

|                              |                                                                                                                                                                                                      |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Branch Name:</b>          | MCA                                                                                                                                                                                                  |
| <b>Program Code:</b>         | CS201                                                                                                                                                                                                |
| <b>Course Name:</b>          | Full Stack Development-I                                                                                                                                                                             |
| <b>Course Code:</b>          | 4CS2010206T                                                                                                                                                                                          |
| <b>Pre-requisite Course:</b> | Basic programming, HTML & CSS, JavaScript fundamentals, OOP concepts, TypeScript basics, client-server architecture, basic database CRUD, command line/npm usage, and Git version control knowledge. |

### Course Objectives:

1. To enable students to design responsive, component-based web interfaces using modern front-end frameworks like ReactJS and Angular.
2. To provide in-depth knowledge of state management techniques such as React Hooks, Context API, Redux, and Angular Two-Way Data Binding for building interactive and scalable web applications
3. To introduce students to Node.js and Express.js for developing secure, efficient, and scalable server-side applications and RESTful APIs.
4. To familiarize students with both SQL (MySQL) and NoSQL (MongoDB) databases for effective data storage, querying, and management in real-world applications.
5. To develop students' ability to integrate front-end, back-end, and database technologies for building complete, industry-relevant full-stack web applications.
6. To enhance students' understanding of modern development practices, including **API security**, **data validation**, and the use of tools like **Mongoose**, **JWT**, and cloud database services.

### Teaching and Examination Scheme:

| Teaching Scheme (Hours per week) |              |               |        | Evaluation Scheme (Marks) |                       |                       |                       |               |
|----------------------------------|--------------|---------------|--------|---------------------------|-----------------------|-----------------------|-----------------------|---------------|
| Lecture (L)                      | Tutorial (T) | Practical (P) | Credit | Theory (Marks)            |                       | Practical (Marks)     |                       | Total (Marks) |
|                                  |              |               |        | University Assessment     | Continuous Assessment | University Assessment | Continuous Assessment |               |
| 3                                | -            | -             | 3      | 60                        | 40                    | -                     | -                     | 100           |

