

KOENIG SOLUTIONS · COURSE OUTLINE

Python Full Stack Web Developer

Learn to build full stack web applications with Python, Django, REST APIs, MySQL and JavaScript, from syntax to deployment.

20 Days

DURATION

106

TOPICS

59

HANDS-ON LABS

COURSE OVERVIEW

This course begins with Python fundamentals, control structures, data structures, functions, file handling and object-oriented programming, then moves into front-end development with HTML5, CSS3 and JavaScript. It continues with Django, covering project setup, models, the ORM, admin panel, forms, authentication and a full CRUD blog/product management project. The course then builds REST APIs with Django REST Framework, integrates them with JavaScript, connects Django to MySQL, and introduces Git, GitHub and deployment. It concludes with a final capstone project combining backend, frontend, database and deployment work.

WHO THIS COURSE IS FOR

Aspiring Web Developers

Python Developers

Front-End Developers

Back-End Developers

Software Engineers

PREREQUISITES

- Knowledge of Web Application Development and Any Programming Language

TOOLS AND REQUIREMENTS

- Python
- MY SQL
- HTML
- CSS
- JS
- Django
- DRF
- GIT

What You Will Learn

- Apply Python control structures, functions and data structures
- Implement object-oriented programming concepts in Python
- Build web page structure and forms using HTML5
- Style responsive layouts with CSS3 Flexbox and Grid
- Develop interactive front-end behaviour with JavaScript
- Build Django applications with models, views and templates
- Implement authentication, permissions and REST APIs with DRF
- Deploy Django projects using Git, GitHub and MySQL

COURSE SYLLABUS

106 Topics

DAY 1 Python Fundamentals & Control Structures

TOPICS

4 Labs

- Overview of Full Stack Development
- Installing Python & IDEs (VS Code, PyCharm)
- Variables, Data Types, Type Conversion
- Input/Output, Print Formatting
- Operators (Arithmetic, Comparison, Logical)
- Conditional Statements (if-else, nested if)
- Loops (for, while, nested loops)
- range(), enumerate(), zip()

DAY 2 Data Structures & Built-in Functions

TOPICS

3 Labs

- **Lists:** operations, methods, nested lists
- **Tuples:** immutability, indexing
- **Sets:** uniqueness, set operations
- **Dictionaries:** keys/values, nesting, methods
- Membership and identity operators
- List comprehensions & dictionary comprehensions
- **Built-in functions:** len(), max(), sum(), etc.

DAY 3 Functions, Modules, File Handling

TOPICS

3 Labs

- Defining and calling functions
- **Function arguments:** default, keyword, *args, **kwargs
- Return statements, scope, global/local variables
- **File handling:** open, read, write, append, context manager
- Creating and importing modules
- **Standard libraries:** math, os, random, datetime

TOPICS

3 Labs

- Object-Oriented Programming overview
- Classes, Objects, `__init__` method
- Instance vs class variables/methods
- Inheritance, Multilevel Inheritance
- Polymorphism, Method Overriding
- **Special methods:** `__str__`, `__len__`, `__repr__`
- Exception handling (try-except-finally, raise)

TOPICS

3 Labs

- HTML document structure & DOCTYPE
- **Tags:** headings, paragraphs, images, links
- **Lists:** ordered, unordered, nested
- **Tables:** rows, colspan, rowspan
- **Forms:** input types, validation attributes
- **Semantic tags:** section, article, aside, footer

TOPICS

3 Labs

- Inline, internal, external styles
- **Selectors:** class, ID, pseudo-classes
- **Box model:** margin, padding, border
- **Layouts:** Flexbox, Grid, Positioning
- Transitions, shadows, backgrounds
- Responsive Design using Media Queries

TOPICS

3 Labs

JavaScript syntax & data types · Variables (var, let, const) · Operators and expressions · Functions, events, DOM manipulation · Conditional logic and loops · Built-in methods (Array, String)

TOPICS

3 Labs

Lambda functions, `map()`, `filter()`, `reduce()` · `zip()`, `enumerate()`, `sorted()`, `any()`/`all()` · Virtual environments using `venv` · Installing packages with `pip` · Creating `requirements.txt` · Basics of unit testing with `unittest`

TOPICS

3 Labs

What is Django? MVT vs MVC · Project & App structure · URL routing and path() · **Views:** function-based and template rendering · Templates and template inheritance · Static files management (CSS, JS)

TOPICS

3 Labs

Creating models and model fields · Running migrations · Model relationships (ForeignKey, OneToOne) · **Django admin:** registration and customization · **Django shell:** ORM operations

TOPICS

3 Labs

Django Form class · ModelForm with validation · Input widgets and customizations · CSRF protection · Messages framework for feedback

