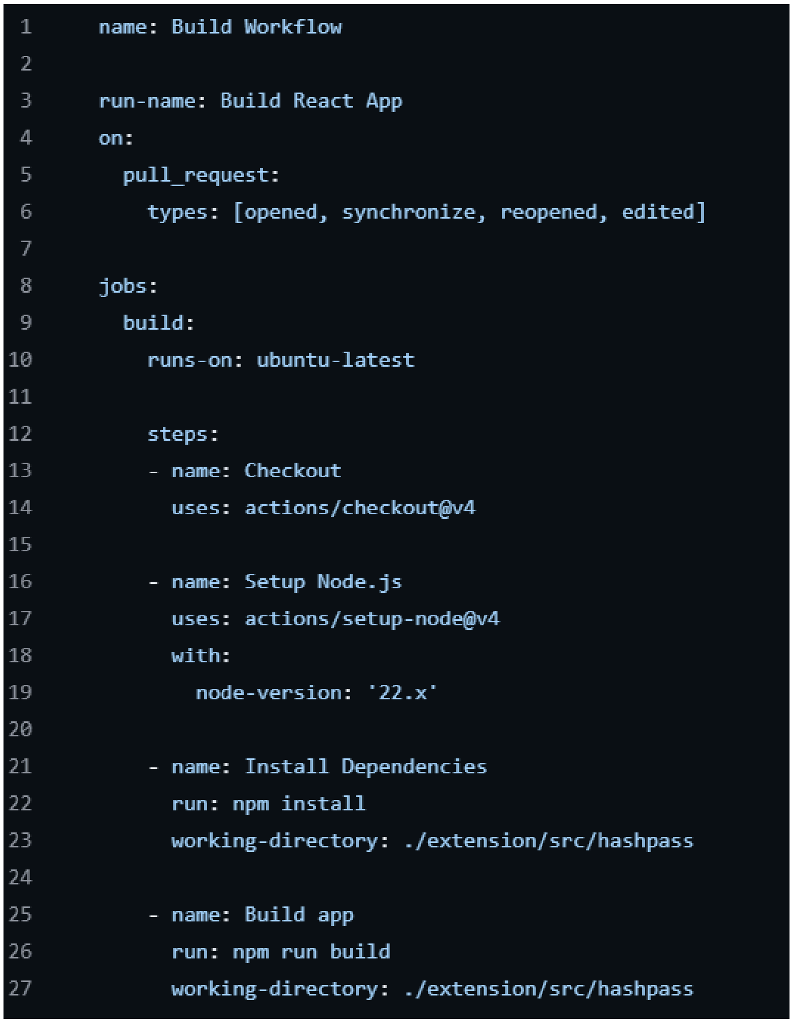
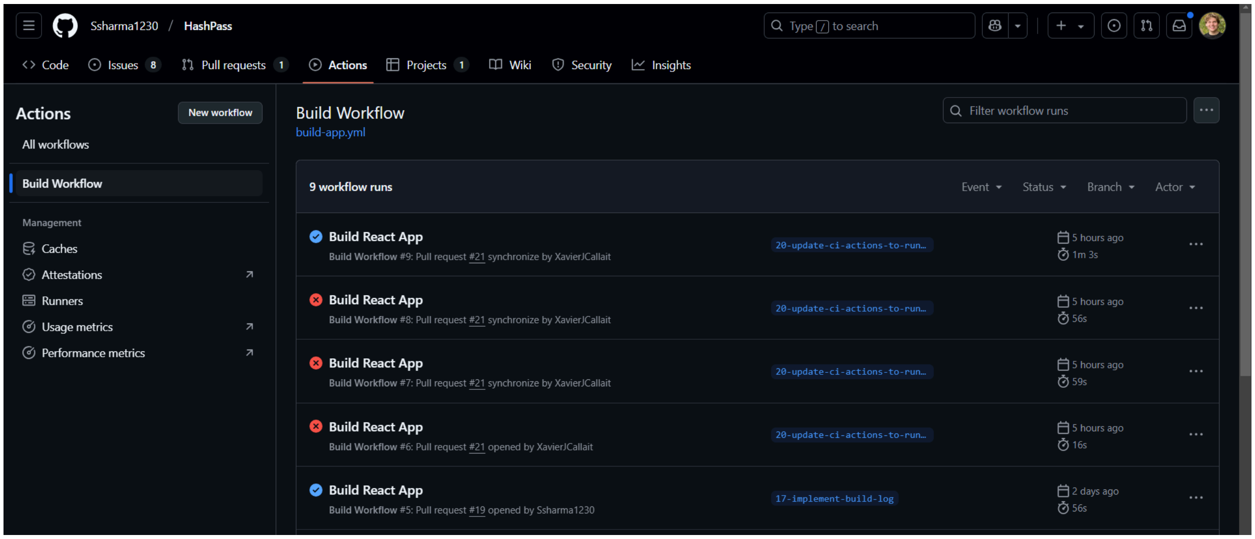
1. For our CI workflow, we decided to use GitHub workflow actions as it allows us to automatically verify and build our project whenever our codebase changes without needing third party software. Our CI builds our react application every time it is run. The trigger events are based on pull requests: each time there is a new pull request, a change to the base branch or to the actual branch of the pull request, or a pull request is closed. The idea is that we want to test our code before we merge, but we felt that if we set it to run everytime we push code, it might run unnecessarily during development. However, by having it run on pull requests, it ensures that it runs before it is merged, and allows for reviewers to verify the code. Below is a link to the workflow yaml file, and some runs of the yaml file. The job stated in the yaml file runs on a virtual machine with the latest ubuntu version. We checkout our repository and install node.js. From there we run npm commands to build our project and also run our test suite. In our package.json, we specify all of the commands and the dependencies that are required for our project to build and be tested. A picture of a our CI workflows along with the code is shown below.



```yaml
name: Build Workflow

run-name: Build React App
on:
  pull_request:
    types: [opened, synchronize, reopened, edited]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout
      uses: actions/checkout@v4

    - name: Setup Node.js
      uses: actions/setup-node@v4
      with:
        node-version: '22.x'

    - name: Install Dependencies
      run: npm install
      working-directory: ./extension/src/hashpass

    - name: Build app
      run: npm run build
      working-directory: ./extension/src/hashpass
```