### Course Contents:

| Unit No | Topic                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Total Hours | Weightage (%) |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------|
| 1       | <b>Front-End Framework – ReactJS</b><br><b>Component-Based Architecture</b> <ul style="list-style-type: none"> <li>● <b>React Overview:</b> <ul style="list-style-type: none"> <li>○ Introduction to ReactJS &amp; its Ecosystem</li> <li>○ Role of React in Single Page Applications (SPAs)</li> <li>○ Comparison with Traditional DOM vs Virtual DOM</li> <li>○ Setting up React environment using Create React App (CRA)</li> </ul> </li> <li>● <b>Components &amp; Architecture:</b> <ul style="list-style-type: none"> <li>○ Component-Based Architecture</li> <li>○ Functional vs Class Components</li> <li>○ Organizing Components: Folder Structure</li> <li>○ Rendering elements and composition of components</li> </ul> </li> <li>● <b>Props &amp; State:</b> <ul style="list-style-type: none"> <li>○ One-way data flow for predictable state management.</li> <li>○ Enhances component reusability and data integrity.</li> </ul> </li> </ul> | 12          | 25            |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |    |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|
|   | <ul style="list-style-type: none"> <li>● <b>Component Lifecycle:</b> <ul style="list-style-type: none"> <li>○ React Phases: Mounting, Updating, Unmounting.</li> <li>○ Hooks like <b>useEffect</b> now manage side-effects, replacing older lifecycle methods.</li> </ul> </li> </ul> <p><b>State Management</b></p> <ul style="list-style-type: none"> <li>● <b>Local State (useState Hook):</b> <ul style="list-style-type: none"> <li>○ Manages dynamic data in functional components.</li> <li>○ Triggers re-rendering on state changes.</li> </ul> </li> <li>● <b>Global State:</b> <ul style="list-style-type: none"> <li>○ <b>Context API:</b> <ul style="list-style-type: none"> <li>■ Built-in, lightweight solution for passing data across multiple components.</li> </ul> </li> <li>○ <b>Redux (Industry Standard):</b> <ul style="list-style-type: none"> <li>■ Centralized state store for large-scale apps.</li> <li>■ Middleware tools like Redux Thunk and Redux Saga for handling async operations.</li> </ul> </li> </ul> </li> <li>● <b>Side Effects (useEffect Hook):</b> <ul style="list-style-type: none"> <li>○ Efficiently handles API calls, subscriptions, timers.</li> <li>○ Dependency array management to optimize performance.</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |    |
| 2 | <p><b>Introduction to Angular</b></p> <ul style="list-style-type: none"> <li>● Used for building Single Page Applications (SPA) with high performance and modular architecture.</li> <li>● Based on component-driven development, promoting reusability and maintainability.</li> <li>● Written in TypeScript, offering better code quality and error handling.</li> </ul> <p><b>TypeScript Basics</b></p> <p><b>Overview of TypeScript:</b></p> <ul style="list-style-type: none"> <li>● Developed by Microsoft; Superset of JavaScript.</li> <li>● Enables <b>static typing</b>, ensuring type safety.</li> <li>● Detects errors at compile-time, improving large-scale project stability.</li> <li>● Essential for modern Angular application development.</li> </ul> <p><b>Key TypeScript Features for Angular:</b></p> <ul style="list-style-type: none"> <li>● Data Types: String, Number, Boolean, Arrays, Tuples, Enums.</li> <li>● Interfaces: Define object structure and contracts, improving consistency.</li> <li>● Classes &amp; Object-Oriented Programming (OOP)</li> <li>● Access Modifiers: public, private, protected.</li> <li>● Generics: Develop reusable and type-safe components and functions.</li> </ul> <p><b>Core Angular Concepts</b></p> <p><b>Modules, Components, and Templates:</b></p> <ul style="list-style-type: none"> <li>● NgModules: Organize application features into functional blocks.</li> </ul> <p>Components:</p> <ul style="list-style-type: none"> <li>○ Building blocks of the Angular UI.</li> <li>○ Consist of: <b>-Template -Class - Styles</b> .</li> </ul> <p><b>Data Binding Mechanisms in Angular</b></p> <p><b>One-Way Data Binding</b></p> <ul style="list-style-type: none"> <li>● Data flows from component to template.</li> <li>● Utilizes interpolation ({{variable}}) and property binding.</li> </ul> <p><b>Two-Way Data Binding</b></p> <ul style="list-style-type: none"> <li>● Real-time synchronization between template and component.</li> <li>● Implemented using [(ngModel)].</li> </ul> | 12 | 25 |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |    |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|
