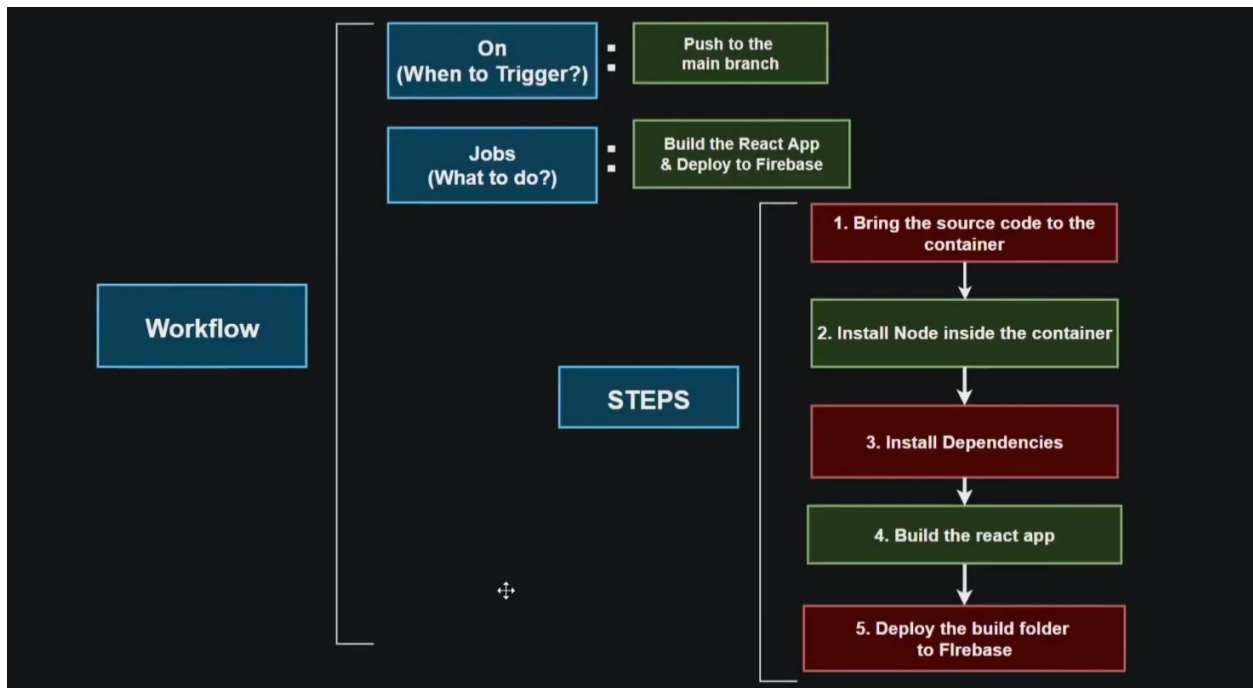
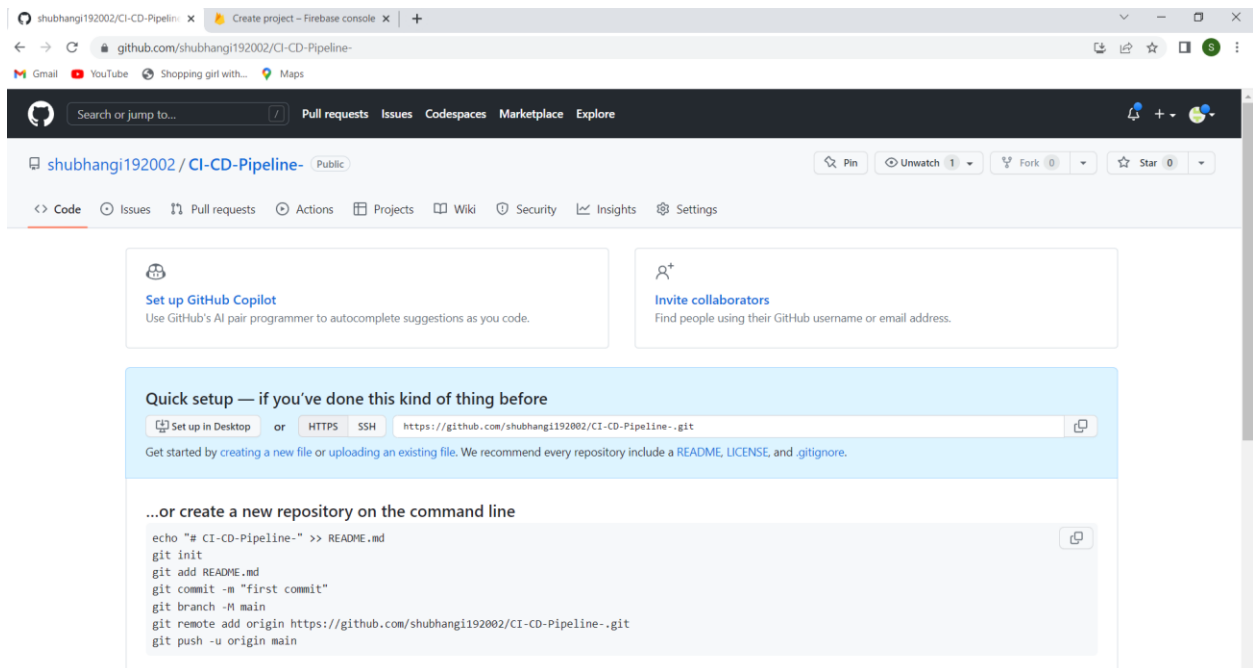


