NEVEAN ADHIAMBO OMONDI

SCT211-0024/2023

ICS 2208

B.SC. COMPUTER SCIENCE

17/01/2025

1a. django-admin startproject myproject

b. cd myproject

python manage.py runserver

2a. Serves as the entry point for various Django management tasks such as running the development server, creating apps and applying database migrations.

b. Contains all configuration settings for your Django project defining how the project interacts with the environment, database, installed apps and other components.

3. In Django, when a request is made, it is first handled by the WSGI/ASGI application (wsgi.py or asgi.py), which directs it to the URL dispatcher defined in urls.py. The URL patterns match the request to a specific view function in views.py, which processes the request by retrieving data, applying logic, or rendering templates (if necessary). Middleware, defined in settings.py, processes the request and response at various stages for tasks like security and session management. Finally, the prepared response, such as an HTML page or JSON data, is sent back to the client.

4. Data is created using the ORM (Object-Relational Mapping) by defining models that map to database tables. After defining a model, migrations are applied to create the corresponding table in the database. Data can then be created using the Django shell, views, forms, or the admin interface. The ORM provides methods like create() to directly insert data, or you can instantiate a model object and save it using save(). This abstraction simplifies database interactions, allowing developers to work with Python objects instead of raw SQL.

5. python manage.py makemigrations

python manage.py migrate

6. from members.models import Member

7. <!DOCTYPE html>

<html>

<head>

<title>Greeting Page</title>

</head>

<body>

<h1>Hello, {{ firstname }}.</h1>

</body>

</html>

8. from django.shortcuts import render

def greet(request):

# Pass the 'firstname' variable to the template

return render(request, 'greeting.html', {'firstname': 'Mwakio'})

9. <!DOCTYPE html>

<html>

<head>

<title>Member List</title>

</head>

<body>

<h1>Member List</h1>

<ul>

{% for member in members %}

<li>{{ member }}</li>

{% endfor %}

</ul>

</body>

</html>

10. a) SQL Injection: An attacker injects malicious SQL code into queries to manipulate or access data.

b) Insecure Direct Object References (IDOR): An attacker manipulates object identifiers to access unauthorized data.

11. from django.db import models

class Plant(models.Model):

name = models.CharField(max\_length=100)

area\_needed = models.DecimalField(max\_digits=5, decimal\_places=2) # Size/Area around plant in square meters

def \_\_str\_\_(self):

return self.name

class GardenRow(models.Model):

plant = models.ForeignKey(Plant, on\_delete=models.CASCADE) # Linking a row to a specific plant

number\_of\_plants = models.PositiveIntegerField() # Number of plants in the row

row\_length = models.DecimalField(max\_digits=5, decimal\_places=2) # Length of the row in meters

def total\_area(self):

return self.number\_of\_plants \* self.plant.area\_needed # Calculate the total area required for all plants in the row

def \_\_str\_\_(self):

return f"Row of {self.number\_of\_plants} {self.plant.name}(s)"