

LAB 5

Name: Ayanika Paul

Batch:D1

Roll no. 22

Reg_no : 220905128

SOLVED:

Q1) Build Your First Web Application in Django

myproject/settings.py

CODE:

```
from pathlib import Path
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
SECRET_KEY = 'django-insecure-k%dwvv4t1_2&2gm(1lh-sm*)d3q4ll7m%w2@_7aa#j+&^47hf'
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
INSTALLED_APPS = [
```

```
    'django.contrib.admin',
```

```
    'django.contrib.auth',
```

```
    'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
```

```
    'django.contrib.messages',
```

```
    'django.contrib.staticfiles',
```

```
]
```

```
MIDDLEWARE = [
```

```
    'django.middleware.security.SecurityMiddleware',
```

```
    'django.contrib.sessions.middleware.SessionMiddleware',
```

```
    'django.middleware.common.CommonMiddleware',
```

```
    'django.middleware.csrf.CsrfViewMiddleware',
```

```
    'django.contrib.auth.middleware.AuthenticationMiddleware',
```

```
    'django.contrib.messages.middleware.MessageMiddleware',
```

```
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'q1.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
WSGI_APPLICATION = 'q1.wsgi.application'
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
```

```

    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

myproject/urls.py

CODE:

```

from django.contrib import admin

from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('webapp.urls')), # Add this line to include webapp.urls for the root URL
]

```

webapp/urls.py

CODE:

```

from django.urls import path # Import path from django.urls

from . import views # Assuming you have views to handle the URLs

urlpatterns = [
    path("", views.index, name='index'), # Replace `url()` with `path()`
]

```

webapp/views.py

CODE:

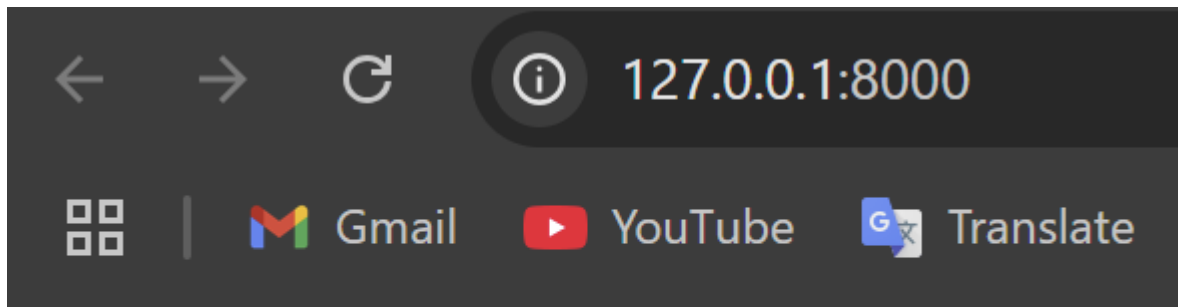
```
from django.shortcuts import render

from django.http import HttpResponse

def index(request):

    return HttpResponse("<h2>HEY! Welcome to Edureka!</h2>")
```

OUTPUT:



HEY! Welcome to Edureka!

2. Displaying a Template

myproject/settings.py

CODE:

```
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-k%dwvv4t1_2&2gm(1lhf-sm*)d3q4ll7m%w2@_7aa#j+&^47hf'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',
```

```
'django.contrib.messages',
'django.contrib.staticfiles',
]
MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'q1.urls'
TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]
WSGI_APPLICATION = 'q1.wsgi.application'
DATABASES = {
'default': {
```

```

        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
STATIC_URL = 'static/'
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

[webapp/views.py](#)

CODE:

```

from django.shortcuts import render
from django.http import HttpResponse
from datetime import date
import calendar

```

```

from calendar import HTMLCalendar

def index(request, year=None, month=None):
    if year is None or month is None:
        today = date.today()
        year = today.year
        month = today.month
    year = int(year)
    month = int(month)
    if year < 1900 or year > 2099:
        year = date.today().year
    month_name = calendar.month_name[month]
    title = f"MyClub Event Calendar - {month_name} {year}"
    cal = HTMLCalendar().formatmonth(year, month)
    return HttpResponse(f"<h1>{title}</h1><p>{cal}</p>")

```

webapp/urls.py

CODE:

```

from django.urls import path, re_path
from . import views

urlpatterns = [
    path("", views.index, name='index'),
    re_path(r'^(?P<year>[0-9]{4})/(?P<month>(0?[1-9]|1[0-2]))/$', views.index, name='index'),
]

```

myproject/urls.py

CODE:

```

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),

```

```
path("", include('webapp.urls')), # Add this line to include webapp.urls for the root URL
]
```

OUTPUT:

MyClub Event Calendar - February 2025

February 2025						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

LAB Q

Q1) Create a web form which allows the teacher to enter the student details like name, date of birth, address, contact number, email id and marks of English, Physics and Chemistry. After filling all the information and on clicking submit button, details should be added to a textarea displayed on the same page. Display the total percentage of marks obtained in a label.

