Full Stack Development in Django

1. (Not there for lab test)

2. Create a Django application to demonstrate Cross-Site Scripting (XSS) vulnerability and its prevention.

The application should:

- Provide an input form where the user can enter their name.
- On submission, display the entered name back to the user.
- Demonstrate how XSS can occur if malicious code such as <script>alert("Hacked")</script> is entered in the input field.
- Modify the application to sanitize the input (e.g., by removing or escaping the <script> tag) and re-display the cleaned output safely.

urls.py

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

urlpatterns = [
    path('', views.index, name='index'),
    path('nameRef', views.nameRef, name='nameRef'),
]
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse

def index(request: HttpRequest):
    return render(request, 'index.html')

def nameRef(request: HttpRequest):
    if request.method == 'POST':
        name = request.POST.get('name')
        if not name:
            return HttpResponse("Please provide a name.")
        name = name.lower()
        if 'script' in name:
            name = name.replace('script', '')
        return HttpResponse(f"Hello, {name}!")
    else:
        return HttpResponse("Invalid request method.")
```

templates/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>XSS Project</title>
</head>
<body>
```

```
    <h1>Cross Site Scripting (XSS) Project</h1>
    <form method="POST" action="/nameRef">
        {% csrf_token %}
        <label for="name">Name:</label>
        <input type="text" id="name" name="name">
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

3. Create a Django web application that fetches and displays live weather information for a given city using the OpenWeatherMap API.

The application should:

- Provide an input form where the user can enter a city name.
- On form submission, make an API request to OpenWeatherMap to retrieve the current weather.
- Extract and display the following details:
    1. City name
    2. Current temperature (in °C)
    3. Weather description (e.g., Clear Sky)
- If the user enters an invalid city name, display a suitable error message.

urls.py

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

urlpatterns = [
    path('', views.index, name='index'),
]
```

views.py

```
import requests
from django.shortcuts import render

def index(request):
    city = request.GET.get('city', 'Tumakuru')
    api_key = ""
    url =
f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&units=metric"
    response = requests.get(url)
    data = response.json()
    if response.status_code == 200 and "main" in data:
        weather_data = {
            "city": city,
            "temperature": data["main"]["temp"],
            "description": data["weather"][0]["description"].title(),
            "icon": data["weather"][0]["icon"],
        }
    else:
        weather_data = {"error": "City not found"}
    return render(request, "index.html", {"weather": weather_data})
```

templates/index.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Weather App</title>
</head>
<body>
    <h1>Weather App</h1>
    <form method="get">
        <input type="text" name="city" placeholder="Enter city" required>
        <button type="submit">Search</button>
    </form>
    {% if weather.error %}
    <p style="color:red;">{{ weather.error }}</p>
    {% else %}
    <h2>{{ weather.city }}</h2>
    <p>Temperature: {{ weather.temperature }}°C</p>
    <p>Description: {{ weather.description }}</p>
    <img src="http://openweathermap.org/img/w/{{ weather.icon }}.png" alt="Weather
icon">
    {% endif %}
</body>
</html>
```

4. Create a Django application that displays a biodata form. The form should collect user details (Name, Age, Email, Phone, Address) and, upon submission, display the entered information on a new webpage.

urls.py

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

urlpatterns = [
    path('', views.biodata_form, name='biodata_form'),
    path('display/', views.display_biodata, name='display_biodata'),
]
```

views.py

```python
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse

def biodata_form(request: HttpRequest):
    if request.method == 'GET':
        return render(request, 'biodata_form.html')
    else:
        return HttpResponse("Invalid request method.")

def display_biodata(request: HttpRequest):
    if request.method == 'POST':
        name = request.POST.get('name', 'N/A')
        age = request.POST.get('age', 'N/A')
        address = request.POST.get('address', 'N/A')
        phone = request.POST.get('phone', 'N/A')
        email = request.POST.get('email', 'N/A')
        context = {
            'name': name,
            'age': age,
            'address': address,
            'phone': phone,
            'email': email,
        }
        return render(request, 'display_biodata.html', context)
    else:
        return HttpResponse("Invalid request method.")
```

templates/biodata_from.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Biodata Form</title>
</head>
<body>
    <form method="post" action="{% url 'display_biodata' %}">
        {% csrf_token %}
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br><br>

