



Subject Code: 01CT1625

Subject Name: Advanced Web Technology

B. Tech. Year – III (Semester VI)

Objective: The objective of the course is to understand modern web technologies based on JavaScript.

Credits Earned: 04 Credits

Course Outcomes: On the completion of the course, students will be able to:

1. Implement NoSQL Database CRUD operations
2. Acquire knowledge about Server-side JS framework to make Database Connectivity
3. Acquire knowledge about functionalities of Client-side and Server-side JS frameworks
4. Explore ReactJS features and create component-based web pages
5. Develop Front-end web pages and connect to the Back-end Databases

Pre-requisite of course:

- Student should have a basic understanding of fundamental languages used by the browser and the server.
- Student should be confident in designing and styling web pages using HTML & CSS.
- Fundamental concepts of JavaScript should be clear and should also be familiar with DOM(Document Object Model) in browser.
- Basic understanding of Node/express will speed up the learning process but is not recommended.

Teaching and Examination Scheme:

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial / Practical Marks		Total Marks
				E	I		V	T	
Theory	Tutorial	Practical		ESE	IA	CSE	Viva	Term Work	
03	00	02	04	50	30	20	25	25	150



Contents:

Unit	Topics	Hours
1	Client-side JS Framework – ReactJS Introduction to ReactJS, Introducing JSX, Rendering Elements, Component and Props, State and Life Cycle, Handling Events, Conditional Rendering, List and Keys, Forms, Lifting State Up, Composition vs Inheritance, Thinking in React, Manual and Unit Testing.	12
2	Server-side JS Framework – Node JS Get started with Node.js, Node Package Manager, Modules Asynchronous Program, Callbacks, Events and Event Loop, Streams and Buffers, Connect Node.js to Database, Web Sockets, API using NodeJS	5
3	Express JS MVC Pattern, Introduction to Express, Routing, HTTP Interaction, Handle Form Data, Handle Query Parameters, Cookies and Sessions, User Authentication, Error Handle, Create and Consume RESTful Services, Use Templates, Functional and Unit Testing	12
4	MongoDB Introduction to MongoDB, CRUD Operations in Mongoddb, Introduction to Mongoose, Core concepts of Mongoose, Extending Models - Working with hooks, Validation of model data, Creating custom static methods, Creating custom instance methods, CRUD operations with Mongoose	5
5	Cloud & Deployment Basics of Devops and Cloud Infrastructure, In-depth of DNS and network terminologies, Introduction to AWS, AWS-ECS, EC2, Docker, Kubernetes, Git Basics, Git with CI/CD, Hosting website on various platforms.	8
Total Hours		42

Suggested Text books / Reference books:

1. Krasimir Tsonev, "Node.js by Example"
2. Title: MongoDB Cookbook Author(s): Amol Nayak
3. Title: Fullstack React: The Complete Guide to ReactJS and Friends Author(s): Anthony Accomazzo
4. <https://reactjs.org/docs/getting-started.html>
5. <https://nodejs.dev/en/learn/>

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching- learning process.

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
15%	30%	20%	15%	10%	10%

Suggested List of Experiments:

1. Create an application for student result management using React JS.
2. Develop a web application which involves database operations using NodeJS
3. Improve any React (+NodeJS) application by adding Error Handling.
4. Develop a web application which involves database operations using Express
5. Create modern, scalable and high-speed Web Applications with React and Node.js +Express
6. Create a RESTful service with Node, Express and MongoDB
7. Frontend: Using ReactJS, make Image Carousel, FAQ/Accordion and Shopping item list with filters.
8. Frontend: Using ReactJS, integrate video player like a youtube clone.
9. Frontend: Using ReactJS, make a e-commerce website (flipkart or amazon) clone
10. Backend: Using NodeJS, build a real time chat application or email sender.
11. Backend: Using NodeJS, build a job search platform or e-commerce platform api to search and filter.
12. Database: Using MongoDB, build the CRUD operations with NodeJS
13. Database: using MongoDB, NodeJS and ReactJS, build a project to fetch and post data from frontend to database.
14. Deployment: Using AWS, deploy the project in EC2 instances or use Lambda functionalities to build APIs.

Additional Frameworks:

1. Frontend: **Next JS** (ReactJS based framework), **Angular 12/13** (requires Typescript knowledge)
2. Backend: **Nest JS**
3. Database: **MySQL, PostgreSQL**
4. Cloud: **Google Cloud Services**