**Deployment Plan**

### 1. Goals

Features we want implemented:

1. Navigation bar

a. Change the navigation bar to be its own html view that is loaded into each page, maintaining consistency, and making maintenance easier.

2. Mobile support/formatting

3. Accessible and easy to understand admin mode

a. Add and remove exhibits

b. Add new sections to the home page

c. View recorded data

d. Protected with log in

4. Pages

a. Home page

b. Exhibits page

c. Science of Play page

d. Map

e. Outside/Continuing Play page

5. Recording data

a. Allow users to track what type of play they see their child do at each exhibit

b. Allow users to vote on how their experience at an exhibit was

How much testing and under what conditions

How many known bugs will be fixed

### 2. Assign roles:

Hunter

* Footer
* Exhibits page
* Mobile formatting
* Documentation

Nicole

* Home page
* Exhibits page
* General UI/UX
  + Font
  + Background gradient
  + Circles
  + Animation on home page
* Admin mode
* User documentation

Ella

* Map

Thomas

* Outside/Continuing Play page

Liam

* Science of Play page
* Tracking info to database (user activity, map page exhibit voting)

Sean

* Django, backend, and database
* Live hosted site
* Admin mode

Testers: Liam, Ella, Thomas

Debuggers: Hunter, Nicole, Sean

Hunter

* General mobile formatting errors (including text, images, and box sizing)

Nicole

* UI/UX bugs and flaws
* App organization flaws

Sean

* Database and backend bugs

### 3. Determine user-developer communication:

How will users report issues to the development team?

On the bottom of each page there is a section to report any issues users may have with the app.

### 4. Determine deployment task dependencies:

Start-to-Start:

* Map formatting. The map buttons must be started in order for the map formatting to be started.

Start-to-Finish:

* Map voting tracking. Map buttons must be started to be able to finish tracking the votes to the database.
* Buttons to track the type of play at each exhibit. Buttons must be started to be able to finish tracking the button presses to the database.

Finish-to-Finish:

* View recorded data. There must be data that is being recorded for the recorded data view/data analyst view to be finished.

### 5. Bug tracking:

How will you keep track of bugs as they appear and as they are fixed?

Bugs will be tracked using a Kanban board and documented using an excel spreadsheet.

### 6. Training:

Who needs training (users, system admins, etc.)?

Admins:

* Add and remove admin users/log-ins
* Add, remove, and edit exhibits
* Add new sections to the home page

Data Analysts:

* View recorded data

### 7. Accountability

Who is responsible for post-deployment changes?

Hunter and Sean

Hunter

* Email: [amon2504@outlook.com](mailto:amon2504@outlook.com)
* Discord: zo333\_

Sean

* Discord:

### 8. BC/DR (business continuity, data recovery) plan:

1. If a major issue occurs
   1. How will you ensure minimal disruption business practices
      1. Pause the hosting of the website for maintenance
   2. How will you recover lost data? How will you make sure that data can be recovered?
      1. Monthly backups
   3. How big of an issue will cause you to halt deployment? And how will you roll back to the previous system while you fix the issue?
      1. An issue with the security of the admin page would be very major and require a halt of the deployment
   4. What is the point of no return - when during deployment can you not halt and not roll back? And what will you do to minimize disruption then?
      1. If the website is currently be used for research purposes or marketing rolling back or halting would likely not be an option
         1. Disruption will be minimized by having the people accountable update/fix the issues and possibly refer to the members who worked on each part for additional information and assistance