

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY GURAJADA-VIZIANAGARAM  
VIZIANAGARAM – 533003  
ANDHRA PRADESH, INDIA  
MARCH 2025.



# UNIVERSITY ATTENDANCE MANAGEMENT SYSTEM

by  
**BATCH-1**

|                             |                   |
|-----------------------------|-------------------|
| ALLADA SAI SATYA SHARAN     | 23VV1A1201        |
| BANDARU DURGA PRASAD        | 23VV1A1202        |
| BANDARU VENKATA RAMALAKSHMI | 23VV1A1203        |
| <b>BEHARA SAHITHYA</b>      | <b>23VV1A1205</b> |

Under the supervision of

**MRS.MADHUMITA CHANDA**

**Department of Information Technology  
JNTU-GV CE, Vizianagaram**



**JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Regd.No : 23VV1A1205**

**CERTIFICATE**

**This is to certify that this is a bonafide record of practical work done by MS. B. Sahithya of II B.Tech II Semester Class in Django framework Lab during the year 2024-25.**

**No.of Tasks Completed and Certified: 13**

**Lecture In-Charge**

**Head of The Department**

**Date:**



DEPARTMENT OF INFORMATION TECHNOLOGY

JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM

Website: www.jntugvcev.edu.in

Subject Name: DJANGO FRAMEWORK

Subject Code: R232212SE01

Year: 2025

Regulation: R23

### COURSE OUTCOMES

| NBA<br>Subject Code | Course Outcomes |   |
|---------------------|-----------------|---|
|                     | <b>CO1</b>      | Design and build static as well as dynamic web pages and interactive web-based applications . |
|                     | <b>CO2</b>      | Web development using Django framework.   |
|                     | <b>CO3</b>      | Analyze and create functional website in Django and deploy Django Web Application on Cloud .  |

### CO-PO Mapping

Mapping of Course Outcomes (COs) with Program Outcomes (POs)

| Course Outcomes |     | Program Outcomes (POs) |      |      |      |      |      |      |      |      |       |       |       |       |       |       |
|-----------------|-----|------------------------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|
|                 |     | PO 1                   | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 | PSO 3 |
|                 | CO1 | 3                      | 1    | 3    | 1    | 3    | 1    | 1    | 1    | 2    | 3     | 2     | 1     | 3     | 3     | 2     |
|                 | CO2 | 3                      | 2    | 3    | 1    | 3    | 1    | 1    | 1    | 2    | 2     | 2     | 2     | 3     | 3     | 3     |
|                 | CO3 | 2                      | 3    | 3    | 3    | 3    | 2    | 2    | 2    | 2    | 3     | 3     | 3     | 3     | 3     | 3     |

Enter correlation levels 1,2 and 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) If there is no correlation, put “-”

Signature of the Course Instructor

## **INDEX**

| <b>SNO</b> | <b>DATE</b> | <b>CONTENT</b>                                   | <b>PAGE</b> | <b>MARKS</b> | <b>REMARKS</b> |
|------------|-------------|--|-------------|--------------|----------------|
| <b>1</b>   | 13-12-2024  | Libraries  | 5-11        |              |                |
| <b>2</b>   | 20-12-2024  | Django Installation Process                      | 12-14       |              |                |
| <b>3</b>   | 27-12-2024  | Django App Creation                              | 15-16       |              |                |
| <b>4</b>   | 03-01-2025  | Connecting Views an app                          | 17-18       |              |                |
| <b>5</b>   | 24-01-2025  | Exploring Views.py                               | 19-23       |              |                |
| <b>6</b>   | 24-01-2025  | App URL.py                                       | 24-25       |              |                |
| <b>7</b>   | 31-01-2025  | Templates  | 26-58       |              |                |
| <b>8</b>   | 17-02-2025  | What is forms.py in Django                       | 59-60       |              |                |
| <b>9</b>   | 21-02-2025  | What is models.py and how to apply models        | 61-63       |              |                |
| <b>10</b>  | 21-02-2025  | Run migrations to create a database table        | 64-65       |              |                |
| <b>11</b>  | 07-03-2025  | Database Integration                             | 66-68       |              |                |
| <b>12</b>  | 27-03-2025  | Deploying Django Applications on cloud platforms | 69-70       |              |                |
| <b>13</b>  | 04-04-2025  | Front end web developer certifications           | 71-73       |              |                |



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Libraries
7. Date of Experiment : 13-12-2024
- 8.Date of Submission of Report :20-12-2024

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

## **DJANGO FRAMEWORK**

### **LIBRARIES:**

#### **Tkinter-GUI Applications:**

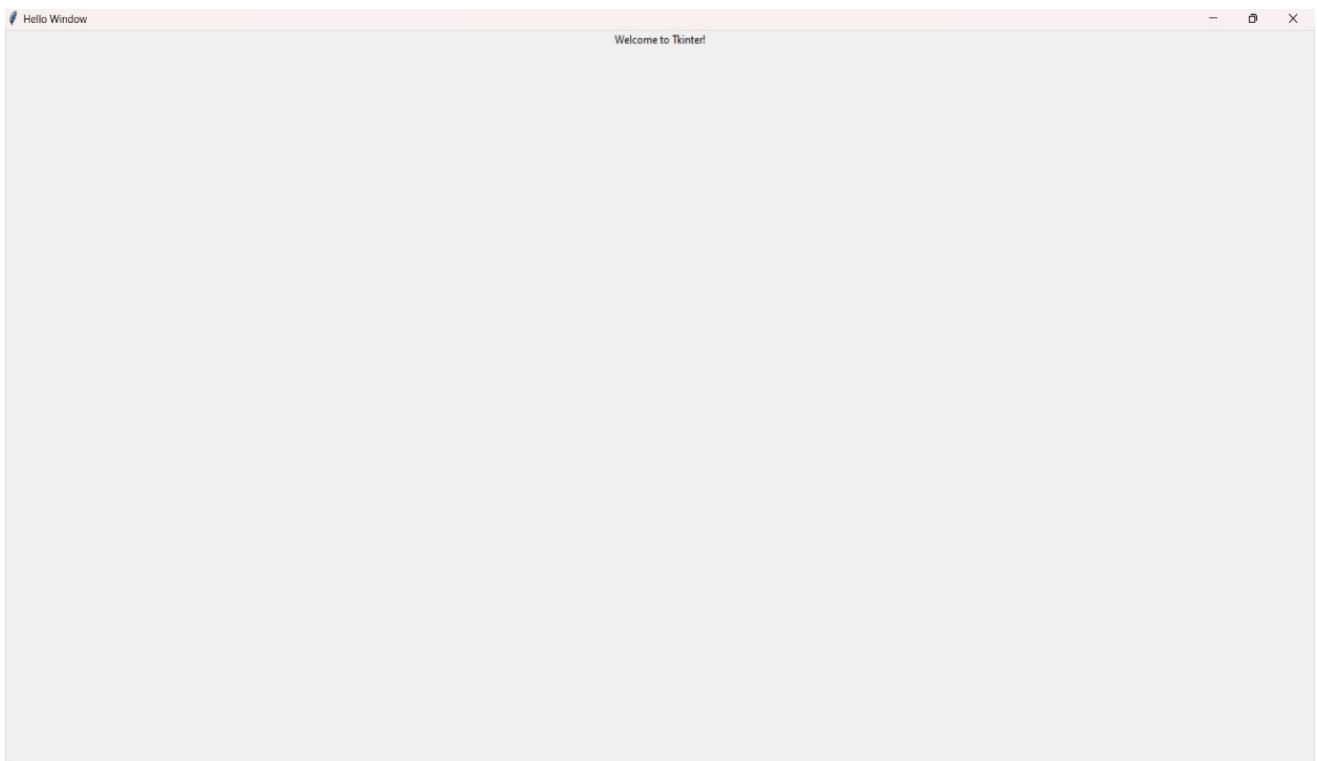
```
#pip install tk
from tkinter import Tk, Label

# Create a window root =
Tk()
root.title("Hello Window")

# Add a label to display text
Label(root, text="Welcome to Tkinter!").pack()

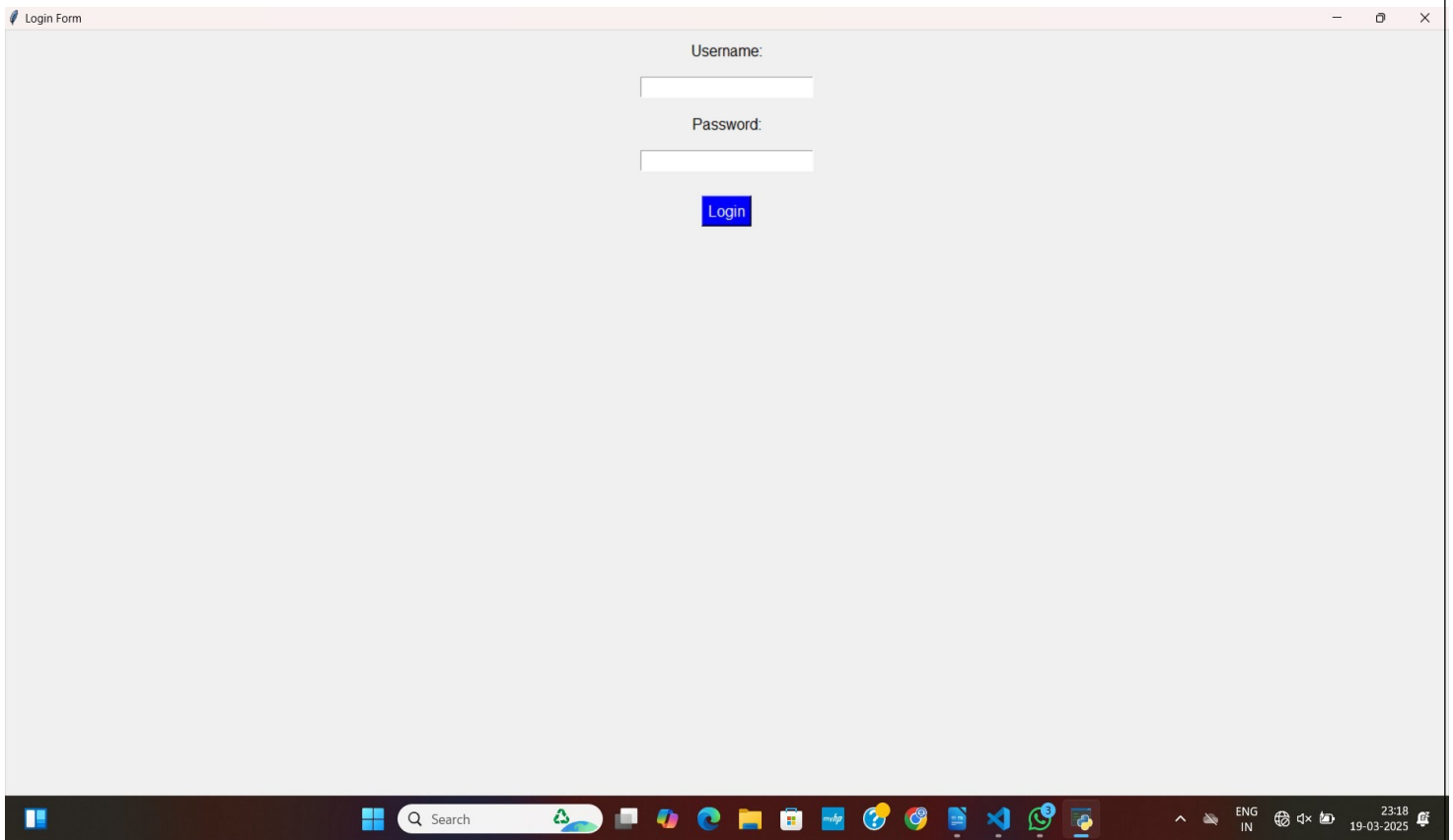
# Run the application
root.mainloop()
```

### **Output:**



## **Requests-HTTP Requests:**

```
# Install tkinter if not already installed: pip install tk import
tkinter as tk from tkinter import messagebox # Function
to validate login def validate_login(): username =
username_entry.get() password = password_entry.get()
# Example credentials if username == "user" and
password == "password": messagebox.showinfo("Login
Success", "Login Successful!") else:
messagebox.showerror("Login Failed", "Invalid
username or password") # Create the main window root
= tk.Tk\(\) root.title("Login Form")
root.geometry("400x300") # Adjusted size for better
presentation # Username label and entry
username_label = tk.Label(root, text="Username:",
font=("Arial", 12)) username_label.pack(pady=10)\
username_entry = tk.Entry(root, font=("Arial", 12))
username_entry.pack(pady=5) # Password label and
entry password_label = tk.Label(root, text="Password:",
font=("Arial", 12)) password_label.pack(pady=10)
password_entry = tk.Entry(root, show="*",
font=("Arial", 12)) password_entry.pack(pady=5) # Login
button login_button = tk.Button(root, text="Logi
command=validate_login, font=("Arial", 12), bg="blue",
fg="white") login_button.pack(pady=20) # Run the
Tkinter event loop root.mainloop()
```

**Output:**

The screenshot displays a web browser window with the title "Login Form". The browser's address bar is empty. The main content area of the browser shows a simple login form centered on a light gray background. The form consists of two text input fields: the first is labeled "Username:" and the second is labeled "Password:". Below these fields is a blue button with the text "Login" in white. The browser's taskbar is visible at the bottom, showing various application icons and the system clock indicating 23:18 on 19-03-2025.

