



Los Angeles Kite Fighting (LAKF) **Technical Documentation Document**

Cael McDermott

Jake Moore

Mellisa Dodson

Ryan Morrell

CS367 - Practicum

5/6/2025

Contents

1. Introduction.....	3
2. User Manual.....	3
2.1 Overview of Available Features.....	3
2.2 Installation / Setup.....	4
2.3 Guided Walkthrough of Common Scenarios.....	4
2.4 User Roles.....	8
2.5 Update Procedure.....	9
2.6 Known Limitations.....	9
3. Troubleshooting.....	9
4. Engineering Documentation.....	10
4.1 System Architecture / Design Summary.....	10
4.2 Code or Configuration Details.....	11
4.3 Future Work or Handoff.....	11
5. Appendices.....	12
5.1 Glossary.....	12
5.2 References / Related Documents.....	12

1. Introduction

1. Introduction

This document provides a structured overview of the LAKF website: how it functions, who it serves, its technical requirements, and areas for improvement. It is designed to guide both everyday users and future developers by combining a user manual, troubleshooting guide, and engineering reference in one place.

2. Scope of this Document

This document is intended for LAKF staff, developers, and maintainers. It covers system setup, operation, maintenance, known limitations, and potential future development.

3. Overview

The LAKF website is a web-based system that allows users to host and join round-robin tournaments for kite fighting. It provides a way to create tournaments, register and manage participants, generate match schedules, track results, and maintain a historical record via an archive — making it easier to organize and run kite fighting competitions online.

2. User Manual

DISCLAIMER: A more direct and streamlined copy of the user manual is available by clicking the pink “?” icon at the top right of the website. For more detailed technical descriptions of how the site works, including installation and setup instructions, see 2.1 and 2.3.

2.1 Overview of Available Features

Currently, the LAKF website features the following:

1. “Host an Event” button, that allows users to create an round-robin event lobby upon the user entering the name, location, their name, and confirmation email of the event. From there, the host is redirected to a lobby page, where the lobby members are visible. From here, the host can kick undesired users from the lobby, as well as start the round-robin tournament.
2. “Join an Event” button, that, upon entering their name and valid email address, allows users to join lobbies by entering the pin number of the lobby. Users that join lobbies are redirected to a lobby page, where all members of the lobby are visible. Additionally, a QR code appears, allowing the user to be redirected to the round-robin tournament. Unlike the host, however, guests of a lobby cannot kick other users. From the lobby page, they may leave the lobby and be redirected toward the main homepage.
3. “View Match Archive” button, that allows the user to see all previous round-robin tournaments in our database. Upon selection of a round-robin tournament, the match number, player name, and player score becomes visible via an accordion-style graph.

2.2 Installation / Setup

Before any development can take place there is some setup to be done.

1. Download [PHP](#) and add the folder containing php.exe to your environment variable path.
 1. Open your php.ini file and ensure that the lines contain “extension=gd” and “extension=mysqli” and remove any semicolons before the line.
 1. Ex. ;extension=mysqli -> extension=mysqli
2. Download a MySQL and a MySQL Client (ex. HeidiSQL), add MySQL to your environment variable path.
3. Download [composer](#). This is the php package manager we will use to get all of our dependencies.
4. Git clone the website files onto your system from the remote
5. In a command line cd the local repository
6. Use composer to download the necessary packages using the following commands
 1. ‘composer require phpmailer/phpmailer’
 1. <https://github.com/PHPMailer/PHPMailer> for reference
 2. ‘composer require endroid/qr-code’
 1. <https://github.com/endroid/qr-code> for reference
7. Finally run the SQL-Setup file in the repository on your SQL workbench of choice to setup the database

2.3 Guided Walkthrough of Common Scenarios

This section provides a visual step-by-step guide for hosting and joining events, as well as viewing the match archive.

Hosting An Event

First Step:
Enter the Name
Of the Event
In the Top Box!

Home Join an Event

Name of Event
Ryan's Kite Party! :)

Location
Occidental College

Your Name
Ryan Morrell

Confirmation Email
anemail@gmail.com

Reset Create Event



Home Join an Event

Second Step:
Enter the
Location of the
Event!

