EXPERIMENT NO: - 10

Name:- Tejas Gunjal **Class:-** D15A **Roll:No: -** 18

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

Theory: -

GitHub Pages

GitHub Pages is a free hosting service that allows users to publish public webpages directly from a GitHub repository. It is particularly useful for static websites, project documentation, and blogs. It seamlessly integrates with Git version control, ensuring automatic updates with every commit. With built-in support for Jekyll, custom domains, and HTTPS (for GitHub subdomains), it provides an easy and efficient way to deploy static sites.

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.

Firebase

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

Some of the features offered by Firebase are:

1. Add the Firebase library to your app and get access to a shared data structure. Any changes made to that data are automatically synchronized with the Firebase cloud and with other clients within milliseconds.
2. Firebase 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 Firebase. Every piece of data has its own URL which can be used in Firebase's client libraries and as a REST endpoint. These URLs can also be entered into a browser to view the data , 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 Firebase
Firebase 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/tejasgunjal021/college_pwa_website

Link to our Hosted Website : -

https://tejasgunjal021.github.io/college_pwa_website/

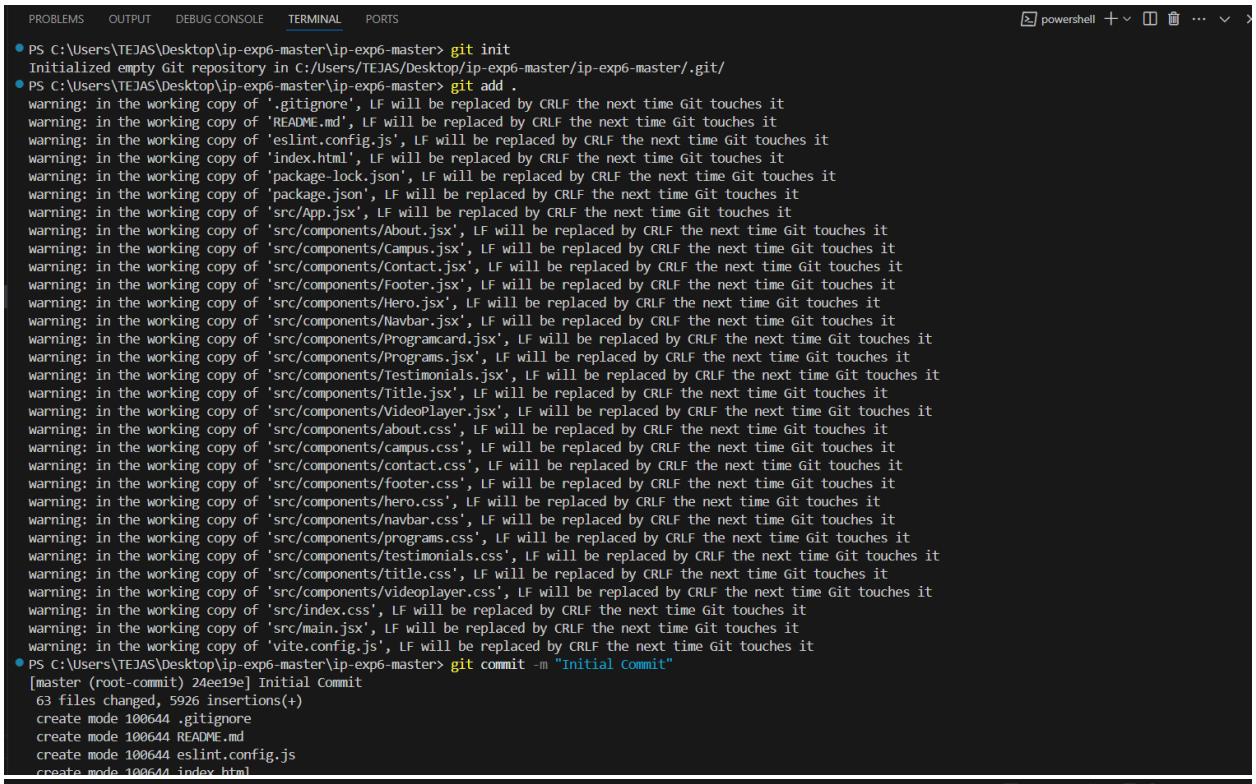
Method Followed [gh- pages] : -

We deployed a **React-based PWA** to **GitHub Pages** using the gh-pages package. The process involved:

1. **Setting up GitHub Pages** – Configuring the repository to support deployment.
2. **Adding homepage and scripts in package.json** – Ensuring proper routing support.
3. **Building the React App** – Creating the production-ready files.
4. **Deploying using gh-pages** – Pushing the build folder to the gh-pages branch.
5. **Verifying the Deployment** – Ensuring the website works correctly on GitHub Pages.

