24-T1 OWL JUDGE

Software Requirement Specification

CS 4850 – Senior Project
Sections 02 & 03, Spring 2025
January 29, 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 Content

I- Introduction

a) **Overview**

The OwlJudge Web Application is a digital platform designed for the task of judging the Computing Showcase (C-Day) events at Kennesaw State University's College of Computing and Software Engineering (CCSE). The current process is largely manual and depends on data entry and paper-based scoring, which are inefficient and error-prone. This project is an attempt to digitize the judging system by developing a cloud hosted, scalable and secure web application that will improve the experience for judges, administrators and participants. The system will support standardized project evaluations with real time score tracking and automatic result aggregation.

b) **Goals:**

- Streamline and automate the capstone judging event process.
- Develop a user-friendly web application for judges and administrators.
- Secure authentication and authorization are implemented to manage roles effectively.
- Scalability and cost efficiency are achieved through cloud-based services (AWS Free Tier / Azure Student Account).
- High availability and data integrity are ensured by PostgreSQL for structured data storage and Redis for performance optimization.
- Real-time monitoring and analytics for event administrators are provided.
- Synthetic test data is used for robust testing and development throughout the product lifecycle.

c) **Definition**

- CCSE: College of Computing and Software Engineering
- C-Day: Computing Project event at Kennesaw State University
- AWS: Amazon Web Services
- Redis: Database using in-memory caching to optimize performance
- PostgreSQL: Database management system

d) **Assumptions**

- The application is hosted on AWS Free Tier

- Application users will have access to internet
- Application users will use modern web browsers such as Chrome, Safari, Firefox…
- To the extent that the system collects and processes personal data, it will do so in accordance with the university's policies on data protection and information security
- The valid judge and project data will be provided by event organizers before each competition

II- General Design Constrains

a) Product Environment

The OwlJudge Web Application will be hosted on cloud services such as AWS Free Tier or Azure student account to ensure accessibility and scalability. The frontend will be built using React or Angular, while the backend will be powered by Node.js or Python. The database will be managed using PostgreSQL, with Redis implemented for caching. The application will be accessible via modern web browsers on both desktop and mobile devices.

b) User Characteristics

The application will be used by two primary user roles:

1. **Admin Users:**
    - Typically university staff or event coordinators.
    - Responsible for setting up events, assigning judges, and monitoring progress.
    - Expected to have basic to moderate familiarity with web applications.

2. **Judges:**
    - Professors, industry professionals, or external evaluators.
    - Responsible for evaluating capstone projects and submitting scores.
    - May have varying levels of technical expertise, requiring an intuitive UI/UX.

c) Mandated Constraints

- **Cloud Hosting Requirement:** The application must be deployed using AWS Free Tier or Azure student account to minimize hosting costs.
- **Security Compliance:** Authentication and authorization must be implemented using industry-standard security protocols (OAuth, JWT).
- **Database Constraints:** PostgreSQL must be used for storing event and scoring data, with Redis for caching to enhance performance.
- **Scalability:** The system must be designed to handle multiple events and judges concurrently without performance degradation.
- **Data Storage & Retention:** All event data should be stored securely, with audit logs maintained for at least one semester.

d) Potential System Evolution

- Future integration of a mobile application to allow judges to evaluate projects using smartphones.
- Implementation of AI-driven analytics to suggest scoring adjustments and prevent bias.
- Enhanced judge collaboration features, including discussion threads and feedback sharing.
- Expanded scalability to support multi-institutional judging events.

III- Functional Requirements

1. User Management & Authentication

- The system will support secure login and authentication for judges, administrators, and participants.
- Role-based access control will ensure that:
- Administrators can manage events, assign judges, and monitor progress.
- Judges can evaluate projects and submit scores.
- Participants can view their scores after judging is complete.
- Users will be able to reset their passwords securely.
- Authentication will be implemented using OAuth, JWT, or Single Sign-On (SSO).

- Judges will only have access to their assigned projects.
- The system will maintain an audit log to track user activity and security events.

2. Event Management

- Administrators will be able to create, edit, and delete events.
- Each event will support customizable scoring criteria and project categories.
- The system will allow bulk import of judge and project data to simplify event setup.
- Judges will receive email notifications upon event creation and assignment.
- A dashboard will provide administrators with real-time event progress and scoring updates.

3. Project & Judge Assignment

- Judges will be assigned to projects manually or automatically based on event settings.
- Assigned judges will receive notifications with details of their evaluation tasks.
- The system will ensure fair and balanced project distribution among judges.
- Judges will have access to detailed project descriptions and supporting documents.

4. Digital Scoring System

- Judges will be provided with an intuitive scoring interface for evaluating projects.
- Scores will be auto-saved periodically to prevent data loss.
- Judges will be able to review and edit scores before final submission.
- Once scores are submitted, they will be locked to prevent unauthorized changes.
- The system will support optional feedback submission along with scores.

