My backend project can be found in the backend folder.

The frontend project can be found in the frontend folder. There are Cypress tests for the frontend. For the tests which are supposed to add something to the database (videogame, addition or news article), I couldn’t find out how to automatically test the uploading of an image, so that part should be performed manually. However, everything else regarding the Cypress tests works.

The project which is supposed to run the tests and be used for Sonarqube testing is in the tests folder. For some reason, when Flyway is applied to the project, the tests start failing which is why the project in the tests folder doesn’t use Flyway or migration tables because they are not needed for testing and all the tests run and the quality gate for SonarQube passes.

The actions which should be performed when changes are made to the GIT repository are written in the .gitlab-ci.yml file. SonarQube is included in the pipelines and it automatically runs and analyses the code.

The documentation for the project is in the documents folder.

The documentation consists of a Research Document, Design Document (with two additional UML diagrams for which there wasn’t enough space in the Word document), Security Report, UX Report, Project Plan, SonarQube screenshots which prove that it works and that I have 80% test coverage of the business layer, a SonarQube diagram and a URS document with Use Cases. There is also a folder with Google LightHouse Reports for all my pages.

The backend has a Dockerfile. The backend container can be started by following these steps:

* Execute the following command to start a MYSQL container: *docker run --name docker\_demo\_db -e MYSQL\_DATABASE=docker\_demo\_db -e MYSQL\_ROOT\_PASSWORD=my-secret-pw -p 3306:3306 -d mysql*
* Go to the backend folder and execute the following command to start a backend image: docker build -t backend .
* Then, to create a container from the image, execute the following command: docker run -d -p 8090:8080 --net=docker\_demo\_network\_staging --env spring\_profiles\_active=staging --name=backend-staging backend

After the commands are executed, a new MySQL container would be created, and the backend container would attach itself to it. It would automatically fill the new database with predefined data through Flyway.