        <label for="age">Age:</label>
        <input type="number" id="age" name="age" required><br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>

        <label for="phone">Phone:</label>
        <input type="tel" id="phone" name="phone" required><br><br>

        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

templates/display_biodata.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Display Biodata</title>
</head>
<body>
    <h1>Biodata Details</h1>
    <p><strong>Name:</strong> {{ name }}</p>
    <p><strong>Age:</strong> {{ age }}</p>
    <p><strong>Email:</strong> {{ email }}</p>
    <p><strong>Phone:</strong> {{ phone }}</p>
    <p><strong>Address:</strong> {{ address }}</p>
</body>
</html>
```

5. Create a web page using jQuery and Django that allows students to submit feedback. The application should: Allow students to enter their Name and Feedback. Display submitted feedback dynamically without

reloading the page using AJAX. Use jQuery events, effects, and traversing to enhance interactivity. Highlight input fields on focus and apply alternating background colors for feedback entries.

urls.py

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

urlpatterns = [
    path('', views.feedback_page, name='feedback_page'),
    path('submit/', views.submit_feedback, name='submit_feedback'),
    path('get-feedbacks/', views.get_feedbacks, name='get_feedbacks'),
]
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, JsonResponse
from django.views.decorators.csrf import csrf_exempt

feedback_storage = []

def feedback_page(request: HttpRequest):
    return render(request, 'feedback.html')

def get_feedbacks(request: HttpRequest):
    return JsonResponse({'feedbacks': feedback_storage})

@csrf_exempt
def submit_feedback(request: HttpRequest):
    if request.method == "POST":
        name = request.POST.get('name')
        feedback = request.POST.get('feedback')

        if name and feedback:
            entry = {'name': name, 'feedback': feedback}
            feedback_storage.insert(0, entry)
            return JsonResponse({'status': 'success', 'feedback': entry})
        else:
            return JsonResponse({'status': 'error', 'message': 'Both fields required.'})
    return JsonResponse({'status': 'error', 'message': 'Invalid request method.'})
```

templates/feedback.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Feedback</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <h2>Submit Feedback</h2>
    <form id="feedback-form">
        <input type="text" id="name" name="name" placeholder="Your Name" required><br><br>
        <textarea id="feedback" name="feedback" placeholder="Your Feedback"
required></textarea><br><br>
        <button type="submit">Submit</button>
    </form>
    <div id="message"></div>
    <h3>Feedback Entries</h3>
    <div id="feedback-list"></div>
    <script>
        $(document).ready(function () {
            var colors = ['#ffebcd', '#e0ffe0'];
            var colorIndex = 0;
            function createFeedbackEntry(fb) {
                let entry = $('<div></div>')
                    .text(fb.name + ': ' + fb.feedback)
                    .css('background-color', colors[colorIndex % colors.length])
                    .css('padding', '10px')
                    .css('margin', '5px 0');
                colorIndex++;
                return entry;
            }

            // Load existing feedbacks on page load
            $.ajax({
                url: "/get-feedbacks/",
                method: 'GET',
                success: function (response) {
                    response.feedbacks.forEach(function (fb) {
                        let entry = createFeedbackEntry(fb);
                        $('#feedback-list').append(entry);
                    });
                }
            });

            // Highlight input on focus and remove on blur
            $('#name, #feedback').on('focus', function () {
                $(this).css('background-color', '#ffffcc');
            }).on('blur', function () {
                $(this).css('background-color', '');
            });

