Complex Systems and DevOps: Deliverable 2

Date: November 25, 2024

Course Details

Course Name: Complex Systems and DevOps **Course Code:** 62582
Semester: Fall 2024

Team Members

Name	Student Number
Christoffer Fink	*s205449*
Kasper Falch Skov	*s205429*
Johan Søgaard Jørgensen	*s224324*
Henrik Lynggaard Skindhøj	*s205464*
Kevin Wang Højgaard	*s195166*
Sebastian Halfdan Lauridsen	*s215769*

Link to GitHub Project

Table of Contents

---

1. Introduction

Project Scope and Objectives

REST-assured Testing

OpenAPI Documentation

5. Frontend Development

React Application Structure The project features a React application for the frontend using Typescript. This is designed to provide and engaging and interactive experience for the users. It has a landing page with multiple sections, and navigation to each implemented game. Furthermore it is developed in a modular way, ensuring that scaling the project up and down can be done easily.

Key sections and features The landing page is the first page of the application and where the user will start when they enter. It is built up by using several components that form the landing page when put together. The header is at the top of the landing page, and serves as a navigation bar. It has the logo of the application and then links to different sections of the application: Home, About, Features and Contact. Below that is the hero section, where you would typically have a banner, welcoming the user onto the page and inviting them to navigate around the application to explore the different features. Below that is the games section where the different implemented games are listed along with buttons that navigate to said games. Below that there is a "Start" section, which is component that invites the user to sign up in order to play the games. Lastly a footer is implemented, which is a generic footer containing copyright information and links to resources such as terms of service and privacy policy. The application also has seperate game pages, which when navigated to, shows each of the games.

Navigation and Routing

For routing in the application react-router-dom is used to enable navigation between different pages. The routes are defined in the index.tsx file, and currently contains routes for blackjack, coinflip and the landing page. This setup makes it easy if new games are added, because they can just be added to the index file.

TypeScript Integration The project is built with TypeScript, which makes it possible to do static type checking. This is super helpful in order to catch errors during development instead of having to catch them at runtime. Typescript and the static type checking is best used when defining interfaces and types for components props and states. This limits the data passed between componenets such that it has to be structured properly for the component to accept it.

Vite Build Tool Implementation for building the application this project uses Vite. This is a popular build tool with some advantages mainly aimed at ease of use for the developers. The advantages include a really fast development server providing instant updates when changes are made to the code. It supports TypeScript out of the box, which minimized the amount of work developers have to focus on configuring build pipelines.

State Management In this project, Reacts built in state management hooks are used. UseNavigation is used from react-router-dom is used to handle the navigation state and allowing the user to navigate to different pages. Local states are also being used within the components with UseState. UseState can be used to handle specific UI states, for example when users are interacting with the application, or when dynamic content rendering is used.

Component Architecture

Security Features

Token Security Implementation

Package Management

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 helpås track changes to files over time, allows several developers to collaborate on the same project while ensuring that they dont accidentally overwrite each others 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 alot of features that can aid the collaboration between developers to a great extend. 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 continious integration and continous deployment.

Continuous Integration/Continuous Deployment

GitHub Actions Configuration

Build Server Setup

Testing Pipeline

Containerization

Docker Implementation

Container Registry

Cloud Deployment

Google Cloud Setup

Netlify Frontend Deployment

Monitoring and Maintenance

7. Conclusion

Project Outcomes

Future Improvements

Lessons Learned