Github Screenshots: -

- **Pushing my PWA application into the github repository**



- **Installing gh-pages Package** – Setting up the deployment tool.

- **Adding homepage and Deployment scripts in package.json**

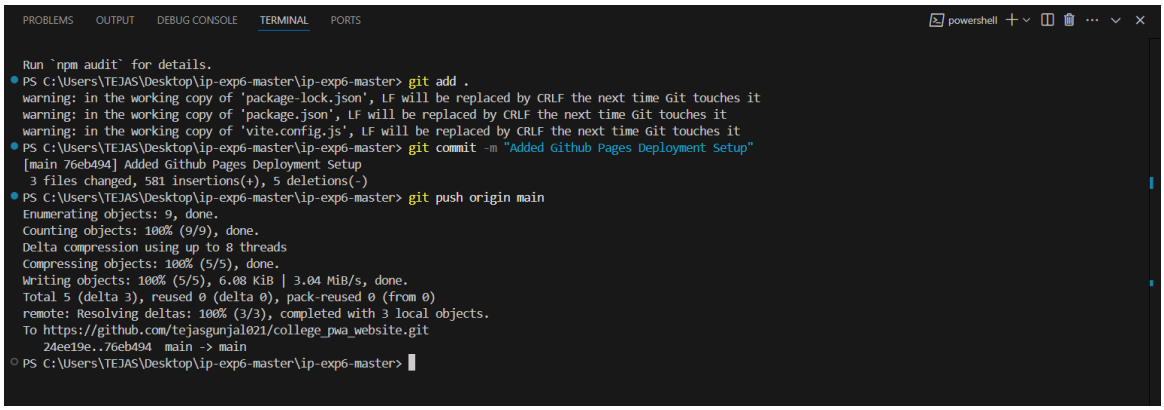
```
{
 "name": "college-website",
 "homepage": "https://tejasgunjal021.github.io/college_pwa_website",

 "private": true,
 "version": "0.0.0",
 "type": "module",
 "scripts": {
  "dev": "vite",
  "build": "vite build",
  "lint": "eslint .",
  "preview": "vite preview",
  "predeploy": "vite build",
  "deploy": "gh-pages -d dist"
 },
 "dependencies": {
  "@fortawesome/fontawesome-free": "^6.6.0",
  "react": "^18.3.1",
  "react-dom": "^18.3.1",
  "react-router-dom": "^6.27.0"
 },
 "devDependencies": {
  "@eslint/js": "^9.11.1",
  "@types/react": "^18.3.10",
  "@types/react-dom": "^18.3.0",
  "@vitejs/plugin-react": "^4.3.2",
  "eslint": "^9.11.1",
  "eslint-plugin-react": "^7.37.0",
  "eslint-plugin-react-hooks": "^5.1.0-rc.0",
  "eslint-plugin-react-refresh": "^0.4.12",
  "gh-pages": "^6.3.0",
  "globals": "^15.9.0",
  "vite": "^5.4.8"
 }
}
```

- **Update vite.config.js**

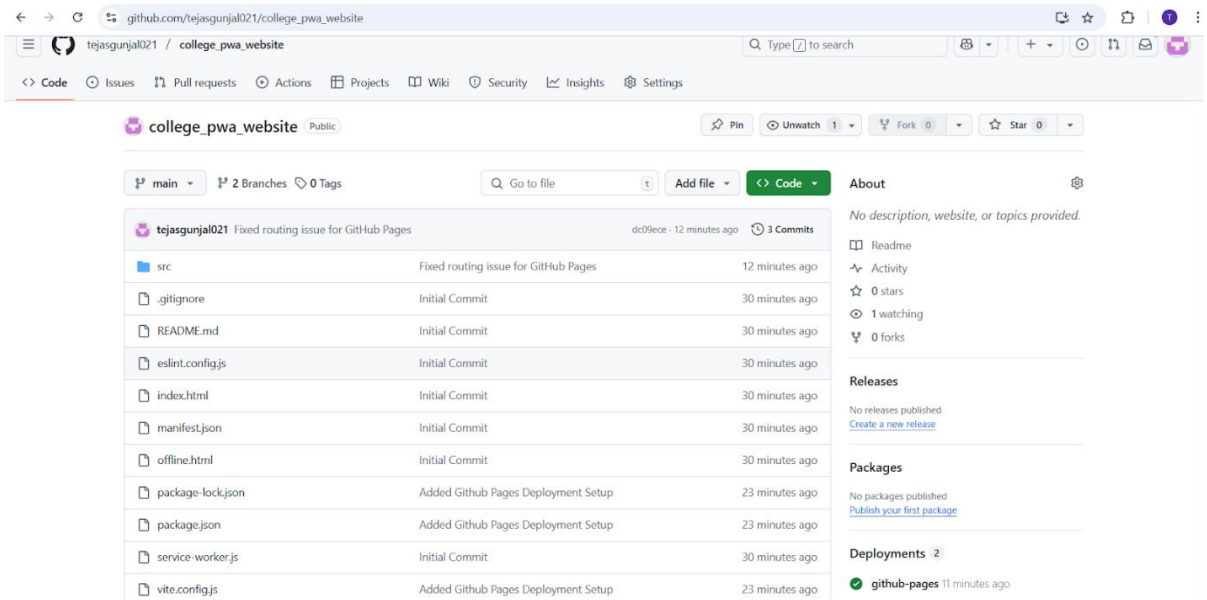
  Since Vite serves the app from / by default, you must configure it to serve from the correct subdirectory.

```
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react";

export default defineConfig({
  plugins: [react()],
  base: "/college_pwa_website/",  // Add this line
});
```

