MAD-PWD Experiment - 10

AIM: To study and implement deployment of Ecommerce PWA to GitHub Pages.

THEORY:
GitHub Pages is a web hosting service provided by GitHub, allowing users to publish static websites directly from their GitHub repositories. Here's a brief overview of GitHub Pages:

1. Hosting for Static Websites:
GitHub Pages is designed to host static websites, which consist of HTML, CSS, and JavaScript files. It does not support server-side scripting languages like PHP or databases like MySQL. This makes it ideal for hosting simple websites, documentation, project pages, or personal blogs.

2. Free Hosting:
One of the significant advantages of GitHub Pages is that it offers free hosting for public repositories. Users can host their websites for free without any additional cost, making it an attractive option for developers, open-source projects, and personal websites.

3. Automatic Deployment:
GitHub Pages automatically deploys websites whenever changes are pushed to the repository's gh-pages branch or the docs folder within the main branch. This streamlined deployment process eliminates the need for manual uploading or configuration, simplifying the publishing workflow.

4. Custom Domains:
GitHub Pages supports custom domains, allowing users to use their own domain names for hosted websites. By configuring DNS settings, users can point their domain to their GitHub Pages site, giving it a professional and personalized web address.

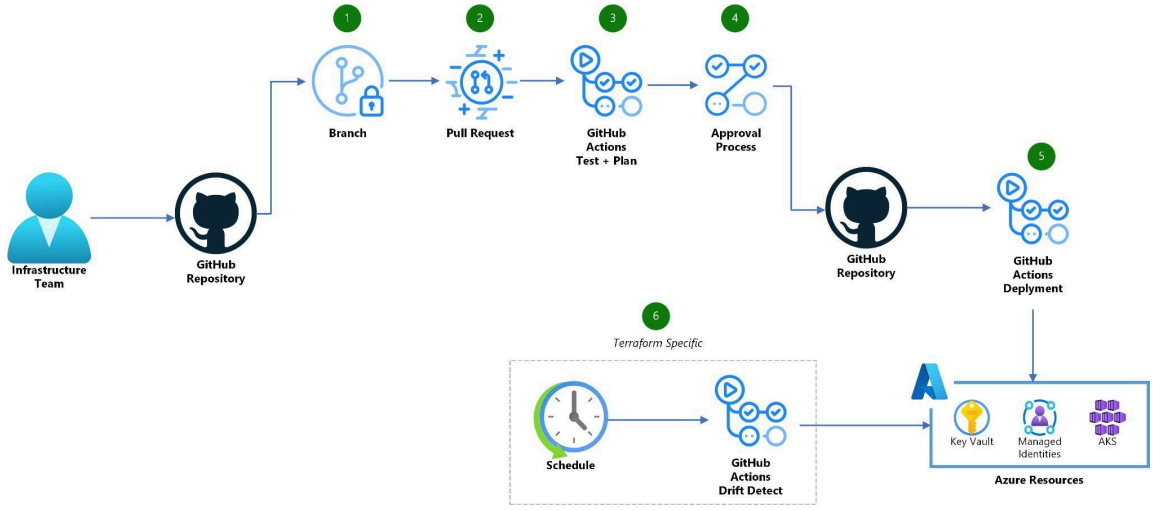
5. HTTPS Encryption:
GitHub Pages provides HTTPS encryption for all hosted websites by default. This ensures that data transmitted between the user's browser and the GitHub Pages server is encrypted, enhancing security and privacy for visitors accessing the website.

6. Built-in Jekyll Support:
GitHub Pages offers built-in support for Jekyll, a static site generator. Users can create and publish Jekyll-powered websites directly from their GitHub repositories. Jekyll simplifies

website development by providing templates, layouts, and other features for generating static content.

7. Project Pages:
GitHub Pages allows users to create Project Pages, which are dedicated websites for individual GitHub projects. Project Pages are hosted under a subdomain of username.github.io/projectname, providing a centralized platform for showcasing project documentation, demos, and information.

