The work is divided in frontend and backend.

The idea is to have two separate services, backend and frontend so that when adding extra features in one of the two, that won't affect the other working service.

The two services are Dockerized for portability and for enabling any developer to work on the application despite the system in use.

The functionality of the application can be tested locally by using docker to build and run the containers.

To simulate a production environment I decided to run de application in Kubernetes. In this way we can make the application scalable, highly available and fault tolerant.
Enabling CICD is straightforward, for example in Azure DevOps we can create a build pipeline that builds the image like we do locally and pushes it to the container registry (i.e Azure contains registry). After the build is completed we trigger the release pipeline where we pull the latest image from the container registry and create a deployment and services in our kubernetes cluster. This all process can be set to be triggered automatically by any new commit to master branch.

A lot of things are missing before moving to production:
- Visibility. There is no logging or monitoring solution in place.
- Security. No certificates, using http instead of https.
- HPA
- Cluster level load balancer, ingress resource and hostname.
- Proper testing, including load testing.