

Documentation generation:

We used JSDocs in order to generate documentation. On a pull request, documentation for index.js is generated into the source directory, which we can merge into the main branch.

doc.yml:

```
name: doc
on:
  pull_request:
    branches:
      - main
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v2
        with:
          ref: ${github.event.pull_request.head.ref}
      - name: Set up Node.js
        uses: actions/setup-node@v1
        with:
          node-version: 12
      - name: Install JSDocs
        run: npm install -g jsdoc
      - name: run JSDoc
        run: jsdoc ./timer/index.js -d ./source
      - name: add files
        run: git add .
      - name: Commit files
        run: |
          git config --local user.email "41898282+github-actions[bot]@users.noreply.github.com"
          git config --local user.name "github-actions[bot]"
          git commit -m "Add changes" -a
          git push
```

Styling and Syntax:

We used ESLint to detect if there are any js style issues and stylelint to detect if there are any css style issues. On a pull request, it will detect the documents with either .js or .css type in the timer folder and use those lint to detect errors.

lint.yml:

```
name: Lint
on:
  pull_request:
```

```

    branches: main
jobs:
  run-linters:
    name: Run linters
    runs-on: ubuntu-latest
    steps:
      - name: Check out Git repository
        uses: actions/checkout@v2
      - name: Install ESLint
        run: npm install --save-dev eslint-config-standard eslint-plugin-promise eslint-plugin-import
  eslint-plugin-node
    name: install stylelint
    run: npm install --save-dev stylelint stylelint-config-standard
  - name: ESLint
    run: npx eslint --init
  - name: Run ESLint
    run: npx eslint "timer/*.js"
  - name: run stylelint
    run: npx stylelint "timer/css/*.css"

```

Testing:

We used Jest to do our unit testing. When a pull request happens, all tests in files that end in .test.js are run and the results are shown in Actions as well as the pull request page.

test.yml:

```

name: test
on:
  pull_request:
    branches:
      - main
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v2
      - name: Install jest
        run: npm install --save-dev jest
      - name: Run jest
        run: npm test

```