8. Community and Collaboration:
GitHub Pages fosters community and collaboration by providing a platform for sharing and showcasing projects with a global audience. Users can contribute to open-source projects, collaborate on documentation, or showcase their portfolios, leveraging the power of GitHub's collaborative ecosystem.



While GitHub Pages offers many benefits, it also has some limitations and drawbacks:

Limited Server-Side Functionality: GitHub Pages only supports static websites, so it lacks server-side functionality such as dynamic content generation, server-side scripting, and database interaction. This limits the types of web applications and functionality that can be hosted on GitHub Pages.

No Support for Non-Static Content: GitHub Pages does not support hosting non-static content such as databases, server-side applications, or serverless functions. As a result, complex web applications requiring dynamic content or backend processing cannot be hosted on GitHub Pages alone.

File Size and Bandwidth Limits: GitHub Pages imposes limits on file size and bandwidth usage, which may restrict the size and traffic of hosted websites. Large files or high traffic volumes may exceed these limits, resulting in performance issues or additional costs for premium hosting services.

Limited Control over Server Configuration: GitHub Pages offers limited control over server configuration and customization options compared to traditional web hosting services. Users have restricted access to server settings, .htaccess files, and other configuration files, limiting customization options for advanced users.

Dependency on GitHub Infrastructure: GitHub Pages is dependent on GitHub's infrastructure and availability. If GitHub experiences downtime or technical issues, hosted websites may become inaccessible or experience disruptions, impacting the reliability and availability of the hosted content.

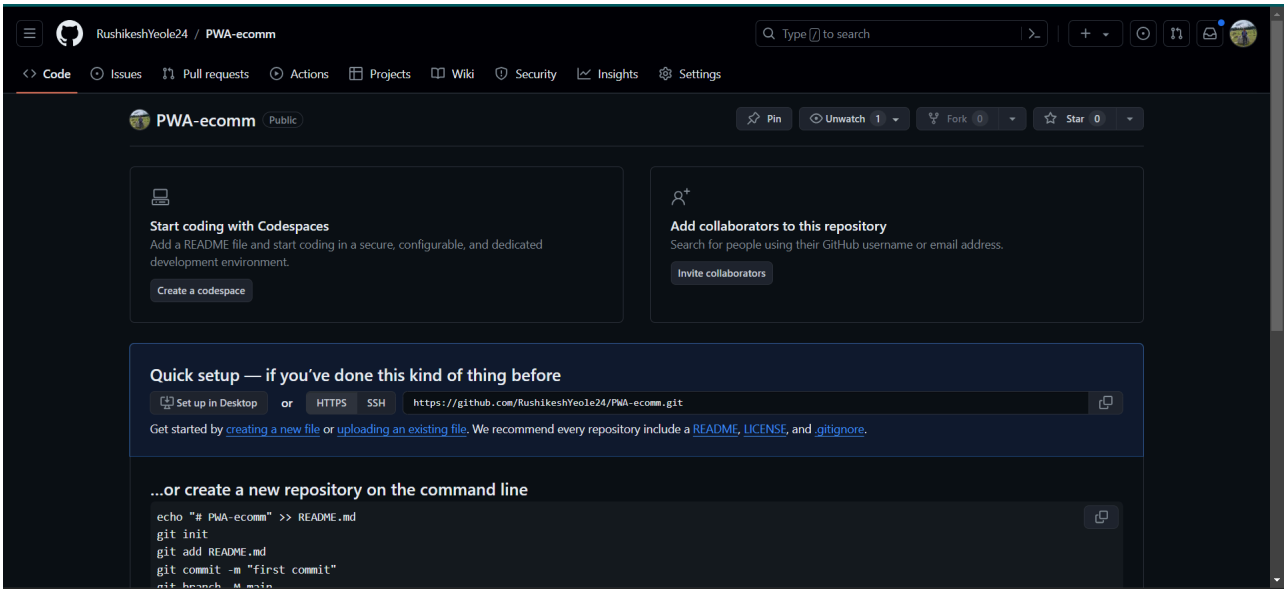
No Support for Server-Side Technologies: GitHub Pages does not support server-side technologies such as PHP, Ruby on Rails, or Node.js. This limits the flexibility and functionality of hosted websites, particularly for applications requiring server-side processing or dynamic content generation.

