

Course title:	Web Engineering		
Course Code:	SEEC-304	Credit Hrs:	03
Division:		Theory:	03
Hours for meeting students (Contact Hours): 3			Practical: 0

## 1. Course Introduction/Objectives

**Web Engineering** is a multidisciplinary field focused on the systematic development, deployment, and maintenance of web-based applications. This course covers foundational topics in frontend and backend development, modern web practices, and essential tools and techniques. It emphasizes performance, usability, security, and emerging trends like AI integration and Progressive Web Applications (PWAs). Students will gain hands-on experience in building, testing, and managing scalable and responsive web solutions.

## 2. Course/Learning Outcomes

After completion of this course, the students are expected to be able to demonstrate the following knowledge, skills, and attitudes:

By the end of this course, students will be able to:

1. Understand the fundamentals of web technologies and how they impact software development. (Bloom's Taxonomy Level: 2 (Understand))
2. Identify and address constraints faced by web developers. Bloom's Taxonomy Level: 3 (Apply)
3. Design and build a basic, functional web application using modern tools. (Bloom's Taxonomy Level: 6 (Create))
4. Evaluate web applications for usability, performance, and security. (Bloom's Taxonomy Level: 5 (Evaluate))
5. Understand modern web trends, such as responsive design, cloud-based applications, and web accessibility. (Bloom's Taxonomy Level: 2 (Understand))

## 3. COURSE OUTLINE

### 1. Introduction to Web Engineering:

- Overview of Web Engineering and its role in modern software development.
- **Categories of Web Applications** (e.g., static, dynamic, single-page applications).
- **Characteristics of Web Applications**: Scalability, security, and responsiveness.
- **Evolution of Web Technologies**: From Web 1.0 to Web 3.0, and how this affects development practices.

### 2. Frontend Development Foundations:

- **HTML and CSS Basics**: Structure and styling of web pages.
- **Responsive Web Design**: Mobile-first approach, using CSS Grid and Flexbox for layouts.
- **Introduction to JavaScript**: Adding interactivity and handling events.
- **ES6+ JavaScript Features**: Arrow functions, async/await, modules, and modern practices.
- **CSS Frameworks** (e.g., Bootstrap): Rapid UI design and mobile responsiveness.
- **Accessibility Basics**: Designing for all users with WCAG guidelines.

### 3. Backend Development Fundamentals:

- **Basics of Server-Side Programming**: Node.js/PHP
- **Databases**: Relational databases (e.g., MySQL, PostgreSQL) vs. NoSQL databases (e.g., MongoDB, Firebase).
- **Setting up Web Servers**: Deploying a simple web server and integrating databases.
- **Authentication**: User login systems (JWT, OAuth).
- **APIs**: Introduction to RESTful APIs, consuming APIs, and building custom APIs.

### 4. Modern Web Development Practices:

- **Version Control**: Introduction to Git, GitHub, and best practices for source code management.
- **CI/CD**: Continuous Integration and Continuous Deployment pipelines.
- **Deployment Platforms**: Overview of platforms like Netlify, Vercel, and Heroku.
- **Containerization**: Introduction to Docker for managing application environments.
- **Progressive Web Applications (PWAs)**: Basics of building apps that work offline and on low-quality networks.
- **Cloud Hosting and CDNs**: Using cloud services and Content Delivery Networks for scaling applications.

### 5. Requirements Engineering for Web Applications:

- **Fundamentals of Requirements Engineering (RE)**: Collecting and analyzing user needs.
- **Adapting RE to Web Applications**: Specific challenges and considerations for web-based systems.

- **Tools and Techniques:** Using JIRA, Confluence, and Agile methodologies (Scrum, Kanban).

## 6. Web Application Architectures:

- **Layered Architecture:** Separating concerns and maintaining scalable systems.
- **Microservices Architecture:** Designing independent, loosely-coupled services.
- **Serverless Architectures:** Introduction to serverless computing (e.g., AWS Lambda, Google Cloud Functions).

## 7. AI-Powered Web Applications:

- **Introduction to Machine Learning and AI:** How AI can enhance web apps (e.g., recommendations, predictions).
- **Using Pre-Trained AI Models:** Integrating AI-driven features such as image recognition and natural language processing.
- **TensorFlow.js:** Running machine learning models directly in the browser.
- **Building an AI-powered Chatbot:** Integrating a chatbot for web interaction.

