Project Step 6

Group 95: Mikasa Hackerman

Team: Eli Mills, Na Kim

Project Title: “Gametyme”

Website URL: http://flip3.engr.oregonstate.edu:9501/

Executive Summary

Since the initial version of our project, we have made a lot of changes as we further developed our application as well as from feedback we received from classmates. From the earlier feedback we received, we made many minor changes such as fixed inconsistencies in naming attributes and entity conventions as well as adding more attributes in Companies and Platforms entities due to these Entities lacking in attributes. In addition, to normalize the tables to 3NF, two additional tables were added. The tables Locations and GamesPlatforms were added to make it easier to reference to the Platforms and Companies tables.While working on the schema, we also decided to update the ON DELETE requirements for all foreign keys in the following tables: Playthroughs, Sessions, Games, Companies, and Platforms.

While further developing the front end of our project, we realized that some of our pages were user-focused rather than admin-focused, thus we had to make additional adjustments to remove user-focused forms such as sign up/login/profile page as well as some queries. Furthermore, the GamesPlatforms intersection table was not being implemented in the UI, so we added the appropriate queries to associate Games with their Platforms. To implement filtering logic in our SQL queries, we initially implemented a search bar on the Games and Users page in the UI and DML queries. However due to time constraints, we removed the search bar feature and replaced it with select dropdowns in forms. We implemented select dynamically populated dropdown menus for Genre, Platforms, Companies, Games, and Playthroughs add/edit CRUD forms in both the UI and SQL queries. In addition, we decided to add filters for our Games page where users can filter by Genre/Company/Platform easily which was then implemented in our UI and queries. To accommodate these changes on the front-end, we enabled mySQL to handle multiple queries at once and had DELETE statements require all attributes to match to prevent accidental deletion.

From the last feedback we received from peers, we received suggestions to add a deletion confirmation pop up that confirms if a user wants to delete a row as well as to have edit forms autofill in the input field. Thus we have made the appropriate changes by having edit forms be prefilled as well as a delete confirmation modal box implemented. While debugging our final draft of our project, we came upon a possible bug where a user may click the “finished playthrough” or “end session” button when the playthrough/session was already completed. To prevent users from running into any errors or crashing our backend, we added a fix to hide the buttons. We have also added basic special character escaping for all text input as an attempt to mitigate SQL injection.

Project Outline

I) Overview:

Gametyme is a database-driven website which provides gamers a platform to track the video games they have played along with their gameplay hours and number of playthroughs. There are over 5 million games in the world, many with their own competitive circles and dedicated fan bases in which tracking the number and duration of playthroughs can be just as important as the game itself!

Gametyme makes it easier for gamers to keep track of the games they play as well as how long each playthrough takes. Gametyme has a large database of 1 million videogames and can support up to 500,000 users. To use our platform, users can simply search for a video game and start a new playthrough. As they make progress on their playthroughs, users can then report their hours by logging new sessions.

For pro members, Gametyme can be used to generate charts, trends, and analytical insights into their performance, which can be especially useful for those in the competitive speedrunning community. Gametyme also aims to foster a supportive gaming community by allowing users to view other users' lists and check out their stats!

II) Database Outline:

a) **Users**: Records the details of Users who have created an account and have logged in games & gameplay hours
   i) user_id: INT, auto_increment, unique, not NULL, PK
   ii) first_name: VARCHAR(45), not NULL
   iii) last_name: VARCHAR(45), not NULL
   iv) username: VARCHAR(45), not NULL
   v) email: VARCHAR(45), not NULL
   vi) **Relationship**:
      1) A 1:M relationship between Users and Playthroughs is implemented with user_id inside of Playthroughs. Users can log multiplePlaythroughs, and each Playthrough can only have one User.

b) **Games**: Records the details of games that users can track hours for or have played
   i) game_id: INT, auto_increment, unique, not NULL, PK
   ii) game_title: VARCHAR(100), not NULL
   iii) game_summary: VARCHAR(1000)
   iv) release_date: DATE
   v) company_id : INT, FK
   vi) genre_id: INT, FK
   vii) **Relationship**:
      1) A M:1 relationship between Games and Genres where genre_id from Genres is implemented as a FK in Games. Genres can have multiple Games, but Games will be limited to one Genre.
      2) A 1:M relationship between Games and Playthroughs is implemented with game_id inside of Playthroughs as a FK.
      3) A M:1 relationship between Games and Companies where company_id from Companies is implemented as a FK in Games. A Company can have produced multiple games, but a Game can only be produced by one Company
      4) A M:M relationship between Games and Platforms which is established through an intersection table named GamesPlatforms. GamesPlatforms will contain both game_id and platform_id as foreign keys. Games are available on multiple Platforms, and many different games can be available on the same Platform.

