Overview

The **Web Development Fundamentals** course is a comprehensive, beginner-friendly program designed to take learners from having no prior coding experience to confidently creating, styling, and publishing their own interactive websites.

Over **30 hours** of immersive, hands-on sessions, students will explore the three core technologies of front-end web development — **HTML**, **CSS**, and **JavaScript** — through a structured, project-based approach.

The course emphasizes *learning by doing*: each stage concludes with a mini-project, allowing learners to immediately apply concepts in practical, real-world contexts. By the end, participants will have built and deployed a complete responsive website and gained the foundational skills required to pursue advanced frameworks like React, Tailwind, or backend development.

Stage 1: Web Introduction & HTML Basics (2 hours)

Objective: Understand how the web works and build the first HTML page.
Topics Covered:

- Overview of the Web: what is a website, how browsers and servers interact.
- Setting up development environment: text editor, browser, project folder.
- HTML Document Structure: <!DOCTYPE html>, <html>, <head>, <body>.
- Basic HTML Tags: headings (<h1>–<h6>), paragraphs (<p>), links (<a href>), images (<img src alt>).
- Embedding media (images) and simple links between pages.
- Saving and opening files locally in a browser.
  Mini-Project: Create an “About Me” page with heading, an image, a paragraph and a link.
  Learning Outcome: Students create a valid HTML page and understand basic page structure.

Stage 2: HTML Layout & Forms (2 hours)

Objective: Explore more HTML tags for layout and user input.
Topics Covered:

- Semantic layout tags: <header>, <nav>, <section>, <article>, <footer>, <aside>.
- Lists: ordered (<ol>), unordered (<ul>), list items (<li>).
- Divs/spans for grouping content: , .
- Forms and input fields: <form>, <input type="text">, <input type="email">, <textarea>, <select>, <option>, <button>.
- Attributes: placeholder, required, minlength, etc.
- Embedding video/audio (optional): <video>, <audio>.
  Mini-Project: Build a survey/registration page with input fields (name, email), select list, submit button.
  Learning Outcome: Students can structure pages with semantic tags and create simple forms for user input.

Stage 3: CSS Fundamentals (4 hours)

Objective: Learn to style HTML pages using CSS and understand the box model and selectors.
Topics Covered:

- Linking CSS to HTML: inline (style=), internal (<style>), external (<link>).
- Selectors: element, class (.classname), ID (#idname).
- Basic properties: color, background-color, font-family, font-size, text-align.
- Box model: content, padding, border, margin. Understanding how spacing works.
- Setting width/height, display types (block, inline, inline-block).
- Using external fonts (Google Fonts) and basic typography.
- Styling links (:hover, :active) and lists.
  Mini-Project: Style the “About Me” and form page: consistent fonts, colors, list styles, spacing.
  Learning Outcome: Students apply styles and control basic layout and typography visually.

Stage 4: CSS Layouts & Responsive Design (4 hours)

Objective: Build page layouts that work on different screen sizes.
Topics Covered:

- Flexbox: display: flex, flex-direction, justify-content, align-items, flex-wrap.
- CSS Grid (basic): display: grid, grid-template-columns, grid-template-rows, grid-gap.
- Media queries: @media (max-width:...) to adapt to mobile screens.
- Relative units: %, vw, vh, rem, versus fixed px.
- Mobile-first design approach.
- Responsive images: max-width: 100%, height: auto.
  Mini-Projects:
- Create a two-column layout (sidebar + main content) that stacks to one on mobile.
- Build a responsive image gallery grid with varying number of columns on mobile vs desktop.
  Learning Outcome: Students develop layouts that adapt to screen size and use modern layout tools (Flexbox/Grid).

Stage 5: CSS Enhancements & Visual Effects (3 hours)

Objective: Add visual polish to websites using advanced CSS features.
Topics Covered:

- Pseudo-classes and pseudo-elements: :hover, :focus, ::before, ::after.
- CSS Transitions: transition property, smooth changes of properties.
- CSS Animations: @keyframes, animation-name, animation-duration, animation-iteration-count.
- Shadows (box-shadow, text-shadow), gradients (linear-gradient, radial-gradient).
- Button styles, hover effects, image effects (filter, transform).
- Accessibility basics: contrast, alt text for images, logical heading order.
  Mini-Project: Create a “stylised card” component: image + title + description, with hover animation and responsive design.