CODE:

[q1/settings.py](#)

```
from pathlib import Path
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
SECRET_KEY = 'django-insecure-m9rls0zebd*%5j+y)8xzf#st#%ks#vq!&sq5d74&y5@uim)zxcg'
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
INSTALLED_APPS = [
```

```
    'formapp',
```

```
    'django.contrib.admin',
```

```
    'django.contrib.auth',
```

```
    'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
```

```
    'django.contrib.messages',
```



```
'django.contrib.staticfiles',
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'q1.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'q1.wsgi.application'
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
```

```

        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
STATIC_URL = 'static/'
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

q1/urls.py

```

from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('formapp.urls')),
]

```

formapp/urls.py

```
from django.urls import path

from . import views

urlpatterns = [

    path("", views.student_form, name='student_form'), ]
```

formapp/views.py

```
from django.shortcuts import render

from .forms import StudentForm

from .models import Student

def student_form(request):

    students = Student.objects.all()

    form = StudentForm()

    if request.method == 'POST':

        form = StudentForm(request.POST)

        if form.is_valid():

            form.save()

            form = StudentForm() # Reset form after submission

    return render(request, 'student_form.html', {'form': form, 'students': students})
```

formapp/views.py

CODE:

```
from django.shortcuts import render

from .forms import StudentForm

from .models import Student

def student_form(request):

    students = Student.objects.all()

    form = StudentForm()

    if request.method == 'POST':

        form = StudentForm(request.POST)

        if form.is_valid():

            form.save()
```

```
form = StudentForm() # Reset form after submission  
return render(request, 'student_form.html', {'form': form, 'students': students})
```

formapp/templates/Student_form.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Student Details Form</title>  
  <style>  
    body {  
      font-family: Arial, sans-serif;  
      margin: 20px;  
      padding: 20px;  
      background-color: #f4f4f4;  
    }  
    .container {  
      width: 60%;  
      background: white;  
      padding: 33px;  
      border-radius: 8px;  
      box-shadow: 0px 0px 10px gray;  
    }  
    label {  
      font-weight: bold;  
    }  
    input, textarea {  
      width: 100%;  
      padding: 8px;  
      margin: 5px 0 15px;
```

```
        border: 1px solid #ccc;
        border-radius: 4px;
    }
    button {
        background: #28a745;
        color: white;
        padding: 10px;
        border: none;
        cursor: pointer;
        width: 100%;
    }
    button:hover {
        background: #218838;
    }
    textarea {
        width: 100%;
        height: 150px;
        margin-top: 15px;
    }
    .result {
        font-size: 18px;
        font-weight: bold;
        color: #333;
    }
</style>
</head>
<body>
<div class="container">
    <h2>Student Details Form</h2>
    <form method="POST">
        {% csrf_token %}
```

```
        {{ form.as_p }}

        <button type="submit">Submit</button>

    </form>

    <h3>Student Records:</h3>

    <textarea readonly>

{% for student in students %}

    Name: {{ student.name }}

    DOB: {{ student.dob }}

    Address: {{ student.address }}

    Contact: {{ student.contact }}

    Email: {{ student.email }}

    Marks - English: {{ student.english }}, Physics: {{ student.physics }}, Chemistry: {{ student.chemistry }}

    Total Percentage: {{ student.percentage }}%

    -----

{% endfor %}

    </textarea>

</div>

</body>

</html>
```

OUTPUT:

Student Details Form

Name:

Ayushi Das

Dob:

12/09/2004

Address:

Delhi

Contact:

863056789

Email:

abc@gmail.com

English:

90

Physics:

77

Chemistry:

67

Submit

Student Records:

DOB: Aug. 12, 2003
Address: Mit Gate 2, Eshwar Nagar, Manipal, Udupi
Contact: 8822872261
Email: ayanika2a18@gmail.com
Marks - English: 90, Physics: 87, Chemistry: 92
Total Percentage: 89.67%

Name: Rithik Das
DOB: Dec. 8, 2004
Address: Manipal,Udupi
Contact: 8833278619
Email: xyz@gmail.com
Marks - English: 80, Physics: 90, Chemistry: 94
Total Percentage: 88.0%

Student Records:

Name: Ayanika Paul
DOB: Aug. 12, 2003
Address: Mit Gate 2, Eshwar Nagar, Manipal, Udupi
Contact: 8822872261
Email: ayanika2a18@gmail.com
Marks - English: 90, Physics: 87, Chemistry: 92
Total Percentage: 89.67%

Name: Rithik Das
DOB: Dec. 8, 2004
Address: Manipal,Udupi
Contact: 8833278619
Email: xyz@gmail.com
Marks - English: 80, Physics: 90, Chemistry: 94
Total Percentage: 88.0%

Name: Ayushi Das
DOB: Dec. 9, 2004
Address: Delhi
Contact: 863056789
Email: abc@gmail.com
Marks - English: 90, Physics: 77, Chemistry: 67

Q2) Create a web form with employee ids in dropdown list. Use a textbox for accepting date of joining. Add a button named “Am I Eligible for Promotion”. On clicking the button, if he has more than 5 years of experience, then display “YES” else “NO” in the label control.

