# WEB APPLICATION PROGRAMMING

**ENCT 302**

**Lecture : 3** **Year : III**
**Tutorial : 1** **Part : I**
**Practical : 3**

**Course Objectives:**
The objective of this course is to provide comprehensive understanding of web technologies for developing responsive and interactive web applications. The course emphasizes the use of modern front-end frameworks and tools, as well as techniques for integrating front-end interfaces with back-end APIs and databases. By the end of the course, students will be able to design, implement and evaluate contemporary applications while exploring current trends and future directions in web development.

**1 Introduction (6 hours)**

- 1.1 Overview of web applications and evolution of web architecture
- 1.2 Client-server architecture, HTTP, HTTPS, URLs, DNS, web browsers
- 1.3 Html basics: Syntax, tags, attributes, forms and inputs, tables, lists, multimedia, semantic HTML5 elements
- 1.4 CSS basics: Selectors, properties, values, box model
- 1.5 CSS framework: Bootstrap

**2 JavaScript and Client-Side Programming (12 hours)**

- 2.1 JavaScript essentials
  - 2.1.1 Data types, variables, control structures, functions
  - 2.1.2 Dom manipulation, events, and form validation
  - 2.1.3 Local storage and session storage
  - 2.1.4 GUI interactions
  - 2.1.5 JavaScript library: jQuery
- 2.2 Modern JavaScript (ES6+)
  - 2.2.1 Arrow functions, destructuring, spread/rest operators
  - 2.2.2 Callbacks, promises, async/await
  - 2.2.3 Modules and imports
- 2.3 Client-side applications
  - 2.3.1 Client-side web application development using React JS library
  - 2.3.2 Traditional multi-page apps vs. single-page apps (SPAs)
  - 2.3.3 Component-based UI and props
  - 2.3.4 State management and data flow
  - 2.3.5 Client-side routing and navigation without reloads
  - 2.3.6 Fetch API: Fetching, displaying data, handling errors and loading states

2.3.7 Comparison of modern JavaScript libraries: React, Angular, Vue

**3 Server-Side Web Programming (9 hours)**

3.1 MVC architecture in web development
3.2 Role of backend in web applications
3.3 Backend web framework: Django
3.4 Handling requests and responses
3.5 Form data handling and sessions
3.6 Routing, middleware, templating concepts
3.7 Overview and comparison of backend frameworks: Django, flask, fastapi, dot NET MVC framework, ruby on rails, java spring boot, node.js
3.8 Database integration: Relational versus NoSQL, CRUD operations, ORM concept
3.9 Authentication and authorization: Cookies, sessions, JWT
3.10 Middleware for logging, error handling, and security

**4 Web Services and APIs (7 hours)**

4.1 API basics: Role in web applications
4.2 REST principles and design, RESTful APIs
4.3 JSON versus XML data exchange
4.4 Data validation and serialization
4.5 Microservices

**5 Web Application Security (6 hours)**

5.1 Common vulnerabilities: XSS, SQL injection, CSRF
5.2 Security best practices: Input validation, sanitization, https, secure cookies, env variables
5.3 Authentication practices and token handling
5.4 Security in full-stack apps: CORS, safe sessions

**6 Web Application Deployment and Modern Trends (5 hours)**

6.1 Full-stack development
6.2 Testing and QA
6.3 DevOps, continuous integration (CI) and continuous delivery (CD)
6.4 Progressive web apps (PWAs), responsive design and usability

**Tutorial (15 hours)**

1. Walkthrough of client-server request flow (DNS→HTTP/HTTPS→ Browser)
2. Guided coding: HTML page with semantic tags, styled with CSS grid/flexbox
3. JavaScript and client-side programming
4. Hands-on exercises with DOM manipulation, event handling
5. Form validation and local storage demo

6. Guided task: Small SPA with component-based UI and fetch API integration
7. Discuss request/response cycle with server framework
8. Guided coding: Simple CRUD using Django/Flask (e.g., library system)
9. Session handling and authentication example
10. REST principles with examples
11. Hands-on exercise: Build a simple REST API endpoint and test with fetch/postman
12. Demonstration of XSS and SQL Injection vulnerabilities
13. Guided exercise: Secure login form with input validation and session/token handling
14. Discussion on PWAs, serverless, microservices
15. Case study/tutorial: Compare a traditional app vs. PWA features

**Practical** **(45 hours)**

1. Build a responsive personal portfolio page using semantic HTML5, CSS grid/flexbox, and media queries
2. Create a dynamic to-do list app with DOM events, localstorage, and JavaScript event handling (using jQuery)
3. Develop a CRUD application (e.g., library system) using a server-side framework (Django/Flask/FastAPI)
4. Create a frontend in React that fetches data from your backend API and displays it dynamically.
5. Extend the CRUD app with user authentication (Login/logout), form validation, and secure token/session handling
6. Develop a comprehensive project that encompasses the complete process of front-end and back-end web application development and deployment. The project should be carried out over at least six to seven lab sessions, covering the following stages: Project proposal, front-end UI/UX design, back-end and database design, API design and integration, testing and deployment, and culminating in a final presentation and demonstration

**Final Exam**

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

| Chapter | Hours | Marks distribution* |
|---|---|---|
| 1 | 6 | 8 |
| 2 | 12 | 16 |
| 3 | 9 | 12 |
| 4 | 7 | 9 |
| 5 | 6 | 8 |
| 6 | 5 | 7 |
| Total | 45 | 60 |

* There may be minor deviation in marks distribution.

**References**


1. Copes, F. (2019). The JavaScript Handbook. (https://flaviocopes.com/the-javascript-handbook-2019-edition/)
2. Nixon, R. (2025). Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites. O'Reilly Media.
3. Vincent, W. S. (2025). Django for Beginners: Build websites with Python and Django.
4. Richardson, L., & Amundsen, M. (2013). RESTful Web APIs: Services for a Changing World. O'Reilly Media.
5. Hoffman, A. (2024). Web Application Security: Exploitation and Countermeasures for Modern Web Applications (2nd ed.). O'Reilly Media.