Web Application Programming
ENCT 302

Year/Part: III/I

Teaching Schedule				Examination Scheme						
				Theory			Practical			Total
L	T	P	Total	Assessment Marks	Final		Assessment Marks	Final		
					Duration (Hrs)	Marks		Duration (Hrs)	Marks	
3	1	3	7	40	3	60	50	-	-	150

Depth Codes

E-Explanation	C-Circuit	D-Definition	DM-Demonstration
DV-Derivation	DW-Drawing	P-Proof	I-Illustration
NUM-Numerical	PRG-Programming	S-State	ACT-Activity-based Learning
MP- Mini Project	EXP-Experiment	REV-Review / Recap	PS- Problem Solving
QA- Question Answer	Q-Quiz	ST- Surprise Test	MT-Mid Term Test

UNIT	Topic/Subtopic	Depth Code	Description of Depth	Actual Plan			Week
				L	T	P	
1	**Introduction**			**6**	**2**	2	**2**
1.1	Overview of Web Applications & Evolution of Web Architecture	E, D, I, QA	Explain the concept of traditional vs. modern web apps , Explain evolution of static to dynamic web application, Explain the concept multi-tier & single-page architectures with diagrams.	1			
1.2	Client–Server Architecture, HTTP, HTTPS, URLs, DNS, Web Browsers	E, D, I, DW, QA	Architecture diagrams, HTTP request–response cycle, DNS lookup flow, browser rendering basics.	1			
1.3	HTML Basics: Syntax, Tags, Attributes, Document Structure	E, DM, I, PRG	Demonstration of basic HTML structure, headings, paragraphs, links, images.	1	0.5	0.5	
1.4	Forms & Inputs, Tables, Lists, Multimedia	DM, PRG, NUM, QA	Practical implementation of forms, interactive inputs, audio/video tags.	0.5	0.5	0.5	
1.5	Semantic HTML5 Elements	E, I, DM	Header, footer, section, article, nav, etc.	0.5	0.25	0.25	

UNIT	Topic/Subtopic	Depth Code	Description of Depth	Actual Plan			Week
				L	T	P	
1.6	CSS Basics: Selectors, Properties, Values, Box Model	E, DM, PRG, PS	CSS rules, inline/internal/external styles, margin/padding/border.	1	0.25	0.25	
1.7	CSS Framework: Bootstrap Introduction	DM, PRG, ACT	Grid system demo, containers, responsive layout activity.	1	0.5	0.5	
2	**JavaScript and Client-Side Programming**			**12**	**4**	4	**4**
2.1	JavaScript Essentials	E, D, DM, QA	Intro to scripting, JS execution model, placement in HTML	0.5			
2.1.1	Data Types, Variables, Control Structures, Functions	E, DM, PS	Practical coding of loops, conditionals, functions.	1	0.25	0.25	
2.1.2	DOM Manipulation, Events, Form Validation	DM, ACT, PRG	Selecting elements, updating UI, event listeners, validation.	1	0.25	0.25	
2.1.3	Local Storage & Session Storage	E, DM, PRG	Storing and retrieving key-value data; building a small demo.	0.5			
2.1.4	GUI Interactions (Basic Dynamic UI)	DM, ACT, I	Hover effects, dynamic menus, modals (JS-only).	1	0.5	0.5	
2.1.5	JavaScript Library: jQuery	E, DM, PRG	DOM shortcuts, event handling, AJAX basics.	1	0.5	0.5	
2.2	Modern JavaScript (ES6+)	E, D, DM, PS	Let/const, arrow functions, destructuring, spread/rest.	1			
2.2.1	sync Operations: Callbacks, Promises, Async/Await	E, I, DM, PS	Understanding async flow; building simple API calls.	0.5	0.25	0.25	
2.2.2	Modules and Imports	DM, PRG	Structuring JS files using ES modules.	0.25			
2.3	Client-Side Applications						
2.3.1	Client-Side Applications: Introduction to React JS	E, I, DM	React JSX, components, rendering.	0.25			
2.3.2	Traditional Multi-Page Apps vs SPAs	E, I, QA	SPA concept, reload-free navigation.	0.5			
2.3.3	React Component-Based UI & Props	DM, ACT, PRG	Building functional components with props.	1	1	1	

UNIT	Topic/Subtopic	Depth Code	Description of Depth	Actual Plan			Week
				L	T	P	
2.3.4	State Management & Data Flow	DM, PS, PRG	useState, one-way data flow, parent-child states.	0.5	0.5	0.5	
2.3.5	Client-Side Routing (React Router)	DM, PS	Navigation without reloading, routing basics.	1	0.5	0.5	
2.3.6	Fetch API: Fetching, Displaying Data, Handling Errors & Loading States	DM, NUM, PRG	Working with real/placeholder API, error handling.	1	0.25	0.25	
2.3.7	Comparison of JS Libraries: React vs Angular vs Vue	E, I, REV	Strengths, weaknesses, use cases.	1			
3	**Server-Side Web Programming**			**9**	**4**	**4**	**4**
3.1	Introduction to MVC Architecture.	E, D, I, QA	Model–View–Controller pattern, flow diagrams, responsibilities.	0.5			
3.2	Role of Backend in Web Applications.	E, I, D	Server logic, routing, business logic, security layers	0.5			
3.3	Backend Web Framework: Django (Overview)	E, DM, I	Django apps, MTV pattern, project structure.	0.5			
3.4	Handling Requests & Responses	DM, PRG, ACT	Writing views, HTTP methods, JSON response.	1	0.5	0.5	
3.5	Form Data Handling & Sessions.	DM, PS, PRG	Form POST processing, CSRF, session usage	1	0.5	0.5	
3.6	Routing, Middleware, Templates.	DM, DW, PRG	URL dispatcher, custom middleware, Template language	1	0.5	0.5	
3.7	Overview & Comparison of Backend Frameworks.	E, I, REV	Django vs Flask vs FastAPI vs .NET MVC vs Rails vs Spring Boot vs Node.js	0.5			
3.8	Database Integration: Relational vs NoSQL, CRUD, ORM	E, DM, PRG	Using Django ORM, basic models, performing CRUD.	2	1	1	

