

Course Title: Python Full Stack Development Course Syllabus

Course Description: This course provides a comprehensive overview of Full Stack Development using Python, covering both frontend and backend technologies. Students will learn how to build dynamic and interactive web applications using Python for backend development and modern frameworks for frontend development. The course includes topics such as Python programming, Django framework, HTML/CSS, JavaScript, React.js, database integration, and deployment strategies.

Prerequisites: Basic understanding of programming concepts and web development. Familiarity with Python programming language is helpful but not required.

Course Objectives:

1. Understand the principles and technologies involved in Full Stack Development using Python.
2. Learn how to develop backend applications using Python and Django framework.
3. Gain proficiency in building responsive and interactive user interfaces using HTML/CSS, JavaScript, and React.js.
4. Develop skills in integrating frontend and backend components to create full-stack web applications.
5. Explore advanced topics such as RESTful APIs, database management, and deployment strategies.

Course Outline:**1. Introduction to Full Stack Development with Python**

- Overview of Full Stack development
- Frontend vs. Backend development
- Introduction to Python and JavaScript

2. Python Programming Basics

- Variables, data types, and operators
- Control structures (if-else, loops)
- Functions and modules

3. Introduction to Django Framework

- Overview of Django framework
- Django project structure and setup
- Model-View-Template (MVT) architecture

4. Building Web Applications with Django

- Creating Django models and database migrations

- Developing views and templates
- Handling URL routing and HTTP requests

5. **RESTful APIs with Django Rest Framework**

- Building RESTful APIs using Django Rest Framework
- Serializers for data serialization and deserialization
- Authentication and permission handling

6. **Frontend Development with HTML/CSS**

- Introduction to HTML for web markup
- Styling web pages using CSS
- Responsive web design principles

7. **JavaScript Fundamentals**

- Introduction to JavaScript programming language
- DOM manipulation and event handling
- Asynchronous JavaScript with Promises and async/await

8. **Introduction to React.js**

- Overview of React.js library
- Setting up React.js development environment
- Components and props in React.js

9. **State Management in React.js**

- Managing state with useState and useContext hooks
- Redux for state management in larger applications
- React Router for client-side routing

10. **Integration of Frontend and Backend**

- Consuming RESTful APIs from frontend applications
- Handling CORS (Cross-Origin Resource Sharing)
- Implementing data exchange between frontend and backend

11. **Database Integration**

- Connecting Django applications to databases (SQLite, PostgreSQL)
- Performing CRUD operations using Django ORM

12. **Deployment and DevOps Practices**

- Deployment strategies for Full Stack Python applications
- Continuous Integration and Continuous Deployment (CI/CD)
- Docker containers for application packaging and deployment

Assessment:

- Weekly assignments to reinforce learning concepts.
- Midterm project: Developing a backend API using Django and a simple frontend using HTML/CSS.
- Final project: Designing and implementing a Full Stack Python web application using Django for backend and React.js for frontend.

Textbook: "Django for Beginners" by William S. Vincent

Additional Resources:

- Online tutorials and documentation (Django official documentation, React.js official documentation, etc.).
- Supplemental readings and materials provided by the instructor.

Grading:

- Assignments: 30%
- Midterm Project: 20%
- Final Project: 40%
- Participation and Attendance: 10%

Attendance Policy: Regular attendance is expected. Students are allowed a maximum of three unexcused absences. Excessive absences may result in a reduction of the final grade.

Office Hours: Instructor office hours will be held twice a week for additional help and clarification.

Csdtdo