

Job Assistance with Interview Opportunity



"Career Placement in Top Companies:
Your Path to Professional Success"



About Zoko World

Welcome to Zoko World, your premier destination for website development services and professional training courses, located in Preet Vihar, Delhi, India. At Zoko World, we are committed to providing you with top-notch education and services to excel in the digital landscape.

Courses Offered:

- 1. MERN Full Stack Development**
(MongoDB, Express, React, and Node.js): Immerse yourself in modern web development with our MERN stack course.
- 2. PHP with MySQL and More:** Harness the power of PHP combined with MySQL and other essential tools. Our PHP course equips you with the skills to create dynamic websites and robust web applications.

Services:

Web development services: In addition to our training courses, Zoko World offers professional web design and development services. Whether you need a stunning website for your business or a custom web application, our experts in Preet Vihar, Delhi, India, are here to bring your vision to life.



₹12 LPA Average Salary



Discover High-Paying Careers with an Average Salary of ₹12 LPA Join Zoko World to embark on a rewarding career path in web development.

Work On Real Projects



Develop practical skills with real projects that mirror industry challenges at Zoko World.

Real Time Doubt Clarification



Get immediate solutions and personalized guidance with our real-time doubt clarification system.

Real Live Class



Engage with industry experts in interactive live classes to deepen your understanding of web development concepts.

MERN (Full Stack Development Course)

Overview: The MERN stack is a powerful combination of technologies used for building modern web applications. MERN stands for MongoDB, Express, React, and Node.js, each playing a crucial role in the development process. This comprehensive course dives deep into each component of the stack, equipping you with the skills needed to create dynamic, high-performance web applications.

1. HTML

1. Introduction to HTML:

- What is HTML and its role in web development.
- Understanding HTML syntax and structure.
- Setting up a basic HTML document.
- Text formatting with HTML tags (headings, paragraphs, lists, etc.).

2. HTML Elements and Attributes:

- Working with different HTML elements (div, span, anchor tags, etc.).
- Using attributes to enhance element functionality (id, class, href, src, etc.).
- Creating hyperlinks to navigate between web pages.
- Embedding images, videos, and other media content.

3. Forms and Input Elements:

- Building interactive forms using HTML.
- Input types (text, password, email, checkbox, radio buttons, etc.).
- Form validation and error handling.
- Submitting form data to a server using HTML forms.

4. Semantic HTML:

- Understanding semantic markup and its importance.
- Semantic tags (header, footer, section, article, aside, etc.).
- Improving accessibility and SEO with semantic HTML.

5. HTML5 Features:

- Introduction to HTML5 and its new features.
- Multimedia elements (audio, video, canvas).
- Geolocation and local storage.
- Responsive design with HTML5 elements.

6. CSS Integration:

- Linking external CSS stylesheets to HTML documents.
- Inline styles and internal stylesheets.
- Styling HTML elements using CSS selectors (class, id, tag selectors).
- Creating responsive layouts with CSS media queries.

7. Best Practices and Optimization:

- Writing clean and maintainable HTML code.
- Optimizing HTML for performance and SEO.
- Accessibility considerations (ARIA roles, alt attributes, tabindex, etc.).
- Cross-browser compatibility and testing.

2. CSS (Cascading Style Sheets)

1. Introduction to CSS:

- What is CSS and its role in web development.
- CSS syntax and basic structure.
- Inline, internal, and external CSS.
- Selectors and specificity.

2. CSS Box Model:

- Understanding the box model (content, padding, border, margin).
- Box sizing and box model properties.
- Creating layouts with CSS.

3. CSS Layout Techniques:

- Floats and clear property.
- CSS Flexbox layout.
- CSS Grid layout.

4. Typography and Fonts:

- Styling text with CSS (font-family, font-size, font-weight, etc.).
- Google Fonts and custom fonts.
- Text alignment and spacing.

5. CSS Colors and Backgrounds:

- Applying colors using CSS (hexadecimal, RGB, RGBA, HSL, HSLA).
- Background images and gradients.
- CSS background properties (background-color, background-image, background-size, etc.).