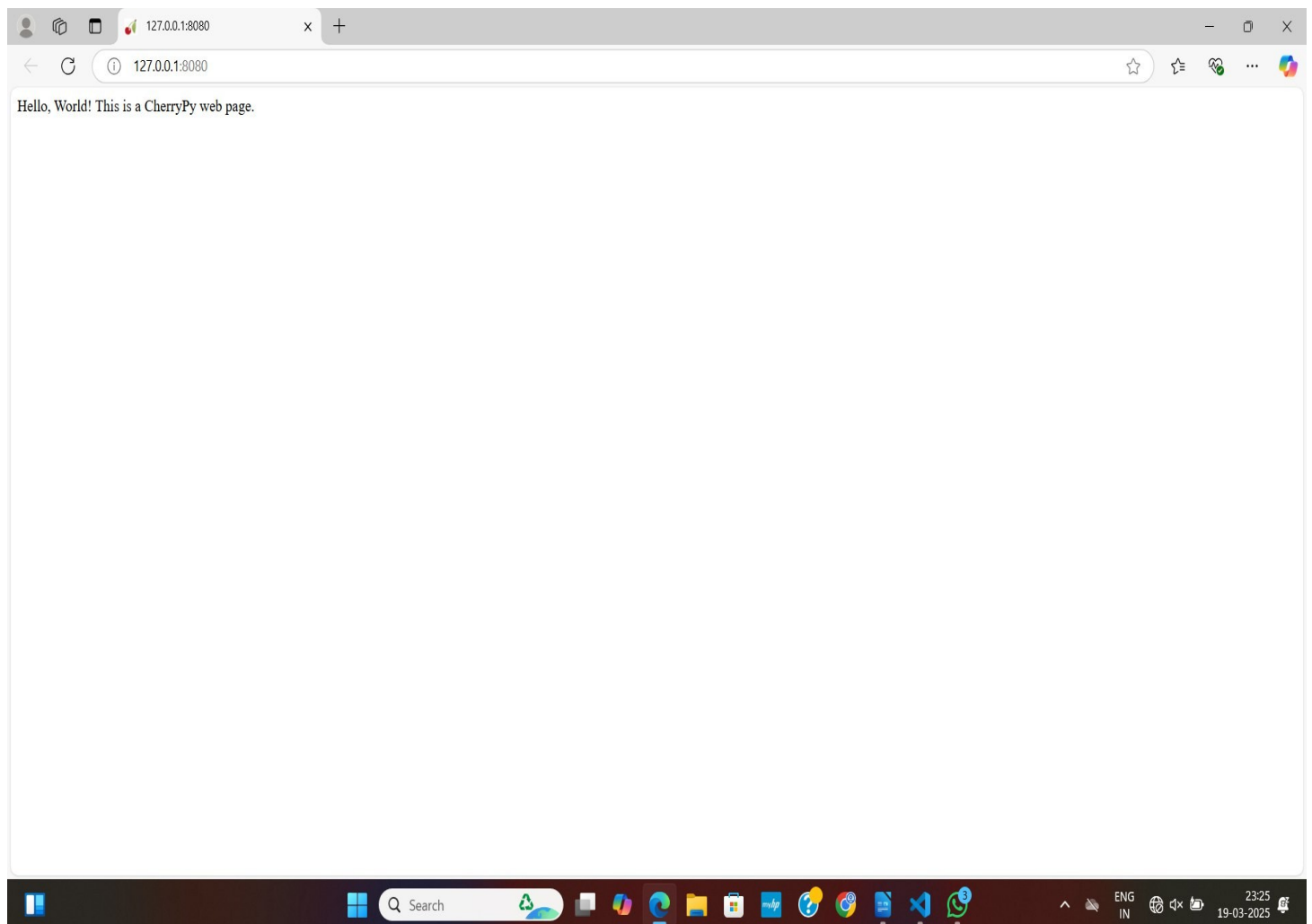


**CherryPy:**

```
import cherrypy

class HelloWorld:
    @cherrypy.expose def
    index(self):
        return "Hello, World! This is a CherryPy web page."

if __name__ == '__main__': cherrypy.quickstart(HelloWorld())
```

**Output:**

## **Flask:**

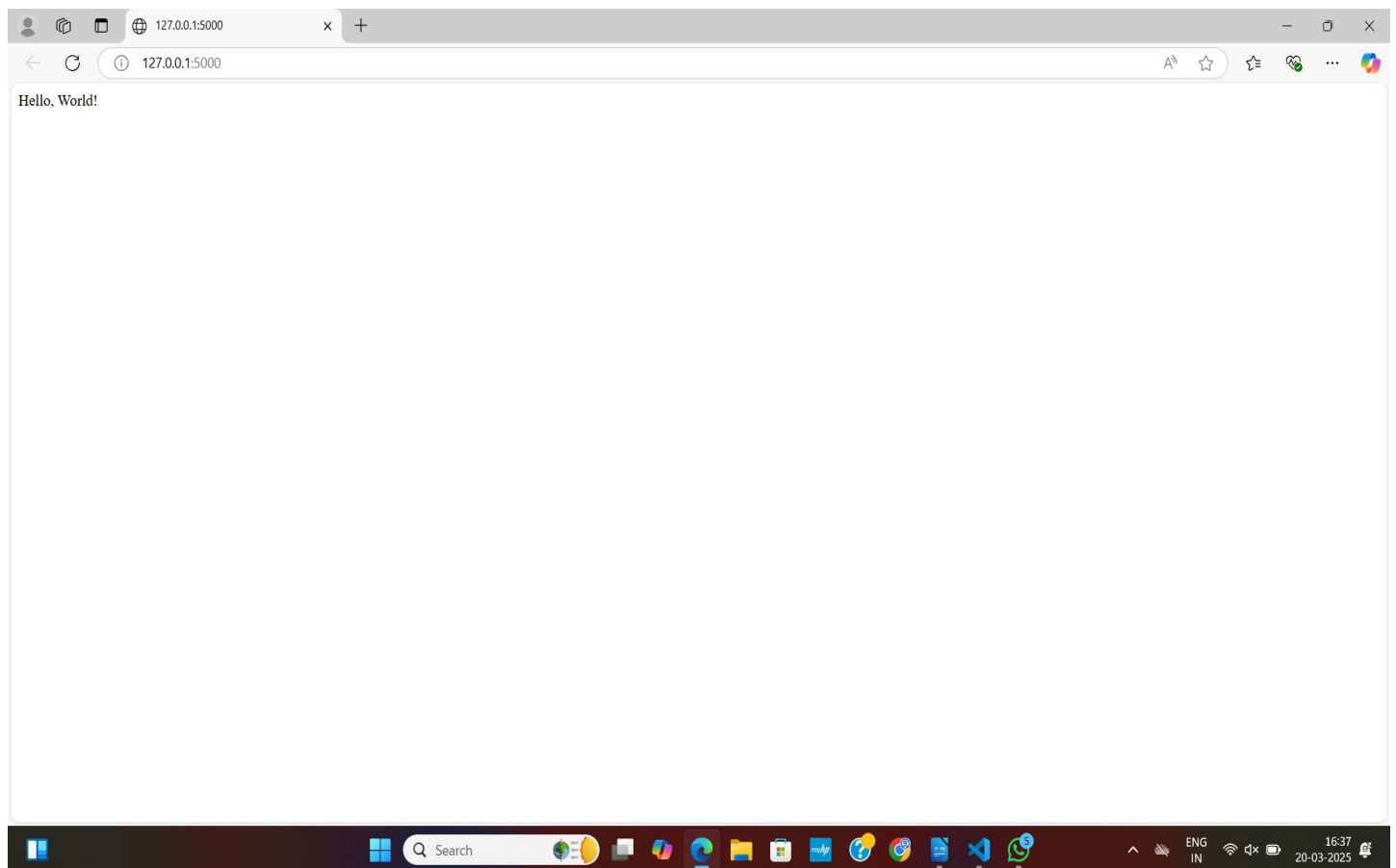
```
from flask import Flask

# Create a Flask application instance
app = Flask(__name__)

# Define a route for the root URL ("/") @app.route('/')
def hello_world(): return
    'Hello, World!'

# Run the Flask application if
__name__ == '__main__':
    app.run(debug=True)
```

## **Output:**



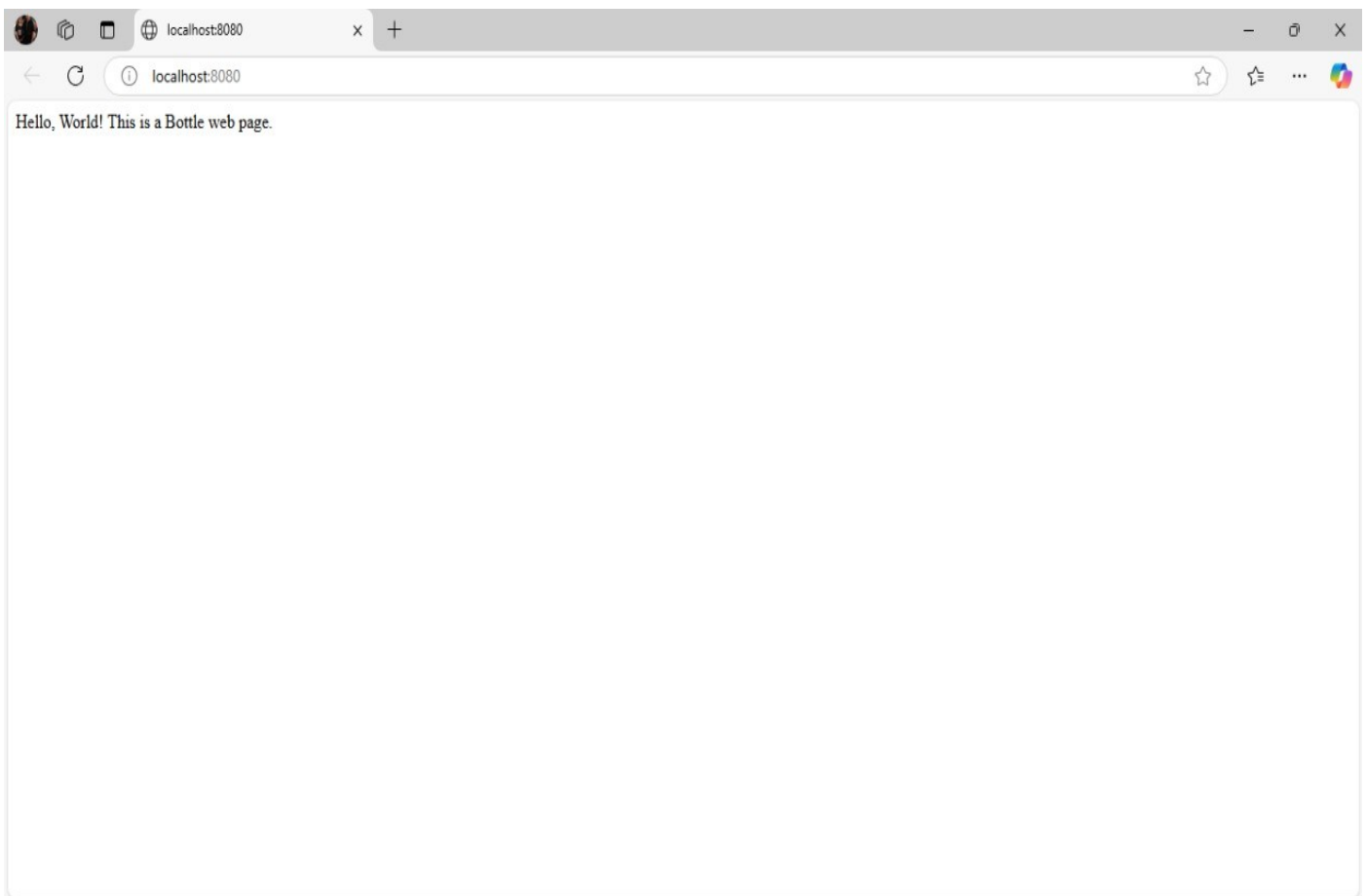
## **Bottle:**

```
from bottle import route, run

@route('/')
def index():
    return "Hello, World! This is a Bottle web page."

run(host='localhost', port=8080)
```