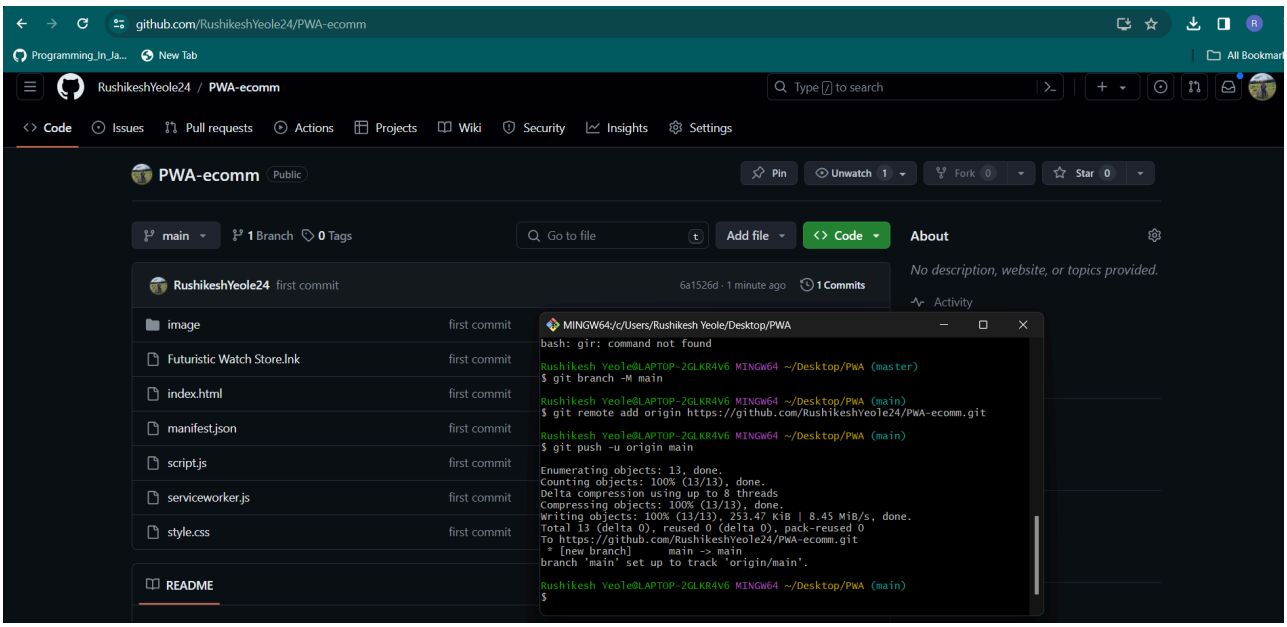
Limited Security Options: GitHub Pages provides basic security features such as HTTPS encryption and GitHub's security measures. However, users have limited control over server security settings and may not be able to implement advanced security measures or custom security configurations.

