Memory Game – Charity Katz Final Project – CSCI E-33a Harvard – Fall 2021

Project Overview

This project consists of a memory game but instead of the normal pictures, it uses text from JSON. The JSON also includes feedback that is displayed each time a match is made.

The especially unique part of this game is its extensibility to be used as a Knowledge Check with the ability to enter the card data into Excel (carda, cardb, and feedback), and save it as a tab-delimited text file. Once saved, the JavaScript converts the text file to JSON, which is then used in the game. This provides an opportunity for non-technical curriculum developers to customize the content in the game.

The moves and time elapsed are tracked during the game, and then saved to a User model along with the number of times the user has won the game and the last time they won.

The saved game information is available on the user's profile page, along with their follower and following counts that can be clicked to see the full list of individuals. In addition, the game information is displayed on a Winners Circle page, which is sortable using the Data Tables that are available using JQuery and Bootstrap. This JQuery also provides pagination for the table.

Users must be logged in to play the game, view profiles, and to view the Winner's Circle.

Files Breakdown

Python Files

Nothing special was done within the project's main python files. However, within the **memory** app, content was changed on most.

- **admin.py** – In order to utilize the User model for my model with additional fields and then to see and have the ability to manipulate those fields within the admin site, this was altered. To configure the UserAdmin, all fieldsets, including those part of the AbstractUser needed to be configured. The additional fields for the created model were adding in an *Additional info* fieldset.

- **models.py** – Per the assignment, at least one model needed to be created. For this, it was a modification of the User model. Five (5) additional fields were included:
    - *following* – This tracks who the user is following using a ManyToManyField
    - *number_of_wins* – This is an IntegerField that increases incrementally by 1 with each win.
    - *best_moves* – This is an IntegerField that tracks the user's best moves. This field is only updated when a user wins the game AND the number of moves is less than the previous best.
    - *last_win* – This is a DateTimeField that is automatically updated each time a user wins the game with the current date and time at the time of the win.
    - *best_time* – This is an IntegerField that stores the number of seconds the game was played before it was won. This is converted to a standard hh:mm:ss view in the tables in which it is displayed.

- **urls.py** – This includes the URLs for the easy to see navigation:
    - *index* – This is where the game is displayed to be played but redirects to the *Login* page when a user has not yet logged in.
    - *login* – This is the route to where users login.
    - *logout* – This is the route for when users logout, at which time they are redirected to the Login page.
    - *register* – If a user has not yet registered, they can visit this route/view.
    - *profile* – This is the route to the user Profile page.

  - *winnerscircle* – This is the route to the Winners Circle page.
  - *gamewon* – This is the only route not clearly visible, and it is used to update the User in the database when they win the game.

- **views.py** – Within the views.py are the various views that are used primarily to interact with the user.
  - *index* – This is where the game is located. However, if a visitor is not authenticated, it redirects them to the Login page.
  - *gamewon* – When the game is won, this view is used to update the user's database record with their moves and time, if they have never won before or it is greater than their currently best moves and/or time. It also updates their record with the current time and date of the win to count as their last_win, and incrementally adds to their win. There is an "or" included to avoid issues where the field is blank because they have yet to play. In that case, it adds 1 to 0. It is protected with a @login_required decorator to prevent them from navigating to it through the URL bar without first logging in, which would display the Memory game.
  - *winnerscircle* – This view is used to pass the list of users with a win to the Winners Circle page where it is then used in a for loop within the table. It is protected with a @login_required decorator to prevent them from navigating to it through the URL bar without first logging in.
  - *login_view* – This is used to log users in if they have already registered.
  - *logout_view* – This is used to log users out.
  - *register* – This is used to register new users. All fields are not used within the create_user variable because there is a limitation by Django.
  - *profile* – This is used to update who a user is following by adding or removing from their following field when they click the button on another user's profile, and it also grabs the followers and who they are following from the database to display, along with the user's bests. @login_required decorator to prevent them from navigating to it through the URL bar without first logging in.

Templates

There are six (6) HTML templates, which are used to display the game and user-directed pages.

- **layout.html** – This includes all of the universal components. In order to utilize Bootstrap and the JQuery Data Tables for sorting, the external JS and CSS are included here. At the end of the Body is the Bootstrap JavaScript bundle, per the advice of Bootstrap. This also includes the navigation bar items. To make it mobile accessible, the navigation bar utilizes Bootstrap's collapse option, which provides a hamburger icon in place of the navigation text as smaller viewports, and it can be expanded to access the navigation elements.
  - *Username*, *Play Memory*, *Winners Circle*, and *Log Out* are displayed when a user is logged in.
  - *Log In* and *Register* are used when a user has yet to log in.

