

**personal-website/major-projects/project02/script.js**

```
1  /*
2    Name: Kevin Blinn
3    Class: CPSC 3750
4    Date Due: July 20th, 2024
5    Assignment: Project 02 – Collect App Phase 1
6  */
7
8  // This function runs when the entire DOM content has been loaded
9  document.addEventListener('DOMContentLoaded', function() {
10
11    // Get the search form element by its ID
12    const searchForm = document.getElementById('searchForm');
13
14    // Check if the search form exists
15    if (searchForm) {
16      // Add an event listener to handle the form submission
17      searchForm.addEventListener('submit', function(e) {
18
19        // Prevent the default form submission
20        e.preventDefault();
21
22        // Get the value of the search input field, trimmed of whitespace
23        const query = document.getElementById('searchQuery').value.trim();
24
25        // Call the searchGames function with the query
26        searchGames(query);
27      });
28    }
29
30    // Parse the URL parameters and Get the 'id' parameter
31    const params = new URLSearchParams(window.location.search);
32    const gameId = params.get('id');
33
34    /// Fetch and display the details of the specified game if it exists
35    if (gameId) {
36      fetchGameDetails(gameId);
37    } else {
38      searchGames('');
39    }
40
41    // Fetch and display stats about the games collection
42    fetchStats();
```

```
43 });
44
45 // Function to search for games based on a query
46 function searchGames(query) {
47
48     // My personal API key for RAWG Video Games Database API
49     const apiKey = '02fdad8f60254eea8c8509bf12e6dcde';
50
51     // Construct the API URL with the query and page size parameters
52     const url = query
53         ? `https://api.rawg.io/api/games?key=${apiKey}&search=${query}&page_size=100`
54         : `https://api.rawg.io/api/games?key=${apiKey}&page_size=100`;
55
56     // Make an AJAX request to the RAWG API
57     $.ajax({
58         url: url,
59         method: 'GET',
60         success: function(response) {
61
62             // Check if the response contains any results and display it
63             if (response.results.length > 0) {
64                 displaySearchResults(response.results);
65             } else {
66                 // Display a message if no results are found
67                 displaySearchResults([]);
68             }
69         },
70         error: function(xhr, status, error) {
71             // Log any errors that occur during the request
72             console.error('Error fetching games:', error);
73         }
74     });
75 }
76
77 // Function to fetch and display details of a specific game
78 function fetchGameDetails(id) {
79
80     // Same steps as before
81     const apiKey = '02fdad8f60254eea8c8509bf12e6dcde';
82     const url = `https://api.rawg.io/api/games/${id}?key=${apiKey}`;
83
84     // Make an AJAX request to the RAWG API
85     $.ajax({
86         url: url,
87         method: 'GET',
```

```
88     success: function(response) {
89         // Display the game details
90         displayGameDetails(response);
91     },
92     error: function(xhr, status, error) {
93         // Log any errors that occur during the request
94         console.error('Error fetching game details:', error);
95     }
96 });
97 }
98
99 // Function to display the search results on the page
100 function displaySearchResults(games) {
101
102     // Get the results container element by its ID
103     const resultsContainer = document.getElementById('results');
104
105     // Clear any existing content in the results container
106     resultsContainer.innerHTML = '';
107
108     // Check if there are any games to display
109     if (games.length === 0) {
110         resultsContainer.innerHTML = '<p>No games match this title.</p>';
111     } else {
112         // Loop through each game and create a card element for it
113         games.forEach(game => {
114             // Create a div element for the game card
115             const gameElement = document.createElement('div');
116
117             // Add a CSS class to the game card
118             gameElement.classList.add('game-card');
119
120             // Set the inner HTML of the game card with the game's information
121             gameElement.innerHTML = `
122                 
123                 <h3>${game.name}</h3>
124                 <p>${game.released || 'No release date available.'}</p>
125                 <a href="details.html?id=${game.id}">View Details</a>`;
126
127             // Append the game card to the results container
128             resultsContainer.appendChild(gameElement);
129         });
130     }
131 }
132
```

```
133 // Function to display the details of a specific game on the page
134 function displayGameDetails(game) {
135
136     // Get the game details container element by its ID
137     const detailsContainer = document.getElementById('gameDetails');
138
139     // Set the inner HTML of the details container with the game's information
140     detailsContainer.innerHTML = `
141         
142         <h2>${game.name}</h2>
143         <p>${game.description_raw || 'No description available.'}</p>
144         <p>Release Date: ${game.released}</p>
