

Experiment No. 10

Aim:

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

Theory:

GitHub Pages

Public web pages are freely hosted and easily published. Public webpages hosted directly from your GitHub repository. Just edit, push, and your changes are live.

GitHub Pages provides the following key features:

1. Blogging with Jekyll
2. Custom URL
3. Automatic Page Generator

Reasons for favoring this over Firebase:

1. Free to use
2. Right out of github
3. Quick to set up

GitHub Pages is used by Lyft, CircleCI, and HubSpot.

GitHub Pages is listed in 775 company stacks and 4401 developer stacks.

Pros

1. Very familiar interface if you are already using GitHub for your projects.
2. Easy to set up. Just push your static website to the gh-pages branch and your website is ready.
3. Supports Jekyll out of the box.
4. Supports custom domains. Just add a file called CNAME to the root of your site, add an A record in the site's DNS configuration, and you are done.

Cons

1. The code of your website will be public, unless you pay for a private repository.
2. Currently, there is no support for HTTPS for custom domains. It's probably coming soon though.
3. Although Jekyll is supported, plug-in support is rather spotty.

Firestore

The Realtime App Platform. Firestore is a cloud service designed to power real-time, collaborative applications. Simply add the Firestore library to your application to gain access to a shared data structure; any changes you make to that data are automatically synchronized with the Firestore cloud and with other clients within milliseconds.

Some of the features offered by Firestore are:

1. Add the Firestore library to your app and get access to a shared data structure. Any changes made to that data are automatically synchronized with the Firestore cloud and with other clients within milliseconds.
2. Firestore apps can be written entirely with client-side code, update in real-time out-of-the-box, interoperate well with existing services, scale automatically, and provide strong data security.
3. Data Accessibility- Data is stored as JSON in Firestore. Every piece of data has its own URL which can be used in Firestore's client libraries and as a REST endpoint. These URLs can also be entered into a browser to view the data and watch it update in real-time.

Reasons for favoring over GitHub Pages:

1. Realtime backend made easy
2. Fast and responsive

Instacart, 9GAG, and Twitch are some of the popular companies that use Firestore
Firestore has a broader approval, being mentioned in 1215 company stacks & 4651 developers stacks

Pros

1. Hosted by Google. Enough said.
2. Authentication, Cloud Messaging, and a whole lot of other handy services will be available to you.
3. A real-time database will be available to you, which can store 1 GB of data.
4. You'll also have access to a blob store, which can store another 1 GB of data.
5. Support for HTTPS. A free certificate will be provisioned for your custom domain within 24 hours.

Cons

1. Only 10 GB of data transfer is allowed per month. But this is not really a big problem, if you use a CDN or AMP.
2. Command-line interface only.
3. No in-built support for any static site generator.

Link to our GitHub repository: https://github.com/Shamaila02/PWA_Fuzzy

Hosted Link: https://shamaila02.github.io/PWA_Fuzzy/

Github Screenshot:

The screenshot shows the GitHub repository page for 'PWA_Fuzzy' by user 'Shamaila02'. The repository is public and has 2 commits. The file list includes 'assets', 'README.md', 'app.js', 'cart.html', 'categories.html', 'checkout.html', 'coupon.html', 'create-account.html', 'empty-cart.html', 'empty-notification.html', 'empty-order-history.html', 'empty-search.html', and 'empty-wishlist.html'. The right sidebar shows 'About' (no description), 'Releases' (no releases published), 'Packages' (no packages published), 'Deployments' (1 deployment by github-pages 5 hours ago), and 'Languages' (a bar chart showing the distribution of languages).

The screenshot shows the 'GitHub Pages' settings page for the 'PWA_Fuzzy' repository. The 'General' tab is selected. The 'Access' section shows 'Collaborators' and 'Moderation options'. The 'Code and automation' section shows 'Branches', 'Tags', 'Rules', 'Actions', 'Webhooks', 'Environments', 'Codespaces', and 'Pages' (which is highlighted). The 'Security' section shows 'Code security and analysis', 'Deploy keys', and 'Secrets and variables'. The 'GitHub Pages' section states that the site is live at https://shamaila02.github.io/PWA_Fuzzy/ and was last deployed by Shamaila02 5 hours ago. The 'Build and deployment' section shows the 'Source' as 'Deploy from a branch' and the 'Branch' as 'main'. The 'Custom domain' section is also visible.

The image displays two screenshots of the GitHub web interface for the repository 'Shamaila02 / PWA_Fuzzy'.

The top screenshot shows the 'Actions' tab. The left sidebar lists 'All workflows', 'Workflows', and 'pages-build-deployment' (which is selected). Below these are 'Management' options: 'Caches', 'Deployments', and 'Runners'. The main content area is titled 'pages-build-deployment' and shows '1 workflow run'. A single run is listed with a green checkmark, titled 'pages build and deployment', and notes it is 'pages-build-deployment #1: by Shamaila02', completed '5 hours ago' in '51s'.

The bottom screenshot shows the 'Deployments' tab. The left sidebar lists 'All deployments' (selected), 'Environments', and 'github-pages'. Below these are options to 'Manage environments', 'Give beta feedback', and 'Opt out of beta view'. The main content area is titled 'All deployments' and shows 'Latest deployments from select environments'. It features a deployment card for 'github-pages' with a green checkmark, noting it was 'Last deployed 5 hours ago' to 'https://shamaila02.github.io/PWA_Fuzzy/'. Below this, a section titled '1 deployments' shows 'FuzzyProject added' with an 'Active' status, noting it was 'Deployed to github-pages by Shamaila02 via pages-build-deployment #1'.

Conclusion: In this experiment we have Successfully deployed the Ecommerce PWA to GitHub Pages.