## 8. Testing Web Applications:

- **Test Methods and Techniques:** Unit testing, integration testing, and end-to-end testing.
- **Test-Driven Development (TDD):** Encouraging writing tests before coding.
- **Automation Tools:** Jest, Mocha (for unit testing) and Cypress, Selenium (for end-to-end testing).
- **Usability Testing:** A/B testing, heatmaps, and user feedback for UI improvements.

## 9. Web Project Management:

- **Managing Web Development Teams:** Roles, collaboration, and communication tools (e.g., Slack, Trello).
- **Client Management:** Handling scope, expectations, and feedback.
- **Project Management Methodologies:** Agile (Scrum, Kanban), Waterfall, and Hybrid approaches.

## 10. Usability of Web Applications:

- **Usability Design Guidelines:** WCAG 2.1, mobile usability, and internationalization.
- **Usability Testing:** Methods like user surveys, A/B testing, and focus groups.
- **Trends in Usability:** Current UX/UI trends and standards.

## 11. Performance of Web Applications:

- **Performance Metrics:** Page load time, Time to Interactive (TTI), First Contentful Paint (FCP), and Web Vitals.
- **Optimization Techniques:** Minimizing HTTP requests, optimizing images, lazy loading, caching strategies.
- **Mobile Performance:** Optimizing for mobile networks and devices.

## 12. Security for Web Applications:

- **Common Web Security Threats:** Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), SQL Injection.
- **Encryption and Secure Communication:** SSL/TLS, HTTPS, and secure APIs.
- **Digital Signatures and Certificates:** Implementing secure authentication and digital communication.
- **Web Application Firewalls (WAF):** Protecting against common vulnerabilities.

## 13. Semantic Web:

- **Introduction to the Semantic Web:** Linked Data, RDF, SPARQL, and their relevance to web development.
- **Web3 Technologies:** Blockchain, decentralized apps (DApps), and the future of the web.

# Weekly Plan

Sr.#	Week	Chapter/Unit	Topics/Concepts covered	Quiz/ Activity/Assignments
1	Week 1		<ul style="list-style-type: none"> <li>- Overview of Web Engineering</li> <li>- Categories of Web Applications</li> <li>- Characteristics of Web Applications</li> <li>- Evolution of Web Technologies (Web 1.0 to Web 3.0)</li> </ul>	Quiz 1 on Web Engineering Concepts
2	Week 2		<ul style="list-style-type: none"> <li>- Deep dive into Web Application Categories</li> <li>- Characteristics: Scalability, security, and responsiveness</li> </ul>	Assignment 1: Web Application Comparison
3	Week 3		<ul style="list-style-type: none"> <li>- HTML and CSS Basics: Structure and styling</li> </ul>	Quiz 2 on HTML, CSS Basics

			- Responsive Web Design: CSS Grid, Flexbox	
4	Week 4		- Introduction to JavaScript: Handling events - ES6+ Features: Arrow functions, async/await, etc.	Assignment 2: Build a Responsive Web Page
5	Week 5		- CSS Frameworks (Bootstrap) - Accessibility Basics: WCAG guidelines	Activity: Group work on Responsive Design
6	Week 6		- Server-Side Programming: Node.js/PHP basics - Introduction to Databases: Relational vs. NoSQL	Quiz 3 on Backend Fundamentals
7	Week 7		- Setting up a Web Server - Integrating Databases - Authentication: JWT, OAuth	Assignment 3: Build a Simple Web App with Authentication
8	Week 8		- APIs: RESTful API Introduction - Consuming APIs	Activity: API Integration Project
9	Week 9		- Introduction to Git, GitHub - CI/CD pipelines	Mid-Term Exam
10	Week 10		- Deployment Platforms: Netlify, Vercel, Heroku - Containerization: Docker	Activity: Set up a project on Netlify/Vercel
11	Week 11		- Progressive Web Apps (PWAs): Basics - Cloud Hosting and CDNs	Assignment 4: Build a PWA
12	Week 12		- Fundamentals of RE - Adapting RE to Web Applications	Quiz 4 on RE and Web App Challenges
13	Week 13		- Layered Architecture - Microservices and Serverless Architectures	Activity: Discuss Web Architecture Case Studies
14	Week 14		- Introduction to Machine Learning and AI - Using Pre-Trained AI Models - TensorFlow.js	Assignment 5: AI-Powered Web Application
15	Week 15		- Common Web Security Threats (XSS, CSRF, SQL Injection) - Encryption, SSL/TLS, WAFs	Activity: Implement Web Security Best Practices
16	Week 16		- Review of all topics covered - Final preparation for the final exam	Final Term Exam