Workflow



Steps:

Step 1: Creating a Github Repo



Step 2: Creating a React-App on Windows powershell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\DELL> cd Desktop
PS C:\Users\DELL\Desktop> cd Certificates
PS C:\Users\DELL\Desktop\Certificates> cd React
PS C:\Users\DELL\Desktop\Certificates\React> npm create-react-app react-app

Creating a new React app in C:\Users\DELL\Desktop\Certificates\React\react-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1423 packages in 4m

235 packages are looking for funding
  run `npm fund` for details

Installing template dependencies using npm...
added 62 packages, and changed 1 package in 15s

235 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

removed 1 package, and audited 1485 packages in 9s

235 packages are looking for funding
  run `npm fund` for details

0 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Success! Created react-app at C:\Users\DELL\Desktop\Certificates\React\react-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes the tooling. In production, you can use the command-line tools directly, without needing the CLI tools to bundle your app.
```

Step 3: Creating firebase workflow on windows Powershell

```
Windows PowerShell
Happy hacking!
PS C:\Users\DELL\Desktop\Certificates\React> cd react-app
PS C:\Users\DELL\Desktop\Certificates\React\react-app> firebase init

##### 
##          ##          ##### 
##    ##    ##    ##    ##    ##    ##    ##    ##    ##    ## 
##           #####       #####       ##### 
##    ##    ##    ##    ##    ##    ##    ##    ##    ##    ## 
##    #####  ##          #####  ##    ##    ##    ##    ## 

You're about to initialize a Firebase project in this directory:

c:\Users\DELL\Desktop\Certificates\React\react-app

Are you ready to proceed? y/n
Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys

-- Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

Please select an option: use existing project / create new project
Select a default Firebase project for this directory: counter-based (counter)
1 Using project counter-based (Counter)

--- Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

What do you want to use as your public directory? build
Configure as a single-page app (rewrite all urls to /index.html)? yes
Set up automatic builds and deploys with GitHub? no
0 Wrote build/index.html

1 Writing configuration info to firebase.json...
1 Writing project information to .firebaserc...

2 Firebase initialization complete!
PS C:\Users\DELL\Desktop\Certificates\React\react-app> firebase login:ci
1 Authenticating with a "login:ci" token is deprecated and will be removed in a future major version of "firebase-tools". Instead, use a service account key with "GOOGLE_APPLICATION_CREDENTIALS": https://cloud.google.com/docs/authentication/getting-started

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fefbe6d47b9e61518b59e03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform-response_type=code&state=320700957&redirect_uri=http%3A%2F%2Flocalhost%3A9005