c) Playthroughs: Created when a User starts a new Playthrough of a Game. Used to track which Games a User has played, when they've started and finished a Game, how many times they've played it, and how many hours they've played it (by summing Session lengths).
   i) playthrough_id: INT, auto_increment, unique, not NULL, PK
   ii) start_timestamp: TIMESTAMP, not NULL
   iii) finish_timestamp: TIMESTAMP
   iv) user_id: INT, FK, not NULL
   v) game_id: INT, FK
   vi) **Relationships:**
      1) A M:1 relationship between Playthroughs and Users where user_id from Users is implemented as a FK in Playthroughs. Users can start multiple Playthroughs, but each Playthrough is only associated with one User.
      2) A M:1 relationship between Playthroughs and Games where game_id from Games is implemented as a FK in Playthroughs.

There can be multiple Playthroughs for the same Game, but only one Game per Playthrough.

3) A 1:M relationship between Playthroughs and Sessions where playthrough_id from Playthroughs is implemented as a FK in Sessions. A Playthrough can be divided across multiple Sessions, but each Session can only be for one Playthrough.

d) **Sessions**: Created whenever a User logs a new session for a particular Playthrough. Records the corresponding Playthrough, the start date and time, and eventually the finish date and time.
   i) session_id: INT, auto_increment, unique, not NULL, PK
   ii) session_start: TIMESTAMP, not NULL
   iii) session_end: TIMESTAMP
   iv) playthrough_id: INT, FK, not NULL
   v) **Relationship**:
      1) A M:1 relationship between Sessions and Playthroughs where playthrough_id from Playthroughs is a FK in Sessions. The M:1 relationship between Sessions and Playthroughs is implemented and established through an intersection table. Playthroughs can be divided into multiple Sessions, but each Session can only belong to one Playthrough.

e) **Genres**: Category entity listing possible genres for Games.
   i) genre_id: INT, auto_increment, unique, not NULL, PK
   ii) genre_name: VARCHAR(45), not NULL
   iii) **Relationship**:
      1) A 1:M relationship between Genres and Games. Games can only be one Genre, but the same Genre can be used for multiple Games.

f) **Companies**: Records the information about each game development studio.
   i) company_id: INT, auto_increment, unique, not NULL, PK
   ii) company_name: VARCHAR(100), not NULL
   iii) location_id: VARCHAR(45), FK
   iv) **Relationship**:
      1) A 1:M relationship between Companies and Games. A Company can produce multiple Games.
      2) A M:1 relationship between Companies and Locations where location_id from Locations is used as a FK in Companies. There

are multiple game companies, but each company has one location for their headquarters.

