

1. Introduction

1.1 Project Scope and Objectives The goal of this project is to design and implement a full-stack gambling platform, with games such as roulette, and coin flip. The project includes the development of both a backend, a frontend, and the use of DevOps practices to improve the development and deployment processes.

The project will provide an secure and intuitive user experience. Software using modern frameworks and tools will be implemented. The project will be implemented with an emphasis on using the practices of DevOps to enable continous integration and deployment.

1.2 Problem Statement and Solution Overview

Problem Statement: In this project several challenges had to be overcome, these include:

- **Scalability:** Making it possible for several different users to play simultaneously, while maintaining uninterrupted gameplay.
- **Security:** Preventing exploitation of tokens, to prevent potential threats.
- **Maintainability:** Creating a system that can be updated and improved without any significant overhead.
- **Deployment:** Developing a deployment process with frequent updates without downtime

Solution Overview: The project addresses these challenges the following ways:

- A **React frontend** using **Vite** and **TypeScript** to create a user-friendly interface.
- A **Quarkus-based backend** written in Java to implement core functionalities such as game logic, authentication, and data management. **PostgreSQL** is used as the database for data storage.
- **DevOps practices:**
 - **Containerization** with Docker to provide environments that are consistent across development, testing, and production.
 - **Version control and automation** using GitHub and GitHub Actions for continuous integration and deployment.
- An emphasis on **secure authentication and authorization**, by using **JWT tokens** to manage user sessions effectively.

These approaches both addresses the technical challenges, and ensures that the platform is secure, and scalable.

1.3 Methodology The project follows the Agile methodology of iterative development in a manner that allows for easier addaptation to changes in require-

ments and the users needs. The process initiates with comprehensive requirements gathering and analysis, where user stories and use cases are written in detail to ensure alignment with end-user expectations. This approach promotes clarity of purpose for the project and lays a solid foundation for later phases in development.

The development phase involves the creation of both the frontend and backend, each designed with scalability, and maintainability in mind. The structure of the frontend is component-based, built with React and TypeScript, promoting reusability. This makes it easier for developers to update and extend the user interface every time something a new feature is implemented. The backend uses a layered architecture based on Quarkus, which isolates concerns into three layers for business logic, API endpoints, and data management. This makes the code cleaner and easier to scale, making it easier to integrate further features.

DevOps practices are integrated into the project to improve the development and deployment processes. Continuous integration and deployment pipelines will be implemented with the help of GitHub Actions where code builds, testing, and deployments are automated. It minimizes manual labour, reduces the possibilities of human errors, and speeds up the release cycle. Docker is used for containerization, which maintains the consistency of the environments throughout the development, testing, and production stages, enhancing reliability and simplifying the deployment process.

The deployment strategy adopted here is cloud-based, with the backend deployed to Google Cloud, and the frontend hosted on Netlify.

By integrating these methodologies, the project ensures a smooth development process, that's responsive to change.