Full Stack Web Development with Python Django

A Recruitment Portal

Full Stack web development has increased its popularity like never before. A full stack developer needs both front-end and back-end skills. Where backend requires a lot of coding skills and front-end requires more of designing & experience perspective for a coder. For backend, we can use PYTHON. The python language is one of the most accessible programming languages available because it has simplified syntax and not complicated, which gives more emphasis on natural language. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

With this program one can get a kick-start to their career in the field of web development from Zero to Hero level. The program comprises of Basic Python Scripting, HTML/CSS/Bootstrap/JS Implementation and Django Framework for backend development.

Why should you learn from us?

- A Team built with Professional Trainers having an experience of delivering for more than 20K students
- An outreach of 300+ colleges pan India
- Associated with multiple Corporate as hiring partners
- Live Interactive Online Sessions with Animations and 360-degree views for topic explanation.
- Daily 15 minutes Query Solving Session at the end of the session.
- Availability of Recorded Sessions with Personalized Accounts for convenient revisions
- And very affordable never-before fees

The Fee structure is mentioned below:

DURATION	6 Week / 90 Hours	For JECRC Students
Course Fee	INR 12500/- + 18 % GST (INR 14,750/-)	INR 5865/-+18%GST =INR 6920/-

Who can attend this program?

- Engineering Undergraduates/Computer Programming Pursuing
- Python/Web Development Hobbyists
- Students willing to Get through with Web Development Career
- Knowledge of any computer programming is advantageous

Project Titles

Final Project (1)

"A Recruitment Portal"

Capstone Projects (3)

1. *Deploying a Django Based Application for House Price Prediction*
2. *Myntra Clone: An E-commerce website with variety of clothes*
3. *Image Blogger Web Application (Instagram/Pinterest Clone)*

Mini Projects (6)

1. *CLI based Student Registration & Login Portal*
2. *Instagram / Pinterest Home Page clone*
3. *Weather Meter Application*
4. *An online Quiz App*
5. *News Portal*
6. *Resume Builder Application*

Full Stack Web Development using Python Django

(Duration – 6 Weeks/90 Hours)

W1	**Introduction & Setup** (2.5 Hours)	• Introduction with **Python & Linux** • What is Web Development • Basic Linux/Windows Commands • Installing & Setting up Python on System • Understanding its Command Line & Scripts • Simple Python Program
	Python Data & Types (6 Hours)	• Python Variables, Rules of Identifiers • Creating, Using and Printing a Variable in Python • Taking User Inputs from Python • Some Common Functions • **Python Keywords** • Interactive Mode & Script Mode Programming • Python Comments • **Python Operators:** Arithmetic, Assignment, Relational, Logical, Bitwise, Identity, Membership • **Python Braces:** (), {}, [] • **Python Data Types/Classes:** int, float, complex, str, list, tuple, dict, set, bool • **Numeric Data Types:** Creation, Modification, Use of Operators, Built-in Functions • Concept of Mutability • **String Data:** Creation, Accessing, Modification*, Use of Operators, Built-in Functions, Use of Triple Quotes • **List:** Creation, Accessing, Modification, Use of Operators, Built-in Functions • **Tuple:** Creation, Accessing, Modification, Use of Operators, Built-in Functions • **Dictionary**: Creation, Accessing, Modification, Use of Operators, Built-in Functions • Type Casting in Python • **Set**: Creation, Accessing*, Modification, Use of Operators, Built-in Functions • **Boolean**: Creation, Use of Operators • Some Common Functions
	Control Statements in Python (2.5 Hours)	• *Concept of Indentation in Python* • *Python Suites: Simple, Nested, Ladder, Hybrid* • *Conditional IF: Use of if-elif-else in Python*
	Python Loops (2 Hours)	• **WHILE**: Use of while statement in Python • *FOR Loops: The for-loop behavior in Python* • *Using range() function* • *Use of break and continue* • *Some Common Functions*
	Comprehensions (2 Hours)	• Writing Python IF in one line • List Comprehension • Use of Conditionals • Nested Comprehensions • Shortcuts to Create Tuples • Dictionary Comprehension