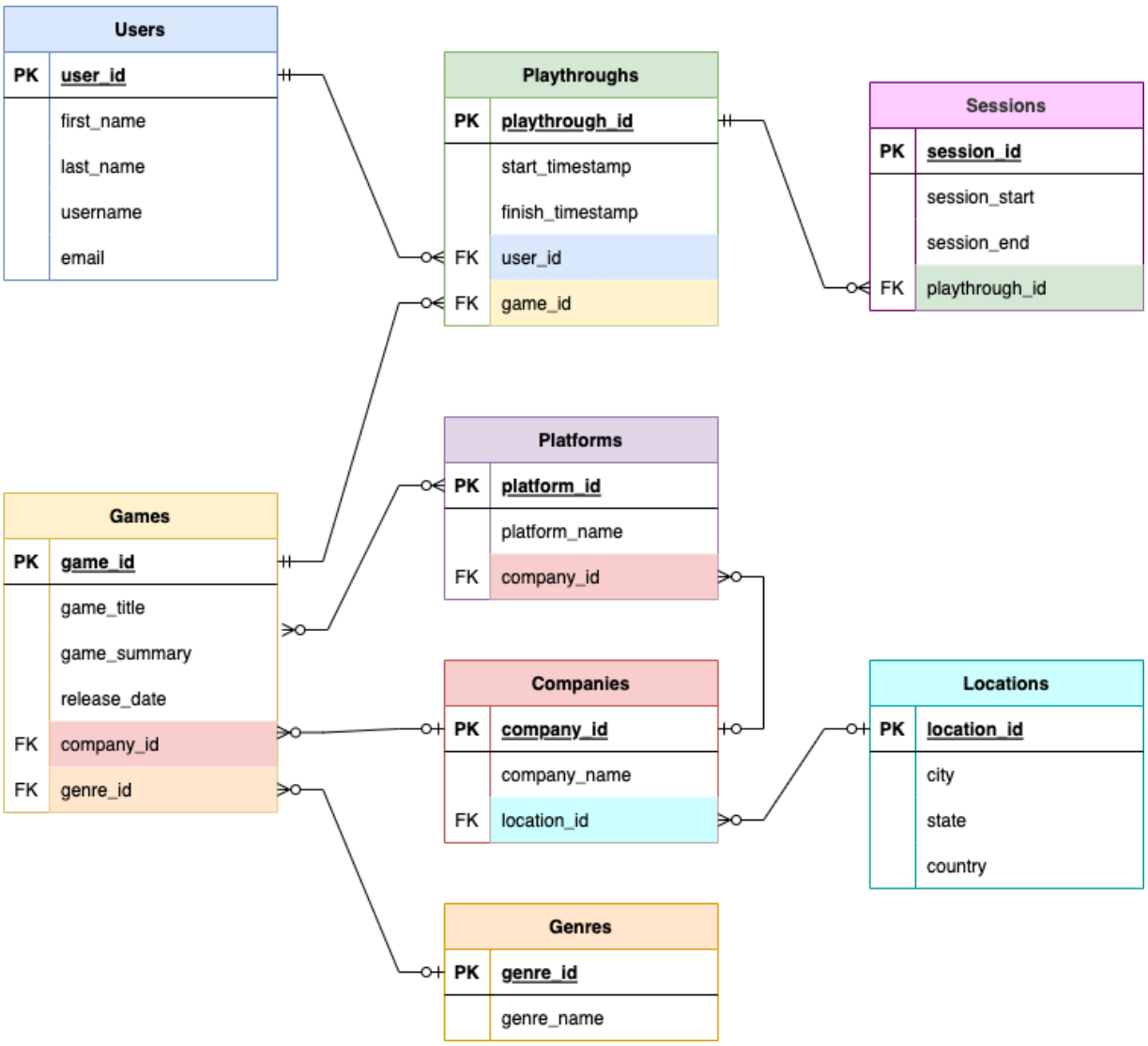
g) **Platforms**: Category entity listing possible gaming platforms.
   i) platform_id: INT, auto_increment, unique, not NULL, PK
   ii) platform_name: VARCHAR(45), not NULL
   iii) company_id: VARCHAR(45), FK
   iv) **Relationship**:
      1) A M:M relationship between Platforms and Games which is established through an intersection table named GamesPlatforms. GamesPlatforms will contain both game_id and platform_id as foreign keys. Games can be on multiple platforms, and the same Platform can be associated with many different Games.
      2) A M:1 relationship between Platforms and Companies, where company_id from Companies is used as a FK in Platforms. A single Company can create many different Platforms, but a given Platform must only belong to a single Company.

h) **Locations**: Category entity listing the locations of game companies.
   i) location_id :INT, not NULL, AUTO INCREMENT, unique, PK
   ii) city: VARCHAR(45), not NULL
   iii) state: VARCHAR(45)
   iv) country : VARCHAR(45), not NULL
   v) **Relationship**:
      1) A 1:M relationship between Locations and Companies where location_id from Locations is used as a FK in Companies. There are multiple game companies, but each company has one location for their headquarters.

