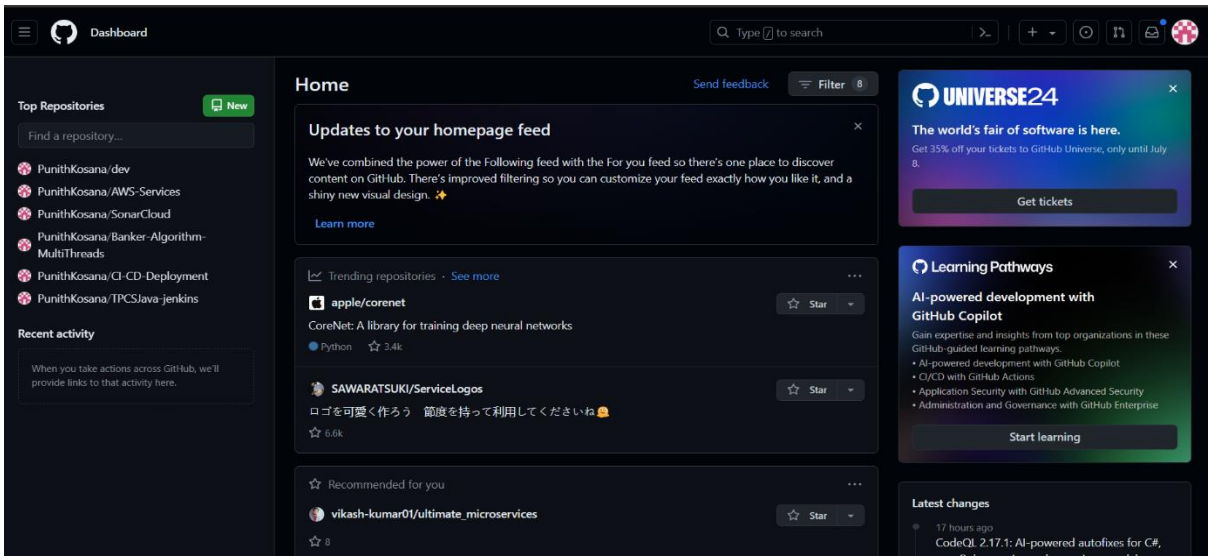
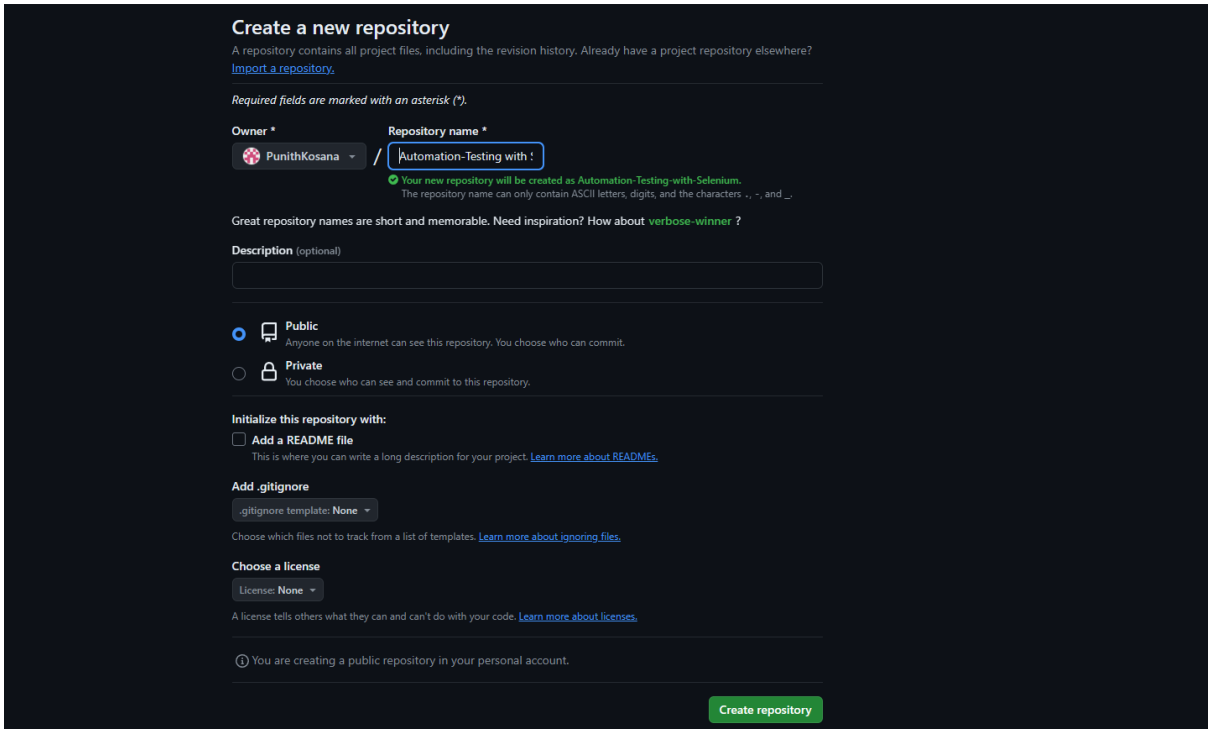
Creating Github Actions Pipeline involving usage of Selenium-Python Automation Testing