- **index.html** – This is the index page and where the game is built. It includes a Bootstrap modal, which also uses the JavaScript previously mentioned. This is displayed when a match is made to provide feedback regarding the match. Because the game uses images and files in the static folder, which are needed by the JavaScript, they are created as variables here for the game because JavaScript would not support grabbing static files but can use the universal variables.

- **profile.html** – This page displays information about the user. If the visitor is the logged in user, they will see additional information such as their email and the number of people they are following, which they can click to open a modal in order to see the list of people. If the visitor is not the user whose page it is, they will see a follow/unfollow button. In addition, for everyone, the user's number of followers is displayed. The number can be clicked to see the list, and the user's bests are available.

- **winnerscircle.html** – This page displays a sortable table with each winner and their bests. It is sortable by each column using the JQuery and Bootstrap Data Tables plugin, which also includes pagination. The names and usernames are clickable, and they redirect to the user's page where the logged in user can follow/unfollow them.

- **login.html** and **register.html** – These include standard login and registration pages, but name has been added to the standard registration page.

Static Files

The static files include two (2) JavaScript files, one (1) CSS file, the cards files, and images for the cards.

- **cards** folder – This folder contains two files. One is Excel, which can be easily edited. The header cannot be changed or the associations will no longer work, but the remaining content can be edited as needed. The first column defines the ID, which is used on the game board and to check for a match. The next two columns are the content for the card, which should have some association, in this case it is adult and child animal names, and then the feedback, which is displayed when a match is made. After this content is edited, it can be saved as a tab-delimited text file, which is used by the JavaScript. The names of the files need to be retained for things to work properly. The game is set to have 20 cards and 10 matches.

- **images** folder – This contains images that are used for the cards. Only one image is used but I had a couple options I was switching between.

- **styles.css** – This is the CSS file for styling components of the various pages. It includes some necessary table styling for the Data Tables, and there are media styles for the viewport sizes to help ensure it is mobile friendly as best as it can be.

- **gameplay**.js – This is the main JavaScript file as it is used for the game. On document load, it initiates the text to JSON conversion by fetching the text file and then splitting it and converting it to JSON. It passes that to build the buildGameBoard function, which is used to build out the game board by creating cards for each of the carda and cardb column items. Then it randomizes the cards for display on the board and adds an event listener to them. When cards are clicked on, they flip, and a match is checked for by comparing the IDs. If they match, some classes are added, and the feedback is displayed. If there are 20 cards with the class "match," the game is won, and the game info is sent to the database. For non-matches, the cards flip back over. One is added to the moves counter for each two cards flipped, and the seconds are counted as well. This is included in a hidden display that the JavaScript pulls after the game is won but also in a formatted string. This was done this way to make it an easy integer to add to the database.

- **winners.js** – This JavaScript does two things. It converts the seconds integer from the database to hh:mm:ss format, and uses the JQuery/Bootstrap plugin to make the table sortable and add pagination.

Goals Outcomes

These are the goals that were outlined in my original submission. They have been accomplished, but additional elements, such as including the best moves and best time on the profile page have been added.

Memory Game – Charity Katz Final Project – CSCI E-33a Harvard – Fall 2021

Goal	Outcome Level	Complete d
Memory Board (10 pairs)	Good	Yes
Match JSON object key:value pairs (key1 and key2 are same id)	Good	Yes
Flips cards when clicked and looks for match when 2 have been clicked	Good	Yes
JS checks for match and updates class if match or flips back	Good	Yes
Feedback is displayed on match	Better	Yes
Uses Excel file saved as tab-delimited text and converted JSON to form objects	Better	Yes
Registration, login, and logout functionality	Good	Yes
Login required for game play (not included in original submission)	Better	Yes
Profile page displaying username, email, and number of wins	Good	Yes
Winners circle page with usernames, number of wins, last win date	Better	Yes
Tracking number of moves and time per game and include best moves and time on winners circle	Best	Yes
Winners circle sortable by wins, best moves, best time	Best	Yes
Paginate winners circle to display 10 winners	Best	Yes
Add follow/unfollow and display on profile followers and following	Best	Yes