Fundamentals

The MERN stack is a powerful framework for full-stack web development, combining MongoDB, Express.js, React.js, and Node.js to create robust, scalable applications. MongoDB serves as the NoSQL database, offering flexibility with JSON-like document storage, which aligns seamlessly with JavaScript-based technologies. Express.js acts as the lightweight and efficient backend framework, enabling smooth communication between the server and database. React.js handles the frontend with a component-based architecture, facilitating dynamic, interactive, and reusable UI development. Finally, Node.js provides the runtime environment for executing JavaScript on the server, ensuring high performance and scalability. At Tetraskills, we leverage the MERN stack to deliver cutting-edge, end-to-end web solutions that meet modern business needs efficiently and effectively.

HTML

Basics :-

- HTML Introduction
  Learn the fundamentals of HTML, including its purpose, structure, and how it's used to create webpages.
- HTML Editors
  Discover the tools required to write and edit HTML code, from basic text editors to advanced IDEs.
- HTML Basic
  Understand the foundation of HTML with examples of how to start building simple webpages.
- HTML Elements
  Explore the building blocks of HTML, including how tags, attributes, and content interact.
- HTML Attributes
  Learn how to enhance HTML elements using attributes like `id`, `class`, and `style`.
- HTML Headings
  Understand how to structure your document using different heading levels (`<h1>` to `<h6>`).
- HTML Paragraphs
  Learn how to create and format paragraphs using the `<p>` tag.

Core :-

- HTML Styles
  Learn how to apply inline styles to HTML elements for basic design and layout.
- HTML Formatting
  Explore text formatting tags like `<b>`, `<i>`, `<u>`, and others to style content.
- HTML Comments
  Learn to add comments in your code to make it more readable and maintainable.
- HTML Links
  Master creating hyperlinks using the `<a>` tag to connect webpages.
- HTML Images
  Understand how to add and optimize images using the `<img>` tag.
- HTML Lists
  Learn to create ordered, unordered, and description lists for better content organization.
- HTML Tables
  Understand table structure with `<table>`, `<tr>`, `<th>`, and `<td>` tags.

Intermediate :-

- HTML Colors
  Learn how to add colors to your HTML using names, HEX, RGB, and HSL values.
- HTML CSS
  Understand how to integrate CSS into HTML to style your webpages.
- HTML Favicon
  Learn how to add a favicon to your webpage using the `<link>` tag.
- HTML Page Title
  Understand the importance of the `<title>` tag in the `<head>` section of your document.
- HTML Block & Inline
  Learn the differences between block-level and inline-level elements.
- HTML Div
  Explore the `<div>` tag, a container element used to group and style sections of HTML.
- HTML Classes
  Learn how to use the class attribute to apply styles to multiple elements.
- HTML Id
  Understand how to use the id attribute for uniquely identifying elements.

- HTML JavaScript
  Learn how to integrate JavaScript into HTML using the `<script>` tag.
- HTML File Paths
  Understand the concept of absolute and relative file paths in HTML.
- HTML Head
  Learn about the `<head>` section and how it contains metadata, styles, and external links.
- HTML Layout
  Explore basic webpage layouts using containers, divs, and other elements.
- HTML Responsive
  Learn the basics of creating responsive webpages using meta tags and layout techniques.

Advanced :-

- HTML Forms
  Master the creation of interactive forms using the `<form>` element.
- HTML Form Attributes
  Learn advanced form attributes like `action`, `method`, `target`, and their use cases.
- HTML Form Elements
  Understand the various form elements like `<input>`, `<textarea>`, `<select>`, and `<button>`.
- HTML Input Types
  Explore the different input types like text, email, password, number, and more.
- HTML Input Attributes
  Learn advanced input attributes like `placeholder`, `maxlength`, `min`, and `required`.
- Input Form Attributes
  Dive deeper into attributes that can be applied to form inputs, enhancing their functionality.
- HTML Iframes
  Learn to embed content from external sources using the `<iframe>` tag.
- HTML Advanced Tables
  Master complex table structures using `colspan` and `rowspan` for multi-column and multi-row data.
- HTML Layout
  Explore basic webpage layouts using containers, divs, and other elements.
- HTML Responsive
  Learn the basics of creating responsive webpages using meta tags and layout techniques.