## **Output:**





**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**

Asst. Professor & HOD

[it@intugvce.edu.in](mailto:it@intugvce.edu.in)

Email:

hod.

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Django installation process
7. Date of Experiment : 20-12-2024
- 8.Date of Submission of Report :27-12-2024

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature Faculty:**

DJANGO FRAMEWORK II B-Tech II semester 2025

## **DJANGO INSTALLATION PROCESS:**

### **Step 1: Install Python**

Django requires Python to run. Follow these steps:

1. Check if Python is already installed

Open the terminal (Command Prompt or PowerShell for Windows) and run:

#### **COMMAND:**

```
python --version
```

If Python is installed, it will show the version. Django requires **Python 3.6 or later**.

### **Step 2: Install Virtual Environment (Recommended)**

A virtual environment helps manage dependencies for different projects separately.

1. Install venv (if not already installed) **COMMAND:**

```
python -m ensurepip --default-pip
```

2. Create a virtual environment

Navigate to the project folder in the terminal and run:

#### **COMMAND:**

```
python -m venv
```

This creates a folder named venv, which contains a separate Python environment.

3. Activate the virtual

#### **environment COMMAND:**

```
venv\Scripts\activate
```

**After activation, you'll see (venv) at the beginning of your terminal.**

### **Step 3: Install Django**

Once the virtual environment is activated, install Django using pip:

#### **COMMAND:**

```
pip install django
```

To verify the installation, run:

**COMMAND:**

```
django-admin --version
```

 This should display the installed Django version.**Step 4: Create a Django Project**

To start a new Django project

**COMMAND:**

```
django-admin startproject myproject
```

Replace myproject with your desired project name.

Move into the project directory:

**COMMAND:**

```
cd myproject
```

**Step 5: Run the Development Server**

Start the built-in Django development server to check if everything is set up correctly:

**COMMAND:**

```
python manage.py runserver
```

You should see output like this:

**COMMAND:**

Starting development server at <http://127.0.0.1:8000/>

Open your browser and go to **<http://127.0.0.1:8000/>** to see the Django welcome page.



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Django app creation
7. Date of Experiment : 27-12-2024
- 8.Date of Submission of Report :03-01-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

## **DJANGO APP CREATION:**

### **Step 1: Navigate to Your Django Project Directory**

Make sure you are inside your Django project folder:

#### **COMMAND:**

```
cd myproject
```

### **Step 2: Create a New Django App**

Run the following command:

#### **COMMAND:**

```
python manage.py startapp myapp
```

Replace myapp with your desired app name.

### **Step 3: Register the App in settings.py**

Open myproject/settings.py and add your app to the INSTALLED\_APPS list:

#### **COMMAND:**

```
INSTALLED_APPS = [  
    # Default Django apps  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
  
    # Your app  
    'myapp',  
]
```

Your Django app is now created and registered. 🚀





**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Connecting views an app
7. Date of Experiment : 03-01-2025
- 8.Date of Submission of Report :24-01-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**\ Signature of Faculty:**

**DJANGO FRAMEWORK II B-Tech II semester 2025**

## **Connecting Views and URLs:**

### **Set Up URLs:**

Project-level URL Configuration:

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('myapp1.urls')), # Include app URLs
]
```

### **App-level URL Configuration**

```
from django.urls import path
from .views import home
urlpatterns = [
    path("", home, name='home'),
]
```

### **Create a Sample View:**

```
from django.http import HttpResponse
def home(request):
    return HttpResponse("<h1>Welcome to My Django App!</h1>")
```

### **Run Migrations:**

```
python manage.py migrate
```

### **Run the Server and Test:**

```
python manage.py runserver
```



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Exploring views.py
7. Date of Experiment : 24-01-2025
- 8.Date of Submission of Report :31-01-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

DJANGO FRAMEWORK II B-Tech II semester 2025

## **Exploring Views.py:**

In Django, views.py is the file where you define functions or classes that handle requests and return responses. Views act as the logic layer of a Django web application, controlling how data is processed and which HTML templates are displayed.

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth import authenticate, login, logout
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from django.contrib.auth.models import User # ✅ Added missing import
from .models import Student, Faculty, StudentAttendance, FacultyAttendance, Branch
from .forms import AddStudentForm, AddFacultyForm, StudentAttendanceForm,
FacultyAttendanceForm
```

# ✅ Index Page

```
def index(request):
    return render(request, "attendance/index.html")
```

# ✅ Register Page (Selection between Student & Faculty)

```
def register(request):
    return render(request, "attendance/register.html")
```

# ✅ Student Registration

```
def student_register(request):
    if request.method == "POST":
        name = request.POST.get("name", "").strip()
        email = request.POST.get("email", "").strip()
        password1 = request.POST.get("password1", "")
        password2 = request.POST.get("password2", "")
        branch_id = request.POST.get("branch", "").strip()
        roll_number = request.POST.get("roll_number", "").strip()

        if not all([name, email, password1, password2, roll_number, branch_id]):
            messages.error(request, "All fields are required.")
            return redirect("student_register")

        if password1 != password2:
            messages.error(request, "Passwords do not match.")
            return redirect("student_register")

        if User.objects.filter(email=email).exists():
            messages.error(request, "Email is already registered.")
            return redirect("student_register")

        user = User.objects.create_user(username=email, email=email,
            password=password1, first_name=name)

        branch = get_object_or_404(Branch, id=branch_id) # ✅ Optimized query
```

```
if Student.objects.filter(roll_number=roll_number, branch=branch).exists():
    messages.error(request, "Roll number already exists in this branch.")
    return redirect("student_register")
```

```
Student.objects.create(user=user, branch=branch, roll_number=roll_number)
messages.success(request, "Student registration successful! Please log in.")
return redirect("login")
```

```
branches = Branch.objects.all()
return render(request, "attendance/student_register.html", {"branches": branches})
```

#### # Faculty Registration

```
def faculty_register(request):
```


```
    if request.method == "POST":
        name = request.POST.get("name", "").strip()
        email = request.POST.get("email", "").strip()
        password1 = request.POST.get("password1", "")
        password2 = request.POST.get("password2", "")
        branch_id = request.POST.get("branch", "").strip()
```

```
    if not all([name, email, password1, password2, branch_id]):
        messages.error(request, "All fields are required.")
        return redirect("faculty_register")
```

```
    if password1 != password2:
        messages.error(request, "Passwords do not match.")
        return redirect("faculty_register")
```

```
    if User.objects.filter(email=email).exists():
        messages.error(request, "Email is already registered.")
        return redirect("faculty_register")
```

```
    user = User.objects.create_user(username=email, email=email,
password=password1, first_name=name)
```

```
    branch = get_object_or_404(Branch, id=branch_id) #  Optimized query
    Faculty.objects.create(user=user, branch=branch)
```

```
    messages.success(request, "Faculty registration successful! Please log in.")
    return redirect("login")
```

```
    branches = Branch.objects.all()
    return render(request, "attendance/faculty_register.html", {"branches": branches})
```

#### # User Login

```
def login_view(request):
```

```
    if request.method == "POST":
        email = request.POST.get("email", "").strip()
        password = request.POST.get("password", "")
```

```

    user = authenticate(request, username=email, password=password)
    if user is not None:
        login(request, user)
        return redirect("home")
    else:
        messages.error(request, "Invalid email or password")
        return redirect("login")

return render(request, "attendance/login.html")

#  User Logout
def user_logout(request):
    logout(request)
    messages.success(request, "You have been logged out successfully.")
    return redirect("login")

#  Home Page (After Login)
@login_required
def home(request):
    return render(request, "attendance/home.html")

#  Student Attendance
@login_required
def student_attendance(request):
    student = getattr(request.user, 'student', None)
    if not student:
        messages.error(request, "Only students can access this page.")
        return redirect("home")



    attendance_records = StudentAttendance.objects.filter(student=student)
    return render(request, "attendance/student_attendance.html", {"attendance_records":
attendance_records})

#  Add Student
@login_required
def add_student(request):
    if request.method == "POST":
        form = AddStudentForm(request.POST)
        if form.is_valid():
            student = form.save(commit=False)
            student.user = request.user #  Associate the logged-in user with the student
            student.save()
            messages.success(request, "Student added successfully.")
            return redirect("home")
        else:
            messages.error(request, "Failed to add student. Please check the form.")
    else:
        form = AddStudentForm()



```

```

return render(request, "attendance/add_student.html", {"form": form})

#  Add Faculty
@login_required
def add_faculty(request):
    if request.method == "POST":
        form = AddFacultyForm(request.POST)
        if form.is_valid():
            faculty = form.save(commit=False)
            faculty.user = request.user #  Associate the logged-in user with the faculty
            faculty.save()
            messages.success(request, "Faculty added successfully.")
            return redirect("home")
        else:
            messages.error(request, "Failed to add faculty. Please check the form.")
    else:
        form = AddFacultyForm()

    return render(request, "attendance/add_faculty.html", {"form": form})

#  Faculty Attendance
@login_required
def faculty_attendance(request):
    if request.method == "POST":
        form = FacultyAttendanceForm(request.POST)
        if form.is_valid():
            attendance = form.save(commit=False)
            attendance.faculty = request.user.faculty #  Associate the attendance with the logged-in faculty
            attendance.save()
            messages.success(request, "Faculty attendance marked successfully.")
            return redirect("home")
        else:
            messages.error(request, "Failed to mark attendance. Please check the form.")
    else:
        form = FacultyAttendanceForm()

    return render(request, "attendance/faculty_attendance.html", {"form": form})
views

```

In Django, the `urls.py` file is responsible for mapping URLs to views. It acts as the router of your application, directing user requests to the correct function in `views.py`.

### **Creating `urls.py` in a Django App:**

Each Django app should have its own `urls.py` file to define app-specific routes.

Steps to Create `urls.py` in a Django App

- 🕒 Inside your Django app folder (myapp1), create a file named **`urls.py`**. Define URL patterns to map URLs to views.



