document(user)


        profile_url = storage.child(“avtars/” + user + “.jpg”).get_url(None)
        doc = doc_ref.get()
        # name = u’{}’.format(doc.to_dict()[‘name’])
        # email=u’{}’.format(doc.to_dict()[‘email’])
        # profile=u’{}’.format(doc.to_dict()[‘profile_url’])


        # form = UploadForm()

        if request.method == ‘POST’:
                # check if the post request has the file part

            file = request.files[‘file’]
                # If the user does not select a file, the browser submits an
                # empty file without a filename.
            if file.filename == ‘’:
                flash(‘No selected file’)
                return redirect(request.url)
            if file and allowed_file(file.filename):
                filename = secure_filename(file.filename)
                file.save(os.path.join(app.config[‘UPLOADED_PHOTOS_DEST’], filename))
                path_local = app.config[‘UPLOADED_PHOTOS_DEST’] + “/” + filename
                path_on_cloud = “avtars/” + user + “.jpg”

                storage.child(path_on_cloud).put(path_local)
                file_url = storage.child(“avtars/” + user + “.jpg”).get_url(None)
                doc_ref.update({
                    u’profile_url’: file_url

                })
                # doc_ref.set({
                #         u’name’: name,
                #         u’email’: email,
                #         u’profile_url’: file_url,
                #
                #     })


                os.remove(app.config[‘UPLOADED_PHOTOS_DEST’]+”/”+ filename)


                success = True

            else:
                success = False
        #files_list = os.listdir(app.config[‘UPLOADED_PHOTOS_DEST’])
```



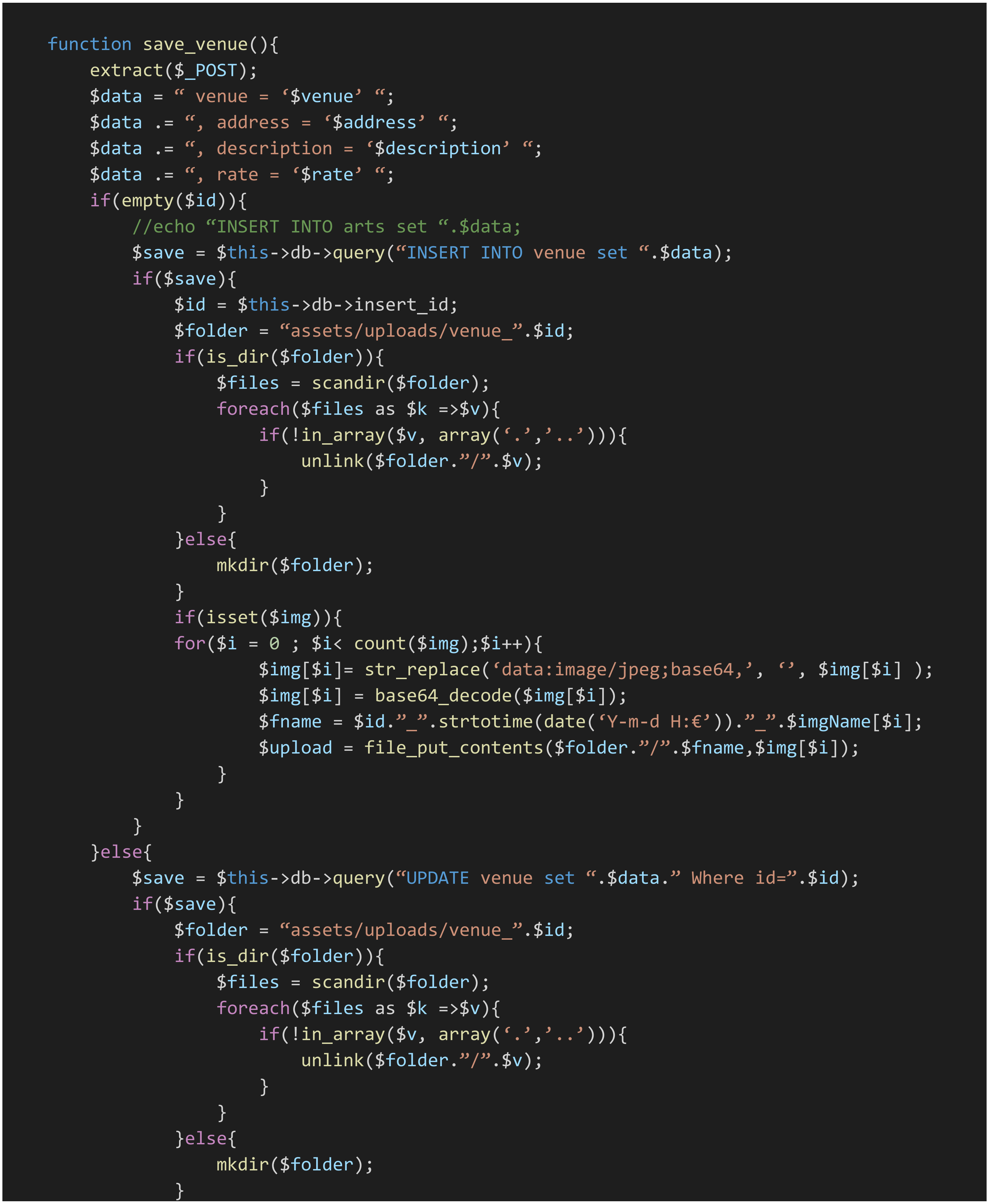
```
        #file_url = photos.url(filename)

        #profile_url=storage.child(“avtars/”+person[“uid”]+”.jpg”).get_url(None)
        u = fb.collection(user)
        d = u.stream()
        a=[]
        c=[]
        t=[]
        m=[]


        for doc in d:
            key.append(str(doc.id))
            a.append(u’{}’.format(doc.to_dict()[‘url’]))
            c.append(u’{}’.format(doc.to_dict()[‘caption’]))
            t.append(u’{}’.format(doc.to_dict()[‘timestamp’]))
            m.append(doc.to_dict()[‘metadata’])


            print(u’{}’.format(doc.to_dict()[‘url’]))
            print(m)
        doc = doc_ref.get()
        name = u’{}’.format(doc.to_dict()[‘name’])
        email=u’{}’.format(doc.to_dict()[‘email’])
        profile=u’{}’.format(doc.to_dict()[‘profile_url’])
        print(name,email,profile)

        return render_template(“welcome.html”, key=key,email = email, name =
name,uid=user,file_url=profile,files_list=a,caption=c,timestamps=t,metadata=m,l=len(a),d=d)
    else:
        return redirect(url_for(‘login’,_external=True))

#If someone clicks on login, they are redirected to /result
@app.route(“/result”, methods = [“POST”, “GET”])
def result():
    trigger=False
    if request.method == “POST”:        #Only if data has been posted
        result = request.form           #Get the data
        email = result[“email”]
        password = result[“pass”]
        try:
            #Try signing in the user with the given information
            user = auth.sign_in_with_email_and_password(email, password)
            session[‘loggedin’] = True
            session[‘id’] = user[‘localId’]
            session[‘username’] = email
            #Insert the user data in the global person
            global person
            person[“is_logged_in”] = True
            # person[“email”] = user[“email”]
            # person[“uid”] = user[“localId”]
            # person[“idToken”] = user[“idToken”]
            #Get the name of the user
            # users_ref = fb.collection(u’users’).document(user[‘localId’])
            # doc = users_ref.get()
            # bal = u’{}’.format(doc.to_dict()[‘name’])
            # print(bal)
            #
            # data = db.child(“users”).get()
            # person[“name”] = u’{}’.format(doc.to_dict()[‘name’])
            # person[“email”]=u’{}’.format(doc.to_dict()[‘email’])


            # return render_template(‘login.html’,trigger=person[“is_logged_in”])
```



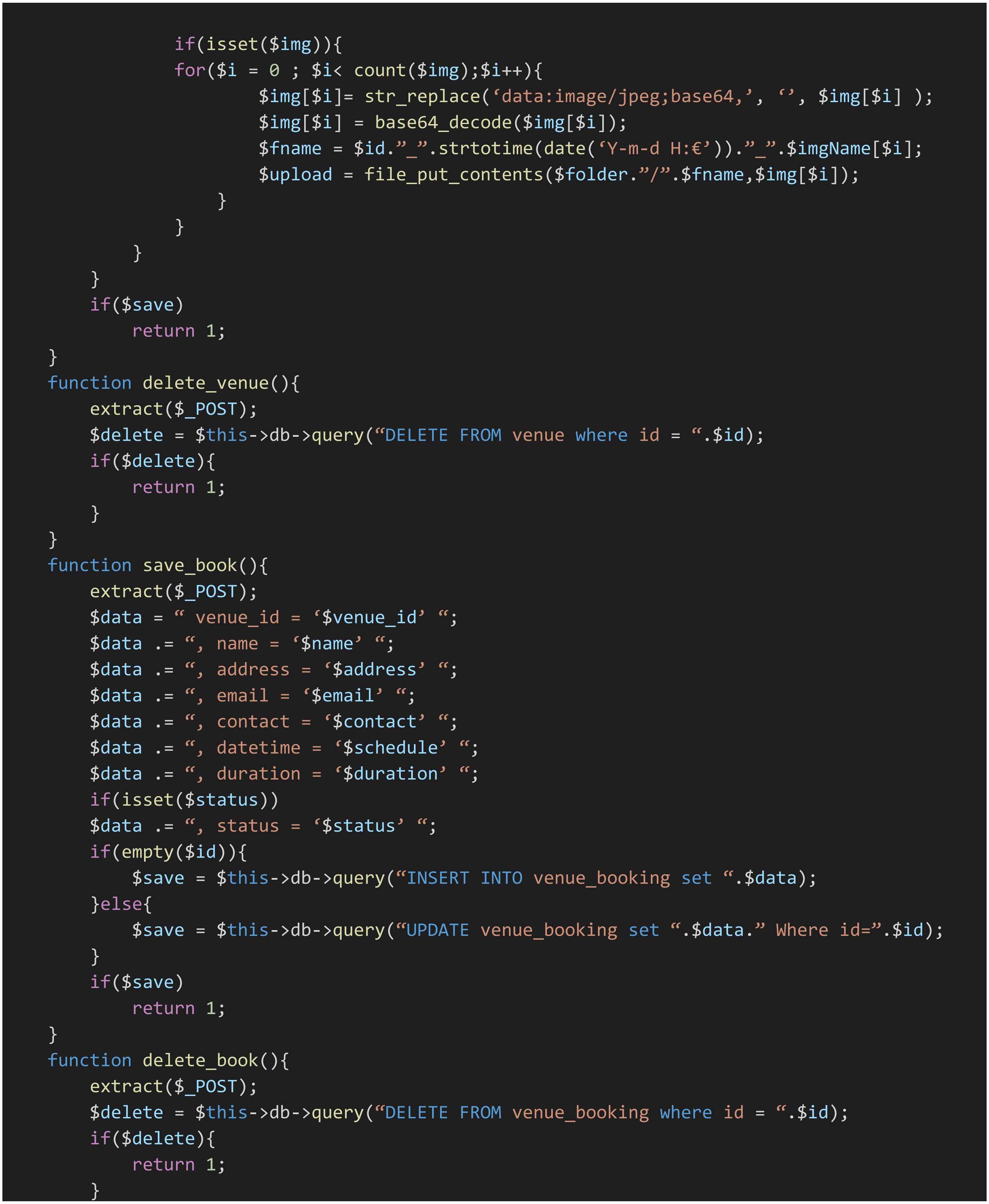
```
            return redirect(url_for(‘welcome’,_external=True,user=user[“localId”]))
        except:
            #If there is any error, redirect back to login
            return redirect(url_for(‘login’,_external=True))
    # else:
    #     if person[“is_logged_in”] == True:
    #
    #         return redirect(url_for(‘welcome’,_external=True))
    #     else:
    #         return redirect(url_for(‘login’,_external=True))

#If someone clicks on register, they are redirected to /register
@app.route(“/register”, methods = [“POST”, “GET”])
def register():
    if request.method == “POST”:        #Only listen to POST
        result = request.form           #Get the data submitted
        email = result[“email”]
        password = result[“pass”]
        name = result[“name”]

        #Try creating the user account using the provided data
        auth.create_user_with_email_and_password(email, password)
        #Login the user
        user = auth.sign_in_with_email_and_password(email, password)
        #Add data to global person
        global person
        person[“is_logged_in”] = True
        # person[“email”] = user[“email”]
        # person[“uid”] = user[“localId”]
        # person[“idToken”] = user[“idToken”]
        # person[“name”] = name
        #Append data to the firebase realtime database
        # data = {“name”: name, “email”: email}
        # db.child(“users”).child(user[“localId”]).set(data)
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute(‘INSERT INTO users VALUES (% s, % s, NULL )’, (name, email))
        mysql.connection.commit()
        doc_ref = fb.collection(u’users’).document(user[“localId”])
        doc_ref.set({
            u’name’: name,
            u’email’:email,
            u’profile_url’:’’

        })



        #Go to welcome page
        return redirect(url_for(‘welcome’,_external=True,user=user[“localId”]))
    else:
            #If there is any error, redirect to register
        return redirect(url_for(‘register’,_external=True))

@app.route(“/logout”, methods = [“POST”, “GET”])
def logout():
    auth.current_user=None
    person = {“is_logged_in”: False, “name”: “”, “email”: “”, “uid”: “”, “profile_url”: “”}
    person[“is_logged_in”]=False
    # person[“name”]=””
    # person[“email”]=””
    # person[“uid”]=””
    # person[“profile_url”]=””
    session.pop(‘loggedin’, None)
    session.pop(‘id’, None)
    session.pop(‘username’, None)
    return redirect(url_for(‘login’,_external=True))

if __name__ == “__main__”:
    # port = int(os.environ.get(‘PORT’, 5000))
    app.run(debug=True)
```

Admin_class.php

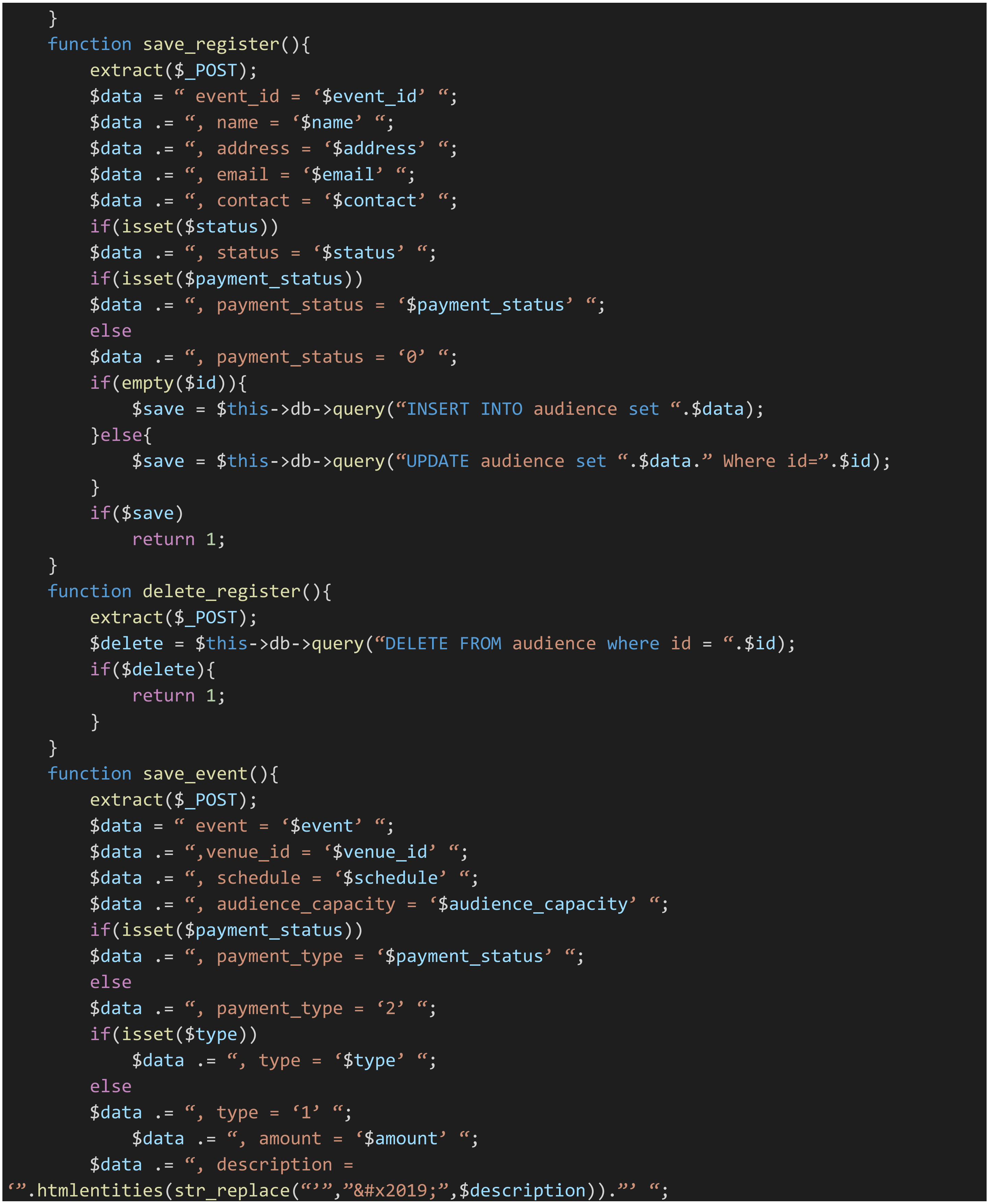
```
<?php
session_start();
ini_set(‘display_errors’, 1);
Class Action {
    private $db;

    public function __construct() {
        ob_start();
    include ‘db_connect.php’;