CSS

Basics :-

- CSS Introduction
  Learn the purpose of CSS, its role in styling webpages, and how it works alongside HTML.
- CSS Syntax
  Understand the basic syntax of CSS, including selectors, properties, and values.
- CSS Selectors
  Explore the different types of selectors and how they target specific HTML elements.
- CSS How To
  Learn how to apply CSS to your HTML using inline styles, internal styles, and external stylesheets.
- CSS Comments
  Discover how to add comments in CSS to make your code more organized and understandable.
- CSS Colors
  Learn how to add colors to your elements using color names, HEX, RGB, and HSL values.
- CSS Backgrounds
  Explore the various background properties to style webpage backgrounds effectively.
- CSS Borders
  Learn how to create and customize borders for your elements.
- CSS Margins
  Understand how to add space around elements using the `margin` property.
- CSS Padding
  Learn how to add space inside elements using the `padding` property.
- CSS Height/Width
  Master controlling the size of elements with the `height` and `width` properties.
- CSS Box Model
  Understand the box model, including content, padding, borders, and margins.

Core :-

- CSS Outline
  Learn how outlines differ from borders and how to style them.
- CSS Text
  Explore the various properties to style text, including color, alignment, spacing, and decoration.
- CSS Fonts
  Learn how to use different fonts in your designs, including custom fonts.
- CSS Links
  Understand how to style hyperlinks using pseudo-classes like `:hover`, `:visited`, and `:active`.
- CSS Lists
  Learn how to style ordered and unordered lists with custom markers and alignments.
- CSS Tables
  Discover how to style tables with borders, padding, and alignments.
- CSS Display
  Understand the `display` property and how it controls the layout of elements.
- CSS Position
  Learn about position properties (`static`, `relative`, `absolute`, `fixed`, `sticky`) and their use cases.
- CSS Z-index
  Understand how to control the stack order of elements using `z-index`.
- CSS Overflow
  Learn to handle overflowing content with the `overflow` property.
- CSS Float
  Discover how to use the `float` property to position elements.
- CSS Align
  Understand alignment techniques for text and elements using CSS.
- CSS Rounded Corners
  Learn to create rounded corners for elements using the `border-radius` property.
- CSS Gradients
  Explore linear and radial gradients for creating background effects.
-

Intermediate :-

- CSS Pseudo-classes
  Learn to style elements based on their state using pseudo-classes like `:hover`, `:focus`, and `:nth-child`.
- CSS Pseudo-elements
  Understand how to style parts of elements using pseudo-elements like `::before` and `::after`.
- CSS Shadows
  Learn how to create box and text shadows to add depth to your designs.
- CSS Navigation Bar
  Understand how to design responsive navigation bars with hover effects.
- CSS Dropdowns
  Learn to create dropdown menus using CSS.
- CSS Image Gallery
  Design beautiful image galleries with CSS.
- CSS Image Sprites
  Understand how to use image sprites to optimize webpage performance.
- CSS Forms
  Learn to style form elements for better user experience.
- CSS Website Layout
  Master basic website layouts using CSS.
- CSS Media Queries
  Understand how to use media queries to create responsive designs.
- CSS Flexbox
  Learn the Flexbox layout model to align and distribute space among items in a container.

Advanced :-

- CSS Animations
  Discover how to create animations using keyframes and transitions.
- CSS 2D Transforms
  Learn how to apply 2D transformations like scaling, rotating, and translating elements.
- CSS 3D Transforms
  Understand how to create 3D transformations for more advanced effects.
- CSS Grid
  Master the Grid layout model for creating complex and responsive layouts.
  - Grid Intro: Learn the basics of the Grid layout.
  - Grid Container: Understand container-specific properties.

    - Grid Item: Explore item-specific properties.
- CSS Variables
  Learn how to define reusable values in your stylesheets using custom properties (CSS variables).
- CSS Box Sizing
  Understand how the `box-sizing` property affects the box model and layout.
- CSS Tooltips
  Learn to create tooltips for better user guidance and interactivity.
- CSS Pagination
  Design pagination for multi-page content.
- CSS Counters
  Understand how to use counters to create numbered lists and custom sequences.
- CSS Transitions
  Master smooth transitions between CSS states.
- CSS User Interface
  Learn to style user interface elements like buttons, sliders, and progress bars.
- CSS Responsive
  Master responsive web design techniques.
    - RWD Intro: Overview of responsive web design.
    - RWD Viewport: Learn to control the viewport for mobile devices.
    - RWD Media Queries: Use media queries to adapt layouts.
    - RWD Grid View: Understand grid-based responsive design.
    - RWD Images: Optimize images for responsiveness.

Tailwind CSS

Basics :-

Getting Started with Tailwind CSS

- Installation and Setup: Learn how to install and set up Tailwind CSS in your project.
- Configuration (`tailwind.config.js`): Understand how to customize Tailwind CSS using the configuration file.