		• Shortcuts to Create Dictionary • Set Comprehension • Tuple/? Comprehension** • Some Common Functions
W2	**User-defined Functions** (4 Hours)	• Defining and calling a custom user defined Function • Types of Functions by parameters/arguments • Default Arguments • Arguments Skipping • Variable Length Arguments *args • Keyword Arguments **kwargs • Passing Functions as Arguments • Recursion • **Anonymous Functions:** Lambda Expressions • Use of map(), filter(), reduce() • **Generators**: Use of Generators, Significance • Creating Custom Generator • Comprehension • Some Common Functions
	Modules & Packages (2 Hours)	• Python Library & Packages • Using Modules by: import, from-import, creating alias • Downloading a new module • Installing a module • Creating User-defined modules • Some Commonly used Modules in Python: time, os, datetime, calendar, math, random • Finding System Time, Showing Calendars • Generating Random Numbers • Performing Typical Mathematical Operations • Creating files, directories • Basics of File Handling
	Errors & Exception Handling (1 Hours)	• Assertions in Python • Errors and Exceptions • Exception Handling: Try, except, finally • User Defined Exceptions
	Database in Python (3 Hours)	• Connecting an SQL based Database • Setting up PostgreSQL Engine • SQL Queries: CRUD Operations, Joining, Ordering, Grouping, Filtering • Basic SQL Operations using python
	MINI Project (1 Hour)	• *Project 1: CLI based Student Registration & Login Portal*
	HTML (4 Hours)	• What is browser: Introduction • Editors • HTML PAGE STRUCTURE • HTML File Save/Open/Modification • What is HTML Elements (Tags and Attributes)

		• Tags: <!DOCTYPE html>, <Meta Tag>, <html>, <head>, <title>, <body>, <heading>, <paragraph>, <Line break> • ***Style Tag*** • HTML Links, Lists, Images, File Paths, Classes, id • Iframes, Table, Div, Span, form, input
W3	**HTML** **(2.5 Hours)**	• Canvas, svg, audio, video
	CSS **(2.5 Hours)**	• Introduction, Syntax • Selectors • Comments, • Colors: color, RGB, HEX, HSL • Backgrounds: Color, Image • Borders: width, color, side • Padding • Height/Width • Text: color, alignment, decoration • Fonts: family, style, size • Display: max-width, position, z-index, overflow, float • Clear • Inline-Block • Align • Templates
	Bootstrap **(2.5 Hours)**	• Introduction with Bootstrap • The responsive web pages • Buttons, Forms, Navbars, Gri*ds* • *Bootstrap Plugins*
	JavaScript **Basics (5 Hours)**	• Introduction • File Open, Save, Modify • Using JS file with HTML; Syntax • Data Type and Variables • let, var, const, prompt, confirm, alert • ***Operators:*** Assignment, Arithmetic, Conditional, Logical, Ternary • ***Conditional & Looping Statements*** • ***Arrays*** • Document Object Modelling • Introduction with JQuery • Modifying properties of HTML using JS and JQuery
	Mini Project **(2.5 Hours)**	• ***Project 2: "Instagram / Pinterest Home Page clone"***
W4	**Django** **(2.5 Hours)**	• What is Django • Django installation • Creating First Django Project • Running a Django Application

	Django Template Views (2.5 Hours)	• Django architecture • MVC and MTV • Starting a project • Django apps • Activating our first app • A view that displays a hyperlink • Mapping the views to URLs • Running our first app • Improving the views using templates • The improved templates • Template inheritance • Sending data from url to view • Sending data from view to template
	Mini Project (1.5 Hours)	• *Project 3: Weather Meter Application*
	Django Models (2.5 Hours)	• Django Models • Connecting Postgre/MySQL/SQLite with Django • Models-Templates-Views Paradigm
	Django Forms (2.5 Hours)	• Django Forms • Form Basics Code • Form Validation • Model Forms
	Mini Project (1.5 Hours)	• *Project 4: An online Quiz Application*
	Mini Project (2 Hours)	• *Project 5: News Portal*
W5	**Django Special Features** (2.5 Hours)	• Relative URLs with Templates • URL Template Inheritance • Configuring Email with Django
	Django User Management (2.5 Hours)	• User Models • Registrations; Logins • Django Passwords • Sessions & Cookies
	Project (2.5 Hours)	• *Deploying a Django Based Application for House Price Prediction*
	Mini Project (1 Hours)	• *Project 6: Resume Builder Application*
	Class based Views (2.5 Hours)	• What is CBV in Django • Template views with CBVs • Detail view and list view • CRUD Views
	Restful APIs (1.5 Hours)	• Introduction with Django API • Creating Restful APIs using Django

W6	**Django Administration** (2.5 Hours)	• Admin of Django & Templates • Sorting, Searching, Filtering • Adding Fields; Editable List View
	Capstone Project (5 Hours)	• *An Ecommerce Platform (Myntra Clone)*
	Project (5 Hours)	• *My Blogging Platform (Like Instagram/Pinterest)*
	FINAL PROJECT (10 Hours)	*"Recruitment Portal"*