24-T1 OWL JUDGE

Development Document

CS 4850 – Senior Project

Sections 02 & 03, Spring 2025

March 9th, 2025

Project Team:

Name	Role	Contact
Lumiere Nathan Kue	Documentation, Group Leader	404-271-1584
Nathanael Fokou	Developer	404-663-5803
Bill Roan Simo	Developer	404-493-0240
Minh Quang Vu	Developer	678-579-7824

Advisor / Instructor: Sharon Perry

Role: Facilitate project progress; advise on project planning and management.

Contact: 770-329-3895

Table of Contents

1. Development

a. System Architecture

- **Hosting Platform:** Initially, we were to implement it with AWS Free Tier using EC2 instances for backend hosting. Due to some modifications to our project in our last meeting from Dr Perry, we are to develop a mobile app.
- **Database:** PostgreSQL as the primary relational database with Redis for caching.
- **Microservices:** The system is modular, divided into authentication, event management, scoring, and reporting services.
- **Frontend:** Built with React Native/flutter and dart to ensure a responsive, mobile-friendly UI.
- **Real-Time Updates:** Implemented using WebSockets to ensure score updates are reflected immediately.

b. Backend Development

- **API**: FastAPI is used for backend APIs, selected for its asynchronous support and high-speed performance.
- **Security:** Implemented OAuth 2.0 with JWT tokens for user authentication.
- **Audit Logging:** Logs all score submissions and administrative actions for security and accountability.

c. Frontend Development Framework:

- Framework: Flutter and dart/ react native for a cross-platform, mobile-first approach
- User interface: Judges access a dashboard displaying assigned projects, scoring criteria, and real-time score tracking.

d. Security Measures

- Data encryption: According to the specification in the email sent by Dr Perry, we are to use the latest version of encryption which is TLS 1.3(We were planning to use version 1.2+ in our previous documentations)
- **Role-Based Access Control:** Restricted access levels for Admins, Judges, and Participants.

e. Database Connection

I. Database Overview

- **Primary Database:** PostgreSQL stores user data, events, and scoring information.
- **Caching System:** Redis caches frequently accessed queries to optimize performance.

II. Connecting to PostgreSQL

The backend establishes a connection to the PostgreSQL database using SQLAlchemy in FastAPI. The connection configuration is as follows:

```
from sqlalchemy import create_engine

from sqlalchemy.orm import sessionmaker

DATABASE_URL = "postgresql://username:password@database-url:5432/dbname"

engine = create_engine(DATABASE_URL)

SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
```

NB: we don't yet have a concrete name for our database. So, dbname is just a sample name. That goes like wise for our database URL name.

Environment Variables

- DB_HOST: This refers to the database host
- DB_USER: This refers to the database username
- DB_PASSWORD: This refers to the database password
- DB_NAME: This refers to the database name

These variables should be stored securely in a .env file and loaded into the backend application before runtime.

2. Project setup

Building the Mobile App for Production

I. Install EAS CLI:

```
npm install -g eas-cli
```

II. Configure the project for Expo Application Services:

```
eas build:configure
```

III. Run the build command for Android/iOS:

```
eas build --platform all
```

IV. Once the build is complete, deploy using Expo's publishing service:

```
expo publish
```

3. Conclusion

This document provides a structured approach to setting up and deploying Judging app. The details outlined ensure that developers can efficiently configure the system, understand its architecture, and connect to the database.