            // Submit feedback with AJAX
            $('#feedback-form').submit(function (e) {
                e.preventDefault();
                $('#message').text('');
                let name = $('#name').val().trim();
                let feedback = $('#feedback').val().trim();
                if (!name || !feedback) {
                    $('#message').text('Please fill both fields.');
                    return;
                }
                $.ajax({
                    url: "/submit/",
                    method: 'POST',
                    data: {
                        'name': name,
                        'feedback': feedback,
                        'csrfmiddlewaretoken': '{{ csrf_token }}'
                    },
                    success: function (response) {
                        if (response.status === 'success') {
                            let entry = createFeedbackEntry(response.feedback);
                            $('#feedback-list').prepend(entry);
                            $('#feedback-form')[0].reset();
                            $('#message').text('Feedback submitted.');
                        } else {
                            $('#message').text(response.message);
                        }
                    },
                    error: function () {
                        $('#message').text('Error submitting feedback.');
                    }
                });
            });
        });
    </script>
</body>
</html>
```

6. Develop a Django application to accept USN, Name, Subject Code, and CIE Marks from a web form and store the details in a database.
Display the list of students whose CIE marks are less than 20.

urls.py

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

urlpatterns = [
    path('', views.add_student, name='add_student'),
    path('low_cie_students/', views.low_cie_students, name='low_cie_students'),
]
```

models.py

```
from django.db import models

class Student(models.Model):
    usn = models.CharField(max_length=20, unique=True)
    name = models.CharField(max_length=100)
    subject_code = models.CharField(max_length=10)
    cie_marks = models.IntegerField()
```

forms.py

```
from django.forms import ModelForm
from .models import Student

class StudentForm(ModelForm):
    class Meta:
        model = Student
        fields = '__all__'
```

admin.py

```
from django.contrib import admin
from .models import Student

@admin.register(Student)
class StudentAdmin(admin.ModelAdmin):
    list_display = ('usn', 'name', 'subject_code', 'cie_marks')
    search_fields = ('name', 'usn', 'subject_code')
    list_filter = ('cie_marks',)
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
from .forms import StudentForm
from .models import Student

def add_student(request: HttpRequest):
    if request.method == "POST":
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Student added successfully.")
    else:
        form = StudentForm()
    return render(request, 'add_student.html', {'form': form})

def low_cie_students(request: HttpRequest):
    students = Student.objects.filter(cie_marks__lt=20)
    return render(request, 'low_cie_students.html', {'students': students})
```

templates/add_student.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Student</title>
</head>
<body>
    <h2>Add Student Details</h2>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Save</button>
    </form>
    <a href="{% url 'low_cie_students' %}">View Low CIE Students</a>
</body>
</html>
```

templates/low_cie_students.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Low CIE Students</title>
</head>
<body>
    <h2>Students with marks less than 20</h2>
    {% if students %}
        <table border="1">
            <tr>
                <th>USN</th>
                <th>Name</th>
                <th>Subject Code</th>
                <th>CIE Marks</th>
            </tr>
            {% for student in students %}
            <tr>
                <td>{{ student.usn }}</td>
                <td>{{ student.name }}</td>
                <td>{{ student.subject_code }}</td>
                <td>{{ student.cie_marks }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No students with CIE marks less than 20.</p>
    {% endif %}
    <br>
    <a href="{% url 'add_student' %}">Add New Student</a>
</body>
</html>
```

7. Create a Django application to accept Student Name, USN, Semester, and Exam Fee Status from a web page and store the data in a database.
Implement a feature to delete all students who have not paid the exam fee.

urls.py

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

urlpatterns = [
    path('', views.add_student, name='add_student'),
    path('students/', views.view_students, name='students'),
    path('delete_unpaid/', views.delete_unpaid_students, name='delete_unpaid')
]
```

models.py

```
from django.db import models

class ExamFee(models.Model):
    name = models.CharField(max_length=100)
    usn = models.CharField(max_length=20, unique=True)
    sem = models.IntegerField()
    exam_fee_status = models.BooleanField(default=False)
```

forms.py

```
from django.forms import ModelForm
from .models import ExamFee

class ExamFeeForm(ModelForm):
    class Meta:
        model = ExamFee
        fields = ['name', 'usn', 'sem', 'exam_fee_status']
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
from .forms import ExamFeeForm
from .models import ExamFee

def add_student(request: HttpRequest):
    if request.method == 'POST':
        form = ExamFeeForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Student added successfully.")
    else:
        form = ExamFeeForm()
    return render(request, 'add_student.html', {'form': form})

def view_students(request: HttpRequest):
    students = ExamFee.objects.all()
    return render(request, 'students.html', {'students': students})