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : APP URL.py
7. Date of Experiment : 24-01-2025
- 8.Date of Submission of Report :31-01-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**



## **APP URL .PY**

```
from django.urls import path
from . import views

urlpatterns = [
    path("", views.index, name="index"),
    path("home/", views.home, name="home"),
    path("login/", views.login_view, name="login"),
    path("logout/", views.user_logout, name="logout"),
    path("register/", views.register, name="register"),
    path("student_register/", views.student_register, name="student_register"),
    path("faculty_register/", views.faculty_register, name="faculty_register"),
    path("student_attendance/", views.student_attendance,
name="student_attendance"),
    path("faculty_attendance/", views.faculty_attendance, name="faculty_attendance"),
    path("add_student/", views.add_student, name="add_student"),
    path("add_faculty/", views.add_faculty, name="add_faculty"),
]
```



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Templates
7. Date of Experiment : 31-01-2025
- 8.Date of Submission of Report :17-02-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

DJANGO FRAMEWORK II B-Tech II semester 2025

## **FRONT END PART**

### **Templates:**

Templates in Django are HTML files that display dynamic content. They separate the frontend (UI) from the backend logic, following the MVT (Model-View-Template) architecture.

**Where to Store Templates?** By default, Django looks for templates in a folder named **templates/** inside your app.

### **LOGIN PAGE:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login and Registration</title>
  <style>
    body {
      font-family: cursive;
      background: url('https://wallpapers.com/images/hd/technology-circuit-pattern-black-
u650scpdr1pa5w1f.webp');
      background-size:
      cover; display: flex;
      justify-content:
      center; align-items:
      center; height:
      100vh; margin: 0;
    }
    .container {
      padding: 20px;
      border-radius:
      8px;
      box-shadow: 0 0 10px rgba(0, 0, 0,
      0.09); width: 300px;
      height: fit-content;
      background-color: rgba(255, 255, 255, 0.8);}
    .container.hidden
    { opacity: 0;
      transform:
      scale(0.9); pointer-
      events: none;
    }.container h2 {
      margin-bottom:
      20px; color: #333;}
    .container form { display:
    flex;
```

```

        flex-direction: column;
    }.container form input,
    .container form
        select { margin-
            bottom: 10px;
            padding: 10px;
            border: 1px solid
                #ccc; border-
                radius: 4px;}
    container form button {
        padding: 10px;
        background-color:
            #555; color: #fff;
        border: none;
        border-radius:
            4px; cursor:
            pointer;}
    .container form
        button:hover
        { background-color:
            #333;}
    .container .toggle-link,
    .container .forgot-
        password { text-align:
            center;
            margin-top: 10px;

            color:
                #4c82ff;
            cursor:
                pointer;
        }
    .container .forgot-
        password { font-size:
            15px;
    }.footer {
        text-align: center;
        color: rgb(246, 246,
            163); position:
            absolute; bottom:
            10px;
            width: 100%;}
</style>
</head>
<body>
    <div class="container" id="login-container">

```

```

<h2>Login</h2>
<form id="login-form">
  <input type="text" placeholder="Username" required>
  <input type="password" placeholder="Password" required>
  <button type="submit">Login</button>

```

```

</form>
<div class="forgot-password" onclick="forgotPassword()">Forgot Password?</div>
<div class="toggle-link" onclick="toggleForm()">Don't have an account?
<b>Register</b></div>
</div>
<div class="container hidden" id="register-container">
  <h2>Register</h2>
  <form id="register-form">
    <input type="text" placeholder="Username" required>
    <input type="password" placeholder="Password" required><input type="email"
    placeholder="Email" required>
    <select required>
      <option value="" disabled selected>Select Role</option>
      <option value="student">Student</option>
      <option value="staff">Staff</option>
      <option value="hod">HOD</option>
    </select><br>
    <button type="submit">Register</button>
  </form>
  <div class="toggle-link" onclick="toggleForm()">Already have an account? Login</div>
</div>

<script>
  document.getElementById('login-form').addEventListener('submit', function(event) {
    event.preventDefault();
    // Perform login validation here
    window.location.href =
    'Homepage.html';
  });
  function toggleForm() {
    const loginContainer = document.getElementById('login-container');
    const registerContainer = document.getElementById('register-
    container'); loginContainer.classList.toggle('hidden');
    registerContainer.classList.toggle('hidden');}
  function forgotPassword() {window.location.href = 'Forgotpasswordpage.html';
  }
</script>
</body>
<footer class="footer">
  <p>&copy; Done by the members of students of IT</p>
</footer>
</html>

```

## **Output:**



## **HOMEPAGE:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>University Attendance Management System</title>
  <link rel="stylesheet" href="styles.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-image: url('https://wallpapers.com/images/high/blue-technology-
computer-interface-q6vspdtjbjhc0d.webp');
      background-size: cover;
      color: #333;
    }
    header {
      background-color: #000000;
      color: rgb(87, 143, 255);
      padding: 5px;
      text-align: center;
      opacity: 60%;
      position: relative;
    }
  }
```

```
    header h1 {
      font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans
Unicode', Geneva, Verdana, sans-serif;
      font-size: 45px;
      margin: 0;
    }
    nav {
      position:
      absolute; top:
      50%;
      left: 10px;
      transform: translateY(-50%);
    }
    .dropdown {
      display: inline-block;
    }
    .dropdown-content
    { display: none;
      position:
```

```

    absolute;
    background-color:
    #f9f9f9; min-width:
    160px;
    box-shadow: 0px 8px 16px 0px
    rgba(0,0,0,0.2); z-index: 1;
}.dropdown-content a { color: black; padding: 12px ;
    text-decoration:
    none; display: block;
}
.dropdown-content a:hover {
    background-color:
    #fb8181; text-align:
    center;
}
.dropdown:hover .dropdown-content
{ display: block;
}
main {
    padding: 2rem;
}
section {
    margin-bottom: 2rem;
    padding: 1rem;
    background-size:
    cover; border-radius:
    8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

```



```

h2 {
  color: #a25f83;
  text-align:
  center; font-
  size: 30px;
}
p {
  text-align: center;
  color: rgb(208, 22,
  22); font-size: 12px;
}.button {
  display: inline-block;
  box-sizing: border-
  box; padding: 0.5rem
  1rem;
  background-color:
  #ffd5d2; position:
  relative;
  color: rgb(5, 5, 5);
  text-decoration:
  none; border-
  radius: 4px;
  transition: background-color 0.3s;
}.button:hover {
  background-color: #45a049;
}footer {
  background-color:
  #282626; color: white;
  text-align: center;
  color: #333;
  padding: 0%;
  position:
  fixed; width:
  100%;
  bottom: 0;
}
</style>
</head>
<body>
<header>
<h1>University Attendance Management System</h1>
<nav>
  <div class="dropdown">
    <button class="dropbtn">Menu</button>
    <div class="dropdown-content">
      <a href="Homepage.html">Home</a>
      <a href="studentdash.html">Student Dashboard</a>
      <a href="About.html">About</a>
      <a href="Features.html">Features</a>
    
```

```
<a href="Studentprofile.html">Profile</a>  
<a href="Contact.html">Contact</a>
```

```
<a href="Logout.html">Logout</a>  
</div>  
</div>  
</nav>  
</  
header>  
</body>  
</html>
```

### Output:

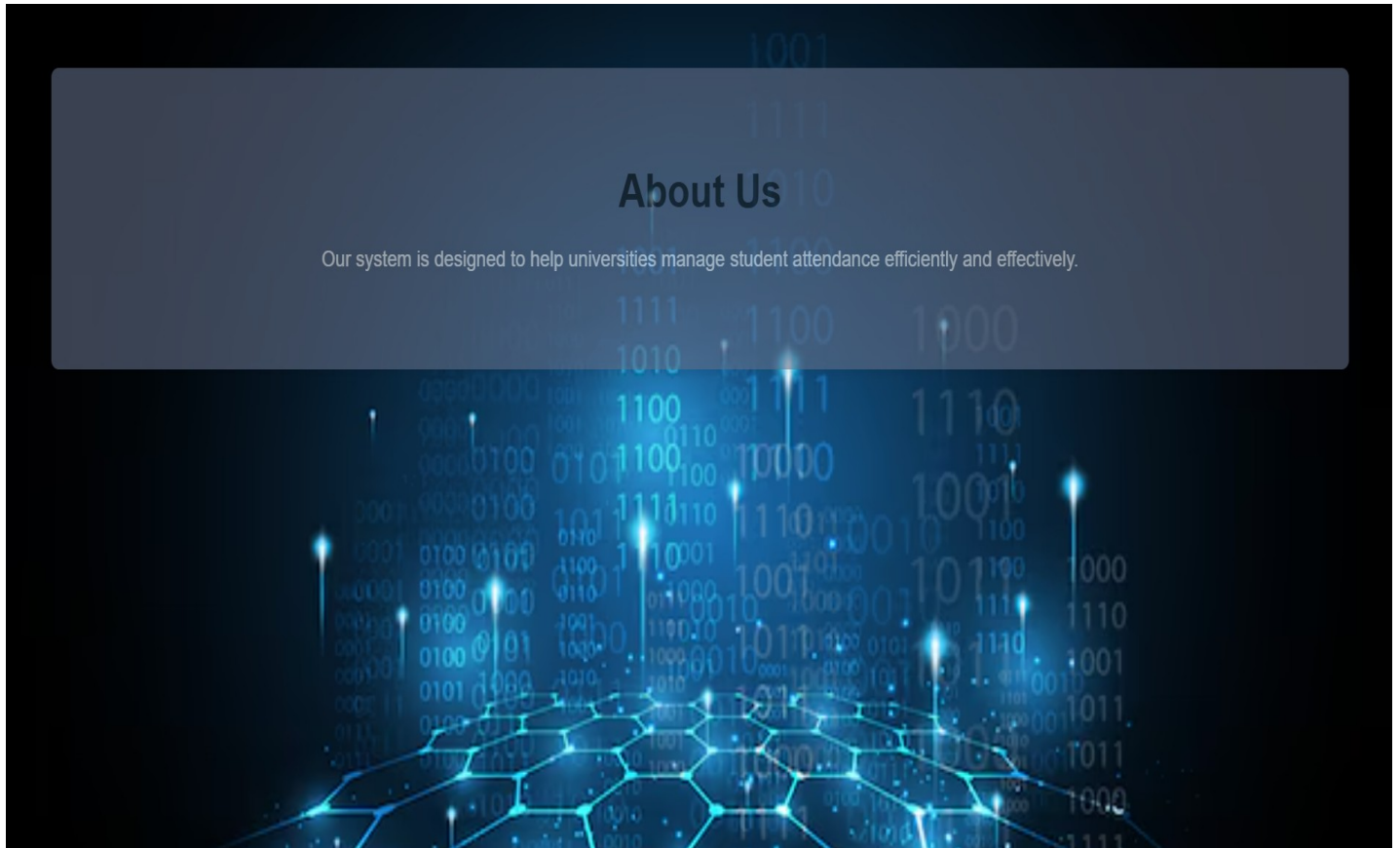


## **ABOUT:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About Us</title>
  <style>
    body {
      font-family: Arial, sans-
        serif; margin: 0;
      padding: 0;
      background-image: url("https://img.freepik.com/premium-photo/security-cyber-
digital- concept-abstract-technology-background_42421-211.jpg?
ga=GA1.1.1274352584.1740070781&semt=ais_hybrid");
      background-size: cover;
      background-attachment:
        fixed; text-align: center;
    }
    #about {
      background-color: rgb(97, 108, 134);
      padding: 50px 20px;
      text-align: center;
      box-shadow: 0 4px 8px rgba(0, 0, 0,
        0.1); margin: 1.5cm;
      border-radius: 8px;
      background-size:
        cover; opacity:
        60%;
    }
    #about h2 {
      font-size: 2.5em;
      margin-bottom:
        20px; color:
        #1a1a1a;
    }
    #about p {
      font-size:
        1.2em; color:
        #ffffff; line-
        height: 1.6;
      margin-bottom: 30px;
    }
    .button {
      display: inline-
        block; padding:
        10px 20px; font-
        size: 1em; color:
```