Open github.com and login into it. Choose New to create new repository.



Name the repository and make it public. Choose Create Repository.



Install selenium.

```
C:\Windows\System32\cmd.e

C:\Users\Punith>pip install selenium
```

This is the website we are going to deploy with github actions.

```
index.html
index.html > html > body > p
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Demo</title>
7   </head>
8   <body>
9       <h1>Welcome to my website</h1>
10      <p>Automation Testing with selenium</p>
11  </body>
12  </html>
13
14
```

This is how website looks like.

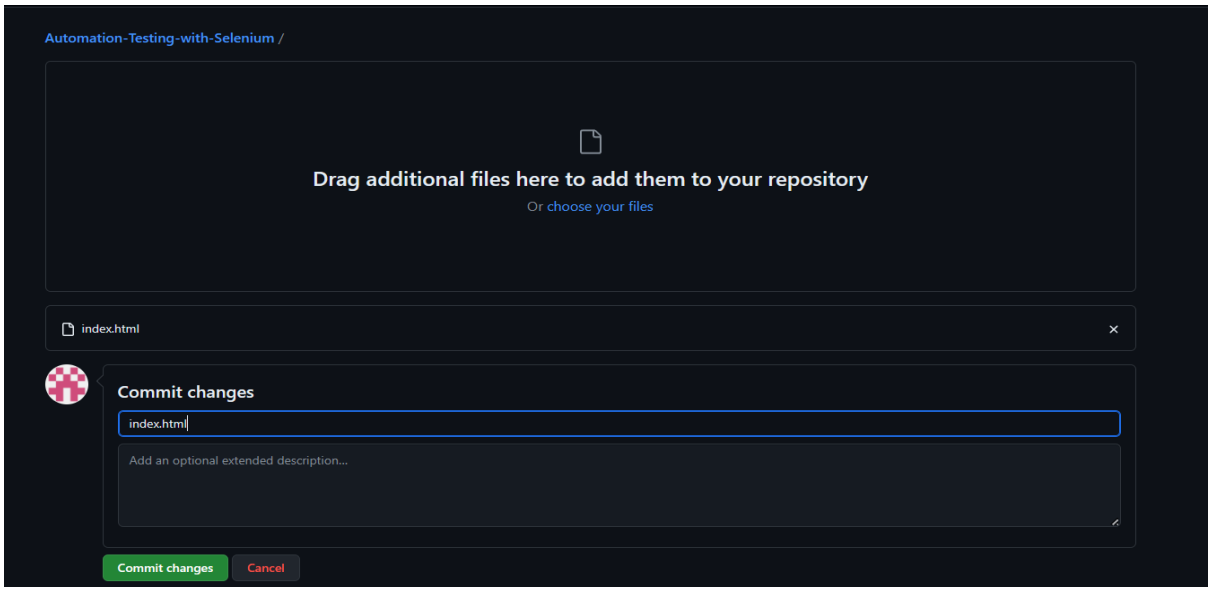
Demo

File C:/Users/Punith/OneDrive/Desktop/Selenium%20with%20Github/index.html

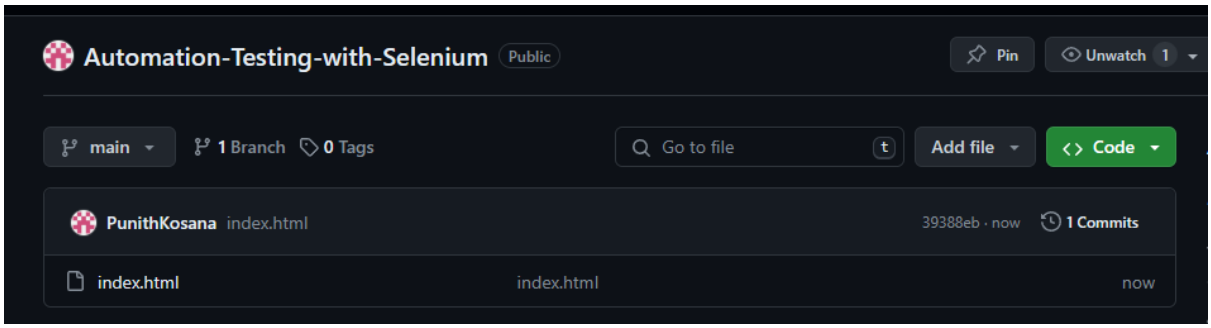
Welcome to my website

Automation Testing with selenium

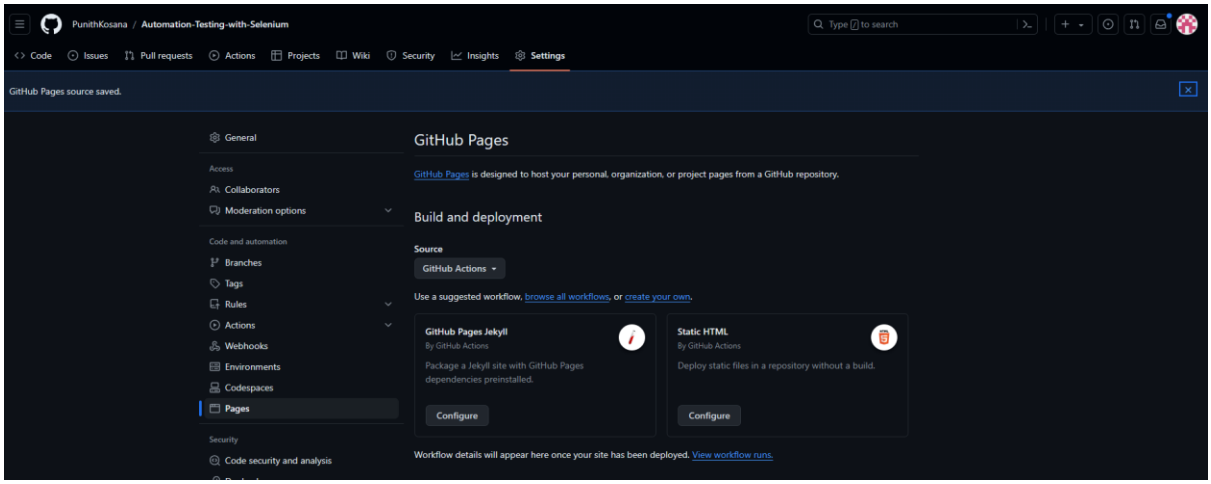
Add the file into repo and commit changes.



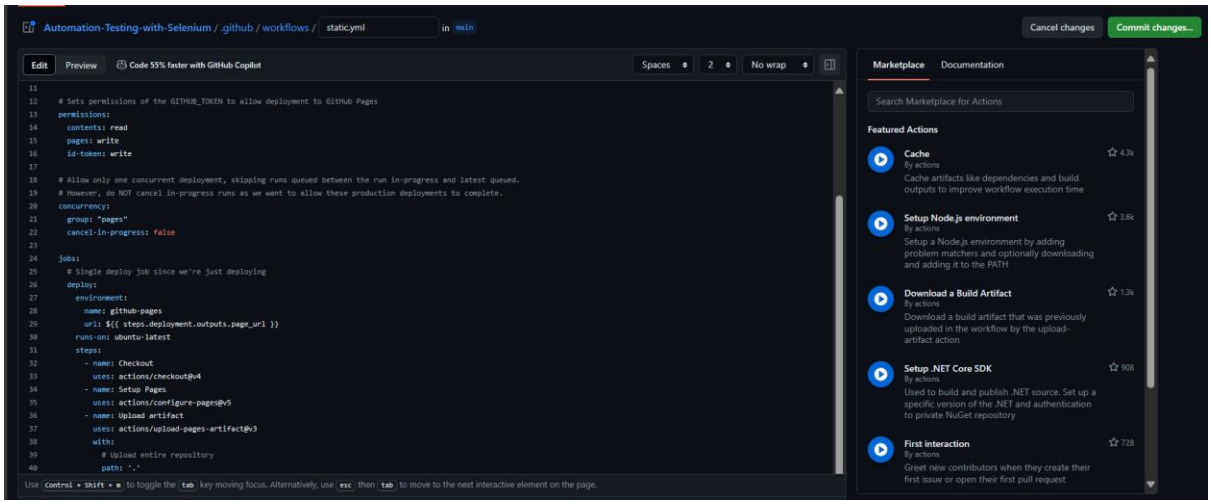
File has been successfully pushed into the repository.



