

Branch Name:	MCA
Program Code:	CS201
Course Name:	Full Stack Development-I
Course Code:	4CS2010206P
Pre-requisite Course:	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	
-	-	4	2	-	-	25	25	50

LAB/ Practical:

Note: The practical list provided is a part for reference only. The course t teacher may Change/formulate it as per his/her methodology and requirement.

Tools and Libraries Used for Practical:

Visual Studio Code (VS Code) - ReactJS / Angular - TypeScript - React Router / Angular Router - React Hooks (useState, useEffect) - CSS3, Bootstrap - Node.js & [Express.js](https://expressjs.com/) - body-parser, cors, dotenv - jsonwebtoken (JWT), bcrypt, express-validator, helmet - MySQL, MySQL Workbench, mysql2 - MongoDB, MongoDB Compass, Mongoose - Git & GitHub

Sr. No	List of Practical's
1	Setup and Configure Front-End Project (ReactJS or Angular) -Aim: To set up a basic front-end development environment using ReactJS or Angular
2	Develop Reusable UI Components (Header, Footer, Cards) -Aim: To create reusable and independent UI components.
3	Implement Data Passing using Props (React) or @Input Decorators (Angular) -Aim: To pass data between parent and child components.
4	Apply State Management (useState in React or Two-Way Data Binding in Angular) Aim: To manage internal component state or form inputs.

5	Handle Events and User Interactions (Button Clicks, Form Submissions) Aim: To implement event handling mechanisms in the front-end.
6	Fetch Data from APIs using useEffect (React) or Lifecycle Hooks (Angular) Aim: To integrate API calls and manage side effects in components.
7	Setup Node.js Server with Express Framework Aim: To create a basic backend server using Node.js and Express.
8	Develop RESTful API with CRUD Operations Aim: To build a backend API with standard Create, Read, Update, and Delete functionalities.
9	Secure API Endpoints using JWT Authentication Aim: To implement JSON Web Token-based authentication for secure APIs.
10	Setup and Design MySQL Database with Tables and Relationships Aim: To create relational database tables with primary and foreign key constraints.
11	Perform SQL CRUD Operations (SELECT, INSERT, UPDATE, DELETE) Aim: To manipulate data in MySQL using SQL queries.
12	Setup MongoDB Database and Collections with Mongoose Aim: To create and configure a NoSQL database using MongoDB and Mongoose ORM.
13	Perform CRUD Operations in MongoDB with Mongoose Integration Aim: To manage documents in MongoDB using Mongoose within Node.js applications.
14	Secure Data with Validations and Error Handling at API and Database Level Aim: To implement validation mechanisms and error handling strategies in the system.
15	Develop a Full Stack Application (React/Angular Front-End + Node.js API + Database) Aim: To integrate front-end, back-end, and database technologies in a complete project.

Text Books:

1. "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" Author: Vasan 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/)" 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	L	L	M	L	L	L	L	L	M	M	H	M
CLO2	H	H	M	M	H	M	L	L	M	M	M	M	H	H
CLO3	H	H	H	M	H	M	L	L	M	M	M	M	H	H
CLO4	H	H	H	H	H	M	M	L	L	M	M	M	H	H
CLO5	H	H	H	H	H	M	M	L	L	M	M	M	H	H
CLO6	H	H	H	H	H	M	M	M	M	H	H	H	H	H

H: High, M: Medium, L: Low