```
#fff;
background-color:
#007BFF; text-
decoration: none;
border-radius: 5px;

    transition: background-color 0.3s ease;
}
.button:hover {
    background-color: #0056b3;
}
</style>
</head>
<body>
    <section id="about">
        <h2>About Us</h2>
        <p>Our system is designed to help universities manage student attendance
efficiently and effectively.</p>
    </section>
</body>
</html>
```

**Output:**

**FEATURES:**

```

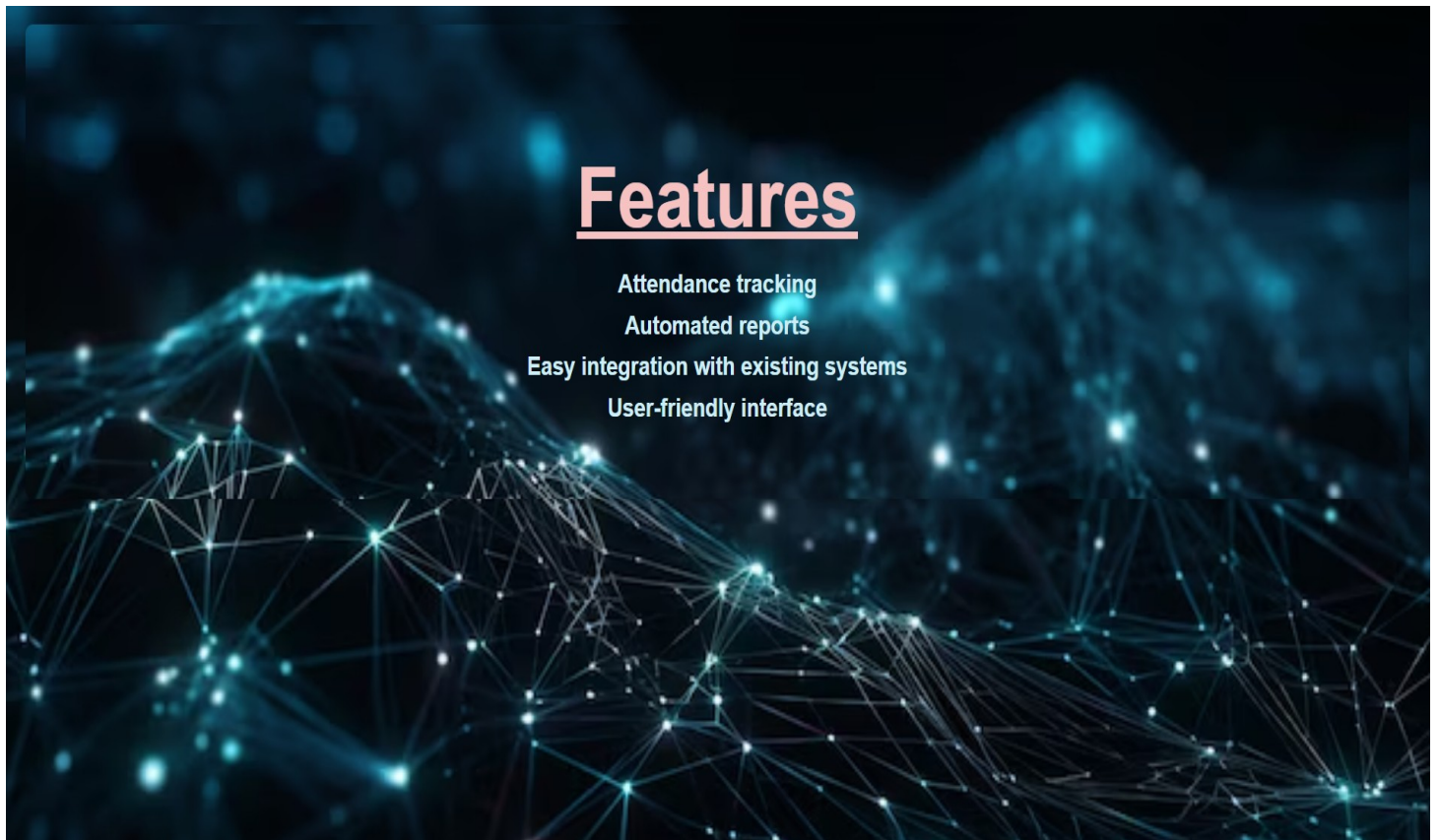
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Features</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-image: url('https://img.freepik.com/free-photo/glowing-molecular-structure-
futuristic-galaxy-backdrop-generated-by-ai_188544-27827.jpg?
ga=GA1.1.1274352584.1740070781&semt=ais_hybrid');
      background-size: cover;
    }
    #features {

      padding: 50px 20px; color:
      beige;
      text-align: center;

    }
    #features h2 {
      font-size: 1.9cm; margin-
      bottom: 20px; color:
      honeydew;
    }
    #features ul {
      list-style: none;
      padding: 0;
    }
    #features ul li { font-
      size: 22px; margin:
      10px ; color:linen;
      text-align: center;
    }
    .button {
      display: inline-block;
      margin-top: 20px;
      padding: 10px 20px; font-
      size: 1em; color: #fff;
      background-color: #007BFF;
      text-decoration: none;
      border-radius: 5px;
      transition: background-color 0.3s ease;
    }
    .button:hover {background-color: #0056b3;}
  
```

```
</style>
</head>
<body>
  <section id="features">
    <h2><u>Features</u></h2>
    <ul>
      <li><b>Attendance tracking</b> </li>
      <li><b>Automated reports</b></li>
      <li><b>Easy integration with existing systems</b></li>
      <li><b>User-friendly interface</b></li>
    </ul>
  </section>
</body>
</html>
```

### Output:



**CONTACT:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contact Us</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-image: url("https://wallpapers.com/images/hd/hexagonal-technology-motion-
grid-6497yh099hkpfd3.webp");
    }
    #contact {
      background-color: #d9f5fb;
      padding: 50px;
      margin: 200px auto;
      max-width: 600px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      border-radius: 8px;
      text-align: center; opacity:
60%; background-size:
cover; color: black;
    }
    #contact h2
    { color: #333;
      margin-bottom: 20px;
      font-size: 40px;
    }
    #contact p { color:
      black; line-height:
      1.6; font-size: 15px;
    }
    #contact a { color:
      #342826;
      text-decoration: none;
      font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
      font-size: 20px;}
    #contact a:hover {
      text-decoration: underline;
    }.button {
display: inline-block; padding:
10px 20px; margin-top: 20px;
background-color: #f5f78d;
color: #fff;

```



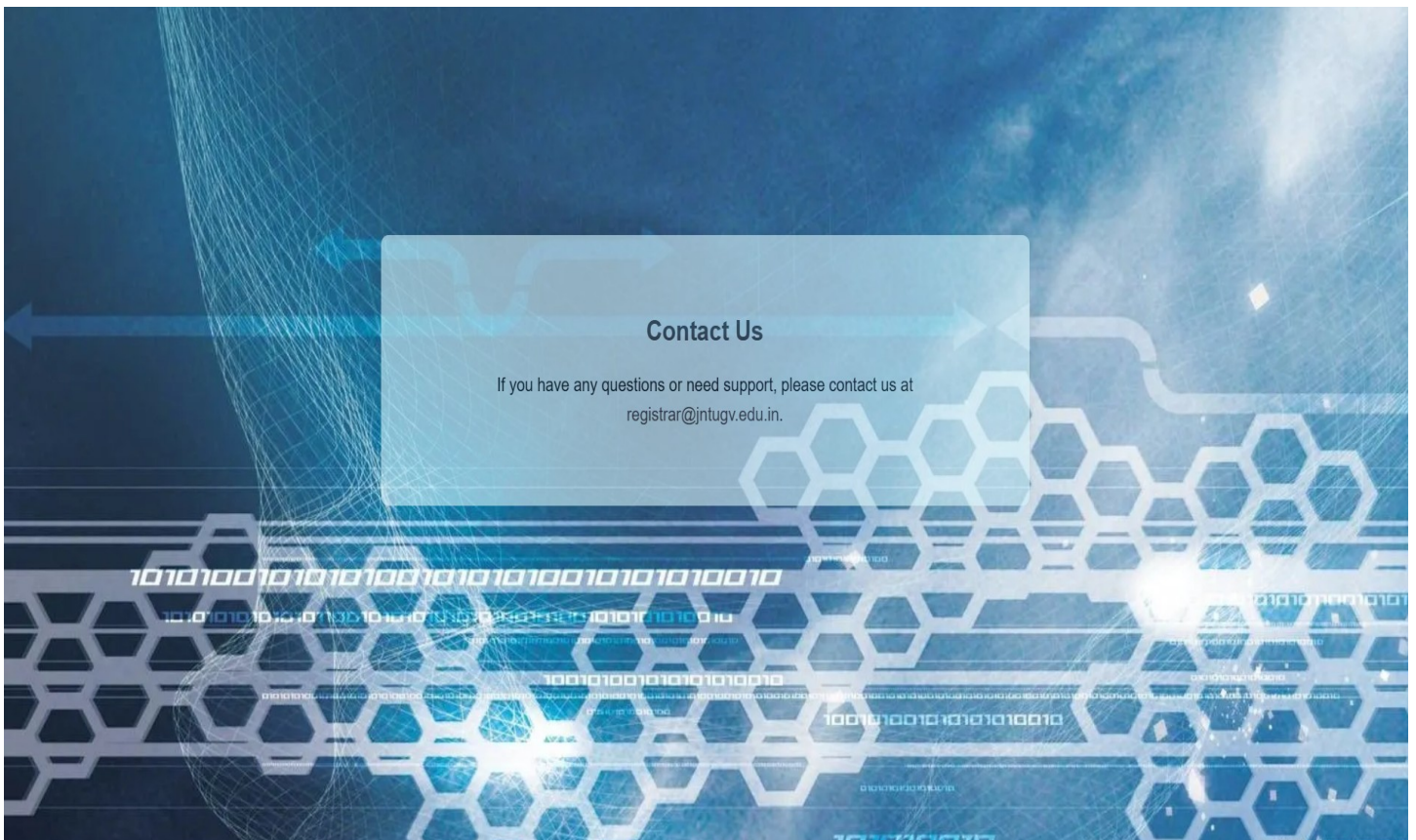
```

text-decoration: none;
border-radius: 5px;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #ffffff;
}
</style>
</head>
<body>
<section id="contact">
<h2>Contact Us</h2>
<p>If you have any questions or need support, please contact us at <a
href="mailto:registrar@jntugv.edu.in">registrar@jntugv.edu.in</a>.</p>

</section>
</body>
</html>

