

Group Project Assignment

[S25] System and Network Administration / Innopolis University

Team Members

Mikhail Panteleev	m.panteleev@innopolis.university	B23-SD-01
Asqar Arslanov	a.arslanov@innopolis.university	B23-SD-01

Project Goal

The project idea has been taken from the [RealWorld](#) project—a specification for a social blogging website used for (a) learning to build “real world”-size applications (b) demonstrating capabilities of new technologies.

To make this project, we’ve separated some of our responsibilities. Asqar’s responsibilities were to find a front end implementation and prepare a report and demo, and Mikhail was responsible for finding a back end implementation and deploying the application. We’ve both contributed to CI/CD and tried to apply best practices in our work.

The deliverables of our project include:

- A Git repository with all the source code.
- Build instructions.
- Dockerfiles and Compose files.
- An automated CI/CD pipeline with code linting.
- A hosted version of the application.

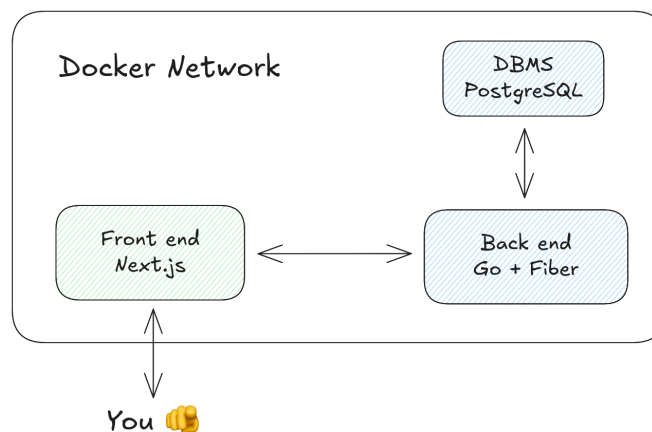
Methodology

The SNA course mainly focuses on DevOps practices to maintain and deploy software applications rather than the architectural side of building software with code. For this reason, we’ve decided not to reinvent the wheel and build upon an existing open source foundation.

The code for the back end has mostly been taken from [dashhhik/realworld-fiber-sqlc](#). The front end code originates from [yukicountry/realworld-nextjs-rsc](#). These implementations work well together (as all RealWorld implementations are supposed to).

The back end is written in Go with Fiber. It uses PostgreSQL as the DBMS. The front end is written in Next.js. It uses the pnpm package manager to make builds faster than with the “traditional” npm. Both services log incoming actions to the console.

The diagram below illustrates the main components of our application:



Development

We've managed to get everything to work with Docker, which makes it easier to launch our application without installing many dependencies on your computer. The project can either be launched using our shell script ([scripts/run-dev.sh](#)) or with the following command:

```
docker compose --file=compose-dev.yml up --build
```

In our CI/CD pipeline, we assess quality of our code using static analysis tools and upload images to DockerHub.

The back end code is analyzed with [golangci-lint](#) . Also, for code style consistency, everything is formatted with [gofumpt](#) . The front end uses [ESLint](#) for linting and avoiding common mistakes. The code is formatted with [Prettier](#) (default config).

In our [README file](#), we show badges that reflect statuses of the [workflows](#) we run to automate our CI/CD pipeline. These workflows re-run on every new push.

Discussion

Overall, we think we've succeeded in making this project. Our application works and it has a fairly decent DevOps infrastructure.

It's worth noting, however, that we've also faced challenges. The most important of them is the fact that we needed to glue together other people's works, which is done much easier when you have written the code yourself and know all its inner details.

The experience we've gained while developing this project is undoubtedly going to be valuable to us in our future careers. The project has made us more proficient in Git, Docker, GitHub Actions, and even tools from the Go and JavaScript development ecosystems.

Our work is published here:

- Repository: <https://github.com/Polinanime/sna25>
- Deployed application: <http://88.218.62.21:80>
- Demo video: https://drive.google.com/file/d/17iWMV7bnA27IKFvBYE_O5VE6OdVxLfYa/view
- Back end image: <https://hub.docker.com/r/polinaanime/sna25-backend>
- Front end image: <https://hub.docker.com/r/polinaanime/sna25-frontend>