Learning Outcome: Students build visually engaging, interactive UI elements using CSS.

Stage 6: JavaScript Basics & DOM Interaction (4 hours)

Objective: Introduce JavaScript to make web pages interactive.
Topics Covered:

- What is JavaScript and where it sits in the web page (<script> tag, external .js file).
- Variables and data types: let, const, string, number, boolean.
- Console logging (console.log), prompt (prompt()), alert (alert()), confirm (confirm()).
- Functions: declaration, calling, parameters, return value.
- Events: onclick, onchange, onkeyup, addEventListener().
- Basic DOM access and manipulation: document.getElementById(), document.querySelector(), innerHTML, textContent.
- Changing styles or classes via JS: element.style.property, element.classList.add/remove.
  Mini-Projects:
- A button that toggles background color of the page.
- A text input and button that take user input and display a message.
  Learning Outcome: Students connect JavaScript code to HTML/CSS and respond to user interactions.

Stage 7: JavaScript Logic & Interactive Features (4 hours)

Objective: Build logic and interactive applications using JavaScript.
Topics Covered:

- Conditional statements: if, else if, else.
- Loops: for, while, iterating through arrays.
- Arrays: creating arrays, push(), pop(), simple array traversal.
- Simple form validation: checking input length, required fields, and giving feedback.

- Dynamic HTML modification: creating and deleting elements (document.createElement(), appendChild(), removeChild()).
- Simple debugging: using console.log, breakpoints in browser dev tools.
  Mini-Projects:
- A quiz app: ask 3–5 questions, track score, show result.
- A to-do list app: add tasks, delete tasks, mark tasks completed.
  Learning Outcome: Students write logic, manage lists/arrays, dynamically update the web page and handle user data.

Stage 8: Final Project – Responsive Interactive Website (5 hours)

Objective: Integrate HTML, CSS and JavaScript to build a full website with interactivity and responsive design.
Project Brief: Choose one of:

- A portfolio site with interactive skill list and contact form.
- A product or service page with image gallery, toggle features and contact form.
- A mini-game or quiz site with interactive elements and scoring.
  Project Requirements:
- Use semantic HTML structure.
- Apply styled CSS for layout (Flexbox/Grid) and responsiveness (media queries).
- Add JavaScript interactivity: button events, form validation, dynamic content display.
- Test the site on desktop and mobile.
  Deliverable: Full website demo, share the link or host via GitHub Pages.
  Learning Outcome: Students consolidate skills, produce a portfolio-worthy project and demonstrate their mastery of web fundamentals.

Stage 9: Deployment & Next Steps (2 hours)

Objective: Publish the website and learn where to progress next.
Topics Covered:

- Hosting basics: free platforms like GitHub Pages or Netlify.
- Basic version control: git init, git add ., git commit, git push. (Optional but introduced.)
- Accessibility and performance basics: alt text for images, proper heading order, mobile testing.
- Next-step topics: CSS frameworks (Bootstrap, Tailwind), JS libraries (jQuery, React), backend basics, APIs.
  Mini-Project: Deploy your final project and share the live link.
  Learning Outcome: Students publish their website publicly and have a roadmap for further learning.

Summary Table

Stage	Duration	Focus
1	2 hrs	Web intro & HTML basics
2	2 hrs	HTML layout & forms
3	4 hrs	CSS fundamentals
4	4 hrs	CSS layouts & responsive design
5	3 hrs	CSS enhancements & visual effects
6	4 hrs	JavaScript basics & DOM interaction
7	4 hrs	JavaScript logic & interactive features
8	5 hrs	Final project: responsive interactive website
9	2 hrs	Deployment & next steps
Total	**30 hrs**	Core web fundamentals

Grade-Level Adaptation

Grade	Typical Age	Suggested Stages
7–8	12–13	Stages 1–3
9	13–14	Stages 1–5
10–11	15–16	Stages 1–8
12	16–18	Full Curriculum