5. Real-Time Score Aggregation & Ranking

- Scores will be updated and aggregated in real time as judges submit evaluations.
- A ranking system will automatically determine top-performing projects.
- The system will have tie-breaker mechanisms to resolve scoring conflicts.
- Administrators will be able to export results in CSV and PDF formats.

6. Reporting & Analytics

- The system will provide real-time analytics to administrators during the event.
- Organizers will be able to track judge activity and event progress.
- Historical event data and scoring trends will be stored for future reference and analysis.

7. Cloud Hosting & Performance Optimization

- The application will be hosted on AWS Free Tier or Azure Student Account.
- PostgreSQL will be used for structured data storage (events, users, scores).
- Redis will be used to cache frequently accessed data to improve performance.
- The system will be designed to handle at least 100 concurrent users without slowing down.

8. Security & Data Protection

- All user data will be encrypted at rest and in transit using TLS 1.2+.
- Passwords will be securely hashed using bcrypt or Argon2.
- The system will automatically log out inactive users after a set period.
- Audit logs will track administrative actions and judge activity for security and accountability.

9. User Experience & Accessibility

- The system will have a responsive design to work smoothly on desktops, tablets, and mobile devices.

- Judges will be able to submit scores quickly and efficiently with minimal steps.
- The interface will comply with WCAG 2.1 accessibility standards for usability.
- Confirmation messages and a progress bar will guide users through the scoring process.

10. System Monitoring & Maintenance

- The system will perform daily backups to prevent data loss.
- Automated error detection and logging will ensure reliability.
- The system will have auto-recovery mechanisms to restore service within 10 minutes in case of failure.
- Future updates and improvements will be modular to support easy enhancements, including potential mobile app integration.

IV- Non-Functional Requirements

1. Security

- The system will employ role based authentication and authorization (Admin, Judge)
- All user data and scores will be encrypted at rest and in transit (TLS 1.2+ encryption)
- Judges can only access their assigned projects
- The system shall keep track of all the scoring and administrative actions in an audit log
- Encrypted using hashed encryption algorithms like bcrypt or Argon2, passwords should be stored securely.

2. Performance and Scalability

- The system will process at least 100 concurrent users without performance degradation
- The time for submission and retrieval of the scores should not be more than 2 seconds even during normal load

- Redis caching should be used to improve performance and avoid overwhelming the database
- Will include provision for future expansion, like a mobile application

3. Usability

- The application should have a responsive web design to suit all screen sizes
- Judges should be able to give scores with just few clicks
- The user interface should be easy to use and understand, and should be accessible to all users, following the WCAG 2.1 accessibility guidelines
- For administrators, a dashboard should provide real-time score tracking

4. Reliability

- The system should be available 99.9% of the time during competition hours
- Data should be automatically backed up every day to avoid loss
- In case of failure, there shall be auto recovery mechanisms in the system to restore service within 10 minutes

5. Maintainability

- The system should be developed using modular system development practices so that it can be easily updated or enhanced
- All major components are to be documented for future developers
- The database schema should be normalized to enable effortless data modification

6. Compliance

- The application will comply with FERPA (Family Educational Rights and Privacy Act) to protect student data. The application will protect student data in compliance with FERPA (Family Educational Rights and Privacy Act)
- The system will follow university's IT security policies

7. Other Requirements

- The system shall support the running of multiple events at the same time
- A logging mechanism should monitor all the significant system events in order to use them for debugging and auditing

V- System Features

V.1 Feature: Event Management

V.1.1 Description and Priority

- **Description:** Admins will be able to create and manage events, assign judges, and monitor progress in real-time.
- **Priority:** High – This is the core functionality of the system, ensuring structured and fair judging.

V.1.2 Use Case: Event Setup

Actors: Admin
Preconditions: Admin must be logged in and have event creation permissions.
Flow:

1. Admin navigates to the "Create Event" page.
2. Admin enters event details (name, date, location, number of projects).
3. Admin assigns judges and sets scoring criteria.
4. Admin finalizes and publishes the event.

V.1.3 Additional Requirements

- Event setup should include **project categories** and **scoring criteria customization**.
- The system should send **email notifications** to assigned judges.
- Real-time **progress tracking dashboard** must be available.

V.2 Feature: Judge Score Submission

V.2.1 Description and Priority

- **Description:** Judges will evaluate projects based on predefined criteria and submit scores through the web application.
- **Priority:** High – This feature ensures the digital scoring process replaces the inefficient manual process.

V.2.2 Use Case: Submitting Scores

Actors: Judge
Preconditions: The judge must be assigned to an event and logged in.
Flow:

1. Judge selects an assigned project.
2. Judge reviews project details and evaluation criteria.
3. Judge inputs scores and optional feedback.
4. Judge submits the scores, and they are logged in the system.

V.2.3 Additional Requirements

- Scores should be **auto-saved periodically** to prevent data loss.
- Judges should receive **reminders** for pending evaluations.
- The system should implement **tie-breaker mechanisms** for scoring conflicts.