**Continuous Integration Report(CIR)**

**Framework: Django**

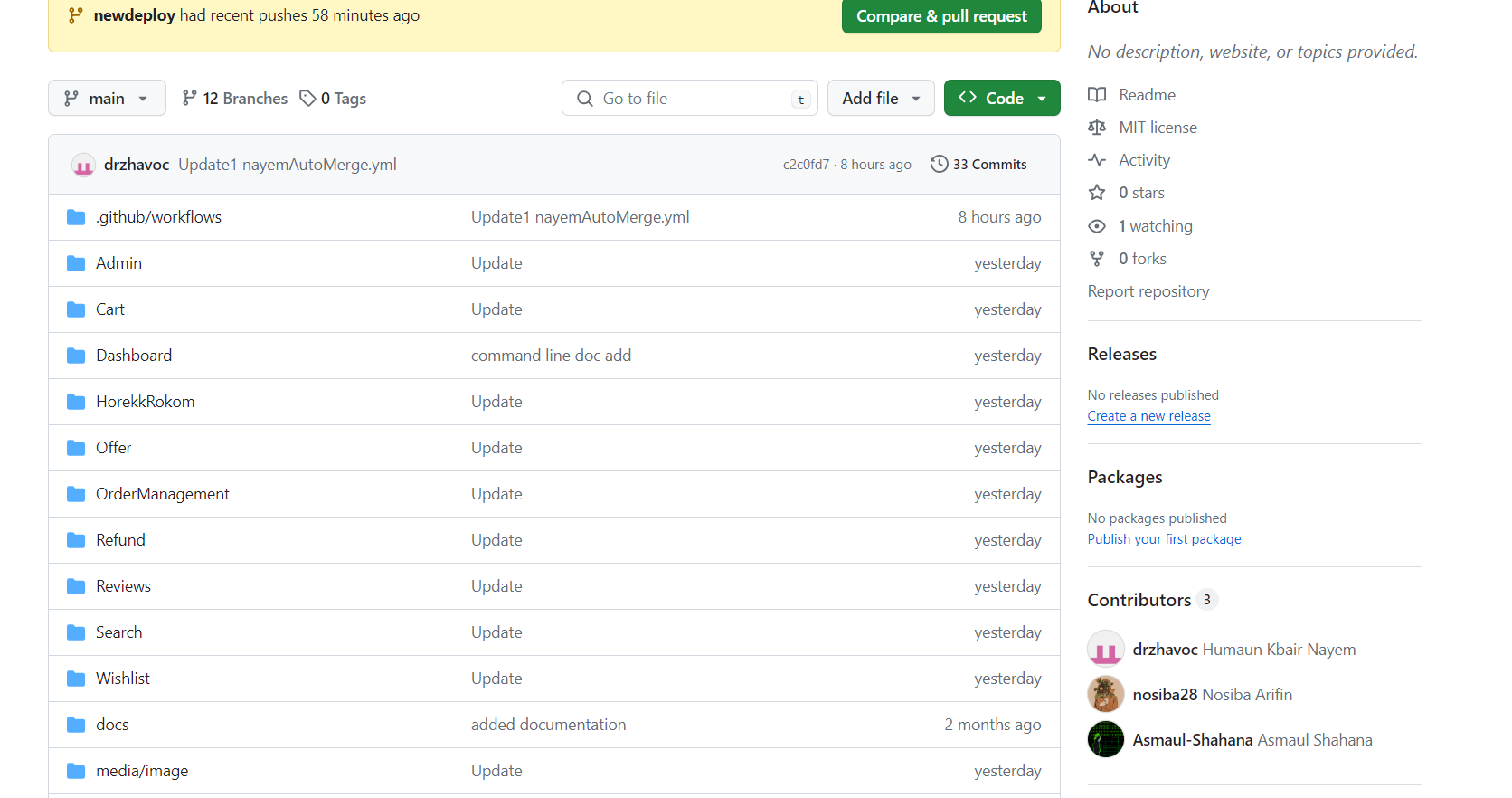
**CI Tool:GitHub**

**Introduction:**   
GitHub CI is a crucial aspect of modern software development that leverages automation to streamline the process of delivering code changes from development to production environments. It's an integral part of the GitHub platform, providing developers with a robust set of tools and workflows to automate testing, build, and deployment processes.