|   | <ul style="list-style-type: none"> <li>Allows instant reflection of input field changes in component variables.</li> </ul> <p><b>Event Binding</b></p> <ul style="list-style-type: none"> <li>Links UI events to component logic.</li> <li>Syntax: (event)="method()".</li> <li>Example: (click)="submitForm()".</li> </ul> <p><b>Property Binding</b></p> <ul style="list-style-type: none"> <li>Sets properties of DOM elements from component data.</li> <li>Syntax: [property]="value"</li> </ul> <p><b>Additional Important Angular Features</b></p> <ul style="list-style-type: none"> <li>Directives - Dependency Injection (DI) - Pipes - Routing and Navigation</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |    |
| 3 | <p><b>Node.js &amp; Express</b></p> <p><b>Introduction to Node.js</b></p> <ul style="list-style-type: none"> <li>Node.js is a powerful, open-source, server-side runtime environment that executes JavaScript code outside the browser.</li> <li>Built on Google Chrome's V8 JavaScript engine, offering high performance.</li> <li>Utilizes an event-driven, non-blocking I/O architecture ideal for building scalable network applications.</li> </ul> <p><b>Setting Up the Server</b></p> <p><b>Node.js Fundamentals</b></p> <ul style="list-style-type: none"> <li>Runs JavaScript on the server-side.</li> <li>Facilitates building fast, scalable, and efficient network applications.</li> <li>Supports asynchronous programming using callbacks, Promises.</li> </ul> <p><b>NPM Ecosystem</b></p> <ul style="list-style-type: none"> <li>NPM (Node Package Manager).</li> <li>Essential packages for back-end development: <ul style="list-style-type: none"> <li>bcrypt: Password hashing for secure authentication.</li> <li>jsonwebtoken (JWT): Handling secure user sessions with token-based authentication.</li> <li>mongoose: Simplifies MongoDB database interaction.</li> </ul> </li> </ul> <p><b>RESTful API Development</b></p> <ul style="list-style-type: none"> <li><b>Express.js Structure</b> <ul style="list-style-type: none"> <li>-Minimalist, modular framework for scalable APIs.</li> <li>-Supports Router, Middleware, and custom logic separation.</li> </ul> </li> <li><b>API Design Principles</b> <ul style="list-style-type: none"> <li>- Follows RESTful standards: GET, POST, PUT, DELETE.</li> <li>-Async handling with async/await.</li> </ul> </li> <li><b>Security Best Practices</b> <ul style="list-style-type: none"> <li>-Validate inputs with express-validator.</li> <li>-Secure endpoints using JWT for authentication.</li> <li>- Prevent abuse with CORS, rate-limiting middleware.</li> </ul> </li> </ul> | 12 | 25 |
| 4 | <p><b>Understanding Databases</b></p> <ul style="list-style-type: none"> <li><b>SQL (MySQL) Key Concepts:</b> <ul style="list-style-type: none"> <li>Structured Query Language (SQL) for data integrity and ACID compliance.</li> <li>Relational database design: Primary keys, foreign keys, normalization.</li> </ul> </li> <li><b>NoSQL (MongoDB) Key Concepts:</b> <ul style="list-style-type: none"> <li>Document-oriented storage using BSON (Binary JSON).</li> <li>Schema-less design for flexible, scalable solutions.</li> <li>Horizontal scaling and sharding for large datasets.</li> </ul> </li> <li><b>Modern Trends:</b> <ul style="list-style-type: none"> <li>Using Mongoose ODM for MongoDB schema design and validation.</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 12 | 25 |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <ul style="list-style-type: none"> <li>Cloud Database Solutions: MongoDB Atlas, Amazon RDS for production deployment.</li> </ul> <p><b>CRUD Operations</b></p> <ul style="list-style-type: none"> <li><b>SQL CRUD:</b> <ul style="list-style-type: none"> <li>Writing efficient SELECT, INSERT, UPDATE, DELETE queries..</li> </ul> </li> <li><b>MongoDB CRUD:</b> <ul style="list-style-type: none"> <li>Using MongoDB drivers and Mongoose for JavaScript-based queries.</li> <li>Aggregation pipelines for complex queries.</li> </ul> </li> <li><b>Data Security:</b> <ul style="list-style-type: none"> <li>Preventing SQL injection with prepared statements.</li> <li>Implementing data validation and error handling at schema and API levels</li> </ul> </li> </ul> |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### Text Books:

1. "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" Author: Vasanth Subramanian Publisher: Apress Edition: 2nd Edition (2019) ISBN: 978-1484243908 Pages: Approx. 620
2. "Modern Full-Stack Development" by Frank Zammetti Publisher: Apress Edition: 1st Edition (2019) Author: Frank Zammetti ISBN: 978-1-4842-5189-0 Pages: Approx. 377
3. "Learning React: Modern Patterns for Developing React Apps" Author(s): Alex Banks, Eve Porcello Publisher: O'Reilly Media Edition: 3rd Edition (2023) ISBN: 978-1098116597 Pages: Approx. 450
4. "Angular Projects: Build Modern Web Apps with Angular" Author(s): Aristeidis Bampakos, Christos Chronis Publisher: Packt Publishing Edition: 2nd Edition (2023) ISBN: 978-1803238802 Pages: Approx. 550
5. "MongoDB: The Definitive Guide" Author(s): Shannon Bradshaw, Kristina Chodorow, Eoin Brazil Publisher: O'Reilly Media Edition: 3rd Edition (2019) ISBN: 978-1491954461 Pages: Approx. 500

### Reference Books:

1. "Eloquent JavaScript: A Modern Introduction to Programming" Author: Marijn Haverbeke Publisher: No Starch Press Edition: 3rd Edition (2018) ISBN: 978-1593279509 Pages: Approx. 472
2. "Beginning [Node.js](https://nodejs.org/en/docs)" Author: Basarat Syed Publisher: Apress Edition: 1st Edition (2014) ISBN: 978-1484201885 Pages: Approx. 308
3. "Angular Development with TypeScript" Author: Yakov Fain, Anton Moiseev Publisher: Manning Publications Edition: 3rd Edition (2022) ISBN: 978-1617295348 Pages: Approx. 600
4. "MongoDB in Action" Author: Kyle Banker et al. Publisher: Manning Publications Edition: 2nd Edition (2016) ISBN: 978-1617291609 Pages: Approx. 480

### List of Open Source Software/learning website:

<https://developer.mozilla.org>  
<https://react.dev>  
<https://angular.io>  
<https://nodejs.org/en/docs>  
<https://www.w3schools.com>  
<https://www.tutorialspoint.com>  
<https://university.mongodb.com>  
<https://github.com/explore>

**Course Learning Outcomes (CLO): On completion of this course, the students will be able to:**

| CLO  | Description                                                                                                                                          | Bloom's Taxonomy Level |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| CLO1 | Understand and explain the fundamentals of ReactJS including component-based architecture, props, state, and lifecycle methods.                      | Understand / Remember  |
| CLO2 | Develop interactive web applications using Angular framework, implementing components, data binding, directives, and routing.                        | Apply / Create         |
| CLO3 | Demonstrate proficiency in TypeScript features such as static typing, interfaces, classes, generics, and OOP principles in Angular development.      | Apply / Analyze        |
| CLO4 | Build scalable server-side applications using Node.js and Express, including RESTful API design, middleware, and security practices.                 | Apply / Create         |
| CLO5 | Design and implement efficient CRUD operations with both SQL (MySQL) and NoSQL (MongoDB) databases, applying data validation and security.           | Apply / Analyze        |
| CLO6 | Integrate front-end and back-end technologies effectively to create a full-stack web application with proper state management and API communication. | Create / Evaluate      |

### Mapping of CLOs with Pos & PSOs

| Course Learning Outcomes | Program Outcomes(POs) |     |     |     |     |     |     |     |     |      |      |      | Program Specific Outcomes(PSOs) |      |
|--------------------------|-----------------------|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|---------------------------------|------|
|                          | PO1                   | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1                            | PSO2 |
| CLO1                     | H                     | M   | M   | L   | H   | L   | M   | L   | M   | L    | L    | L    | H                               | M    |
| CLO2                     | H                     | H   | H   | L   | H   | L   | M   | M   | M   | M    | M    | M    | H                               | H    |
| CLO3                     | H                     | H   | M   | M   | H   | L   | M   | M   | M   | L    | M    | M    | M                               | H    |
| CLO4                     | H                     | H   | H   | M   | H   | M   | M   | H   | M   | M    | M    | M    | H                               | H    |
| CLO5                     | H                     | H   | H   | H   | H   | M   | M   | M   | M   | M    | M    | M    | H                               | M    |
| CLO6                     | H                     | H   | H   | M   | H   | M   | H   | M   | H   | M    | M    | H    | H                               | H    |

**H:High, M:Medium, L:Low**