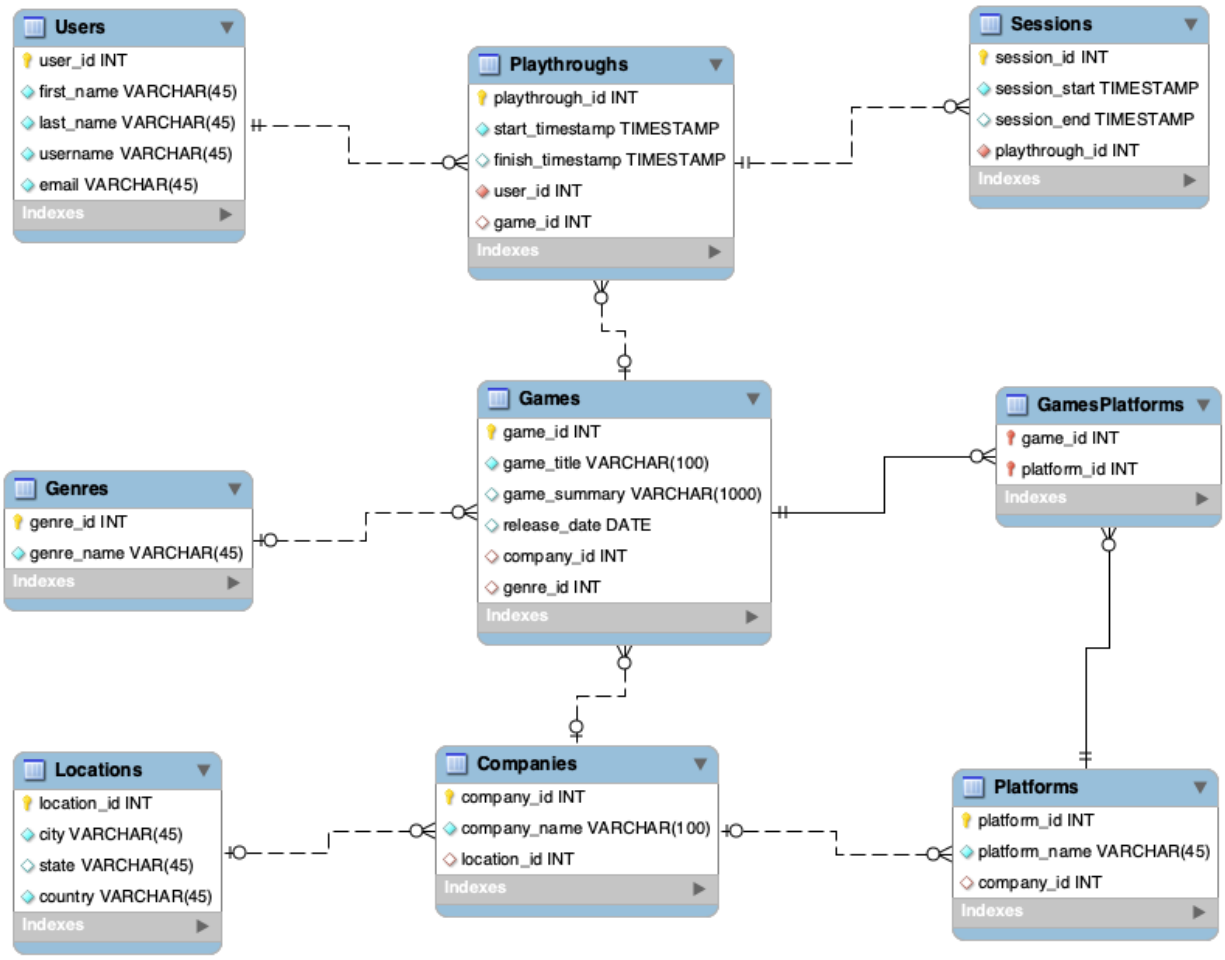
Entity Table Assignments:

- Eli Mills - Genres, Companies, Platforms, Locations
- Na Kim - Users, Games, Playthroughs, Sessions

III) Entity-Relationship Diagram



IV) Schema


Users
user_id INT
first_name VARCHAR(45)
last_name VARCHAR(45)
username VARCHAR(45)
email VARCHAR(45)
Indexes
Playthroughs
playthrough_id INT
start_timestamp TIMESTAMP
finish_timestamp TIMESTAMP
user_id INT
game_id INT
Indexes
Sessions
session_id INT
session_start TIMESTAMP
session_end TIMESTAMP
playthrough_id INT
Indexes
Games
game_id INT
game_title VARCHAR(100)
game_summary VARCHAR(1000)
release_date DATE
company_id INT
genre_id INT
Indexes
GamesPlatforms
game_id INT
platform_id INT
Indexes
Genres
genre_id INT
genre_name VARCHAR(45)
Indexes
Locations
location_id INT
city VARCHAR(45)
state VARCHAR(45)
country VARCHAR(45)
Indexes
Companies
company_id INT
company_name VARCHAR(100)
location_id INT
Indexes
Platforms
platform_id INT
platform_name VARCHAR(45)
company_id INT
Indexes

