Mastering Web Development: A Comprehensive Workshop

By K P RANJITH KUMAR

Day 1: Introduction to Web Development & Building a Static Website

- Introduction to Web Development:

  Overview of how websites work, understanding the front-end and back-end.

- HTML Basics:
  - Introduction to HTML
  - Structure of an HTML document
  - Tags, Elements, and Attributes
  - Creating and organizing content with headings, paragraphs, lists, images, links, tables, and forms.
- CSS Basics:
  - Styling HTML with CSS
  - Selectors, Properties, and Values
  - Inline, internal, and external CSS
  - Understanding the box model, margins, padding, and borders.
  - Basic layout with Flexbox
- JavaScript Basics:
  - Introduction to JavaScript
  - Variables, data types, and operators
  - Functions and events in JavaScript

    - DOM manipulation
    - Basic form validation
- Practical: By the end of the day, students will have created a basic static website using HTML, CSS, and JavaScript, which they will enhance in the following days.

Day 2: Advanced Front-End Development & GitHub Pages Deployment

- Advanced CSS Techniques:
    - Positioning elements: relative, absolute, fixed, and sticky positioning
    - Advanced layout techniques with Flexbox and CSS Grid
    - Responsive Design: media queries and flexible grids
    - Using Google Fonts and integrating Font Awesome for icons.
- Advanced JavaScript Concepts:
    - JavaScript ES6 features (let, const, arrow functions, etc.)
    - Working with arrays and objects
    - Manipulating the DOM further (adding and removing elements, event delegation)
    - Simple animations using CSS and JavaScript
- GitHub Basics:
    - Introduction to version control and Git
    - Setting up a GitHub repository
    - Pushing code to GitHub
- GitHub Pages Deployment:
    - Deploying a static site using GitHub Pages
    - Customizing URLs and managing updates.
- Practical: Deploy the static site developed on Day 1 to GitHub Pages and work on improving the design using advanced CSS and JavaScript.

Day 3: Working with APIs, Node.js & Express

- Understanding APIs:
    - Introduction to APIs and JSON
    - Fetching data from an API using JavaScript (Fetch API)
    - Displaying API data on the website (e.g., weather data, user information)
- Introduction to Node.js:
    - What is Node.js and why use it for server-side development?
    - Installing Node.js and setting up the development environment
    - Introduction to npm and managing packages
- Express.js Basics:
    - Introduction to Express.js and its uses
    - Setting up a basic Express server
    - Handling GET and POST requests in Express
    - Using body-parser to handle form data
- Practical: By the end of the day, students will have set up a Node.js and Express server, allowing them to capture and display user data in the terminal using localhost.

Day 4: MongoDB Integration & User Authentication

- Introduction to Databases:
    - Introduction to databases and how they store data
    - Differences between SQL and NoSQL databases
    - Why use MongoDB?
- MongoDB Setup:
    - Installing MongoDB locally
    - Introduction to MongoDB Compass for data visualization

    - Understanding collections and documents
- Performing CRUD Operations:
    - Connecting Node.js to MongoDB with Mongoose
    - Creating, Reading, Updating, and Deleting data in MongoDB
    - Understanding Mongoose schemas and models
- User Authentication:
    - Introduction to user authentication concepts
    - Hashing passwords with bcrypt
    - Implementing Sign Up and Login functionality
    - Using JSON Web Tokens (JWT) for secure sessions
- Practical: Students will implement a simple registration and login system with MongoDB as the database, allowing users to securely sign up and log in.

Day 5: Building a Full Dynamic Website

- Project Setup:
    - Reviewing and combining all the concepts learned from the previous days
    - Structuring the project for scalability (frontend, backend, database)
- Creating a Fully Functional Website:
    - Frontend: Enhancing the website's user interface and user experience (UI/UX)
    - Backend: Handling user requests, processing form data, interacting with MongoDB

    - Dynamic content: Displaying data fetched from the database on the frontend
    - Implementing complete user authentication (Sign Up, Login, Logout)
- Final Project Review: By the end of the day, students will have built and deployed a fully functional dynamic website that interacts with a database, includes user authentication, and is live for the world to see.
