

Below is a comprehensive syllabus for a Full Stack .NET course covering both frontend and backend development:

### **Course Title: Full Stack .NET Development**

**Course Description:** This course provides a comprehensive overview of Full Stack .NET development, covering frontend and backend technologies. Students will learn how to build dynamic and interactive web applications using the .NET framework for backend development and modern frontend frameworks such as React.js or Angular for frontend development. The course includes topics such as C# programming, ASP.NET Core, Entity Framework, SQL Server, JavaScript, TypeScript, and deployment strategies.

**Prerequisites:** Basic understanding of programming concepts and web development. Familiarity with C# programming language and HTML/CSS is helpful but not required.

### **Course Objectives:**

1. Understand the principles and technologies involved in Full Stack .NET development.
2. Learn how to develop backend applications using .NET Core and ASP.NET Core.
3. Gain proficiency in building responsive and interactive user interfaces using frontend frameworks like React.js or Angular.
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:**

#### **Module 1: Introduction to Full Stack .NET Development**

- Overview of Full Stack development
- Frontend vs. Backend development
- Introduction to .NET ecosystem

#### **Module 2: C# Programming Basics**

- Variables, data types, and operators
- Control structures (if-else, switch, loops)
- Object-oriented programming concepts

**Module 3: Introduction to ASP.NET Core**

- Overview of ASP.NET Core framework
- Creating and configuring ASP.NET Core projects
- Understanding MVC (Model-View-Controller) architecture

**Module 4: Building Web APIs with ASP.NET Core**

- Creating RESTful APIs using ASP.NET Core
- Handling HTTP requests and responses
- Authentication and authorization with JWT

**Module 5: Database Integration with Entity Framework Core**

- Introduction to Entity Framework Core
- Connecting ASP.NET Core applications to SQL Server
- Performing CRUD operations using Entity Framework Core

**Module 6: Frontend Development with React.js or Angular**

- Overview of React.js or Angular framework
- Setting up development environment
- Components and data binding

**Module 7: State Management in React.js or Angular**

- Managing state with Redux (for React.js) or RxJS (for Angular)
- Implementing routing and navigation

**Module 8: Authentication and Authorization**

- Implementing user authentication and authorization in ASP.NET Core
- Integrating authentication with frontend applications

**Module 9: Advanced Topics in .NET Development**

- Testing and debugging ASP.NET Core applications
- Performance optimization techniques
- Error handling and logging best practices

**Module 10: Deployment and DevOps Practices**

- Deployment strategies for Full Stack .NET applications

- Continuous Integration and Continuous Deployment (CI/CD)
- Containerization with Docker and deployment on platforms like Azure

**Module 11: Real-world Projects and Case Studies**

- Working on real-world projects and case studies
- Designing and implementing end-to-end Full Stack .NET solutions
- Presenting findings and insights from projects

**Module 12: Capstone Project**

- Developing a comprehensive Full Stack .NET project
- Identifying a business problem or scenario
- Designing and implementing a solution using skills learned throughout the course

**Assessment:**

- Weekly assignments to reinforce learning concepts.
- Midterm project: Developing a backend API using ASP.NET Core and a simple frontend using React.js or Angular.
- Final project: Designing and implementing a Full Stack .NET web application using ASP.NET Core for backend and React.js or Angular for frontend.

**Textbook:** "Pro ASP.NET Core MVC" by Adam Freeman

**Additional Resources:**

- Online tutorials and documentation (Microsoft documentation, ASP.NET Core documentation, React.js or Angular 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.

CsdT Centre