Assignment No.: 10
Title: Deployment of E-commerce PWA using GitHub Pages
Name: Harshit Raheja
Class: D15B
Roll Number: 45

Aim:
To study and implement deployment of an E-commerce Progressive Web App (PWA) using GitHub Pages.

Theory:
GitHub Pages is a free and simple static site hosting service provided by GitHub. It allows developers to host public webpages directly from a GitHub repository with minimal setup. Updates go live instantly upon pushing code to the configured branch, typically gh-pages.

Key Features of GitHub Pages:

1. Supports **Jekyll** for blog-aware content.
2. Allows **custom domain integration** via CNAME file and DNS setup.
3. Offers **automatic page generation** tools.
4. Can be integrated with any static site including PWAs.
5. Used by companies like **Lyft**, **CircleCI**, and **HubSpot**.

Pros of GitHub Pages:

- Familiar GitHub interface and version control.
- Extremely simple setup process.
- Built-in support for static websites.
- Free for public repositories.

Cons of GitHub Pages:

- Requires public repo unless using GitHub Pro.
- Lacks HTTPS support for custom domains (as of now).
- Limited support for advanced Jekyll plugins.

Firebase is a real-time backend platform from Google, commonly used for dynamic web applications.

Key Features of Firebase:

1. Real-time synchronization of data across clients using JSON data model.
2. Strong data security and scalability.
3. Supports RESTful endpoints and URL-based access.
4. Full-featured platform with **authentication**, **cloud messaging**, and **blob storage**.
5. Used by **Twitch**, **9GAG**, and **Instacart**.

Pros of Firebase:

- Google-hosted, highly reliable.
- HTTPS support for custom domains with auto SSL.
- 1 GB real-time DB + 1 GB blob storage.
- CLI-based deployment is powerful and fast.

Cons of Firebase:

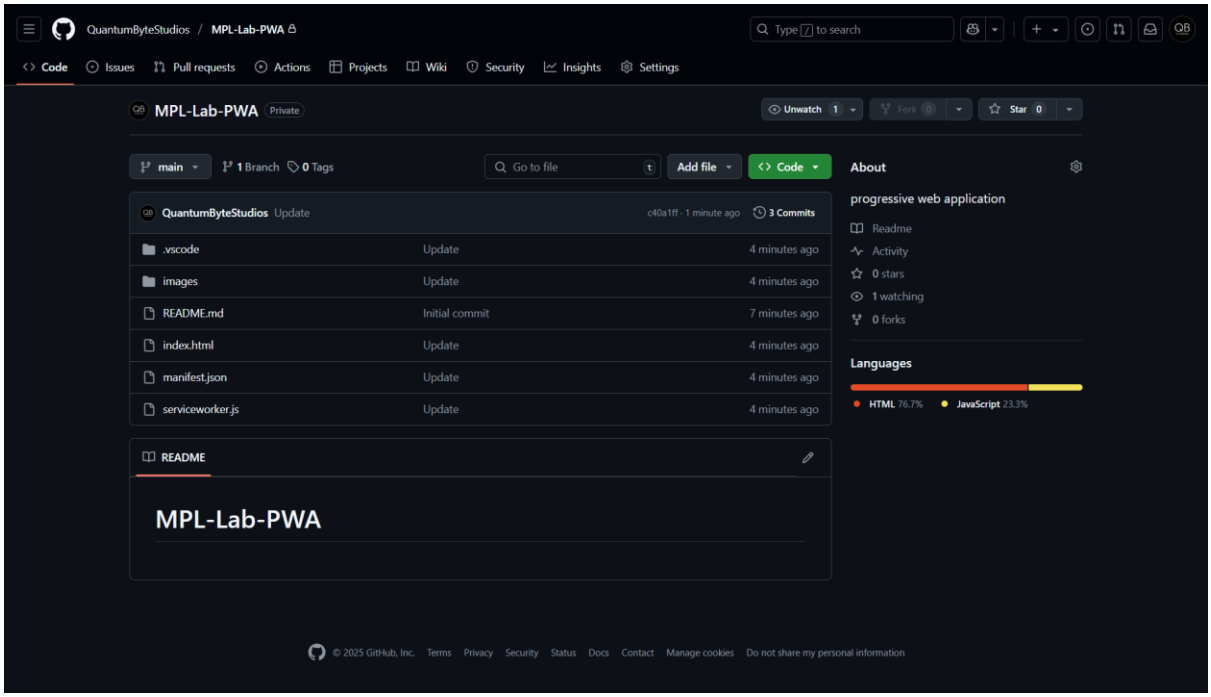
- 10 GB/month bandwidth limit (can be managed with CDN).
- No native support for static site generators like Jekyll.
- Requires comfort with command-line tools.

Why GitHub Pages for Deployment in This Experiment:

- Simpler and quicker for static PWAs like the one used here.
- No backend complexity required.
- Direct integration with Git and browser makes updates seamless.

GitHub Deployment Process (Summary):

1. Create a GitHub repo.
2. Push your PWA files (index.html, manifest, service worker, etc.).
3. Go to repository settings → Pages → Choose branch (main or gh-pages) → Save.
4. Your PWA is now live at https://quantumbytestudios.github.io/MPL-Lab-PWA/



Link: https://github.com/QuantumByteStudios/MPL-Lab-PWA

Output:

The E-commerce PWA is successfully deployed and accessible via GitHub Pages. Future updates can be managed through Git push, triggering instant deployment.