AeroAspire-SDE Intern

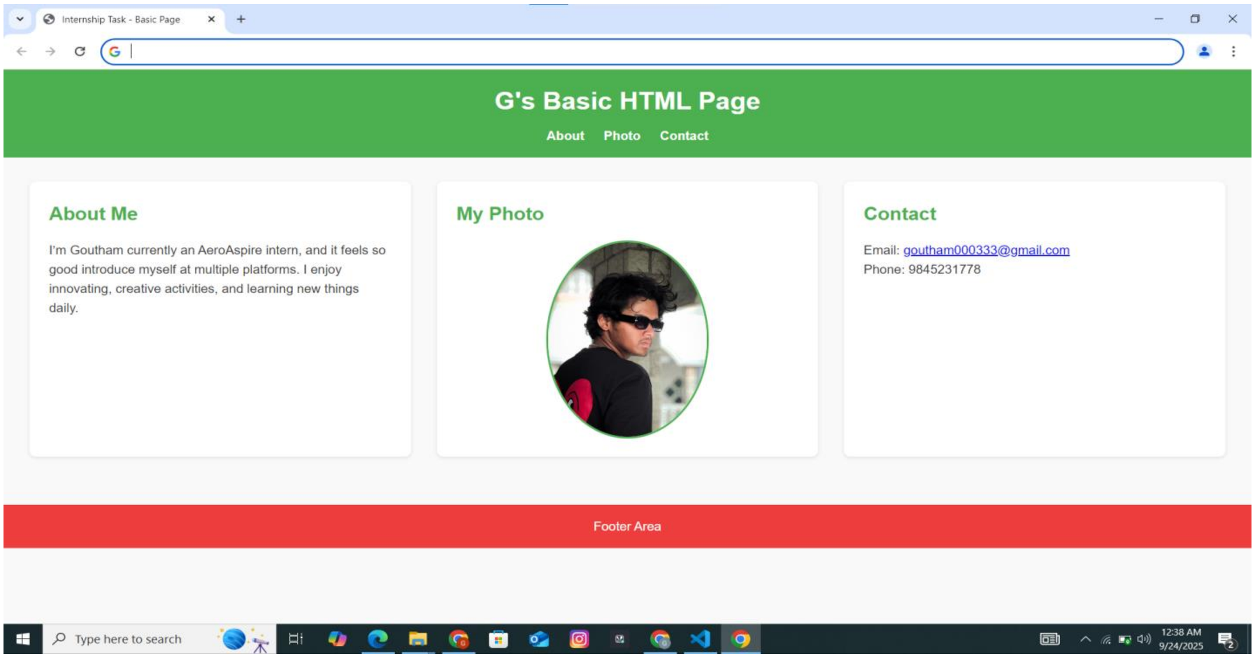
Goutham V

Week 1 – Day2(23rd September)

Task:

Build basic HTML page: About / Photo / Contact sections, Style the sections; header/nav/footer; layout using Flexbox or Grid.

Below is the webpage created,



G's Basic HTML Page

About Photo Contact

About Me

I'm Goutham currently an AeroAspire intern, and it feels so good introduce myself at multiple platforms. I enjoy innovating, creative activities, and learning new things daily.