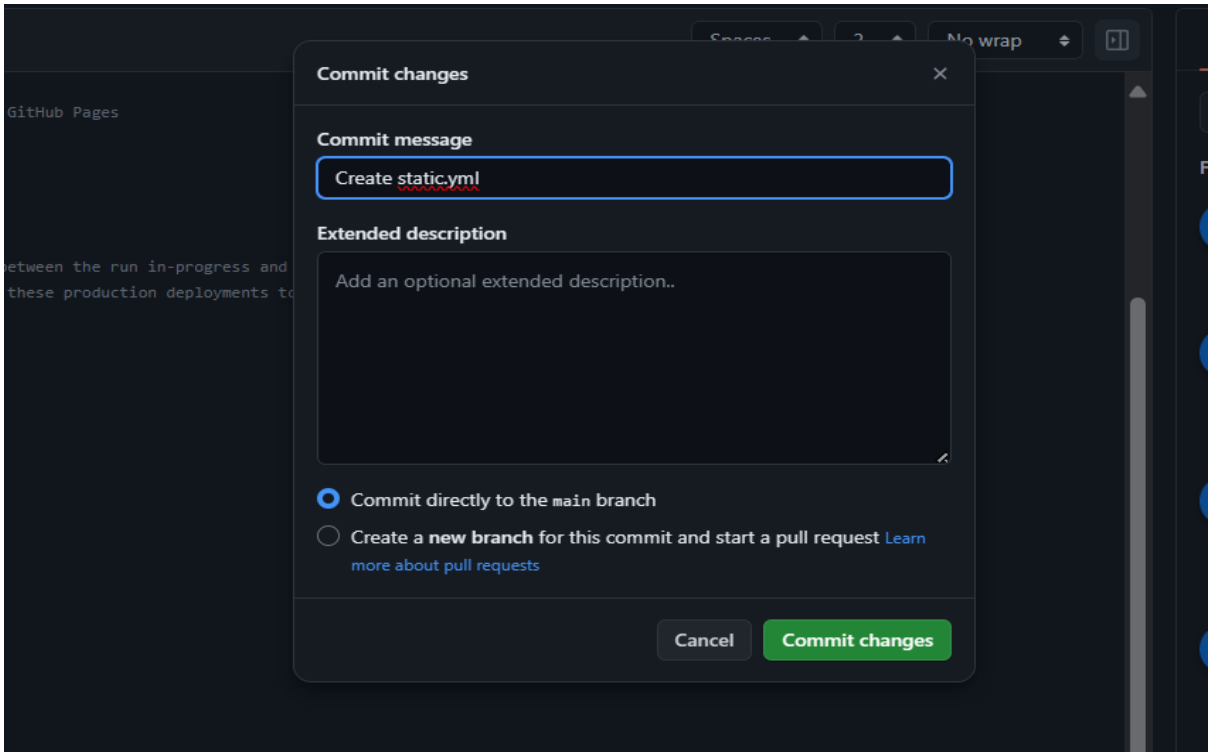
Go to Settings>Github Pages. Configure Static HTML. Make sure to choose the source as Github Actions.



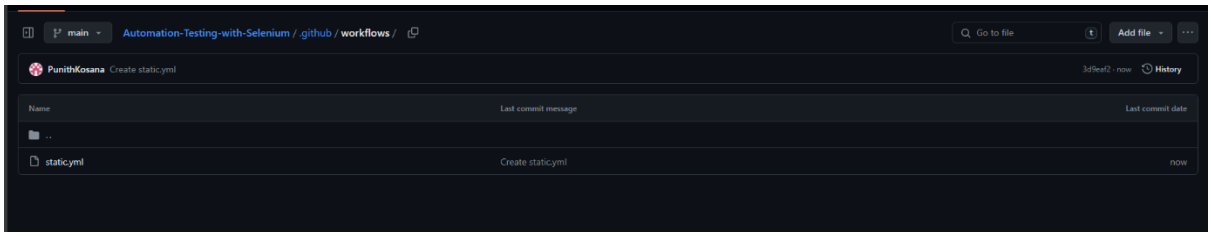
This is the code used to deploy onto the github pages.