```

### Output:



**ATTENDANCE ACTION:**

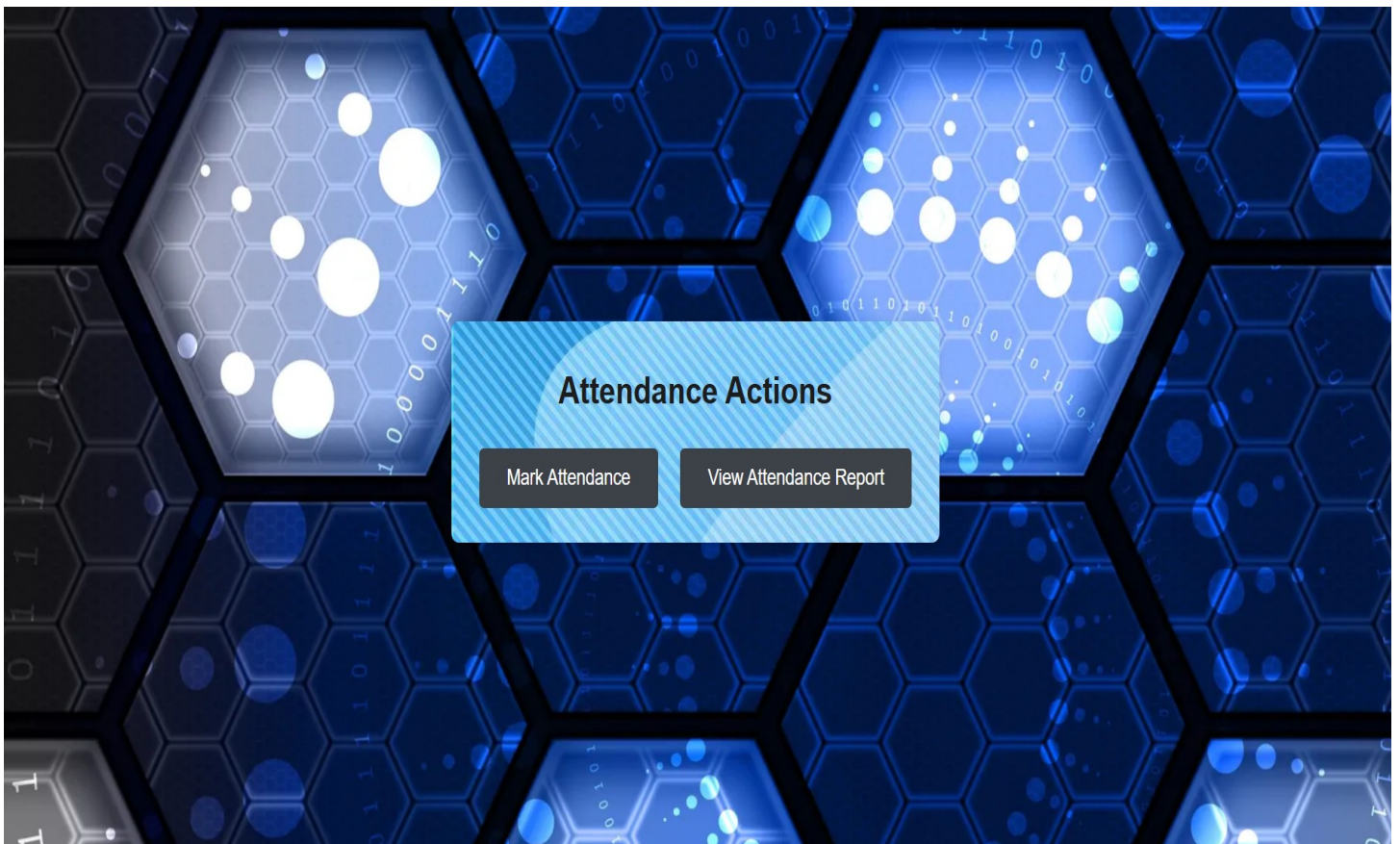
```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Attendance Actions</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-image: url('https://wallpapers.com/images/hd/technology-futuristic-hexagon-
pattern-50pc6fr3nqktardn.webp');
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background-image: url('https://wallpapers.com/images/hd/blue-abstract-technology-
circuit-5i5r4rslj1yqs72v.webp');
      background-position: 2px;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      text-align: center;
    }
    h1 {
      color: #1d1d1d;
    }
    .button {
      display: inline-block;
      margin: 10px; padding:
      15px 30px; font-size:
      18px; color: #fff;
      background-color: #3c4248;
      border: none;
      border-radius: 5px;
      cursor: pointer;
      text-decoration: none;
      transition: background-color 0.3s, transform 0.3s;
    }
    .button:hover {background-color: #f86b6d; transform: scale(1.05);}
    .button:active {

```

```
background-color: #d9534f;
transform: scale(0.95);
}
</style>
</head>
<body>
  <div class="container">
    <h1>Attendance Actions</h1>
    <a href="Markattendance.html" class="button">Mark Attendance</a>
    <a href="Viewreport.html" class="button">View Attendance Report</a>
  </div>
</body>
</html>
```

### Output:



## **STUDENT DASHBOARD:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Student Dashboard</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-image: url('https://img.freepik.com/free-photo/digital-blue-hud-interface-laptop-concept_53876-15831.jpg?ga=GA1.1.1274352584.1740070781&semt=ais_hybrid');
      background-size: cover;
      margin: 0;
      padding: 2%;
    }
    .container { width:
      100%; margin:
      auto;
      overflow: hidden;
    }
    header { background:
      #333; color: #fff;
      padding-top: 10px;
      min-height: 70px;
    }
    header a { color:
      #fff;
      text-decoration: none; text-
      transform: uppercase; font-
      size: 16px;
    }
    header ul
      { padding: 0;
        list-style:georgian;
      }
    header li { float:
      left;
      display: inline;
      padding: 0 20px 0 20px;
    }
    header #branding { float:
      left;
    }
    header #branding h1
      { margin: 0;

```

```

}
header nav {float: right; margin-top: 10px;
}
.main-content { padding:
    20px; background:
    #fff; margin-top: 20px;
}
.card {

```

```

    background: #fff; border:
    1px solid #ddd; margin-
    bottom: 20px; padding:
    20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.card h2 { margin-top:
    0;
}
.attendance-table
{ width: 100%;
border-collapse: collapse;
}
.attendance-table th, .attendance-table td
{ border: 1px solid #fbbaba;
padding: 8px; text-
align: center;
}
.attendance-table th
{ background-color: beige;
}
.btn {
display: inline-block;
padding: 10px 20px; font-
size: 16px;
cursor: pointer ;
text-align: center; text-
decoration: none; outline:
none;
color: #fdeaea; background-
color: #333; border: none;
border-radius: 15px;
box-shadow: 0 9px #999;
}
.btn:hover {background-color: #333}
.btn:active {
background-color: #333;
box-shadow: 0 5px #666;
transform: translateY(4px);
}
</style>

```



```
</head>
<body>
  <header>
    <div class="container">
      <div id="branding">
        <h1>Student Dashboard</h1>
      </div>
      <nav>
        <ul>
          <li><a href="Homepage.html">Home</a></li>
          <li><a href="Attendanceaction.html">Attendance</a></li>

        </ul>
```

```
      </nav>
    </div>
  </header>
</body>
</html>
```

### Output:



**MARK ATTENDANCE:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Student Attendance</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-image: url('https://wallpapers.com/images/hd/futuristic-technology-
computer- interface-k4qiauqbznzpwsebe.webp');
      background-position:
        cover; margin: 0;
      padding: 0;
      display:
        flex;
      justify-content:
        center; align-items:
        center; height:
        100vh;
    }
    .container {
      background-color: rgba(213, 248, 247, 0.8); /* Adjust the opacity here
      */ padding: 20px;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      width: 300px;
      text-align: center;
    }
    h1 {
      color: #333;
    }
    label {
      display: block;
      margin: 15px 0
      5px;
      color: #333; /* Darker text color */
    }
    input[type="text"], input[type="date"]
    { width: 75%;
      padding: 10px;
      margin-bottom:
        10px;
      border: 1px solid
        #ddd; border-radius:

```

```

    5px;
}
button {
    background-color:
    #e4f9cd; color: #fff;
    padding: 10px
    20px; border:
    none;
    border-radius: 5px; cursor: pointer;
    transition: background-color 0.3s; color:

```

```

    #333;
}
button:hover {
    background-color: #f9c659;
}
.alert {
    display:
    none;
    position:
    fixed; top:
    20px;
    left: 50%;
    transform:
    translateX(-50%);
    background-color:
    #4caf50; color: white;
    padding: 15px;
    border-radius:
    5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    animation: slideInOut 3s;
}
@keyframes slideInOut {
    0% { transform: translateX(-50%) translateY(-100%); opacity:
    0; } 10% { transform: translateX(-50%) translateY(0); opacity:
    1; } 90% { transform: translateX(-50%) translateY(0); opacity:
    1; }
    100% { transform: translateX(-50%) translateY(-100%); opacity: 0; }
}
</style>
<script>
    function handleSubmit(event) {
        event.preventDefault();
        // Show the animated alert
        const alertBox = document.getElementById('alertBox');
        alertBox.style.display = 'block';
        setTimeout(() =>
            { alertBox.style.display = 'none';

```



```

        // Redirect to the student dashboard
        window.location.href =
'file:///C:/Users/shara/Project/Batch1/Batch1_app/templates/studentdash.html';
    }, 3000); // Match the duration of the animation
    }
</script>
</head>
<body>
    <div class="container">
        <h1>Attendance Form</h1>
        <form onsubmit="handleSubmit(event)">
            <label for="studentName">Student Name</label>
            <input type="text" id="studentName" name="studentName" required>

```


```

<input type="text" id="rollNumber" name="rollNumber" pattern="[A-Za-z0-9]+"
required>
        <label for="date">Date</label>
        <input type="date" id="date" name="date" required>

        <button type="submit">Mark Attendance</button>
    </form>
</div>
    <div id="alertBox" class="alert">Attendance marked successfully!</div>
</body>
</html>

<label for="rollNumber">Roll Number</label>

```

**Output:**

The image shows a web application interface for an attendance form. The form is a light blue rectangular box centered on a dark background with futuristic, glowing green and blue geometric patterns, including hexagons and circular lines. The form has the following elements:

- Attendance Form** (Title)
- Student Name** (Label) with a text input field.
- Roll Number** (Label) with a text input field.
- Date** (Label) with a text input field showing the placeholder "dd-mm-yyyy" and a calendar icon.
- Mark Attendance** (Submit Button) in a yellow box.

In the background, there are faint, glowing green and blue geometric patterns, including hexagons and circular lines. On the right side, there is a small, glowing green rectangular box with some text inside, and a small, glowing green circular box with some text inside. The overall aesthetic is futuristic and high-tech.

**FORGOT PASSWORD:**

```

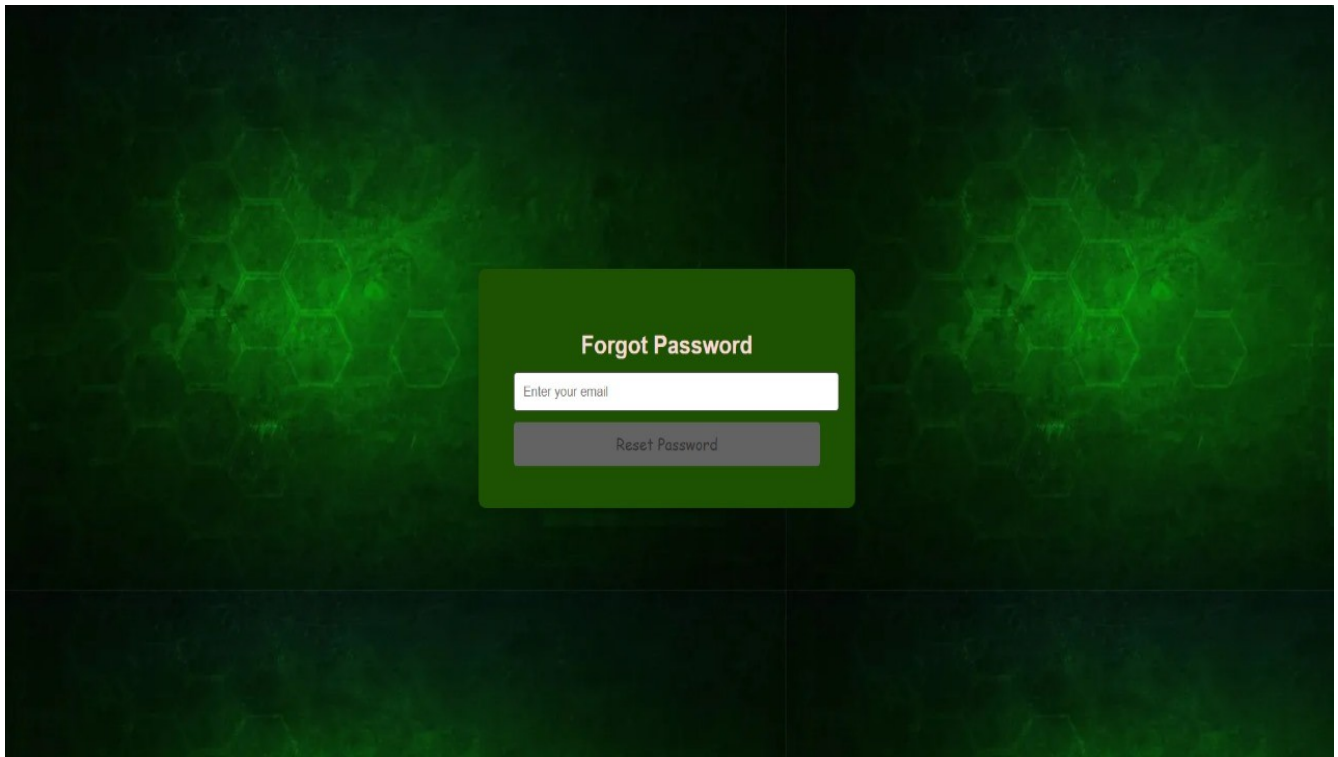
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Forgot Password</title>
  <style>
    body {
      font-family: Arial, sans-
      serif; background-color:
      #f4f4f4; display: flex;
      justify-content:
      center; align-items:
      center; height:
      100vh; margin: 0;
      background-image: url('https://wallpapers.com/images/high/green-
      technology-1280-x-800- wallpaper-8w1x8eu5rzgz8r3l.webp');
    }
    .container {
      background-color:
      #1e5203; padding:
      40px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0,
      0.1); width: 350px;
      text-align: center;
    }
    h2 {
      margin-bottom: 2px;

      color: #fbd9d9;
    }
    input[type="email"]
    { width: 100%;
      padding: 10px;
      margin: 10px 0;
      border: 1px solid
      #454545; border-
      radius: 4px;
    }
    button {
      width:
      100%;
      padding:
      10px;
      background-color:
      #626363; border: none;

