

Course Title: Full Stack Python Development

Course Description:

This course offers a comprehensive introduction to full stack development using Python as the core language for backend development. Students will learn to integrate Python with modern front-end technologies and frameworks to create complete web applications.

Course Objectives:

- Understand the essentials of web development.
- Develop proficiency in Python for backend programming.
- Learn front-end technologies such as HTML, CSS, and JavaScript.
- Design, build, and deploy a full-stack web application.

Prerequisites:

- Basic knowledge of programming concepts.
- Some familiarity with Python or another high-level programming language is helpful but not required.

Weekly Syllabus Outline:

Week 1: Introduction to Full Stack Development

- What is full stack development?
- Overview of the Python programming language.
- Setting up the development environment (Python, pip, virtual environments).

Week 2: HTML and CSS Basics

- HTML5 fundamentals: Elements, attributes, and document structure.
- CSS for styling and layout techniques.
- Introduction to Bootstrap for responsive design.

Week 3: JavaScript and the Front-End

- Basics of JavaScript: Syntax, variables, and control structures.
- DOM manipulation and event handling.
- Introduction to AJAX for asynchronous web applications.

Week 4: Deep Dive into Python

- Advanced Python concepts (comprehensions, decorators, generators).
- Introduction to object-oriented programming in Python.

Week 5: Python Web Frameworks: Flask

- Introduction to Flask.
- Building simple web applications with Flask.
- Routing and template rendering.

Week 6: Working with Databases

- Database basics: SQL vs NoSQL.
- Integrating Python applications with SQLite.
- Using SQLAlchemy as an ORM tool.

Week 7: Python Web Frameworks: Django

- Introduction to Django.
- Django's architecture and design philosophies.
- Building models and handling migrations.

Week 8: Advanced Django

- Admin interface customization.
- User authentication and authorization.
- Form handling and validation with Django.

Week 9: JavaScript Frameworks: Introduction to React (Optional)

- Basics of React and its ecosystem.
- Creating dynamic front-ends using React.
- Integrating React with a Django backend.

Week 10: REST APIs with Django REST Framework

- Designing RESTful APIs with Django REST Framework.
- Serializers and class-based views.
- Authentication and permissions in APIs.

Week 11: Testing in Python

- Unit testing with pytest.
- Functional testing with Selenium.

- Testing Django applications.

Week 12: Deployment and Production

- Deploying Python applications to platforms like Heroku and AWS.
- Using Docker containers for deployment.
- Environment management and security best practices.

Week 13: Advanced Topics and Best Practices

- Celery for background task processing.
- Caching strategies for web applications.
- WebSockets for real-time communication.

Week 14: Capstone Project

- Project initiation and planning.
- Implementing full stack features.
- Peer reviews and iterative development.

Week 15: Final Presentations and Course Review

- Student project demonstrations.
- Review of key topics and final Q&A.
- Course feedback and next steps.

Assessment Methods:

- Weekly hands-on coding assignments.
- Quizzes to reinforce critical concepts.
- A comprehensive capstone project involving all aspects of full-stack development.