6. CSS Transitions and Animations:

- Transition effects (hover effects, fade-in/out, etc.).
- CSS animations (keyframes, animation properties, timing functions).
- Creating interactive and engaging UI elements with animations.

7. Responsive Design with CSS:

- Media queries and responsive design principles.
- Designing for different screen sizes (mobile-first approach).
- Flexibility and adaptability in CSS layouts.

8. CSS Frameworks:

- Introduction to CSS frameworks (Bootstrap, Foundation, etc.).
- Using pre-built CSS components and utilities.
- Customizing and extending CSS frameworks.

9. CSS Preprocessors:

- Overview of CSS preprocessors (Sass, Less).
- Variables, mixins, and nesting in CSS preprocessors.
- Compiling CSS preprocessors into standard CSS.

10. CSS Best Practices:

- Writing clean and maintainable CSS code.
- Organizing CSS stylesheets (file structure, naming conventions).
- Optimization for performance and efficiency.
- Cross-browser compatibility and testing.

3. JavaScript

1. Introduction to JavaScript:
 - What is JavaScript and its role in web development.
 - JavaScript syntax and basic structure.
 - Variables, data types, and operators.
 - Functions, loops, and conditional statements.
2. DOM Manipulation:
 - Understanding the Document Object Model (DOM).
 - Accessing and modifying DOM elements.
 - Event handling and responding to user interactions.
 - Dynamic content creation and updates.
3. JavaScript Functions:
 - Function declarations vs. function expressions.
 - Parameters and arguments in functions.
 - Scope and closures in JavaScript.
 - Arrow functions and their benefits.
4. Arrays and Objects:
 - Working with arrays (creating, accessing, modifying, iterating).
 - Object-oriented programming concepts in JavaScript.
 - Creating and manipulating objects (properties, methods, prototypes).
 - JSON (JavaScript Object Notation) and data serialization.
5. Asynchronous JavaScript:
 - Introduction to asynchronous programming.
 - Callback functions and handling asynchronous tasks.
 - Promises and `async/await` for asynchronous operations.
 - Fetch API for making HTTP requests.
6. ES6 and Modern JavaScript:
 - Features of ECMAScript 6 (ES6) and beyond.
 - `let` and `const` variables.
 - Template literals and string interpolation.
 - Destructuring, spread operator, and rest parameters.
7. JavaScript Libraries and Frameworks:
 - Introduction to popular JavaScript libraries and frameworks (e.g., jQuery, React, Vue.js, Angular).
 - Benefits and use cases of using libraries and frameworks.
 - Integrating external libraries into JavaScript projects.
8. Debugging and Error Handling:
 - Debugging techniques in JavaScript (`console.log`, breakpoints, etc.).
 - Common JavaScript errors and how to handle them.
 - Error handling strategies (try-catch blocks, error objects, etc.).
9. Browser Storage and Cookies:
 - Working with browser storage (`localStorage`, `sessionStorage`).
 - Managing cookies with JavaScript.
 - Storing and retrieving data on the client-side.
10. JavaScript Security:
 - Common security threats in JavaScript applications.
 - Cross-site scripting (XSS) prevention techniques.
 - Best practices for secure coding in JavaScript.

4. MongoDB

1. Introduction to MongoDB:

- What is MongoDB and its role in modern database management.
- Understanding NoSQL databases and their advantages.
- MongoDB vs. traditional SQL databases.
- Installing MongoDB and setting up a local development environment.

1. MongoDB Basics:

- MongoDB data model (documents, collections, databases).
- CRUD operations (Create, Read, Update, Delete) in MongoDB.
- Querying documents using MongoDB's query language.
- Indexing and performance optimization techniques.

1. Schema Design and Data Modeling:

- Designing efficient MongoDB schemas.
- Embedding vs. referencing documents.
- Relationships between collections (one-to-one, one-to-many, many-to-many).
- Normalization and denormalization strategies.

1. Advanced MongoDB Operations:

- Aggregation framework for complex queries and data manipulation.
- Working with MongoDB Atlas (cloud-based MongoDB service).
- Replica sets and high availability configurations.
- Sharding and horizontal scaling for large-scale applications.