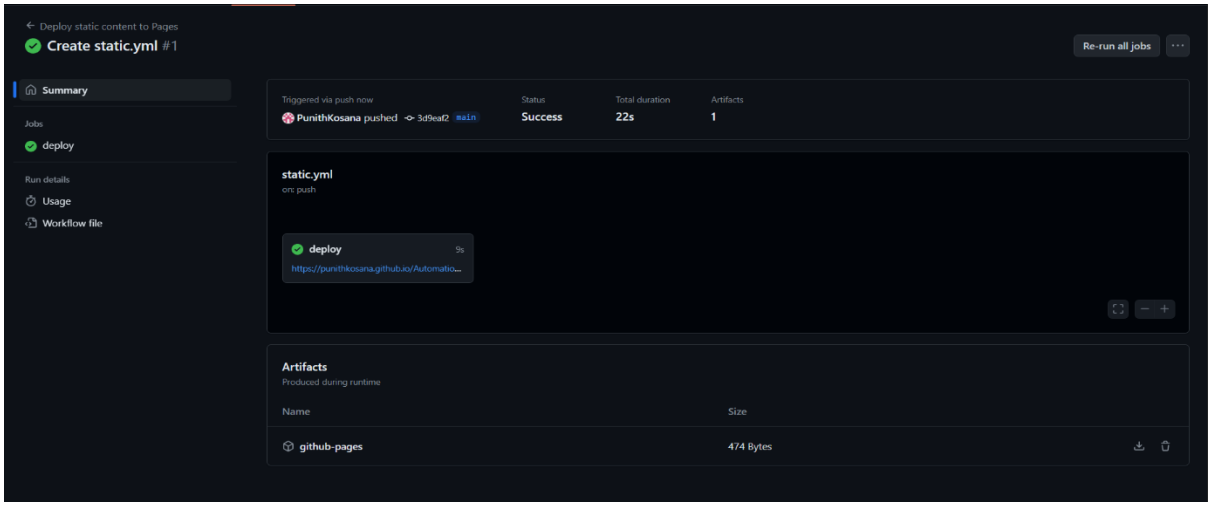
Commit the changes.



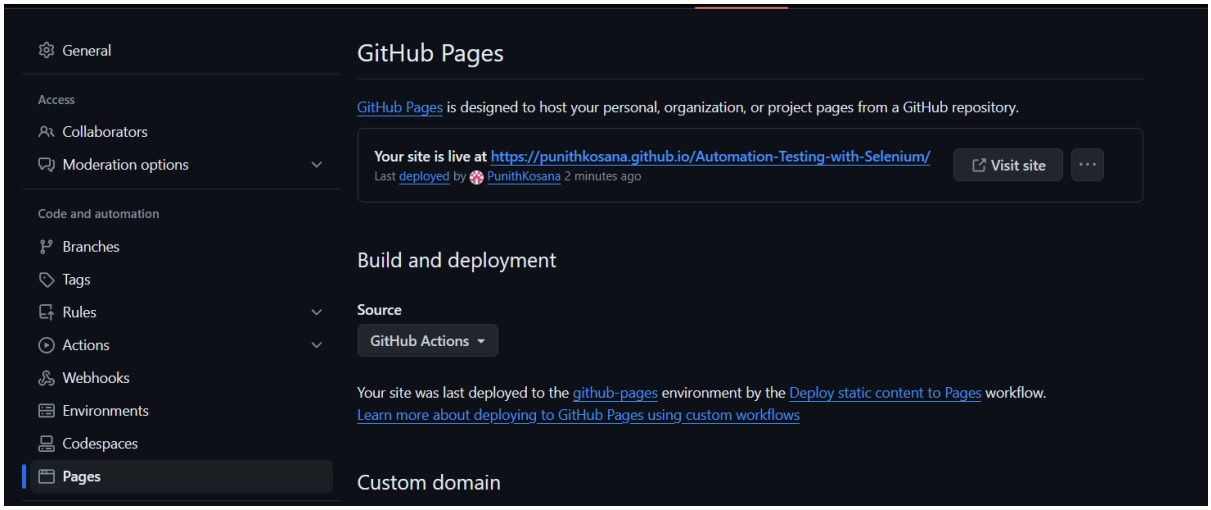
The github workflow has been created successfully.



