

## Department of MCA

### SYLLABUS of FY MCA 2024-25 (Autonomous)

#### SEMESTER-2

COURSE TITLE		ADVANCE WEB TECHNOLOGIES			CREDITS	4
COURSE CODE		MCA206	COURSE CATEGO RY	ME2-2.2	L-P-T	2-2-0
Version	1.0	Approval Details		07-2024		
ASSESSMENT SCHEME						
First Periodica l Assessment	Second Periodica l Assessment	Seminar/ Assignments /Project		Practical Assessment/ Test	Attendanc e	ES E
10%	10%	10%		10%	10%	50%
Course Descriptio n	Advance Web Technologies Web development typically refers to the coding and programming side of website production. When you learn web development, you might start out writing a simple page of HTML text and build up to creating more complex, feature-rich applications designed to be accessed from various Internet-connected devices.					
Course Objectiv e	1. To impart the design, development and implementation of Dynamic Web Pages. 2. To implement the Latest properties of Node, Angular, Express 3. To implement the Concept of NodeJS. 4. To develop programs for the Web using Angular and SPA. 5. To design and implement dynamic websites with a good sense of design and the latest technical aspects.					
Course Outcom e	CO1: Understand Outline the basic concepts of Advance Internet Technologies CO2: Apply Design appropriate user interfaces and implements webpage based on given problem Statement CO3: Apply Implement concepts and methods of NodeJS CO4: Apply Implement concepts and methods of Angular CO5: Analyze Build Dynamic web pages using server-side programming with Database Connectivity					
Prerequisites: Student must have hands-on working knowledge of HTML, CSS, JavaScript						

MODULE: Topic	Sessions	Weightage (%)	CO Mapping
<b>MODULE 1: React JS</b>			
<b>Topic:</b> 1. 1 Introduction and Setup to React ,1.2 Component-based Architecture 1.3 Handling Events and Forms, 1.4 React Router, 1.5 State Management with Redux, 1.6 React Hooks App Component and JSX Functional, Component, Adding CSS, 1.7 Click Events, 1.8 PROPS, Forms, Use State, Use Effects, Custom Hook, 1.9 Router Links, State Management	7	20%	CO-1

<b>MODULE 2: Node JS</b>			
<b>Topic:</b> 2.1 Node JS, 2.2 introduction and how it works, 2.3 installation of node js , 2.4 REPL, 2.5 NPM, 2.6 How modules work, Webserver Creation, 2.7 Events	<b>6</b>	<b>20%</b>	<b>CO2</b>
<b>MODULE 3: Angular JS</b>			
<b>Topic:</b> 3.1 Angular (Latest Stable Version), 3.2 Introduction (Features and Advantage), 3.3 Type Script, Modules, Components, 3.4 Directives, Expression, Filters, Dependency Injection, 3.5 Services, 3.6 Routing, SPA (Single Page Application), Data binding, property binding, Event Binding, Two-way data binding, String Interpolation.	<b>7</b>	<b>20%</b>	<b>CO-3</b>
<b>MODULE 4: Mongo DB</b>			
<b>Topic:</b> 4.1 SQL and NoSQL Concepts, 4.2 Database Integration – Mongo DB, 4.3 CRUD with Mongoose & Mongo DB I, 4.4 CRUD with Mongoose & Mongo DB II, 4.5 Mongo Data Validation, Modelling Relationships, 4.6 Authentication, Authorization I, Authentication, Authorization II, Authentication, Authorization III, 4.7 Profanity Filtering and Post Moderation, 4.8 Caching & Performance, 4.9 Caching & Performance continue, Protecting express App	<b>6</b>	<b>20%</b>	<b>CO-4</b>
<b>Module 5: ExpressJS</b>			
<b>Topic:</b> 5.1 Introduction to Express ,5.2 Building Restful APIs using express, 5.3 Building Restful APIs using express II, Express Middleware's, Express Middleware's continue, 5.4 Rendering/Template engines, Routing	<b>2</b>	<b>10%</b>	<b>CO-5</b>
<b>Module 6: Hands on Application Development</b>			
<b>Front-end:</b> Creating interactive user interfaces with React/Angular. <b>Back-end:</b> Developing APIs with Node.js and Express.js. <b>Database:</b> Designing and interacting with a MongoDB database.  The emphasis is on practical experience. need to develop code, debug, test, and potentially work in a team.  Essentially, this section aims to equip you with the skills to build real-world web applications using industry-standard technologies.	<b>2</b>	<b>10%</b>	<b>ALL CO</b>

