

Sports Event Platform

Project Documentation

Guilherme Cruz

July 2025

Introduction

This project is a sports event management platform built using Flask. It allows users to register, create and manage events, and sign up as participants. The goal was to provide a simple and user-friendly interface to plan and track sport-related activities, while implementing features common in production-grade systems like photo uploads, email notifications, and calendar exports.

Technology Stack and Justification

Flask: Chosen due to my familiarity and its simplicity. Flask allowed quick prototyping and modular design using Blueprints, making the app structure clean and maintainable.

SQLite: Selected for being lightweight and easy to integrate with Flask during development. It requires no external setup and supports relational data modeling effectively.

Bootstrap: Used for frontend styling to ensure responsiveness and ease of layout design without starting from scratch.

Flask-Mail: Integrated to provide email notifications upon account registration, event changes, or cancellations. Email support is controlled via environment configuration to avoid sending during development.

Architecture and Configuration

The application follows the factory pattern using `create_app()` in `__init__.py`. All routes are organized using Blueprints for separation of concerns (`auth.py`, `events.py`, etc.).

A `config.py` file holds all configuration constants, while `app_secrets.py` stores sensitive credentials and is ignored by Git. Logging is directed to a file using `RotatingFileHandler` for persistent storage of runtime information.

Key Features

- User authentication with optional profile pictures
- Event creation with attributes like date, location, sport type, and participant limit
- Dynamic registration with duplicate prevention and validation
- Role-based permissions (e.g. event editing restricted to creators or admins)
- Event cancellation, soft deletion, and participant management
- Automatic email notifications for key actions (registration, edits, cancellations)
- Calendar export functionality using the iCalendar standard

Development and Testing Process

Version control was handled with Git and a clean `.gitignore` to avoid uploading local or sensitive data. Logs were kept separate from stdout for better traceability.

I used ChatGPT throughout development to assist with debugging and discovering new libraries like Flask-Mail. This allowed me to overcome blockers more efficiently and learn unfamiliar tools while keeping a fast development pace.

Task Management and Workflow Organization

To maintain structure and productivity throughout development, I used **ClickUp** as a task management system. Each feature or fix was created as a separate task, and I logged the time spent on each task to track development effort.

For traceability, I included the corresponding task ID in every Git commit message, following the format: `#taskid: commit description`. This ensured a clear link between code changes and tracked tasks, helping maintain accountability and making it easier to revisit specific changes or decisions.

This workflow helped me stay organized, measure progress, and document my development process in a way that mimics professional software engineering practices.

Future Improvements

While the current version is fully functional in a development environment, several enhancements could be implemented to bring the platform closer to production-level deployment:

- **Deployment to the Web:** Host the application on a cloud platform with production-ready settings and proper domain configuration.
- **Database Migration to a Remote Server:** Replace SQLite with PostgreSQL or MySQL and configure access to an online database for multi-user support and persistence.
- **Admin Dashboard:** Add a backend dashboard for administrators to monitor users, events, and application status.
- **Event Recommendations:** Use activity history or sport preferences to suggest relevant events to users.
- **Data Analytics:** Add a dashboard for analyzing participation statistics, popular sports, and user engagement.

Conclusion

This project helped me consolidate knowledge in full-stack web development and good software practices. From modular architecture to environmental configuration, it reflects a thoughtful balance between practical implementation and clean maintainability. I'm proud of the result and confident it can scale further with new features or integrations.