1. MongoDB Security:

- MongoDB authentication and authorization mechanisms.
- Role-based access control (RBAC) in MongoDB.
- Data encryption and securing MongoDB deployments.
- Best practices for MongoDB security.

1. Data Import/Export and Backup/Restore:

- Importing and exporting data to/from MongoDB.
- Backup and restore strategies for MongoDB databases.
- Automation tools for managing MongoDB backups.
- Data migration between MongoDB instances.

1. Working with MongoDB Drivers:

- Overview of MongoDB drivers for different programming languages (Node.js, Python, Java, etc.).
- Connecting to MongoDB using a programming language.
- Performing CRUD operations programmatically with MongoDB drivers.

1. MongoDB and Web Development:

- Integrating MongoDB with web applications (using Node.js as an example).
- Building RESTful APIs with MongoDB as the backend database.
- Handling user authentication and session management with MongoDB.
- Real-world examples and case studies of MongoDB in web development

5. React

1. Introduction to React:

- What is React and its role in front-end web development.
- Understanding the React component-based architecture.
- Virtual DOM and its advantages in React applications.
- Setting up a React development environment.

2. React Basics:

- JSX (JavaScript XML) syntax and its integration with React.
- Components in React (functional components vs. class components).
- Props and state in React components.
- Handling events and managing component lifecycle.

3. React Hooks:

- Introduction to React Hooks (useState, useEffect, useContext, etc.).
- Managing state and side effects with React Hooks.
- Custom hooks and their usage in React applications.
- Benefits of using Hooks over class-based components.

4. React Router:

- Navigation and routing in React applications.
- Setting up routes and handling navigation using React Router.
- Route parameters and dynamic routing.
- Nested routes and route guards for authentication.

5. Managing State with Redux:

- Introduction to Redux for state management in React.
- Redux store, actions, reducers, and middleware.
- Connecting React components to the Redux store.
- Asynchronous actions and using Redux Thunk for async operations.

6. Forms and Data Handling:

- Handling forms and form submissions in React.
- Controlled vs. uncontrolled components.
- Validating form input and displaying error messages.
- Fetching and displaying data from APIs in React applications.

7. Component Styling:

- CSS-in-JS libraries (styled-components, Emotion) for styling React components.
- CSS modules and scoped styles in React.
- Theming and global styles in React applications.
- Best practices for styling React applications.

8. React Performance Optimization:

- Identifying and optimizing performance bottlenecks in React apps.
- Memoization and useCallback/useMemo hooks for performance optimization.
- Code-splitting and lazy loading components for better performance.
- React DevTools for performance profiling and debugging.

9. Testing React Applications:

- Unit testing React components with Jest and React Testing Library.
- Snapshot testing and component rendering tests.
- Testing Redux store and async actions.
- Integration testing and end-to-end testing strategies.

10. React Best Practices and Patterns:

- Writing clean and maintainable React code.
- Component composition and reusable UI patterns.
- Error handling and logging in React applications.
- Code organization and project structure best practices.

6. Node js

1. Introduction to Node.js:

- What is Node.js and its role in server-side JavaScript development.
- Understanding event-driven, non-blocking I/O in Node.js.
- Setting up a Node.js development environment.
- Node.js architecture and core modules.

2. Node.js Basics:

- Working with modules in Node.js (CommonJS modules, npm packages).
- File system operations in Node.js (reading/writing files, working with directories).
- Handling asynchronous operations with callbacks and Promises.
- Error handling and debugging techniques in Node.js.

3. Express.js Framework:

- Introduction to Express.js for building web applications with Node.js.
- Setting up an Express.js server and routing requests.
- Middleware concepts in Express.js (body parser, logger, error handling middleware).
- Handling static files, form data, and HTTP requests/responses in Express.

4. Working with Databases:

- Connecting Node.js applications to databases (MongoDB, MySQL, PostgreSQL, etc.).
- CRUD operations with databases using Node.js and Express.
- Using ORM/ODM libraries (Mongoose, Sequelize) for database interactions.
- Database migrations and seeding data.

5. Authentication and Authorization:

- Implementing user authentication with Passport.js.
- Sessions and cookies for managing user sessions.
- JSON Web Tokens (JWT) for stateless authentication.
- Role-based access control (RBAC) in Node.js applications.

