

# 10-Day Web Development Course

HTML + CSS + Bootstrap + JavaScript

A beginner-friendly, hands-on course plan with daily study material and assessments.

## What You Get

- 10 day plan (topics + key takeaways).
- Daily starter code snippets.
- Daily assessments (quiz + practical task).
- Final project rubric + quiz answer keys.

## Course Outline

- Day 1: HTML Foundations
- Day 2: Semantic HTML and Forms
- Day 3: CSS Basics and the Box Model
- Day 4: CSS Layout with Flexbox and Positioning
- Day 5: Responsive Design and Reusable Components
- Day 6: Bootstrap Fundamentals
- Day 7: Bootstrap Components and a Landing Page
- Day 8: JavaScript Basics and DOM
- Day 9: JS Forms, localStorage, and Dynamic Lists
- Day 10: Final Project and Evaluation

# Day 1 - HTML Foundations

## Topics

- HTML page structure: doctype, html, head, body
- Text content: headings, paragraphs, line breaks
- Links and images (with alt text)
- Lists (ul/ol) and clean file structure

## Key Takeaway

- Use a consistent folder per day and name files clearly (index.html, images/).

## Starter Code

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Day 1 - Profile</title>
</head>
<body>
<h1>Hello Web</h1>
<p>This is my first HTML page.</p>

<a href="https://example.com" target="_blank" rel="noreferrer">Visit</a>
</body>
</html>
```

## Practice

- Create a page with headings, paragraphs, and a list.

## Assessment - Quiz (Questions)

1. HTML stands for what?	2. Which tag contains metadata and links to CSS?
3. Largest heading tag?	4. Tag used to create a link?
5. Tag used to display an image?	6. Attribute to open link in a new tab?
7. Ordered list tag?	8. Attribute used for image description?

## Assessment - Practical Task

Build a one-page Profile (name, photo, short bio, skills list, and a contact link). No CSS yet.

## Submission Checklist

- Valid HTML structure (doctype, head, body)
- At least one image with alt text
- At least one external link that works

## Day 2 - Semantic HTML and Forms

### Topics

- Semantic layout: header, nav, main, section, footer
- Tables: table, tr, th, td (basic usage)
- Forms: input types, label, select, textarea, button
- Built-in validation: required, minlength, type=email

### Key Takeaway

- Semantic tags improve readability and SEO; they act like meaningful divs.

### Starter Code

```
<form>
<label for="name">Full Name</label>
<input id="name" type="text" required />

<label for="email">Email</label>
<input id="email" type="email" required />

<label for="dept">Department</label>
<select id="dept" required>
<option value="">Choose</option>
<option>CSE</option>
<option>ECE</option>
</select>

...
```

### Practice

- Create a semantic page skeleton with header/nav/main/footer.

### Assessment - Quiz (Questions)

1. Semantic tag commonly used for navigation?	2. Tag to connect text with a form field?
3. Input type for email validation?	4. Attribute to make a field mandatory?
5. Table header cell tag?	6. Multi-line input tag?
7. Dropdown list tag?	8. Which element groups the main unique page content?

### Assessment - Practical Task

Create a Student Registration page with semantic layout and a form (name, email, phone, department, gender, skills, comments).

### Submission Checklist

- Uses header/nav/main/footer
- All inputs have labels
- Uses correct input types (email, tel, etc.)

## Day 3 - CSS Basics and the Box Model

### Topics

- Ways to add CSS: external (recommended), internal, inline
- Selectors: element, class, id; basic specificity
- Typography: font-size, font-weight, line-height
- Box model: margin, padding, border; border-radius and shadows

### Key Takeaway

- Start with typography and spacing before colors and effects.

### Starter Code

```
.card{
width: 320px;
border: 1px solid #ddd;
padding: 16px;
border-radius: 12px;
}
.card h2{ margin: 0 0 8px; }
.muted{ opacity: 0.75; }
```

### Practice

- Style your headings and paragraph spacing for readability.

### Assessment - Quiz (Questions)

1. Symbol for class selector?	2. Symbol for id selector?
3. Property for text color?	4. Property for background color?
5. Space inside the border?	6. Space outside the border?
7. Property for rounded corners?	8. How do you link external CSS?

### Assessment - Practical Task

Style your Day 1 profile into a centered card layout with clean spacing, typography, and a button-style link.

### Submission Checklist

- External CSS file used
- Consistent spacing (margin/padding)
- Readable typography (size/line-height)

# Day 4 - CSS Layout with Flexbox and Positioning

## Topics

- Layout foundations: display types and container width
- Flexbox: justify-content, align-items, gap, flex-wrap
- Positioning: relative, absolute, sticky (practical uses)
- Build a responsive card grid

## Key Takeaway

- Use flex-wrap for responsive rows and gap for consistent spacing.

## Starter Code