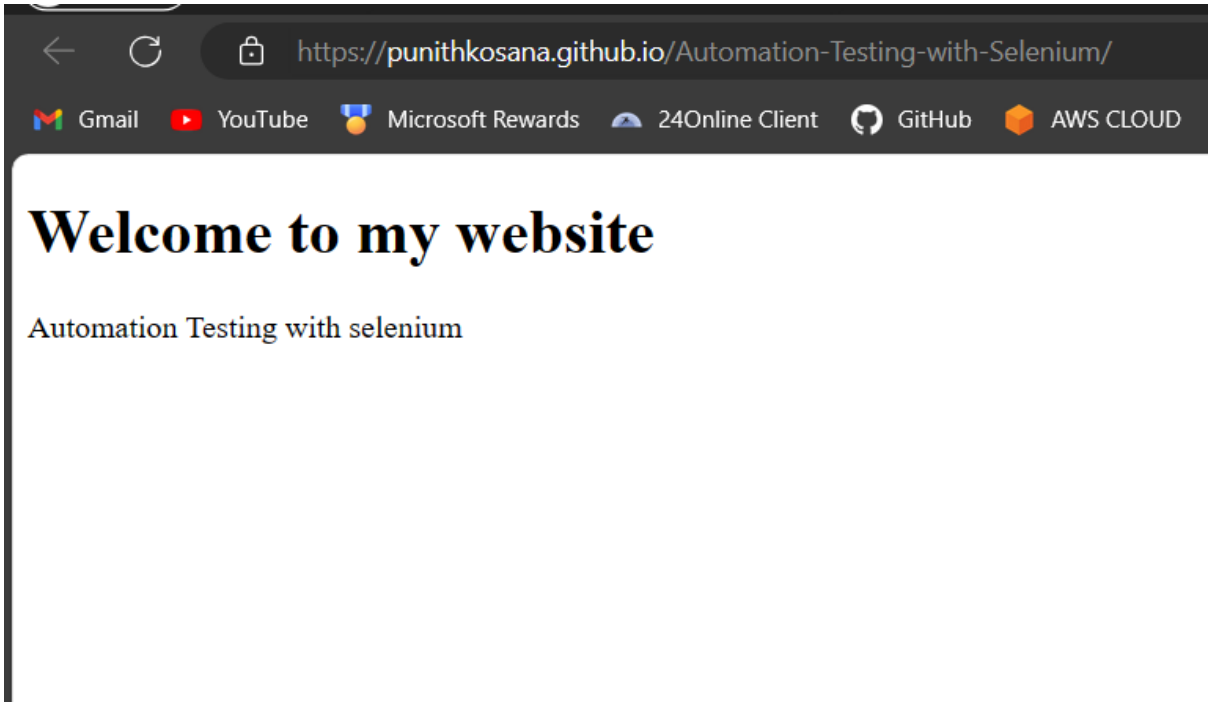
Choose static.yml and click view runs. The deployment has been successful.



Open Github Pages and visit the site.



The website have been deployed.



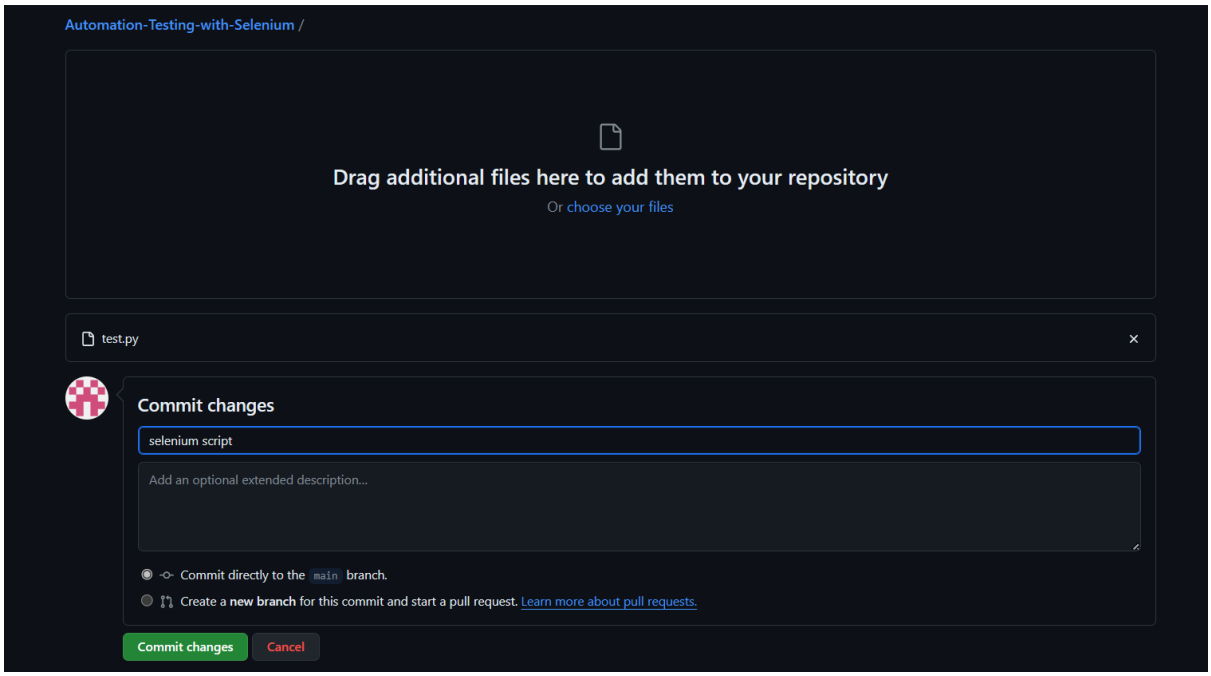
This is the python-selenium script used to automate the process. Whenever the changes made in the repo, the scripts executes and the screenshot of the website will be pushed into the repository automatically.