TOPICS

3 Labs

- Managing static and media files
- ImageField and FileField
- Django User model, Login, Logout, Signup
- @login_required, Sessions & context processors

TOPICS

3 Labs

Designing full CRUD project · Handling comments and likes · Using Django's Q object for search · Advanced form rendering with crispy-forms

TOPICS

3 Labs

REST principles and architecture · Installing and configuring DRF · **Serializers:** basic & ModelSerializer · Function-based API views · Testing APIs with Postman

DAY 15 DRF ViewSets, Auth & Permissions

TOPICS

3 Labs

ViewSets and Routers · Pagination, Filtering, Ordering · Token authentication · Permissions (IsAuthenticated, custom)

DAY 16 API Integration with JavaScript

TOPICS

3 Labs

Fetch API in JavaScript · GET and POST requests · Handling JSON response · Error handling and retries

DAY 17 MySQL with Django

TOPICS

3 Labs

- Installing MySQL and GUI tools (MySQL Workbench)
- Setting up MySQL database
- Django settings configuration
- Migrations in MySQL
- **ORM optimization:** select_related, prefetch_related

DAY 18 Git, GitHub & Deployment Basics

TOPICS

3 Labs

- **Git basics:** clone, commit, push, pull, branches
- Linking GitHub repository
- Managing .env files and environment variables
- Deployment on Render/Heroku/Railway

DAY 19 Final Capstone Project - Development

TOPICS

3 Labs

Project planning (Idea, Features, Models) · Setup full CRUD with Django and DRF · MySQL integration · Frontend integration (HTML/JS)

DAY 20 Final Capstone Project - Completion & Presentation

TOPICS

Lab

Code cleanup and refactoring · Final testing and QA · Documentation and README writing · Demo

Practise What You Learn

MODULE	ACTIVITY
Python Fundamentals & Control Structures	■ Age group categorizer ■ Prime number checker ■ Multiplication table generator ■ Number guessing game
Data Structures & Built-in Functions	■ Student database (dictionary of lists) ■ Product inventory manager ■ Word frequency counter in a paragraph
Functions, Modules, File Handling	■ To-do list stored in a file ■ Password generator module ■ Calendar planner using datetime
OOP in Python	■ Employee management using classes ■ Vehicle model hierarchy (inheritance) ■ Library system with exception handling
HTML5 - Page Structure & Forms	■ Personal portfolio webpage ■ Registration form with validation ■ Product listing with images in a table
CSS3 - Styling and Layout	■ Style the Day 5 portfolio page ■ Responsive pricing table ■ Flexbox-based navigation bar
JavaScript Essentials	■ Interactive contact form with validation ■ Simple calculator ■ DOM-based image gallery toggler
Python Intermediate + Virtual Environment	■ CSV reader/writer tool ■ Project scaffold with modules and venv ■ Unit tests for utility functions
Django Introduction & Project Setup	■ Setup “Quote App” with 3 pages ■ Create and render context-based views ■ Use static CSS in template
Django Models, ORM, Admin Panel	■ Product and Category model with admin interface ■ Add image field and description ■ Admin filters, search, and inline editing

MODULE	ACTIVITY
Forms, Validations & Messages	■ User signup form using ModelForm ■ Bootstrap-styled feedback forms ■ Message display after submit
Static/Media Files & Django Authentication	■ Profile uploader with media files ■ User login/logout with custom UI ■ Authenticated dashboard
Django Project - Blog/Product Management	■ Complete blog system with images ■ Product management app with categories ■ Search/filter products by name/category
REST API Basics with Django REST Framework	■ Product list and detail APIs ■ Create/update/delete API using Postman ■ Return custom messages and status codes
DRF ViewSets, Auth & Permissions	■ Secure product API with login token ■ Add filters for search ■ Create custom permission for superuser
API Integration with JavaScript	■ Fetch product list and render in HTML ■ Submit form data via JS to Django API ■ Live product search with debounce
MySQL with Django	■ Switch project from SQLite to MySQL ■ Create indexes and FK constraints ■ Performance testing of queries
Git, GitHub & Deployment Basics	■ Push Django project to GitHub ■ Create .env and hide secrets ■ Deploy Django app and access live
Final Capstone Project - Development	■ Build all backend features (API + admin) ■ Frontend forms and templates ■ Auth, session, and image uploads
Final Capstone Project - Completion & Presentation	Final commit and GitHub push

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

Classroom Training

Live Online Classes (ILO)

Fly-Me-A-Trainer (FMAT)

Flexi (Self-Paced)

1-on-1 Training

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

Visit koenig-solutions.com

✉ info@koenig-solutions.com