2. Utility-First Workflow

- Understanding the Utility-First Approach: Learn the philosophy behind Tailwind CSS and its utility-first methodology.
- Applying Classes for Styling Directly in HTML/JSX: Discover how to use Tailwind's utility classes to style elements directly in your code.

Core :-

Typography

- **Font Sizes, Weights, Styles: Learn how to style text with different font sizes, weights, and styles.**
- **Text Alignment and Decoration: Understand how to align text and apply decorations like underlines or line-through.**

2. Spacing Utilities

- **Padding and Margin: Explore how to control the spacing inside and around elements using padding and margin utilities.**
- **Gap (for Flex/Grid Layouts): Learn to set gaps between items in flexbox or grid layouts.**

3. Colors

- **Text Colors, Background Colors, and Border Colors: Apply colors to text, backgrounds, and borders using utility classes.**
- **Customizing the Color Palette: Understand how to extend or modify the default color palette in the Tailwind CSS configuration file.**

Intermediate :-

Flexbox and Grid Layouts

- Flexbox Utilities: Learn to use Tailwind's flexbox utilities for flexible and dynamic layouts.
- Grid-Based Responsive Layouts: Discover how to create responsive designs using grid utilities.

2. Responsive Design

- Breakpoints (`sm`, `md`, `lg`, `xl`, `2xl`): Understand how to apply styles conditionally based on screen size.
- Conditional Styling with Media Queries: Use media query utilities to create adaptive designs.

3. Hover, Focus, and Active States

- Pseudo-Class Utilities: Explore utilities for hover, focus, active, and other interactive states.
- Styling Interactive Elements: Learn to style buttons, links, and other interactive elements effectively.

4. Typography Enhancements

- Line Height, Letter Spacing: Customize the readability of text with line height and letter spacing utilities.

- Custom Font Families: Learn to integrate and apply custom fonts in your Tailwind projects.

Advanced :-

Customizing Tailwind

- Adding Custom Themes and Plugins: Learn how to add custom themes and integrate plugins into your Tailwind setup.
- Extending the Default Configuration: Understand how to extend Tailwind's default configuration to include additional utilities and styles.

2. Animations and Transitions

- Using Transition, Duration, Ease: Master creating smooth transitions and controlling their timing and easing.
- Tailwind's Animation Utilities: Discover Tailwind's built-in animation classes for simple and effective animations.

3. Dark Mode

- Implementing and Toggling Dark Mode: Learn to enable and switch between dark and light modes in your projects.
- Theming with Tailwind: Understand how to design and customize themes for both light and dark modes.

For React-Specific Use :-

Dynamic Class Binding

- Using `classnames` or `clsx` Packages: Learn how to dynamically bind Tailwind classes in JavaScript/React using popular helper libraries.
- Conditional Class Application in JSX: Understand how to apply classes conditionally based on state or props.

2. Reusable Components with Tailwind

- Structuring UI Components (e.g., Buttons, Cards): Explore how to design and organize reusable UI components with Tailwind CSS.
- Prop-Based Styling for Reusability: Learn to use component props to apply dynamic Tailwind styles for consistent and adaptable designs.

3. Headless UI Integration

- Leveraging Components like Modals, Dropdowns, etc.: Integrate Tailwind with Headless UI to build accessible and customizable UI components like modals and dropdowns.

Javascript

Basics :-

- Introduction to JavaScript Course Topics
  Overview of what you will learn in this course.
- How Websites Work with JavaScript | Beginner's Guide
  Learn how JavaScript integrates with HTML and CSS to make websites interactive.
- What is JavaScript? | Programming Essentials
  Introduction to JavaScript as a programming language and its key features.
- History of JavaScript | Evolution and Milestones
  A brief history of JavaScript and its evolution over the years.
- Creating First JavaScript File in VS Code | Step-by-Step Guide
  Learn to set up and write your first JavaScript file using VS Code.
- Values and Variables in JavaScript
  Understand how to declare variables and work with values in JavaScript.
- Data Types in JavaScript | Web Development Basics
  Explore different data types like strings, numbers, and booleans.
- Bonus: Advanced Data Types in JavaScript | Programming Tips
  Learn about advanced data types like objects and arrays.
- Expressions and Operators in JavaScript
  Discover how to write expressions and use arithmetic, comparison, and logical operators.
- Functions in JavaScript
  Learn how to create reusable blocks of code using functions.

Core :-

- Arrays in JavaScript | Programming Challenges
  Understand arrays and how to perform operations like adding and removing elements.
- Strings in JavaScript | String Operations
  Learn string manipulation techniques such as concatenation and slicing.

