



E-HOOYIA

Backend Training

Presented by:
Hooyia



+237 691435485
+237 697907096



e-hooyia.com



Bafoussam, Cameroon

contact@e-hooyia.com

Course Duration: 3 Months	2
Month 1: Introduction to Python (4 Weeks)	2
Week 1: Python Basics.....	2
Week 2: Functions and Modules.....	3
Week 3: Data Structures.....	3
Week 4: File Handling and Libraries.....	4
Month 2: Web Development with Django (4 Weeks)	4
Week 5: Django Basics.....	4
Week 7: Django Views and Templates.....	5
Week 8: Advanced Django Features.....	6
Month 3: API Development and Deployment (4 Weeks)	6
Week 9: Introduction to APIs.....	6
Week 10: Authentication and Permissions.....	7
Week 11: Deployment Strategies.....	7
Week 12: Final Project.....	8
Course Objectives	8

Course Proposal: Python/Django/API/Deployment

Join Python and Django Backend Development Program and advance your skills from beginner to expert. This intensive course focuses on hands-on projects that equip you with essential backend development techniques, preparing you for a successful career in the tech industry.

Course Structure

Here's the revised course outline structured for a 5-day week format over 3 months:

Course Duration: 3 Months

Month 1: Introduction to Python (4 Weeks)

Week 1: Python Basics

- Day 1: Introduction to Python
 - Overview of Python and its applications. Setting up the development environment (Anaconda, Jupyter Notebook, or IDE).
- Day 2: Syntax and Variables
 - Understanding Python syntax and variable declaration. Introduction to basic data types.
- Day 3: Control Structures
 - Conditional statements (if, elif, else) and loops (for, while).
- Day 4: Practical Project: Simple Calculator
 - Develop a command-line calculator that performs basic arithmetic operations.
- Day 5: Review and Assessment
 - Recap of the week's concepts and a quiz to assess understanding.

Week 2: Functions and Modules

- Day 6: Function Definitions
 - Learn to create and call functions, including parameters and return values.

- Day 7: Modules and Packages
 - Introduction to importing modules and using Python packages.
- Day 8: Error Handling
 - Understanding exceptions and using try-except blocks.
- Day 9: Practical Project: To-Do List Application
 - Create a command-line to-do list application utilizing functions and modules.
- Day 10: Review and Assessment
 - Summary of key points and a quiz on functions and modules.

Week 3: Data Structures

- Day 11: Introduction to Data Structures
 - Overview of data structures in Python.
- Day 12: Lists and Tuples
 - Detailed exploration of lists and tuples, including methods and operations.
- Day 13: Dictionaries and Sets
 - Understanding dictionaries and sets, their methods, and use cases.
- Day 14: Practical Project: Data Analysis
 - Conduct data analysis using lists and dictionaries to solve a real-world problem.
- Day 15: Review and Assessment
 - Recap of data structures and a quiz to assess knowledge.

Week 4: File Handling and Libraries

- Day 16: File Handling Basics
 - Learn to read from and write to files using Python.

- Day 17: Working with CSV Files
 - Introduction to handling CSV files using Python's built-in libraries.
- Day 18: Introduction to NumPy
 - Explore NumPy and its applications for numerical computations.
- Day 19: Introduction to Pandas
 - Overview of Pandas for data manipulation and analysis.
- Day 20: Practical Project: Text-Based Game
 - Build a simple text-based game utilizing file handling and data structures.

Month 2: Web Development with Django (4 Weeks)

Week 5: Django Basics

- Day 21: Introduction to Django
 - Overview of Django, its architecture, and installation.
- Day 22: Setting Up Your Environment
 - Configure a Django project and its settings.
- Day 23: Understanding MVC Architecture
 - Learn about the Model-View-Controller (MVC) architecture in Django.
- Day 24: Creating Your First Django App
 - Build a simple Django app and understand its structure.
- Day 25: URL Routing
 - Learn how to configure URL routing in Django applications.

Week 6: Django Models and Databases

- Day 26: Introduction to Models
 - Understand the concept of models and database design.
- Day 27: Object-Relational Mapping (ORM)

- Learn about Django's ORM for database interactions.
- Day 28: Migrations
 - Understand database migrations and how to manage them.
- Day 29: Practical Project: Blog Enhancement
 - Enhance the blog application with a comments feature using models.
- Day 30: Admin Interface
 - Explore Django's built-in admin interface for managing models.

Week 7: Django Views and Templates

- Day 31: Introduction to Views
 - Learn about function-based and class-based views.
- Day 32: Templates
 - Understand Django's templating engine and how to create dynamic web pages.
- Day 33: Static Files
 - Learn how to serve static files in Django applications.
- Day 34: Practical Project: User Authentication
 - Implement user authentication in the blog using views and templates.
- Day 35: Review and Assessment
 - Recap the week's concepts and a quiz.

Week 8: Advanced Django Features

- Day 36: Middleware
 - Understand middleware and how it modifies requests and responses.
- Day 37: Signals

- Learn about Django signals for decoupling applications.
- Day 38: Custom Management Commands
 - Create custom management commands for repetitive tasks.
- Day 39: Practical Project: Admin Dashboard
 - Build an admin dashboard for managing blog content and users.
- Day 40: Review and Assessment
 - Summarize advanced features of Django and take a quiz.

Month 3: API Development and Deployment (4 Weeks)

Week 9: Introduction to APIs

- Day 41: Understanding APIs
 - Overview of APIs and RESTful services.
- Day 42: Setting Up Django REST Framework (DRF)
 - Install and configure DRF in your Django project.
- Day 43: Building Your First API
 - Create a basic API endpoint using DRF.
- Day 44: Serializers
 - Learn about serializers for converting complex data types.
- Day 45: Practical Project: Blog API
 - Build a RESTful API for the blog application.

Week 10: Authentication and Permissions

- Day 46: User Authentication
 - Implement user authentication in your API.

- Day 47: Permissions and Access Control
 - Understand permission classes in DRF.
- Day 48: Token-Based Authentication
 - Learn to implement token-based authentication for APIs.
- Day 49: Practical Project: Secure the API
 - Enhance the blog API to secure user data and manage permissions.
- Day 50: CORS (Cross-Origin Resource Sharing)
 - Learn about CORS and how to configure it in Django.

Week 11: Deployment Strategies

- Day 51: Introduction to CI/CD
 - Overview of Continuous Integration and Continuous Deployment.
- Day 52: Using Docker
 - Learn the basics of Docker for containerization.
- Day 53: Dockerizing a Django Application
 - Step-by-step process to containerize the Django application.
- Day 54: Deploying to Cloud Platforms
 - Overview of deploying Django applications on platforms like Heroku or AWS.
- Day 55: Practical Project: Containerized Deployment
 - Deploy the Django application using Docker.

Week 12: Final Project

- Day 56: Capstone Project Introduction
 - Overview of the final project requirements and expectations.
- Day 57-58: Project Development
 - Work on the capstone project, integrating learned skills.

- Day 59: Final Touches
 - Implement final touches and prepare for presentation.
- Day 60: Peer Review and Feedback
 - Present your project to peers for feedback and evaluation.

Course Objectives

- Master Python and Django for backend development.
- Develop and deploy robust APIs.
- Build a portfolio of practical projects to showcase your skills.
- Prepare for a career in backend development within the tech industry.