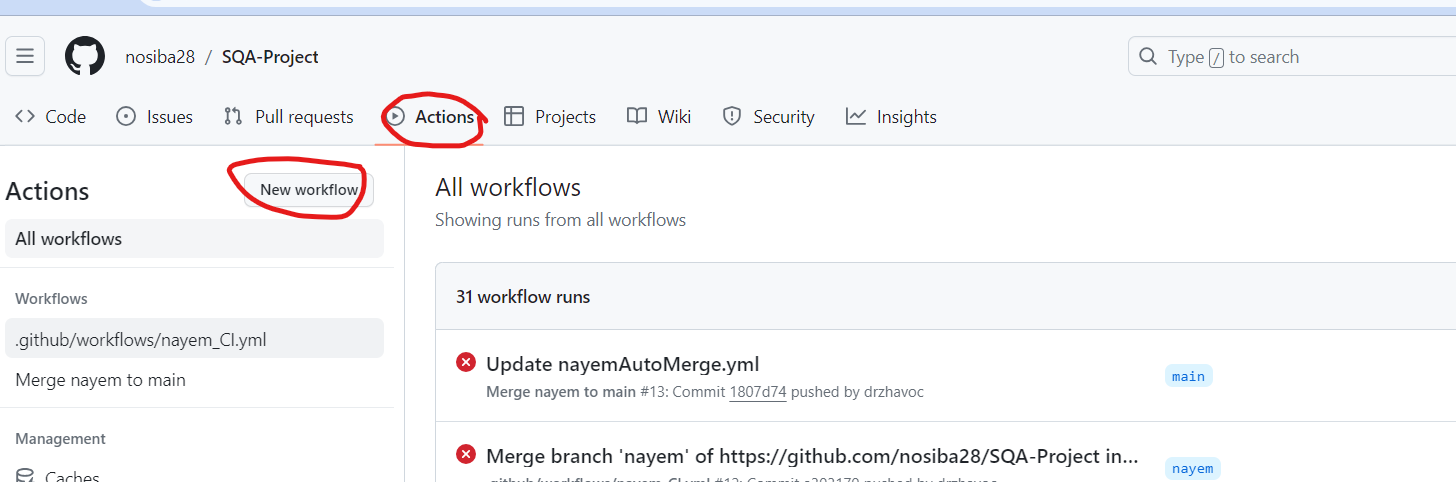
There are several tools that can be used like Jenkins, CircleCI, JitLab, GitHub Actions etc. As we used GitHub and It is very much a neighbour feature, That’s why we selected GutHub actions as our CI tool.

Working Procedure:

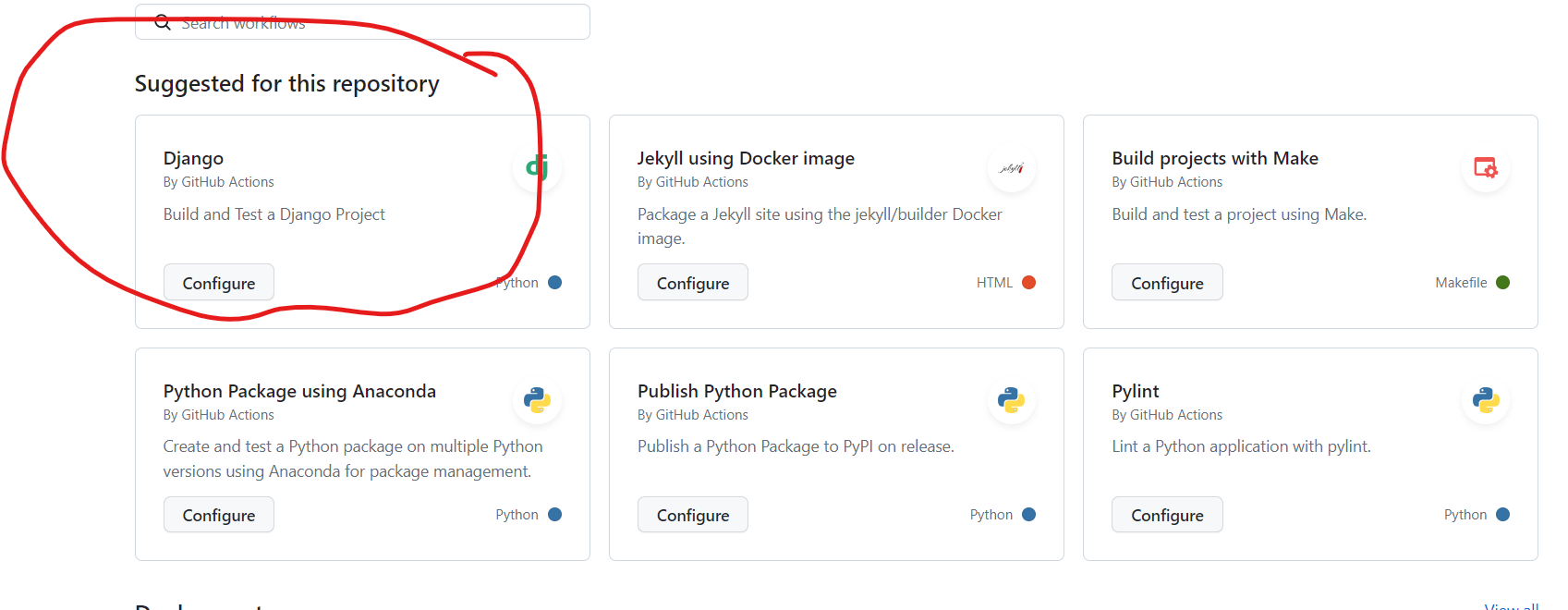
1.Merging files to all main branch



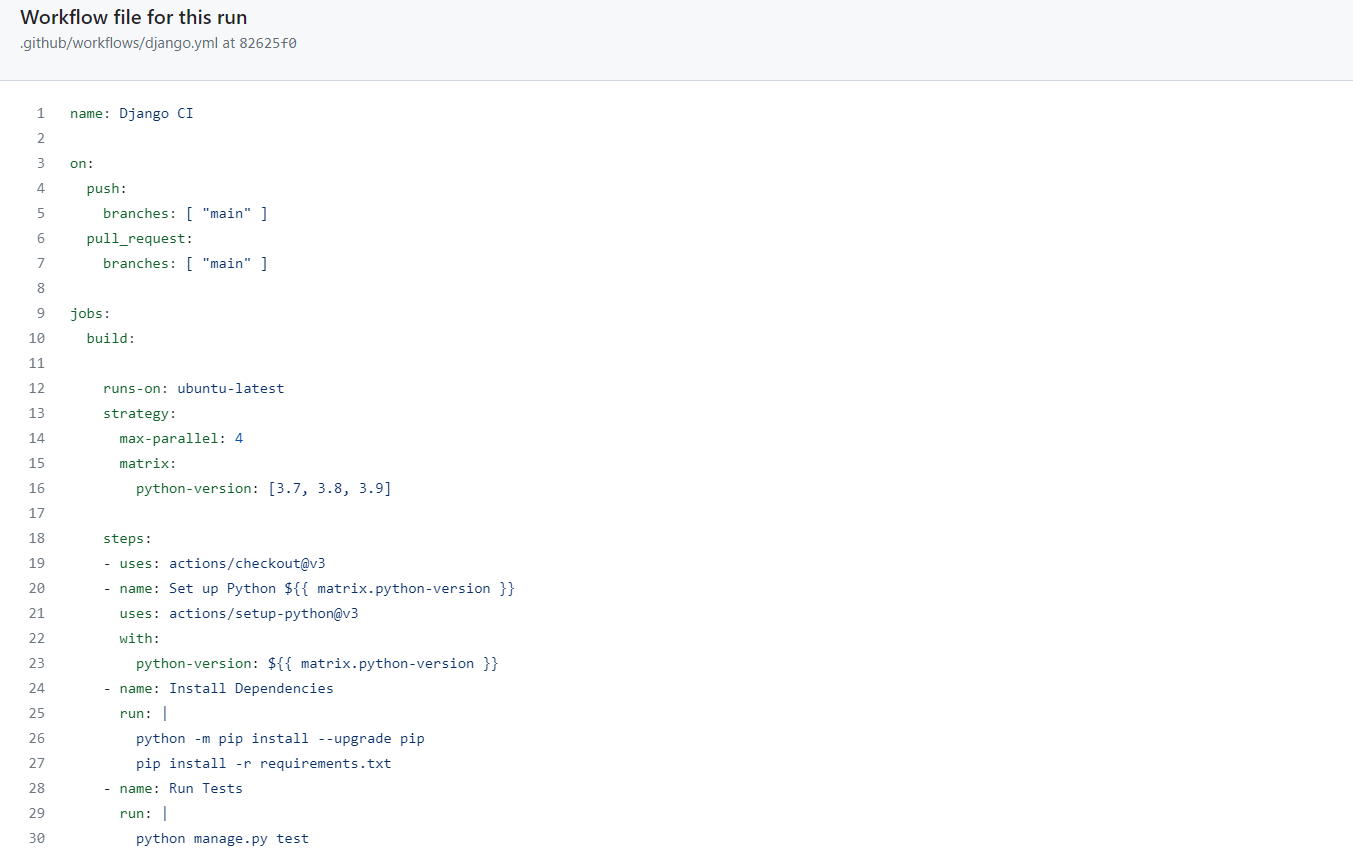
**2.** [**Go to the “GitHub Actions” option and create a “new wrokflow”**](https://github.com/EmmanuelEmu/CSE-455_SQA/actions)

****

**3.** [**Choose a frame work according to our project for CI, “Python application-Django”**](https://github.com/EmmanuelEmu/CSE-455_SQA/actions/new)



**5.** [**Update Django .yml file belongs to in “.github/workflows” in main repo on Github path for commiting the CI test and build**](https://github.com/EmmanuelEmu/CSE-455_SQA/tree/main/.github/workflows)



**6.**[**Build States execution**](https://github.com/EmmanuelEmu/CSE-455_SQA/actions/runs/8051658336/job/21990047715)

**6.1 Accept phases:**

1. Set up Job
2. Run actions/checkout@v3
3. Set up Python 3.7
4. Install dependencies
5. Post Run actions/checkout@v3
6. Complete job

**6.2 Error phase:**

1. Run test