- Math Objects in JavaScript | Advanced Concepts
  Explore the Math object to perform mathematical operations like rounding or generating random numbers.
- Window, Document, and Browser Object Models | Web Development Concepts
  Understand the browser environment and how JavaScript interacts with it.
- Events in JavaScript
  Learn how to handle user interactions like clicks and keypresses.
- Project: Background Gradient Generator
  Create a tool to generate custom gradients using JavaScript.
- Update on Hosting
  Understand the basics of hosting and deploying JavaScript projects.
- LocalStorage in JavaScript
  Learn how to store data persistently in the browser using LocalStorage.

Intermediate :-

- Date & Time Objects
  Work with dates and times using JavaScript's built-in Date object.
- Timing-based Events in JavaScript
  Learn about setTimeout and setInterval for creating timed actions.
- Objects in JavaScript
  Understand how to create and work with objects in JavaScript.
- ECMAScript 2015 – 2023
  Explore modern JavaScript features introduced in recent ECMAScript versions.
- Lexical & Scope Chaining
  Learn about variable scope and how JavaScript resolves variables.
- Closures in JavaScript
  Understand closures and their use cases in programming.
- First-Class Functions – Callbacks & Higher-Order Functions
  Discover the concept of first-class functions and how to use callbacks and higher-order functions.

Advanced :-

- Promises
  Learn how to handle asynchronous operations using promises.
- Project: Dad Jokes with Promises
  Build a project that fetches jokes using APIs and handles them with promises.

- Project: Dad Joke with Async Await & Try Catch
  Enhance the jokes project by using async/await and error handling techniques.
- Advanced JavaScript
  Dive deeper into advanced JavaScript concepts like prototypes and event loops.
- Final Project: CRUD Operation Using the API
  Build a full-featured project to Create, Read, Update, and Delete data using APIs.

React

Basics :-

- Introduction | What is React
  Understand the basics of React and its core purpose.
- Set Up React Environment | Install React project in Windows
  Learn how to set up a React environment on Windows.
- Install React in macOS | React.js setup in MacOS
  Step-by-step guide to setting up React on macOS.
- Use and Learn React without install | Vite Code editor
  Explore how to use React without installation.
- Difference Between Framework and Library
  Understand the key differences between a framework and a library.
- Hello World | React.js app Code Flow
  Create a simple "Hello World" app and understand React's code flow.
- File and Folder Structure
  Learn the standard structure of a React project.
- What is Component | Make First Component
  Understand React components and create your first one.
- Importing and Exporting Components
  Learn how to import and export components in React.
- Write Markup with JSX
  Understand JSX and how it integrates with React.
- JSX Exercise
  Practice creating JSX code.
- JSX Exercise Implementation
  Apply the JSX concepts with an exercise solution.
-

Core :-

- JavaScript in JSX with Curly Braces
  Learn how to embed JavaScript in JSX.
- React.js Click Event and Function Call
  Handle user interactions with click events.
- State in React JS
  Understand state management in React.
- Project: Toggle or Hide Show
  Build a toggle/hide-show feature using React.
- Multiple Conditions of Else If
  Handle multiple conditions in React logic.
- Props in React.js | Pass Data Between Components
  Learn how to pass data between components using props.
- Default Props
  Set default values for props in React.
- Get Input Field Value | On Change Event
  Handle input values dynamically with `onChange`.
- Handle Checkbox in React
  Learn how to work with checkboxes in forms.
- Looping with Map Function
  Render lists using the map function.
- Reuse Component in Loop
  Reuse components dynamically within loops.
- Array Nested Looping with Components
  Work with nested arrays in React.
-

Intermediate :-

1. What are Hooks in React.js
   Introduction to React hooks and their purpose.
2. UseState Hooks
   Manage state in functional components with `useState`.
3. Use of useEffect Hook in React.js
   Understand side effects and use the `useEffect` hook.
4. UseEffect Hook for Lifecycle Methods in React
   Mimic lifecycle methods in functional components.
5. Dynamic and Conditional Inline Styles
   Apply styles dynamically in React components.

6. Project: How API Call in UseEffect (CRUD Operation)
   Fetch, create, update, and delete data using APIs in React.
7. Context API
   Share state and props globally with Context API.
8. UseReducer Hook
   Learn advanced state management with `useReducer`.
9. UseRef Hook
   Work with refs to manipulate DOM elements.
10. UseLayoutEffect Hook
   Optimize rendering with `useLayoutEffect`.
11. UseMemo Hook
   Improve performance by memoizing expensive calculations.
12. UseCallback Hook
   Memoize functions for better performance.
13. Custom Hook
   Create reusable custom hooks in React.