- **Updating the changes in the repository**



- **Code pushed to the repository**

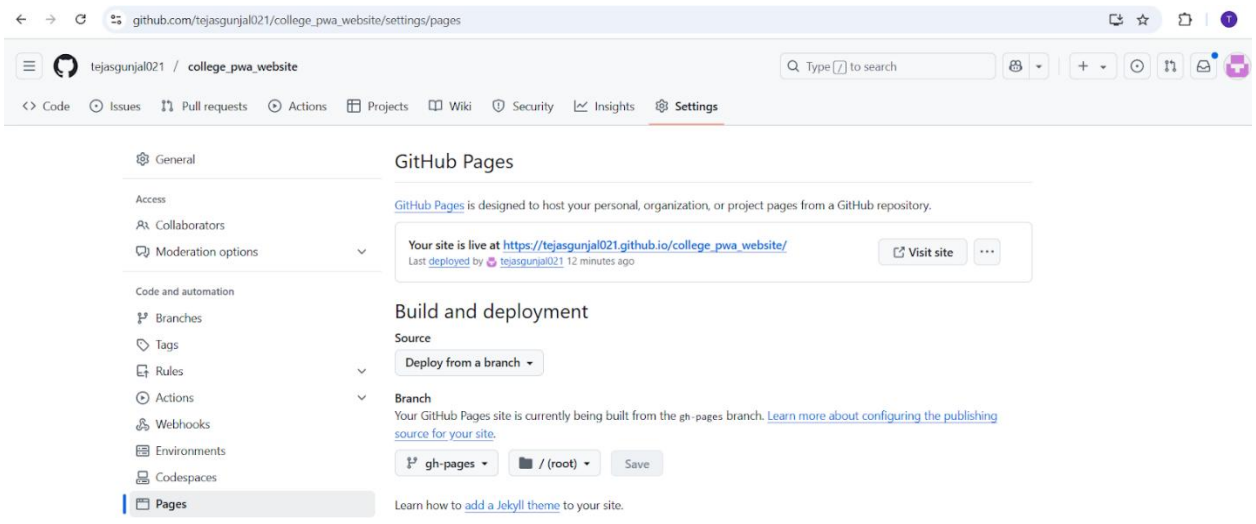
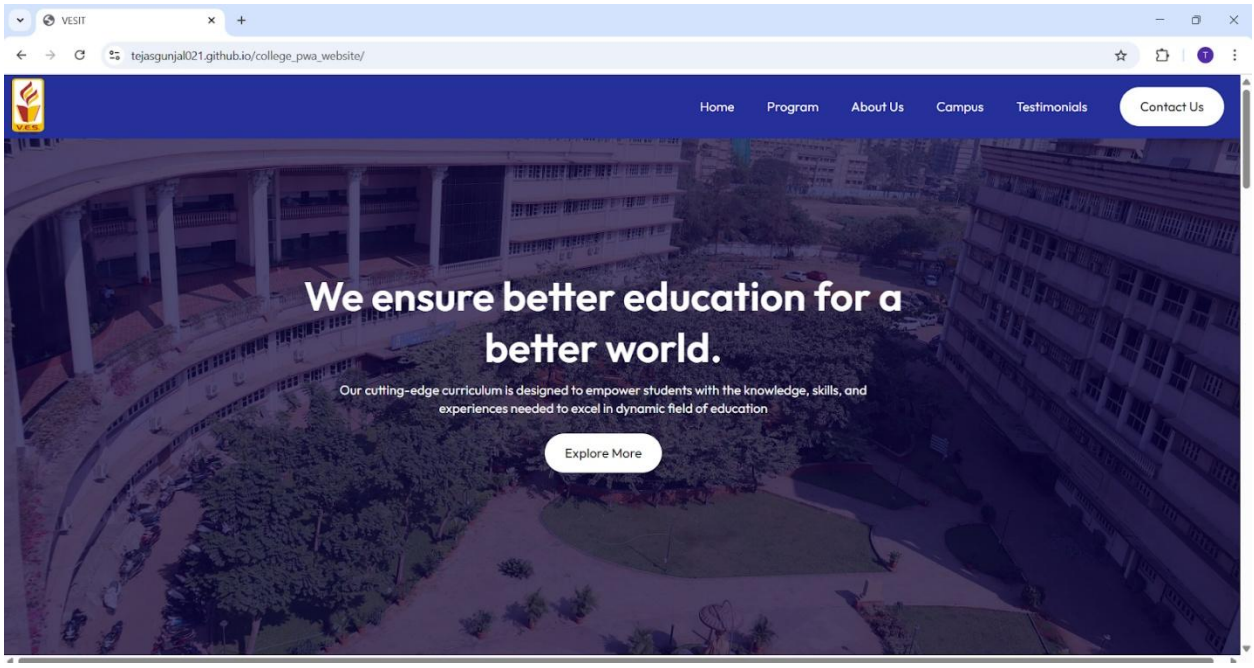


