Curriculum Overview

Title: Introduction to Web Technology

Level: Beginner

Total Semesters: 3

Course Duration (Years): 2

Industry Focus: Web Development

Assessment Type: Mixed

Semester-wise Breakdown

Semester 1: Foundations of Web Development

Subjects:

- Introduction to HTML5 and CSS3
- Basics of Web Structure & Design
- Understanding the Document Object Model (DOM)
- Introducing JavaScript Fundamentals
- Building Simple Web Pages
- Responsive Web Design Principles

Projects:

- Creating a personal website using HTML5, CSS3, and Bootstrap.
- Interactive quizzes using JavaScript to enhance user engagement.

Key Outcomes:

- Competent in creating basic web pages using HTML5 and CSS3.
- Ability to understand and manipulate the Document Object Model (DOM).

- Foundational knowledge of JavaScript for web development.

- Experience with responsive design principles.

Semester 2: Intermediate Web Development Techniques

Subjects:

- Advanced JavaScript (ES6 features, DOM manipulation)

- Introduction to Web APIs (Fetch API, AJAX)

- Understanding of SEO and Metadata

- Building Dynamic Web Applications

- Introduction to Version Control with Git

- Interactive projects using Node.js and Express.js

Projects:

- Developing a simple e-commerce website with product listings, user authentication, and a basic shopping cart.

- Implementing a weather application that fetches data from an external API.

- Building a CRUD (Create, Read, Update, Delete) application for managing blog posts.

Key Outcomes:

- Proficient use of advanced JavaScript concepts like ES6 features and DOM manipulation.

- Skills in integrating web APIs for dynamic content.

- Understanding of search engine optimization (SEO).

- Experience with server-side programming using Node.js, Express.js, and MongoDB.

- Familiarity with version control using Git.

Semester 3: Web Development Specialization & Capstone Project

Subjects:

- Front-end frameworks (React or Vue.js)

- Introduction to Back-end frameworks (Express.js or Django)

- Database Management (MongoDB, PostgreSQL)

- Testing and Debugging in Web Applications

- Modern Web Performance Optimization

- Building a comprehensive Capstone Project

Projects:

- Developing a full-fledged web application such as a social media platform, an online forum, or a project management tool utilizing front-end (React/Vue.js) and back-end (Express.js/Django) technologies.

- Implementing robust testing strategies for web applications using tools like Jest or PyTest.

- Optimizing the application's performance using modern techniques and tools.

Key Outcomes:

- Proficiency in one of the front-end frameworks (React, Vue.js).

- Strong back-end framework knowledge (Express.js, Django) for handling server-side operations.

- Expertise in database management systems like MongoDB or PostgreSQL.

- Ability to write comprehensive tests and debug web applications effectively.

- Completion of a real-world web development project that integrates front-end, back-end, and database technologies.

Tools & Technologies

- HTML5, CSS3 (Bootstrap)

- JavaScript ES6

- DOM Manipulation

- AJAX/Fetch API

- Git & GitHub

- Version Control Concepts

- Responsive Design Principles

- Node.js & Express.js

- MongoDB or PostgreSQL

- React or Vue.js Frameworks

- Testing Frameworks (Jest, PyTest)

- Web Performance Optimization Tools

Career Opportunities

- Junior Web Developer

- Front-end/Back-end Developer

- Full Stack Developer

- Web Application Tester

- UI/UX Designer (with coding skills)

- E-commerce Developer

- Content Management System (CMS) Specialist

- Digital Marketing Specialist with web development skills

- Consultant for web technology solutions in various industries

Learning Outcomes

Upon completion of this curriculum, students will be capable of:

1. Designing and developing dynamic web applications using HTML5, CSS3, JavaScript, and modern frameworks (React/Vue.js).

2. Implementing server-side operations with Node.js, Express.js, or Django.

3. Managing databases efficiently using MongoDB or PostgreSQL.

4. Applying responsive design principles for cross-platform compatibility.

5. Integrating APIs for enhancing web application functionality (e.g., fetching weather data).

6. Writing comprehensive tests and debugging web applications.

7. Optimizing web application performance for faster load times and better user experience.

8. Collaborating effectively on development projects, utilizing version control systems like Git.