CONTINUOUS INTEGRATION(CI)

Continuous integration is a coding philosophy and set of practices that drive development teams to implement small changes and check in code to version control repositories frequently. Because most modern applications require developing code in different platforms and tools, the team needs a mechanism to integrate and validate its changes.

The technical goal of CI is to establish a consistent and automated way to build, package, and test applications. With consistency in the integration process in place, teams are more likely to commit code changes more frequently, which leads to better collaboration and software quality.

CONTINUOUS DELIVERY(CD)

Continuous delivery picks up where continuous integration ends. CD automates the delivery of applications to selected infrastructure environments. Most teams work with multiple environments other than the production, such as development and testing environments, and CD ensures there is an automated way to push code changes to them.

Continuous integration and continuous delivery require continuous testing because the objective is to deliver quality applications and code to users. Continuous testing is often implemented as a set of automated regression, performance, and other tests that are executed in the CI/CD pipeline.

CAN WE USE IT?

Ofcourse we can use a CI/CD pipeline in our project. We are creating a dashboard so when we update something we should see its results instantaneously.

For frontend being able to see the preview of pull requests without having to locally run them makes development cycle a lot faster.

HOW WE USED IT?

We use CI/CD in our both backend and frontend.

For our frontend it checks format and the lint rules of the codebase runs tests, checks the test coverage and also creates a preview deploy for each pull request and deploys the main branch to production.

For backend it runs the tests for the project and also checks the test coverage and also we have manual trigger that deploys to production.

We use github actions for the pipelines, docker and docker compose for our backend and netlify for our frontend preview and production deployments.