test.py

test.py > ...

```python
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
import time

# Start the WebDriver and open the HTML page
service = Service(executable_path='/usr/local/bin/chromedriver')
options = webdriver.ChromeOptions()
options.add_argument("--headless")
options.add_argument('--no-sandbox')
driver = webdriver.Chrome(service=service, options=options)

driver.get("https://punithkosana.github.io/Automation-Testing-with-Selenium/")

time.sleep(5)  # Adding a delay to see the result

# Assert some condition to verify the result
assert "" in driver.title

# Take a screenshot
timestamp = time.strftime("%Y%m%d-%H%M%S")
screenshot_file = f"screenshot_{timestamp}.png"
driver.save_screenshot(screenshot_file)

# Close the WebDriver
driver.close()
```

Upload the python file and commit changes.

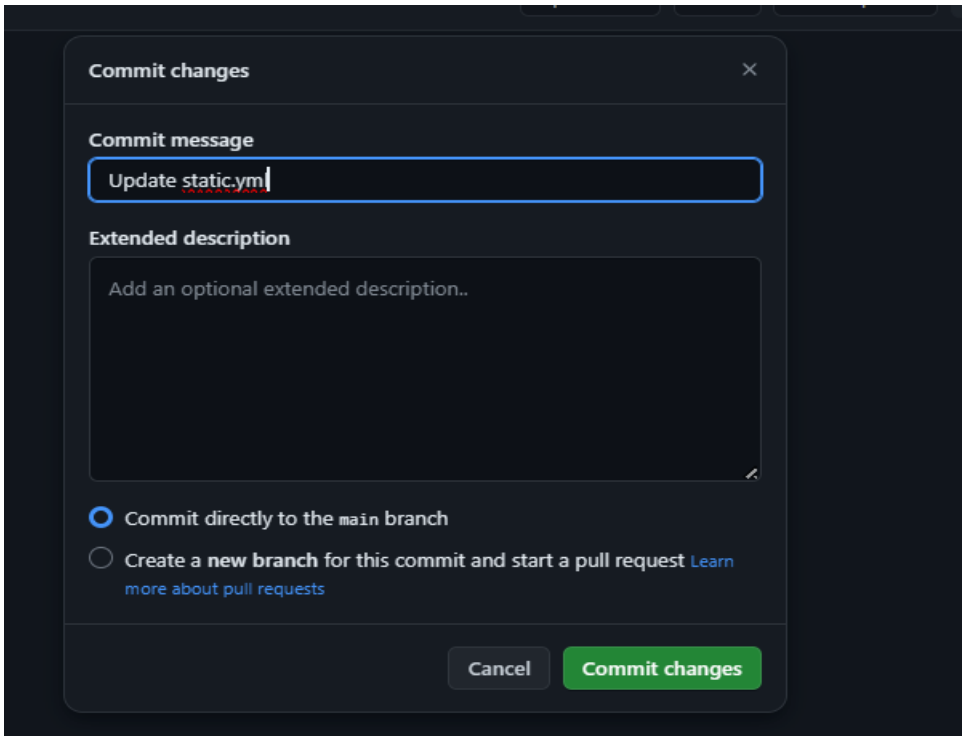


Add these steps that has to perform and test at the time of deployment in the static.yml file.