    $this->db = $conn;
    }
    function __destruct() {
        $this->db->close();
        ob_end_flush();
    }

    function login(){
        extract($_POST);
        $qry = $this->db->query(“SELECT * FROM users where username = ‘”.$username.”’ And
password = ‘”.md5($password).”’ “);
        if($qry->num_rows > 0){
            foreach ($qry->fetch_array() as $key => $value) {
                if($key != ‘passwors’ && !is_numeric($key))
                    $_SESSION[‘login_’.$key] = $value;
            }
                return 1;
        }else{
            return 3;
        }
    }
    function logout(){
        session_destroy();
        foreach ($_SESSION as $key => $value) {
            unset($_SESSION[$key]);
        }
        header(“location:login.php”);
    }
    function save_user(){
        extract($_POST);
        $data = “ name = ‘$name’ “;
        $data .= “, username = ‘$username’ “;
        if(!empty($password))
        $data .= “, password = ‘”.md5($password).”’ “;
        $data .= “, type = ‘$type’ “;
        if($type == 1)
            $establishment_id = 0;
        $data .= “, establishment_id = ‘$establishment_id’ “;
        $chk = $this->db->query(“Select * from users where username = ‘$username’ and id
!=’$id’ “)->num_rows;
        if($chk > 0){
            return 2;
            exit;
        }
        if(empty($id)){
            $save = $this->db->query(“INSERT INTO users set “.$data);
        }else{
            $save = $this->db->query(“UPDATE users set “.$data.” Where id = “.$id);
        }
        if($save){
            return 1;
        }
    }
    function delete_user(){
        extract($_POST);
        $delete = $this->db->query(“DELETE FROM users where id = “.$id);
        if($delete)
            return 1;
    }
    function signup(){
        extract($_POST);
        $data = “ name = ‘$name’ “;
        $data .= “, contact = ‘$contact’ “;
        $data .= “, address = ‘$address’ “;
        $data .= “, username = ‘$email’ “;
        $data .= “, password = ‘”.md5($password).”’ “;
        $data .= “, type = 3”;
        $chk = $this->db->query(“SELECT * FROM users where username = ‘$email’ “)->num_rows;
        if($chk > 0){
            return 2;
            exit;
        }
            $save = $this->db->query(“INSERT INTO users set “.$data);
        if($save){
            $qry = $this->db->query(“SELECT * FROM users where username = ‘”.$email.”’ And
password = ‘”.md5($password).”’ “);
            if($qry->num_rows > 0){
                foreach ($qry->fetch_array() as $key => $value) {
                    if($key != ‘passwors’ && !is_numeric($key))
                        $_SESSION[‘login_’.$key] = $value;
                }
            }
            return 1;
        }
    }

    function save_settings(){
        extract($_POST);
        $data = “ name = ‘”.str_replace(“’”,”&#x2019;”,$name).”’ “;
        $data .= “, email = ‘$email’ “;
        $data .= “, contact = ‘$contact’ “;
        $data .= “, about_content = ‘”.htmlentities(str_replace(“’”,”&#x2019;”,$about)).”’
“;
        if($_FILES[‘img’][‘tmp_name’] != ‘’){
                        $fname = strtotime(date(‘y-m-d H:€’)).’_’.$_FILES[‘img’][‘name’];
                        $move =
move_uploaded_file($_FILES[‘img’][‘tmp_name’],’assets/uploads/’. $fname);
                    $data .= “, cover_img = ‘$fname’ “;

        }

        // echo “INSERT INTO system_settings set “.$data;
        $chk = $this->db->query(“SELECT * FROM system_settings”);
        if($chk->num_rows > 0){
            $save = $this->db->query(“UPDATE system_settings set “.$data);
        }else{
            $save = $this->db->query(“INSERT INTO system_settings set “.$data);
        }
        if($save){
        $query = $this->db->query(“SELECT * FROM system_settings limit 1”)->fetch_array();
        foreach ($query as $key => $value) {
            if(!is_numeric($key))
                $_SESSION[‘settings’][$key] = $value;
        }

            return 1;
                }
    }
```



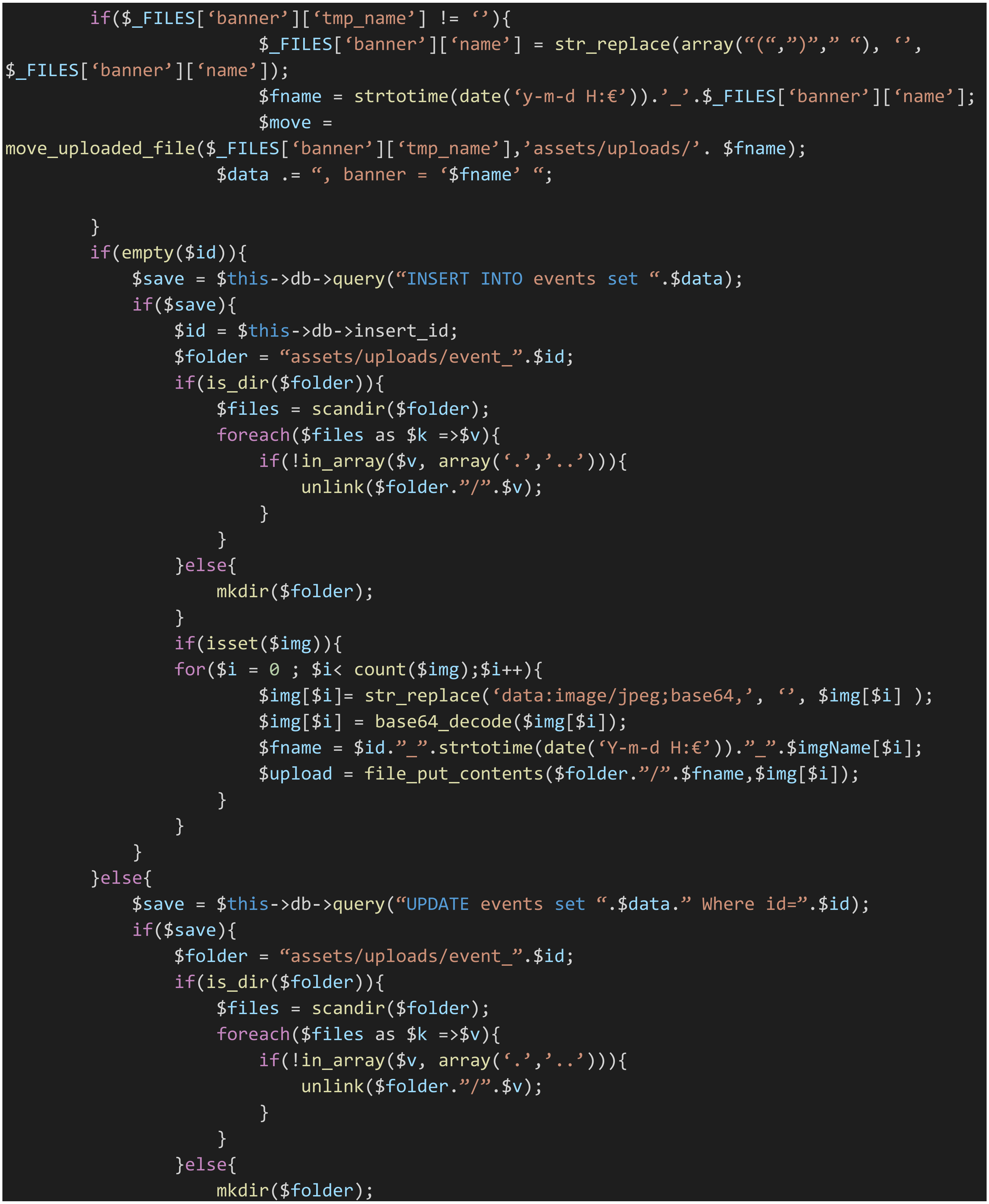
```
function save_venue(){
    extract($_POST);
    $data = “ venue = ‘$venue’ “;
    $data .= “, address = ‘$address’ “;
    $data .= “, description = ‘$description’ “;
    $data .= “, rate = ‘$rate’ “;
    if(empty($id)){
        //echo “INSERT INTO arts set “.$data;
        $save = $this->db->query(“INSERT INTO venue set “.$data);
        if($save){
            $id = $this->db->insert_id;
            $folder = “assets/uploads/venue_”.$id;
            if(is_dir($folder)){
                $files = scandir($folder);
                foreach($files as $k =>$v){
                    if(!in_array($v, array(‘.’,’..’))){
                        unlink($folder.”/”.$v);
                    }
                }
            }else{
                mkdir($folder);
            }
            if(isset($img)){
            for($i = 0 ; $i< count($img);$i++){
                    $img[$i]= str_replace(‘data:image/jpeg;base64,’, ‘’, $img[$i] );
                    $img[$i] = base64_decode($img[$i]);
                    $fname = $id.”_”.strtotime(date(‘Y-m-d H:€’)).”_”.$imgName[$i];
                    $upload = file_put_contents($folder.”/”.$fname,$img[$i]);
                }
            }
        }
    }else{
        $save = $this->db->query(“UPDATE venue set “.$data.” Where id=”.$id);
        if($save){
            $folder = “assets/uploads/venue_”.$id;
            if(is_dir($folder)){
                $files = scandir($folder);
                foreach($files as $k =>$v){
                    if(!in_array($v, array(‘.’,’..’))){
                        unlink($folder.”/”.$v);
                    }
                }
            }else{
                mkdir($folder);
            }
```



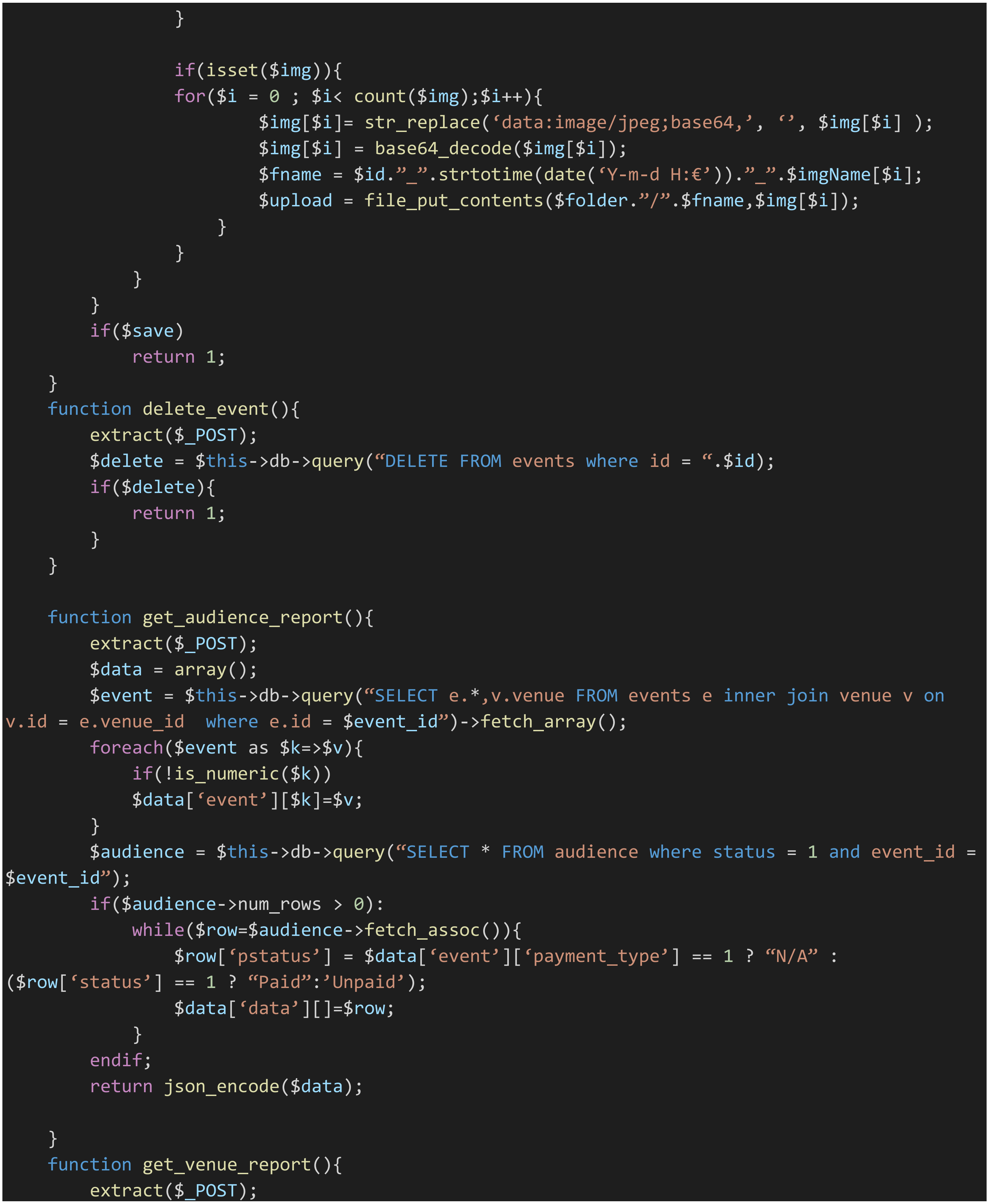
```
                if(isset($img)){
                for($i = 0 ; $i< count($img);$i++){
                        $img[$i]= str_replace(‘data:image/jpeg;base64,’, ‘’, $img[$i] );
                        $img[$i] = base64_decode($img[$i]);
                        $fname = $id.”_”.strtotime(date(‘Y-m-d H:€’)).”_”.$imgName[$i];
                        $upload = file_put_contents($folder.”/”.$fname,$img[$i]);
                    }
                }
            }
        }
        if($save)
            return 1;
    }
    function delete_venue(){
        extract($_POST);
        $delete = $this->db->query(“DELETE FROM venue where id = “.$id);
        if($delete){
            return 1;
        }
    }
    function save_book(){
        extract($_POST);
        $data = “ venue_id = ‘$venue_id’ “;
        $data .= “, name = ‘$name’ “;
        $data .= “, address = ‘$address’ “;
        $data .= “, email = ‘$email’ “;
        $data .= “, contact = ‘$contact’ “;
        $data .= “, datetime = ‘$schedule’ “;
        $data .= “, duration = ‘$duration’ “;
        if(isset($status))
        $data .= “, status = ‘$status’ “;
        if(empty($id)){
            $save = $this->db->query(“INSERT INTO venue_booking set “.$data);
        }else{
            $save = $this->db->query(“UPDATE venue_booking set “.$data.” Where id=”.$id);
        }
        if($save)
            return 1;
    }
    function delete_book(){
        extract($_POST);
        $delete = $this->db->query(“DELETE FROM venue_booking where id = “.$id);
        if($delete){
            return 1;
        }
```



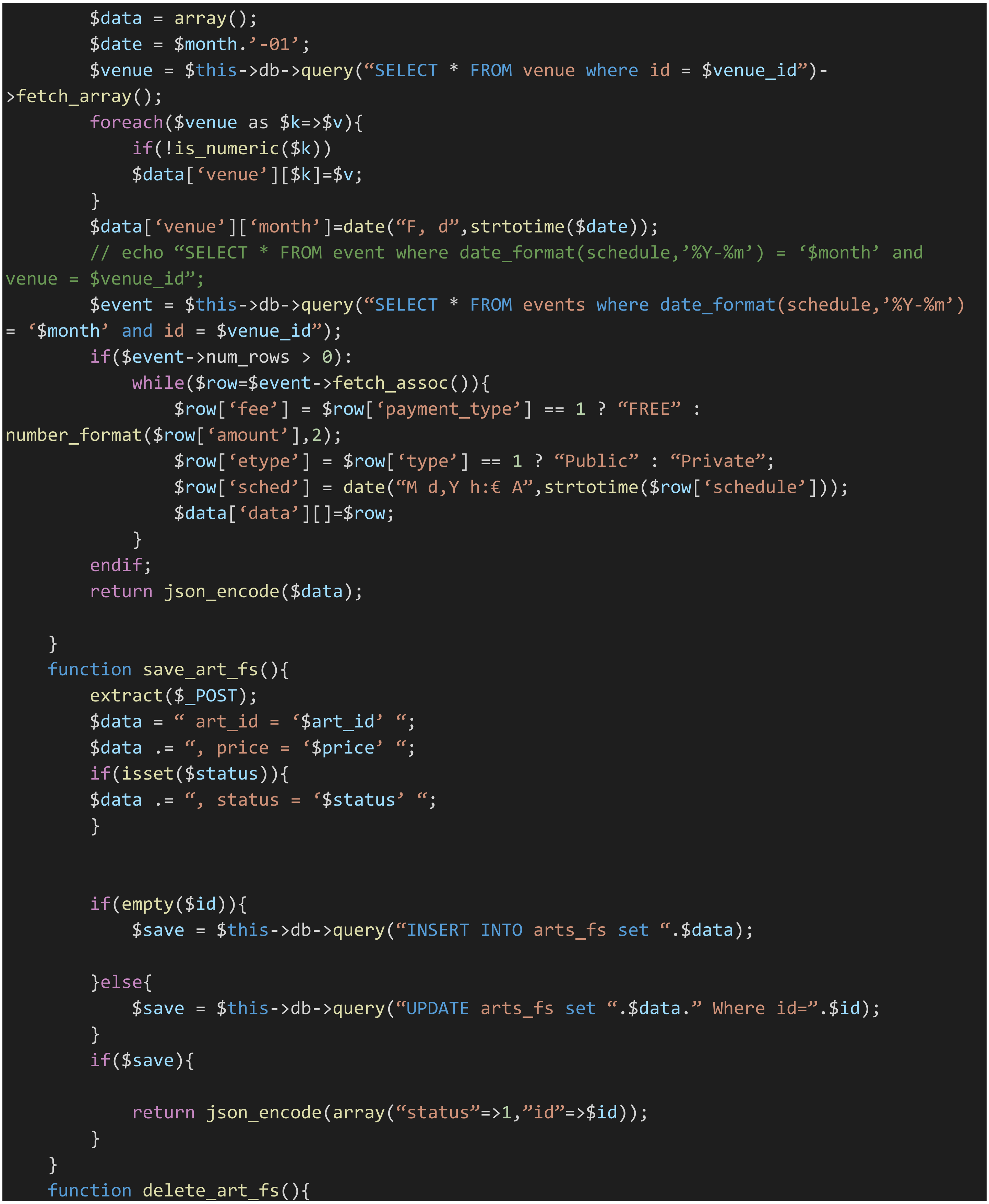
```
  }
  function save_register(){
    extract($_POST);
    $data = “ event_id = ‘$event_id’ “;
    $data .= “, name = ‘$name’ “;
    $data .= “, address = ‘$address’ “;
    $data .= “, email = ‘$email’ “;
    $data .= “, contact = ‘$contact’ “;
    if(isset($status))
    $data .= “, status = ‘$status’ “;
    if(isset($payment_status))
    $data .= “, payment_status = ‘$payment_status’ “;
    else
    $data .= “, payment_status = ‘0’ “;
    if(empty($id)){
      $save = $this->db->query(“INSERT INTO audience set “.$data);
    }else{
      $save = $this->db->query(“UPDATE audience set “.$data.” Where id=”.$id);
    }
    if($save)
      return 1;
  }
  function delete_register(){
    extract($_POST);
    $delete = $this->db->query(“DELETE FROM audience where id = “.$id);
    if($delete){
      return 1;
    }
  }
  function save_event(){
    extract($_POST);
    $data = “ event = ‘$event’ “;
    $data .= “,venue_id = ‘$venue_id’ “;
    $data .= “, schedule = ‘$schedule’ “;
    $data .= “, audience_capacity = ‘$audience_capacity’ “;
    if(isset($payment_status))
    $data .= “, payment_type = ‘$payment_status’ “;
    else
    $data .= “, payment_type = ‘2’ “;
    if(isset($type))
      $data .= “, type = ‘$type’ “;
    else
    $data .= “, type = ‘1’ “;
      $data .= “, amount = ‘$amount’ “;
    $data .= “, description =
‘”.htmlentities(str_replace(“’”,”&#x2019;”,$description)).”’ “;
```



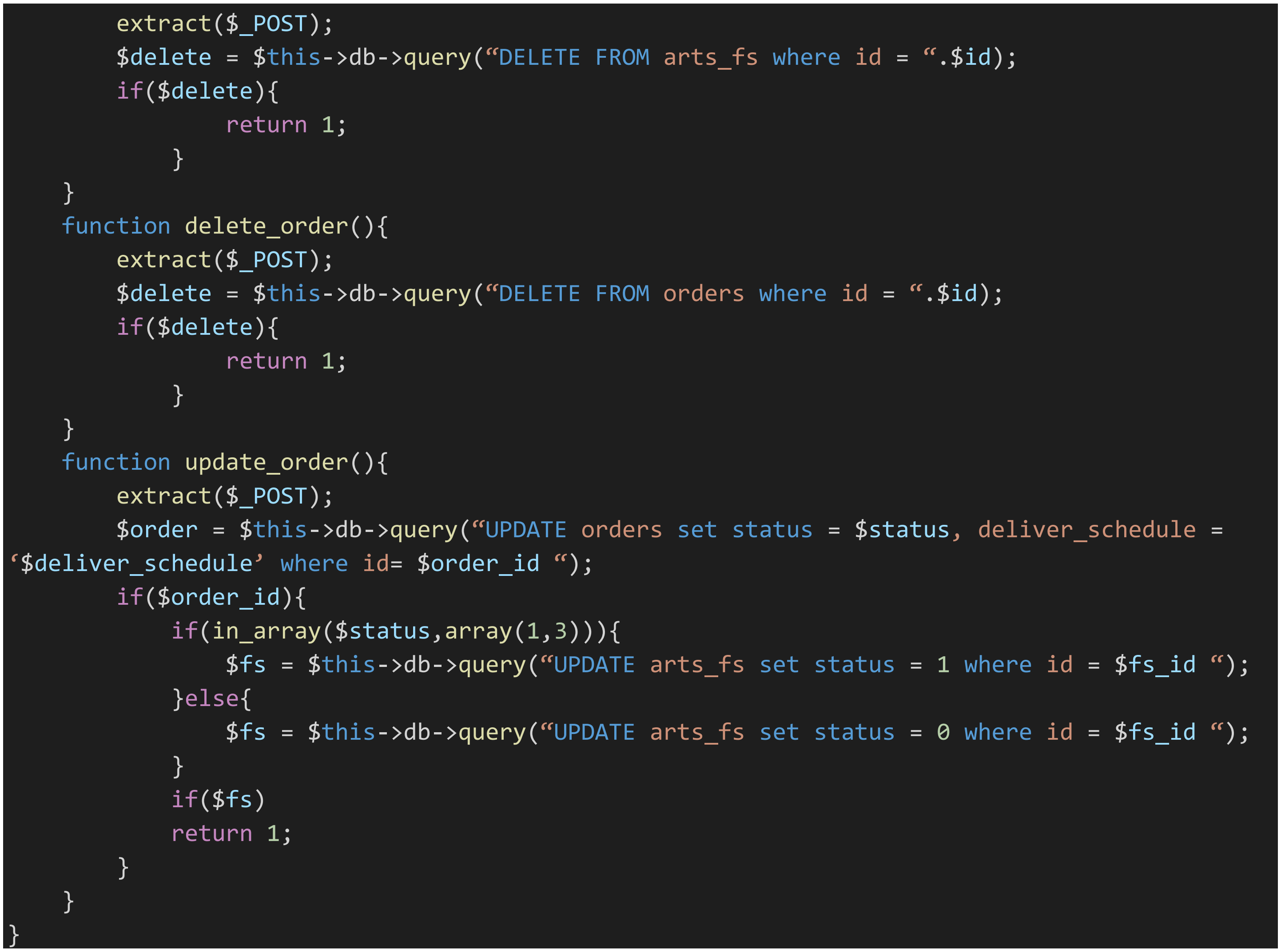
```
        if($_FILES[‘banner’][‘tmp_name’] != ‘’){
                        $_FILES[‘banner’][‘name’] = str_replace(array(“(“,”)”,” “), ‘’,
$_FILES[‘banner’][‘name’]);
                        $fname = strtotime(date(‘y-m-d H:€’)).’_’.$_FILES[‘banner’][‘name’];
                        $move =
move_uploaded_file($_FILES[‘banner’][‘tmp_name’],’assets/uploads/’. $fname);
                    $data .= “, banner = ‘$fname’ “;