UNIT	Topic/Subtopic	Depth Code	Description of Depth	Actual Plan			Week
				L	T	P	
3.9	Authentication & Authorization: Cookies, Sessions, JWT	E, DM, PS	Login flow, cookies vs sessions, intro to JWT.	1	0.5	0.5	
3.1	Middleware for Logging, Error Handling, Security.	DM, PRG, ACT	Build simple custom middleware for logging & error handling	1	1	1	
4	**Web Services & APIs**			**7**	**3**	**3**	**3**
4.1	API Basics: Role in Web Applications	E, D, I, QA	What APIs are, why they are used, browser–server communication.	0.5			
4.2	REST Principles and RESTful API Design	E, D, DW, DM	HTTP verbs, endpoints, statelessness, resource modeling, API structure.	1	0.5	0.5	
4.3	JSON vs XML Data Exchange	E, D, I, NUM	JSON syntax, XML structure, differences, parsing demo.	0.5	0.5	0.5	
4.4	Data Validation & Serialization	DM, PS, PRG	Validating request data, serializing/deserializing objects (Django REST or similar).	2	0.5	0.5	
4.5	Microservices Architecture Basics	E, I, DM	Microservices concepts, API gateway, loose coupling, communication.	1			
4.6	Building & Testing a Simple REST API	DM, PRG, ACT	Implementing endpoints, returning JSON, testing via Postman/browser.	2	1.5	1.5	
5	**Web Application Security**			**6**	**1**	**1**	**1**
5.1	Common Vulnerabilities: XSS, SQL Injection, CSRF	E, D, I, DM	Explanation + demonstration of vulnerabilities with small examples.	1	0.25	0.25	
5.2	Security Best Practices (Input Validation, Sanitization, HTTPS, Secure Cookies, Env Variables)	E, DM, PS	Secure coding practices, showing validation/sanitization techniques.	2	0.25	0.25	
5.3	Authentication Practices & Token Handling	E, D, DM, PS	Password hashing, sessions, JWT flow, secure token handling.	2	0.25	0.25	
5.4	Security in Full-Stack Apps: CORS, Safe Sessions	E, DM, ACT	CORS policies, session hardening, browser security settings.	1	0.25	0.25	

UNIT	Topic/Subtopic	Depth Code	Description of Depth	Actual Plan			Week
				L	T	P	
6	**Web Application Deployment and Modern Trends**			**5**	**1**	**1**	**1**
6.1	Full-Stack Development Overview	E, I, D	Understanding frontend–backend integration, build pipeline, API integration	1			
6.2	Testing and QA	E, DM, PS	Unit tests, integration tests, UI testing basics, automation overview.	1	0.5	0.5	
6.3	DevOps, CI/CD	E, D, DM	Pipelines, GitHub Actions, Docker intro, deployment workflow.	1	0.5	0.5	
6.4	Progressive Web Apps (PWAs)	E, I, DM	Service workers, caching, offline-first, manifest files.	1			
6.5	Responsive Design & Usability	DM, ACT, I	Mobile-first approach, breakpoints, UX considerations.	1			

Web Application Programming
ENCT 302

Answer ALL questions:

1. Explain the concept of semantic web [**4 Marks**]
2. Design an HTML form for student registration with the following: name, email, phone, program selection (dropdown), and a submit button. **[4 Marks]**
3. Explain DOM manipulation and event handling using JavaScript with suitable code examples. Demonstrate add/remove DOM nodes dynamically. A college wants to develop a Single Page Application (SPA) for course selection using React JS. [**10 Marks**]
    A. Draw an architecture diagram for SPA using React (components, props, state, router).
    B. Explain the flow of data from a backend API to UI components using fetch() and state hooks.
    C. Compare React with Angular and Vue.
4. List and explain any three features of Modern JavaScript with code snippets (ES6+). [**6 Marks**]
5. Explain Model-View-Controller (MVC) architecture with a labeled diagram. [**4 Marks**]
6. Write a Django flow for: [**8 Marks**]
    A. URL routing
    B. View handling form data
    C. Template rendering
    D. Session storage

    Design the complete request to response cycle using a block diagram.
7. Explain JSON versus XML, with examples, advantages, and data parsing workflow [**4 Marks**]
8. Consider a hospital management system designing Microservices Architecture. Explain with an architectural diagram highlighting: [**5 Marks**]
    A. API gateway
    B. Service communication
    C. Independent deployments
    D. Database per service strategy
9. Explain SQL Injection, XSS, and CSRF with examples. [**4 marks**]
10. Explain why CORS is needed in full-stack applications. [**4 marks**]
11. Write short notes non: [**3 Marks + 4 Marks**]
    A. Continuous Integration and Continuous Delivery
    B. Progressive web apps