

## EXPERIMENT 10

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

**Theory:**

GitHub Pages is a feature provided by GitHub that allows users to host static websites directly from a GitHub repository. It is commonly used to publish personal, project, or organizational websites without requiring any additional web hosting service.

GitHub Pages supports static content such as HTML, CSS, JavaScript, and images. Developers often use it to host portfolios, documentation, or demo projects. The site can be served from the root of a repository or a specific branch like main or gh-pages.

Although GitHub Pages supports Jekyll, a static site generator that converts Markdown into HTML, it is not mandatory to use it. Users can host simple HTML and CSS files without any additional configuration.

Key features include free hosting for public repositories, support for custom domains, easy integration with Git workflows, and fast deployment. These benefits make it a popular choice for students, developers, and open-source projects.

To publish a site with GitHub Pages, a repository is selected and configured to serve content from a specific branch or folder. Once the files are pushed to the configured source, GitHub automatically builds and hosts the site. The website is then accessible through a URL in the format `https://username.github.io/repository-name`.

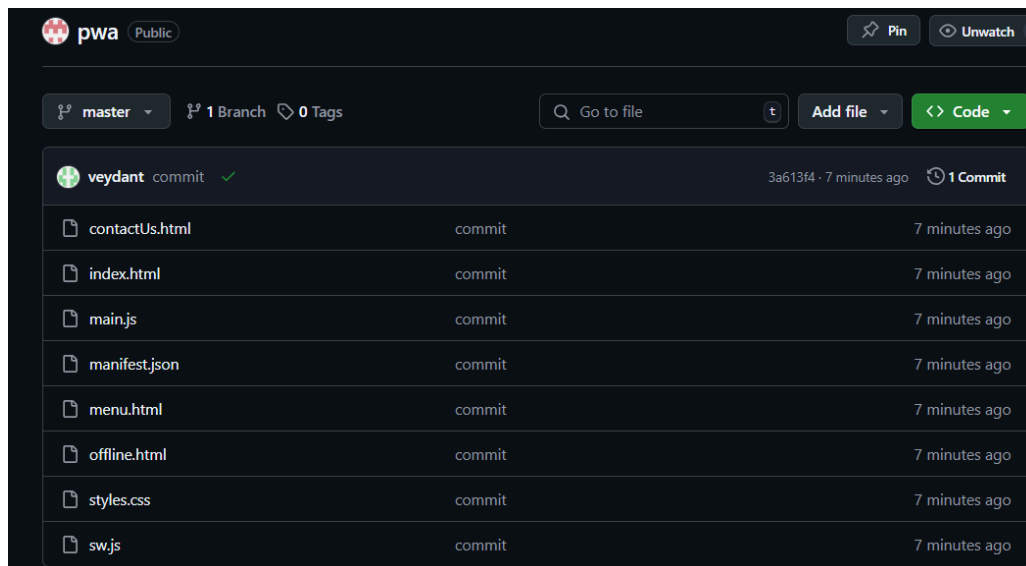
GitHub Pages is best suited for lightweight websites, especially those that do not require dynamic server-side processing.

**Steps:**

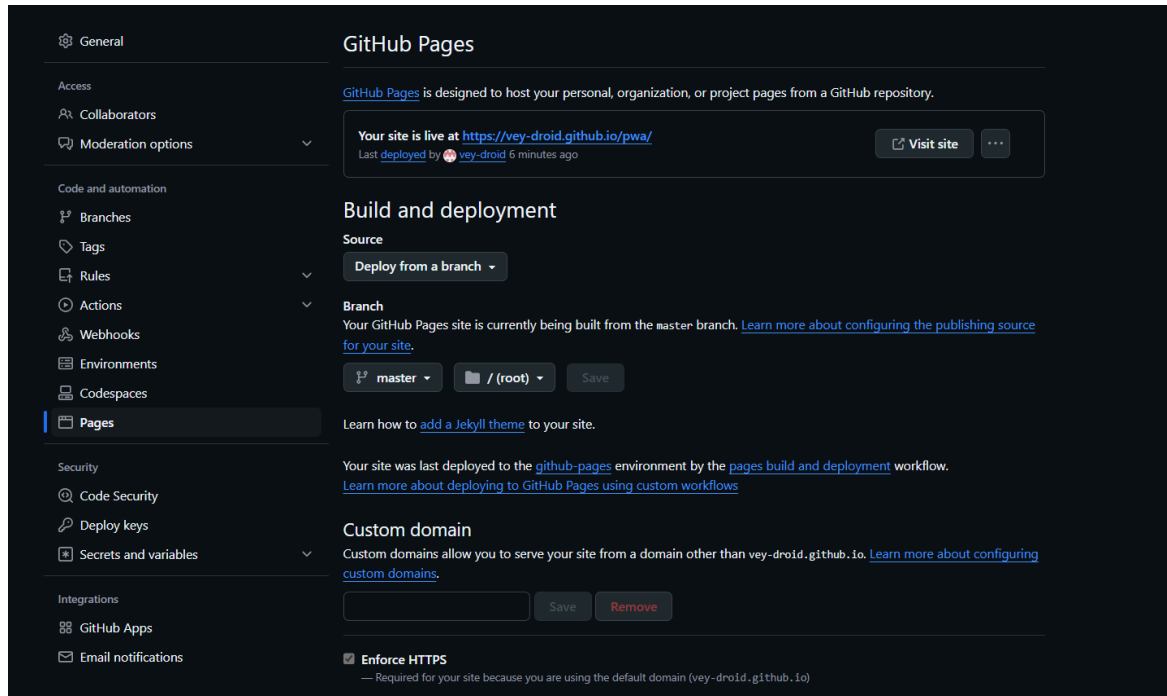
1. Create and Initialize a github repository and upload the code

```
PS C:\Users\veyda\Desktop\New folder\mpl\ecommerce-pwa>
git init
>>
Initialized empty Git repository in C:/Users/veyda/Desktop/New folder/mpl/ecommerce-pwa/.git/
PS C:\Users\veyda\Desktop\New folder\mpl\ecommerce-pwa> git add .
>>
>>
8 files changed, 243 insertions(+)
create mode 100644 contactUs.html
create mode 100644 index.html
create mode 100644 main.js
create mode 100644 manifest.json
create mode 100644 menu.html
create mode 100644 offline.html
create mode 100644 styles.css
create mode 100644 sw.js
PS C:\Users\veyda\Desktop\New folder\mpl\ecommerce-pwa> git remote add origin https://github.com/vey-droid/pwa
PS C:\Users\veyda\Desktop\New folder\mpl\ecommerce-pwa> git push -u origin master
>>
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 2.92 KiB | 748.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/vey-droid/pwa
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\veyda\Desktop\New folder\mpl\ecommerce-pwa>
```

2. Project is successfully uploaded to github.



3. Then go to settings->Pages there select root folder(/) as per your folder structure. Then click on save then it will get deployed and site link is created.



Github link: <https://vey-droid.github.io/pwa/>

## Conclusion:

We explored and carried out the deployment of an E-commerce Progressive Web App (PWA) using GitHub Pages. This platform turned out to be a convenient and reliable option for hosting PWAs, offering seamless integration with GitHub and free hosting for public repositories. The deployment process highlighted how developers can easily make their PWAs live without dealing with complex server setups. Overall, GitHub Pages proved to be a practical solution for hosting static PWA projects that have minimal backend needs.