Build Workflow
build-app.yml

9 workflow runs

Build React App
Build Workflow #9: Pull request #21 synchronize by XavierJCallait
20-update-ci-actions-to-run…
5 hours ago
1m 3s

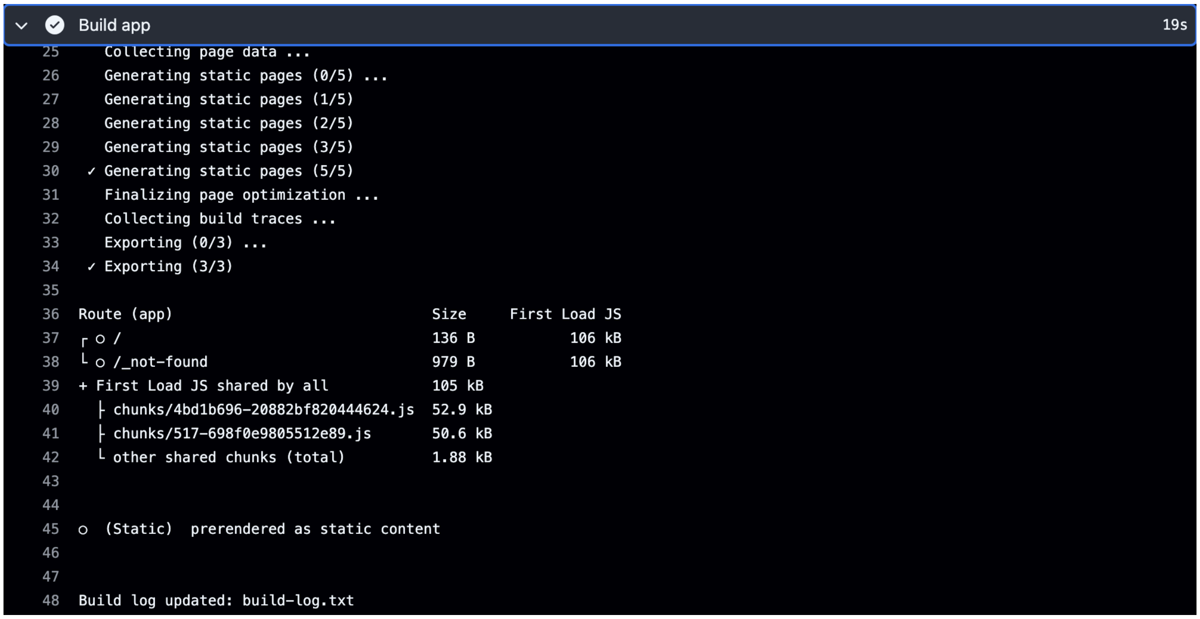
Build React App
Build Workflow #8: Pull request #21 synchronize by XavierJCallait
20-update-ci-actions-to-run…
5 hours ago
56s