<b>TEXT BOOKS</b>	
<b>1.</b>	<b>"Data Structures and Algorithms with JavaScript" by Michael McMillan</b> <a href="https://www.amazon.com/Data-Structures-Algorithms-Michael-McMillan/dp/1449364934">https://www.amazon.com/Data-Structures-Algorithms-Michael-McMillan/dp/1449364934</a>
<b>2</b>	<b>"JavaScript: The Good Parts" by Douglas Crockford</b> <a href="https://www.amazon.com/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742">https://www.amazon.com/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742</a>
<b>REFERENCE BOOKS</b>	
<b>1.</b>	<b>"Eloquent JavaScript" by Marijn Haverbeke</b> <a href="https://eloquentjavascript.net">https://eloquentjavascript.net</a>
<b>2.</b>	<b>"Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein</b>

	<a href="https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844">https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844</a>
<b>E BOOKS</b>	
1.	"JavaScript Data Structures and Algorithms" by Sammie Bae <a href="https://opendatastructures.org/ods-cpp.pdf">https://opendatastructures.org/ods-cpp.pdf</a>
2	"Open Data Structures (in pseudocode)" by Pat Morin: <a href="https://opendatastructures.org/ods-python.pdf">https://opendatastructures.org/ods-python.pdf</a>
<b>MOOC</b>	
1.	"Algorithms and Data Structures" on edX <a href="https://www.edx.org/course/algorithm-and-data-structures">https://www.edx.org/course/algorithm-and-data-structures</a>
2.	NPTEL Data Structures and Algorithms <a href="https://archive.nptel.ac.in/courses/106/102/106102064/">https://archive.nptel.ac.in/courses/106/102/106102064/</a>

Handson Practical Questions:	Sessions	Weight age (%)	CO Mapping
<b>MODULE 1: React JS</b>			
1. <b>Dynamic E-commerce Product Page:</b> Develop a product display component using <b>PROPS</b> , manage availability status with <b>useState</b> , and implement a pop-up on <b>click events</b> .	7	20%	CO-1
2. <b>Multi-Step Form for Job Application:</b> Create a <b>React form</b> with input validation using <b>useState</b> and <b>useEffect</b> , enable navigation with <b>Redux</b> , and allow users to save progress.			
3. <b>Ride-Sharing App with React Router:</b> Implement <b>React Router</b> for pages like Home, Book Ride, and Ride History, use <b>Router Links</b> , and manage ride data with <b>Redux</b> .			
4. <b>Movie Recommendation System with Custom Hooks:</b> Build a recommendation system using a <b>Custom Hook</b> for API fetching, update suggestions dynamically with <b>useEffect</b> , and track selected movies using <b>useState</b> .			
5. <b>Social Media Dashboard with Real-Time Updates:</b> Create a post feed using <b>useState</b> for dynamic updates, fetch posts			

with <b>useEffect</b> , and manage likes/comments globally with <b>Redux</b> .			
6. <b>Event Booking System with State Management:</b> Design an event listing component with <b>PROPS</b> , handle bookings dynamically with <b>useState</b> , and manage global event availability using <b>Redux</b> .			
<b>MODULE 2: Node JS</b>			
1. <b>Real-Time Chat Application:</b> Develop a real-time chat app using <b>Node.js</b> and <b>WebSockets</b> , manage events with the <b>EventEmitter module</b> , and handle multiple user connections dynamically.			
2. <b>CLI-Based Task Manager:</b> Create a <b>command-line task manager</b> using <b>Node.js REPL</b> , allowing users to add, delete, and list tasks, and integrate <b>NPM packages</b> for file storage.			
3. <b>E-commerce Backend with NPM Modules:</b> Build an <b>e-commerce backend</b> using <b>Node.js</b> , manage dependencies with <b>NPM</b> , and structure the project using <b>custom modules</b> for handling users, products, and orders.			
4. <b>Web Server for Blog Platform:</b> Develop a simple <b>web server</b> using <b>Node.js HTTP module</b> , serve HTML pages dynamically, and implement <b>routing</b> for different blog sections.			
5. <b>Event-Driven File Logger:</b> Implement a <b>file logging system</b> using <b>Node.js events</b> , where log messages are stored in a file when triggered by different system actions.			
6. <b>API Server for IoT Data Processing:</b> Create a <b>Node.js API</b> to receive real-time IoT sensor data, process it using <b>event-driven architecture</b> , and store logs using <b>file system modules</b> .	6	20%	CO2
<b>MODULE 3: Angular JS</b>			

