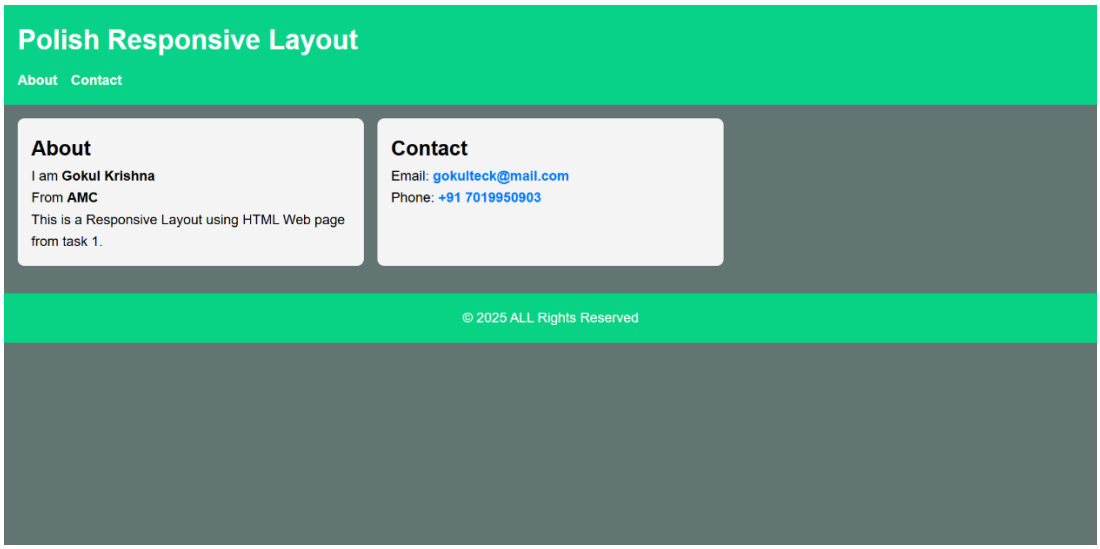
AeroAspire-SDE Intern
Gokul Krishna S

Week 1 – Day 4 (September 26)

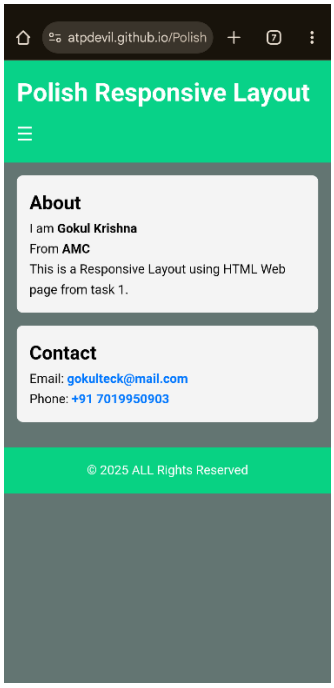
Task/Assignment :

Polish responsive layout;
deploy on GitHub Pages;
write README.

Desktop Layout :