6. RESTful APIs with Node.js:

- Designing and building RESTful APIs using Express.js.
- RESTful routing and HTTP methods (GET, POST, PUT, DELETE).
- Request validation and error handling in REST APIs.
- Documenting APIs using tools like Swagger/OpenAPI.

7. Real-time Web Applications with Socket.io:

- Introduction to real-time communication with Socket.io.
- Building chat applications and real-time dashboards with Socket.io.
- Broadcasting events and handling WebSocket connections.
- Scalability considerations for real-time applications.

8. Testing Node.js Applications:

- Unit testing Node.js applications with Mocha and Chai.
- Integration testing and mocking dependencies.
- Test coverage analysis and reporting.
- Automated testing workflows with continuous integration (CI) tools.

9. Security in Node.js Applications:

- Common security vulnerabilities in Node.js applications (SQL injection, XSS, CSRF, etc.).
- Implementing security best practices (input validation, data sanitization, etc.).
- Securing APIs with rate limiting, authentication, and authorization.
- HTTPS configuration and TLS/SSL certificates.

10. Deployment and DevOps:

- Deploying Node.js applications to cloud platforms (AWS, Azure, Heroku, etc.).
- Containerization with Docker for Node.js applications.
- CI/CD pipelines for automated deployment.
- Monitoring and logging Node.js applications in production.

7. Express.js

1. Introduction to Express.js:

- What is Express.js and its role in web development.
- Understanding the middleware concept in Express.js.
- Setting up a basic Express.js application.
- Express.js vs. other web frameworks.

2. Routing and Middleware:

- Handling routes and HTTP methods in Express.js.
- Route parameters and query parameters.
- Middleware functions and their role in request processing.
- Creating custom middleware for logging, authentication, etc.

3. Templating Engines with Express:

- Using template engines (EJS, Pug, Handlebars) with Express.js.
- Rendering dynamic views and passing data to templates.
- Layouts, partials, and template inheritance.
- Integrating front-end frameworks (React, Vue.js) with Express.

4. Working with Forms and Data:

- Handling form submissions in Express.js applications.
- Parsing form data (URL-encoded, multipart) using middleware.
- File uploads with Express.js and Multer middleware.
- Working with JSON data and API endpoints.

5. Database Integration:

- Connecting Express.js applications to databases (MongoDB, MySQL, PostgreSQL, etc.).
- Performing CRUD operations with databases using Express and Mongoose/Sequelize.
- ORM/ODM concepts for database interaction.
- Transactions and data consistency in database operations.

6. Authentication and Authorization:

- Implementing user authentication with Passport.js middleware.
- Session management and cookies in Express.js applications.
- JSON Web Tokens (JWT) for stateless authentication.
- Role-based access control (RBAC) using middleware.

7. Error Handling and Logging:

- Handling errors and exceptions in Express.js.
- Error handling middleware and global error handling.
- Logging requests and responses for debugging and analytics.
- Handling 404 (Not Found) and other HTTP status codes.

8. RESTful APIs with Express:

- Designing and building RESTful APIs using Express.js.
- RESTful routing and resource endpoints.
- Request validation and error handling in API endpoints.
- Documenting APIs using tools like Swagger/OpenAPI.

9. Security Best Practices:

- Implementing security measures in Express.js applications.
- Input validation and data sanitization techniques.
- Protecting against common web vulnerabilities (SQL injection, XSS, CSRF, etc.).
- HTTPS configuration and TLS/SSL certificates.

10. Testing and Debugging:

- Unit testing Express.js applications with Mocha, Chai, and Supertest.
- Integration testing for API endpoints and database interactions.
- Debugging Express.js applications using tools like Node Inspector.
- Test coverage analysis and continuous integration (CI) workflows.

8. Preparing for a job

1. Research the Company:

- Understand the company's products/services, culture, mission, and values.
- Research recent news, projects, or achievements related to the company.

2. Understand the Job Role:

- Review the job description thoroughly to understand the role's responsibilities and requirements.
- Identify key skills, experiences, and qualifications the employer is looking for.

