



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

W.E. F. Academic Year:	2026-27
Semester:	5 th
Category of the Course:	PEC-03

Prerequisite:	Basic knowledge of Python, Database, HTML, CSS, Java Script, Bootstrap.
Rationale:	Web applications form the backbone of modern software systems used in education, business, healthcare, and government sectors. Python is widely adopted for web development due to its simplicity, readability, and strong open-source ecosystem. Although students have prior knowledge of basic Python and front-end technologies, industry-oriented web development requires skills in object-oriented programming and database-driven application design. This subject introduces Python Object-Oriented Programming and database concepts essential for building robust server-side applications. Students learn to develop dynamic web applications using popular open-source frameworks such as Flask and Django. The course also exposes learners to REST APIs, enabling integration between different software systems. Emphasis is given to web application security, authentication, and secure data handling. Students gain hands-on experience in developing and deploying web applications using open-source tools. Overall, this subject enhances practical skills and prepares students for entry-level roles in Python-based web and full-stack development.

Course Outcome:

After Completion of the Course, Student will be able to:

No	Course Outcomes	RBT Level
01	Apply Python object-oriented programming concepts to develop modular and reusable code.	Apply
02	Use Python database concepts to create, retrieve, update, and delete data in database-driven applications.	Apply
03	Develop dynamic web applications using the Flask framework with routing, templates, and database integration.	Apply
04	Develop structured web applications using the Django framework with models, views, templates, and forms.	Apply
05	Implement REST APIs, security features, and deploy Django applications to production environments.	Apply

**Revised Bloom's Taxonomy (RBT)*



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
L	T	PR	C	Theory		Tutorial / Practical		
				ESE (E)	PA(M)	PA(I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Python Object-Oriented Programming (OOP) • Introduction to OOP • Classes, Objects, Attributes and Methods • Constructors (__init__), destructors (__del__) and self parameter • Encapsulation and Data hiding: ○ Public, protected (_), and private (__) attributes/methods ○ Accessing data using methods • Inheritance: ○ Single inheritance ○ Multilevel inheritance ○ Use of super() • Polymorphism ○ Method overriding ○ Operator overloading • Abstraction ○ Abstract classes ○ Use of abc module • Magic methods (Dunder Methods): __str__, __repr__, __len__, __add__, __eq__ etc.	9	20
2.	Python Database Programming • Introduction to SQLite and PostgreSQL • Python Database Connectivity using SQLite (sqlite3 module) ○ Connecting to database ○ Executing SQL queries	7	16



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

	- SQL Commands using sqlite3 - DDL: CREATE TABLE, ALTER TABLE, DROP TABLE - DML: INSERT, SELECT (with WHERE and ORDER BY), UPDATE, DELETE - Aggregate functions: COUNT, SUM, AVG, MIN, MAX - GROUP BY and HAVING clause - Parameterized Queries to prevent SQL injection (using ? placeholders) - Introduction to ORM (Object Relational Mapping) - Need for ORM in modern web applications - ORM concept - Brief introduction to ORM tools used in Python web frameworks (SQLAlchemy, Django ORM)		
3.	Flask Framework - Introduction to Flask - Introduction to web frameworks - Micro (Flask) vs. Full-stack (Django) Frameworks - Flask framework features and advantages - Flask application Structure - Routing, Views and URL Handling - Routing concept in Flask - Defining routes with @app.route() decorator - URL Variable rules, converters - HTTP methods: specifying methods=['GET', 'POST'] - Using url_for() for dynamic link generation - Dynamic UI with Jinja2 Templating Engine - Jinja2 template syntax - Rendering templates using render template() - Variables: {{variable}}, Filters {{name title}} - Control structures: {% if %}, {% for %} - Template inheritance: {% extends 'base.html' %}, {% block content %} - Linking CSS, JavaScript using the static folder/CDN - Forms Handling with WTForms - Form classes: class LoginForm(FlaskForm): with fields/validators - Template integration: {{form.hidden_tag()}}, {{form.field.Label}} - Processing: form.validate_on_submit(), accessing form.data	10	22