        }
        if(empty($id)){
            $save = $this->db->query(“INSERT INTO events set “.$data);
            if($save){
                $id = $this->db->insert_id;
                $folder = “assets/uploads/event_”.$id;
                if(is_dir($folder)){
                    $files = scandir($folder);
                    foreach($files as $k =>$v){
                        if(!in_array($v, array(‘.’,’..’))){
                            unlink($folder.”/”.$v);
                        }
                    }
                }else{
                    mkdir($folder);
                }
                if(isset($img)){
                for($i = 0 ; $i< count($img);$i++){
                        $img[$i]= str_replace(‘data:image/jpeg;base64,’, ‘’, $img[$i] );
                        $img[$i] = base64_decode($img[$i]);
                        $fname = $id.”_”.strtotime(date(‘Y-m-d H:€’)).”_”.$imgName[$i];
                        $upload = file_put_contents($folder.”/”.$fname,$img[$i]);
                    }
                }
            }
        }else{
            $save = $this->db->query(“UPDATE events set “.$data.” Where id=”.$id);
            if($save){
                $folder = “assets/uploads/event_”.$id;
                if(is_dir($folder)){
                    $files = scandir($folder);
                    foreach($files as $k =>$v){
                        if(!in_array($v, array(‘.’,’..’))){
                            unlink($folder.”/”.$v);
                        }
                    }
                }else{
                    mkdir($folder);
```



```
                }

                if(isset($img)){
                for($i = 0 ; $i< count($img);$i++){
                        $img[$i]= str_replace(‘data:image/jpeg;base64,’, ‘’, $img[$i] );
                        $img[$i] = base64_decode($img[$i]);
                        $fname = $id.”_”.strtotime(date(‘Y-m-d H:€’)).”_”.$imgName[$i];
                        $upload = file_put_contents($folder.”/”.$fname,$img[$i]);
                    }
                }
            }
        }
        if($save)
            return 1;
    }
    function delete_event(){
        extract($_POST);
        $delete = $this->db->query(“DELETE FROM events where id = “.$id);
        if($delete){
            return 1;
        }
    }

    function get_audience_report(){
        extract($_POST);
        $data = array();
        $event = $this->db->query(“SELECT e.*,v.venue FROM events e inner join venue v on
v.id = e.venue_id  where e.id = $event_id”)->fetch_array();
        foreach($event as $k=>$v){
            if(!is_numeric($k))
            $data[‘event’][$k]=$v;
        }
        $audience = $this->db->query(“SELECT * FROM audience where status = 1 and event_id =
$event_id”);
        if($audience->num_rows > 0):
            while($row=$audience->fetch_assoc()){
                $row[‘pstatus’] = $data[‘event’][‘payment_type’] == 1 ? “N/A” :
($row[‘status’] == 1 ? “Paid”:’Unpaid’);
                $data[‘data’][]=$row;
            }
        endif;
        return json_encode($data);

    }
    function get_venue_report(){
        extract($_POST);
```

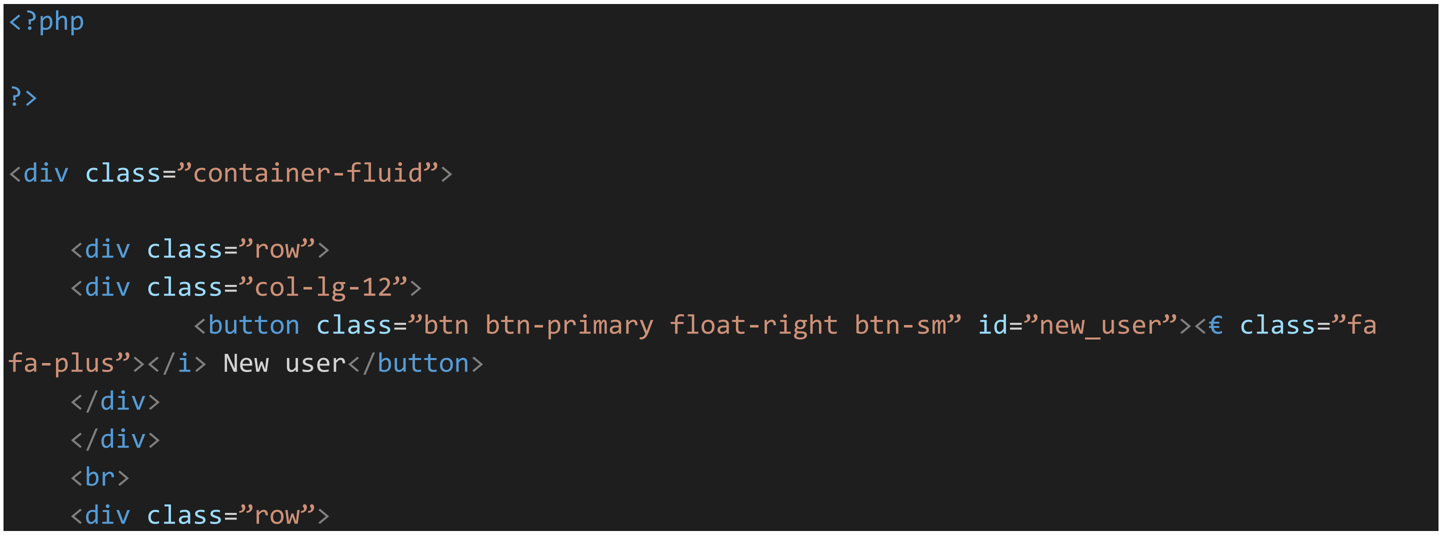


```
    $data = array();
    $date = $month.’-01’;
    $venue = $this->db->query(“SELECT * FROM venue where id = $venue_id”)-
>fetch_array();
    foreach($venue as $k=>$v){
      if(!is_numeric($k))
      $data[‘venue’][$k]=$v;
    }
    $data[‘venue’][‘month’]=date(“F, d”,strtotime($date));
    // echo “SELECT * FROM event where date_format(schedule,’%Y-%m’) = ‘$month’ and
venue = $venue_id”;
    $event = $this->db->query(“SELECT * FROM events where date_format(schedule,’%Y-%m’)
= ‘$month’ and id = $venue_id”);
    if($event->num_rows > 0):
      while($row=$event->fetch_assoc()){
        $row[‘fee’] = $row[‘payment_type’] == 1 ? “FREE” :
number_format($row[‘amount’],2);
        $row[‘etype’] = $row[‘type’] == 1 ? “Public” : “Private”;
        $row[‘sched’] = date(“M d,Y h:€ A”,strtotime($row[‘schedule’]));
        $data[‘data’][]=$row;
      }
    endif;
    return json_encode($data);

  }
  function save_art_fs(){
    extract($_POST);
    $data = “ art_id = ‘$art_id’ “;
    $data .= “, price = ‘$price’ “;
    if(isset($status)){
    $data .= “, status = ‘$status’ “;
    }


    if(empty($id)){
      $save = $this->db->query(“INSERT INTO arts_fs set “.$data);

    }else{
      $save = $this->db->query(“UPDATE arts_fs set “.$data.” Where id=”.$id);
    }
    if($save){

      return json_encode(array(“status”=>1,”id”=>$id));
    }
  }
  function delete_art_fs(){
```

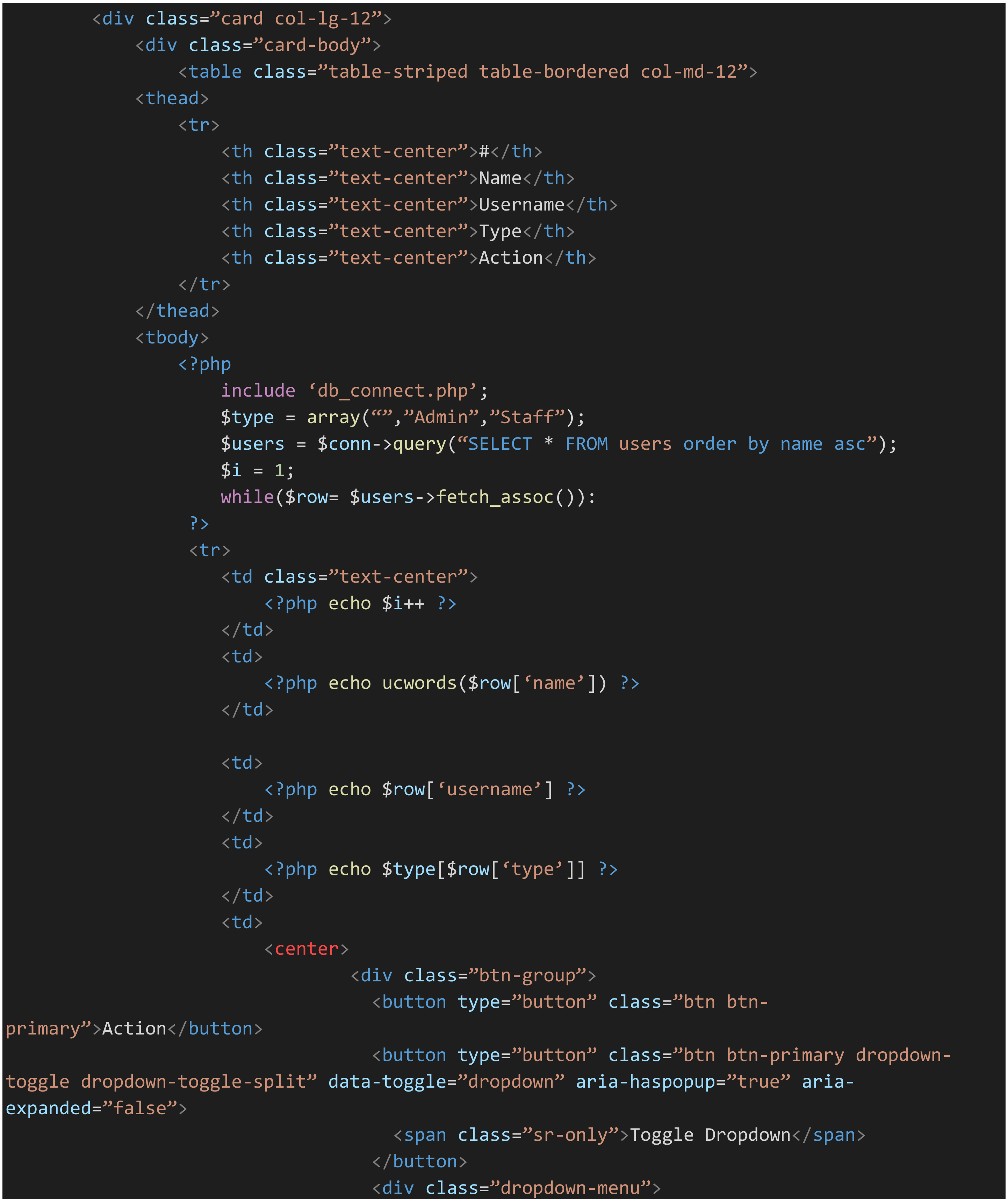