3. Prepare Your Resume:

- Update your resume with relevant skills, experiences, and achievements.
- Customize your resume to align with the job requirements.
- Be ready to discuss any gaps or transitions in your career.

4. Practice Common Interview Questions:

- Prepare answers for common interview questions such as "Tell me about yourself," "Why do you want this job?" and "What are your strengths and weaknesses?"
- Practice behavioral interview questions using the STAR method (Situation, Task, Action, Result).

5. Technical Skills Assessment:

- If the role requires technical skills, review and practice technical questions or tasks related to the job (coding challenges, case studies, etc.).
- Showcase your skills and experiences through relevant projects or portfolio items.

6. Mock Interviews:

- Conduct mock interviews with friends, family, or career advisors to practice answering questions and receive feedback.
- Focus on your communication skills, body language, and confidence during mock interviews.

7. Prepare Questions to Ask:

- Prepare thoughtful questions to ask the interviewer about the company, team, role, or future projects.
- Show your interest in the company and demonstrate your understanding of the role.

8. Dress Appropriately:

- Choose professional attire that is suitable for the company culture and industry.
- Pay attention to grooming and appearance to make a positive impression.

9. Plan Your Journey:

- Plan your route to the interview location in advance to avoid being late.
- Bring necessary documents such as your resume, portfolio, identification, and any reference letters.

10. Follow-Up:

- Send a thank-you email or note to the interviewer(s) after the interview.
- Express gratitude for the opportunity and reiterate your interest in the position.

Master Certification in Full-Stack Development

Upon successful completion of the MEAN stack full stack web development course and meeting the evaluation criteria, students will be awarded a certification accredited by Zoko World.

Certification Benefits:

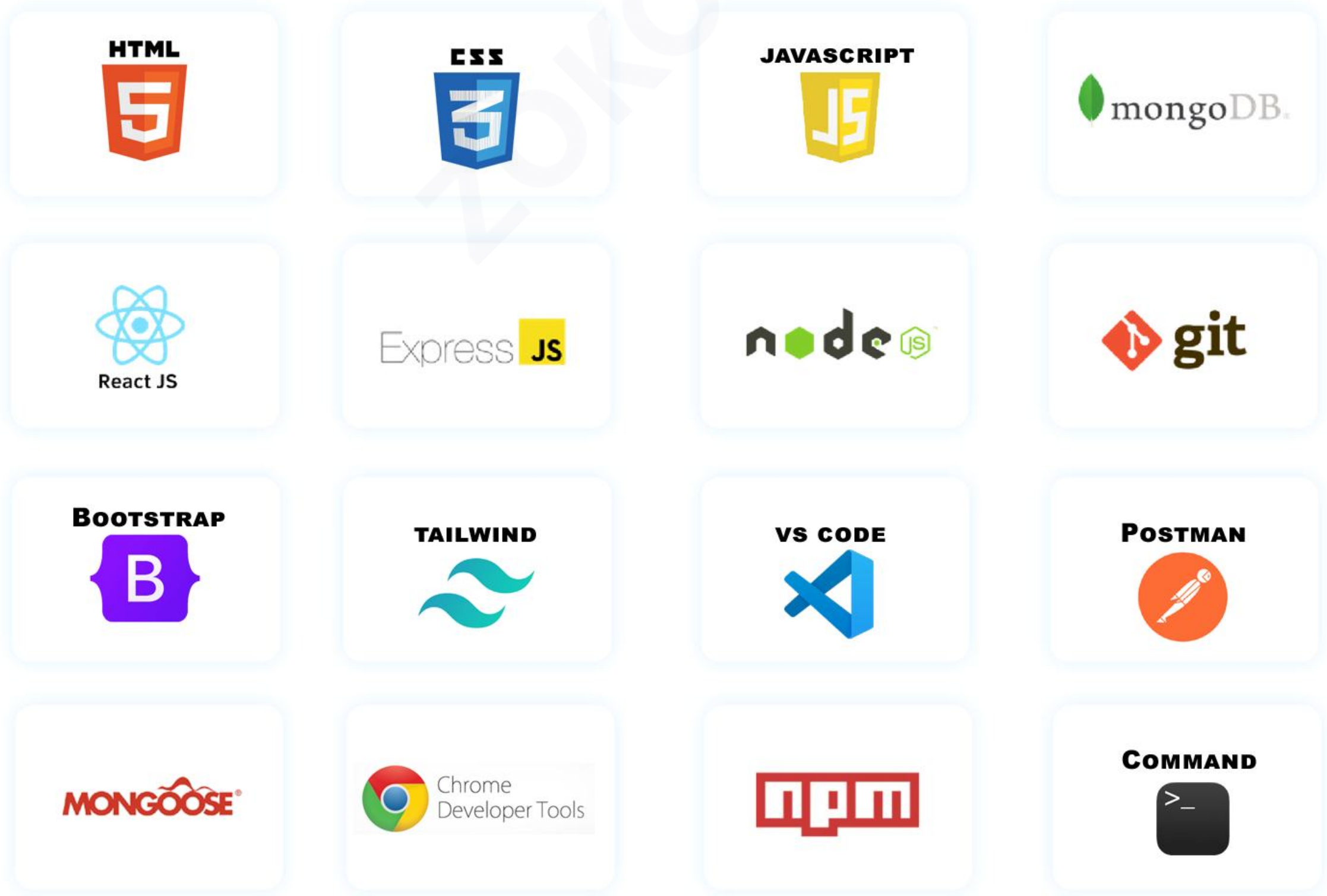
- **Industry Recognition:** The certification is widely recognized within the tech industry, showcasing students' proficiency in MEAN stack development.
- **Career Advancement:** Holding a certified credential enhances students' employability and opens doors to job opportunities in web development roles.
- **Validation of Skills:** The certification serves as tangible proof of the skills and knowledge gained throughout the course, giving students a competitive edge in the job market.
- **Professional Growth:** It signifies a commitment to continuous learning and professional growth, positioning students as valuable assets to potential employers.