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

	● Flask with Database ○ Connecting Flask with SQLite database ○ Creating database and performing CRUD operations ○ Displaying database data on web pages ● Sessions, Cookies, and Flash Messages ○ Sessions: session['key'] = value, session.pop('key') ○ Cookies: response.set_cookie('key', 'value') ○ Flash messages: flash(), display with get_flashed_messages()		
4.	Django Framework ● Django Introduction and Project Structure ○ Overview and features of Django ○ Django project/app structure ○ MVT architecture: Model (data), View (logic), Template (presentation) ● Views and URL Mapping ○ URL configuration: path(), include() ○ Function-based views (FBV): render(), redirect() ○ Class-based views (CBV): ListView, DetailView, CreateView ○ Context processors, RequestContext; HTTP methods handling ● Models and Django ORM ○ Defining models: classes inheriting from models.Model ○ Migrations: makemigrations, migrate, understanding migration files ○ Django ORM basics ○ CRUD operations using Django ORM ● Django Admin Interface: ○ Enabling admin: create superuser, register models in admin.py ○ Customizing admin ● Templates and Template Inheritance ○ Django template language: Template variables, template tags, common template filters ○ Bootstrap integration ● Forms and ModelForms ○ Built-in forms: Form class vs. ModelForm ○ Creating ModelForms ○ Form rendering: {{ form.as_p }}, manual field rendering ○ File uploads: FileField, MEDIA_URL/ROOT config ○ Validation and error display in templates	11	24



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

5.	Advance Django: REST APIs, Security and Deployment • Authentication and Security ○ Django's built-in auth system (User model, groups, permissions) ○ Registration, login, and logout views ○ protecting views with decorators/mixins (@login_required, LoginRequiredMixin), and permission checks with PermissionRequiredMixin ○ Web Security Best Practices for protection against common vulnerabilities: Cross-Site Request Forgery (CSRF): middleware + tokens, Cross-Site Scripting (XSS): auto-escaping, SQL injection (ORM usage), HTTPS enforcement • Django Middleware & Sessions • Django REST Framework (DRF) ○ Introduction to DRF ○ Serializers: ModelSerializer for Post/Category, validation (validate(), create()) ○ Function-based views using @api_view, Class-based views using APIView and Generic Views ○ Routers: DefaultRouter(), URL auto-generation (/api/posts/) • Deployment ○ Environment variables: using python-dotenv or django-environ for secrets ○ Deploy a simple Django app locally and on cloud platforms like PythonAnywhere.	8	18
	Total	45	100

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	U Level	A Level	N Level	E Level	C Level
25	35	40	0	0	0

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. William S. Vincent, "Django for Beginners", 5th edition, Still River Press
2. Miguel Grinberg, "Flask Web Development: Developing Web Applications with Python", 2nd edition, Shroff/O'Reilly



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

3. Eric Matthes, “Python Crash Course”, 3rd edition, Penguin Random House India Pvt. Ltd.
4. Antonio Melé, “Django 4 By Example”, 4th edition, Packt Publishing Ltd.
5. Julia Solórzano and Mark Lavin, “Lightweight Django: A Crash Course in 100 Short Lessons”, 1st edition, O'Reilly

(b) Open source software and website:

1. <https://docs.djangoproject.com>
2. <https://flask.palletsprojects.com>
3. <https://www.django-rest-framework.org>
4. <https://www.fullstackpython.com>
5. <https://www.geeksforgeeks.org/python/python-web-development>
6. <https://www.geeksforgeeks.org/python/flask-tutorial>
7. <https://www.w3schools.com/django>
8. <https://www.tutorialspoint.com/flask>

Suggested Course Practical List:

1. Write Python programs to demonstrate OOP concepts such as classes, objects, attributes, methods, encapsulation, and inheritance.
2. Write Python programs to demonstrate OOP concept polymorphism (method overriding) and abstract class (using abc module).
3. Write a Python program to connect to SQLite database using sqlite3 module, create a table, and execute basic queries.
4. Write a Python program to perform full CRUD operations on a SQLite database using sqlite3 module (parameterized queries to prevent SQL injection).
5. Set up a Python virtual environment, install Flask, and develop a simple "Hello World" web application to understand Flask project structure, basic routing, and Jinja2 templates.
6. Develop a Flask web application to handle GET and POST requests using HTML forms; integrate static files (CSS, JavaScript, images) and Bootstrap for styling.
7. Develop a Flask web application to perform CRUD operations by connecting to an SQLite3 database.
8. Develop a Flask web application to implement a user registration and login system with session management using an SQLite3 database.
9. Set up a Python virtual environment, install Django, and develop a simple "Hello World" web application to understand Flask project structure, basic routing, and Jinja2 templates.
10. Develop a Django web application with database integration by creating a sample model to store and retrieve data (e.g. student records, customer feedback, or book information).
11. Create an admin user using the Django Admin panel and manage application data through the admin interface.
12. Implement a login/logout (user authentication) system using Django's built-in Auth features and decorators with CSRF protection.
13. Develop a Django CRUD application with session-based features (e.g., user-specific data) and proper form handling using Django forms or ModelForms.



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

14. Develop a simple REST API using Django REST Framework (DRF) to perform CRUD operations on a database table.
15. Develop and deploy a web application using either Flask or Django framework.

List of Laboratory/Learning Resources Required:

1. Computer with latest configurations and Internet connectivity.
2. Python Environment
3. IDEs: VS Code, PyCharm
4. Web Browser: Google Chrome, Firefox
5. Python Packages: Flask, Flask-SQLAlchemy, WTForms, Flask-WTF, Django, djangorestframework (DRF)
6. Database Management Systems: SQLite (sqlite3)
7. API Testing Tool: Postman
8. Version Control: git (Optional)

Suggested Project List:

1. Develop a Personal Portfolio Website with multi-page (Home, About, Projects, Contact) using Flask and Jinja2 templates.
2. Develop a Task Management "To-Do" App, where users can add tasks, mark them as completed, and delete them.
3. Develop an Online Notes Manager that allows users to save, edit, delete and view notes.
4. Develop a Weather App using External API, where a user enters a city name and the app displays the current weather by fetching data from the OpenWeatherMap API using the requests library.
5. Develop a Student Management System website to store student information with add, delete, search and update facility.
6. Develop a URL Shortener web application, where a user submits a long URL and receives a shortened version. When the short URL is visited, the app redirects them to the original link.
7. Develop a Simple Blog Engine, where an admin can write posts via the Django Admin panel, and public users can read them and leave comments.
8. Develop a Simple Library Management System for managing books in library.
9. Develop an Online Quiz System, where user can take a quiz and scores are displayed after submission.
10. Develop a Product Catalog website, where products can be added, updated, or deleted and Users can view product details.
11. Develop an Expense Tracker website, where user can add expenses, view expenses and filter expenses based on category/date.
12. Develop a Recipe Sharing website, where user can add/edit/delete recipes (after login), view all recipes and basic search based on recipe or ingredient.



GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering

Level: Diploma

Branch: Computer Engineering / Computer Science and Engineering

Subject Code: DI05000091

Subject Name: Web Development using Python

Suggested Activities for Students:

- a) Students can solve programming and Python problems on platforms such as HackerRank, CodeChef, LeetCode, and Codeforces to improve programming logic.
- b) Students should enroll in online courses related to Python Web Development on platforms such as SWAYAM, Coursera, edX, or Udemy.
- c) Students should develop a small web application using Flask or Django, such as a To-Do list, blog application, or student record management system and deploy it on free platforms such as PythonAnywhere, Render, or Railway.
- d) Students should explore open-source Python web projects on GitHub and understand project structure and code organization.
- e) Students can form or participate in a coding or development club in the institute to collaborate on projects and share knowledge.
- f) Students can prepare presentations/charts on topics such as Flask vs Django, Django MVT architecture, REST APIs, Web Security, or Cloud Deployment.