➢ Deploy to github pages

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\TEJAS\Desktop\ip-exp6-master\ip-exp6-master> npm run deploy

> college-website@0.0.0 predeploy
> vite build

vite v5.4.14 building for production...
✓ 82 modules transformed.
dist/index.html                                0.58 kB │ gzip:  0.33 kB
dist/assets/manifest-G75MMNiJ.json             0.71 kB │ gzip:  0.35 kB
dist/assets/fa-v4compatibility-C9RhG_FT.woff2  4.80 kB
dist/assets/fa-v4compatibility-CCth-dXg.ttf   10.84 kB
dist/assets/msg-BJ1-p5Un.png                  13.63 kB
dist/assets/logo-Bu6wwlcJ.png                 14.24 kB
dist/assets/icon-C1OtVbCZ.png                 17.12 kB
dist/assets/play-DZJCAgKb.png                 17.42 kB
dist/assets/right-arrow-BcmZ5rr2.png          17.59 kB
dist/assets/left-arrow-D-vK3ks6.png           18.44 kB
dist/assets/user1-DjTToPK1.png                24.41 kB
dist/assets/ai-D8sSgONg.jpg                   25.19 kB
dist/assets/fa-regular-400-BjRzuEpd.woff2     25.47 kB
dist/assets/user2-Dvg8F2pi.png                27.28 kB
dist/assets/location-CsUE0Atg.png             30.25 kB
dist/assets/user6-g8XIt7bn.png                32.17 kB
dist/assets/user3-CZ6tJ5BY.png                36.06 kB
dist/assets/user4-BtQSOxcZ.png                36.71 kB
dist/assets/user5-CXG3eEZr.png                38.78 kB
dist/assets/iot-By66MvuU.jpg                  40.74 kB
dist/assets/mail-Bryx0Fa9.png                 42.74 kB
dist/assets/electronics-CxeaB2SO.jpg          44.67 kB
dist/assets/mechanics-7TpPCQUc.jpg            57.48 kB
dist/assets/fa-regular-400-DZaxPHgR.ttf       68.06 kB
dist/assets/fa-brands-400-D_cYUPeE.woff2     118.68 kB
dist/assets/fa-solid-900-CTAAxXor.woff2      158.22 kB
dist/assets/robotics-Bmec4tgd.jpg            180.11 kB
dist/assets/fa-brands-400-D1LuMI3I.ttf       210.79 kB
dist/assets/program-1-DYXM3znK.png           270.08 kB
dist/assets/fa-solid-900-D0aA9rwL.ttf        426.11 kB
dist/assets/gallery-1-BGCASvfo.png           517.44 kB
dist/assets/about-BMKaheDG.png             1,327.94 kB
dist/assets/college-Dx6yeVSG.png           4,579.16 kB
dist/assets/gallery-0-CHbQgws-.png         7,094.24 kB
```

➢ Enable Github Pages

1. Go to your repository on GitHub.
2. Click on **Settings** > **Pages**.
3. In that you will see the github hosted link for your application.



- **Hosted Website: -** https://tejasgunjal021.github.io/college_pwa_website/