def delete_unpaid_students(request: HttpRequest):
    deleted_students = list(ExamFee.objects.filter(exam_fee_status=False))
    ExamFee.objects.filter(exam_fee_status=False).delete()
    return render(request, 'delete.html', {'deleted_students': deleted_students})
```

templates/add_student.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Student</title>
</head>
<body>
    <h2>Add Student Details</h2>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Student</button>
    </form>
    <a href="{% url 'students' %}">View All Students</a>
</body>
</html>
```

templates/students.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Students</title>
</head>
<body>
    <h2>Students List</h2>
    {% if students %}
        <table border="1">
            <tr>
                <th>Name</th>
                <th>USN</th>
                <th>Sem</th>
                <th>Exam Fee Status</th>
            </tr>
            {% for student in students %}
            <tr>
                <td>{{ student.name }}</td>
                <td>{{ student.usn }}</td>
                <td>{{ student.sem }}</td>
                <td>{{ student.exam_fee_status }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No students found.</p>
    {% endif %}
    <a href="{% url 'add_student' %}">Add New Student</a>
    <a href="{% url 'delete_unpaid' %}"><button>Delete unpaid students</button></a>
</body>
</html>
```

templates/delete.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Deleted Students</title>
</head>
<body>
    <h2>Deleted Students who have not paid the exam fee</h2>
    {% if deleted_students %}
        <table border="1">
            <tr>
                <th>Name</th>
                <th>USN</th>
                <th>Exam Fee Status</th>
            </tr>
            {% for student in deleted_students %}
            <tr>
                <td>{{ student.name }}</td>
                <td>{{ student.usn }}</td>
                <td>{{ student.exam_fee_status }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No unpaid students were deleted.</p>
    {% endif %}
    <a href="{% url 'students' %}">Back to Students List</a
</body>
</html>
```

8. Using Django, create an HR management system with an Employee model containing the fields: Employee Name, Email, Phone, Date of Hiring, Job Title, and Salary.
Accept employee details through a web form, store them in a database, and display all employees whose salary exceeds ₹50,000.

urls.py

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

urlpatterns = [
    path('', views.add_employee, name='add_employee'),
    path('employees/', views.employee_list, name='employee_list'),
]
```

models.py

```
from django.db import models
```

```
class Employee(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    phone = models.CharField(max_length=15)
    job_title = models.CharField(max_length=100)
    salary = models.DecimalField(max_digits=10, decimal_places=2)=
```

forms.py

```
from django.forms import ModelForm
from .models import Employee

class EmployeeForm(ModelForm):
    class Meta:
        model = Employee
        fields = "__all__"
```

views.py

```
from django.shortcuts import render
from django.http import HttpResponse, HttpRequest
from .models import Employee
from .forms import EmployeeForm

def add_employee(request: HttpRequest):
    if request.method == "POST":
        form = EmployeeForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Employee added successfully.")
    else:
        form = EmployeeForm()
    return render(request, 'add_employee.html', {'form': form})

def employee_list(request: HttpRequest):
    employees = Employee.objects.filter(salary__gt=50000)
    return render(request, 'employee_list.html', {'employees': employees})
```

templates/add_employee.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Employee</title>
</head>
<body>
    <h1>Add Employee</h1>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Employee</button>
    </form>
    <a href="{% url 'employee_list' %}">View Employee List</a>
</body>
</html>
```

templates/employee_list.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee List</title>
</head>
<body>
    <h1>Employee List</h1>
    {% if employees %}
        <table border="1">
            <tr>
                <th>Name</th>
                <th>Email</th>
                <th>Phone</th>
                <th>Job Title</th>
                <th>Salary</th>
            </tr>
            {% for employee in employees %}
            <tr>
                <td>{{ employee.name }}</td>
                <td>{{ employee.email }}</td>
                <td>{{ employee.phone }}</td>
                <td>{{ employee.job_title }}</td>
                <td>{{ employee.salary }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No employees found.</p>
    {% endif %}
    <a href="{% url 'add_employee' %}">Add New Employee</a>
</body>
</html>
```

9. Build an Exam Management System using Django to store student information, and display the list of students who have secured an **"O" grade**.

urls.py

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

urlpatterns = [
    path('', views.add_student, name='add_student'),
    path('o_grade_students', views.list_ograde, name='list_ograde'),
]
```