V) Example Data

Users

user_id	first_name	last_name	username	email
1	Eren	Yeager	Titan	fightme@gmail.com
2	Mikasa	Ackerman	Erennn	ackerman@gmail.com
3	Link	Zelda	Korok	zelda@gmail.com

Games

game_id	game_title	game_summary	release_date	company_id	genre_id
1	Elden Ring	This is an open world dark souls like game where you also die a lot	2022-02-25	1	1
2	The Legend of Zelda: Breath of the Wild	This is an open world Zelda game where you hunt koroks	2017-03-03	2	2
3	Overwatch	This is a multiplayer shooting game where everyone needs healing too much	2016-05-24	3	3

Playthroughs

playthrough_id	start_timestamp	finish_timestamp	user_id	game_id
1	2022-01-04 02:15:01	NULL	1	1
2	2021-11-01 01:30:40	2022-01-12 10:24:10	2	3
3	2020-05-20 04:20:32	2020-12-01 10:02:01	3	2

Sessions

session_id	session_start	session_end	playthrough_id
1	2020-12-11 11:01:03	2020-12-11 21:01:03	3
2	2022-02-01 02:02:13	2022-02-01 03:02:13	2
3	2022-04-11 05:09:23	2022-04-11 10:09:23	1

Genres

genre_id	genre_name
1	action
2	adventure
3	shooter

Companies

company_id	company_name	location_id
3	Blizzard	3
1	FromSoftware	1
5	Microsoft	4
2	Nintendo	2
4	Sony	1

Platforms

platform_id	platform_name	company_id
1	Switch	2
2	PC	5
3	Playstation 4	4
4	Playstation 5	4
5	Xbox One	5
6	Xbox Series X|S	5

Locations

location_id	city	state	country
1	Tokyo	NULL	Japan
2	Kyoto	NULL	Japan
3	Irvine	California	United States
4	Redmond	Washington	United States

