1. Personal Portfolio Website

Concepts Covered:

- **HTML**:
  - Semantic tags: `<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`.
  - Forms: Contact form with fields (name, email, message).
  - Tables: Skillset summary in tabular format.
  - Media embedding: Adding profile images or background videos.
- **CSS**:
  - Layouts: Flexbox or Grid for structuring the webpage.
  - Styling: Font styles, colors, hover effects.
  - Responsive design: Media queries for mobile and desktop.
  - Animations: Transition effects for buttons or hover states.
- **JavaScript**:
  - Variables and functions: Create reusable code for form validation.
  - DOM manipulation: Dynamically update skills or project sections.
  - Event listeners: Interactive navbar or form submission.
  - Promises/async-await: Simulate a fake form submission.

Features:

1. **Home Section**:
   - A brief introduction and an image.
2. **Skills Section**:
   - Display skills using a table.
3. **Projects Section**:
   - Cards showing project details.
4. **Contact Form**:
   - Fields for name, email, and message.
   - Add basic validation using JavaScript.
5. **Responsive Design**:
   - Ensure it looks good on mobile and desktop.

2. To-Do List Application

Concepts Covered:

- **HTML**:
  - Semantic structure: `<section>`, `<ul>`, `<button>`, `<input>`.
  - Forms: Input for adding tasks.
- **CSS**:
  - Flexbox or Grid for layout.
  - Media queries to make the app responsive.
  - Animations: Smooth addition/removal of tasks.
- **JavaScript**:
  - Variables: Store tasks in arrays or objects.
  - Functions: Add, edit, delete, and mark tasks as complete.
  - DOM manipulation: Render tasks dynamically in the list.
  - Event listeners: Button clicks for adding or removing tasks.
  - Promises/async-await: Mock saving tasks to a server.

Features:

1. **Add Task**:
   - Input field to type a task and a button to add it.
   - Tasks appear in a dynamic list.
2. **Mark Complete**:
   - Strike-through completed tasks.
3. **Delete Task**:
   - Button to remove a task.
4. **Save and Load Tasks**:
   - Use `localStorage` for persistence or mock an API call.
5. **Responsive Design**:
   - Adjust layout for mobile and desktop.