```

```
border-radius: 4px;  
color: #383737; font-  
size: 16px;
```

```
    cursor:  
    pointer;  
    font-family: cursive;}  
button:hover {  
    background-color: #d2e3f5;}  
.message {  
    margin-top:  
    20px; color:  
    #666;}  
</style>  
</head>  
<body>  
    <div class="container">  
        <h2>Forgot Password</h2>  
        <form action="Resetpassword.html" method="POST">  
            <input type="email" name="email" placeholder="Enter your email" required>  
            <button type="submit">Reset Password</button>  
        </form>  
    </div>  
</body>  
</html>
```

**OUTPUT:**

**RESET PASSWORD:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Reset Password</title>
  <style>
    body {
      font-family: Arial, sans-
      serif; background-color:
      #f4f4f4; display: flex;
      justify-content:
      center; align-items:
      center; height:
      100vh; margin: 0;
      background-image: url('https://wallpapers.com/images/hd/technology-
      futuristic- building-67pwn4hh7umuplt4.webp');}
    .container {
      background-color: rgba(255, 255, 255, 0.8); /* Adjust the opacity
      here */ opacity: 62%;
      padding: 30px;
      border-radius:
      8px;
      box-shadow: 0 0 10px rgba(0, 0, 0,
      0.1); width: 300px;
      margin-left: 20px
      ;}
    h2
    {
      margin-bottom:
      20px; color: #333;}
    input[type="text"],
    input[type="password"] { width: 92.5%;
      padding: 10px;
      margin: 10px 0;
      border: 1px solid
      #ccc; border-
      radius: 4px;}
    button {
      width:
      100%;
      padding: 10px;

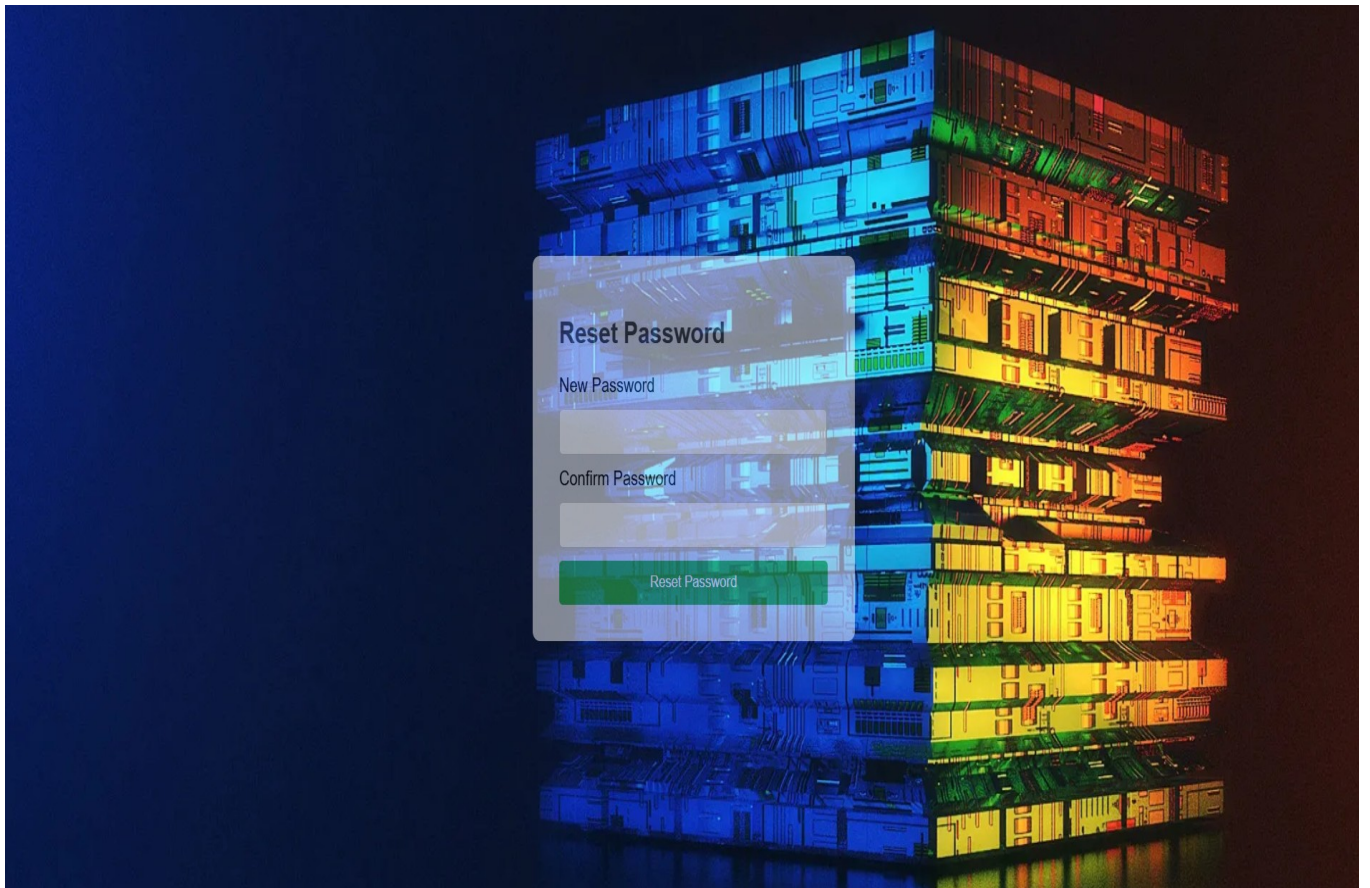
      background-color:
      #28a745; color: #fff;
      border: none; border-
      radius: 4px; cursor:

```

```

        pointer;}
        button:hover {background-color: #218838;}
    </style>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css">
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <script>
        function validatePasswords() {
            var newPassword = document.getElementById('new-password').value;
            var confirmPassword = document.getElementById('confirm-
password').value; if (newPassword !== confirmPassword) {
                Swal.fire({ icon
                    : 'error', title:
                    'Oops...',
                    text: 'Passwords do not match!',
                    showClass: {
                        popup: 'animate__animated animate__fadeInDown'
                    },
                    hideClass: {
                        popup: 'animate__animated animate__fadeOutUp'}});\return false;}
                \Swal.fire({icon: 'success',title: 'Success',
                    text: 'Password reset
                    successful!',showClass: {
                        popup: 'animate__animated animate__fadeInDown'},
                    hideClass: {popup: 'animate__animated animate__fadeOutUp'}});
                return true;}
            </script>
        </head>
        <body>
            <div class="container">
                <h2>Reset Password</h2>
                <form action="Loginpage.html" method="POST" onsubmit="return
                validatePasswords()">
                    <label for="new-password">New Password</label>

```

**Output:**



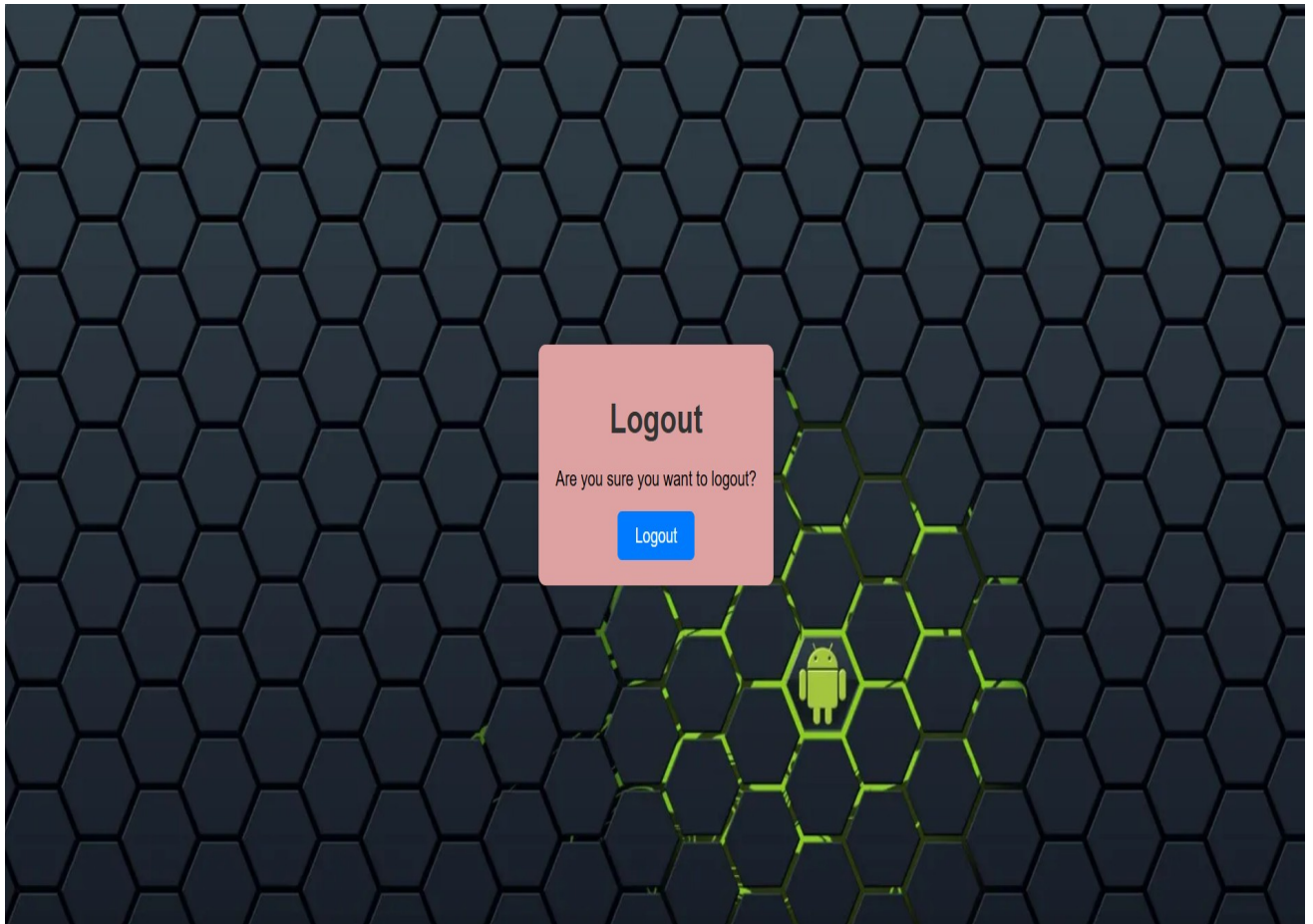
**LOGOUT:**