145         <p>Rating: ${game.rating || 'N/A'}</p>
146         <p>Developers: ${game.developers.map(d => d.name).join(', ')}</p>
147         <p>Genres: ${game.genres.map(g => g.name).join(', ')}</p>
148         <p>Platforms: ${game.platforms.map(p => p.platform.name).join(', ')}</p>`;
149 }
150
151 /* FUNCTIONS FOR GETTING THE STATS */
152
153 // Function to fetch and display various stats about the games collection
154 function fetchStats() {
155
156     // API key for RAWG Video Games Database API
157     const apiKey = '02fdad8f60254eea8c8509bf12e6dcde';
158
159     // Fetch total number of games
160     fetchTotalGames(apiKey);
161
162     // Fetch most popular game
163     fetchMostPopularGame(apiKey);
164
165     // Fetch latest released game
166     fetchLatestReleasedGame(apiKey);
167
168     // Fetch highest rated game
169     fetchHighestRatedGame(apiKey);
170 }
171
172 // Function to fetch and display the total number of games in the collection
173 function fetchTotalGames(apiKey) {
174     // Construct the API URL with the page size parameter
175     const url = `https://api.rawg.io/api/games?key=${apiKey}&page_size=1`;
176
177     // Make an AJAX request to the RAWG API
```

```
178 $.ajax({
179   url: url,
180   method: 'GET',
181   success: function(response) {
182     // Display the total number of games
183     $('#totalGames').text(response.count);
184   },
185   error: function(xhr, status, error) {
186     // Log any errors that occur during the request
187     console.error('Error fetching total games:', error);
188     // Display an error message if the request fails
189     $('#totalGames').text('Error loading');
190   }
191 });
192 }
193
194 // Function to fetch and display the most popular game in the collection
195 function fetchMostPopularGame(apiKey) {
196   // Construct the API URL with the ordering parameter for most recently added games
197   const url = `https://api.rawg.io/api/games?key=${apiKey}&ordering=-added&page_size=1`;
198
199   // Make an AJAX request to the RAWG API
200   $.ajax({
201     url: url,
202     method: 'GET',
203     success: function(response) {
204       // Check if there are any results
205       if (response.results.length > 0) {
206         // Display the name of the most popular game
207         $('#mostPopularGame').text(response.results[0].name);
208       } else {
209         // Display 'N/A' if no results are found
210         $('#mostPopularGame').text('N/A');
211       }
212     },
213     error: function(xhr, status, error) {
214       // Log any errors that occur during the request
215       console.error('Error fetching most popular game:', error);
216       // Display an error message if the request fails
217       $('#mostPopularGame').text('Error loading');
218     }
219   });
220 }
```

```
221
222 // Function to fetch and display the latest released game in the collection
223 function fetchLatestReleasedGame(apiKey) {
224     // Construct the API URL with the ordering parameter for latest released games
225     const url = `https://api.rawg.io/api/games?key=${apiKey}&ordering=-released&page_↵
size=1`;
226
227     // Make an AJAX request to the RAWG API
228     $.ajax({
229         url: url,
230         method: 'GET',
231         success: function(response) {
232             // Check if there are any results
233             if (response.results.length > 0) {
234                 // Display the name of the latest released game
235                 $('#latestReleasedGame').text(response.results[0].name);
236             } else {
237                 // Display 'N/A' if no results are found
238                 $('#latestReleasedGame').text('N/A');
239             }
240         },
241         error: function(xhr, status, error) {
242             // Log any errors that occur during the request
243             console.error('Error fetching latest released game:', error);
244             // Display an error message if the request fails
245             $('#latestReleasedGame').text('Error loading');
246         }
247     });
248 }
249
250 // Function to fetch and display the highest rated game in the collection
251 function fetchHighestRatedGame(apiKey) {
252     // Construct the API URL with the ordering parameter for highest rated games
253     const url = `https://api.rawg.io/api/games?key=${apiKey}&ordering=-rating&page_↵
size=1`;
254
255     // Make an AJAX request to the RAWG API
256     $.ajax({
257         url: url,
258         method: 'GET',
259         success: function(response) {
260             // Check if there are any results
261             if (response.results.length > 0) {
262                 // Display the name of the highest rated game
263                 $('#highestRatedGame').text(response.results[0].name);
```

```
264     } else {
265         // Display 'N/A' if no results are found
266         $('#highestRatedGame').text('N/A');
267     }
268 },
269 error: function(xhr, status, error) {
270     // Log any errors that occur during the request
271     console.error('Error fetching highest rated game:', error);
272     // Display an error message if the request fails
273     $('#highestRatedGame').text('Error loading');
274 }
275 });
276 }
277
```