

6. DevOps Implementation

Version Control Practices

Git Workflow Git is an open-source version control system that can be used between developers to help manage and track changes in their repositories. It has become the standard tool that developers use to manage software development projects ranging from small personal projects to huge projects in big organizations. The tool helps track changes to files over time, allows several developers to collaborate on the same project while ensuring that they don't accidentally overwrite each other's code, and helps provide a history of changes which developers can also use to revert to previous iterations of the same file if problems arise. Git is also being used in this project. Currently the project consists of three repositories. One for the frontend, backend and the report. In each of these repositories each group member has their own branch in which they can make changes without interfering with the other group members work. When a group member has then fully developed or written something, it can be merged into the dev branch.

GitHub Integration Github is an extension to the git version control system, that also serves as a hosting platform for git repositories. It comes with a lot of features that can aid the collaboration between developers to a great extent. For this project, a github organization has been made. That way the group can have their three repositories in a shared space. To each repository the group also has a work board, where tasks can be created such that each group member can see what needs to be done. This helps create an overview of the project status. When group members have developed or written something, they can create a pull request, "asking" for their branch to be merged into a different branch. It is essentially an invitation for the rest of the group to review what has been committed to the repository before merging it into the development branch. This is a way for developers to collaborate and catch errors before they reach a shared branch. Pull requests can also be used to setup continuous integration and continuous deployment.

Continuous Integration/Continuous Deployment

GitHub Actions Configuration

Build Server Setup

Testing Pipeline For the backend of the project, we're using a GitHub Actions workflow as a testing pipeline. The workflow is triggered on push events as well as pull request events into the dev and main branch. The **build** part of the workflow has steps included to set up the JDK, generating fake JWT keys and building the project using Maven. Following that there is a **test** job which

depends on the build job. This means that the test job will not run if the build job fails. If the build job has succeeded, It will download the build artifacts, set the JWT key environment variables, and run the tests for the backend using Maven. This pipeline runs, as said previously, every time a pull request is made, making it super useful, as it prevents any pull requests that do not either build or pass all the tests from being merged into the dev or main branch.

Containerization

Docker Implementation

Container Registry

Cloud Deployment

Google Cloud Setup

Netlify Frontend Deployment

Monitoring and Maintenance