```

<!D-bottom: 20px; color: #333;
}
.logout-container
  button
    { background-color:
      #007bff; color: #fff;
      border: none;
      padding:
        10px 20px;
      border-
        radius: 5px;
      cursor:
        pointer;
      font-size: 16px;
    }
.logout-container
  button:hover
    { background-color:
      #0056b3;
    }
</style>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css">
</head>
<body>
  <div class="logout-container">
    <button onclick="logout()">Logout</button>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
  <script>
    function logout()
    { Swal.fire({
      title: 'Logged Out',
      text: 'You have been
        logged out.', icon:
        'success',
      showConfirmButton:
        false, timer: 2000,
      willClose: () => {
        window.location.href = 'Loginpage.html';
      }
    })}
  </script></body>
</html>

```

**Output:**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvce.edu.in](mailto:it@intugvce.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : What is forms.py in django
7. Date of Experiment : 17-02-2025
- 8.Date of Submission of Report :21-02-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

## What is forms.py in Django:

In Django, forms.py is used to handle user input efficiently and securely. It allows developers to create and manage forms without manually writing HTML and validation logic.

### Why Use forms.py:

- 🕒 Simplifies form creation
- 🕒 Handles input validation automatically
- 🕒 Integrates with Django models
- 🕒 Prevents security risks like SQL Injection & CSRF attacks

### Types of Forms in Django:

- 🕒 Django Forms (forms.Form) – Used for manually creating forms
- 🕒 Model Forms (forms.ModelForm) – Used to create forms directly from a Django model

#### code:

```
from django import forms
from .models import Student, Faculty, StudentAttendance, FacultyAttendance

class AddStudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = ['user', 'roll_number', 'branch', 'year']

class AddFacultyForm(forms.ModelForm):
    class Meta:
        model = Faculty
        fields = ['user', 'branch', 'is_class_teacher']

class StudentAttendanceForm(forms.ModelForm):
    class Meta:
        model = StudentAttendance
        fields = ['student', 'date', 'status']

class FacultyAttendanceForm(forms.ModelForm):
    class Meta:
        model = FacultyAttendance
        fields = ['faculty', 'date', 'status']
```



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvce.edu.in](mailto:it@intugvce.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : What is models.py and how to apply models
7. Date of Experiment : 21-02-2025
- 8.Date of Submission of Report :07-03-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

## What is models.py in Django:

In Django, models.py is where you define the database structure using Python code. Django models act as a bridge between the database and the application, allowing you to create, read, update, and delete records easily.

## Why Use Django Models:

No need to write raw SQL queries, Automatically creates tables in the database and

## How to Apply Models:

Create the Model in models.py:

- 🕒 Write your models inside the models.py file.

### Code:

```
from django.db import models
from django.contrib.auth.models import User

# Branch Model
class Branch(models.Model):
    name = models.CharField(max_length=100, unique=True)
    year_count = models.IntegerField(default=4) # Number of years in the branch, e.g., 4 for B.Tech

    def __str__(self):
        return self.name

# Student Model
class Student(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='student')
    roll_number = models.CharField(max_length=20, unique=True)
    branch = models.ForeignKey(Branch, on_delete=models.SET_NULL, null=True,
        related_name='students')
    year = models.IntegerField(default=1) # Can be used to store the year the student is in

    def __str__(self):
        return f"{self.user.first_name} - {self.roll_number}"

# Faculty Model
class Faculty(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='faculty')
    branch = models.ForeignKey(Branch, on_delete=models.SET_NULL, null=True,
        related_name='faculties')
    designation = models.CharField(max_length=100, default='Not Assigned') # Default value for designation
    is_class_teacher = models.BooleanField(default=False) # Add this field to the Faculty model

    def __str__(self):
        return f"{self.user.first_name} - {self.designation}"
```

```

# Student Attendance Model
class StudentAttendance(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE,
related_name='attendance_records')
    date = models.DateField()
    status = models.CharField(max_length=20, choices=[('Present', 'Present'), ('Absent', 'Absent')])

    def __str__(self):
        return f"{self.student} - {self.date} - {self.status}"

# Faculty Attendance Model
class FacultyAttendance(models.Model):
    faculty = models.ForeignKey(Faculty, on_delete=models.CASCADE,
related_name='attendance_records')
    date = models.DateField()
    status = models.CharField(max_length=20, choices=[('Present', 'Present'), ('Absent', 'Absent')])

    def __str__(self):
        return f"{self.faculty} - {self.date} - {self.status}"

```



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvce.edu.in](mailto:it@intugvce.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Run migrations to create a database table
7. Date of Experiment : 21-02-2025
- 8.Date of Submission of Report :07-03-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**



## Run Migrations to Create Database Tables:

After defining your models, run the following commands to apply them to the database:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

**Register Models in admin.py:**

### Admin.py:

```
from django.contrib import admin  
from .models import Branch, Student,  
Faculty, StudentAttendance,  
FacultyAttendance  
  
admin.site.register(Branch)  
admin.site.register(Student)  
admin.site.register(Faculty)  
admin.site.register(StudentAttendance)  
admin.site.register(FacultyAttendance)
```



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvce.edu.in](mailto:it@intugvce.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Database integration
7. Date of Experiment : 07-03-2025
- 8.Date of Submission of Report :27-03-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

## Database Integration:

A database is an organized collection of data that is stored and managed in a way that makes it easy to retrieve, update, and manage information. Databases are used to store data in a structured format, allowing efficient access, management, and modification.

### Step 1: Check if SQLite3 is Already Installed

sqlite3 --version

Step 2: Install SQLite3 (if not already installed)

#### On Windows:

1. Download the SQLite3 command-line tool from the official website:  
SQLite Downloads
2. Download the **"Precompiled Binaries for Windows"** (usually a ZIP file).
3. Extract the ZIP file and place sqlite3.exe in a folder (e.g., C:\sqlite).
4. Add the folder to your system's **PATH** environment variable:  
Search for "Environment Variables" in the Start menu.  
Click on "Environment Variables."  
In "System variables," select "Path" and click "Edit."  
Add the folder path (e.g., C:\sqlite) and click OK.

On macOS:

sqlite3 --version

### Step 4: Using SQLite3 with Django

Since Django uses SQLite3 as the default database, you don't need to install any additional drivers. Just make sure your settings.py file has the following configuration:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

### Step 5: Run Migrations to Create the Database

After setting up, run:

python manage.py migrate

### Step 1: Open the Django DB Shell

Make sure your virtual environment is activated, then use the following command:

```
python manage.py db shell
```

**Step 2: Common SQLite Commands**

Once you're inside the SQLite shell, you can use the following commands:

List All Tables:

```
.tables
```

**Step3: View Table Schema:**

```
.schema table_name;
```

**Step4: Show All Data in a Table:**

```
SELECT * FROM table_name;
```

Exit the SQLite Shell:

```
.exit
```



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Deploying django applications on colud platforms
7. Date of Experiment : 27-03-2025
- 8.Date of Submission of Report :04-04-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams,Architecture, workflow,Flowchart   | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

DJANGO FRAMEWORK II B-Tech II semester 2025

## **Deploying Django Web Application on Cloud**

### **What is Deployment?**

Deployment is the process of making a Django web application live on the internet so users can access it. This involves hosting your app on a cloud server like AWS, Google Cloud, Digital Ocean, Heroku, or PythonAnywhere.

### **Features:**

Scalability – Handle more users without performance issues. Security – Protect user data with SSL and secure databases. Global Accessibility – Users can access your app from anywhere.

Continuous Deployment – Easily update your app with new features.

Here's a step-by-step guide to Register on GitHub, Create a Django website with login and registration pages, and Configure Django to handle static files.

### **Step 1: Register on GitHub**

1. Go to [GitHub](https://github.com) and click Sign up.
2. Enter your Username, Email, and Password.
3. Complete the verification and click Create Account.
4. Verify your email by clicking the link in your inbox.

### **Step 2: Push to GitHub**

Initialize Git in your project: `git init`

2.Connect to GitHub:

`git remote add origin`

Add and commit changes: `git add .`

`git commit -m "Initial Commit: Login and Registration App"`

3.Push to GitHub:

`git branch -M main`

`git push -u origin main`

You have successfully built a Django website with login, registration, and static file management.

Your code is now available on GitHub.

### **GITHUB LINK:**

<https://github.com/sahithya-behara/UniversityAttendanceSystem>



**DEPARTMENT OF INFORMATION TECHNOLOGY  
JNTU-GURAJADA VIZIANAGARAM  
COLLEGE OF ENGINEERING VIZIANAGARAM (A)  
VIZIANAGARAM**

**Dr.Ch. Bindu Madhuri**  
Asst. Professor & HOD

Email: hod. [it@intugvcev.edu.in](mailto:it@intugvcev.edu.in)

1. Name of the Laboratory : Django Framework lab
2. Name of the Student :B.Sahithya
3. Roll No :23VV1A1205
4. Class : II Btech II Semester
5. Academic Year :2024-25
6. Name of Experiment : Front end web developer certifications
7. Date of Experiment : 04-04-2025
- 8.Date of Submission of Report :04-04-2025

| S.No | Ability And Activity  | Weightage Of Marks | Day To Day Evaluation Score |
|------|---|--------------------|-----------------------------|
| 1    | Aim Objective, Tools required   | 3                  |                             |
| 2    | Theory, Algorithm and Observations  | 3                  |                             |
| 3    | Implementation  | 3                  |                             |
| 4    | Schematic diagrams, Architecture, workflow, Flowchart                                       | 3                  |                             |
| 5    | Tidiness of his/her working area, proper maintenance of system during and after experiment. | 3                  |                             |
|      | Total Score   | 15                 |                             |

**DATE:**

**Signature of Faculty:**

# FRONT END WEB DEVELOPER CERTIFICATES

## HTML CERTIFICATE



Navigate your next

### COURSE COMPLETION CERTIFICATE

The certificate is awarded to

**Sahithya Behara**

for successfully completing the course

**HTML5 - The Language**

on March 1, 2025



Issued on: Saturday, March 1, 2025  
To verify, scan the QR code at <https://verify.onwingspan.com>



*Congratulations! You make us proud!*



Thirumala Arohi  
Executive Vice President and Global Head  
Education, Training & Assessment (ETA)  
Infosys Limited

## CSS CERTIFICATE



Navigate your next

### COURSE COMPLETION CERTIFICATE

The certificate is awarded to

**Sahithya Behara**

for successfully completing the course

**CSS3**

on March 6, 2025



Issued on: Thursday, March 6, 2025  
To verify, scan the QR code at <https://verify.onwingspan.com>



*Congratulations! You make us proud!*



Thirumala Arohi  
Executive Vice President and Global Head  
Education, Training & Assessment (ETA)  
Infosys Limited



## **JAVA SCRIPT CERTIFICATE**



### COURSE COMPLETION CERTIFICATE

The certificate is awarded to

**Sahithya Behara**

for successfully completing the course

**JavaScript**

on March 6, 2025



*Congratulations! You make us proud!*

Thirumala Arohi  
Executive Vice President and Global Head

## **FRONT END WEB DEVELOPER CERTIFICATE**



### CERTIFICATE OF ACHIEVEMENT

The certificate is awarded to

**Behara Sahithya**

for successfully completing

**Front End Web Developer Certification**

on March 6, 2025



*Congratulations! You make us proud!*

Thirumala Arohi  
Executive Vice President and Global Head  
Education, Training & Assessment (ETA)  
Infosys Limited

Issued on: Thursday, March 6, 2025  
To verify, scan the QR code at <https://verify.onwingspan.com>