models.py

```
from django.db import models

class Student(models.Model):
    usn = models.CharField(max_length=10, unique=True)
    name = models.CharField(max_length=100)
    sem = models.IntegerField(max_length=2)
    grade = models.CharField(max_length=2)
```

forms.py

```
from django.forms import ModelForm
from .models import Student
```

```
class StudentForm(ModelForm):
    class Meta:
        model = Student
        fields = "__all__"
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
from .forms import StudentForm
from .models import Student

def add_student(request: HttpRequest):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Student added successfully!")
    else:
        form = StudentForm()
    return render(request, 'add_student.html', {'form': form})

def list_ograde(request: HttpRequest):
    students = Student.objects.filter(grade='O')
    return render(request, 'o_grade_students.html', {'students': students})
```

templates/add_student.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Student</title>
</head>
<body>
    <h2>Add Student Details</h2>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Student</button>
    </form>
    <a href="{% url 'list_ograde' %}">View O Grade Students</a>
</body>
</html>
```

templates/o_grade_students.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>O Grade Students</title>
</head>
<body>
    <h1>O Grade Students</h1>
    {% if students %}
        <table border="1">
            <tr>
```

```
                <th>Name</th>
                <th>USN</th>
                <th>Sem</th>
                <th>Grade</th>
            </tr>
            {% for student in students %}
            <tr>
                <td>{{ student.name }}</td>
                <td>{{ student.usn }}</td>
                <td>{{ student.sem }}</td>
                <td>{{ student.grade }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No O grade students found.</p>
    {% endif %}
</body>
</html>
```

10. Create a Django application that accepts the following fields: USN, Name, and Company Name, and stores them in a database.
Display all students who are placed in Amazon.

urls.py

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

urlpatterns = [
    path('', views.add_student, name='add_student'),
    path('amazon_students', views.list_amazon, name='list_amazon'),
]
```

models.py

```
from django.db import models

class Student(models.Model):
    usn = models.CharField(max_length=10, unique=True)
    name = models.CharField(max_length=100)
    company_name = models.CharField(max_length=100)
```

forms.py

```
from django.forms import ModelForm
from .models import Student

class StudentForm(ModelForm):
    class Meta:
        model = Student
        fields = "__all__"
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
from .forms import StudentForm
from .models import Student

def add_student(request: HttpRequest):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Student added successfully!")
    else:
        form = StudentForm()
    return render(request, 'add_student.html', {'form': form})

def list_amazon(request: HttpRequest):
    students = Student.objects.filter(company_name='Amazon')
    return render(request, 'amazon_students.html', {'students': students})
```

templates/add_student.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Student</title>
</head>
<body>
    <h2>Add Student Details</h2>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Student</button>
    </form>
    <a href="{% url 'list_amazon' %}">View Amazon Students</a>
</body>
</html>
```

templates/amazon_students.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Amazon Students</title>
</head>
<body>
    <h1>Amazon Students</h1>
    {% if students %}
        <table border="1">
            <tr>
                <th>Name</th>
                <th>USN</th>
                <th>Company Name</th>
            </tr>
            {% for student in students %}
            <tr>
                <td>{{ student.name }}</td>
                <td>{{ student.usn }}</td>
                <td>{{ student.company_name }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No Amazon students found.</p>
    {% endif %}
</body>
</html>
```

11. Implement a Django application to accept an ID, Title, Name, and Branch for a faculty member and store the information in a database.
Display all faculty members who belong to the CSE branch and have the title Professor.

urls.py

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

urlpatterns = [
    path('', views.add_faculty, name='add_faculty'),
    path('cse_faculty', views.list_cse_faculty, name='list_cse_faculty'),
]
```

models.py

```
from django.db import models

class Faculty(models.Model):
    faculty_id = models.CharField(max_length=10, unique=True)
    title = models.CharField(max_length=50)
    name = models.CharField(max_length=100)
    branch = models.CharField(max_length=50)
```