Here's the list of

Tools and technologies

Commonly covered in a Full Stack Web Development course



Meet Our Expert Instructors



Sanjay Kumar
Web Developer & Trainer
(4 Year Experience)



Prem Singhania
Web Developer & Trainer
(8 Year Experience)



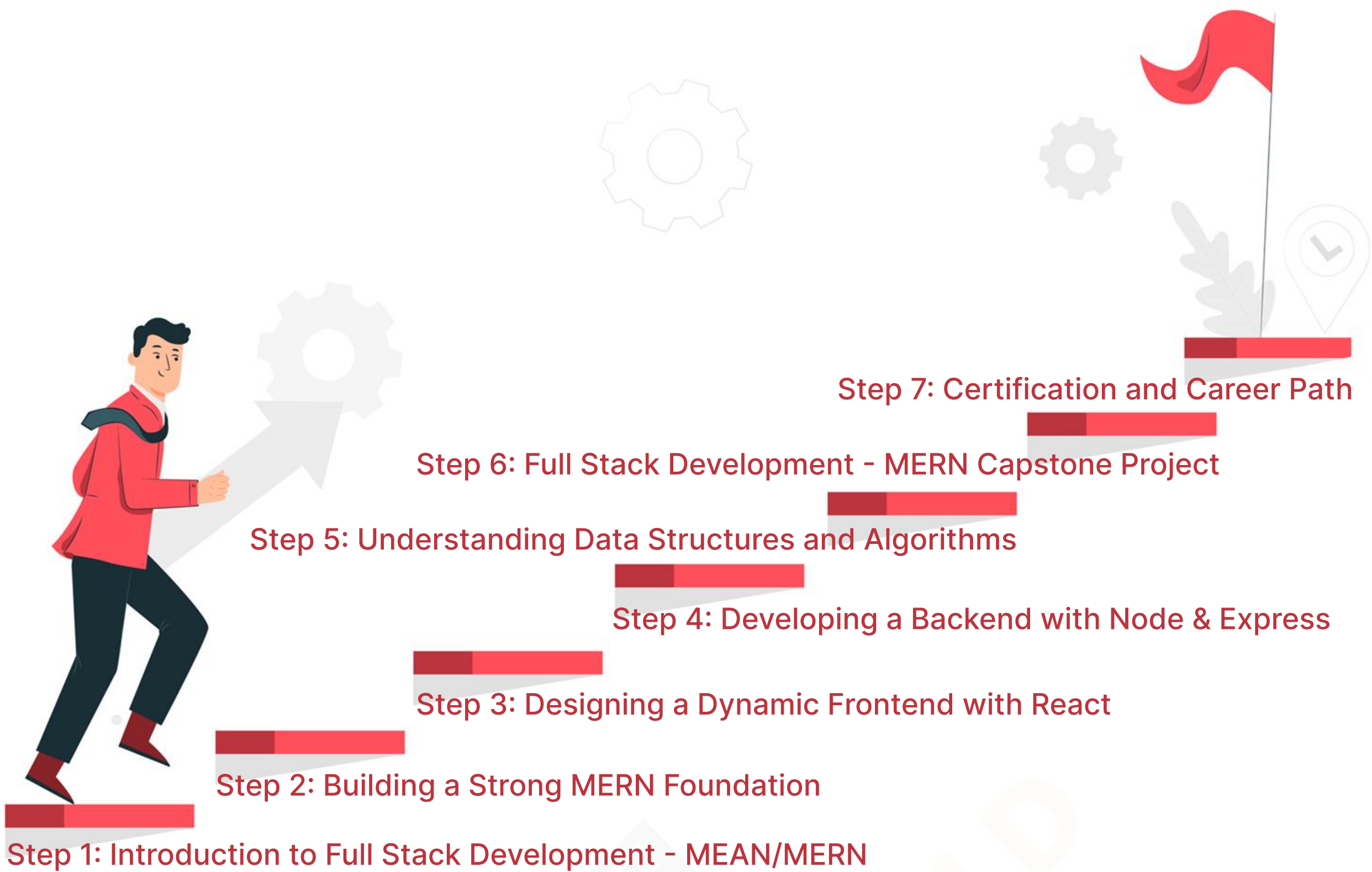
Amit Kumar
Web Developer & Trainer
(5 Year Experience)



Vishal Verma
Web Developer & Trainer
(12 Year Experience)

“A Step-by-Step Roadmap with MERN Stack”

Master Web Development in 6 Months



Demand for Full-Stack Developers

The requirement for full-stack developers in India is substantial. Per the National Association of Software and Services Companies (NASSCOM) report, the demand for full-stack developers in India is anticipated to surge by 30% by this year's end.

 Jobs 100K+ Full stack job posting available in India <i>Source: Indeed</i>	 Growth 30% Year-over-year increase in demand <i>Source: LinkedIn</i>	 Salary ₹6L-₹20L Average annual salary <i>Source: Glassdoor</i>
--	---	--

How to get start ?

Course Information:

- Course Name: **MERN Stack Full Stack Web Development Course**
- Duration: 6 months
- Total Fee: Rs-25000/-
- Class Mode: Online
- Class Schedule: Monday to Saturday, 6 AM to 11 PM
- Class will be 3 days in weeks, (1-2 hours)

Call - [+917827219784](tel:+917827219784)



BECOME A FULL STACK DEVELOPER

Join Free Live Demo Class

Join our free demo live class to explore full-stack web development! Dive into coding, programming, and creating dynamic web apps.

→ Interactive Sessions:

Engage in interactive sessions led by seasoned instructors with extensive experience in the field of web development.

→ Hands-On Learning:

Dive into hands-on learning activities where you'll get to code, build, and troubleshoot real-world projects.

→ Explore Key Concepts:

Gain insights into key concepts such as front-end development, back-end scripting, database management, and more.

→ Ask Questions:

Interact with our instructors, ask questions, and receive valuable guidance tailored to your learning journey.

→ Career Insights:

Get a glimpse into the vast career opportunities available in the world of full stack web development and how our course can pave the way for your success.

Call - [+917827219784](tel:+917827219784)

Contact us

Official Address:
G-20, Preet Vihar, New Delhi - 110092

Registered Address
101/2 Rajiv Garden, Loni, Ghaziabad, Uttar
pradesh, 201102

Phone
+917827219784
E-mail
zokoworldservice@gmail.com