1. <b>E-Learning Platform with Dynamic Content:</b> Develop an <b>Angular SPA</b> for an e-learning platform, using <b>modules and components</b> to display courses, and implement <b>routing</b> for navigating between lessons.			
2. <b>Real-Time Weather Dashboard:</b> Build a <b>weather dashboard</b> using <b>Angular services</b> to fetch live data, apply <b>dependency injection</b> for API calls, and display results dynamically with <b>data binding</b> .			
3. <b>Task Management App with Two-Way Binding:</b> Create a <b>task manager</b> where users can add, edit, and delete tasks, utilizing <b>two-way data binding</b> for instant UI updates and <b>event binding</b> for task actions.	7	20%	CO-3
4. <b>E-commerce Product Filter System:</b> Implement an <b>Angular product catalog</b> with <b>directives and filters</b> to allow users to search and filter products dynamically based on categories and price range.			
5. <b>Finance Dashboard with String Interpolation:</b> Develop a <b>finance dashboard</b> that displays <b>real-time stock market data</b> using <b>string interpolation</b> to update UI dynamically with fetched data.			
6. <b>Online Booking System with Angular Routing:</b> Build an <b>online booking system</b> for hotels, implementing <b>Angular routing</b> for seamless page transitions and <b>SPA architecture</b> for a smooth user experience.			
<b>MODULE 4: Mongo DB</b>			
1. <b>User Management System with Authentication:</b> Develop a <b>user registration and login system</b> using <b>MongoDB and Mongoose</b> , implement <b>authentication and authorization</b> to restrict access to certain pages.	6	20%	CO-4
2. <b>E-Commerce Inventory Management:</b> Create an <b>inventory management system</b> with <b>CRUD operations</b> using			

<p><b>Mongoose</b>, ensuring proper <b>data validation and schema relationships</b> for products, categories, and suppliers.</p> <p>3. <b>Blog Platform with Profanity Filtering:</b> Build a <b>blogging platform</b> where users can post articles, implement <b>profanity filtering and post moderation</b> to automatically review and flag inappropriate content.</p> <p>4. <b>Real-Time Analytics Dashboard with Caching:</b> Design a <b>real-time analytics dashboard</b> that stores user interactions in <b>MongoDB</b>, optimize performance with <b>caching techniques</b>, and prevent duplicate API calls.</p> <p>5. <b>Secure API for Financial Transactions:</b> Implement a <b>secure REST API</b> for processing transactions, integrate <b>authentication and authorization</b> with JWT, and protect sensitive endpoints from unauthorized access.</p> <p>6. <b>Online Learning Platform with Role-Based Access:</b> Develop an <b>e-learning platform</b> where users have different roles (admin, instructor, student), using <b>MongoDB relationships</b> to manage course enrollments and <b>protect routes</b> based on roles.</p>			
<b>Module 5: ExpressJS</b>			
<p>1. <b>Task Management API with Express:</b> Develop a <b>RESTful API</b> using <b>Express.js</b> for a task management system, implementing <b>CRUD operations</b> and handling requests with <b>middleware</b>.</p> <p>2. <b>E-Commerce Product Catalog API:</b> Create an <b>Express.js API</b> to serve product details, implement <b>routing</b> for different product categories, and use <b>middleware</b> for logging and authentication.</p> <p>3. <b>Movie Review Platform with Templating:</b> Build a <b>movie review website</b> using <b>Express.js</b> and a <b>templating engine (EJS/Pug)</b> to render dynamic content and implement <b>route-based navigation</b>.</p>	2	10%	CO-5

<p>4. <b>Authentication System with Middleware:</b> Design an <b>Express-based login system</b> using <b>JWT authentication</b>, protect routes with <b>custom middleware</b>, and handle role-based access control.</p> <p>5. <b>News Aggregator with API Integration:</b> Create a <b>news aggregator platform</b> using <b>Express.js</b>, fetch news from external APIs, and use <b>middleware</b> for request validation and rate limiting.</p> <p>6. <b>Food Delivery App Backend:</b> Develop a <b>food delivery backend API</b> using <b>Express.js</b>, implement <b>nested routing</b> for users, restaurants, and orders, and optimize request handling with <b>middleware functions</b>.</p>			
--	--	--	--