**REQUIREMENTS**

**High-Level Design**

* **(M)**Database from SQL - Using SQL as our main database to store necessary information based on the user's activities.
* **(M)**User interface - Our user interface will include a user login screen, allow the user to view all museum activities and events, give the user an option to change the website's language, inform parents on the importance of play and the types of play there are. It will also suggest activities for parents to complete with their kids.
* (**M**)HTML through Django - Using Django as both a backend and front end. Will be using the Template function within Django to display HTML docs that we created for the main displays
* **(M)**Digital design - For our digital design we are going to be basing our visuals on the website shared with us by the museum called vroom. Our home page will explain what play is and its importance, have buttons to learn about types of play, explore exhibits and find educational activites.
* **(M)**Admin interface - Our admin interface will include a variety of features. The admins will access this interface using an admin-specific username. Once logged in, The admins will be able to adjust the layout of the page to their liking(GUIS) and will be able to adjust the text on the page as necessary. Some other features would include changing text font, text color, etc. We will add more features as we discover the needs of the clients.
* **(M)**Data collection - For data collection, we are going collect the time stamps of time spent on an activity and whether or not the activity was liked or disliked
* **(M)**Hardware Platform - Web application (All devices have access) – Formatting of our site will work for phones and computers
* (**S**)Model Type: KANBAN - We are using the KANBAN type because it is simple and easy to keep the team on the same page regarding what needs to be done.

**Low-Level Design**

* **(C)**Calendar – Calendar links to the museum’s existing calendar page to help inform users
* (**M**)One databases with 5 tables written in mySQL
* (**S**)Exhibit Class
  + Exhibit name
  + Floor level
  + information
* ●  (**M**)Admin interface accessible through Admin login
  + Text Theme -The admin will be able to select the desired text and click a drop-down box including a wide variety of fonts to choose from.
  + Logos - The admin will be able to select a GUI that holds the current logo, and upload a PNG or similar file from their computer to import the logo they would like.
  + Text - The admin will be able to click on a text box and edit the text within as they would like.
  + Colors - We will create the website with a color theme similar to a museum to start, but Admins will be able to select bodies of text, background, etc, and select from the dropdown list of colors to change it to.
  + The layout of the page - The admin will be able to edit the layout of the page as they see fit using a GUI-type format. The Admin will be able to just drag and drop text boxes, logo(PNG) boxes, etc to wherever they they see fit on the web application.
  + Save button to apply changes - Have a button that when pressed will apply all changes made by the admin. If the admin terminates the process without pressing the save button, all changes will be reverted back to how they were prior to the admin making changes.
* **(S)** A contact page for getting in touch with the museum (questions/suggestions)
* **(S)** An “About” page that explains the purpose of the museum and the site.
* **(C)**Admin Class- For our admin class, we want to allow the museum employees to sign into the website and make changes to the events and exhibits. Because they’re not tech-heavy, we want to make it easy to edit for non-programmers but also have to make it harder for them to alter certain parts of the website so they can’t break or ruin any features. As well, they will be able to see the data that's collected so they can use it for their research.
* ●  **(M)**User Class
  + ○  Username/PFP - Users can select their username and profile picture for their

account. If they have already created an account, this will show the login options

for their account.

* + ○  Age Range - Users will select the birthday of their child when creating an

account, or if they sign in as a guest. Guest sign-in will not save information, but this information will be stored for users who create accounts. Birthday will be used so it can make different age recommendations as the child gets older.

* + ○  Gender - Users will select their child's gender which will be used for data collection. If a user continues as a guest they will input this information every time they login. If they create an account, they will only input once.
  + ○  Admin - (True or False) - User will select if they’re an admin or user, and there will be a passkey that can be entered to create admin accounts and prevent non-admins from creating admin accounts.
  + ○  Location of the app being used(i.e. at home or at a museum) - Users will select whether they’re signing in from home or the museum, which will allow for different activities to be recommended based on the location.
  + **○  etc( add as needed)**