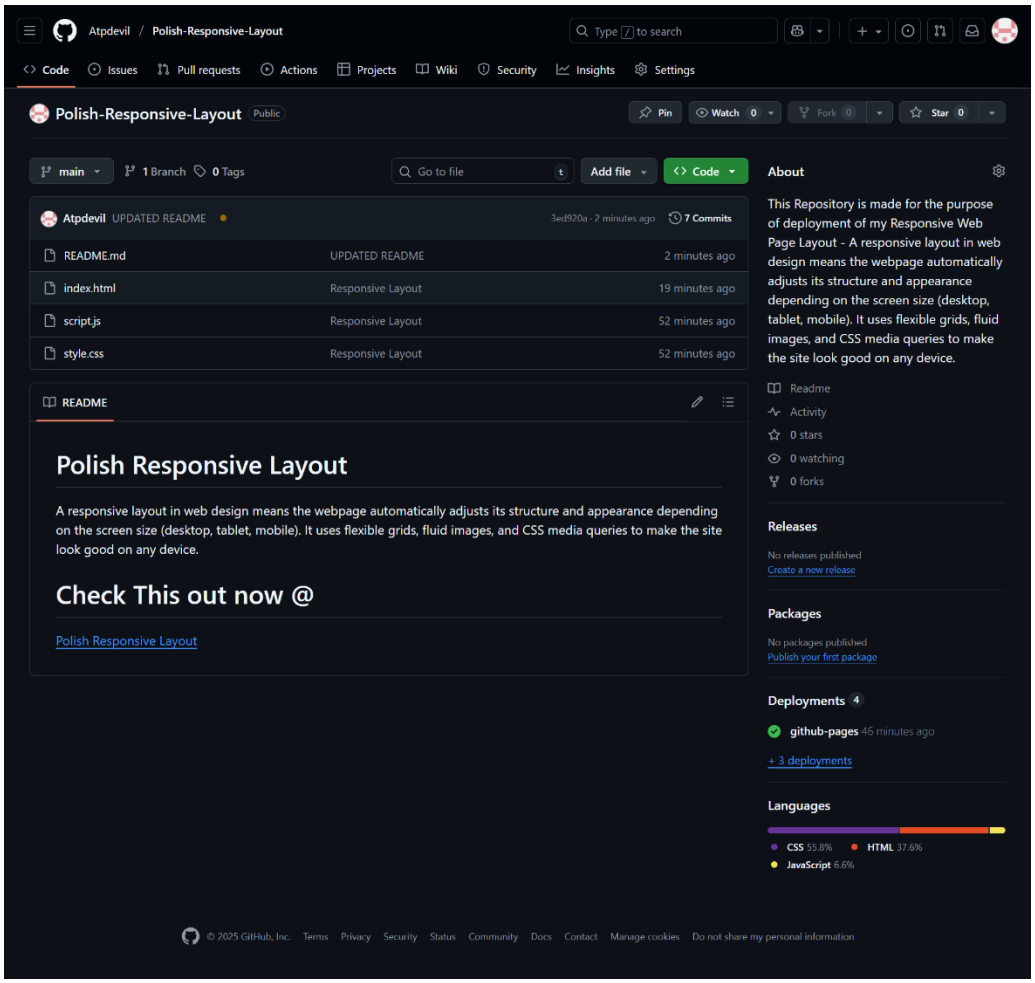


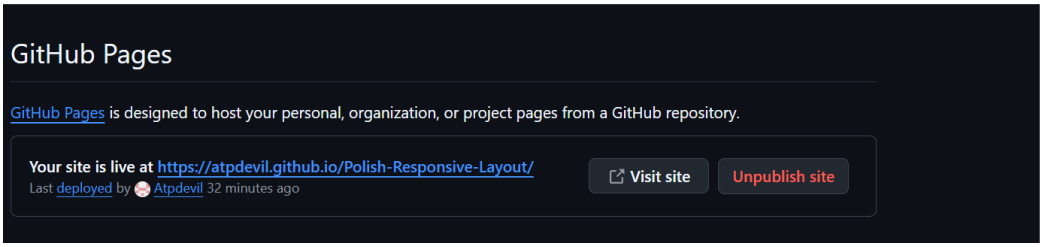
Mobile Layout :



➢ Steps Involved :

- I started with my previous Day 1 task HTML page and enhanced it into a polished responsive layout**.**
- Improved the HTML structure by using semantic tags (header, main, section, footer) for better readability and accessibility.
- Designed the layout using CSS Grid for desktop and tablet views, ensuring content is well organized in multiple columns.
- Implemented responsive breakpoints (@media queries) for tablet and mobile devices:
  - 3-column grid on desktop.
  - 2-column grid on tablets.
  - Single-column stacked layout on mobile.
- Created **a** hamburger menu with JavaScript that toggles navigation links on mobile screens for better usability.
- Verified responsiveness across desktop, tablet, and mobile devices to confirm layout adaptability.
- The project is now live on GitHub Pages for public access.





Questions/Reflections :

1. What steps did you follow to deploy via GitHub Pages?

    1. **Created a GitHub Repository :**
        - Made a new repository on GitHub named Polish-Responsive-Layout
        - Added a professional description about my daily tasks and assignments
    2. **Pushed Project Files to GitHub :**
        - Initialized Git in my local project folder using git init.
        - Added files with git add . and committed them using git commit -m "Initial commit".
        - Connected to the GitHub repository using git remote add origin <repo-link>.
        - Pushed the code with git push -u origin main.
    3. **Enabled GitHub Pages :**
        - Went to the repository settings in GitHub.
        - Opened the **Pages** section under Code and Automation.
        - Selected the **main branch** as the source and set the folder to /root.
        - Saved the settings.
    4. **Deployed Website :**
        - GitHub automatically built and deployed the project.
        - The live link was generated in the GitHub Pages section, typically in the format: https://<username>.github.io/<repository-name>/
    5. **Verified the Deployment**
        - Opened the live link in desktop, tablet, and mobile browsers.

- Checked responsiveness, layout, and functionality (hamburger menu, contact section).

- Confirmed that updates are reflected when pushing new commits

2. What difficulties you faced in deployment?

- At first, I faced an error while pushing my code because my local branch and the remote branch didn't match (main vs master). I had to fix this by renaming the branch and syncing it with GitHub.
- Another difficulty was understanding GitHub Pages settings**.** Initially, I couldn't see my live site because I hadn't selected the correct branch under the Pages settings.
- Overall, these issues taught me how to handle branch conflicts, file path issues, and deployment settings, which made the process smoother.

3. How do responsive breakpoints work (CSS media queries)?

- Responsive breakpoints use CSS media queries (@media) to apply styles conditionally based on device width or other features.
- Typical breakpoints are set at common device widths like 600px, 768px (tablet), 992px (desktop), and 1200px+ (large screens).
- Example:

```
@media only screen and (max-width: 600px) {
        body {
            background-color: lightblue;
          }
}
```

4. Why are hover/focus/active states important for UX?

- Hover states (:**hover**) provide visual feedback when a user places a cursor over an element, enhancing discoverability and interactivity.
- Focus states (:**focus**) help users, especially those navigating via keyboard, know which element is currently selected, which is vital for accessibility.

- Active states (:**active**) show when an interactive element (like a button or link) is being activated, further reinforcing responsiveness.
- These states make interfaces more intuitive, guide user actions, and meet accessibility standards for predictable, usable web design.