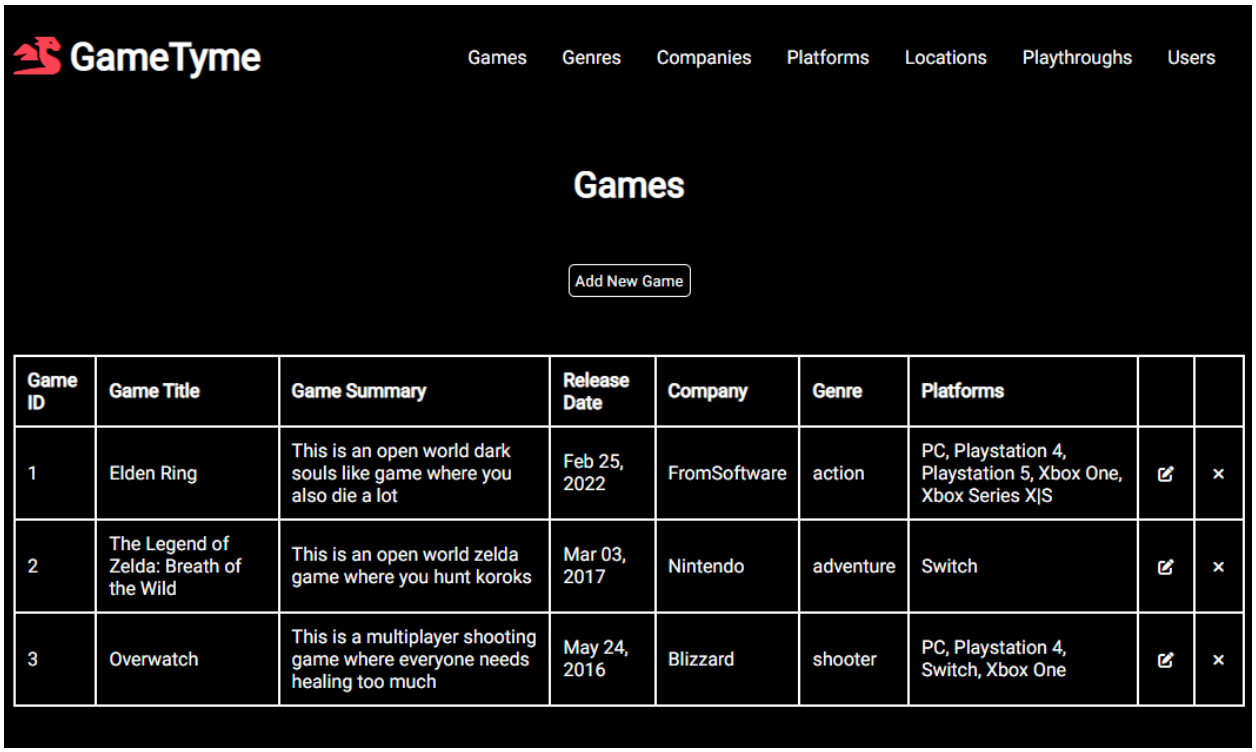
GamesPlatforms

game_id	platform_id
1	2
1	3
1	4
1	5
1	6
2	1
3	1
3	2
3	3
3	5

VI) Screen Captures

Games

READ/BROWSE/DISPLAY Games Page

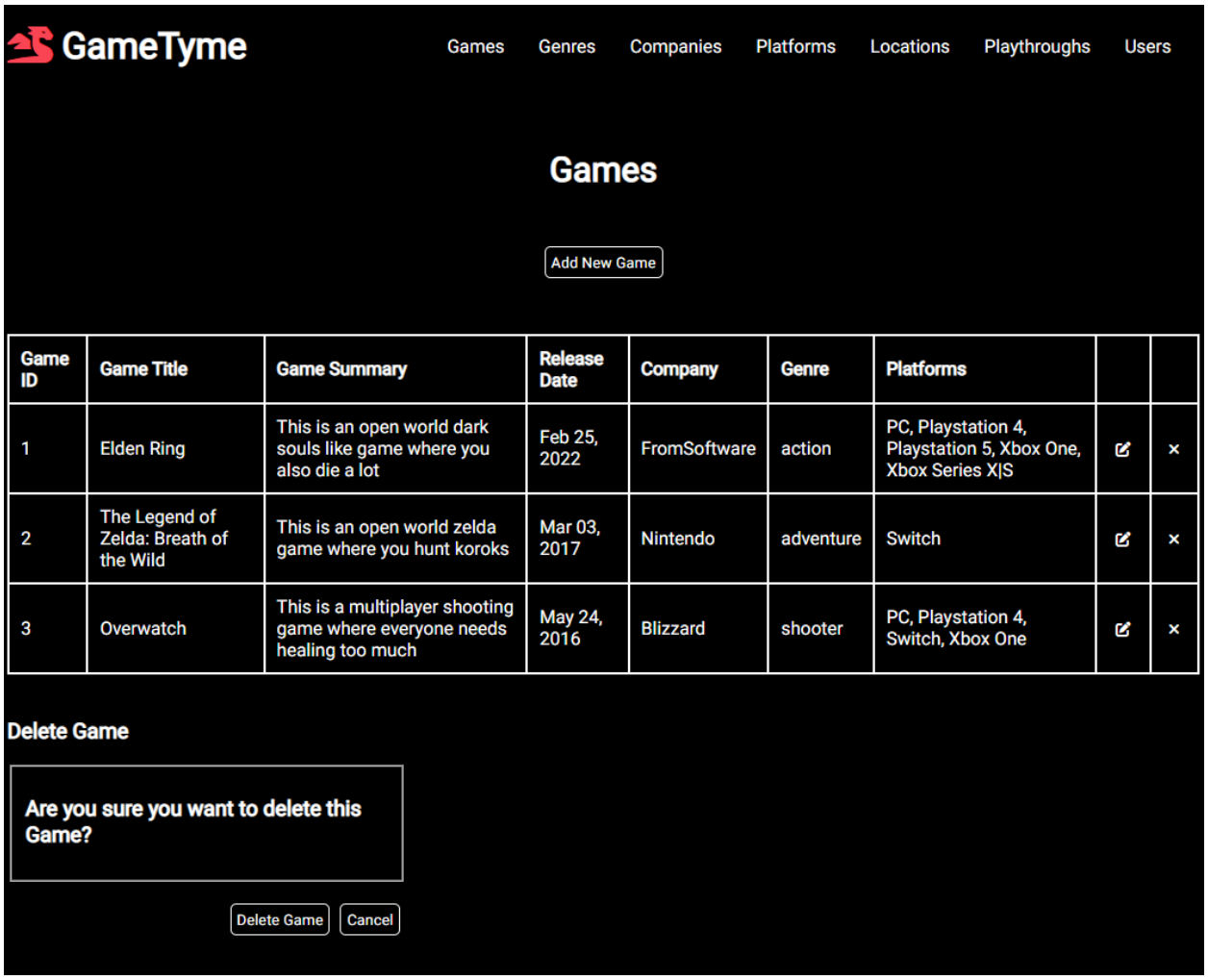


Game ID	Game Title	Game Summary	Release Date	Company	Genre	Platforms		
1	Elden Ring	This is an open world dark souls like game where you also die a lot	Feb 25, 2022	FromSoftware	action	PC, Playstation 4, Playstation 5, Xbox One, Xbox Series X|S	✎	×