forms.py

```
from django.forms import ModelForm
from .models import Faculty

class FacultyForm(ModelForm):
    class Meta:
        model = Faculty
        fields = "__all__"
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
from .forms import FacultyForm
from .models import Faculty

def add_faculty(request: HttpRequest):
    if request.method == 'POST':
        form = FacultyForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Faculty added successfully!")
    else:
        form = FacultyForm()
    return render(request, 'add_faculty.html', {'form': form})

def list_cse_faculty(request: HttpRequest):
    faculty = Faculty.objects.filter(branch='CSE', title='Professor')
    return render(request, 'cse_faculty.html', {'faculty': faculty})
```

templates/add_faculty.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Faculty</title>
</head>
<body>
    <h2>Add Faculty Details</h2>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Faculty</button>
    </form>
    <a href="{% url 'list_cse_faculty' %}">View CSE Faculty</a>
</body>
</html>
```

templates/cse_faculty.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSE Faculty</title>
</head>
<body>
    <h1>CSE Faculty</h1>
    {% if faculty %}
        <table border="1">
            <tr>
                <th>Name</th>
                <th>ID</th>
                <th>Title</th>
                <th>Branch</th>
            </tr>
            {% for f in faculty %}
            <tr>
                <td>{{ f.name }}</td>
                <td>{{ f.faculty_id }}</td>
                <td>{{ f.title }}</td>
                <td>{{ f.branch }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No CSE faculty found.</p>
    {% endif %}
</body>
</html>
```

12. Design a Django application to accept Name, USN, Department, and Grade from a web page and store the data in a database.
Provide functionality to update the grade of a student based on the Name specified by the user, and display the updated results.

urls.py

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

```
urlpatterns = [
    path('', views.add_student, name='add_student'),
    path('update/', views.update_student, name='update_student'),
    path('list/', views.list_students, name='list_students'),
]
```

models.py

```
from django.db import models

class Student(models.Model):
    usn = models.CharField(max_length=10, unique=True)
    name = models.CharField(max_length=100)
    department = models.CharField(max_length=100)
    grade = models.CharField(max_length=2)
```

forms.py

```
from django import forms
from .models import Student

class StudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = "__all__"

class StudentUpdateForm(forms.Form):
    name = forms.CharField(max_length=100)
    grade = forms.CharField(max_length=2)
```

views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
from .forms import StudentForm, StudentUpdateForm
from .models import Student

def add_student(request: HttpRequest):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponse("Student added successfully!")
    else:
        form = StudentForm()
    return render(request, 'add_student.html', {'form': form})

def update_student(request: HttpRequest):
    if request.method == 'POST':
        form = StudentUpdateForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            name = cd['name']
            grade = cd['grade']
            try:
                student = Student.objects.get(name=name)
                student.grade = grade
                student.save()
                return HttpResponse("Student grade updated successfully!")
            except Student.DoesNotExist:
                return HttpResponse("Student not found.")
    else:
        form = StudentUpdateForm()
    return render(request, 'update_student.html', {'form': form})

def list_students(request: HttpRequest):
    students = Student.objects.all()
    return render(request, 'list_students.html', {'students': students})
```

templates/add_student.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Student</title>
</head>
<body>
    <h1>Add Student</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Student</button>
    </form>
    <br>
    <a href="{% url 'list_students' %}">View Student List</a>
    <br>
    <a href="{% url 'update_student' %}">Update Student</a>
</body>
</html>
```

templates/update_student.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Update Student</title>
</head>
<body>
    <h1>Update Student</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Update Student</button>
    </form>
    <br>
    <a href="{% url 'list_students' %}">View Student List</a>
    <br>
    <a href="{% url 'add_student' %}">Add Student</a>
</body>
</html>
```

templates/list_students.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>All Students</title>
</head>
<body>
    <h1>All Students</h1>
    {% if students %}
        <table border="1">
            <tr>
                <th>Name</th>
                <th>USN</th>
                <th>Grade</th>
            </tr>
            {% for student in students %}
            <tr>
                <td>{{ student.name }}</td>
                <td>{{ student.usn }}</td>
                <td>{{ student.grade }}</td>
            </tr>
            {% endfor %}
        </table>
    {% else %}
        <p>No students found.</p>
    {% endif %}
    <br>
    <a href="{% url 'add_student' %}">Add Student</a>
    <br>
    <a href="{% url 'update_student' %}">Update Student</a>
</body>
</html>
```