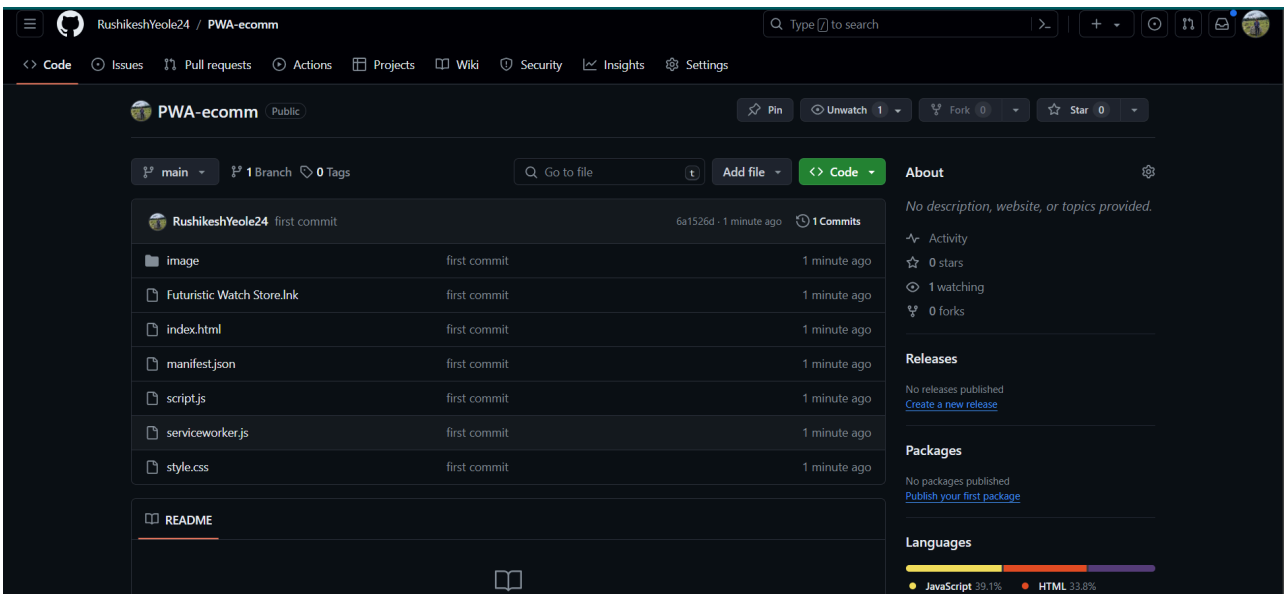
GitHub Repository Dependency: GitHub Pages hosting is closely tied to GitHub repositories. If a repository is deleted or becomes inaccessible, the associated GitHub Pages website may also become unavailable, potentially resulting in data loss or disruptions to website access.

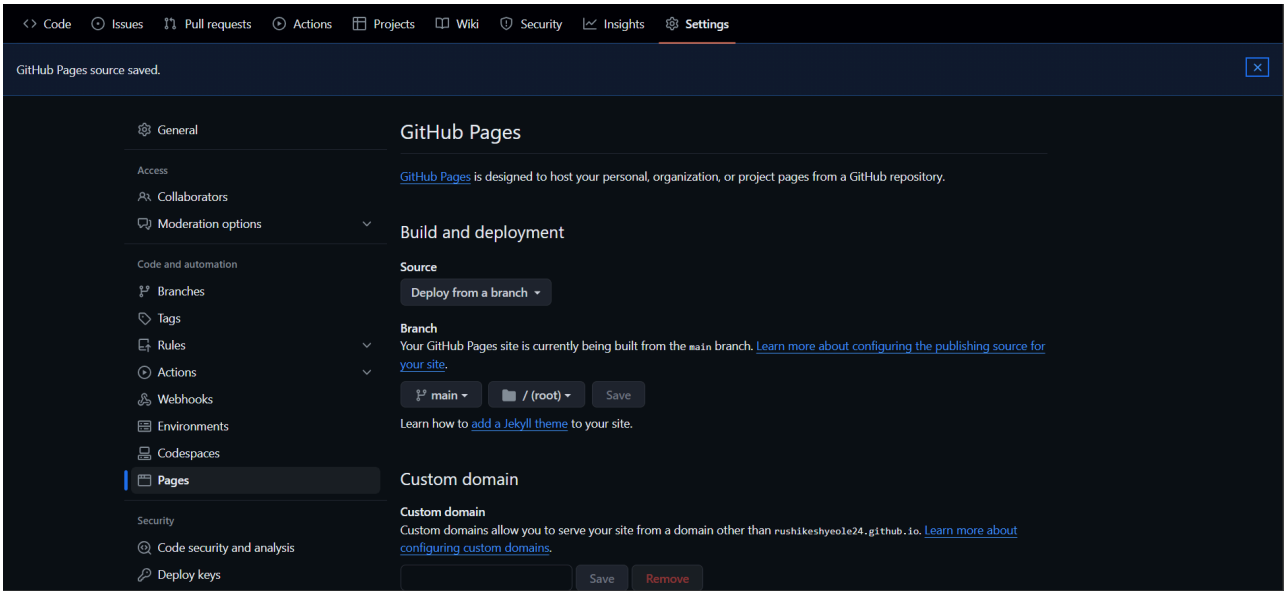
In summary, while GitHub Pages offers a convenient and cost-effective solution for hosting static websites, it may not be suitable for hosting complex web applications or websites requiring server-side functionality, advanced customization, or scalability beyond its limitations. Users should consider these drawbacks when evaluating GitHub Pages as a hosting solution for their projects.