Name of Event
Ryan's Kite Party! :)

Location
Occidental College

Your Name
Ryan Morrell

Confirmation Email
anemail@gmail.com

Reset Create Event



Home Join an Event

Third Step:
Add Your
Name!

Name of Event
Enter Event Name

Location
Enter Location

Your Name
Enter Your Name

Confirmation Email
Enter your email

Reset Create Event



Home Join an Event

Final Step:
Add your
Desired Email,
And Create the
Event!

Name of Event
Enter Event Name

Location
Enter Location

Your Name
Enter Your Name

Confirmation Email
Enter your email

Reset Create Event

Joining An Event (No QR Code)

First Step:
Enter the Pin Number
of the Event!

Home Host an Event

Pin Number*:
17

Flier Name*:
Ryan

Confirmation Email:*

someemail@gmail.com

Reset Join Event



Second Step:
Add Your Name!

Home Host an Event

Pin Number*:
17

Flier Name*:
Ryan

Confirmation Email:*

someemail@gmail.com

Reset Join Event



Third Step:

Add Your Desired Email
and Join the Event!

Home

Host an Event

Pin Number*:

17

Flier Name*:

Ryan

Confirmation Email*:

someemail@gmail.com

Reset

Join Event



Joining An Event (QR Code)

Ryan's Kite Tournament!

★ Ready to play? Scan the QR code to join the lobby! 🎈✨



PIN: 89

Scan the Host's
QR Code With
Your Device, and
Join the Lobby!

Lobby



Cael

Viewing Match Archive

The interface consists of two main sections. The top section, titled "First Step: Select a Match!", displays a list of events with their IDs and names. The bottom section, titled "Second Step: View the Match Data!", displays the details for a selected event, including a table of match results.

Event List (Top Section):

- Event ID: 62 – Zoom Kite
- Event ID: 53 – Office Hours
- Event ID: 52 – Things and stuff
- Event ID: 51 – Freaky Fliers
- Event ID: 19 – fdsafads
- Event ID: 17 – fdaf
- Event ID: 14 – fdaf

Match Data (Bottom Section):

Match #	Player 1	Player 1 Score	Player 2	Player 2 Score
2	Guy who eats kites	3	Guy who eats kites	0
3	Guy who eats kites	0	Ryan	3
5	Guy who eats kites	2	Ryan	0

2.4 User Roles

- **Host / Organizer**
 - Creates and manages new tournaments
 - Configures event details (name, location, date)
 - Manages participants and starts the tournament
 - Oversees the match schedule and results
- **Participant / Player**
 - Joins existing tournaments using a provided lobby PIN
 - Enters personal details (name, email)
 - Views the lobby page, match schedule, and results
 - Competes in assigned matches

Currently, there are no additional administrative or moderator roles beyond the host's in-system controls. Future system improvements could include expanded role types or more detailed permissions.

2.5 Update Procedure

Use `git pull` in the local project directory to retrieve the latest code changes from the remote repository.

Run `composer update` to ensure all PHP dependencies are up to date.

Restart your local server or refresh the web environment as required. This can be accomplished by opening an integrated terminal in the `frequent-fliers-html` directory and launching the local server using the command:

```
php -S localhost:8000
```

Test the system by loading the homepage and checking key functionalities (e.g., hosting/joining events, viewing the archive) to ensure the update was successful.

2.6 Known Limitations

Currently, there are a few known limitations with this current build of the LAKF website:

1. The system currently supports only round-robin tournament formats; other formats like single-elimination or double-elimination are not yet implemented.
2. Admin controls are limited to basic functions (e.g., kicking users); there is no role-based access control or advanced moderation system.
3. The website requires a PHP and MySQL server environment; it has not been tested or deployed on alternative stacks (e.g., Node.js, PostgreSQL).
4. The QR code functionality depends on third-party libraries, which may require periodic updates or maintenance.

3. Troubleshooting

Here are some common errors users may encounter and methods for troubleshooting them:

Problem: The website does not load

Cause: PHP is not installed or not added to the system PATH

Solution: Verify that PHP is properly installed and that the directory containing `php.exe` is included in the system environment variables.

Problem: Database connection fails

Cause: Incorrect MySQL credentials or missing database setup

Solution: Check the configuration file for correct database username, password, and host. Ensure the MySQL server is running and the required database has been set up.

