

Branch Name:	MCA
Program Code:	CS201
Course Title:	Full Stack Development-II Practical
Course Code:	4CS2010307P
Pre-requisite Course:	Basic knowledge of HTML, CSS, JavaScript, React fundamentals, component-based architecture, basic Node.js concepts, REST API basics, database CRUD operations, and version control using Git.

Course Objective:

The objectives of the course are to:

1. Enhance students' knowledge of advanced React concepts including hooks, state management, and performance optimization.
2. Enable students to develop dynamic and scalable front-end applications using modern React techniques.
3. Provide understanding of backend integration using Node.js, Express, and RESTful APIs.
4. Familiarize students with database integration using SQL and NoSQL technologies.
5. Develop the ability to build complete full-stack applications with proper authentication, validation, and security practices.
6. Introduce best practices in application development, testing, deployment, and version control.

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

List of Experiments:

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

Sr. No	Practical Experiments
1.	Setup and configure an advanced React project using Vite or Create React App with proper folder structure and modular organization.
2.	Develop reusable and independent UI components using functional component architecture in React.
3.	Implement state management using useState and manage complex state logic using useReducer hook.
4	Handle side effects such as API calls and lifecycle events using useEffect hook.
5	Implement global state management using Context API to share data across components without prop drilling.
6	Implement conditional rendering and dynamic list rendering using map() with proper key handling.
7	Develop client-side routing using React Router including BrowserRouter, Routes, and navigation links.
8	Fetch and display data from external APIs using Fetch API or Axios with proper error and loading handling.

9	Implement form handling using controlled components with input validation techniques.
10	Optimize React application performance using memorization techniques such as React.memo, useMemo, and useCallback.
11	Setup a backend server using Node.js and Express framework with basic configuration.
12	Implement routing and middleware handling in Express applications for request processing.
13	Develop RESTful APIs using HTTP methods such as GET, POST, PUT, and DELETE.
14	Test and validate API endpoints using tools like Postman.
15	Design and connect a MySQL database with Node.js application using appropriate drivers.
16	Perform SQL CRUD operations including SELECT, INSERT, UPDATE, and DELETE queries.
17	Setup and configure MongoDB database and define schemas using Mongoose ORM.
18	Perform CRUD operations in MongoDB using Mongoose integration with Node.js.
19	Integrate React frontend with Node.js backend APIs for full-stack communication.
20	Develop a complete full-stack application with authentication, CRUD operations, and basic deployment concepts.

Text Books:

1. Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, 2nd Edition, Apress.
2. Alex Banks, Eve Porcello, *Learning React: Modern Patterns for Developing React Apps*, 3rd Edition, O'Reilly.
3. Ethan Brown, *Web Development with Node and Express*, 2nd Edition, O'Reilly Media.
4. Frank Zammetti, *Modern Full-Stack Development*, Apress.

Reference Books:

1. Marijn Haverbeke, "Eloquent JavaScript: A Modern Introduction to Programming", 3rd Edition, No Starch Press.
2. David Herron, "Node.js Web Development", 5th Edition, Packt Publishing.
3. Shama Hoque, "Full-Stack React Projects", 1st Edition, Packt Publishing.
4. Kyle Banker et al., "MongoDB in Action", 2nd Edition, Manning Publications.
5. Kyle Simpson, "You Don't Know JS Yet", 2nd Edition, O'Reilly Media.

List of Open Source Software/learning website:

- <https://react.dev>
- <https://nodejs.org>
- <https://expressjs.com>
- <https://developer.mozilla.org>
- <https://www.mongodb.com/docs>
- <https://www.w3schools.com>
- <https://www.tutorialspoint.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 advanced React concepts including hooks, state management, and performance techniques.	1.Remembering, 2. Understanding
CLO2	Develop scalable front-end applications using React with routing, API integration, and component interaction.	3.Applying 6.Creating
CLO3	Design and implement backend services using Node.js and Express with RESTful API principles.	3.Applying 4.Analyzing
CLO4	Perform database operations using SQL and MongoDB with proper data handling and validation.	3.Applying 4.Analyzing
CLO5	Integrate frontend, backend, and database to build full-stack applications.	6.Creating
CLO6	Apply authentication, security, and best practices in full-stack application development.	5.Evaluating, 6. Creating

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	PO 10	PO11	PO12	PSO1	PSO2
CLO1	H	M	M	–	H	–	–	–	–	–	–	L	H	M
CLO2	H	H	H	M	H	–	–	M	L	L	–	L	H	H
CLO3	H	H	H	M	H	–	–	M	–	–	–	L	H	H
CLO4	H	H	H	H	H	–	–	–	–	–	–	L	H	M
CLO5	H	H	H	M	H	–	–	M	M	L	M	L	H	H
CLO6	H	H	H	M	H	M	–	H	M	L	M	L	H	H

H:High, M:Medium, L:Low