**Beginner Level**

At the beginner level, you will learn the basics of HTML, including how to create and structure HTML documents, add headings, paragraphs, links, images, and lists.

1. Introduction to HTML
   1. Basic syntax and structure of an HTML document
   2. The role of HTML in web development
   3. Setting up a basic HTML page
2. HTML Elements
   1. Headings and paragraphs
   2. Links and anchors
   3. Images and attributes
   4. Lists and tables
3. HTML Forms
   1. Form elements and attributes
   2. Input types (text, radio buttons, checkboxes, etc.)
   3. Labeling and grouping form elements
4. Semantic HTML
   1. The importance of semantic HTML
   2. Semantic tags (header, footer, section, article, etc.)
   3. Using ARIA attributes for accessibility
5. Project: Building a Basic Website
   1. Applying the concepts learned in the previous lessons to create a simple, static website

**Intermediate Level**

At the intermediate level, you will build upon your knowledge of HTML and learn how to create more complex web pages with advanced layout and styling techniques.

1. CSS Basics
   1. Introduction to CSS and how it relates to HTML
   2. Selectors and declaration blocks
   3. Basic CSS properties (color, font, background, etc.)
2. Box Model and Layout
   1. Understanding the box model
   2. Positioning and floating elements
   3. Creating multi-column layouts
3. Responsive Design
   1. Media queries and responsive design principles
   2. Mobile-first design approach
   3. Flexible grids and fluid layouts
4. HTML5 APIs
   1. Geolocation API
   2. Drag and drop API
   3. Local Storage API
5. Project: Creating a Responsive Website
   1. Applying the concepts learned in the previous lessons to create a responsive website that adapts to different screen sizes and devices.

**Advanced Level**

At the advanced level, you will learn more advanced HTML and CSS concepts, such as CSS preprocessors, advanced layout techniques, and web accessibility.

1. CSS Preprocessors
   1. Introduction to CSS preprocessors (Sass, Less, etc.)
   2. Variables, mixins, and nesting
   3. Using preprocessors to streamline and simplify CSS code
2. Advanced Layout Techniques
   1. Flexbox layout
   2. CSS Grid layout
   3. Combining flexbox and grid for complex layouts
3. Web Accessibility
   1. Accessibility principles and guidelines
   2. Semantic HTML for accessibility
   3. ARIA attributes for accessibility
4. HTML5 and Beyond
   1. Web components
   2. Web animations and SVG
   3. WebSocket API for real-time communication
5. Project: Creating a Complex Web Application
   1. Applying the concepts learned in the previous lessons to create a complex web application with a responsive and accessible design, using advanced layout techniques and HTML5 APIs.