

Below is a comprehensive syllabus for a React.js course:

Course Title: React.js Development

Course Description: This course provides an in-depth understanding of React.js for building modern and interactive web applications. Students will learn the fundamentals of React.js, including components, props, state management, and lifecycle methods. The course covers advanced topics such as routing, forms handling, state management with Redux, and integration with backend APIs.

Prerequisites: Basic understanding of HTML, CSS, and JavaScript. Familiarity with ES6 syntax is recommended but not required.

Course Objectives:

1. Understand the fundamentals of React.js and its component-based architecture.
2. Learn how to build reusable and maintainable UI components using JSX.
3. Gain proficiency in state management and handling user interactions in React.js applications.
4. Develop skills in routing, forms handling, and integrating React.js with backend APIs.
5. Explore best practices and advanced techniques for building scalable React.js applications.

Course Outline:

Module 1: Introduction to React.js

- Overview of React.js and its benefits
- Setting up development environment (Node.js, npm, create-react-app)
- Introduction to JSX (JavaScript XML)

Module 2: Components and Props

- Understanding components and props in React.js
- Creating functional and class components
- Passing data between components using props

Module 3: State and Lifecycle Methods

- Managing component state in React.js

- Working with lifecycle methods (componentDidMount, componentDidUpdate, etc.)
- Using setState for updating state

Module 4: Handling Events and Forms

- Handling user events in React.js
- Controlled vs. uncontrolled components
- Handling forms and form validation in React.js

Module 5: React Router

- Introduction to React Router for client-side routing
- Configuring routes and navigation
- Passing parameters and route matching

Module 6: Styling in React.js

- Styling components with CSS and inline styles
- Using CSS frameworks with React.js (Bootstrap, Material-UI)
- CSS-in-JS libraries for styling (styled-components, emotion)

Module 7: State Management with Redux

- Introduction to Redux for state management
- Setting up Redux store, actions, and reducers
- Connecting React components to Redux store

Module 8: Asynchronous Operations and Middleware

- Handling asynchronous operations with Redux Thunk
- Integrating middleware for managing side effects
- Making API requests and updating Redux store

Module 9: Advanced React Patterns

- Higher Order Components (HOCs) and Render Props
- Context API for state management
- Error boundaries and Suspense for handling errors and loading states

Module 10: Testing React Components

- Introduction to testing frameworks (Jest, React Testing Library)

- Writing unit tests for React components
- Snapshot testing and testing best practices

Module 11: Integrating with Backend APIs

- Making HTTP requests from React.js applications
- Fetch API vs. third-party libraries (Axios)
- Handling authentication and authorization

Module 12: Deployment Strategies

- Deployment options for React.js applications
- Hosting React.js applications on platforms like Netlify, Vercel, or AWS
- Continuous Integration and Continuous Deployment (CI/CD) pipelines

Module 13: Real-world Projects and Case Studies

- Working on real-world projects and case studies
- Designing and implementing end-to-end React.js applications
- Presenting findings and insights from projects

Module 14: Capstone Project

- Developing a comprehensive React.js project
- Identifying a business problem or scenario
- Designing and implementing a solution using skills learned throughout the course

Assessment:

- Weekly assignments to reinforce learning concepts.
- Midterm project: Developing a basic React.js application with components, props, and state management.
- Final project: Designing and implementing a comprehensive React.js web application addressing a real-world scenario.

Textbook: "Learning React: A Hands-On Guide to Building Web Applications Using React and Redux" by Kirupa Chinnathambi

Additional Resources:

- Online tutorials and documentation (React.js official documentation, Redux documentation, etc.).

- Supplemental readings and materials provided by the instructor.

Grading:

- Assignments: 30%
- Midterm Project: 20%
- Final Project: 40%
- Participation and Attendance: 10%

Attendance Policy: Regular attendance is expected. Students are allowed a maximum of three unexcused absences. Excessive absences may result in a reduction of the final grade.

Office Hours: Instructor office hours will be held twice a week for additional help and clarification.

Csdtd Centre