Build React App
Build Workflow #7: Pull request #21 synchronize by XavierJCallait
20-update-ci-actions-to-run…
5 hours ago
59s

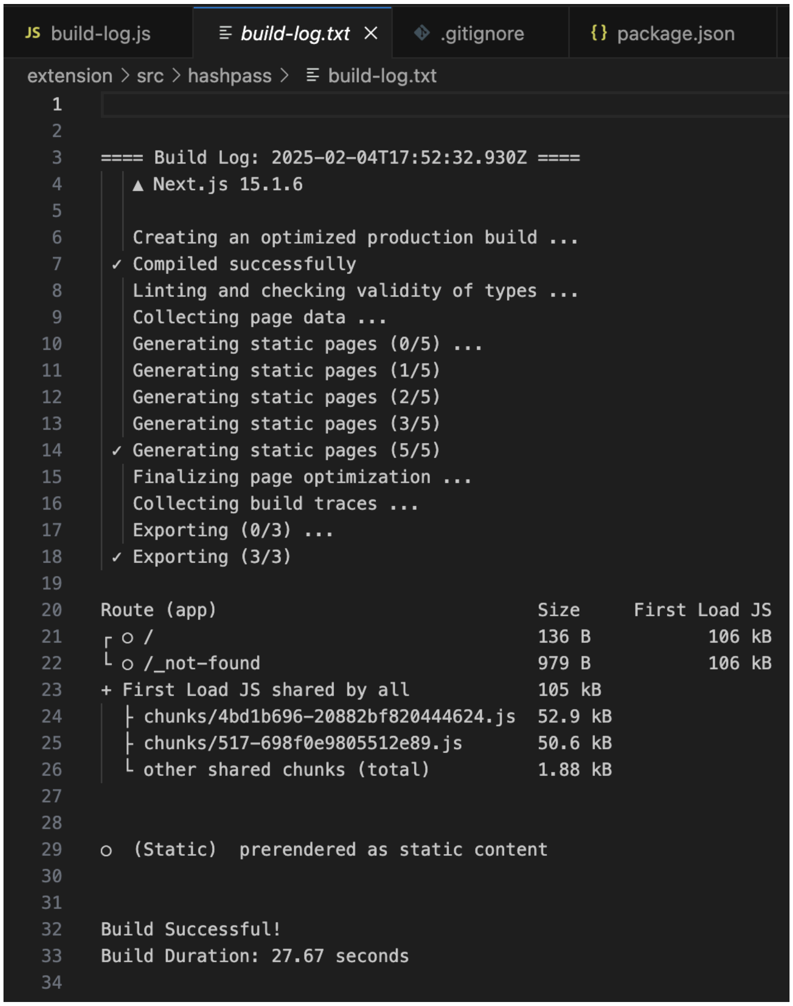
Build React App
Build Workflow #6: Pull request #21 opened by XavierJCallait
20-update-ci-actions-to-run…
5 hours ago
16s

Build React App
Build Workflow #5: Pull request #19 opened by Ssharma1230
17-implement-build-log
2 days ago
56s