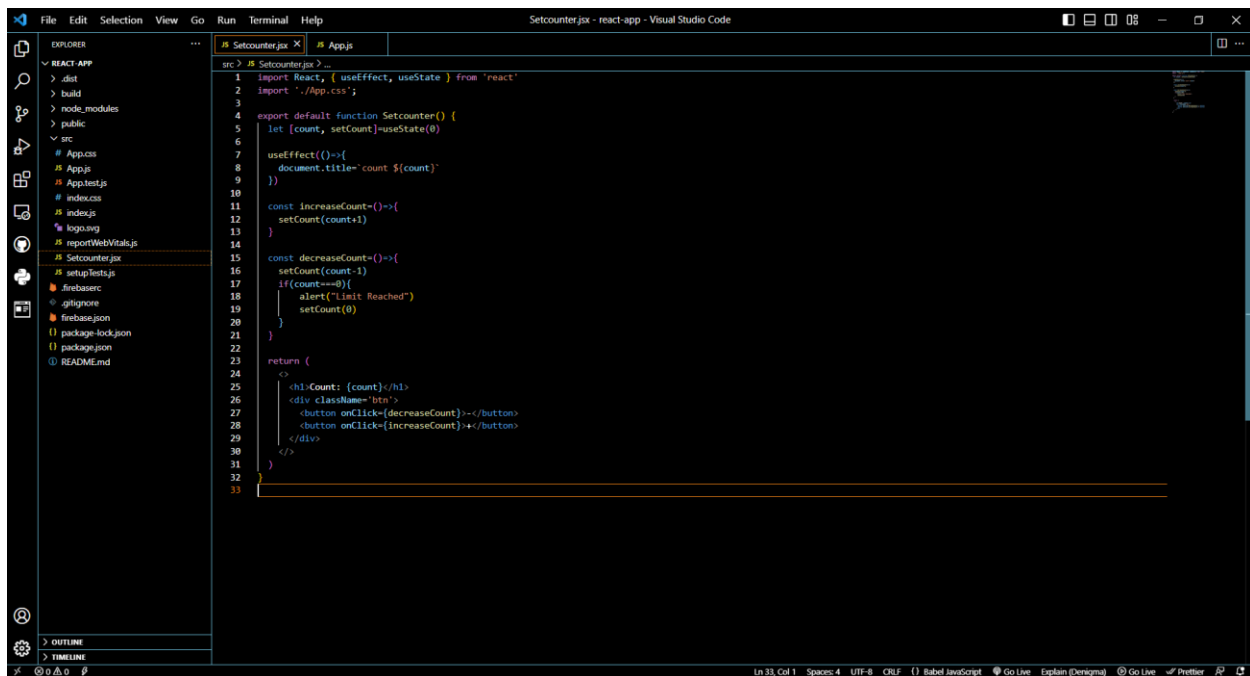
Waiting for authentication...

1 Success! Use this token to login on a CI server:
1//BgAMGVYDNKRncgVlRAAGBASnf-19IrLOvux8bwq0-18kUcG496kc0BrAsPpOfvASH3Aeahy9Z929UPagJbb1bhSPfBRMFCH

Example: firebase deploy --token "FIREBASE_TOKEN"

PS C:\Users\DELL\Desktop\Certificates\React\react-app> cd ..
PS C:\Users\DELL\Desktop\Certificates\React\react-app>
```

Step 4: Creating file for counter as Setcounter.jsx



The screenshot shows the Visual Studio Code editor with the 'Setcounter.jsx' file open. The file contains the following code:

```
1 import React, { useEffect, useState } from 'react'
2 import './App.css'
3
4 export default function Setcounter() {
5   let [count, setCount]=useState(0)
6
7   useEffect(()=>{
8     document.title=`count ${count}`
9   })
10
11   const increaseCount=()=>{
12     setCount(count+1)
13   }
14
15   const decreaseCount=()=>{
16     setCount(count-1)
17     if(count==0){
18       alert("Limit Reached")
19       setCount(0)
20     }
21   }
22
23   return (
24     <>
25     <h1>Count: {count}</h1>
26     <div className="btn">
27       <button onClick={decreaseCount}></button>
28       <button onClick={increaseCount}></button>
29     </div>
30   )
31 }
32
33
```

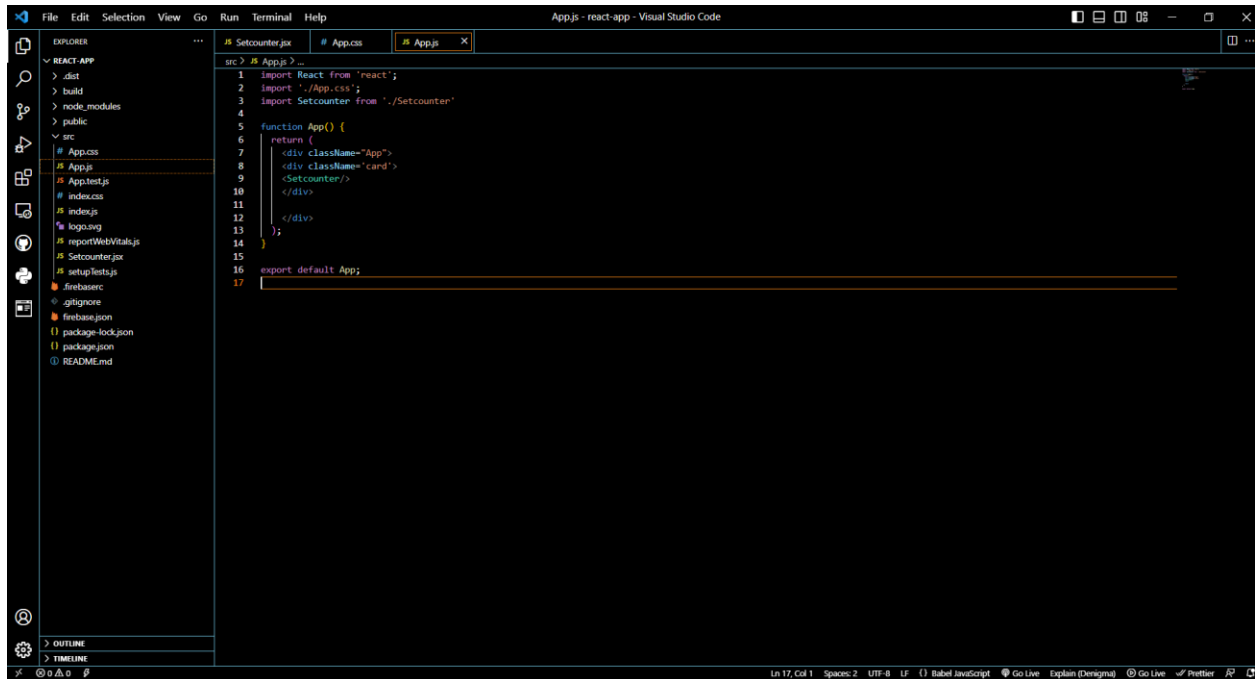
Step 5: Apply Style to Setcounter.jsx file , for that create App.css file



The screenshot shows the Visual Studio Code editor with the 'App.css' file open. The file contains the following CSS code:

```
1 .App{
2   height: 100vh;
3   display: flex;
4   flex-direction: column;
5   align-items: center;
6   justify-content: center;
7   background-color: #464b74;
8   color: #fff;
9 }
10
11 .card{
12   width: 400px;
13   height: 150px;
14   background-color: #fff;
15   color: #333;
16   padding: 30px 20px;
17   box-shadow: 0 0 7px #464b74;
18   border-radius: 20px;
19   display: flex;
20   flex-direction: column;
21   align-items: center;
22   text-align: center;
23 }
24
25 button{
26   padding: 5px 30px;
27   cursor: pointer;
28   background-color: #ff9900;
29   color: #000;
30   font-size: 20px;
31   margin: 0 20px;
32   border: none;
33   border-radius: 10px;
34   text-align: center;
35 }
```

Step 6: Calling Setcounter.jsx file ,App.css file in App.js



The screenshot shows the Visual Studio Code editor with a React app project. The Explorer sidebar on the left shows the file structure, including the 'src' directory with files like 'App.js', 'App.css', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'Setcounter.jsx', 'setupTests.js', 'firebase', 'firebase.json', 'package-lock.json', 'package.json', and 'README.md'. The main editor area displays the 'App.js' file with the following code:

```
src > # App.js > ...
1 import React from 'react';
2 import './App.css';
3 import Setcounter from './Setcounter';
4
5 function App() {
6   return (
7     <div className="App">
8       <div className="card">
9         <Setcounter/>
10      </div>
11    </div>
12   );
13 }
14
15 export default App;
```

Step 6: Do necessary changes in App.test.js file



The screenshot shows the Visual Studio Code editor with the same React app project. The Explorer sidebar on the left shows the file structure, including the 'src' directory with files like 'App.js', 'App.css', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'Setcounter.jsx', 'setupTests.js', 'firebase', 'firebase.json', 'package-lock.json', 'package.json', and 'README.md'. The main editor area displays the 'App.test.js' file with the following code:

```
src > # App.test.js > ...
1 import { render, screen } from '@testing-library/react';
2 import App from './App';
3
4 test('renders correctly', () => {
5   render(<App />);
6   const buttons = screen.getAllByRole('button');
7   expect(buttons).toHaveLength(2);
8 })
```

Step 7: Commit changes to github from git bash

```
MINGW64~/Users/DELL/Desktop/Certificates/React/react-app
$ git init
Reinitialized existing Git repository in C:/Users/DELL/Desktop/Certificates/React/react-app/.git/

MINGW64~/Desktop/Certificates/React/react-app (main)
$ git add .
warning: in the working copy of 'mintty.exe.stackdump', LF will be replaced by CRLF the next time Git touches it

MINGW64~/Desktop/Certificates/React/react-app (main)
$ git commit -m "first commit"
[main e2b7bf0] first commit
1 file changed, 22 insertions(+)
create mode 100644 mintty.exe.stackdump

MINGW64~/Desktop/Certificates/React/react-app (main)
$ git branch -M main

MINGW64~/Desktop/Certificates/React/react-app (main)
$ git remote add origin https://github.com/shubhangi192002/CI-CD-Pipeline-.git

MINGW64~/Desktop/Certificates/React/react-app (main)
$ git push -u origin main
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 4 threads
Compressing objects: 100% (28/28), done.
Writing objects: 100% (28/28), 173.08 Kib | 2.62 MiB/s, done.
Total 28 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/shubhangi192002/CI-CD-Pipeline-.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

