

CI/CD Practices and Post deployment

- CI are used to run tests on each commit, ensuring that the code is always in a deployable state. This helps catch issues early and ensures that new changes don't break existing functionality. I already added a CI script that runs tests and linter.
- End-to-end tests are run during the CI process to validate that the code works as expected. This helps maintain code quality and reduces the risk of bugs or issues making it to production (Cypress, Playwright).
- Adding continuous Deployment (CD) script - after successful integration and testing, the code is automatically deployed to production or staging environments. This is done using CI/CD pipelines that automate the process of deployment.
- Version control - ensuring that each deployment is associated with a version number. In case of issues, rollbacks are possible to revert to a previous stable version.
- Track application performance - for example integrate Grafana in the post deployment period.
- Create Jira tasks for every feature, improvement and bug. Also, it would be good to have all requirements, app descriptions and how it works documented in a Confluence. Later it will be easier to onboard new developers, manage releases, and understand the evolution of the solution over time.
- Add more tests in jest - testing each page, whether it opens properly, as well as interaction with the components on it (filters, pagination), adding, removing products, from cart, submitting the form.