**FRONT END DEVELOPMENT FRAMEWORKS RECORD**

(24SDCS01R)

**SYLLABUS:**

**CO-1: Git, Version Control, and Introduction to HTML/CSS**

Introduction to Git and Version Control Systems, Installing Git and Initial Setup, Git Basics (git init, git add, git commit), Understanding Git Workflow, Branching and Merging, Working with Remote Repositories (git clone, git pull, git push), Resolving Merge Conflicts, Git Logs and Viewing History, Git Reset, Revert, and Checkout, Working with .gitignore, Creating and Managing Branches, Basic Git Commands for Daily Use, Introduction to HTML5 structure, Semantic HTML5 tags, Forms and input types, Media elements (audio, video, images), CSS syntax and selectors, Introduction to responsive web design concepts.

**CO-2: Advanced HTML5, CSS3, and Core JavaScript**

Advanced HTML5 elements (section, aside, nav, header, footer), HTML5 form enhancements (placeholder, required, autofocus, pattern), CSS positioning (static, relative, absolute, fixed, sticky), CSS media types and feature queries, CSS transitions and keyframe animations, Introduction to JavaScript, JavaScript syntax and data types, operators, functions and arrow functions, conditional statements and loops, objects, arrays, set and maps, DOM manipulation, event handling, ES6+ features (let, const, destructuring, spread/rest operators), promises and async/await, fetch API, form validation, error handling, localStorage and sessionStorage, modular JavaScript, classes and objects, Built In Objects, working with JSON, debugging and developer tools.

**CO-3: React Fundamentals and UI Development**

Introduction to React, JSX syntax and expressions, functional components, props and state management, event handling in React, conditional rendering, list rendering and keys, useEffect and useState hooks, component lifecycle overview, forms and controlled components, React Router for navigation, lifting state up, component communication, structure of React apps, reusable components and atomic design principles, state management using Redux Toolkit, UI libraries like Material UI and Tailwind CSS, deployment using Vite.

**CO-4: Advanced React, Testing, and Backend with Node.js/Express**

Context API for global state management, custom hooks for reusable logic, lazy loading and code splitting for performance, error boundaries for robust applications, unit testing with Jest and React Testing Library, introduction to Node.js (event loop, non-blocking I/O), setting up a Node.js project with npm, building a simple server with core modules (http), introduction to Express.js (setup, routing), creating RESTful endpoints (GET, POST), middleware basics (logging, parsing JSON, helmet for security), input validation and sanitization.