```
        extract($_POST);
        $delete = $this->db->query("DELETE FROM arts_fs where id = ".$id);
        if($delete){
                return 1;
            }
    }
    function delete_order(){
        extract($_POST);
        $delete = $this->db->query("DELETE FROM orders where id = ".$id);
        if($delete){
                return 1;
            }
    }
    function update_order(){
        extract($_POST);
        $order = $this->db->query("UPDATE orders set status = $status, deliver_schedule = 
'$deliver_schedule' where id= $order_id ");
        if($order_id){
            if(in_array($status,array(1,3))){
                $fs = $this->db->query("UPDATE arts_fs set status = 1 where id = $fs_id ");
            }else{
                $fs = $this->db->query("UPDATE arts_fs set status = 0 where id = $fs_id ");
            }
            if($fs)
            return 1;
        }
    }
}
```

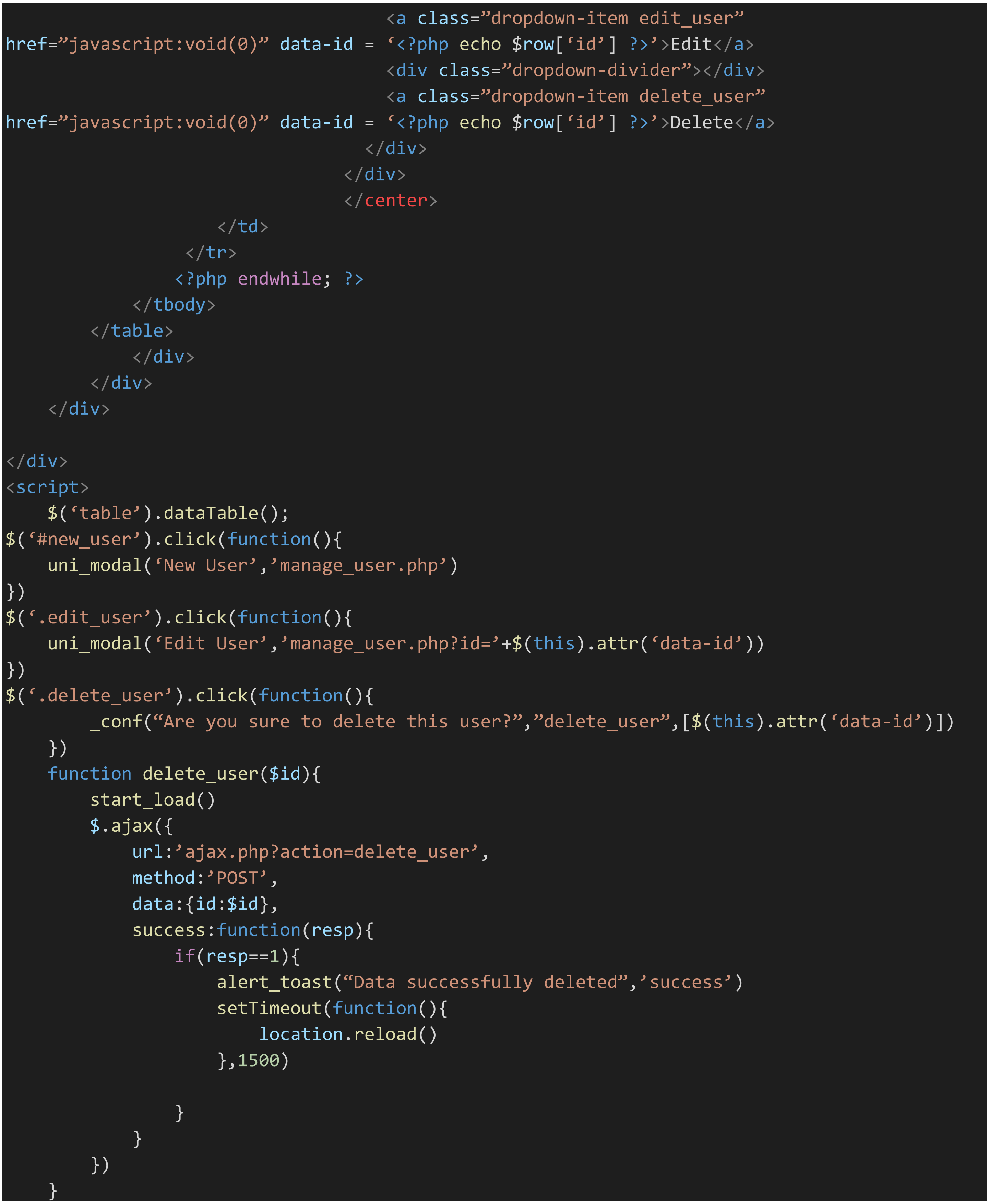
Users.php:



```
<?php

?>

<div class="container-fluid">

    <div class="row">
    <div class="col-lg-12">
            <button class="btn btn-primary float-right btn-sm" id="new_user"><€ class="fa 
fa-plus"></i> New user</button>
    </div>
    </div>
    <br>
    <div class="row">
```



```
        <div class=”card col-lg-12”>
            <div class=”card-body”>
                <table class=”table-striped table-bordered col-md-12”>
            <thead>
                <tr>
                    <th class=”text-center”>#</th>
                    <th class=”text-center”>Name</th>
                    <th class=”text-center”>Username</th>
                    <th class=”text-center”>Type</th>
                    <th class=”text-center”>Action</th>
                </tr>
            </thead>
            <tbody>
                <?php
                    include ‘db_connect.php’;
                    $type = array(“”,”Admin”,”Staff”);
                    $users = $conn->query(“SELECT * FROM users order by name asc”);
                    $i = 1;
                    while($row= $users->fetch_assoc()):
                 ?>
                 <tr>
                    <td class=”text-center”>
                        <?php echo $i++ ?>
                    </td>
                    <td>
                        <?php echo ucwords($row[‘name’]) ?>
                    </td>

                    <td>
                        <?php echo $row[‘username’] ?>
                    </td>
                    <td>
                        <?php echo $type[$row[‘type’]] ?>
                    </td>
                    <td>
                        <center>
                                <div class=”btn-group”>
                                  <button type=”button” class=”btn btn-
primary”>Action</button>
                                  <button type=”button” class=”btn btn-primary dropdown-
toggle dropdown-toggle-split” data-toggle=”dropdown” aria-haspopup=”true” aria-
expanded=”false”>
                                    <span class=”sr-only”>Toggle Dropdown</span>
                                  </button>
                                  <div class=”dropdown-menu”>
```

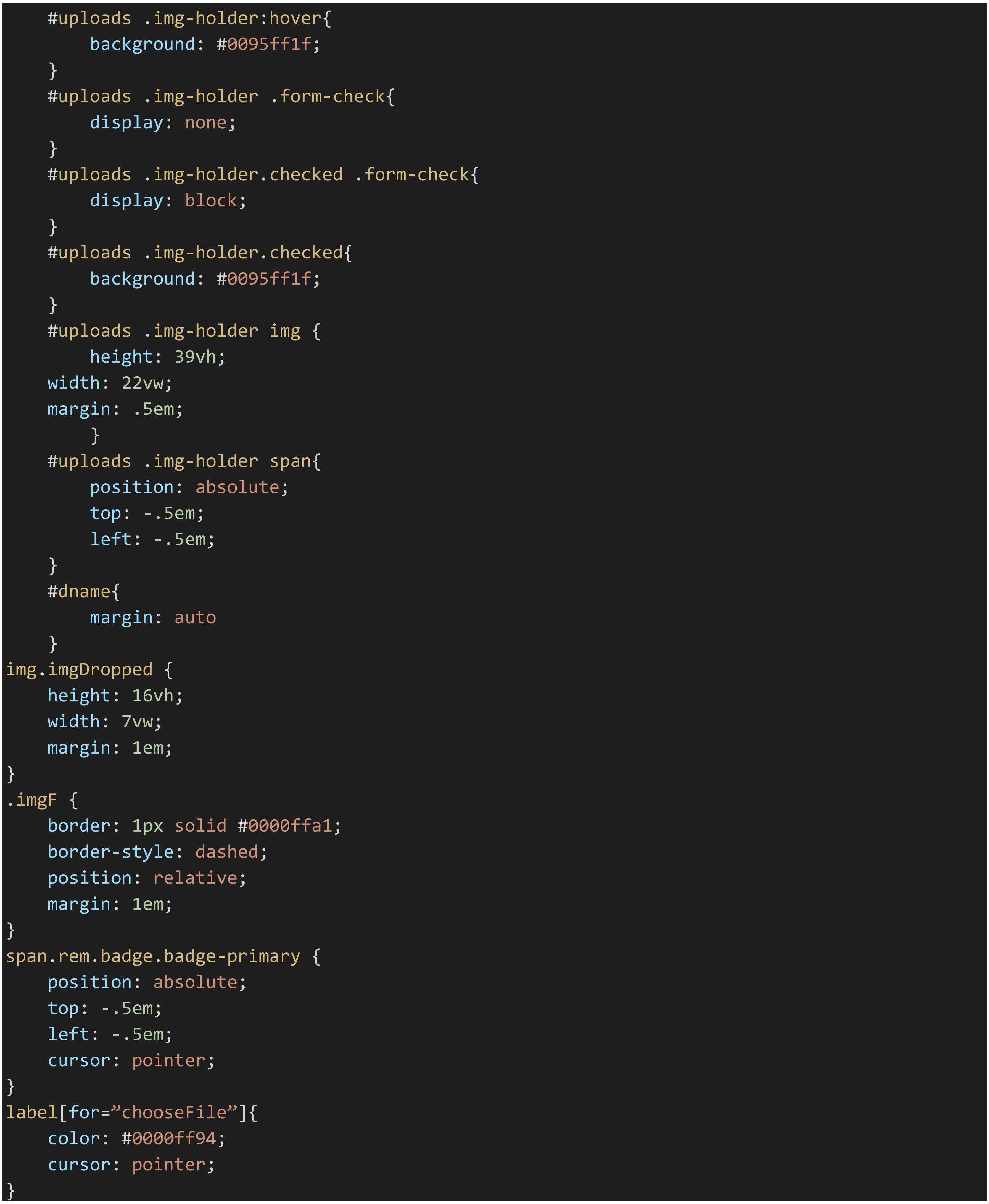


```
                                    <a class=”dropdown-item edit_user”
href=”javascript:void(0)” data-id = ‘<?php echo $row[‘id’] ?>’>Edit</a>
                                    <div class=”dropdown-divider”></div>
                                    <a class=”dropdown-item delete_user”
href=”javascript:void(0)” data-id = ‘<?php echo $row[‘id’] ?>’>Delete</a>
                                  </div>
                                </div>
                                </center>
                    </td>
                 </tr>
                <?php endwhile; ?>
            </tbody>
        </table>
            </div>
        </div>
    </div>

</div>
<script>
    $(‘table’).dataTable();
$(‘#new_user’).click(function(){
    uni_modal(‘New User’,’manage_user.php’)
})
$(‘.edit_user’).click(function(){
    uni_modal(‘Edit User’,’manage_user.php?id=’+$(this).attr(‘data-id’))
})
$(‘.delete_user’).click(function(){
        _conf(“Are you sure to delete this user?”,”delete_user”,[$(this).attr(‘data-id’)])
    })
    function delete_user($id){
        start_load()
        $.ajax({
            url:’ajax.php?action=delete_user’,
            method:’POST’,
            data:{id:$id},
            success:function(resp){
                if(resp==1){
                    alert_toast(“Data successfully deleted”,’success’)
                    setTimeout(function(){
                        location.reload()
                    },1500)

                }
            }
        })
    }
```

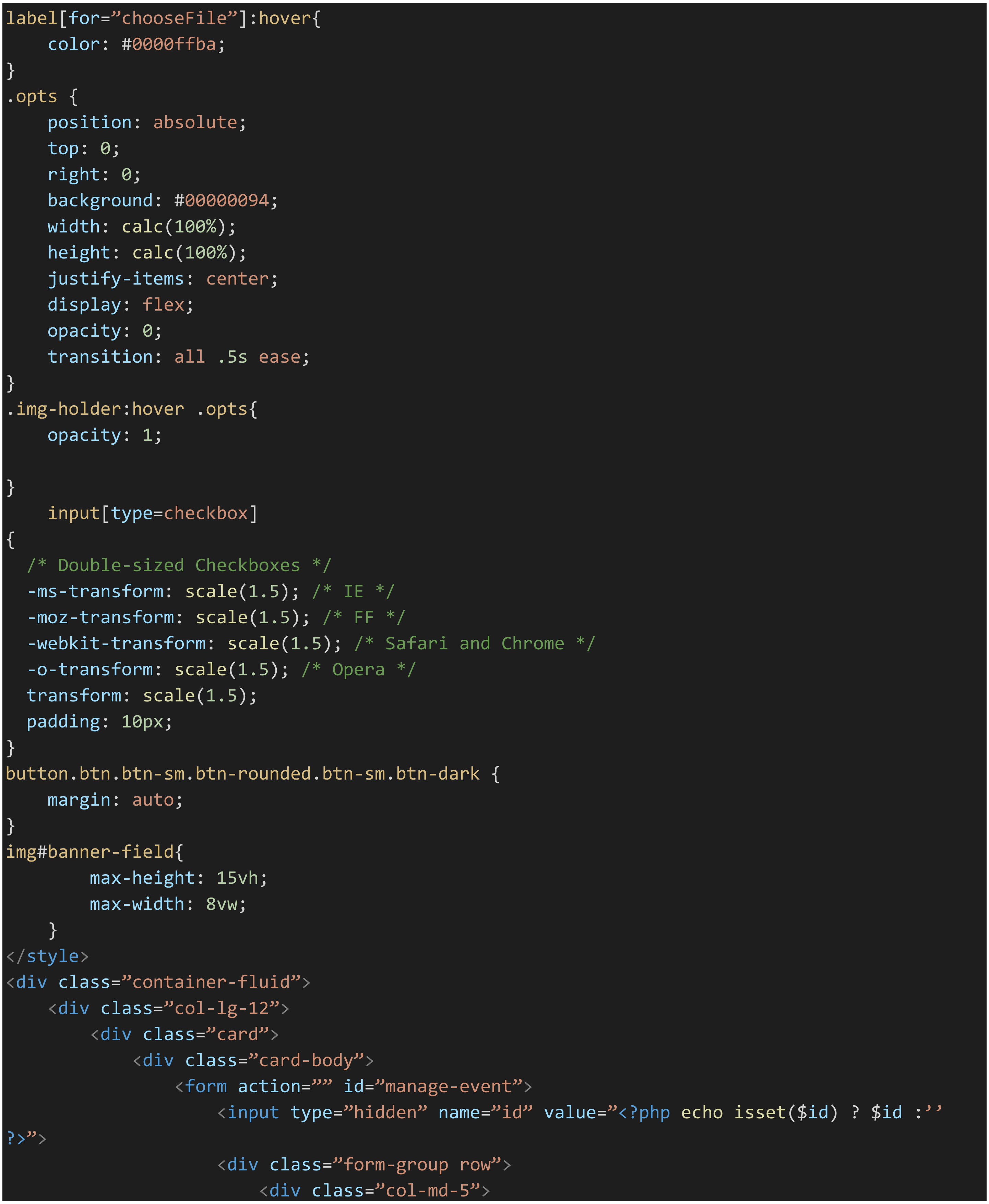
```
</script>
```

MangeEvents.php

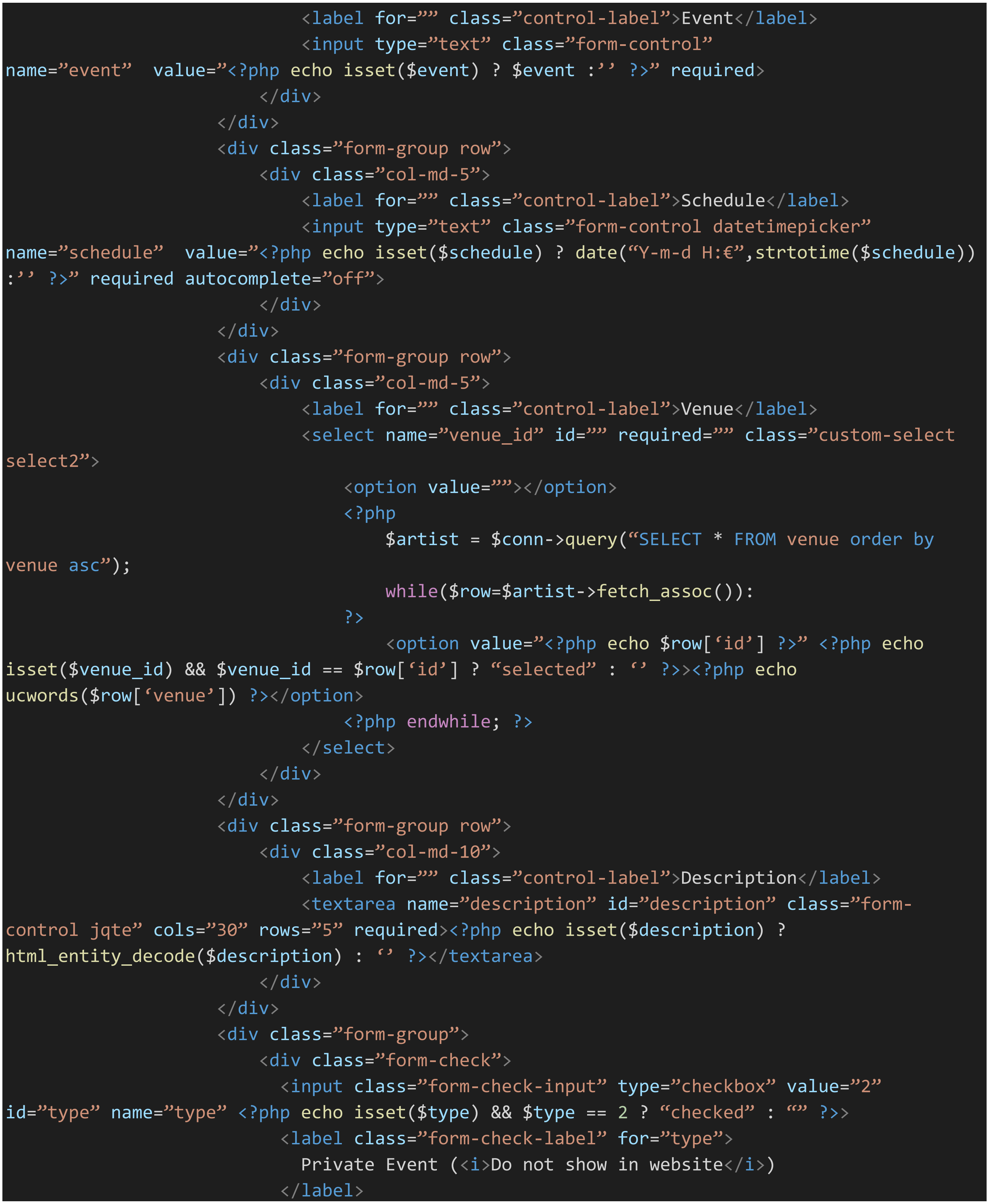
```
<?php include ‘db_connect.php’ ?>
<?php
if(isset($_GET[‘id’])){
$qry = $conn->query(“SELECT * FROM events where id= “.$_GET[‘id’]);
foreach($qry->fetch_array() as $k => $val){
    $$k=$val;
}
}
?>
<style>