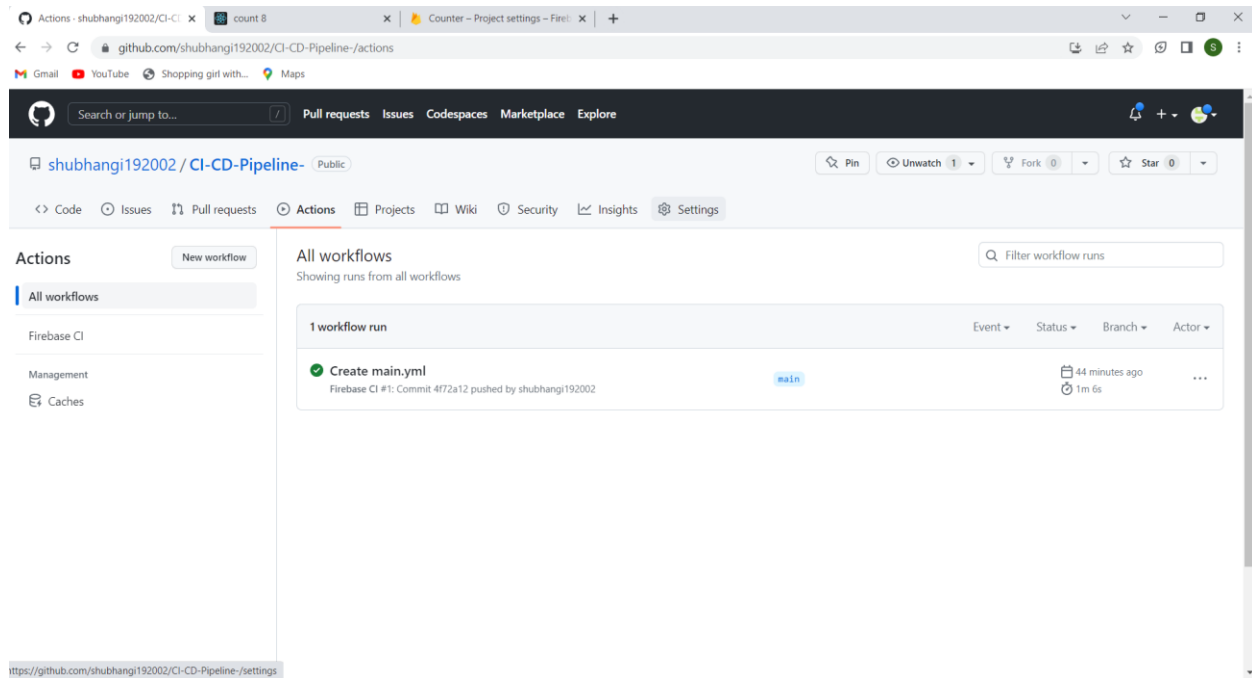
MINGW64~/Desktop/Certificates/React/react-app (main)
$ |
```

Step 8: Creating Workflow

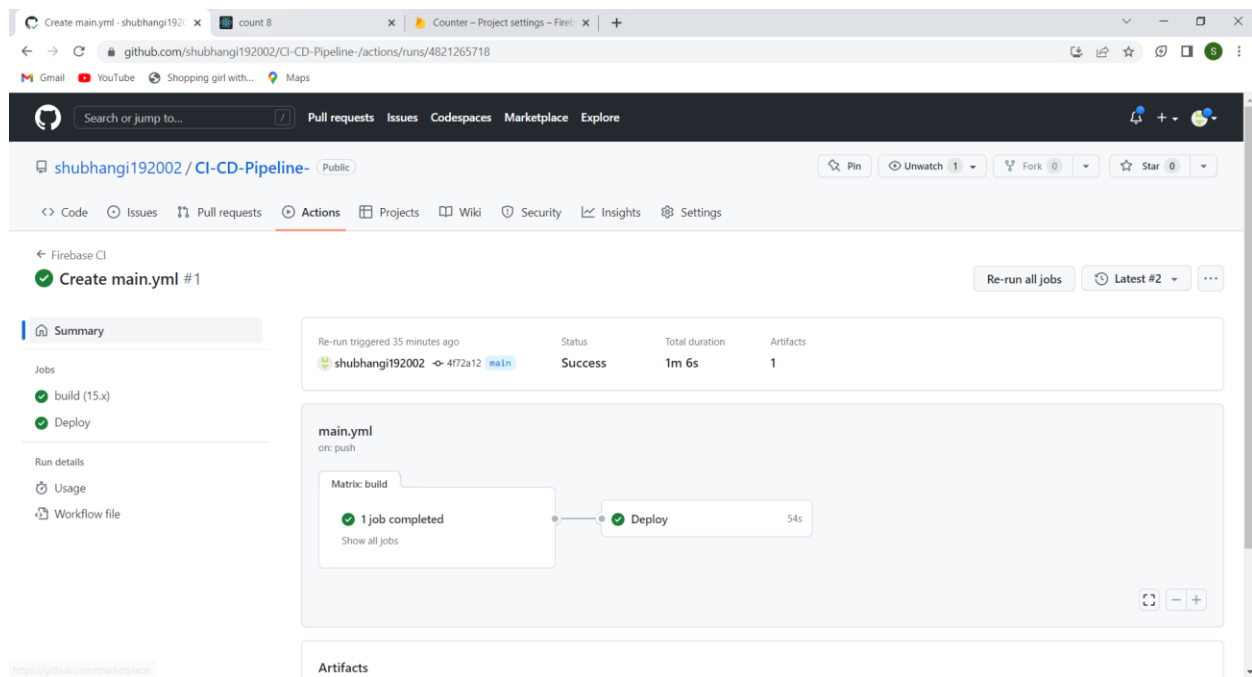
The screenshot shows a GitHub repository page for 'shubhangi192002 / CI-CD-Pipeline-'. The 'Code' tab is active, displaying a file named 'main.yml' in the 'github / workflows /' directory. The file content is a GitHub Actions workflow for Firebase CI. The right sidebar shows the 'Marketplace' for GitHub Actions, featuring actions like 'Setup Node.js environment', 'Upload a Build Artifact', and 'Setup Java JDK'.

