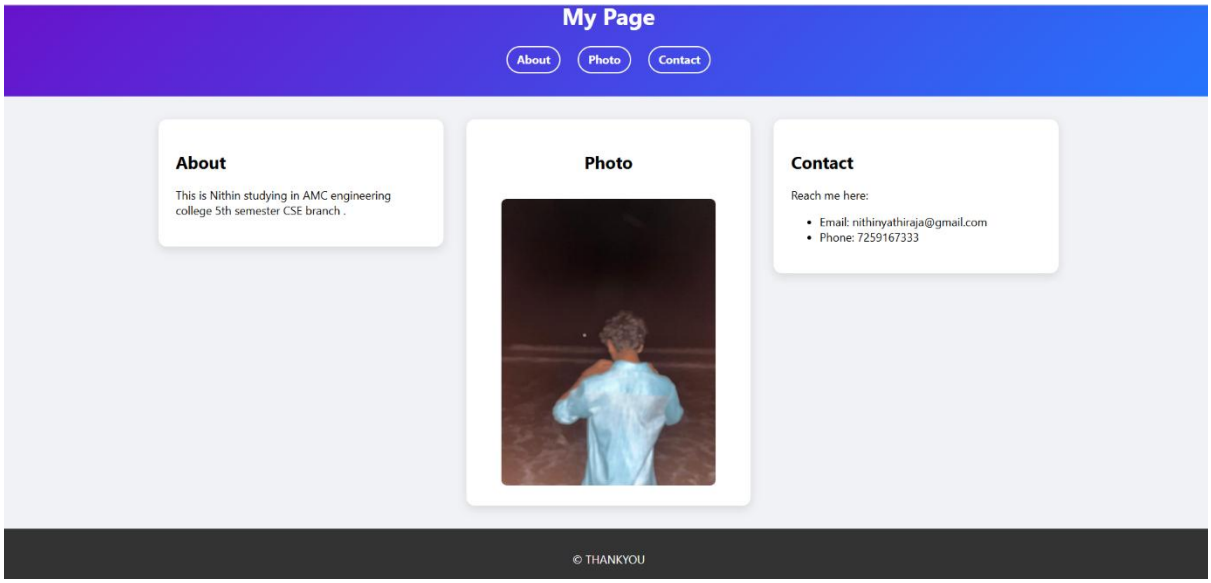
AEROASPIRE INTERN

NITHIN K Y

DAY 1:- Task :- Build basic HTML page: About / Photo / Contact sections, Style the sections; header/nav/footer;layout using flexbox or grid



Questions :-

- **1. What is <section> vs ?**

  The <section> tag defines a section in a document.

  It groups content that shares a common theme.

  Each section is typically introduced with a heading

  The tag defines a division or a section in an HTML document.

The tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the tag.

The difference between both are:-

 is a non-semantic container, used mainly for styling, layout, or JavaScript it doesn't add meaning to the content.

<section> is a semantic container, used to group thematically related content, usually with a heading, and is more SEO and accessibility friendly.

- # 2. Why semantics matter?

Semantic elements = elements with a meaning.

Provide the user a good way to navigate and interact with your site. Make your HTML semantic. Semantics is about using the correct element in HTML.

Semantic HTML matters because it makes webpages easier for search engines, screen readers, and developers to understand. It improves SEO, accessibility, and code readability, beyond just visual layout.

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

- HTML is written – The developer creates an HTML file, optionally linking CSS and JavaScript.
- HTML file is requested and downloaded – The browser fetches the HTML file from the server via HTTP/HTTPS.
- HTML parsing begins – The browser reads the HTML text and parses it into a DOM (Document Object Model) tree.
- CSS is fetched and parsed – CSS from <style> tags or external stylesheets is parsed into a CSSOM (CSS Object Model).
- JavaScript is parsed and executed – Scripts may modify the DOM or CSSOM dynamically; blocking scripts pause HTML parsing unless marked async or defer.
- DOM and CSSOM are combined – The browser merges the DOM and CSSOM to create the Render Tree, representing visible content.
- Layout (Reflow) – The browser calculates the position and size of each element in the render tree based on CSS box model and layout rules.
- Painting – The visual representation of each element (text, colors, images, borders, etc.) is drawn into layers.

- Compositing – All painted layers are assembled together, handling stacking (z-index), transforms, opacity, etc.
- Pixels are rendered to the screen – The final composed image is displayed in the browser window.

4. How does semantic HTML improve accessibility and SEO?

Semantic HTML improves accessibility by helping screen readers understand the structure of a page, so people with disabilities can navigate it easily. It improves SEO because search engines can identify important sections like headings, articles, or navigation, making content more likely to rank higher.

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

When you open a webpage, the browser reads (parses) the HTML to build a structure called the DOM (tree of elements). Then it parses the CSS to figure out how each element should look (colors, sizes, fonts). After that, it combines HTML + CSS into a render tree. Finally, the browser lays out and paints everything on the screen for you to see.

6. How Flexbox handles alignment when container resizes?

Flexbox is smart because it adjusts items automatically when the container size changes.
If the container becomes smaller, items shrink or wrap to fit.
If the container grows, items expand or spread out with space between them.
justify-content controls alignment horizontally (left, right, center, space).
align-items controls alignment vertically (top, center, bottom).
This makes layouts flexible and responsive without extra code.

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

The CSS box model explains how every element on a webpage is a box with four parts:

Content – The text or image inside the box.
Padding – The space between the content and the border (like inner spacing).
Border – The line that goes around the padding and content.
Margin – The space outside the border that separates the box from other elements.

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

- Inline styles (style="...") → highest priority, always wins over anything in CSS files.
- IDs (#header) → stronger than classes, attributes, or elements.
- Classes, attributes, pseudo-classes (.menu, [type="text"], :hover) → medium priority.
- Element selectors (div, p, h1) → lowest priority, used if nothing else applies.

- # 9.How would you approach making a layout responsive?

- Use flexible units like %, vw, vh, or em instead of fixed pixels.
- Use CSS Flexbox or Grid to let elements adjust automatically when the screen changes.
- Add media queries to apply different styles for different screen sizes
- Make images and videos flexible with max-width: 100%; height: auto;.
- Test on different screen sizes (desktop, tablet, mobile) and adjust as needed.