My Photo

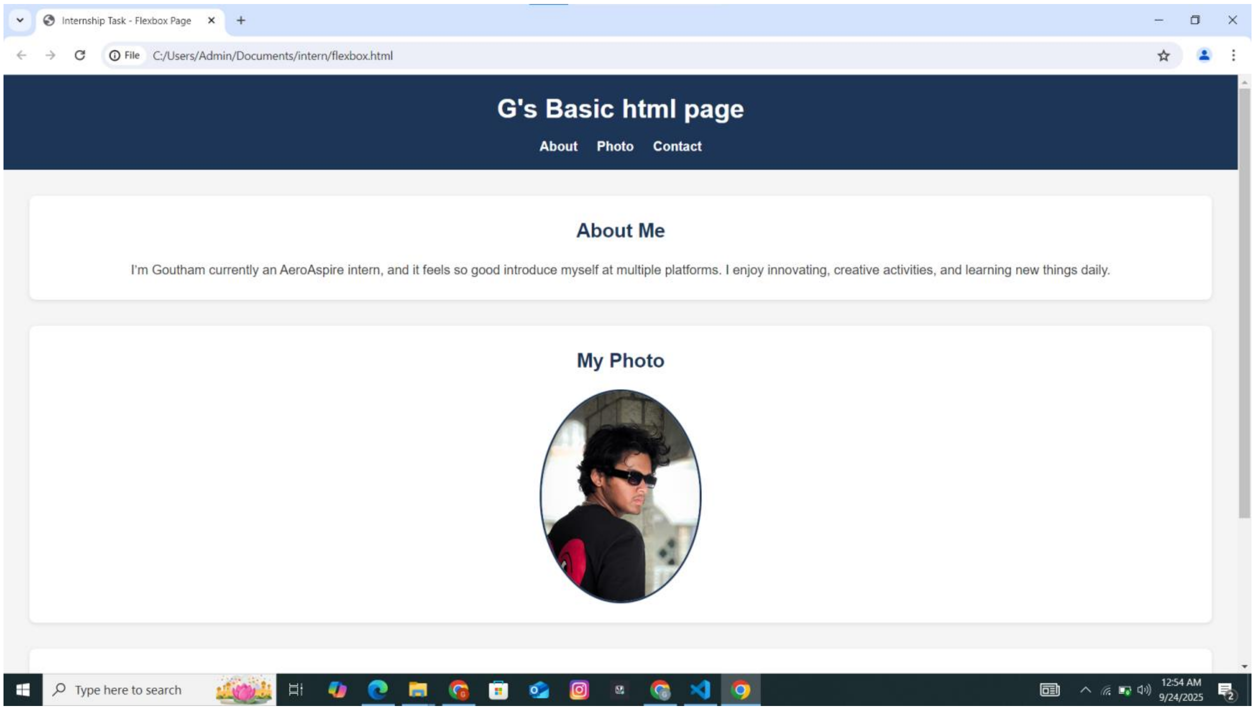
Contact

Email: goutham000333@gmail.com
Phone: 9845231778

Footer Area

Webpage layout using Grid

This page uses a grid layout to organize content into three main sections: "About Me," "My Photo," and "Contact." The sections are neatly arranged side by side, giving the page a clean and structured look. The header is handled with a green background and includes navigation links. There’s also a red footer area at the bottom for a finished touch.


Internship Task - Flexbox Page
File C:/Users/Admin/Documents/intern/flexbox.html
Type here to search
12:54 AM
9/24/2025

G's Basic html page

About Photo Contact

About Me

I'm Goutham currently an AeroAspire intern, and it feels so good introduce myself at multiple platforms. I enjoy innovating, creative activities, and learning new things daily.

My Photo

Webpage layout using FlexBox

This page uses Flexbox to arrange its content in a more vertically centered and spaced-out style. The "About Me" section stretches across the page at the top, followed by "My Photo" centered beneath it. The style is more open and balanced, with a header and simplified layout that adjusts well to different screen sizes.

Reflection,

1.What is <section> vs <div>?

-A `<section>` is a semantic HTML tag used for grouping related content together, like a topic or chapter on your web page. For example, you might use `<section>` for an article or a blog post.

-A `<div>` is a generic container for flow content and doesn't add any specific meaning; it just helps organize things visually without saying what the content is about.

2. Why semantics matter?

-Semantic tags like `<section>`, `<header>`, and `<footer>` tell browsers and developers what each part of your page means. This makes code easier to read and helps search engines understand your site better. Semantics also improve accessibility for people using screen readers because tools can identify sections more precisely.

3. What is the flow from writing HTML → rendering by browser?

- Start with writing structure and content using HTML tags.
- The browser parses the HTML and creates a Document Object Model (DOM).
- CSS is loaded, parsed, and combined with the DOM to compute styles.
- The browser lays out elements as per the DOM and CSS rules, then paints everything onto the screen.

4. How does semantic HTML improve accessibility and SEO?

-Semantic HTML gives browsers and assistive technologies clues about what each part of the webpage does, making it easier for people with disabilities to navigate and understand the pages. It also boosts SEO because search engines look for these tags to figure out your site's structure and show better results.

5. Describe how the browser parses HTML + CSS to render layout.

- Browser reads the HTML and builds the DOM tree.
- It also reads CSS and creates the CSSOM.
- Both trees are used to generate the render tree, where visual structure is calculated.
- Browser figures out box sizes, positions, colors, then displays the layout on the page.

6. How Flexbox handles alignment when container resizes?

- Flexbox lets items in a container expand, shrink or wrap automatically depending on available space.
- Properties like `justify-content` and `align-items` control the alignment and spacing of children, even if the container size changes.

7. Describe the CSS box model and how margin/padding/border/content interact.

- Content: The actual text or image.
- Padding: Space around the content, inside the border.
- Border: Surrounds the padding (if any) and content.
- Margin: Space outside the border, separating elements from each other.
- Together, these determine the total size and spacing of an element on the page.

8. What is the flow of CSS specificity: inline styles, IDs, classes, element selectors?

a) Inline styles (added directly in an element's `style` attribute) have the highest specificity.
b) ID selectors (like `#main`) come next.
c) Class selectors (like `.item`), attributes, and pseudo-classes follow.
d) Element (type) selectors have the lowest specificity.
e) If multiple rules apply, the rule with higher specificity will be used.

9. How would you approach making a layout responsive?

-To build a responsive layout, start with flexible units like percentages, `em`, and `rem` instead of fixed pixels. Use media queries to adjust styles for different device sizes, and take advantage of layout tools like Flexbox and CSS Grid for fluid arrangements. Testing across different devices and ensuring usable touch targets is also key for real-world, mobile-friendly design.