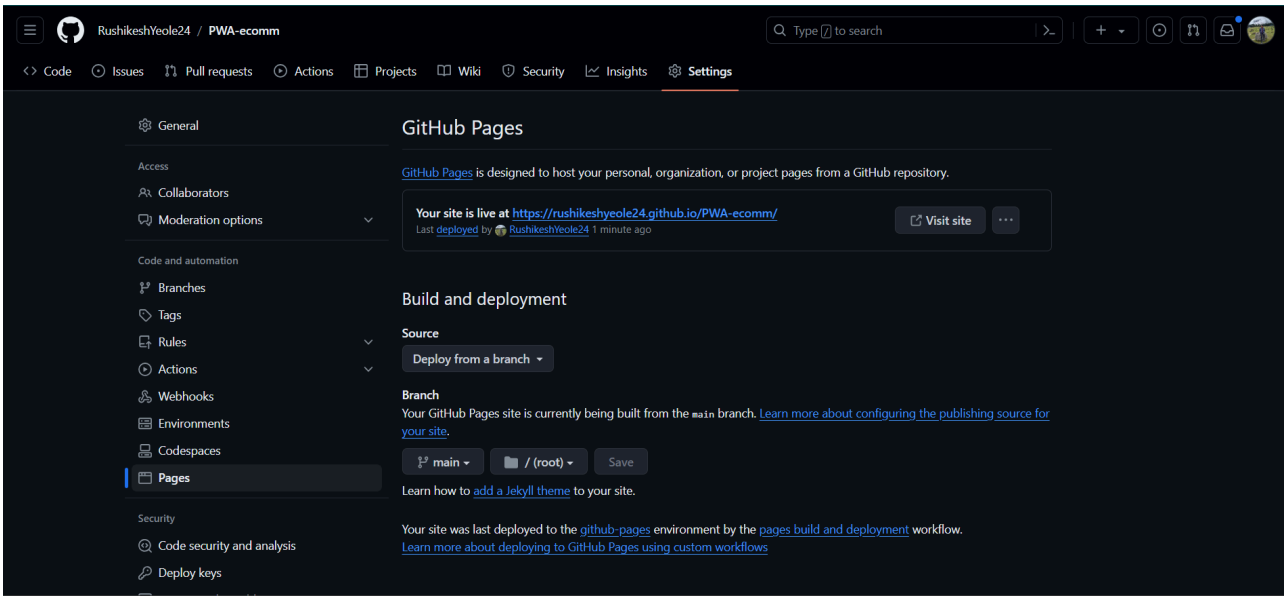
IMPLEMENTATION:











Github repo link: https://github.com/RushikeshYeole24/PWA-ecomm

Deployed website link: https://rushikeshyeole24.github.io/PWA-ecomm/

CONCUSSION: Deploying your Progressive Web App (PWA) on GitHub provides an accessible platform for users to access your application seamlessly. Leveraging GitHub Pages ensures reliability and scalability while enabling offline functionality. With automatic deployment and custom domain support, your PWA offers an enhanced user experience. By harnessing the power of GitHub's infrastructure, your PWA is poised to reach a wider audience and thrive in the digital landscape.