# Python for Web and Data Applications

**Module 1:** Python Fundamentals

- Installing Python, pip, virtual environments
- Variables, Data Types
- Operators & Expressions
- Conditional Statements
- Loops (for, while)
- Functions & Lambda Expressions
- Modules & Packages
- Exception Handling
- Working with files (read/write JSON, CSV, text)

**Module 2:** Object-Oriented Programming in Python

- Classes and Objects
- _init_ and other dunder methods
- Inheritance and Polymorphism
- Encapsulation

**Module 3:** Web Development with Python

**Flask (Lightweight Framework)**

- Flask Basics and Installation
- Routing and Views
- Rendering HTML Templates (Jinja2)
- Form Handling
- URL Parameters and Query Strings
- Session Management and Cookies
- RESTful API creation
- Blueprint for modular projects

**Django (Full-Stack Framework)**

- Django Setup and Architecture (MVC/MVT)
- Models, Views, Templates
- Admin Panel and ORM
- URL Routing and Forms
- Django REST Framework (DRF)

**Module 4:** Frontend Integration

- HTML/CSS/Javascript Basics
- Bootstrap for responsive design
- Templating with Jinja2 or Django Templates

- Sending and receiving data with Fetch/AJAX
- Connecting Frontend with Flask/Django API

**Module 5:** Python for Data Applications

**Data Analysis Tools**
- NumPy: Arrays, Matrix Operations
- Pandas: DataFrames, Series, Filtering, Merging, GroupBy
- Reading/Writing CSV, Excel, JSON
- Handling missing values
- Data transformation

**Data Visualization**
- Matplotlib: Line, Bar, Pie, Scatter plots
- Seaborn: Heatmaps, Boxplots, Histograms
- Plot customization

**Module 6:** Database Integration

- Relational Databases (MySQL / PostgreSQL / SQLite)
- CRUD operations
- Python DB-API
- Using sqlite3 or SQLAlchemy (ORM)
- NoSQL with MongoDB using PyMongo

**Module 7:** REST APIs & JSON

- REST architecture concepts
- Building RESTful APIs in Flask/Django
- Consuming external APIs using requests
- Sending/Receiving JSON
- API Authentication with JWT or API Key

**Module 8:** Deployment

- Environment variables and configuration
- Hosting Flask/Django apps on:
- Heroku / Render / Railway
- PythonAnywhere
- Introduction to Docker (optional)
- Git & GitHub basics for version control

**Module 9:** Security Essentials

- Input validation & sanitization
- Preventing SQL Injection, XSS
- HTTPS and SSL basics

- User Authentication and Authorization
- Hashing passwords with bcrypt or werkzeug

**Module 10:** Project Work

- Build real-world Python web and data applications such as:
- Blog or CMS with Flask/Django
- Data Dashboard for real-time analytics
- Weather App using external APIs
- CRUD app with SQL or MongoDB
- Data Visualization Web App
- Student/Employee Management System

==========

**END**

==========