Week 1: JavaScript Fundamentals and Core Concepts

Day 1: JavaScript Basics

- **Learn:**
  - JavaScript syntax: variables (`let`, `const`, `var`), data types (`string`, `number`, `boolean`, `undefined`, `null`), and operators.
- **Practice:**
  - Write a program to convert Celsius to Fahrenheit.
  - Create a simple calculator (add, subtract, multiply, divide).
- **Resources:**
  - [MDN JavaScript Basics]()

Day 2: Control Flow

- **Learn:**
  - Conditionals (`if`, `else if`, `else`, `switch`).
  - Loops (`for`, `while`, `for...of`).
- **Practice:**
  - Write a program that checks if a number is even or odd.
  - Create a loop that prints all numbers divisible by 3 from 1 to 50.

Day 3: Functions

- **Learn:**
  - Function declarations, expressions, arrow functions.
  - Parameters and return values.
- **Practice:**
  - Write a function that calculates the factorial of a number.
  - Create a function to reverse a string.
- **Resources:**
  - [JavaScript Functions]()

Day 4: Arrays and Objects

- **Learn:**
  - Arrays: methods like `.push()`, `.pop()`, `.map()`, `.filter()`.
  - Objects: creating, accessing, and modifying properties.

- **Practice:**
  - Create an array of numbers and calculate their sum.
  - Write an object representing a student (name, age, courses) and a function to print their details.

Day 5: The DOM

- **Learn:**
  - Select and manipulate HTML elements using `document.querySelector`, `getElementById`.
  - Add event listeners.
- **Practice:**
  - Create a button that, when clicked, changes the text of an HTML element.
  - Create a simple to-do list where items can be added dynamically.
- **Resources:**
  - MDN DOM Guide

Day 6: Asynchronous JavaScript

- **Learn:**
  - Callbacks, Promises, and `async/await`.
  - Fetch data from an API (e.g., JSONPlaceholder).
- **Practice:**
  - Fetch and display a list of users from an API.
  - Write a function that waits for 2 seconds before printing a message.
- **Resources:**
  - JavaScript Promises

Day 7: HTML and CSS Refresher

- **Learn:**
  - HTML structure: semantic tags (`header`, `footer`, `article`).
  - CSS basics: selectors, box model, flexbox.
- **Practice:**
  - Create a simple portfolio webpage with sections for About, Skills, and Contact.
- **Resources:**
  - HTML & CSS Crash Course

Week 2: Advanced JavaScript and Front-End Development

Day 8: ES6+ Features

- **Learn:**
    - Destructuring, template literals, `let`/`const`, default parameters, spread/rest operators.
- **Practice:**
    - Create a function that accepts any number of arguments and returns their sum.
    - Use template literals to generate HTML dynamically.
- **Resources:**
    - MDN ES6 Guide

Day 9: Event Handling and Forms

- **Learn:**
    - Handle form submissions.
    - Validate user input using JavaScript.
- **Practice:**
    - Create a contact form that validates fields (e.g., email format, required fields).
- **Resources:**
    - Handling Events

Day 10-11: React Basics

- **Learn:**
    - React setup (via CDN or Create React App).
    - JSX syntax, components, props, and state.
- **Practice:**
    - Create a React app that displays a list of items fetched from an API.
- **Resources:**
    - React Docs

Day 12: Git and Version Control

- **Learn:**
    - Basic Git commands: `clone`, `commit`, `push`, `pull`.
- **Practice:**
    - Initialize a Git repository for your project.
    - Push your code to GitHub.

- **Resources:**
  - GitHub Guide

Day 13: Responsive Design and Accessibility

- **Learn:**
  - CSS media queries for responsiveness.
  - ARIA roles for accessibility.
- **Practice:**
  - Make your portfolio page responsive for mobile and tablet devices.
- **Resources:**
  - CSS Tricks: Media Queries

Day 14: Build and Showcase a Project

- **Project Ideas:**
  - **Interactive To-Do List**: A web app where users can add, edit, and delete tasks.
  - **Weather App**: Use an API to fetch and display weather data for a given city.
- **Resources:**
  - Netlify or Vercel for deployment.