

Welcome Note:

Dear Students,

We are excited to announce an upcoming workshop on Full-Stack Web Development organized by the BigO CP Club at RNSIT College. This workshop aims to equip you with the essential skills and knowledge required to become proficient in building modern web applications.

Whether you are a beginner or have some experience in web development, this workshop will provide you with a comprehensive understanding of the front-end and back-end technologies used in full-stack development. You will learn industry-standard tools and frameworks such as HTML, CSS, JavaScript, Node.js, Express.js, and React.js, along with databases like MySQL or MongoDB.

We encourage only those students who have a genuine interest in web development to participate in this workshop. It is an excellent opportunity to enhance your skills, expand your knowledge, and boost your career prospects in the rapidly growing field of web development.

Don't miss out on this exciting learning experience! Register now to secure your spot and embark on a journey to become a proficient full-stack web developer.

Best regards,

Full-Stack Web Development Workshop Syllabus



1. Introduction to Full-Stack Web Development

- Overview of web development concepts and technologies
- Introduction to the client-server architecture
- Overview of the front-end (HTML, CSS, JavaScript) and back-end (Node.js, Express.js) components

2. Front-End Development

A. HTML and CSS Fundamentals

- Introduction to HTML5 and its structure
- CSS basics: selectors, properties, and styling
- Building layouts and responsive designs

B. JavaScript Fundamentals

- Introduction to JavaScript: variables, data types, and operators
- Functions, control flow, and loops
- Objects and Object-oriented Programming (OOP) concepts
- Rest and spread operators for handling arrays and objects
- ES6 and ES7 syntaxes and features, such as arrow functions, template literals, destructuring assignments, and default parameters
- Array methods like map, filter, reduce, and forEach for data manipulation and transformation
- Introduction to module bundlers (e.g., Webpack or Rollup) for managing dependencies and bundling code
- Managing JavaScript code across different files
 - Organizing code into modules
 - Importing and exporting code using CommonJS modules and ES6 modules

C. Introduction to React.js

- Overview of React.js and its benefits
- Components, props, and state
- Using useState hook for managing component state
- Implementing side effects and lifecycle management with useEffect hook
- Context API and useContext hook for managing global state
- React Router for handling client-side routing
- Building reusable and modular components
- Best practices for component composition and state management

d. CSS Options (Choose one)

- Option 1: Using a UI library like Material-UI for styling and UI components
- Option 2: Working with vanilla CSS for custom styling and design
- Advanced CSS techniques and features (e.g., flexbox, grid, animations) can be explored based on the chosen option

3. Front-End Frameworks and Libraries

- Introduction to UI frameworks (e.g., Material-UI)
- Integrating Material-UI components into React.js applications
- Customizing the theme and styles

4. Back-End Development

A. Introduction to Node.js

- Overview of Node.js and its features
- Setting up a Node.js project
- NPM and package management

B. Building RESTful APIs with Express.js

- Introduction to Express.js framework
- Routing and handling HTTP requests
- Middleware and error handling

C. Working with Databases

- Introduction to databases: SQL vs. NoSQL
- Connecting to databases (MySQL or MongoDB)
- Querying and manipulating data

5. Authentication and Security

- Overview of authentication and authorization concepts
- Implementing user registration and login functionality
- Introduction to password hashing and encryption
- Securing APIs and protecting routes

6. Deployment and Production-Ready Applications

- Deployment options for web applications (e.g., Heroku, Netlify, AWS, Firebase)
- Building a React app for production
- Optimizing performance and minimizing bundle sizes
- Configuring environment variables for different environments
- Implementing continuous integration and continuous deployment (CI/CD) with GitHub Actions
- Automating deployment workflows
- Monitoring and error tracking in production environments

7. Project Development

- Students will choose one project from the suggested project ideas
- Step-by-step guidance and support in implementing the chosen project
- Regular checkpoints and reviews to ensure progress

8. Project Presentations and Conclusion

- Students present their completed projects
- Q&A session and feedback
- Recap of key learnings and next steps for further learning

Sample Project for Hands on learning:

1. Database Choice:

- Students can choose between SQL or NoSQL databases.
- SQL options: MySQL, PostgreSQL, SQLite, etc.
- NoSQL options: MongoDB, Firebase, Cassandra, etc.

2. CSS Options:

- Option 1: Using a UI library like Material-UI for styling and UI components.
- Option 2: Working with vanilla CSS for custom styling and design.

Todo List Application with User Authentication:

- Students will assist to develop a todo list application where users can create and manage their tasks.
- Features will include user registration and login, task creation and deletion, task prioritization, and task reminders.

Potential Project Ideas:

1. E-commerce Website:

- Students can build an e-commerce website where users can browse products, add them to a cart, and proceed to checkout.
- Features may include product categories, search functionality, user authentication, and payment integration.

2. Social Media Platform:

- Students can create a social media platform where users can create profiles, post updates, follow other users, and engage in discussions.
- Features may include user feeds, comments, likes, notifications, and real-time updates.

3. Task Management Application:

- Students can develop a task management application where users can create, assign, and track tasks.
- Features may include user roles, task categorization, due dates, and task progress tracking.

4. Recipe Sharing Website:

- Students can build a recipe sharing website where users can share and discover new recipes.
- Features may include recipe categories, user ratings, comments, and search functionality.

5. Blogging Platform:

- Students can create a blogging platform where users can publish articles, manage their blogs, and interact with readers.
- Features may include blog post categories, tags, comments, and user subscriptions.