**AEC240303  Full Stack development**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subject**  **Total Credit** | **Teaching scheme (per week)** | | | **Examination scheme** | | | | |
| **Internal [60%]** | | **External [40%]** | | **Total** |
| **Theory** | **Tutorial** | **Practical** | **Theory** | **Practical** | **Theory** | **Practical** |
| 2 | 0 | 1 | 1 | 00 | 60 | 00 | 40 | 100 |

**Prerequisites**

* HTML & CSS – Familiarity with HTML5 structure, form elements, and CSS styling
* Basic JavaScript Knowledge – Understanding of ES6 syntax, functions, arrays, promises, and DOM manipulation.
* Web Development Fundamentals – Understanding client-server model, HTTP methods.

**Course Objectives**

* Introduce students to modern frontend development using ReactJS and its ecosystem.
* Provide hands-on experience with React components, JSX, routing, hooks, and API integration.
* Familiarize students with backend development using Node.js and Express.js for building RESTful services.
* Enable students to perform CRUD operations using MongoDB and understand NoSQL database concepts.
* Equip students with full-stack development skills by integrating client-side and server-side technologies.

|  |  |  |
| --- | --- | --- |
| **Units** | **Contents** | **Weightage** |
| **Unit 1** | **Introduction to ReactJS**  Introduction, Advantages, Setup, React Render,Elements-folder structure  Node Package Manager (NPM), Package.json, Virtual DOM  React JSX-Comments, Nested Element, Attributes, Styling in JSX, Components, Class, Functional component, Arrow component, Events, Props  React Routing, Events, Lists, Keys, Map, Filter |  |
| **Unit 2** | **Advanced ReactJS**  Hooks (useState, useEffect, useReducer, useContext)  Forms, API integration with AXIOS  JSON Introduction: JSON vs XML, JSON syntax  JSON Datatypes: Number, Boolean, string, JSON.parse(), JSON.stringify(), Objects, Arrays, Date |  |
| **Unit 3** | **Introduction to NodeJs**  Introduction, Setup, REPL,  Callbacks:Demonstrating callback using simple JS functions like setInterval(), setTimeout(),  NPM Global Module:NodeMon,  Core modules :File system, OS Modules, Path Modules  HTTP Module, Render Response, Read HTML File Server, Routing, JSON Response  URL, Querystring, How to create, export and use our own modules  Module Wrapper Function, JSON Processing, Events |  |
| **Unit 4** | **Introduction to ExpressJs and MongoDB**  Introduction, Features, Environment Setup  Routing, JSON response passing, HTTP Methods- GET, POST, Middleware  How to Link HTML, CSS & JS file in Express JS, Cookies, Session  File Upload, RESTful APIs  Introduction to NoSQL Database, Features of MongoDB  Difference between RDBMS and NoSQL, MongoDB vs MySQL,  Installation, Create & Drop Database  Create, Drop Collection,  CRUD Operation:Create, Read, Update, Delete, Find  Operators: Comparison, Logical, Arithmetic, Field Update |  |

**Learning Outcomes**

* Develop responsive web interfaces using ReactJS, JSX, and modern JavaScript techniques.
* Utilize React Hooks and routing to manage application state and navigation.
* Implement backend logic with Node.js and Express.js, handling routes, middleware, and server responses.
* Perform CRUD operations and manage data using MongoDB as a NoSQL database.
* Integrate RESTful APIs and deploy full-stack web applications effectively.

**Lab Sessions :**

**ReactJS – Basics (Practical 1–10)**

1. Create a basic ReactJS app and display “Hello World”.
2. Setup folder structure for a React project using create-react-app.
3. Create a JSX element and add inline styling and attributes.
4. Use comments inside JSX and render nested elements.
5. Build a functional component and display props in it.
6. Convert a functional component to a class component and compare both.
7. Create a button and handle onClick event to change state.
8. Build a simple list using .map() and render it in a component.
9. Create a small component tree with 3–4 child components and pass props.
10. Implement basic React Router with two routes and navigation links.

**ReactJS – Advanced (Practical 11–18)**

1. Create a counter app using useState.
2. Build a digital clock using useEffect.
3. Make a theme switcher using useContext.
4. Demonstrate a basic reducer using useReducer to manage state.
5. Design a simple login form and handle form submission with useState.
6. Use Axios to fetch data from a public API and render it in a list.
7. Convert a JSON object to string using JSON.stringify() and display it.
8. Parse a JSON string using JSON.parse() and render data in a component.

**Node.js – Basics (Practical 19–23)**

1. Install Node.js and write a “Hello from Node” program using REPL.
2. Create a script using setTimeout() and setInterval() as callbacks.
3. Build a simple CLI app to read system info using the OS module.
4. Read and write to a file using the fs (File System) module.
5. Setup a local Node server with http module and display HTML response.

**Node.js – Intermediate (Practical 24–26)**

1. Demonstrate the use of URL and querystring modules to parse URL parameters.
2. Create and export a custom module; import and use it in another file.
3. Use nodemon to auto-reload your Node.js application on file changes.

**Reference Books:**

* [“Full stack React – The complete guide to ReactJS and friends”,](https://demo.smarttrainerlms.com/uploads/0003/trainings/course/45/modules/fullstack-react-book-r30_1510302324482009603.pdf) Anthony Accomazzo, Ari Lerner, Nate Murray, Clay Allsopp, David Gutman, and Tyler McGinnis,Fullstack.io