    .jqte_editor{
        min-height: 30vh !important
    }
    #drop {
    min-height: 15vh;
    max-height: 30vh;
    overflow: auto;
    width: calc(100%);
    border: 5px solid #929292;
    margin: 10px;
    border-style: dashed;
    padding: 10px;
    display: flex;
    align-items: center;
    flex-wrap: wrap;
}
    #uploads {
        min-height: 15vh;
    width: calc(100%);
    margin: 10px;
    padding: 10px;
    display: flex;
    align-items: center;
    flex-wrap: wrap;
    }
    #uploads .img-holder{
        position: relative;
        margin: 1em;
        cursor: pointer;
    }
```



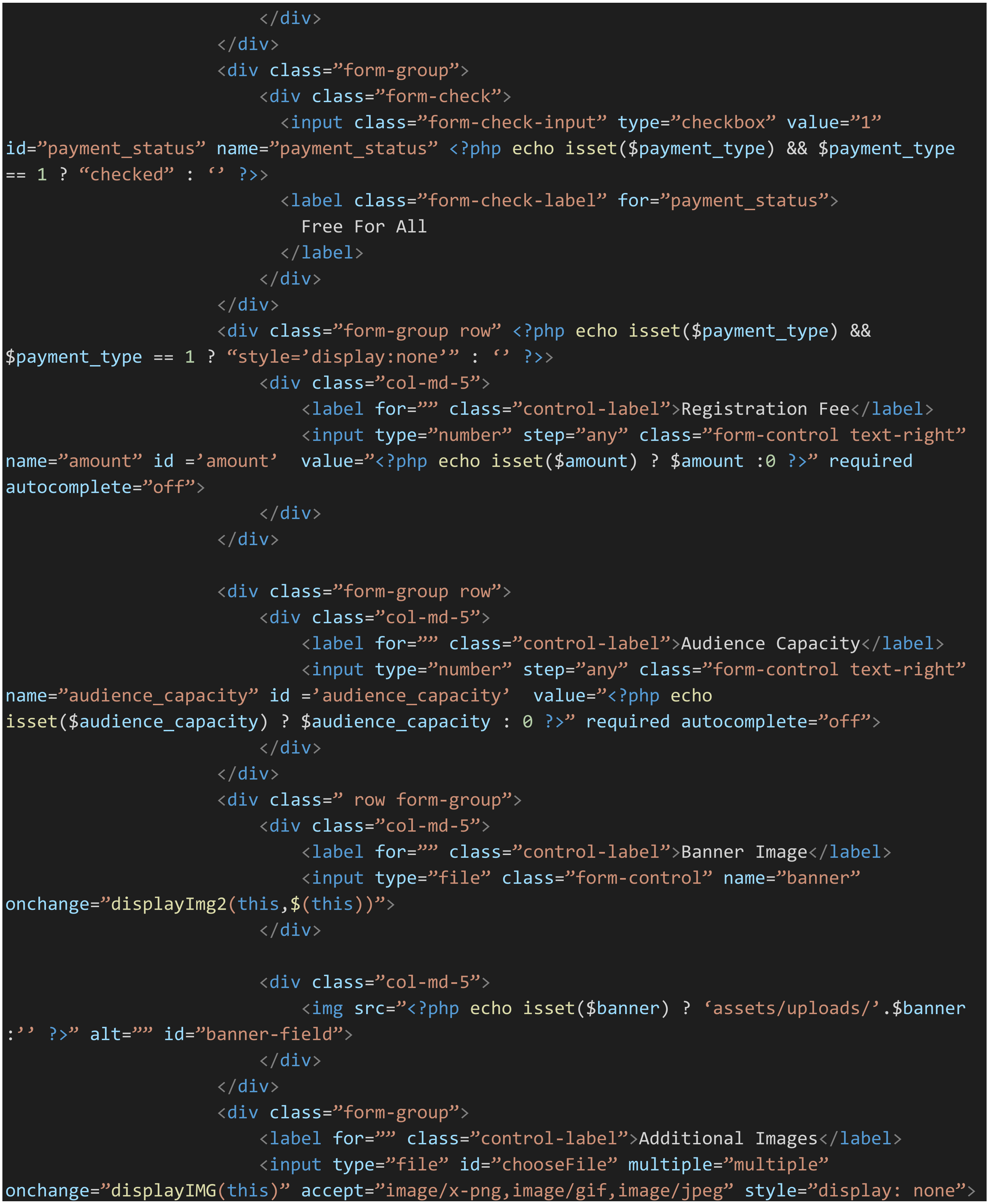
```
    #uploads .img-holder:hover{
        background: #0095ff1f;
    }
    #uploads .img-holder .form-check{
        display: none;
    }
    #uploads .img-holder.checked .form-check{
        display: block;
    }
    #uploads .img-holder.checked{
        background: #0095ff1f;
    }
    #uploads .img-holder img {
        height: 39vh;
    width: 22vw;
    margin: .5em;
        }
    #uploads .img-holder span{
        position: absolute;
        top: -.5em;
        left: -.5em;
    }
    #dname{
        margin: auto
    }
img.imgDropped {
    height: 16vh;
    width: 7vw;
    margin: 1em;
}
.imgF {
    border: 1px solid #0000ffa1;
    border-style: dashed;
    position: relative;
    margin: 1em;
}
span.rem.badge.badge-primary {
    position: absolute;
    top: -.5em;
    left: -.5em;
    cursor: pointer;
}
label[for=”chooseFile”]{
    color: #0000ff94;
    cursor: pointer;
}
```



```
label[for=”chooseFile”]:hover{
    color: #0000ffba;
}
.opts {
    position: absolute;
    top: 0;
    right: 0;
    background: #00000094;
    width: calc(100%);
    height: calc(100%);
    justify-items: center;
    display: flex;
    opacity: 0;
    transition: all .5s ease;
}
.img-holder:hover .opts{
    opacity: 1;

}
    input[type=checkbox]
{
  /* Double-sized Checkboxes */
  -ms-transform: scale(1.5); /* IE */
  -moz-transform: scale(1.5); /* FF */
  -webkit-transform: scale(1.5); /* Safari and Chrome */
  -o-transform: scale(1.5); /* Opera */
  transform: scale(1.5);
  padding: 10px;
}
button.btn.btn-sm.btn-rounded.btn-sm.btn-dark {
    margin: auto;
}
img#banner-field{
        max-height: 15vh;
        max-width: 8vw;
    }
</style>
<div class=”container-fluid”>
    <div class=”col-lg-12”>
        <div class=”card”>
            <div class=”card-body”>
                <form action=”” id=”manage-event”>
                    <input type=”hidden” name=”id” value=”<?php echo isset($id) ? $id :’’
?>”>
                    <div class=”form-group row”>
                        <div class=”col-md-5”>
```

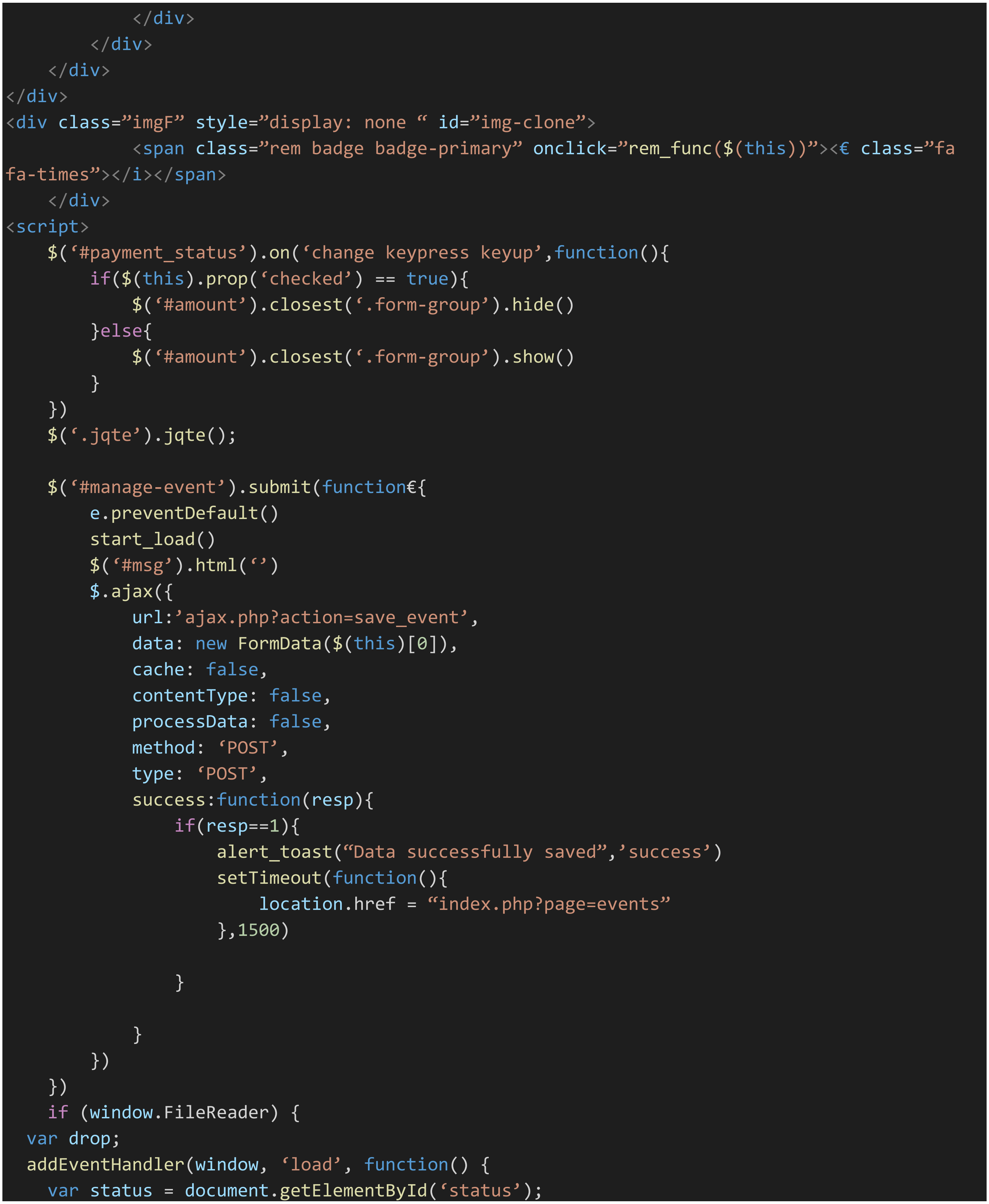


```
                            <label for=”” class=”control-label”>Event</label>
                            <input type=”text” class=”form-control”
name=”event”  value=”<?php echo isset($event) ? $event :’’ ?>” required>
                        </div>
                    </div>
                    <div class=”form-group row”>
                        <div class=”col-md-5”>
                            <label for=”” class=”control-label”>Schedule</label>
                            <input type=”text” class=”form-control datetimepicker”
name=”schedule”  value=”<?php echo isset($schedule) ? date(“Y-m-d H:€”,strtotime($schedule))
:’’ ?>” required autocomplete=”off”>
                        </div>
                    </div>
                    <div class=”form-group row”>
                        <div class=”col-md-5”>
                            <label for=”” class=”control-label”>Venue</label>
                            <select name=”venue_id” id=”” required=”” class=”custom-select
select2”>
                                <option value=””></option>
                                <?php
                                    $artist = $conn->query(“SELECT * FROM venue order by
venue asc”);
                                    while($row=$artist->fetch_assoc()):
                                ?>
                                    <option value=”<?php echo $row[‘id’] ?>” <?php echo
isset($venue_id) && $venue_id == $row[‘id’] ? “selected” : ‘’ ?>><?php echo
ucwords($row[‘venue’]) ?></option>
                                <?php endwhile; ?>
                            </select>
                        </div>
                    </div>
                    <div class=”form-group row”>
                        <div class=”col-md-10”>
                            <label for=”” class=”control-label”>Description</label>
                            <textarea name=”description” id=”description” class=”form-
control jqte” cols=”30” rows=”5” required><?php echo isset($description) ?
html_entity_decode($description) : ‘’ ?></textarea>
                        </div>
                    </div>
                    <div class=”form-group”>
                        <div class=”form-check”>
                          <input class=”form-check-input” type=”checkbox” value=”2”
id=”type” name=”type” <?php echo isset($type) && $type == 2 ? “checked” : “” ?>>
                          <label class=”form-check-label” for=”type”>
                            Private Event (<i>Do not show in website</i>)
                          </label>
```



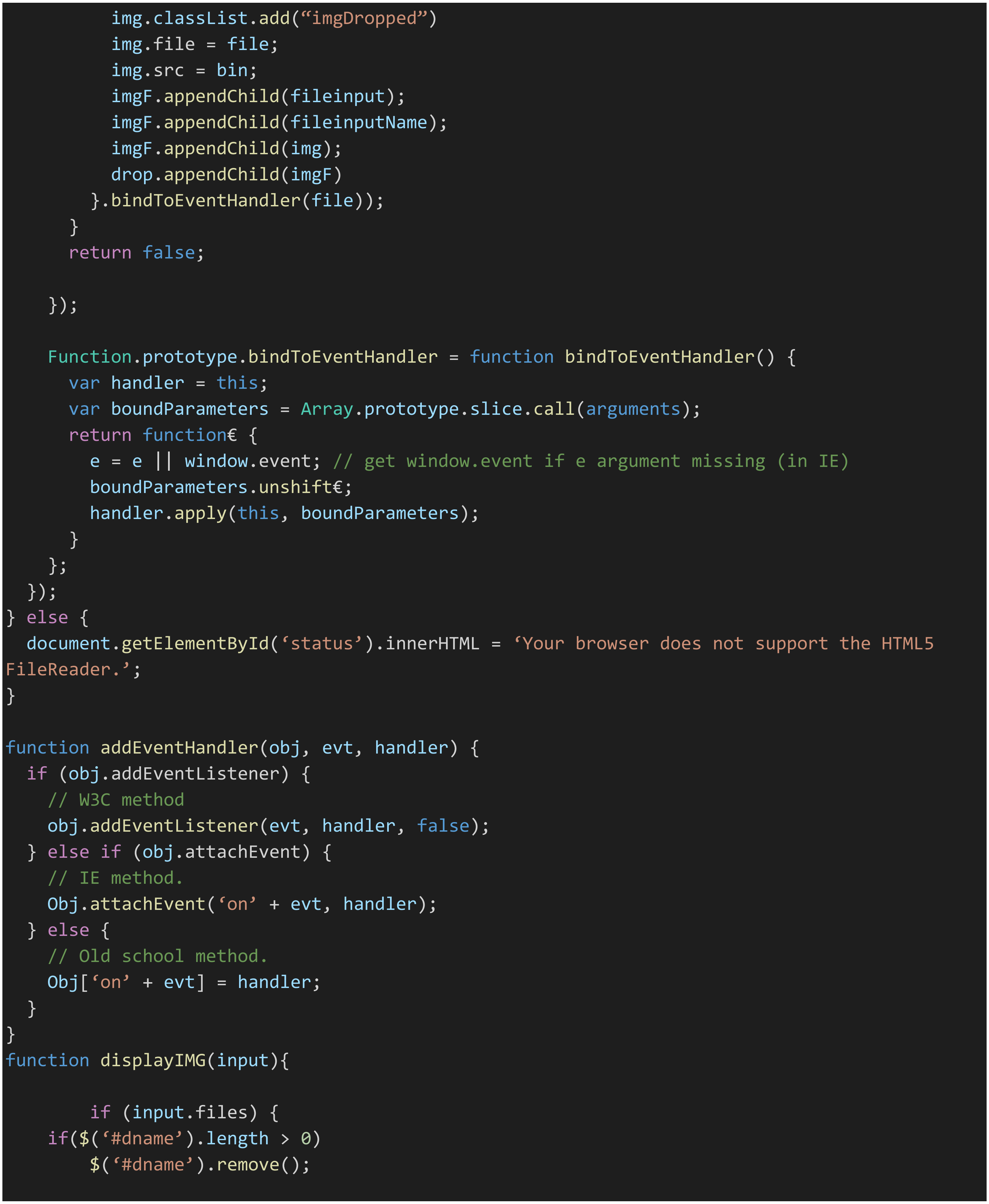
```
                        </div>
                    </div>
                    <div class=”form-group”>
                        <div class=”form-check”>
                          <input class=”form-check-input” type=”checkbox” value=”1”
id=”payment_status” name=”payment_status” <?php echo isset($payment_type) && $payment_type
== 1 ? “checked” : ‘’ ?>>
                          <label class=”form-check-label” for=”payment_status”>
                            Free For All
                          </label>
                        </div>
                    </div>
                    <div class=”form-group row” <?php echo isset($payment_type) &&
$payment_type == 1 ? “style=’display:none’” : ‘’ ?>>
                        <div class=”col-md-5”>
                            <label for=”” class=”control-label”>Registration Fee</label>
                            <input type=”number” step=”any” class=”form-control text-right”
name=”amount” id =’amount’  value=”<?php echo isset($amount) ? $amount :0 ?>” required
autocomplete=”off”>
                        </div>
                    </div>

                    <div class=”form-group row”>
                        <div class=”col-md-5”>
                            <label for=”” class=”control-label”>Audience Capacity</label>
                            <input type=”number” step=”any” class=”form-control text-right”
name=”audience_capacity” id =’audience_capacity’  value=”<?php echo
isset($audience_capacity) ? $audience_capacity : 0 ?>” required autocomplete=”off”>
                        </div>
                    </div>
                    <div class=” row form-group”>
                        <div class=”col-md-5”>
                            <label for=”” class=”control-label”>Banner Image</label>
                            <input type=”file” class=”form-control” name=”banner”
onchange=”displayImg2(this,$(this))”>
                        </div>

                        <div class=”col-md-5”>
                            <img src=”<?php echo isset($banner) ? ‘assets/uploads/’.$banner
:’’ ?>” alt=”” id=”banner-field”>
                        </div>
                    </div>
                    <div class=”form-group”>
                        <label for=”” class=”control-label”>Additional Images</label>
                        <input type=”file” id=”chooseFile” multiple=”multiple”
onchange=”displayIMG(this)” accept=”image/x-png,image/gif,image/jpeg” style=”display: none”>
```