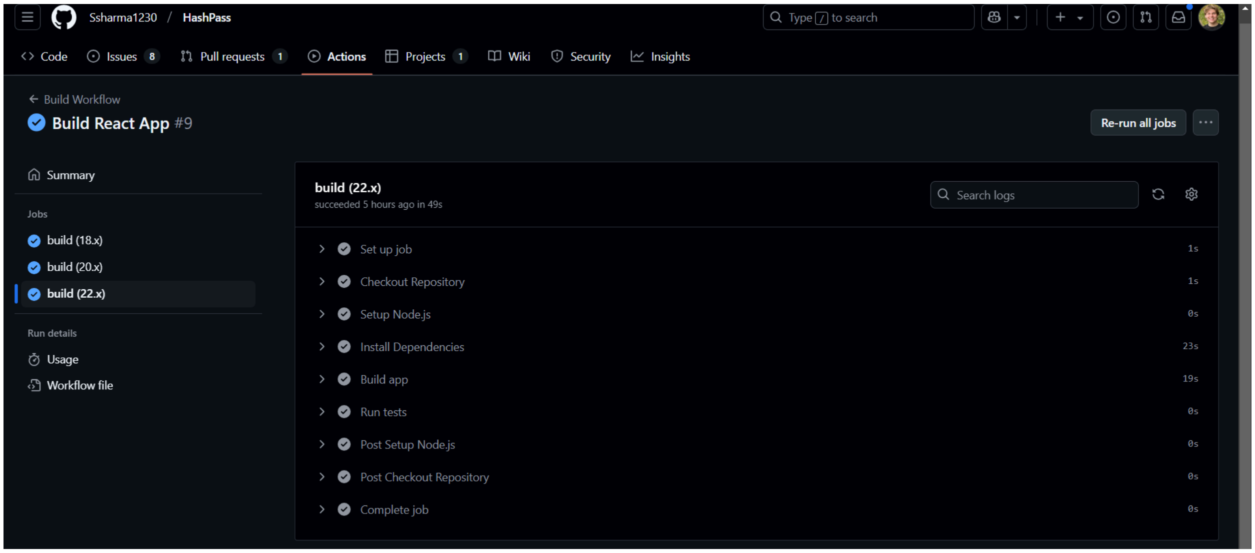
2. The build log is implemented to output the status and the metadata of the build to a file called “build-log.txt”. This file, however, is part of the .gitignore file we have to ensure the repository stays cleaner and will not have bigger files. The screenshot below is from the successful build where it shows the log file has been updated on line 48. The screenshot under that is what the log file shows after a successful build.



```
25     Collecting page data ...
26     Generating static pages (0/5) ...
27     Generating static pages (1/5)
28     Generating static pages (2/5)
29     Generating static pages (3/5)
30   ✓ Generating static pages (5/5)
31     Finalizing page optimization ...
32     Collecting build traces ...
33     Exporting (0/3) ...
34   ✓ Exporting (3/3)
35
36  Route (app)                              Size     First Load JS
37  ┌ ○ /                                    136 B          106 kB
38  └ ○ /_not-found                          979 B          106 kB
39  + First Load JS shared by all            105 kB
40    ├ chunks/4bd1b696-20882bf820444624.js  52.9 kB
41    ├ chunks/517-698f0e9805512e89.js       50.6 kB
42    └ other shared chunks (total)          1.88 kB
43
44
45  ○  (Static)  prerendered as static content
46
47
48  Build log updated: build-log.txt
```



```
==== Build Log: 2025-02-04T17:52:32.930Z ====
  ▲ Next.js 15.1.6

  Creating an optimized production build ...
✓ Compiled successfully
  Linting and checking validity of types ...
  Collecting page data ...
  Generating static pages (0/5) ...
  Generating static pages (1/5)
  Generating static pages (2/5)
  Generating static pages (3/5)
✓ Generating static pages (5/5)
  Finalizing page optimization ...
  Collecting build traces ...
  Exporting (0/3) ...
✓ Exporting (3/3)

Route (app)                              Size     First Load JS
┌ ○ /                                    136 B          106 kB
└ ○ /_not-found                          979 B          106 kB
+ First Load JS shared by all            105 kB
  ├ chunks/4bd1b696-20882bf820444624.js  52.9 kB
  ├ chunks/517-698f0e9805512e89.js       50.6 kB
  └ other shared chunks (total)          1.88 kB


○  (Static)  prerendered as static content


Build Successful!
Build Duration: 27.67 seconds

```



Build React App #9

build (18.x)
build (20.x)
build (22.x)

build (22.x)
succeeded 5 hours ago in 49s

```
Set up job                 1s
Checkout Repository        1s
Setup Node.js              0s
Install Dependencies      23s
Build app                 19s
Run tests                  0s
Post Setup Node.js         0s
Post Checkout Repository   0s
Complete job               0s
```

Tested success: https://github.com/Ssharma1230/HashPass/actions/runs/13171019603
Tested failure: https://github.com/Ssharma1230/HashPass/actions/runs/13170981772

3. As we use Javascript and Typescript as part of a React app, when we run ``npm run build`` it creates a build directory. We decided to zip up this directory as that is the way the code is uploaded when creating a Google Chrome extension. We ran the build command twice and zipped up each build directory upon their respective build. These were uploaded to the diffoscope tool which had the outputs attached to this submission. The differences were either from the zip file having different time stamps or the builds had different SHA 256 hashes when creating the names of the build files themselves. The elements and source code remained the same in the executables.