

PWA EXPERIMENT NO : 10

NAME : Mahek Taneja

CLASS : D15B

ROLL_NO: 58

Aim:

To study and implement deployment of Ecommerce PWA to GitHub Pages

Theory:

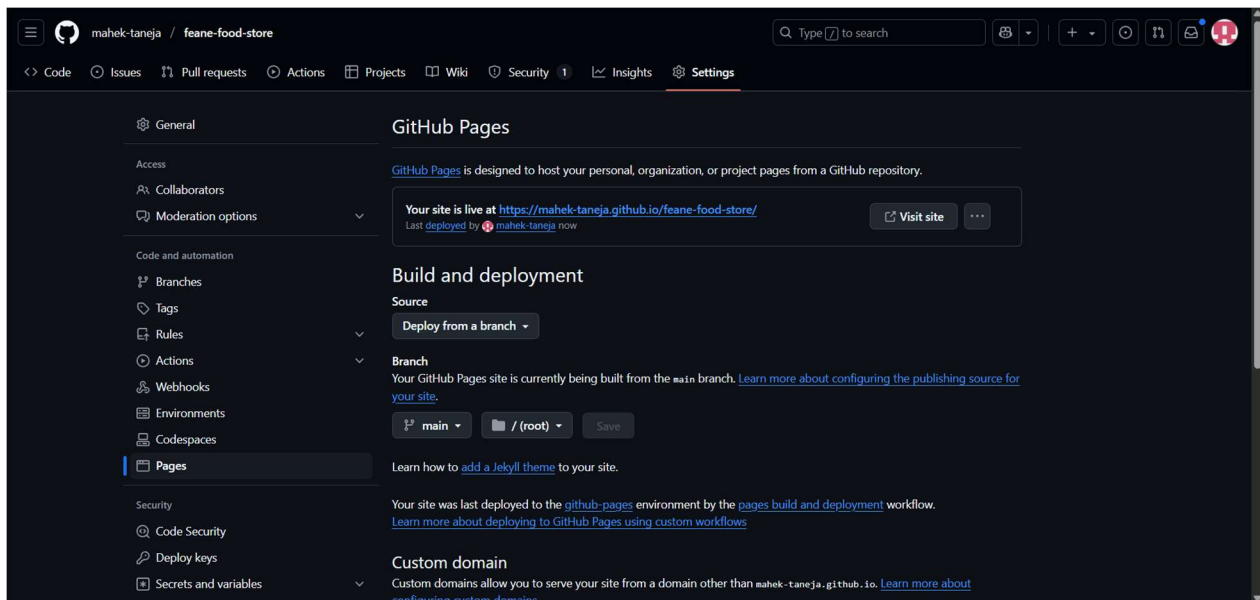
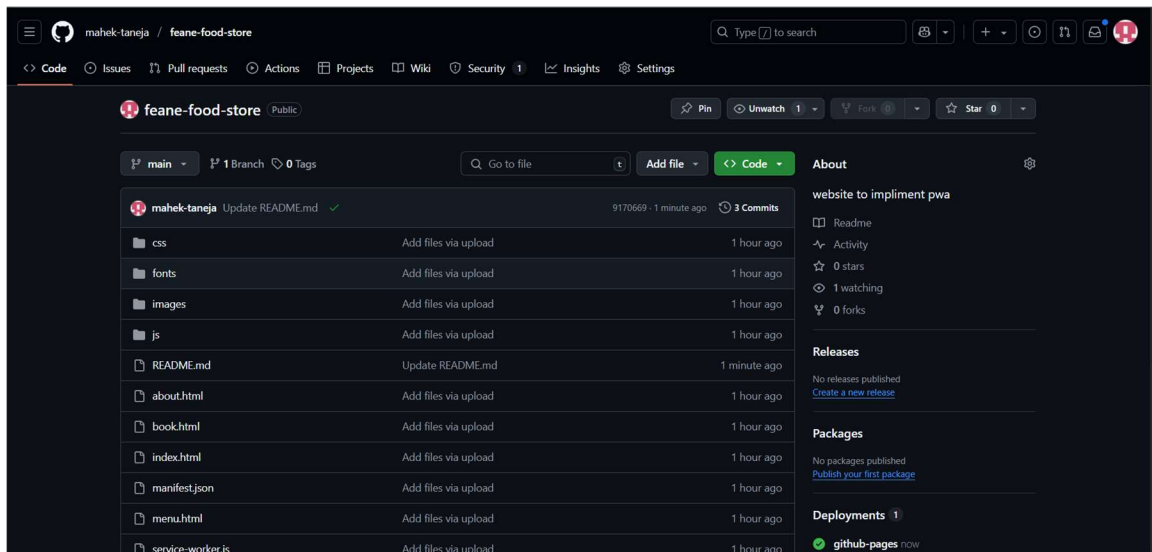
Progressive Web Apps (PWAs) are web applications that combine the best features of web and mobile apps. They are designed to work offline, load quickly, and provide a native app-like experience to users. Deploying a PWA ensures that users can install the app on their devices and interact with it even without a stable internet connection.

