30-Day Front-End Development Curriculum for Interns

# Week 1: HTML Foundations

## Day 1: Introduction to Web Development

Overview of front-end and back-end.  
Tools setup: VS Code, Git, and Chrome DevTools.  
Basic structure of an HTML document (doctype, head, body).  
Introduction to tags: <p>, <a>, <div>, <span>, <h1> to <h6>, etc.  
Homework: Create a simple HTML page with headings and paragraphs.

## Day 2: Forms and Input Elements

Forms (<form>, <input>, <textarea>, <select>, <button>).  
Form validation and input types.  
Action and method in forms.  
Homework: Create a form to collect user registration details.

## Day 3: Links and Lists

Anchor tags (<a>) and links (internal and external).  
Ordered and unordered lists (<ul>, <ol>, <li>).  
Homework: Build a navigation menu using lists and links.

## Day 4: Images and Media

Embedding images with the <img> tag.  
Adding videos (<video>), audio (<audio>), and iframes (<iframe>).  
Homework: Create a multimedia webpage that includes images and videos.

## Day 5: Semantic HTML and Accessibility

Understanding semantic HTML: <header>, <footer>, <section>, <article>, <nav>, etc.  
Importance of accessibility in web development.  
Homework: Build a blog page using semantic HTML.

## Day 6: Project Day

Project: Create a portfolio homepage using only HTML (no CSS or JavaScript yet).

## Day 7: Review and Q&A

Recap of HTML topics.  
Open Q&A session.  
Code review and feedback on the portfolio project.

# Week 2: CSS Fundamentals

## Day 8: Introduction to CSS

Inline, internal, and external CSS.  
Basic selectors (element, class, id).  
Adding CSS to the portfolio project.  
Homework: Style your portfolio homepage.

## Day 9: CSS Box Model

Understanding the box model (content, padding, border, margin).  
Display types: block, inline, inline-block.  
Homework: Style a container using the box model.

## Day 10: Flexbox Layout

Introduction to Flexbox and its properties (justify-content, align-items, flex-direction).  
Building responsive layouts with Flexbox.  
Homework: Create a responsive card layout using Flexbox.

## Day 11: CSS Grid

Introduction to CSS Grid and its properties (grid-template-columns, grid-template-rows, gap, etc.).  
Homework: Build a basic webpage layout using CSS Grid.

## Day 12: Colors, Backgrounds, and Gradients

Applying colors using color, background-color, and gradients.  
Using images as backgrounds.  
Homework: Add background images and gradients to your portfolio.

## Day 13: Fonts and Typography

Working with web-safe fonts and Google Fonts.  
Font properties: font-family, font-size, line-height, text-align, etc.  
Homework: Style text and typography in your portfolio.

## Day 14: CSS Transitions and Animations

Adding simple transitions and hover effects.  
Introduction to CSS animations.  
Homework: Create button animations and hover effects.

# Week 3: JavaScript Basics

## Day 15: Introduction to JavaScript

Basic syntax: variables, data types, and operators.  
Using console.log for debugging.  
Homework: Create a simple script that performs basic arithmetic operations.

## Day 16: Functions and Loops

Declaring functions and using loops (for, while).  
Scope and hoisting.  
Homework: Write a script to generate a list of numbers using loops.

## Day 17: DOM Manipulation

Selecting and modifying DOM elements using JavaScript (getElementById, querySelector, etc.).  
Homework: Add interactivity to your portfolio by updating the DOM.

## Day 18: Events and Event Listeners

Event handling: click, hover, submit, etc.  
Adding event listeners in JavaScript.  
Homework: Create an interactive form with validation using events.

## Day 19: Arrays and Objects

Introduction to arrays and objects.  
Basic array methods (push, pop, map, filter).  
Homework: Build a small array manipulation project (e.g., to-do list).

## Day 20: JavaScript Project Day

Project: Build a basic to-do list application using HTML, CSS, and JavaScript.

## Day 21: Review and Q&A

Recap of JavaScript topics.  
Open Q&A session.  
Code review and feedback on the to-do list project.

# Week 4: React.js Introduction

## Day 22: Introduction to React.js

Overview of React and single-page applications (SPAs).  
Setting up a React project using Create React App (CRA).  
JSX and components.  
Homework: Create a simple React component.

## Day 23: Props and State in React

Understanding props and state in React components.  
Building dynamic components using props.  
Homework: Create a component that accepts props and displays a list.

## Day 24: Event Handling and Forms in React

Handling events in React.  
Creating and managing forms in React.  
Homework: Build a form component that manages state.

## Day 25: Component Lifecycle and Hooks

Introduction to React component lifecycle.  
Understanding hooks (useState, useEffect).  
Homework: Implement hooks in your React project.

## Day 26: Routing in React

Introduction to React Router for navigation.  
Setting up routes and creating multiple pages in React.  
Homework: Add routing to your project.

## Day 27: API Integration in React

Fetching data from APIs using fetch or Axios.  
Displaying API data in React components.  
Homework: Build a weather app that fetches data from an API.

## Day 28: React Project Day

Project: Build a small e-commerce product page or a blog using React, with routes and state management.

## Day 29: Review and Code Walkthrough

Review React concepts.  
Q&A session.  
Code review and feedback on the React project.

# Week 5: Final Projects and Wrap-up

## Day 30: Final Project Presentation and Feedback

Interns will present their final projects (portfolio or React app).  
Code reviews, feedback, and improvements.  
Discussion on next steps: learning paths, advanced topics (e.g., Redux, TypeScript).