VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JnanaSangama, Belagavi - 590 018, Karnataka

Acharya Institute of Technology

Acharya Dr Sarvepalli. Radhakrishnan Road

Acharya P.O Soladevanahalli, Bengaluru-560107

LABORATORY MANUAL

FULLSTACK DEVELOPMENT

21CS62

VI Semester

Prepared by:

1. Mr. Vinayak Raju Kage

2. Mr. Gowtham Raj

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA)

Table of contents

SL	Name of Program	Page No
1	Installation of Python, Django and Visual Studio code editors can be demonstrated.	
2	Creation of virtual environment, Django project and App should be demonstrated	
3	Develop a Django app that displays current date and time in server	
4	Develop a Django app that displays date and time four hours ahead and four hours before as an offset of current date and time in server.	
5	Develop a simple Django app that displays an unordered list of fruits and ordered list of selected students for an event	
6	Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages contact us, About Us and Home page of any website.	
7	Develop a Django app that performs student registration to a course. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field.	
8	For student and course models created in Lab experiment for Module2, register admin interfaces, perform migrations and illustrate data entry through admin forms.	
9	Develop a Model form for student that contains his topic chosen for project, languages used and duration with a model called project.	
10	For students enrolment developed in Module 2, create a generic class view which displays list of students and detailview that displays student details for any selected student in the list.	
11	Develop example Django app that performs CSV and PDF generation for any models created in previous laboratory component.	

12	Develop a registration page for student enrolment as done in Module 2 but without page refresh using AJAX.	
13	Develop a search application in Django using AJAX that displays courses enrolled by a student being searched.	

Laboratory Experiments:

1. Installation of Python, Django and Visual Studio code editors can be demonstrated
2. Creation of virtual environment, Django project and App should be demonstrated
3. Develop a Django app that displays current date and time in server
4. Develop a Django app that displays date and time four hours ahead and four hours before as an offset of current date and time in server.
5. Develop a simple Django app that displays an unordered list of fruits and ordered list of selected students for an event
6. Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages contact us, About Us and Home page of any website
7. Develop a Django app that performs student registration to a course. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field
8. For student and course models created in Lab experiment for Module2, register admin interfaces, perform migrations and illustrate data entry through admin forms.
9. Develop a Model form for student that contains his topic chosen for project, languages used and duration with a model called project
10. For students enrolment developed in Module 2, create a generic class view which displays list of students and detail view that displays student details for any selected student in the list.
11. Develop example Django app that performs CSV and PDF generation for any models created in previous laboratory component.
12. Develop a registration page for student enrolment as done in Module 2 but without page refresh using AJAX.
13. Develop a search application in Django using AJAX that displays courses enrolled by a student being searched.

Evaluation Rubrics for lab Programs (Max marks 20)

A. Lab write-up and execution rubrics(Max marks 8)

		Good	**Average**
a.	**Understanding of problem (3 marks)**	Demonstrate goodknowledge of language constructs and programming practice (3)	Moderate understanding of language constructs (1)
b.	**Execution and testing (3marks)**	Program handles all possible conditions and results with satisfying results. (3)	Partial executions /poor error handling (1)
c.	**Result and documentation (2 marks)**	Meticulousdocumentation of changes made and results obtained are in proper format (2)	Moderate formatting of output and average documentation (1)

B. VIVA Rubrics: (Max marks 2)

		Good	**Average**
	Conceptual understanding (2 marks)	Explain the complete program with the related concepts.(5)	Adequately provides explanation.(3)

C. Marks for Lab Record:10

1.Installation of Python, Django and Visual Studio code editors can be demonstrated.