```
.row{
  display: flex;
  gap: 12px;
  flex-wrap: wrap;
}
.box{
  flex: 1 1 220px;
  padding: 14px;
  border: 1px solid #ddd;
  border-radius: 12px;
}
```

## Practice

- Make a 3-column card row using flex-wrap and gap.

## Assessment - Quiz (Questions)

1. Property to enable flex layout?	2. Horizontal alignment in flex?
3. Vertical alignment in flex?	4. Allow wrapping to next line?
5. Spacing between flex items?	6. Sticky positioning property?
7. Absolute positioning is relative to what?	8. Flex shorthand order is?

## Assessment - Practical Task

Create a Services section with 6 cards using Flexbox. It must wrap nicely on mobile.

## Submission Checklist

- Uses flex container + wrapping
- Cards have consistent height/padding
- Looks good on mobile and desktop

# Day 5 - Responsive Design and Reusable Components

## Topics

- Responsive basics: media queries and mobile-first thinking
- Responsive images: max-width: 100%
- Create reusable UI components (button, badge, card)
- Optional: CSS variables for theme tokens

## Key Takeaway

- Design mobile-first: make it work on small screens, then scale up.

## Starter Code

```
.container{ max-width: 1000px; margin: 0 auto; padding: 16px; }
img{ max-width: 100%; height: auto; }

@media (max-width: 600px){
.container{ padding: 10px; }
}
```

## Practice

- Add media queries for mobile spacing and typography.

## Assessment - Quiz (Questions)

1. Media queries start with which keyword?	2. Best rule for responsive images?
3. Unit relative to root font size?	4. Mobile-first breakpoints often use?
5. Where are CSS variables commonly declared?	6. Hide an element completely?
7. Center a block element?	8. Make text bold?

## Assessment - Practical Task

Make previous pages mobile-friendly and build 3 reusable components: button, card, badge.

## Submission Checklist

- Mobile layout works (no horizontal scroll)
- Reusable classes used across pages
- Consistent spacing scale

# Day 6 - Bootstrap Fundamentals

## Topics

- Bootstrap setup via CDN
- Grid: container, row, col; breakpoints
- Utilities: spacing, text, background, borders
- Components: buttons, cards, alerts

## Key Takeaway

- Use Bootstrap utilities to reduce custom CSS for common spacing and alignment.

## Starter Code

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<div class="container py-4">
<div class="row g-3">
<div class="col-12 col-md-6 col-lg-4">
<div class="card p-3 shadow-sm">
<h5 class="mb-1">Card Title</h5>
<p class="text-muted mb-0">Bootstrap card content</p>
</div>
</div>
</div>
</div>
```

## Practice

- Recreate a layout using container/row/col only.

## Assessment - Quiz (Questions)

1. Bootstrap base wrapper class?	2. Bootstrap grid row class?
3. What does col-md-6 mean?	4. What does g-3 control?
5. Padding utility p-3 means?	6. Muted text utility class?
7. Primary button classes?	8. Bootstrap includes what kind of classes mostly?

## Assessment - Practical Task

Rebuild your Services section using Bootstrap grid and cards (minimal custom CSS).

## Submission Checklist

- Correct grid usage across breakpoints
- Consistent spacing using utilities
- Cards look consistent

# Day 7 - Bootstrap Components and a Landing Page

## Topics

- Navbar (responsive) and basic navigation structure
- Forms styling, badges, tables (as needed)
- Modals (optional) and simple interactions
- Build a complete landing page layout

## Key Takeaway

- Navbar structure matters: brand, toggler, collapse, links.

## Starter Code

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
<div class="container">
<a class="navbar-brand" href="#">MySite</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#nav">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="nav">
<ul class="navbar-nav ms-auto">
<li class="nav-item"><a class="nav-link" href="#home">Home</a></li>
<li class="nav-item"><a class="nav-link" href="#about">About</a></li>
</ul>
</div>
</div>
</nav>
```

## Practice

- Create a responsive navbar with collapse.

## Assessment - Quiz (Questions)

1. Main navbar class?	2. Responsive toggle button class?
3. Collapse wrapper class?	4. Form input styling class?
5. Rounded utility example?	6. Shadow utility example?
7. Center text utility class?	8. Vertical spacing utility example?

## Assessment - Practical Task

Build a full landing page: Navbar, Hero, 3 sections (Features/Services/Testimonials), and Footer.

## Submission Checklist

- Complete page structure and sections
- Responsive navbar works
- Consistent spacing and typography

# Day 8 - JavaScript Basics and DOM

## Topics

- Variables: let/const; data types
- Functions, conditions, loops (practical use)
- DOM selection: getElementById, querySelector
- Events: click, input; classList, textContent

## Key Takeaway

- Use event listeners instead of inline onclick for cleaner code.

## Starter Code

