

23IT22SC – PYTHON WITH DJANGO

(Skill Enhancement Course)

(IT)

Course Category:	Skill Enhancement Course	Credits:	2
Course Type:	Practical	Lecture-Tutorial-Practical:	1-0-2
Prerequisite:	Web Design Frameworks Version Control systems	Sessional Evaluation: Univ. Exam Evaluation: Total Marks:	30 70 100
Objectives:	Students undergoing this course are expected: <ul style="list-style-type: none">• Design and build static as well as dynamic web pages and interactive web-based applications• Web development using Django framework.• Analyse and create functional website in Django and deploy Django Web Application on Cloud		

Course Outcomes	Upon successful completion of the course, the students will be able to:	
	CO1	Design Websites. (L6)
	CO2	Apply Styling to web pages. (L4)
	CO3	Make Web pages interactive Using Django (L6)
	CO4	Design Forms for applications. (L6)
	CO5	Choose Control Structure based on the logic to be implemented. (L3)
Course Content	<p style="text-align: center;"><u>UNIT-I</u></p> <p>Python libraries for web development: Collections-Container datatypes, Tkinter-GUI applications, Requests-HTTP requests, BeautifulSoup4-web scraping, Scrapy, Zappa, Dash, CherryPy, Turbo Gears, Flask, Web2Py, Bottle, Falcon, Cubic Web, Quixote, Pyramid.</p> <p>Sample Experiments:</p> <ol style="list-style-type: none">1. Write a Python GUI program to import Tkinter package and create a window. Set its title and add a label to the window.2. Write a Python program that designs a simple login form with labels and Entry widgets, arranging them in a grid using the Grid geometry manager.3. Write a program using BeautifulSoup4 library for web scraping for a given URL4. Develop a sample Hello World page using Flask framework5. Develop a sample web page using CherryPy / Web2Py / Bottle Framework <p style="text-align: center;"><u>UNIT-II</u></p> <p>Introduction to Django Framework: Understanding Django environment, Features of Django and Django architecture, MVC and MTV, Urls and Views, Mapping the views to URLs, Django Template, Template inheritance Django Models, creating model for site, Converting the model into a table, Fields in Models, Integrating Bootstrap into Django, creating tables, creating grids, Creating carousels.</p>	

Sample Experiments:

1. Create a Sample “Hello World” Application using Django
2. Create a Login and Registration Page using MVC architecture in Django Framework
3. Create a sample page in Django by integrating Bootstrap.
4. Create an application with Tables, grids in Django
5. Create a Django App with Carousels feature.

UNIT-III

Integrating Accounts & Authentication on Django: Introduction to Django Authentication System, Security Problem & Solution with Django Creating Registration Form using Django, Adding Email Field in Forms, configuring email settings, sending emails with Django, Adding Grid Layout on Registration Page, Adding Page Restrictions, Login Functionality Test and Logout.

Sample Experiments:

1. Create a registration page using Authentication System
2. Create an application in Django to send emails using email settings and Grid Layout
3. Create an application in Django using page restriction / authentication with Login and Logout Functionality
4. Create a sample form using Django Forms

UNIT-IV

Connecting SQLite with Django: Database Migrations, Fetch Data from Database, Displaying Data on Templates, Adding Condition on Data, sending data from url to view, sending data from view to template, saving objects into database, sorting objects, filtering objects, deleting objects, Difference between session and cookie, Creating sessions and cookies in Django.

Sample Experiments:

1. Create an app in Django which fetches data from database and show as list and also save objects in database
2. Create an app in Django for performing CRUD operations on records in a database
3. Create an app in Django which uses session management and cookies to store and manage user sessions.

UNIT-V

Deploying Django Web Application on Cloud: Creating a functional website in Django, Four Important Pillars to Deploy, registering on Heroku and GitHub, Push project from Local System to GitHub, working with Django Heroku, Working with Static Root, Handling WSGI with gunicorn, setting up Database & adding users.

Sample Experiments:

1. Create a website in Django with login, and registration page.

	<ol style="list-style-type: none"> 2. Register on GitHub, and Heroku and deploy the website on Heroku with all the functionalities developed. 3. Configure Django to handle static files.
Text Books & References Books	<p>TEXT BOOKS:</p> <ol style="list-style-type: none"> 1. Tom Aratyn, “Building Django 2.0 Web Applications: Create enterprise-grade, scalable Python web applications easily with Django 2.0”, 2nd Edition 2018, Packt Publishing. 2. Harry Percival, “Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium and JavaScript”, 2nd Edition 2019, Kindle Edition. 3. Daniel Rubio, Apress, “Beginning Django Web Application Development and Deployment with Python”, 2nd Edition 2017, Apress.
E-Resources	<ol style="list-style-type: none"> 1. https://www.w3schools.com/html 2. https://www.w3schools.com/django 3. https://www.w3schools.com/js/ 4. https://www.w3schools.com/nodejs 5. https://www.w3schools.com/typescript