GitHub Pages is a free hosting service provided by GitHub that allows developers to publish static websites directly from a repository. It is commonly used for deploying front-end web projects, including PWAs.

To deploy an **E-commerce PWA** to GitHub Pages, the following steps are typically involved:

1. **Create and test the PWA locally** – ensure it includes a manifest file and service worker.
2. **Build the production version** of the app using a build tool (like `npm run build`).
3. **Push the build folder** (or configure the correct branch) to the GitHub repository.
4. **Configure GitHub Pages** in the repository settings to point to the correct branch or folder (main, docs, or /build depending on setup).
5. **Ensure all resources** (images, scripts, manifest, etc.) are linked correctly for the deployed path.

For frameworks like React, Vue, or Angular, additional configurations (like setting homepage in package.json) are often necessary to ensure the app routes correctly when deployed.



Conclusion:

Deploying an E-commerce PWA to GitHub Pages is a practical and cost-effective way to host and share your application online. It allows developers to easily showcase their projects and make them accessible to users as installable, offline-capable web apps. The process of deploying to GitHub Pages also reinforces the importance of organizing project files, setting correct paths, and testing deployment configurations. Successfully hosting a PWA demonstrates an understanding of both frontend development and deployment workflows, making it a crucial skill in modern web development.

Dimensions: Responsive ▾ 473 x 642 100% ▾ No throttling ▾

Application

Manifest

Service workers

Storage

Storage

Local storage

Session storage

Extension storage

IndexedDB

Cookies

Private state tokens

Interest groups

Shared storage

Cache storage

Storage buckets

Background services

Back/forward cache

Background fetch

Background sync

Bounce tracking mitigat...

Notifications

Payment handler

Periodic background sync

Speculative loads

Push messaging

Reporting API

Frames

top

Identity

Name Feane Restaurant

Short name Feane

Description Delicious fast food from Feane!

Computed App ID <https://mahek-taneja.github.io/index.html> [Learn more](#)

Note: id is not specified in the manifest, start_url is used instead. To specify an App ID that matches the current identity, set the id field to /index.html.

Presentation

Start URL [/index.html](#)

Theme color #ff6600

Background color #ffffff

Orientation

Display standalone

Protocol Handlers

Define protocol handlers in the [manifest](#) to register your app as a handler for custom protocols when your app is installed.

Need help? Read [URL protocol handler registration for PWAs](#).

Icons

Show only the minimum safe area for maskable icons

Need help? Read the [documentation on maskable icons](#).

Window Controls Overlay

Feane

Fast Food Restaurant

Doloremque, itaque aperiam facilis rerum, commodi, temporibus sapiente ad mollitia laborum quam quisquam esse error unde.

Order Now