Python download link https://www.python.org/downloads/
To check out python installed version **py –version**
VScode **https://code.visualstudio.com/download**
Django installation commands to be typed in cmd

- py -m pip install Django

  or

  `py -m pip install Django==5.0.4 #version`

2. Creation of virtual environment, Django project and App should be demonstrated

- Create virtual environment!

```
# Install virtualenv if you haven't already
py -m pip install --user virtualenv

# Create a new virtual environment
py -m venv myenv (myenv is virtual environment name)

# Activate the virtual environment
python manage.py runserver
```

- **Installing Django**

  Once the virtual environment is activated, you can install Django

```
python -m pip install --upgrade pip
 # Install Django
      pip install Django
```

- **Creating a Django Project**

```
# Navigate to your desired directory
cd /path/to/your/directory
# Create a new Django project
django-admin startproject myproject
```

- **Running the Django Development Server**

```
# Run the development server
python manage.py runserver
```

- **Creating a Django App**

```
# Ensure you are in the project directory
cd /path/to/your/directory/myproject

# Create a new app called 'myapp'
python manage.py startapp myapp
```

- **Running Migrations**

```
python manage.py makemigrations
python manage.py migrate
```

- **Creating Superuser (Admin User)**

```
python manage.py createsuperuser
python manage.py help
python manage.py runserver
```

3. develop a Django app that displays current date and time in server

To create environment → python -m venv myenv
Activate the environment → .\myenv\Scripts\activate

#Instal Django → pip install django #if u created environment newly

---

Create a new Django project:
django-admin startproject time_project
cd time_project

Create a new Django app
python manage.py startapp time_app

---

time_app/views

```
from django.http import HttpResponse
from datetime import datetime
def current_datetime(request):
    now = datetime.now()
    html = "<html><body>Current date and time: {}</body></html>".format(now)
    return HttpResponse(html)
```

time_project/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('time/', include('time_app.urls')),
]
```

#Create a new file called urls.py inside the time_app directory

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

to run

```
Python manage.py runserver
```

4. Develop a Django app that displays date and time four hours ahead and four hours before as an offset of current date and time in server.

```
django-admin startproject time_project
cd time_project
```

Create a new Django app

```
python manage.py startapp time_app
```

time_app/views

```
from django.http import HttpResponse
from datetime import datetime
def current_datetime(request):
    now1 = datetime.now() - timedelta(hours=4)
    now2 = datetime.now() + timedelta(hours=4)
    html = "<html><body>Current date and time 4hrs ago: {}<br>.format(now1)
    Current date and time 4hrs after: <br>{}</body></html>".format(now2)

    return HttpResponse(html)
```

time_project/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('time/', include('time_app.urls')),
]
```

#Create a new file called urls.py inside the time_app directory

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

to run

```
Python manage.py runserver
```

5. Develop a simple django app that displays an unordered list of fruits and ordered list of selected students for an event

```
django-admin startproject listapp
cd listapp
```

Create a new Django app

```
python manage.py startapp lists
```

list/model.py

```
from django.db import models
class Fruit(models.Model):
    name = models.CharField(max_length=100)
    def __str__(self):
        return self.name
class Student(models.Model):
    name = models.CharField(max_length=100)
    def __str__(self):
        return self.name
```

list/admin.py

```
from django.contrib import admin
from .models import Fruit, Student
admin.site.register(Fruit)
admin.site.register(Student)
```

list/views.py

```
from django.shortcuts import render
from .models import Fruit, Student
def fruit_list(request):
    fruits = Fruit.objects.all()
    return render(request, 'lists/fruit_list.html', {'fruits': fruits})
def student_list(request):
    students = Student.objects.all()
    return render(request, 'lists/student_list.html', {'students': students})
```

lists/templates/lists/fruit_list.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Fruit List</title>
</head>
<body>
    <h1>Unordered List of Fruits</h1>
    <ul>
```

```
        {% for fruit in fruits %}
            <li>{{ fruit.name }}</li>
        {% endfor %}
    </ul>
</body>
</html>
```

lists/templates/lists/student_list.html

```
<!DOCTYPE html>

<html>
<head>
    <title>Student List</title>
</head>
<body>
    <h1>Ordered List of Selected Students</h1>
    <ol>
        {% for student in students %}
            <li>{{ student.name }}</li>
        {% endfor %}
    </ol>
</body>
</html>
```

List/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('fruits/', views.fruit_list, name='fruit_list'),
    path('students/', views.student_list, name='student_list'),
]
```