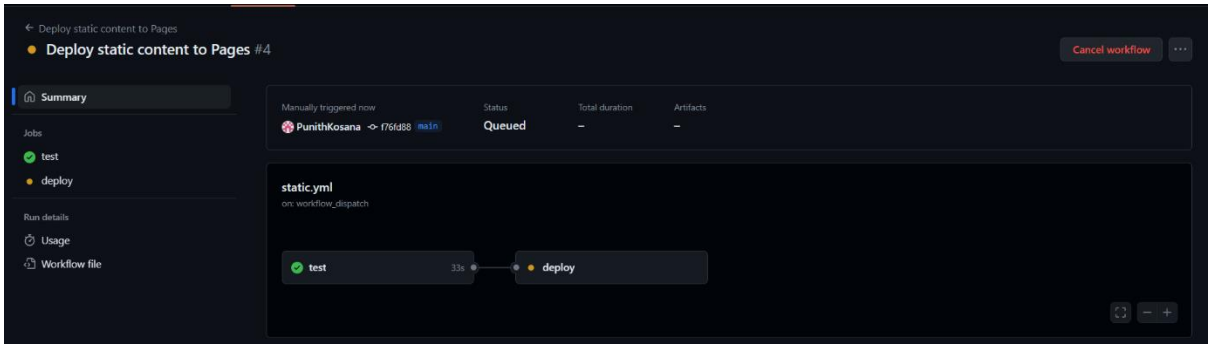
```yaml
    # Steps to execute within the job
    steps:
      # Step to checkout the repository code
      - name: Checkout Repository
        uses: actions/checkout@v2

      # Step to set up Python environment
      - name: Set up Python
        uses: actions/setup-python@v2
        with:
          python-version: "3.12"  # Version of Python to set up

      # Step to install Chrome WebDriver for Selenium
      - name: Install Chrome WebDriver
        run: |
          LATEST=$(wget -q -O - https://googlechromelabs.github.io/chrome-for-testing/LATEST_RELEASE_STABLE)
          wget https://storage.googleapis.com/chrome-for-testing-public/$LATEST/linux64/chromedriver-linux64.zip
          unzip chromedriver-linux64.zip
          cd chromedriver-linux64
          sudo mv chromedriver /usr/local/bin/
          sudo chmod +x /usr/local/bin/chromedriver

      # Step to install Selenium library
      - name: Install Selenium
        run: pip install selenium

      # Step to execute Selenium tests
      - name: Run Selenium tests
        run: python test.py


      - name: Save screenshot to repository
```

```yaml
      # Step to install Selenium library
      - name: Install Selenium
        run: pip install selenium

      # Step to execute Selenium tests
      - name: Run Selenium tests
        run: python test.py


      - name: Save screenshot to repository
        if: ${{ success() }}  # Condition to run only if tests succeed
        run: |
          git config --global user.email "actions@github.com"
          git config --global user.name "GitHub Actions"
          git add screenshot_*.png
          git commit -m "Screenshot taken successfully"
          git push
  deploy:
    needs: test
    environment:
      name: github-pages
      url: ${{ steps.deployment.outputs.page_url }}
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: Setup Pages
        uses: actions/configure-pages@v5
      - name: Upload artifact
        uses: actions/upload-pages-artifact@v3
        with:
          # Upload entire repository
          path: '.'
      - name: Deploy to GitHub Pages
        id: deployment
        uses: actions/deploy-pages@v4
```

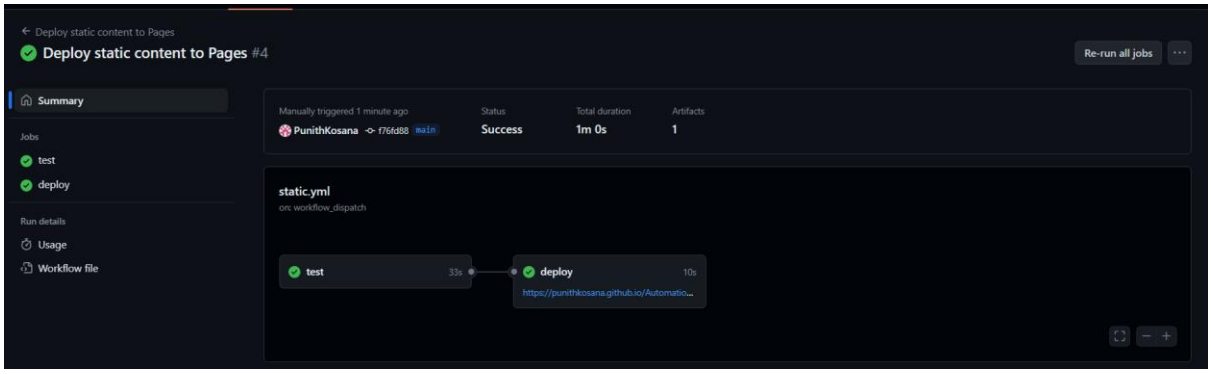
Commit the changes.



The test has been executed successfully.



The deployment also done successfully.



The screenshot has been taken and pushed into the repository.