**Previous Backlog**

**Blue = Sprintable**

**Green = in sprint**

**Yellow = Needs more grooming**

**Pink = Re-evaluate at a later date**

--------------------------------------------------------------------------------------------------

**Previous Backlog**

--------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------

**Theme:** Data! Preservation and Migration

**Sprint Goal:** Preserve data generated or submitted to site and safely migrate data between releases

A person in a military jacket

Description automatically generated

--------------------------------------------------------------------------------------------------

**PBI: As a business owner, I want data on the production instance to be kept to the next production release for analytics and security.**

Acceptance criteria

* Prior to release, all production data that must persist needs to be known
* Post release, the production data that must persist needs to be verified to have not changed from data prior to release

Obstacles

* Does the data actually need to be on the site?

**PBI: As a product owner, I want to be able to safety rollback in the case a critical error is found in production to ensure users are minimally impacted.**

Acceptance criteria

* A deployed production instance can be downgraded to last release instance that did not have discovered critical issues. There exists documentation of the process.

Obstacles

* What sort of things does a rollback need to do? What is lost during a rollback?

**PBI: Deploy the microservice alongside webapp**

Acceptance criteria

* Deploy the webapp and the microservice and show the communication working

Obstacles

* None

**PBI: As a business owner, I want data on the staging instance to be kept to the next staging release for analytics and security.**

Acceptance criteria

* Prior to release, all staging data that must persist needs to be known
* Post release, the staging data that must persist needs to be verified to have not changed from data prior to release

Obstacles

* Does the data actually need to be on the site between releases?

**PBI: As a business owner, I want to know if there are any legal notices needed with data collection to better inform users of what is being collected**

Acceptance criteria

* Determine if there are any legal notices that must be given with data being collected (game stats, submission files, etc.)
* Document potential issues somewhere
  + Give to HR

Obstacles

* None

--------------------------------------------------------------------------------------------------

**Current Backlog**

--------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------

**Theme:** CI/CD

**Sprint Goal:** Explore CI/CD topics and automate parts of development pipeline

--------------------------------------------------------------------------------------------------

**PBI: Clean up**

(This is technically part of our definition of done but here it is anyway)

Acceptance criteria

* Update readme with information about where the project is and where it needs to go
* Add starting point to the readme
* Verify documentation is present in repo and up to date
* Ensure Definition of Done is updated in documentation

Obstacles

* None

**PBI: As a dev, I want to be able to have my code tested when I push it to a branch, so the testing process is streamlined, and feedback is quicker**

Acceptance criteria

* Have code be automatically tested when its pushed (and generate feedback based on success status)

Clarifications

* Feedback can be in one or more forms (email, text, repo workflow diagram, merging, etc.)

Obstacles

* Tests cases need to be written and used

**PBI: As a team member, I want to be able to deploy safely to production with a button or trigger so that the release process is quicker and reliable**

Acceptance criteria

* With a CI/CD tool (like GitHub actions), demonstrate the solution being built and deployed to the main branch into the production environment.
* Reject if the deployment fails tests on either the testing or production environments.
* Reject if the building or deployment fails.

Obstacles

* Strongly advise GitHub actions for this

--------------------------------------------------------------------------------------------------

**Remaining Backlog**

--------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------

**Theme:** Gateways 😐

**Sprint Goal:** Understand what Gateways and Gateway APIs are and create a basic submission gateway

--------------------------------------------------------------------------------------------------

**PBI: KA: Understand what gateway apis are and how they work**

Acceptance criteria

* Provide a written or verbal report (1 or 2 paragraphs will suffice) that explains what gateway APIs are. Present to the team on your findings. Once everyone has verified there understanding with the original team member, the goal is met.

Obstacles

* None

**PBI: As a developer, I want the microservices to use a gateway api for communication to better prepare for scaling with many users**

Acceptance criteria

* Create a gateway API such that it can facilitate the communication between 1 or more microservices and the webapp. Demonstrate the communication.

Obstacles

* None

-----------------------------------------

Excess Backlog

-----------------------------------------

**PBI: As a dev, I want to be able to store submissions somewhere on the site so that the submission is preserved for processing and implementation.**

Acceptance criteria

* Submission data submitted from the client needs to be stored and identified by the webapp.

Obstacles

* None

**PBI: As a dev, I want users to only be able to submit safe file formats (.js, .py, .txt, etc) to ensure better site safety and make submissions uniform.**

Acceptance criteria

* Limit submissions to certain file types or formats.
* Update submission status to reflect success or failure based on type
* Update criteria to inform user of what file types or formats are acceptable
* Submissions should be accepted/rejected based on type of file that is submitted and that status should be reflected to the front-end.

Obstacles

* What are safe file formats?

**PBI: KA: Learn how microservices communicate**

Acceptance criteria

* A member of the team needs to learn how microservices communicate
* Knowledge transfer needs to occur
  + The member needs to present on this to the team
  + Documentation of information needs to be generated
* At least a member of the team is able to generate documentation of how microservices communicate. Then, that member(s) should walk the team through the explanation. Each member can then provide a summary (written or verbal) which needs to be verified by the original member(s). Once everyone has a verified understanding, the goal has been met.