Listsapp/urls.py

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

listapp/settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'lists',  # Add your app name here
]
```

Migration and database changes

```
python manage.py makemigrations
python manage.py migrate
```

create user in command mode

```
python manage.py createsuperuser
```

restart or run server

```
python manage.py runserver
```

in browser

http://127.0.0.1:8000/lists/students/
http://127.0.0.1:8000/lists/fruits/

6. Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages contact us, About Us and Home page of any website

```
django-admin startproject mywebsite
cd mywebsite

python manage.py startapp pages
```

pages/urls.py

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

urlpatterns = [
    path('', views.home, name='home'),
    path('about/', views.about, name='about'),
    path('contact/', views.contact, name='contact'),
]
```

Pages/views.py

```
from django.shortcuts import render

def home(request):
    return render(request, 'home.html')

def about(request):
    return render(request, 'about.html')

def contact(request):
    return render(request, 'contact.html')
```

pages/templates/layout.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}My Website{% endblock %}</title>
    <style>
        /* Basic CSS for layout */
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }
        header {
            background-color: #333;
            color: #fff;
            padding: 10px 0;
            text-align: center;
        }
        nav {
            display: flex;
            justify-content: center;
        }
        nav a {
            color: #fff;
            text-decoration: none;
            padding: 10px 20px;
        }
        nav a:hover {
            background-color: #555;
        }
        footer {
            background-color: #333;
            color: #fff;
            text-align: center;
            padding: 10px 0;
            position: absolute;
            bottom: 0;
            width: 100%;
        }
    </style>
</head>
<body>
    <header>
        <h1>My Website</h1>
        <nav>
            <a href="{% url 'home' %}">Home</a>
            <a href="{% url 'about' %}">About Us</a>
            <a href="{% url 'contact' %}">Contact Us</a>
        </nav>
    </header>

    <div id="content">
        {% block content %}
        <!-- This block will be overridden by specific content of each page -->
        {% endblock %}
    </div>
    <footer>
        <p>&copy; {% now 'Y' %} My Website | Developed by Your Name</p>
    </footer>
</body>
</html>
```

pages/templates/home.html

```
{% extends 'layout.html' %}

{% block title %}Home - My Website{% endblock %}

{% block content %}
   <h2>Welcome to Our Home Page!</h2>
   <p>This is the Home Page content of My Website.</p>
   <!-- Add more content specific to the home page here -->
{% endblock %}
```

pages/templates/about.html

```
{% extends 'layout.html' %}
{% block title %}About Us - My Website{% endblock %}
{% block content %}
   <h2>About Us</h2>
   <p>Learn more about our company or organization.</p>
   <!-- Add more content specific to the about page here -->
{% endblock %}
```

pages/templates/contact.html

```
{% extends 'layout.html' %}
{% block title %}Contact Us - My Website{% endblock %}
{% block content %}
   <h2>Contact Us</h2>
   <p>Get in touch with us using the form below or via other contact information.</p>
   <!-- Add more content specific to the contact page here -->
{% endblock %}
```

Mywebsite/urls.py

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

urlpatterns = [
   path('admin/', admin.site.urls),
   path('', include('pages.urls')),  # Include the URLs from the pages app
]
```

mywebsite/settings.py

```
add--- import os
'DIRS': [os.path.join(BASE_DIR, 'pages/templates')], #replacet this in templates
```

To run python manage.py runserver
http://127.0.0.1:8000/home

7. Develop a Django app that performs student registration to a course. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field.