### Assignments/Activities/Quizzes

Sr.No	Week	Title	Description	Assignments/Presentation/Quizzes
1.	Week 1	Quiz 1: Web Engineering Concepts	Introduction to the course and basic web engineering concepts.	Quiz on Web Engineering Overview, Categories, and Evolution
2.	Week 2	Assignment 1: Web Application Comparison	Compare different types of web applications (static, dynamic, SPA, etc.).	Assignment on Web Application Categories and Characteristics
3.	Week 3	Quiz 2: HTML & CSS Basics	Assessment on HTML structure and basic CSS styling.	Quiz on HTML and CSS Basics
4.	Week 4	Assignment 2: Build a	Create a responsive webpage using HTML, CSS Grid, and Flexbox.	Assignment on Responsive Web Page Design

		Responsive Web Page		
5.	Week 5	Activity: Group Work on Responsive Design	Group project to create a responsive design using Bootstrap and CSS frameworks.	Group Activity on Designing a Responsive Web Page
6.	Week 6	Quiz 3: Backend Fundamentals	Quiz on backend concepts like server-side programming and database types.	Quiz on Node.js/PHP, Databases, and Authentication Basics
7.	Week 7	Assignment 3: Build a Simple Web App with Authentication	Develop a basic web app with user login and authentication using JWT/OAuth.	Assignment on Web App with Authentication
8.	Week 8	Activity: API Integration Project	Integrate external APIs into a web application (e.g., weather data, maps).	Activity on API Integration
9.	Week 9	Mid-Term Exam	Covers topics from weeks 1 to 8, including frontend and backend fundamentals.	Mid-Term Exam on all course topics until Week 8
10.	Week 10	Activity: Set Up Project on Deployment Platforms	Deploy a project on platforms like Netlify, Vercel, or Heroku.	Activity on Deployment using Netlify/Vercel/Heroku
11.	Week 11	Assignment 4: Build a PWA	Develop a Progressive Web Application (PWA) that works offline.	Assignment on Building a Progressive Web Application
12.	Week 12	Quiz 4: RE and Web App Challenges	Quiz on Requirements Engineering (RE) and adapting RE to web applications.	Quiz on RE Concepts, Tools, and Techniques for Web Apps
13.	Week 13	Activity: Discuss Web Architecture Case Studies	Group discussion on Layered, Microservices, and Serverless architectures.	Activity on Web Architecture Case Studies
14.	Week 14	Assignment 5: AI-Powered Web Application	Build an AI-powered web application using TensorFlow.js or pre-trained models.	Assignment on AI Integration in Web Applications
15.	Week 15	Activity: Implement Web Security Best Practices	Implement common security practices (e.g., SSL/TLS, WAF) in a sample web app.	Activity on Web Security Implementation
16.	Week 16	Final Term Exam	Final comprehensive exam covering all topics.	Final Exam on all course topics

\*Mode is Physical/Individual/Group etc.

#### 4. Reference Books

- HTML and CSS: Design and Build Websites by Jon Duckett.
- JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett.
- Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke.
- Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics by Jennifer Robbins.
- Web Engineering: The Discipline of Systematic Development of Web Applications by Gerti Kappel et al.

#### 5. Evaluation / Assessment Scheme

**Mid, Sessional and Final:** Sessional includes as mentioned in clause 10.1 of semester regulations. i.e,

test(s)/quiz(s)/assignments/presentation(s)/seminar(s)/, class participation/attendance and term paper.

Sr.#	Category				
	Midterm	Final Term	Assignments/presentation	Quiz(s)	Total
<b>Marks</b>	30	50	10	10	100
<b>Weightage</b>					

Evaluation Criteria	Marks
Midterm Exam	<b>30</b>
Final Exam	<b>50</b>
Quiz / Assignment/ Activities	<b>18</b>
Attendance*	<b>2</b>
<b>Total</b>	<b>100</b>