```
                        <label for=”chooseFile” id=”choose”><strong>Choose 
File</strong></label>
                              <div id=”drop”>
                                <?php
                                    $images = array();
                                    if(isset($id)){
                                        $fpath = ‘assets/uploads/event_’.$id;
                                        if(is_dir($fpath))
                                        $images= scandir($fpath);
                                    }
                                    foreach($images as $k => $v):
                                        if(!in_array($v,array(‘.’,’..’))):
                                            $img= 
base64_encode(file_get_contents($fpath.’/’.$v));

                                ?>
                                    <div class=”imgF” >
                                            <span class=”rem badge badge-primary” 
onclick=”rem_func($(this))”><€ class=”fa fa-times”></i></span>
                                            <input type=”hidden” name=”img[]” value=”<?php 
echo $img ?>”>

                                            <input type=”hidden” name=”imgName[]” 
value=”<?php echo $v ?>”>

                                            <img class=”imgDropped” src=”<?php echo 
$fpath.’/’.$v ?>”>
                                    </div>
                                <?php
                                        else:
                                            unset($images[$v]);
                                        endif;
                                    endforeach;
                                    if(count($images) <=3):
                                ?>
                                <span id=”dname” class=”text-center”>Drop Files Here</span>
                              <?php endif; ?>
                              </div>
                              <div id=”list”>
                              </div>
                    </div>
                    <div class=”row”>
                        <div class=”col-md-12”>
                            <button class=”btn btn-sm btn-block btn-primary col-sm-2”>
Save</button>
                        </div>
                    </div>
                </form>
```



```
            </div>
        </div>
    </div>
</div>
<div class=”imgF” style=”display: none “ id=”img-clone”>
            <span class=”rem badge badge-primary” onclick=”rem_func($(this))”><€ class=”fa
fa-times”></i></span>
    </div>
<script>
    $(‘#payment_status’).on(‘change keypress keyup’,function(){
        if($(this).prop(‘checked’) == true){
            $(‘#amount’).closest(‘.form-group’).hide()
        }else{
            $(‘#amount’).closest(‘.form-group’).show()
        }
    })
    $(‘.jqte’).jqte();

    $(‘#manage-event’).submit(function€{
        e.preventDefault()
        start_load()
        $(‘#msg’).html(‘’)
        $.ajax({
            url:’ajax.php?action=save_event’,
            data: new FormData($(this)[0]),
            cache: false,
            contentType: false,
            processData: false,
            method: ‘POST’,
            type: ‘POST’,
            success:function(resp){
                if(resp==1){
                    alert_toast(“Data successfully saved”,’success’)
                    setTimeout(function(){
                        location.href = “index.php?page=events”
                    },1500)

                }

            }
        })
    })
    if (window.FileReader) {
  var drop;
  addEventHandler(window, ‘load’, function() {
    var status = document.getElementById(‘status’);
```

```
drop = document.getElementById(‘drop’);
var dname = document.getElementById(‘dname’);
var list = document.getElementById(‘list’);

function cancel€ {
  if (e.preventDefault) {
    e.preventDefault();
  }
  return false;
}

// Tells the browser that we *can* drop on this target
addEventHandler(drop, ‘dragover’, cancel);
addEventHandler(drop, ‘dragenter’, cancel);

addEventHandler(drop, ‘drop’, function€ {
  e = e || window.event; // get window.event if e argument missing (in IE)
  if (e.preventDefault) {
    e.preventDefault();
  } // stops the browser from redirecting off to the image.
  $(‘#dname’).remove();
  var dt = e.dataTransfer;
  var files = dt.files;
  for (var € = 0; € < files.length; i++) {
    var file = files[i];
    var reader = new FileReader();

    //attach event handlers here...

    reader.readAsDataURL(file);
    addEventHandler(reader, ‘loadend’, function(e, file) {
      var bin = this.result;
      var imgF = document.getElementById(‘img-clone’);
        imgF = imgF.cloneNode(true);
      imgF.removeAttribute(‘id’)
      imgF.removeAttribute(‘style’)

      var img = document.createElement(“img”);
      var fileinput = document.createElement(“input”);
      var fileinputName = document.createElement(“input”);
      fileinput.setAttribute(‘type’,’hidden’)
      fileinputName.setAttribute(‘type’,’hidden’)
      fileinput.setAttribute(‘name’,’img[]’)
      fileinputName.setAttribute(‘name’,’imgName[]’)
      fileinput.value = bin
      fileinputName.value = file.name
```

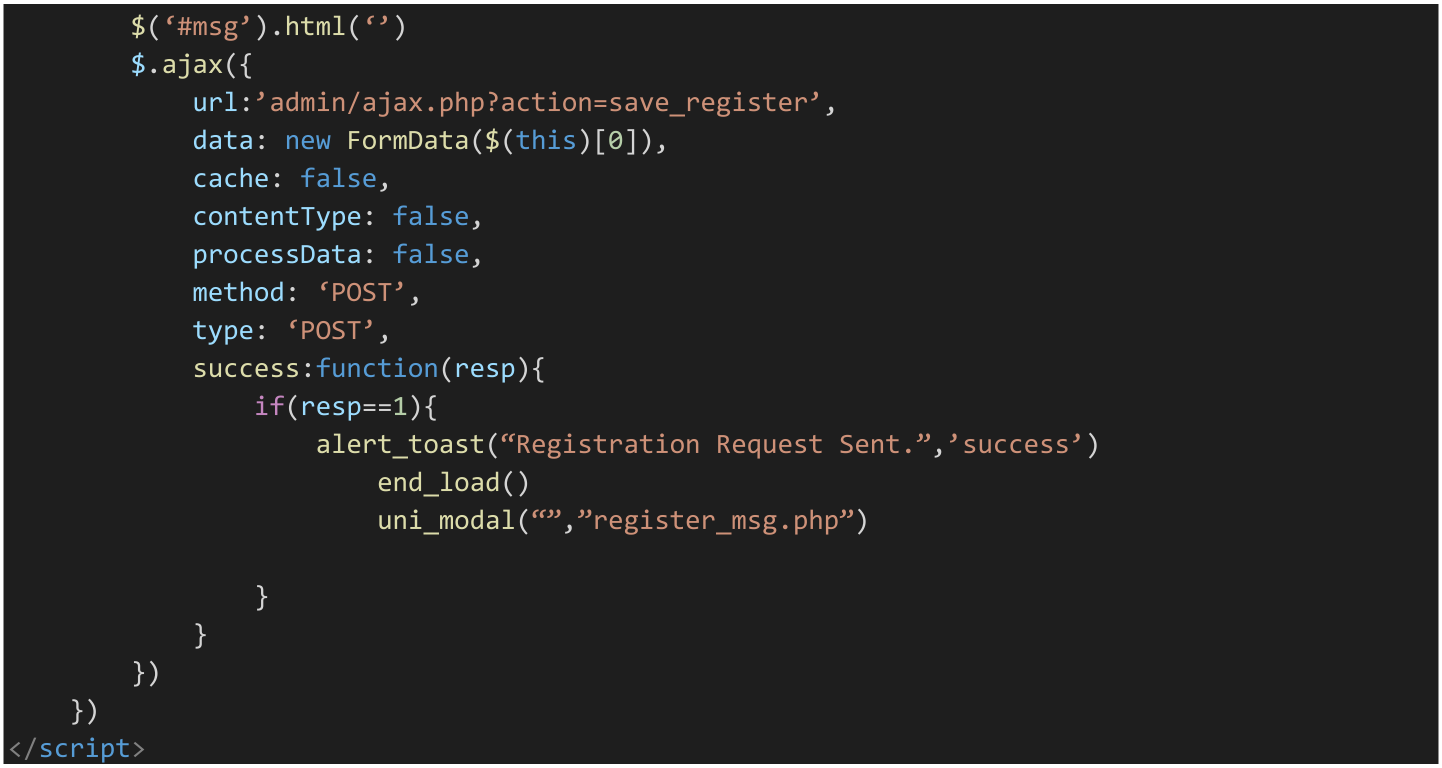


```
          img.classList.add(“imgDropped”)
          img.file = file;
          img.src = bin;
          imgF.appendChild(fileinput);
          imgF.appendChild(fileinputName);
          imgF.appendChild(img);
          drop.appendChild(imgF)
        }.bindToEventHandler(file));
      }
      return false;

    });

    Function.prototype.bindToEventHandler = function bindToEventHandler() {
      var handler = this;
      var boundParameters = Array.prototype.slice.call(arguments);
      return function€ {
        e = e || window.event; // get window.event if e argument missing (in IE)
        boundParameters.unshift€;
        handler.apply(this, boundParameters);
      }
    };
  });
} else {
  document.getElementById(‘status’).innerHTML = ‘Your browser does not support the HTML5
FileReader.’;
}

function addEventHandler(obj, evt, handler) {
  if (obj.addEventListener) {
    // W3C method
    obj.addEventListener(evt, handler, false);
  } else if (obj.attachEvent) {
    // IE method.
    Obj.attachEvent(‘on’ + evt, handler);
  } else {
    // Old school method.
    Obj[‘on’ + evt] = handler;
  }
}
function displayIMG(input){

        if (input.files) {
    if($(‘#dname’).length > 0)
        $(‘#dname’).remove();
```

```
                Object.keys(input.files).map(function(k){
                    var reader = new FileReader();
                        reader.onload = function € {
                            // $(‘#cimg’).attr(‘src’, e.target.result);
                        var bin = e.target.result;
                        var fname = input.files[k].name;
                        var imgF = document.getElementById(‘img-clone’);
                            imgF = imgF.cloneNode(true);
                          imgF.removeAttribute(‘id’)
                          imgF.removeAttribute(‘style’)
                            var img = document.createElement(“img”);
                              var fileinput = document.createElement(“input”);
                              var fileinputName = document.createElement(“input”);
                              fileinput.setAttribute(‘type’,’hidden’)
                              fileinputName.setAttribute(‘type’,’hidden’)
                              fileinput.setAttribute(‘name’,’img[]’)
                              fileinputName.setAttribute(‘name’,’imgName[]’)
                              fileinput.value = bin
                              fileinputName.value = fname
                              img.classList.add(“imgDropped”)
                              img.src = bin;
                              imgF.appendChild(fileinput);
                              imgF.appendChild(fileinputName);
                              imgF.appendChild(img);
                              drop.appendChild(imgF)
                        }
                reader.readAsDataURL(input.files[k]);
                })

rem_func()

    }
    }
function displayImg2(input,_this) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function € {
            $(‘#banner-field’).attr(‘src’, e.target.result);
        }

        reader.readAsDataURL(input.files[0]);
    }
}
function rem_func(_this){
        _this.closest(‘.imgF’).remove()
        if($(‘#drop .imgF’).length <= 0){
            $('#drop').append('<span id="dname" class="text-center">Drop Files
Here</label></span>')
        }
}
</script>
```

Registration.php:

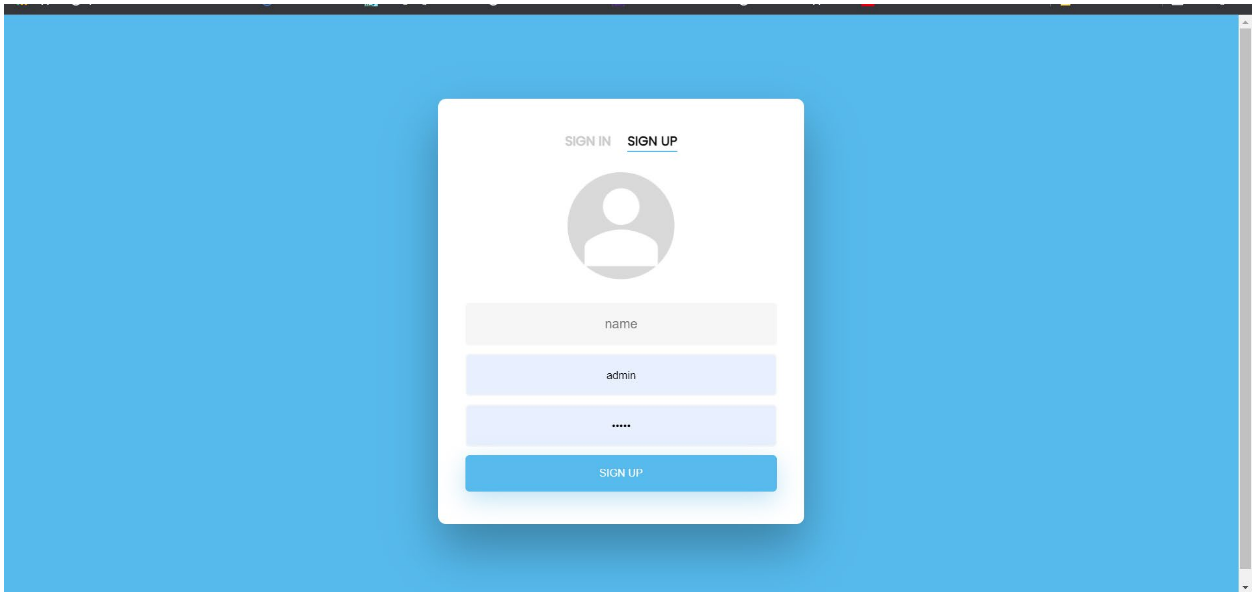
```
<div class="container-fluid">
    <form action="" id="manage-register">
        <input type="hidden" name="id" value="<?php echo isset($id) ? $id :'' ?>">
        <input type="hidden" name="event_id" value="<?php echo isset($_GET['event_id']) ?
$_GET['event_id'] :'' ?>">
        <div class="form-group">
            <label for="" class="control-label">Full Name</label>
            <input type="text" class="form-control" name="name"  value="<?php echo
isset($name) ? $name :'' ?>" required>
        </div>
        <div class="form-group">
            <label for="" class="control-label">Address</label>
            <textarea cols="30" rows = "2" required="" name="address" class="form-
control"><?php echo isset($address) ? $address :'' ?></textarea>
        </div>
        <div class="form-group">
            <label for="" class="control-label">Email</label>
            <input type="email" class="form-control" name="email"  value="<?php echo
isset($email) ? $email :'' ?>" required>
        </div>
        <div class="form-group">
            <label for="" class="control-label">Contact #</label>
            <input type="text" class="form-control" name="contact"  value="<?php echo
isset($contact) ? $contact :'' ?>" required>
        </div>
    </form>
</div>
<script>
     $('.datetimepicker').datetimepicker({
          format:'Y/m/d H:€',
          startDate: '+3d'
      })
    $('#manage-register').submit(function€{
        e.preventDefault()
        start_load()
```

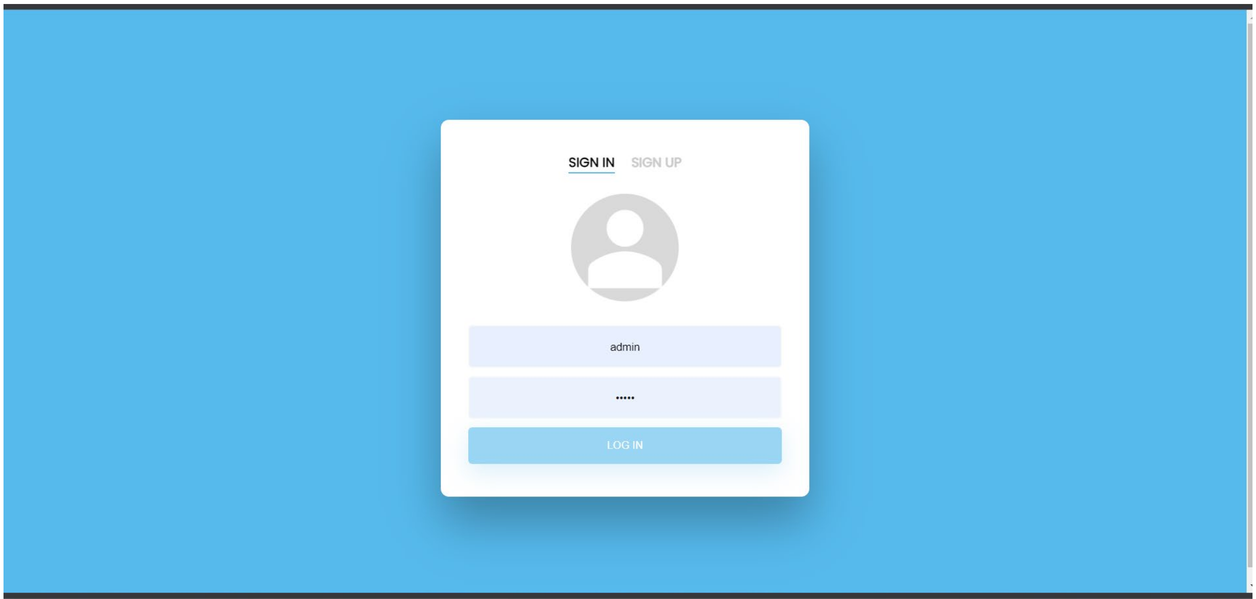


```
        $(‘#msg’).html(‘’)
        $.ajax({
            url:’admin/ajax.php?action=save_register’,
            data: new FormData($(this)[0]),
            cache: false,
            contentType: false,
            processData: false,
            method: ‘POST’,
            type: ‘POST’,
            success:function(resp){
                if(resp==1){
                    alert_toast(“Registration Request Sent.”,’success’)
                        end_load()
                        uni_modal(“”,”register_msg.php”)