To create environment → python -m venv myenv
Activate the environment → .\myenv\Scripts\activate

#Instal Django → if u created environment newly

Create a new Django project:

```
django-admin startproject student_registration
cd student_registration
```

Create a new Django app

```
python manage.py startapp registration
```

Add the new app to your project's settings:
In student_registration/settings.py, add 'registration' to the INSTALLED_APPS

```
INSTALLED_APPS = [
   ...
   'registration',
]
```

Create the models in registration/models.py:

```
from django.db import models

class Student(models.Model):
   first_name = models.CharField(max_length=50)
   last_name = models.CharField(max_length=50)
   email = models.EmailField(unique=True)

   def __str__(self):
      return f'{self.first_name} {self.last_name}'

class Course(models.Model):
   name = models.CharField(max_length=100)
   description = models.TextField()

   def __str__(self):
      return self.name

class Enrollment(models.Model):
   student = models.ForeignKey(Student, on_delete=models.CASCADE)
   course = models.ForeignKey(Course, on_delete=models.CASCADE)
```

```
    enrolled_at = models.DateTimeField(auto_now_add=True)

    class Meta:
        unique_together = ('student', 'course')

    def __str__(self):
        return f'{self.student} enrolled in {self.course}'
```

Run migrations to create the database schema:

```
python manage.py makemigrations
python manage.py migrate
```

---

admin interface registration/admin.py

```
from django.contrib import admin
from .models import Student, Course, Enrollment
admin.site.register(Student)
admin.site.register(Course)
admin.site.register(Enrollment)
```

Create views for registration and displaying enrolled students in registration/views.py:

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Student, Course, Enrollment
from django.http import HttpResponse

def course_list(request):
    courses = Course.objects.all()
    return render(request, 'registration/course_list.html', {'courses': courses})

def enroll_student(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    students = Student.objects.all()

    if request.method == 'POST':
        student_id = request.POST.get('student')
        student = get_object_or_404(Student, id=student_id)
        Enrollment.objects.create(student=student, course=course)
        return redirect('course_list')

    return render(request, 'registration/enroll_student.html', {'course': course, 'students': students})

def enrolled_students(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    enrollments = Enrollment.objects.filter(course=course)
    return render(request, 'registration/enrolled_students.html', {'course': course, 'enrollments':
enrollments})
```

Create URLs for these views in registration/urls.py:

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

urlpatterns = [
    path('', views.course_list, name='course_list'),
    path('course/<int:course_id>/enroll/', views.enroll_student, name='enroll_student'),
    path('course/<int:course_id>/students/', views.enrolled_students, name='enrolled_students'),
]
```

Project folder student_registration/urls.py

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

Create templates for each view:

registration/templates/registration/course_list.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Courses</title>
</head>
<body>
    <h1>Courses</h1>
    <ul>
        {% for course in courses %}
            <li>
                {{ course.name }} - <a href="{% url 'enroll_student' course.id %}">Enroll</a> -
                <a href="{% url 'enrolled_students' course.id %}">View Students</a>
            </li>
        {% endfor %}
    </ul>
</body>
</html>
```

registration/templates/registration/enroll_student.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Enroll Student</title>
</head>
<body>
   <h1>Enroll Student in {{ course.name }}</h1>
   <form method="post">
      {% csrf_token %}
      <select name="student">
         {% for student in students %}
            <option value="{{ student.id }}">{{ student.first_name }} {{ student.last_name
}}</option>
         {% endfor %}
      </select>
      <button type="submit">Enroll</button>
   </form>
</body>
</html>
```

registration/templates/registration/enrolled_students.html:

```
<!DOCTYPE html>
<html>
<head>
   <title>Enrolled Students</title>
</head>
<body>
   <h1>Students Enrolled in {{ course.name }}</h1>
   <ul>
      {% for enrollment in enrollments %}
         <li>{{ enrollment.student.first_name }} {{ enrollment.student.last_name }}</li>
      {% endfor %}
   </ul>
</body>
</html>
```

To create superuser

```
Python manage.py createsuperuser
```

To run
Python manage.py runserver

Ouput link → http://127.0.0.1:8000/admin/ add student name, course, enrolment

Test app → http://127.0.0.1:8000/registration/