Problem: Composer install or update fails

Cause: Composer is not installed or required packages are missing

Solution: Install Composer and run `composer install` or `composer update` to download necessary dependencies.

Problem: QR codes do not generate

Cause: Missing `endroid/qr-code` package

Solution: Run `composer require endroid/qr-code` to install the QR code generation library.

Problem: Email notifications do not send

Cause: PHPMailer is not configured correctly

Solution: Check the PHPMailer configuration settings, including SMTP server details, username, and password.

4. Engineering Documentation

4. Engineering Documentation

All project files (except `.json` and `.lock` files) include a header comment at the top that lists the **Last Editor**, **Date**, **Course**, and **File Name**, along with a brief description of what the file does in relation to the website's functionality. Additional inline comments are placed above more complex functions, briefly explaining their purpose and how they contribute to the system's behavior.

4.1 System Architecture / Design Summary

The LAKF website is built as a PHP-based web application, supported by a MySQL database for persistent data storage.

- **Frontend:** HTML, CSS, and JavaScript for user interface and interaction.
- **Backend:** PHP scripts handling tournament logic, participant management, and data processing.
- **Database:** MySQL, storing tournament details, player data, and archived match results.
- **Third-party libraries:** PHPMailer (for email notifications) and Endroid QR Code (for QR code generation).
- The system follows a modular file structure, with separate scripts for event hosting, joining, match tracking, and result archiving.

4.2 Code or Configuration Details

- **Project directory:** Organized into folders separating frontend assets, backend scripts, and configuration files.
- **Configuration files:** Store database connection details and system constants.
- **Dependency management:** Handled using Composer, ensuring all required PHP libraries are installed and up to date.
- **Database setup:** Provided via an SQL script (`SQL-Setup.sql`) that creates the necessary tables and relationships.
- **Key files:**
 - `host_lobby.php` → Handles event creation
 - `join_lobby.php` → Manages participant registration
 - `archive.php` → Displays past tournament results
 - `create-bracket-email.php` → Sends confirmation and notification emails

⚠ QR Code Configuration Note:

The QR code generation includes a direct link that is hardcoded to the current hosting environment (e.g., `ff.cianci.io/*someaddress*`). If the site is moved to another hosting platform or tested locally, this link must be updated in the relevant backend code or configuration file to match the new domain or localhost address.

• 4.3 Future Work or Handoff

To continue development or expand the system, future developers should:

- Review the header comments in each file for guidance on file purpose and key functions.
- Check the Composer dependencies regularly for updates or replacements.
- Consider adding role-based permissions or advanced admin controls for improved moderation.
- Expand tournament formats beyond round-robin to include knockout or league styles.
- Improve the frontend experience with responsive design for mobile compatibility.
- Document any new features or updates with consistent commenting and version control practices.
- Establish and manage social media accounts (e.g., Twitter, Instagram, Facebook) using the designated project Google account (`kitefightla@gmail.com`). Store account passwords securely in a private, shared document or password manager — do not include them in this public-facing documentation.
- Currently, hosts can kick themselves from their own lobby. Remove this functionality.

5. Appendices

5.1 Glossary

- **Host / Organizer** → A user who creates and manages a kite fighting tournament event.
- **Participant / Player** → A user who joins an existing tournament to compete.
- **Lobby PIN** → A unique code generated for each hosted event that participants use to join the correct lobby.
- **Round-Robin** → A tournament format where each participant competes against every other participant.
- **Composer** → A dependency manager for PHP used to install and manage external packages.
- **PHPMailer** → A third-party PHP library for sending email notifications.
- **Endroid QR Code** → A third-party PHP library for generating QR codes.
- **SQL Script** → A file containing structured commands used to create or update database tables and relationships.

5.2 References / Related Documents

- LAKF GitHub Repository → [LAKF GitHub Repository](#)
- PHPMailer Documentation → <https://github.com/PHPMailer/PHPMailer>
- Endroid QR Code Documentation → <https://github.com/endroid/qr-code>
- Composer Documentation → <https://getcomposer.org/doc/>
- PHP Documentation → <https://www.php.net/docs.php>
- MySQL Documentation → <https://dev.mysql.com/doc/>