Group Name: Team Rocket
Team members: Tommy Agarwal, Paul Daniel Bernabe, Sophia Singh, Jun Wu

Introduction

The purpose of this project report is to provide a description of Rocket Game List, our web application project. Rocket Game List is a video game record-keeping web application, designed to allow users to save a video game list and keep track of the various games they play. The application consists of various functionalities to allow gamers the best game list experience. Rocket Games gives users the ability to sign up for an account, search through the database for games, add/remove games from their list and view their list. The objective of this project is to create a user-friendly web application that lets game fanatics search for their favorite games and create a personalized list of games they have played or want to play.

Back-end - Server Side Functionality

We utilized PHP and MySQL Workbench for the back end of our project implementation. The key features of our server-side implementation are:

1. User authentication and registration

Our application allows users to create an account using their email address, username, and password. If a user is signing up, the server validates their account information and the database checks if the user account already exists. If not, the new user information will be inserted into the MySQL database. If an account has been created, the user can directly log in using their username and password. The PHP application then creates a connection to the MySQL database server. After validating the login attempt, the user is directed to the video game list page. We use PHP sessions to keep track of the login status of the user and keep the user in login status when they change pages. Additionally, users can log out from their accounts and the session will be unset.

2. Game list management

Once logged in, the user is redirected to the games list page. The PHP server establishes a connection with the MySQL database and retrieves the user's game list data. The rows of games in the user list will be sent to the client side. Users can then remove games from this list or add a game to their completed game list on the client side. The user's option to add, remove and change the completion status will retrieve by the server. The server will update the database with the newly created list information.

3. Add the game to the list

Following the user searching for a game, the PHP server then checks if the user has clicked the 'add to list' button. This functionality gives users permission to add games from their search query to their personal list or completed games list. The server will first check if the user has already login and if the game is already in the list. If both validation passed, the server will

retrieve the game ID from the RAWG API call, and the logged-in user ID, and executes the SQL statement in order to retrieve the game information and insert it into the game list.

4. Display the game list
When the user visits the List page, the server will check if the user has logged in. If the user has logged in, then the server will retrieve all the game IDs of the user's game list from the database. The server will do an API call for each of the game IDs and retrieve all the game information from the API call, then send the information to the client side and display them.

Client Side Functionality

We used the RAWG API and applied HTML, CSS, and JavaScript for the front end of this application. The client-side functionality includes the following components:

1. API requests to retrieve game information
The rawg.io API provides access to the vast majority of video games in their database. It also includes critical information about the game including its cover image, title, description, genre, etc. We utilize this API in order to give our users the functionality to search for games and retrieve important information. We send an HTTP request using the fetch() method to the rawg.io API. The request is sent with the user's search query as a parameter, returning the result on a card with basic game information. JSON is used for the purpose of parsing the response data.

2. Add games to the list
After the user searches for a game, the results will be displayed as cards, each card has the game's image, name, genre, release date, and the "add to list" button. After the user clicks the button, the client side will be sent the game ID to the server. The server will validate the request and then insert the game ID into the database. The result of the insertion will prompt the user with an alert message such as "Game added to list", "Game already in the list", or "Please login first" on the client side.

3. Sign Up/Login pages
The tools that were employed to create the signup and log-in pages were HTML and CSS. Users are given an easy-to-use and visually appealing form to enter and validate their account information. The information entered by the user will be sent to the server for validation. Error messages will be shown on the page if the entered information has issues such as incorrect password, empty fields, or username already being used.

4. Card Display
We integrated CSS and HTML to display the game information cards. We gave a clean, visually appealing look to this. The basic design includes the game name, basic information, and the option to remove/add from the list. Users can view the game information cards after they log in and visit the List page.

5. Navigation Bar

Users have access to all the page links through the user-friendly navigation bar. This feature was implemented utilizing HTML CSS and JavaScript. Using HTML, we created a navigation bar with 5 links redirecting to different pages of the web application: Home, Login, Sign Up, Search, and List. After the user login, the user will only be able to visit Search and List page, and an additional option of Logout to logout and return to the Home page. We used CSS to style the navigation bar and make it visually appealing.

GitHub Link: https://github.com/SuperJunlw/Rocket-Game-List

Website Link on the server: http://cos-cs106.science.sjsu.edu/~012714650/rocket_list/home.php