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 Games 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, and add/remove games from 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 and login 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 onto 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.
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 are then displayed as cards. Users can then remove games from this list or add a game to their completed game list. This will update the database with the newly created list information.
3. Game search
   The PHP server connects to the MySQL database and using the information collected from the user, searches for the game information. It retrieves the search keyword from the form submission, creates an sql query, binds parameters in the database, executes and returns the search result as a card.

4. Add game to list
   Following the user searching 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 onto their personal list or completed games list. It retrieves the game ID from the form submission, the logged in user ID, and executes the SQL statement in order to retrieve the game information and insert it onto the game list.

Client Side Functionality

We 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 a 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. Sign Up/Log In 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 validate their account information.
3. 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.
4. 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 4 links redirecting to different pages of the web application: Log in, Sign Up, Search, and List. We used CSS to style the navigation bar and make it visually appealing. Finally, JavaScript was used to implement the functionality into the navigation bar.