```
1 name: Firebase CI
2
3 on:
4   push:
5     branches:
6       - main
7
8 jobs:
9   build:
10    runs-on: ubuntu-latest
11
12    strategy:
13      matrix:
14        node-version:
15          - '15.x'
16
17    steps:
18      - uses: actions/checkout@v3
19      - name: Use Node.js version ${matrix.node-version}
20        uses: actions/setup-node@v3
21        with:
```

Step 8: Generating Token from windows powershell

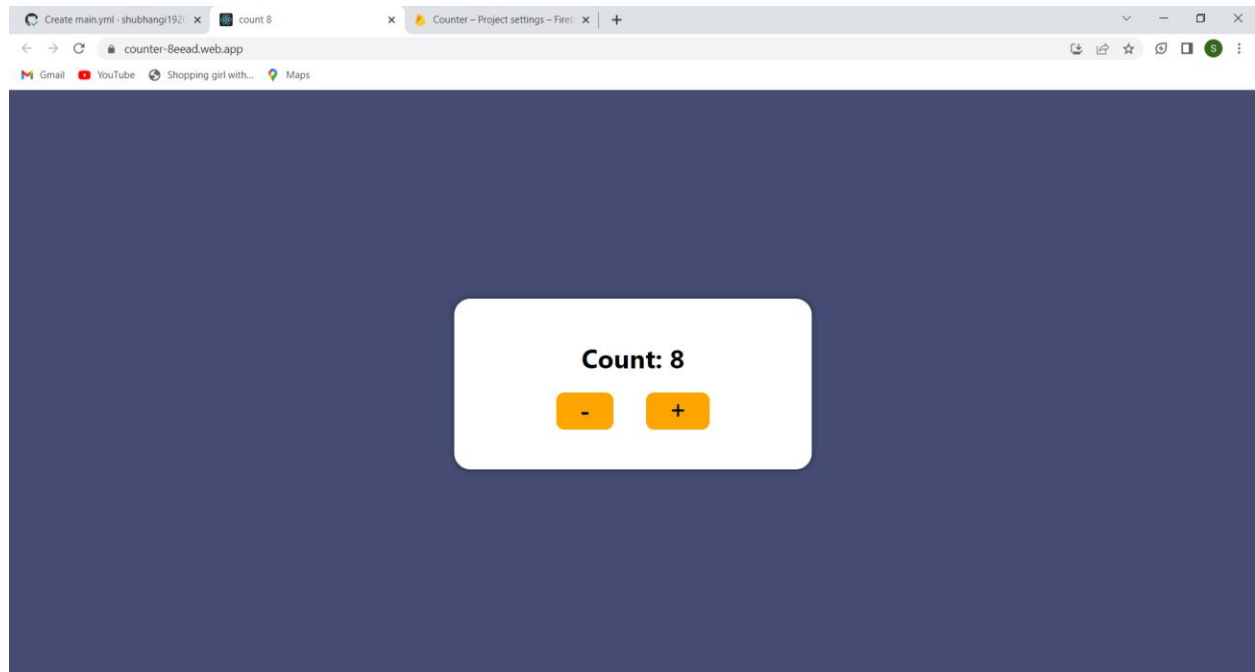


Step 10: Build & Deploy from 'Github Action'



Deployed Website

Hosting URL: <https://counter-8eead.web.app/>



Conclusion

- We have successfully deploy React CI/CD pipeline with GitHub Actions and Firebase which provide the efficient way for build, test and deploy react applications. firebase provides a platform for hosting and deploying the application. By building up a CI/CD pipeline, developers can ensure the stability and reliability of their applications while also increasing development speed and efficiency.