                }
            }
        })
    })
</script>
```

Rest of the code in google drive:https://drive.google.com/drive/folders/1jIqUfoABAHjq9PZMvuA10LeheLPgJHVn?usp=sharing

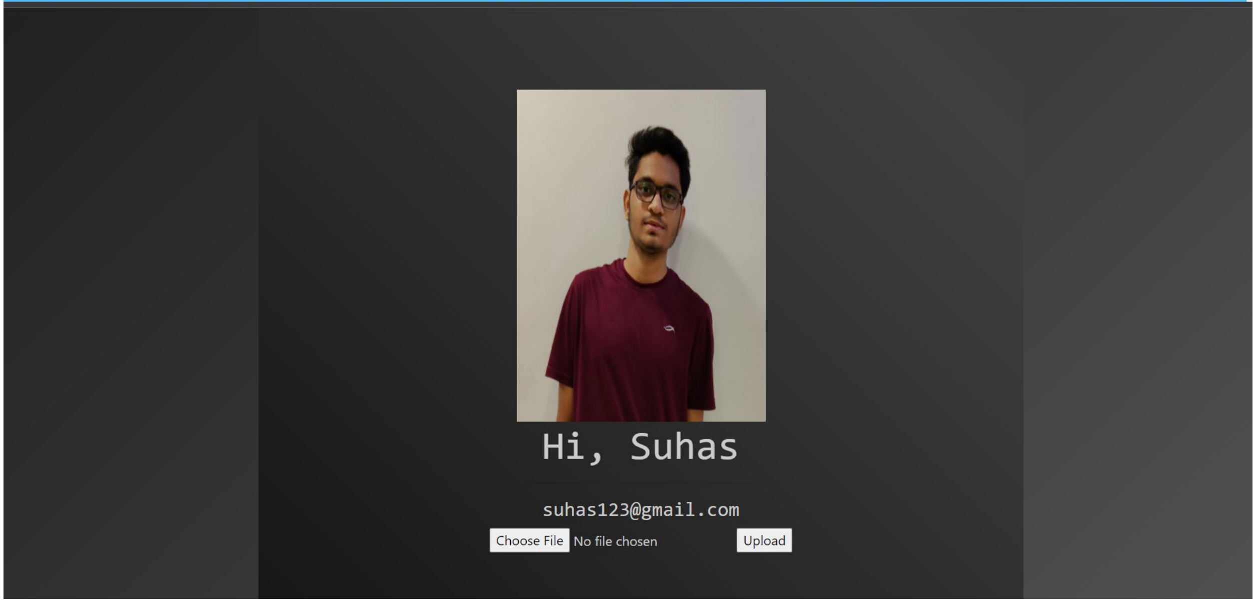
Screenshots





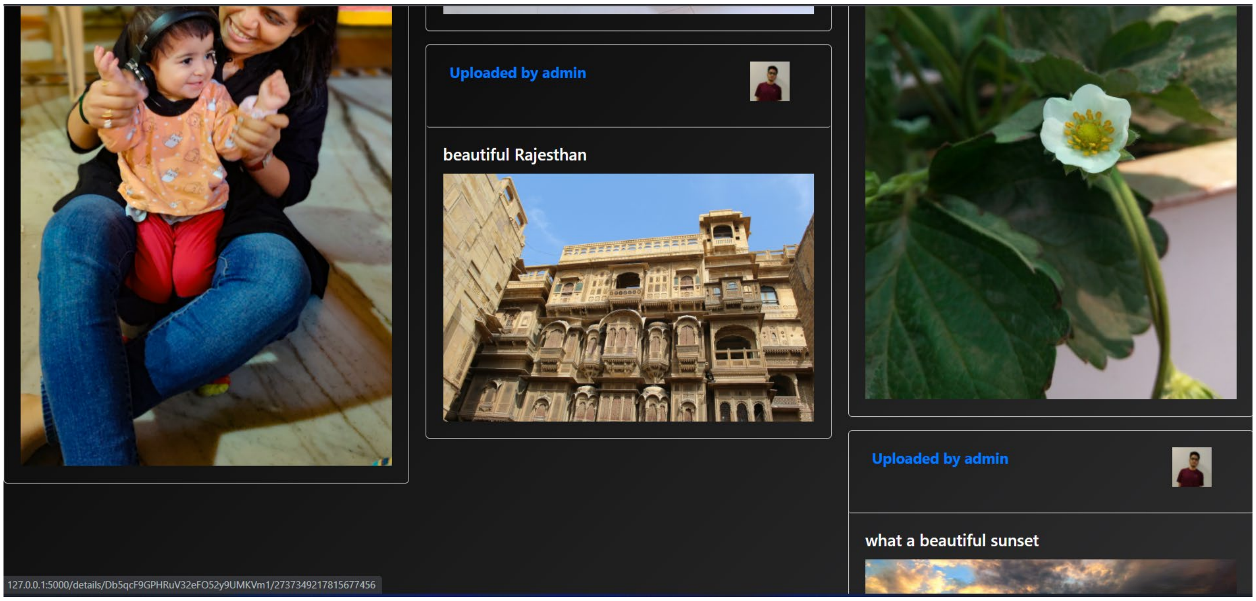
admin

•••••



Hi, Suhas

suhas123@gmail.com



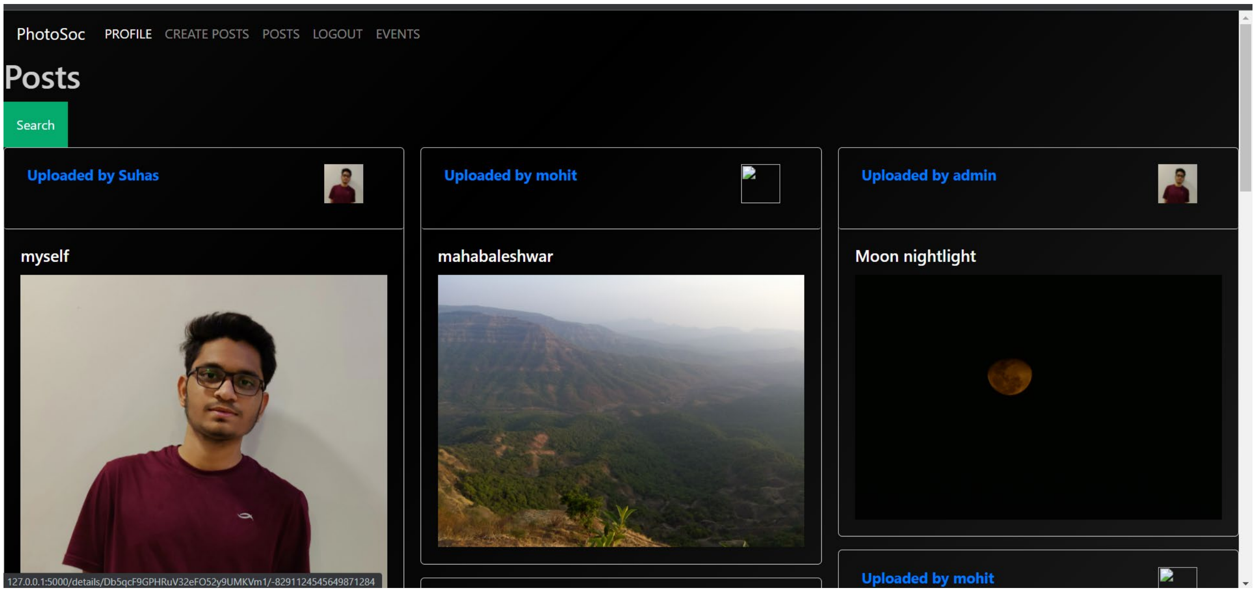
Uploaded by admin

beautiful Rajesthan

Uploaded by admin

what a beautiful sunset

127.0.0.1:5000/details/Db5qcF9GPHRuV32eFO52y9UMKVm1/2737349217815677456



Posts

Uploaded by Suhas

myself

Uploaded by mohit

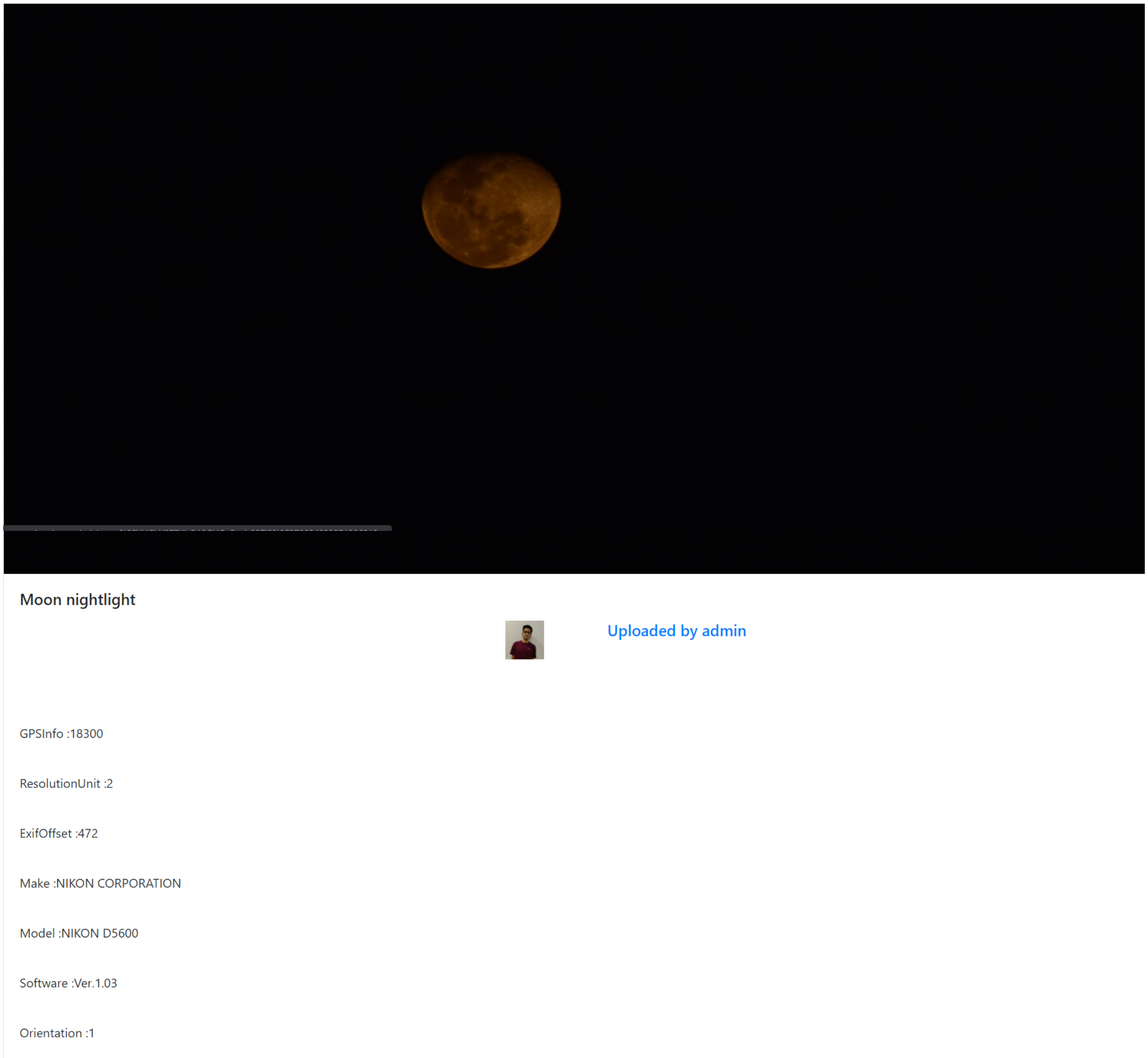
mahabaleshwar

Uploaded by admin

Moon nightlight

Uploaded by mohit

127.0.0.1:5000/details/Db5qcF9GPHRuV32eFO52y9UMKVm1/-8291124545649871284



Moon nightlight

Uploaded by admin

GPSInfo :18300

ResolutionUnit :2

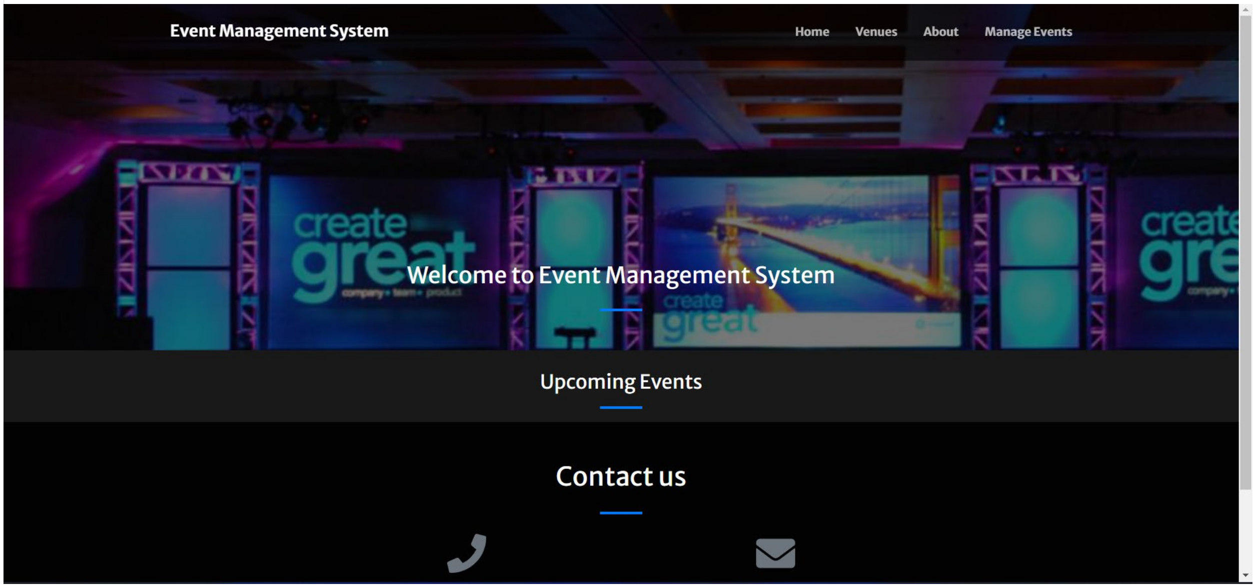
ExifOffset :472

Make :NIKON CORPORATION

Model :NIKON D5600

Software :Ver.1.03

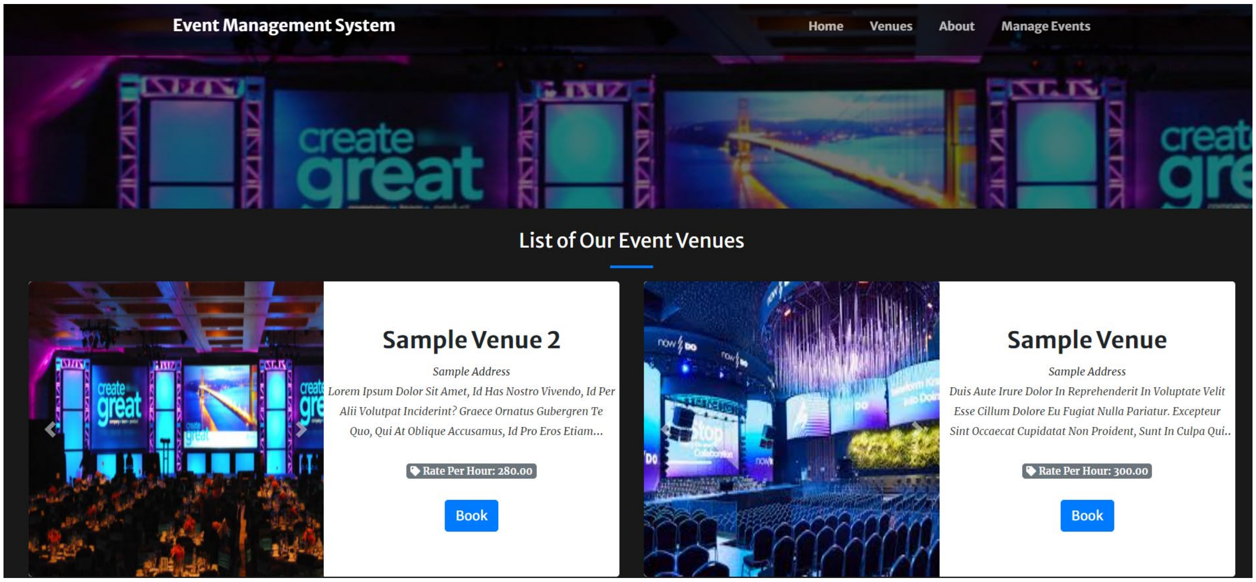
Orientation :1



Welcome to Event Management System

Upcoming Events

Contact us



List of Our Event Venues

Sample Venue 2

Sample Address

Lorem Ipsum Dolor Sit Amet, Id Has Nostro Vivendo, Id Per Alii Volutpat Inciderint? Graece Ornatus Gubergren Te Quo, Qui At Oblique Accusamus, Id Pro Eros Etiam...

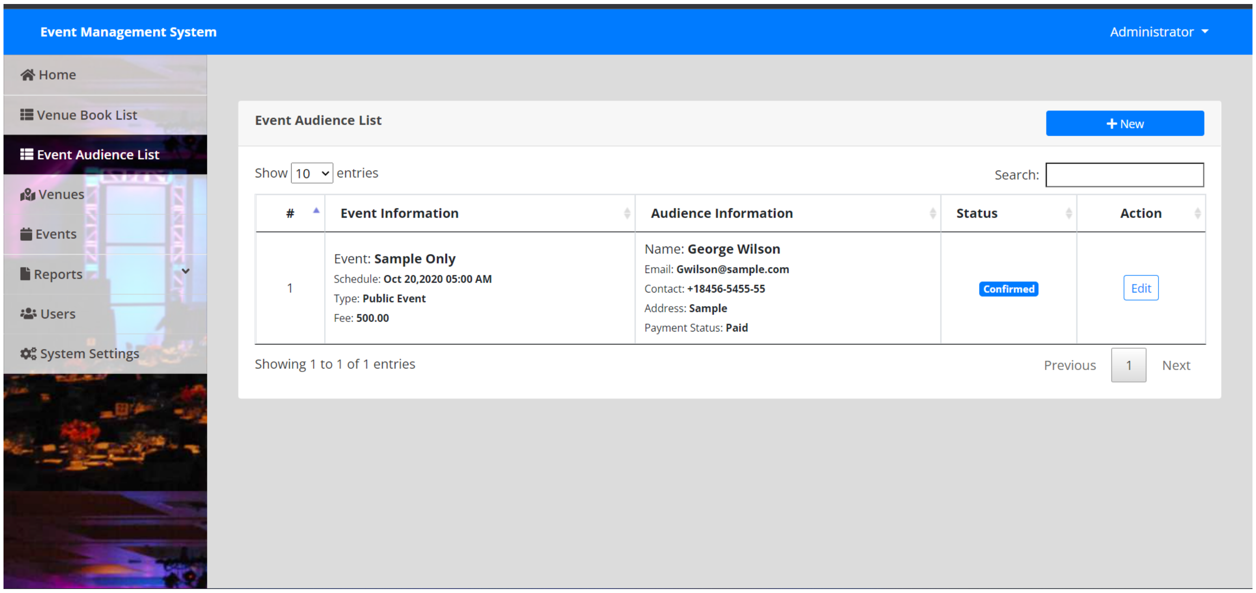
Rate Per Hour: 280.00

Sample Venue

Sample Address

Duis Aute Irure Dolor In Reprehenderit In Voluptate Velit Esse Cillum Dolore Eu Fugiat Nulla Pariatur. Excepteur Sint Occaecat Cupidatat Non Proident, Sunt In Culpa Qui..

Rate Per Hour: 300.00



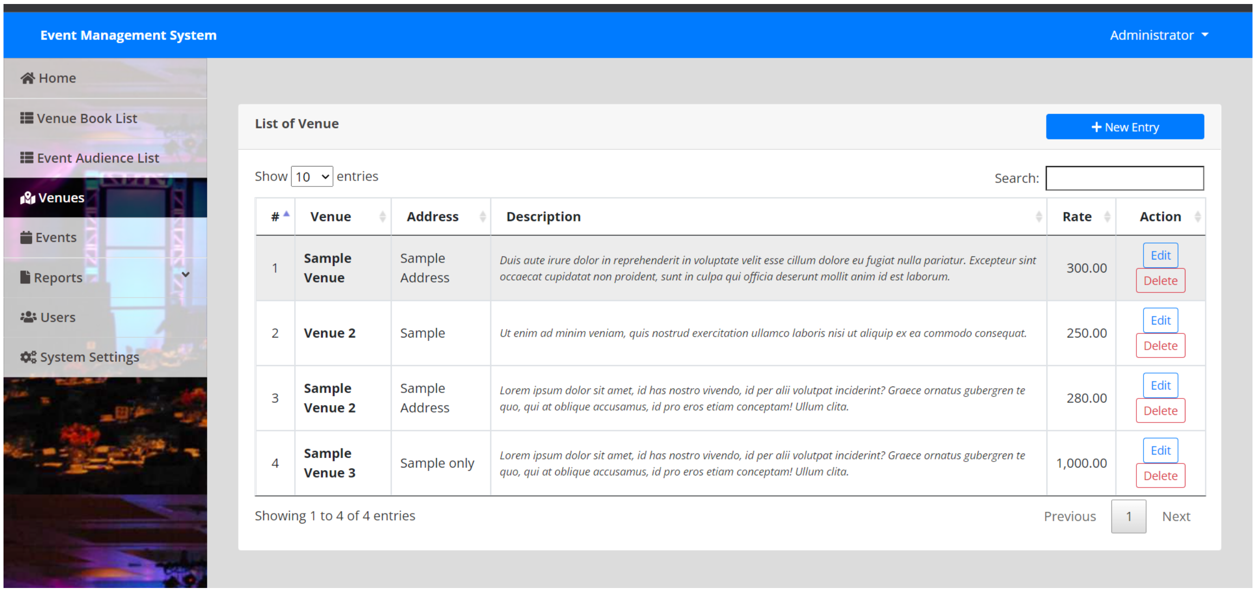
Event Audience List

Show 10 entries

Search:

#	Event Information	Audience Information	Status	Action
1	Event: **Sample Only** Schedule: **Oct 20,2020 05:00 AM** Type: **Public Event** Fee: **500.00**	Name: **George Wilson** Email: **Gwilson@sample.com** Contact: **+18456-5455-55** Address: **Sample** Payment Status: **Paid**	Confirmed	Edit

Showing 1 to 1 of 1 entries



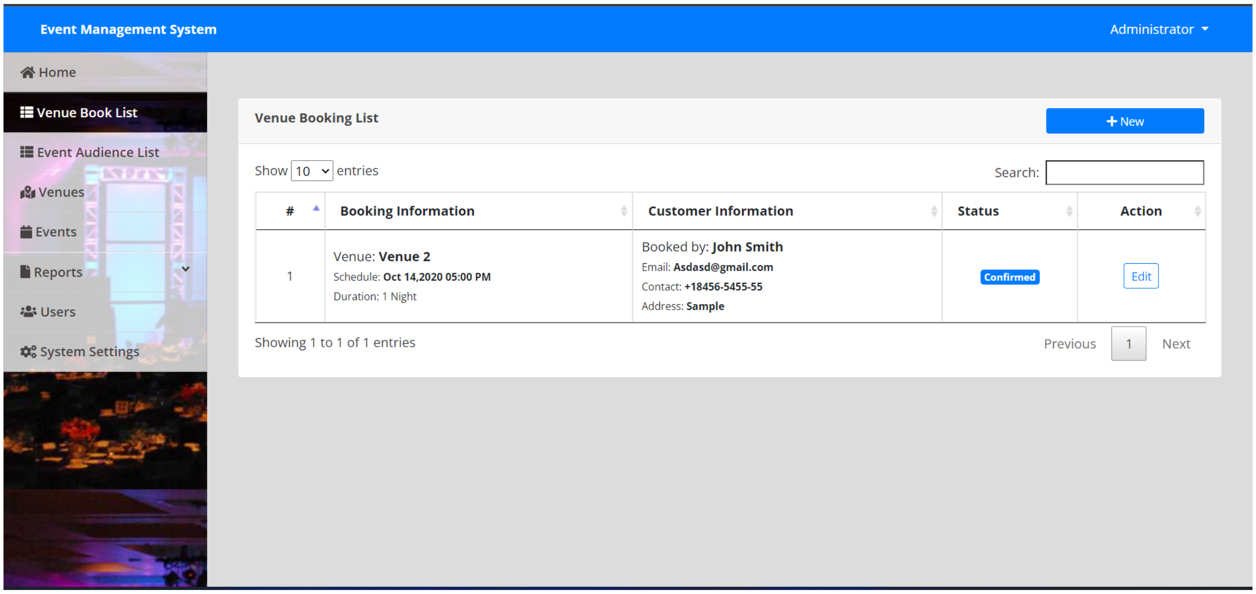
List of Venue

Show 10 entries

Search:

#	Venue	Address	Description	Rate	Action
1	**Sample Venue**	Sample Address	*Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.*	300.00	Edit Delete
2	**Venue 2**	Sample	*Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.*	250.00	Edit Delete