```
<button id="toggleBtn">Toggle</button>
<p id="msg">Hello</p>

<script>
const btn = document.getElementById('toggleBtn');
const msg = document.getElementById('msg');

btn.addEventListener('click', () => {
  msg.classList.toggle('d-none'); // d-none works great with Bootstrap
});
</script>
```

## Practice

- Add a theme toggle that persists during the session.

## Assessment - Quiz (Questions)

1. Block-scoped variable keyword?	2. Constant (cannot be reassigned) keyword?
3. Select one element (CSS selector)?	4. Add a class to an element?
5. Click event name?	6. Print to console?
7. Update text content property?	8. Arrow function uses which symbol?

## Assessment - Practical Task

Add 3 interactions to your landing page: theme toggle, smooth scroll, and dynamic footer year.

## Submission Checklist

- All interactions work without errors
- Uses event listeners (no inline JS)
- Uses class toggles for UI states

# Day 9 - JS Forms, localStorage, and Dynamic Lists

## Topics

- Form handling: submit event and preventDefault
- Validation patterns (show errors clearly)
- localStorage basics: setItem/getItem, JSON stringify/parse
- Render a list dynamically from an array

## Key Takeaway

- Store arrays/objects using JSON.stringify and read back with JSON.parse.

## Starter Code

```
const items = JSON.parse(localStorage.getItem("notes") || "[]");

function save(){
localStorage.setItem("notes", JSON.stringify(items));
}
```

## Practice

- Create a notes list that renders from an array.

## Assessment - Quiz (Questions)

1. Stop form from reloading the page?	2. localStorage stores values as what type?
3. Convert object to string?	4. Convert string to object/array?
5. Read from localStorage?	6. Write to localStorage?
7. Add item to an array?	8. Remove item by index uses which method?

## Assessment - Practical Task

Build a Notes/To-Do widget with add, delete, and persistence using localStorage.

## Submission Checklist

- Renders list items dynamically
- Add and delete both work
- Data persists after refresh

# Day 10 - Final Project and Evaluation

## Topics

- Plan folder structure and shared components
- Build 3 pages: Home, About, Contact
- Bootstrap layout + small custom CSS polish
- JavaScript: validation + one dynamic UI feature + localStorage use

## Key Takeaway

- Keep shared elements consistent across pages (nav, footer, buttons).

## Starter Code

```
Suggested project structure:  
project/  
index.html  
about.html  
contact.html  
assets/  
css/style.css  
js/app.js  
img/
```

## Practice

- Make a 3-page site with shared styling.

## Assessment - Quiz (Questions)

1. Semantic tag for footer?	2. CSS class selector symbol?
3. Flex horizontal alignment property?	4. Bootstrap grid uses how many columns?
5. Select element by id in JS?	6. Prevent form refresh?
7. Make image responsive in CSS?	8. Convert to JSON string?

## Assessment - Practical Task

Final Project: A responsive 3-page site (Home/About/Contact) with JS validation and a localStorage feature (newsletter signup or saved notes).

## Submission Checklist

- Responsive layout across devices
- Clean HTML semantics and structure
- Bootstrap used correctly (grid/components)

## Final Project Rubric

Suggested submission format:

- Project folder with index.html, about.html, contact.html, and an assets/ folder.
- A README.txt with: features implemented, how to run, and any known limitations.
- At least 1 screenshot per page (optional but helpful for evaluation).

### Scoring (50 points)

Category	Points
UI layout + responsiveness	15
Correct Bootstrap usage	10
Clean HTML semantics	10
JavaScript features working	10
Folder structure + readability	5

## Appendix - Quiz Answer Keys

Answers are listed for questions 1-8 in order.

**Day 1:** 1) HyperText Markup Language | 2) <head> | 3) <h1> | 4) <a> | 5) <img> | 6) target="\_blank" | 7) <ol> | 8) alt

**Day 2:** 1) <nav> | 2) <label> | 3) type="email" | 4) required | 5) <th> | 6) <textarea> | 7) <select> | 8) <main>

**Day 3:** 1) . | 2) # | 3) color | 4) background-color | 5) padding | 6) margin | 7) border-radius | 8) <link rel="stylesheet" href="...">>

**Day 4:** 1) display: flex | 2) justify-content | 3) align-items | 4) flex-wrap: wrap | 5) gap | 6) position: sticky | 7) nearest positioned ancestor | 8) grow shrink basis

**Day 5:** 1) @media | 2) max-width: 100% | 3) rem | 4) min-width | 5) :root | 6) display: none | 7) margin: 0 auto | 8) font-weight

**Day 6:** 1) .container | 2) .row | 3) Half width on md and up | 4) Gutter gap | 5) Padding | 6) text-muted | 7) btn btn-primary | 8) Utilities and components

**Day 7:** 1) navbar | 2) navbar-toggler | 3) collapse navbar-collapse | 4) form-control | 5) rounded / rounded-3 | 6) shadow / shadow-sm | 7) text-center | 8) my-\* or py-\*

**Day 8:** 1) let | 2) const | 3) querySelector | 4) classList.add | 5) click | 6) console.log | 7) textContent | 8) =>

**Day 9:** 1) event.preventDefault() | 2) String | 3) JSON.stringify | 4) JSON.parse | 5) getItem | 6) setItem | 7) push | 8) splice

**Day 10:** 1) <footer> | 2) . | 3) justify-content | 4) 12 | 5) getElementById | 6) preventDefault | 7) max-width: 100% | 8) JSON.stringify