Python & Django Backend Development

Introduction

The Python Programming Certification is designed to teach the basics of python programming. Python is a general-purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. With this certification, students will be able to create complete web applications using python libraries and Django Framework. Python programmers are highly paid in the IT industry and python applications are highly demanding on freelancing platforms. It benefits students who have completed intermediate in computers and have basic programming skills.

CURRICULUM:

Sr. No.	Contents

1	**Python Data Structures** • Lists, tuples, sets, dictionaries • Common methods & operations • Nested data structures • Mini exercise: student record system
2	**Functions & Modules** • Defining & calling functions • Parameters & return values • Default, keyword, variable arguments • Importing modules, writing custom modules • Mini exercise: calculator using functions
3	**OOP in Python (Backend Relevant)** • Classes & objects • Constructor (__init__) • Inheritance • Polymorphism (method overriding) • Mini project: employee–manager system
4	**Python for Backend Tools** • File handling (txt, JSON, CSV) • Exception handling (try-except) • Virtual environments & dependency management • Git basics (init, commit, push, pull, branch) • Mini exercise: log analyzer with Git version control

5	**Django Setup & Basics** • Why Django for backend • Setting up Django project & apps • Project structure explained • URL routing & basic views • Templates introduction (rendering HTML) • Mini project: “Hello Django” app
6	**Models & Databases** • Django ORM (models, migrations) • CRUD with QuerySets • Relationships (OneToOne, ForeignKey, ManyToMany) • Django admin basics • Mini project: blog posts stored in database
7	**Templates & Forms** • Django template language (tags, filters, loops, conditions) • Static & media files • Django forms & ModelForms • CSRF protection • Mini project: create post form for blog
8	**Authentication & User Management** • Django built-in user model • Login, signup, logout • Permissions & groups • Password reset & hashing • Mini project: user authentication system

9	**Django REST Framework (DRF) – Basics** • What is REST API • DRF setup • Serializers & ModelSerializers • API Views & ViewSets • Mini project: blog API (CRUD endpoints)
10	**DRF Advanced** • Routers & nested routes • JWT authentication • Permissions & throttling • Connecting DRF with database • Mini project: user-authenticated blog API
11	**Django Advanced & Sockets** • Middleware • Signals (pre_save, post_save) • Django Channels setup • WebSockets with Django (real-time communication) • Mini project: real-time chat room
12	**Capstone Projects & Deployment** • Deployment basics (Gunicorn, Nginx, Docker intro) Final Project Assignment (choose one): • Blogging platform (web + API + auth) • Task management system (DRF + JWT + DB) • Real-time chat app (Django Channels + WebSockets) • E-commerce backend (products, cart, orders, users, payments)

Learning Outcomes:

By the end of this course, participants will be able to:

- Develop complete web application using python.
- Work with Django Framework
- Python applications

Course Benefits:

- To develop web applications using python
- Understanding of python programming
- Working with Django Framework

Skill-Wise Earnings:

Skill Level	Avg Monthly Salary
Junior	75k-100k
Mid-Level	100k - 170k
Advanced	250k- 450k
Freelancer	Earn in millions

Affiliation & Collaboarations