Horse API Overview

Project Summary

I took on this project in 2023 as part of my ongoing efforts to give back to the communities that I live in. I currently reside in Houlton Maine, which has a large culture around farming along with other activities like horse riding. This project aimed to fix a problem that I quickly saw as I was volunteering with Aroostook Riding Club (One of 2 major riding clubs in Aroostook County), the clubs data was almost all on paper creating traceability issues along with more work than was necessary. I worked with club officers to set out business requirements and from there built out my technical requirements. From this a 3 year plan was created with input from club officers, starting with a Google Sheets page to start just keeping track of data and eventually working into a Rest API with a UI to integrate into currently existing or new web sites. In 2024 word of mouth allowed me to get the attention of the other major riding club (Pine and Spurs). After talking to them and showing them the inner workings of the application I was able to scale the project and application to take on multiple clubs with separate data sets. In total between the 2 clubs the project is servicing over 100 users as of 2025 with the possibility of scaling to encompass more clubs in the Maine state area.

Business Requirements

- Cost Efficiency
    - The solution must have minimal to no recurring costs, as the organizations are small nonprofits with limited budgets.
- Data Troubleshooting Accessibility
    - The system must provide simple tools and error messages that allow non-technical staff to identify and resolve common data issues without developer involvement.
- System Integration
    - The solution must integrate seamlessly with existing systems and workflows, avoiding duplication of effort and ensuring data consistency.
- User Interface & Data Validation
    - The user interface must be intuitive and visually appealing, enabling ease of use for staff with varying technical skills.
    - The system must validate inputs to prevent unwanted, incomplete, or incorrect data from being entered.

Technical Requirements

- Cloud Hosting
  - The system must utilize free or low-cost cloud-based platforms to minimize hosting and infrastructure expenses.
- Familiar Tooling
  - The solution must leverage tools already in use by the organizations, such as Google Sheets, to reduce training needs and adoption barriers.
- Automated Data Validation
  - The user interface must read from existing datasets and apply automated validation rules to prevent invalid or duplicate entries.
- System Simplicity & Reliability
  - The architecture must minimize dependencies and overall footprint to reduce points of failure and simplify long-term maintenance.
- Data Accessibility
  - The system must allow users to view, enter, and edit data in a familiar spreadsheet-like format (e.g., via Google Sheets or an equivalent interface).

Project Acceptance Criteria

Cost Efficiency / Cloud Hosting

- The solution must run on free or low-cost cloud services (e.g., Google Workspace, Google Apps Script).
- No additional paid licenses or infrastructure should be required to use core functionality.
- Deployment and ongoing usage should not exceed the nonprofit's existing budget.

Data Troubleshooting Accessibility

- Non-technical staff can identify data issues through clear error messages or visual indicators.
- Email based error messages explain what went wrong and how to resolve it in plain language.
- Staff can correct common issues (e.g., missing required fields, invalid formats) without developer support.

System Integration / Familiar Tooling

- Users can connect the solution to existing Google Sheets with no manual file exports/imports.
- Data entered or updated in the system is reflected in Google Sheets in near real-time.
- No duplication of effort is required (e.g., data does not need to be retyped into multiple systems).

User Interface & Data Validation

- The interface prevents submission of incomplete, invalid, or duplicate entries.
- Validation rules automatically run on data before it is accepted.
- The interface design is simple, visually clean, and can be used without formal training.
- Admin users (non-technical) are able to complete basic tasks (add/edit/view records) without assistance.

System Simplicity & Reliability

- The number of moving parts (scripts, connectors, services) is minimized.
- The solution functions correctly if one component fails (graceful error handling).
- The system requires minimal ongoing technical maintenance (no manual server patching, updates, etc.).

Data Accessibility

- Users can view, add, and edit data in a spreadsheet-like interface.
- Authorized staff can make updates without needing developer access.
- Data changes are logged or timestamped to provide traceability.

Project Timeline

Year 1

- Set up structured Google Sheets as the primary data repository.
- Define consistent data schema (columns, validation rules).
- Add simple data validation in Sheets (dropdowns, required fields).
- Train non-technical staff on data entry best practices.
- Create built-in Sheets formulas for reporting (SUMIF, QUERY, FILTER).
- Implement protections (range locking, edit permissions).
- Document workflows and troubleshooting steps.

Deliverable : Fully functional spreadsheet-based system, automated enough for daily operations, with minimal developer maintenance.

Year 2

- Create basic Google Form to integrate into the spreadsheet
- Create a basic read only API to allow the google form to read data from the sheet
- Create an audit trail for user entries
- Pilot new forms application to the public and get feedback
- Document API and new Forms data flow.

Deliverable : Hybrid solution — staff still use Sheets, but some operations are API-driven with a lightweight Google Forms front-end.

Year 3

- Develop a robust API layer hosted on Google Cloud Functions.
- Implement secure authentication for staff applications (OAuth, Google Identity).
- Create a polished HTML/CSS/JS web interface for member entries and data viewing.
- Create robust reporting system utilizing Google Sheets
- Create a legal audit trail for waivers and electronic signatures
- Run beta system and collect feedback
- Finalize documentation, training, and prepare for any ongoing maintenance.

