

Python Backend Development Course

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	Introduction to Python • Setup (Python, VSCode, venv) • Writing & running Python scripts • Variables, data types, operators • Input & output • First simple program

Sr. No.	Contents
2	Control Flow • If-else conditions • Loops (for, while) • Loop control (break, continue, pass) • Small exercises: calculator, number guessing
3	Data Structures • Lists, tuples, sets, dictionaries • Common methods & operations • Iterating through collections • Mini project: student record system
4	Functions • Defining & calling functions • Arguments & return values • Scope (local vs global) • Importing & using modules • Mini project: function-based calculator
5	File Handling & Error Handling • Reading/writing text & JSON files • with statement • Try-except-finally for error handling • Mini project: simple log file writer

Sr. No.	Contents
6	Object-Oriented Programming (Backend Relevant) • Classes & objects • Constructor (<code>__init__</code>) • Inheritance • Polymorphism (method overriding) • Mini project: employee system with inheritance
7	Development Tools • Virtual environments (venv, pip) • Installing & freezing dependencies • Git & GitHub basics (init, add, commit, push, pull, branch) • Exercise: Push mini Python project to GitHub
8	Databases • Introduction to databases (SQL vs NoSQL) • SQLite with Python (sqlite3 module) • CRUD operations • Connecting Python with MySQL/PostgreSQL (basic) • Mini project: store user data in database
9	Flask Basics • Flask setup & structure • Routes & views • Handling requests (GET, POST) • Rendering templates with Jinja2 • Mini project: simple blog with Flask + SQLite

Sr. No.	Contents
10	FastAPI Basics • FastAPI setup • Path & query parameters • Request body with Pydantic • Returning JSON responses • Mini project: To-Do API with FastAPI + SQLite
11	APIs & Authentication • REST API concepts (CRUD, endpoints) • Connecting Flask/FastAPI to DB • JWT Authentication (login, signup) • Mini project: user authentication API
12	WebSockets & Final Projects • Introduction to WebSockets • Flask-SocketIO basics • FastAPI with WebSockets (real-time communication) - FastAPI with WebSockets (real-time communication) ◦ Chat app with FastAPI & WebSockets ◦ Task management system (Flask + DB + JWT) ◦ E-commerce backend (products, users, orders, DB) ◦ Blogging API (FastAPI + DB + auth)

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