q2/settings.py

CODE:

from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

```
SECRET_KEY = 'django-insecure-(tcd%&4g!vc(^r9t=shhpwhfacf=287x2=gntd17f3lm)_weov'
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
INSTALLED_APPS = [
```

```
    'employeeapp',
```

```
    'django.contrib.admin',
```

```
    'django.contrib.auth',
```

```
    'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
```

```
    'django.contrib.messages',
```

```
    'django.contrib.staticfiles',
```

```
]
```

```
MIDDLEWARE = [
```

```
    'django.middleware.security.SecurityMiddleware',
```

```
    'django.contrib.sessions.middleware.SessionMiddleware',
```

```
    'django.middleware.common.CommonMiddleware',
```

```
    'django.middleware.csrf.CsrfViewMiddleware',
```

```
    'django.contrib.auth.middleware.AuthenticationMiddleware',
```

```
    'django.contrib.messages.middleware.MessageMiddleware',
```

```
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
```

```
]
```

```
ROOT_URLCONF = 'q2.urls'
```

```
TEMPLATES = [
```

```
{
```

```
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
```

```
    'DIRS': [
```

```
        BASE_DIR / 'employeeapp/templates', # Add this line to include the correct path
```

```
    ],
```

```
    'APP_DIRS': True,
```

```
    'OPTIONS': {
```

```
        'context_processors': [
```



```
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
},
]
WSGI_APPLICATION = 'q2.wsgi.application'
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
```

```
USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

q2/urls.py

CODE:

```
from django.contrib import admin

from django.urls import path, include

from employeeapp import views # Import views from your app

urlpatterns = [

    path('admin/', admin.site.urls),

    path('employee/', include('employeeapp.urls')),

    path("", views.check_promotion, name='home'), # Serve the check_promotion page at the root

]
```

employeeapp/urls.py

CODE:

```
from django.urls import path

from . import views

urlpatterns = [

    path('check_promotion/', views.check_promotion, name='check_promotion'),

]
```

employeeapp/views.py

CODE:

```
from django.shortcuts import render

from datetime import datetime

def check_promotion(request):

    error_message = None # Ensure the variable is always defined

    eligible = None # Ensure eligible is always defined

    if request.method == 'POST':
```

```

employee_id = request.POST.get('employee_id')

joining_date_str = request.POST.get('joining_date')

if not joining_date_str: # If the joining date is missing
    error_message = "Please provide the date of joining."
else:
    try:
        joining_date = datetime.strptime(joining_date_str, '%Y-%m-%d')
        experience = (datetime.now() - joining_date).days // 365
        eligible = experience >= 5 # True if 5 or more years of experience
    except ValueError:
        error_message = "Invalid date format. Please enter a valid date."

    return render(request, 'check_promotion.html', {'eligible': eligible, 'error_message':
error_message})

```

employeeapp/templates/check_promotion.html

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Check Promotion Eligibility</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

            margin: 0;

        }
    
```

```
.container {  
    background: white;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  
    width: 350px;  
    text-align: center;  
}  
  
h2 {  
    color: #333;  
    margin-bottom: 20px;  
}  
  
form {  
    display: flex;  
    flex-direction: column;  
}  
  
label {  
    text-align: left;  
    font-weight: bold;  
    margin-top: 10px;  
}  
  
select, input {  
    padding: 8px;  
    margin-top: 5px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    width: 97%;  
}  
  
button {  
    background-color: #28a745;  
    color: white;
```

```
padding: 10px;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

margin-top: 15px;
}

button:hover {

background-color: #218838;

}

.result {

font-size: 18px;

font-weight: bold;

margin-top: 15px;

color: #333;

}

.error {

color: red;

font-size: 14px;

margin-top: 10px;

}

</style>

</head>

<body>

<div class="container">

<h2>Check Promotion Eligibility</h2>

<form method="POST">

{% csrf_token %}

<label for="employee_id">Employee ID:</label>

<select id="employee_id" name="employee_id" required>

<option value="">Select Employee</option>
```

```
<option value="1">Employee 1</option>
<option value="2">Employee 2</option>
<option value="3">Employee 3</option>
</select>
<label for="joining_date">Date of Joining:</label>
<input type="date" id="joining_date" name="joining_date" required>
<button type="submit">Am I Eligible for Promotion?</button>
</form>
{% if error_message %}
    <p class="error">{{ error_message }}</p>
{% endif %}
{% if eligible is not none %}
    <p class="result">Eligibility for promotion:
        <span style="color: {% if eligible %}green{% else %}red{% endif %};">
            {% if eligible %}YES{% else %}NO{% endif %}
        </span>
    </p>
{% endif %}
</div>
</body>
</html>
```

OUTPUT:

Check Promotion Eligibility

Employee ID:

Employee 1

Date of Joining:

29 - 06 - 2023

Am I Eligible for Promotion?

Eligibility for promotion: **NO**

Check Promotion Eligibility

Employee ID:

Employee 2

Date of Joining:

05 - 01 - 2020

Am I Eligible for Promotion?

Eligibility for promotion: **YES**