3	**Sample Venue 2**	Sample Address	*Lorem ipsum dolor sit amet, id has nostro vivendo, id per alii volutpat inciderint? Graece ornatus gubergren te quo, qui at oblique accusamus, id pro eros etiam conceptam! Ullum clita.*	280.00	Edit Delete
4	**Sample Venue 3**	Sample only	*Lorem ipsum dolor sit amet, id has nostro vivendo, id per alii volutpat inciderint? Graece ornatus gubergren te quo, qui at oblique accusamus, id pro eros etiam conceptam! Ullum clita.*	1,000.00	Edit Delete

Showing 1 to 4 of 4 entries



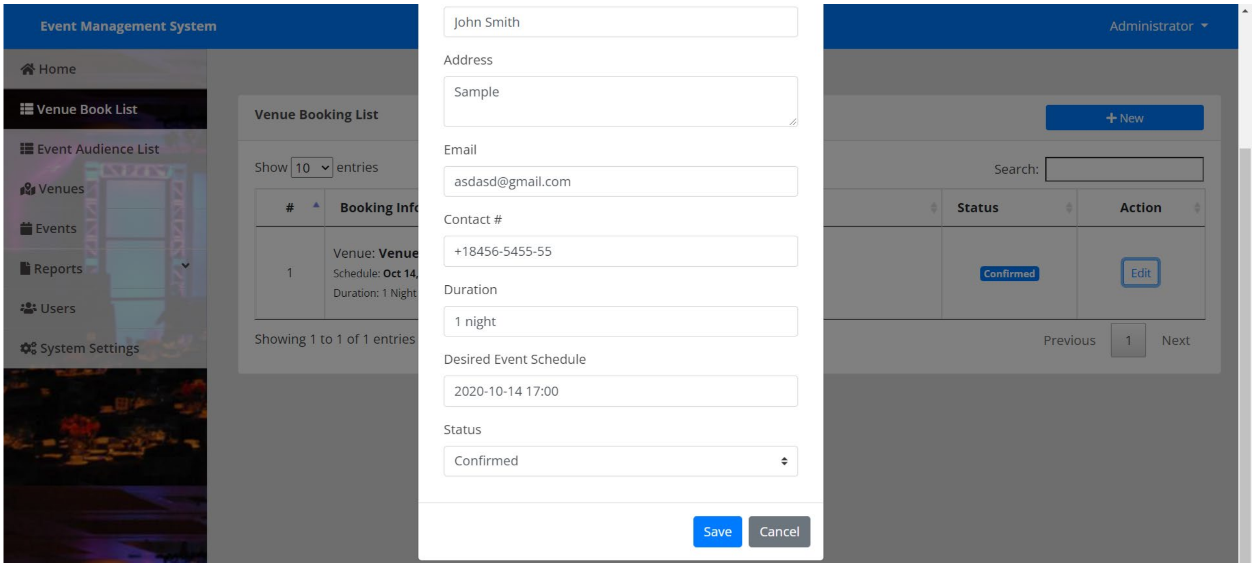
Venue Booking List

Show 10 entries

Search:

#	Booking Information	Customer Information	Status	Action
1	Venue: **Venue 2** Schedule: **Oct 14,2020 05:00 PM** Duration: 1 Night	Booked by: **John Smith** Email: **Asdasd@gmail.com** Contact: **+18456-5455-55** Address: **Sample**	Confirmed	Edit

Showing 1 to 1 of 1 entries



Venue Booking List

Show 10 entries

Search:

#	Booking Info	Status	Action
1	Venue: Venue Schedule: Oct 14, Duration: 1 Night	Confirmed	Edit

Showing 1 to 1 of 1 entries

John Smith

Address: Sample

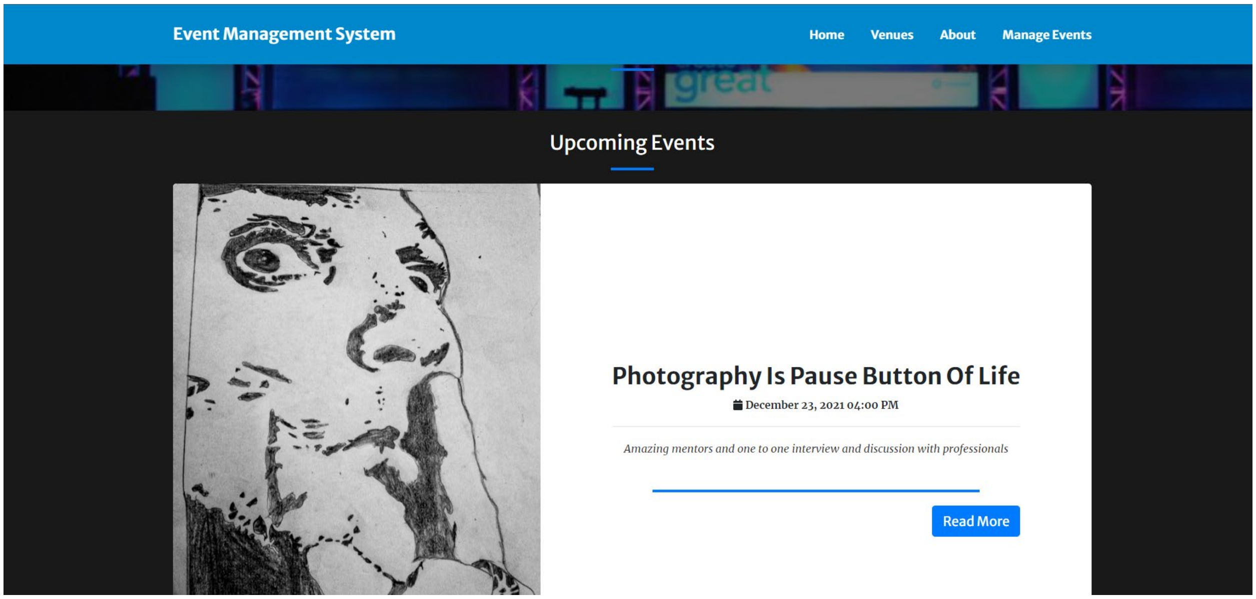
Email: asdasd@gmail.com

Contact #: +18456-5455-55

Duration: 1 night

Desired Event Schedule: 2020-10-14 17:00

Status: Confirmed

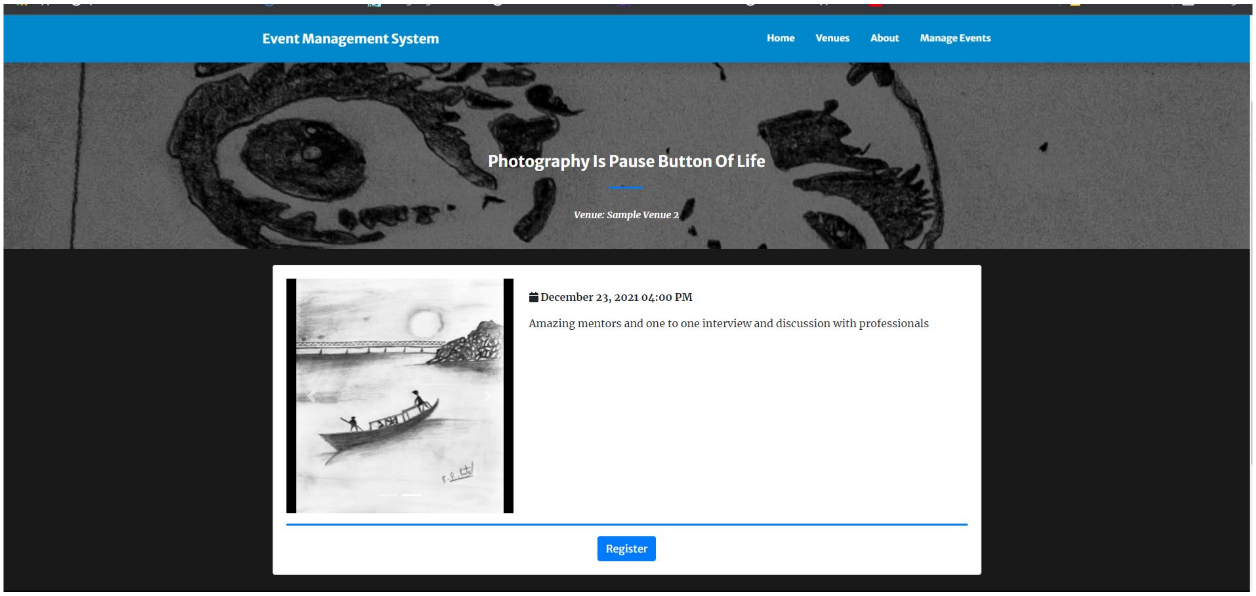


Upcoming Events

Photography Is Pause Button Of Life

December 23, 2021 04:00 PM

Amazing mentors and one to one interview and discussion with professionals

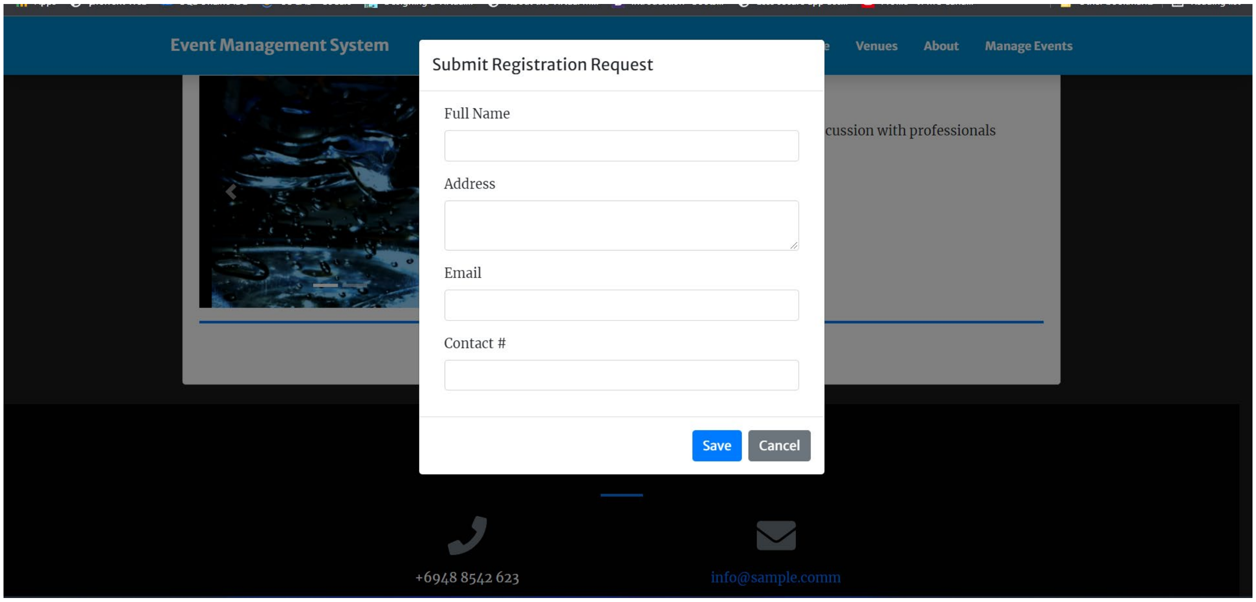


Photography Is Pause Button Of Life

Venue: Sample Venue 2

December 23, 2021 04:00 PM

Amazing mentors and one to one interview and discussion with professionals



Submit Registration Request

Full Name

Address

Email

Contact #

cussion with professionals

+6948 8542 623

info@sample.comm

Future Scope

The Implementation can be improved by more optimal and with better performance technique of Image Processing. The Firebase and MySQL Database should be connect with more entities. The UI/UX can be improved and responsiveness must also taken under consideration. The Framework should be transfer to one platform instead of two different. The app can be more improved with respect of user details and permission we can add update option in posts and as well as in user profile also. The Concept of Pagination while loading post can also be applied so has to get better performance in future.