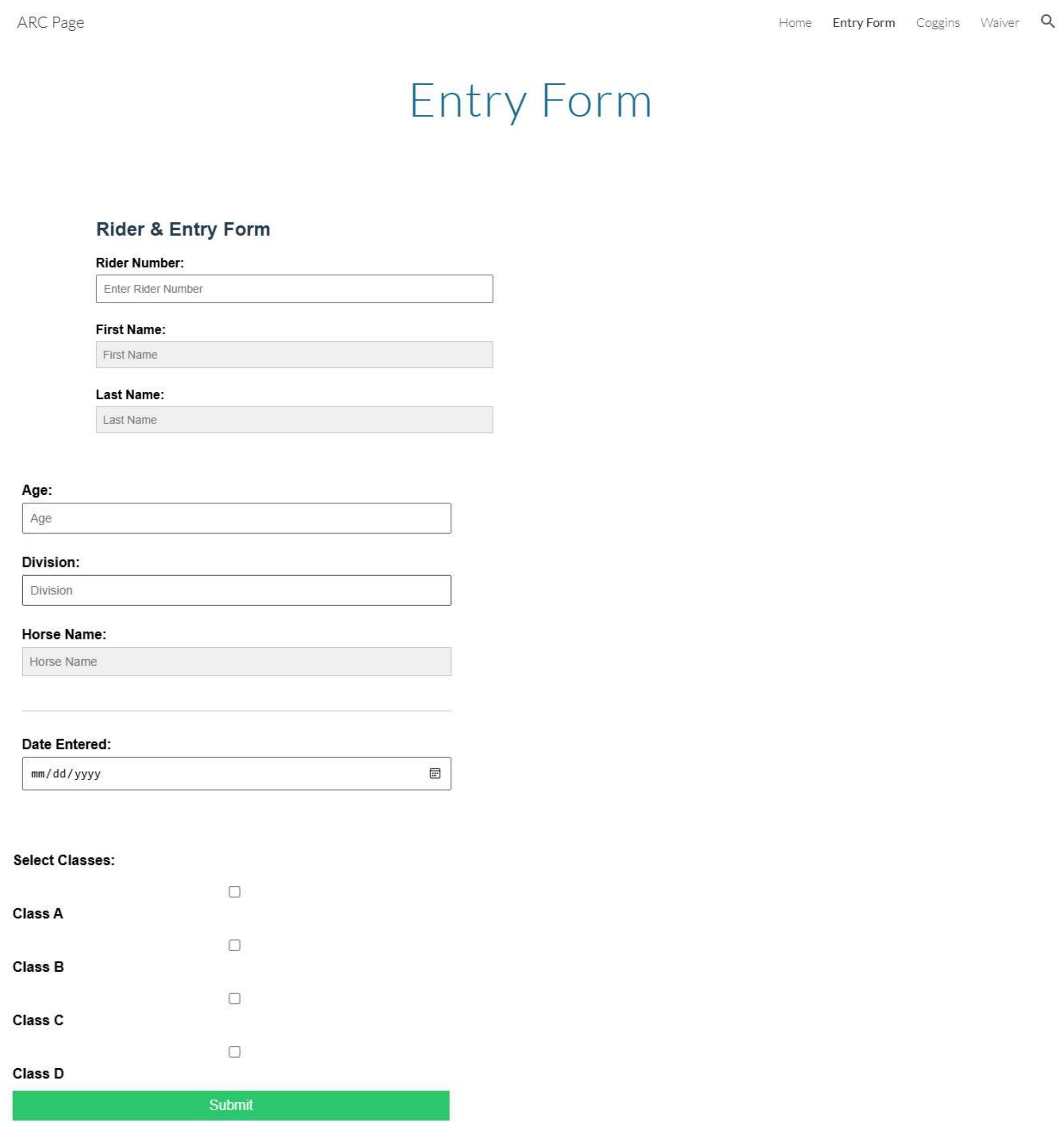
Deliverable : API system with a user-friendly front-end, Google Sheets reporting integration, and seamless external system connections.

Beta Site Images

(Note, source code is not being published here due to security concerns)

Entry Forms

This page allows for riders to enter into competitions


ARC Page

Entry Form

Rider & Entry Form

Rider Number: Enter Rider Number

First Name: First Name

Last Name: Last Name

Age: Age

Division: Division

Horse Name: Horse Name

Date Entered: mm/dd/yyyy

Select Classes:

Class A ☐

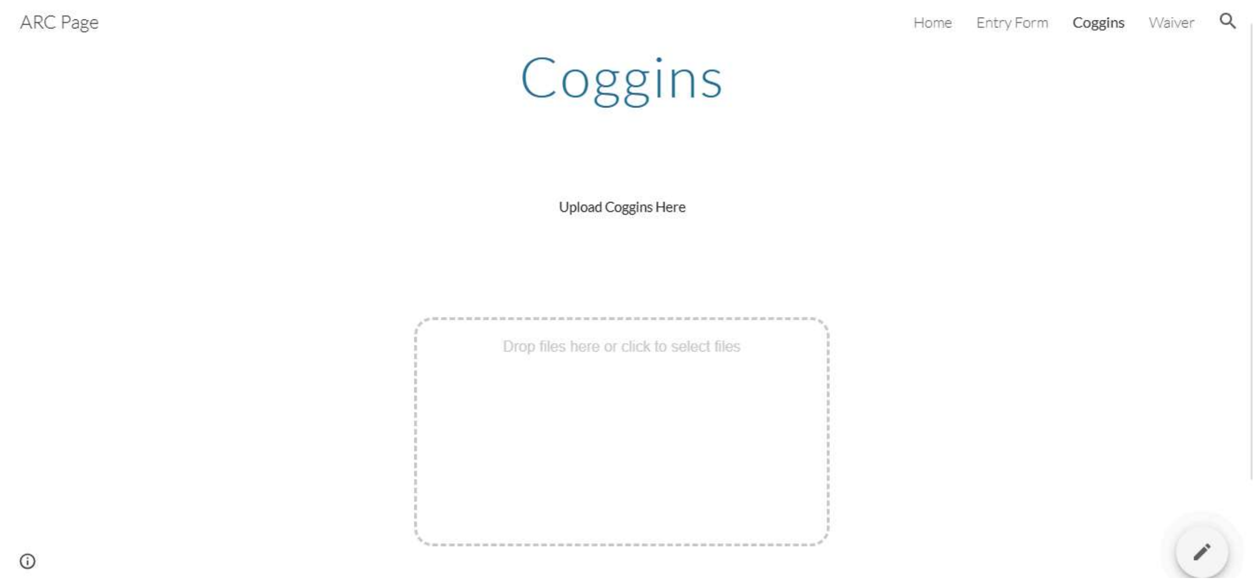
Class B ☐

Class C ☐

Class D ☐

Coggins

This page allows for riders to submit their Coggins (A horse disease) form to allow them to compete. This process goes through an internal audit process due to this being a state and federal requirements.



ARC Page

Home Entry Form Coggins Waiver

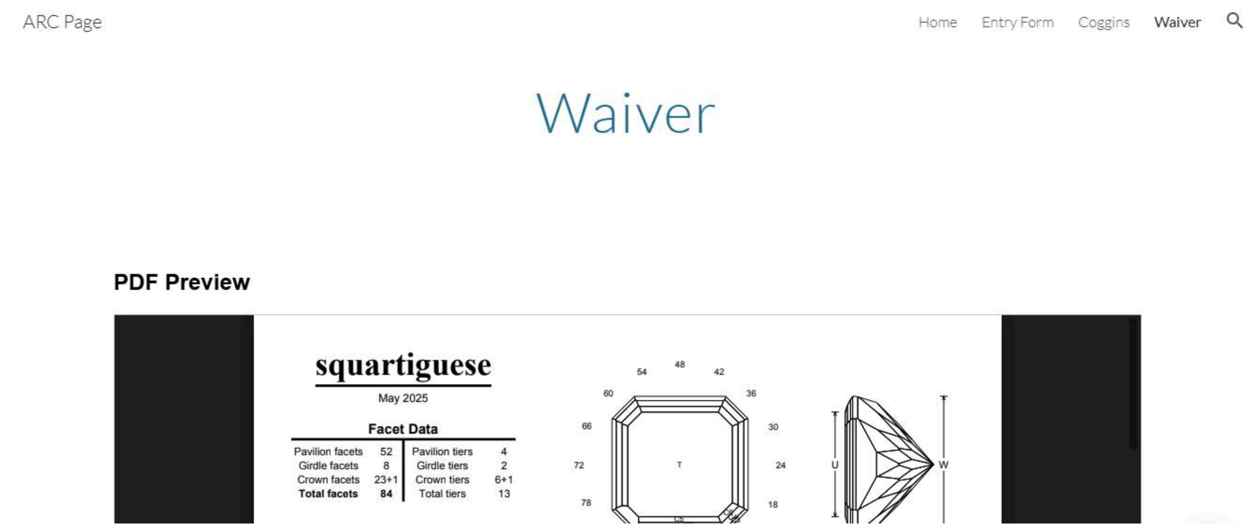
Coggins

Upload Coggins Here

Drop files here or click to select files

Waiver

This page allows for riders to submit their injury waiver form to allow them to compete. This process goes through an internal audit process due to liability requirements. (Note, PDF is a filler due to privacy concerns. Fun fact I cut gems as a hobby)



ARC Page

Home Entry Form Coggins Waiver

Waiver

PDF Preview

squartiguese

May 2025

Facet Data

Pavilion facets	52	Pavilion tiers	4
Girdle facets	8	Girdle tiers	2
Crown facets	23+1	Crown tiers	6+1
Total facets	**84**	Total tiers	13

Parent/Guardian Name (Optional):

Type name here

Clear Guardian Signature

Participant Name (Required):

Type name here

Clear Participant Signature Submit & Save Signed PDF