Advanced :-

- Intro React-Router-Dom
  Introduction to React Router for navigation.
- What is React Routing
  Learn the fundamentals of routing in React.
- Why Use Client-Side Routing
  Understand the benefits of client-side routing.
- React Router Overview
  A detailed overview of React Router.
- Key Concepts of React Router
  Learn essential concepts like route matching and navigation.
- How Does Routing Work in React
  Understand the mechanism behind React routing.
- Setting Up React Router
  Set up React Router in your project.
- Dynamic Routing with URL Parameter
  Learn to create routes with dynamic parameters.
- Nested Routes
  Understand and implement nested routes.
- Programmatic Navigation
  Navigate between pages programmatically using React Router.
- Protected Routes (Authentication)
  Implement routes that require user authentication.

- Handling 404 Pages (Not Found)
  Create custom 404 error pages for undefined routes.

NodeJS

Introduction to Node.js :-

Goal: Understand the basics of Node.js and set up your environment.

- Introduction to Node.js:

– What is Node.js?
– Why use Node.js?
– Node.js architecture and event-driven model.

- Setup:

– Install Node.js and npm (Node Package Manager).
– Verify installation with `node -v` and `npm -v`.

- First Steps:

– Write a simple “Hello World” program.
– Execute JavaScript files using Node.js.

Understanding Core Concepts :-

Goal: Learn about the core modules and concepts in Node.js.

- Modules:

– Understanding built-in modules (e.g., `fs`, `http`, `path`).
– Learn how to import and use modules.

- File System:

– Reading from and writing to files.
– Working with directories.

- Creating a Simple Server:

– Use the `http` module to create a basic web server.
– Understand request and response objects.

Deep Dive into Asynchronous Programming :-

Goal: Master asynchronous programming in Node.js.

- Callbacks:

– Understanding the callback pattern.
– Handling errors in callbacks.

- Promises:

– Introduction to Promises.
– Converting callback-based functions to use Promises.

- Async/Await:

– Using async/await syntax for cleaner asynchronous code.
– Error handling in async/await.

ExpressJS

Working with Express.js :-

Goal: Learn to use Express.js for building web applications.

- Introduction to Express.js:

– Setting up a basic Express application.
– Understanding middleware.

- Routing:

– Define and use routes.
– Handle different HTTP methods.

- Introduction to Postman:

– How to handle different Http models

MongoDB

Connecting to a Database :-

Goal: Learn to integrate a database with your Node.js application.

- Introduction to Databases:
– Overview of SQL vs NoSQL databases.
– Install and set up a simple database (e.g., MongoDB).

- Using Mongoose with MongoDB:
– Set up Mongoose for MongoDB.
– Create schemas and models.
– Perform basic CRUD operations.

Building a Project :-

Goal: Build a simple project.

- Project Setup:
– Define the project (e.g., a simple RESTful API).
– Set up the project structure.

- Building the Backend:
– Implement routes and controllers.
– Connect to the database and implement CRUD operations.

- Testing and Debugging:
– Use tools like Postman to test your API endpoints.
– Debug and fix any issues.

- Project Completion:
– Add any final features or enhancements.
– Document your code and create a README file.

Git and Github

Git Basics :-

Introduction to Git

- What is Version Control? Why Git?
- Installing Git (Windows/Mac/Linux)
- Setting up Git for the First Time:
    - `git config --global user.name "Your Name"`

    - `git config --global user.email "youremail@example.com"`
- Initializing a Repository: `git init`
- Adding Files to Staging:
    - `git add .`
    - `git add <file>`
- Committing Changes: `git commit -m "Initial Commit"`
- Viewing Commit History: `git log`
- Understanding the Three Stages:
    - Working Directory
    - Staging Area
    - Repository

Core Git Concepts :-

Branching and Merging

- What Are Branches?
- Creating Branches: `git branch <branch-name>`
- Switching Branches: `git checkout <branch-name>`
- Merging Branches: `git merge <branch-name>`
- Resolving Merge Conflicts

Undoing Changes

- Undoing Changes in the Working Directory: `git checkout -- <file>`
- Resetting Commits:
    - `git reset --soft`
    - `git reset --mixed`
    - `git reset --hard`
- Deleting a Branch: `git branch -d <branch-name>`

Connecting Git and GitHub :-

Linking Git and GitHub

- Linking Local Repository to GitHub:
    - `git remote add origin <repository-URL>`
- Pushing Code to GitHub:
    - `git push -u origin main`
- Pulling Code from GitHub:
    - `git pull origin main`

Collaborating with GitHub

- Forking a Repository
- Cloning a Repository: `git clone <repository-URL>`
- Creating and Managing Pull Requests
- Reviewing and Merging Pull Requests