

Project Summary

In the context of the 5th edition of Dungeons and Dragons, this database models a game with a DM, players, and player characters. It includes all player character skills, attributes, inventory, and other game-relevant aspects, such as status effects and character class and race. The database will allow players within a game to keep track of all of their characters and their individual inventories, abilities, and dynamic characteristics such as status-effects, and character level.

Tech Stack

- Oracle
- PHP
- HTML/CSS

To-Do

Frontend (Alexa/Diana)

- Create design documentation for layout/UI flow (Alexa)
 - Landing page
 - Sign up
 - Log in
 - Main screen
 - Sidebar
 - Main interaction area
 - list of characters
 - list of games w/ game descriptions and DM info
 - item/spell/attack descriptions
- Implement frontend (Diana)
 - Setup: follow [Tutorial 7](#)
 - Landing page
 - Main screen

**Testing will occur as each item is implemented

Backend (Max/Diana/Alexa)

- Schema edits, tweaks, and changes (Max/Diana)
- Implement all tables from Milestone 2 (Max)
- Single SQL script to create all the tables and data in the database (Max)
- Design user queries (Add/Delete/Edit/Sort) (Max/Alexa/Diana)
- Implement user queries (Add/Delete/Edit/Sort) (Max/Alexa/Diana)
- Design user queries (Additional functionality) (Max/Alexa)
- Implement user queries (Additional functionality) (Max/Alexa)
- Final full-suite backend testing

**Testing will occur as each item is implemented

Front-to-Back Integration (Alexa/Max/Diana)

- Connect Frontend to Backend (Alexa/Max/Diana)
 - TBD
- Testing (Alexa/Max/Diana)

Timeline

Rough deadlines for task completion. Not meant to be restrictive; we can work ahead if possible.

March 11th:

1. Add Milestone 1 to repo (Alexa)
2. Add Milestone 2 to repo (Max)
3. Add initial Milestone 3 Timeline to repo (Diana)

March 14th:

4. Schema edits, tweaks, and changes done and added to repo (Diana/Max)
5. Design documentation completed and uploaded to repo (Alexa)
6. Add updated Milestone 3 Timeline to repo (Diana)
 - a. Before uploading: cross off all todos/timeline items that have been completed. Update challenges with things left/things we get stuck on.

March 17th:

7. Landing page implementation started (Alexa/Diana)
8. Implement all tables from Milestone 2 started (Max)
9. Single SQL script to create all tables and data in database (Max)
10. Landing page backend integration started (Alexa/Diana/Max)

March 19th:

11. All landing page frontend and backend implementation finished (Alexa/Diana/Max)
12. Start main screen frontend implementation (Diana)
13. Start user query design/implementation (Alexa/Max/Diana)
 - a. Add/Delete/Edit/Sort

March 26th:

14. All main screen frontend and backend implementation finished (Alexa/Diana/Max)
15. Start main screen backend integration (Diana/Alexa/Max)
16. Start additional user query design/implementation (Alexa/Max)
17. Start additional user query frontend integration (Diana/Alexa/Max)

April 2nd:

18. Finish additional user query design/implementation (Alexa/Max)
19. Begin final testing

April 5th:

20. Finalise and submit project!

Challenges

- Ensuring work by different people works together properly
- Robust and comprehensive forms of testing
- Proper GitHub formatting and style

MILESTONE INFORMATION

Deliverables

- ~~1. A brief (~2-3 sentences) summary of your project. You can reuse the summary from milestone 2.~~
2. **Timeline and task breakdown/assignment:** The breakdown should be at a level of detail that demonstrates that the group has spent time meaningfully considering what there is left to do. Note that we are not asking you to predict every single possible task that you will need to do. We want to see that the group understands the scope of what is left to do and is prepared to accomplish the remaining tasks in a reasonable manner.
 - a. **Each task should be specifically assigned to a group member (or combination of group members).** It is in your best interest to be as explicit as possible about who will work on what.
 - b. In the event that there is a dispute between group members, this is one of the first things the course staff will look at when evaluating the situation. If it is clear to us what has been agreed on, it will speed up the process of conflict resolution.
 - c. Unless otherwise stated, it is assumed that all group members will work equally on the project. This does not mean that everyone needs to work on each task together. **This means that the overall division of the work is equal. If this is not the case, state the work percentage breakdown for each member.** This will serve as a written acknowledgement between all group members that there will be an uneven distribution of work. The member who does not do their fair share of work will have a penalty applied to their final project grade.
 - d. While each member is not expected to know about every single line of code in the project, it is expected that all members can talk about the overall architecture of the project.
 - e. **The timeline should contain enough detail for your project mentor to determine that you understand that you need to produce a GUI for your full project.**
- ~~3. The deliverables from milestones 1 and 2 have been added to the repository.~~
4. **Each group member has made a commit to the repository.** The commits do not have to be code related. For example, one group member can commit the milestone 1 deliverables, another the milestone 2 deliverables, and the third member the milestone 3 timeline.
 - a. During the meeting, your project mentor may ask each group member to demonstrate that they are able to make a commit. It is in your best interest to get everything set up now. In the event of a group dispute or disagreement over the level of contribution made by each member, we will be looking at the commit history. Git is an essential skill and this is a great way to get some practice at using it in a group context.

Rubric

- Each team member has committed to the department provided repository at least once. The commits do not have to be code related. An example of three commits could be: 1) a README file, 2) the file(s) for milestone 1, and 3) the file(s) for milestone 2. **(1 pt)**
- ~~— A brief summary of the project has been provided. (1 pt)~~
- The project TA has been provided with a list of tasks that the team has left to do before the project is completed. Each task has been given a reasonable deadline. **(2 pts)**