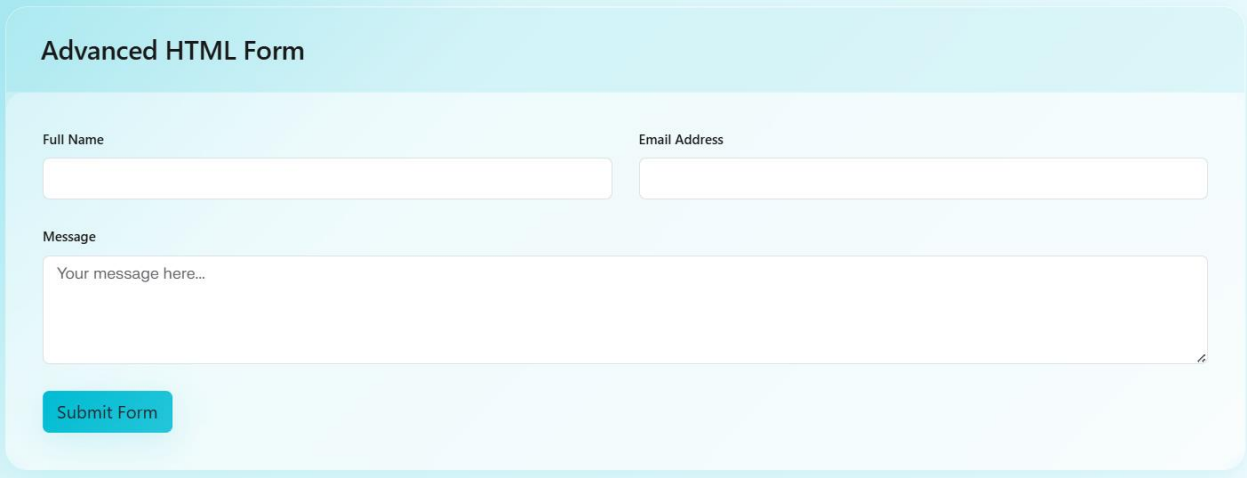
HTML (HyperText Markup Language)

HTML is the standard language for creating the **structure and content** of web pages. . It defines various content types, including headings, paragraphs, images, and links, using a system of "tags." Semantic tags are a fundamental component of modern HTML (HTML5). These tags, like , , or , define the meaning of the content they enclose. Semantic tags are essential for search engine optimization (SEO) because they help search engines better index the content and for accessibility because they make the page layout easier for screen readers to understand.

How the Demos Work: A Detailed Explanation

1. **Semantic Article Demo**: This demo shows a perfectly structured, semantic HTML article to illustrate best practices.
    - **Structure**: The index. html file incorporates certain HTML5 tags to make each section of the article meaningful. As an example, the whole article is enclosed by an <article> tag, the title and date is enclosed in a tag of <header> tag, the main body is enclosed in a tag of <main> tag and the name and details of the author is enclosed in a tag of <footer> tag. Such an unambiguous structure is very advantageous to the assistive technologies, as well as search engines. The code in this instance is then displayed on the page using pre and code tags, which have the specific purpose to display some pre-formatted code.

2. **Advanced HTML Form with Validation**: This demonstrates a modern, interactive form that provides real-time feedback to the user.



- **Structure**: Form in index.html is designed with usability and accessibility in consideration. A label tag is provided on each input field; this is necessary with screen readers. Such attributes as required make sure that the user completes the necessary fields and Autocomplete also makes sure that the user wastes time as the browser suggests the most common data such as the name of the user or his or her email. Each input field is followed by a special div with the purpose to store the validation messages.
- **Real-Time Validation**: The smarts of the form are in the app.js file. It has a function established to monitor the input fields. A validation function is executed when a user clicks out of a field (an occurrence known as blur). This is a validation that will ensure that the field is not empty or the email address is in an acceptable format. Depending on the outcome, it dynamically includes to the input field a success or error class. Rules in the style.css change the border colour of the input to green during success or red during error and are used to provide instant visual feedback. The JavaScript also changes the text in the message div to inform the user what is wrong (e.g. this field is required) or it is good (Looks good!).
- **Submission Handling** The submission of the form is done by another JavaScript function in the file app.js. This works when the submit button is clicked such that the page is not reloaded (the default behavior of browsers when a form is clicked).

It reruns the validation of all fields then. When it is all good, it mimics transferring the data by displaying a loading spinner over the button during a few seconds. Once a simulated delay is over it will show the user success message and will clear the form preparing it to submit again.