2	The Legend of Zelda: Breath of the Wild	This is an open world zelda game where you hunt koroks	Mar 03, 2017	Nintendo	adventure	Switch	✎	×
3	Overwatch	This is a multiplayer shooting game where everyone needs healing too much	May 24, 2016	Blizzard	shooter	PC, Playstation 4, Switch, Xbox One	✎	×

CREATE/INSERT/ADD NEW Game Page

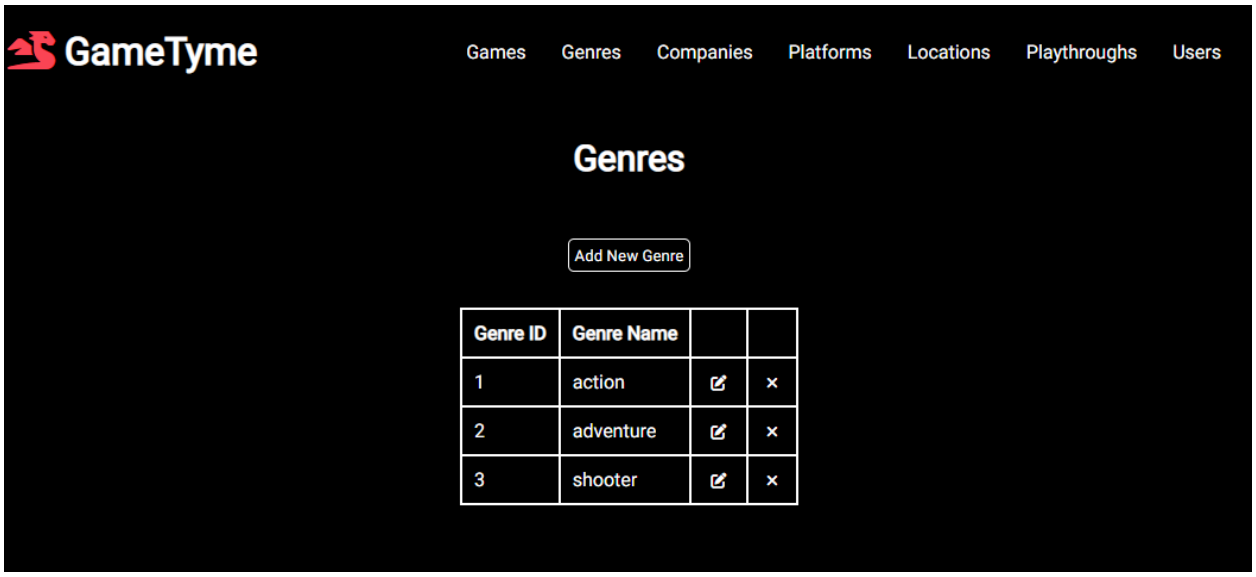
EDIT/UPDATE Game Page

DELETE Game Page

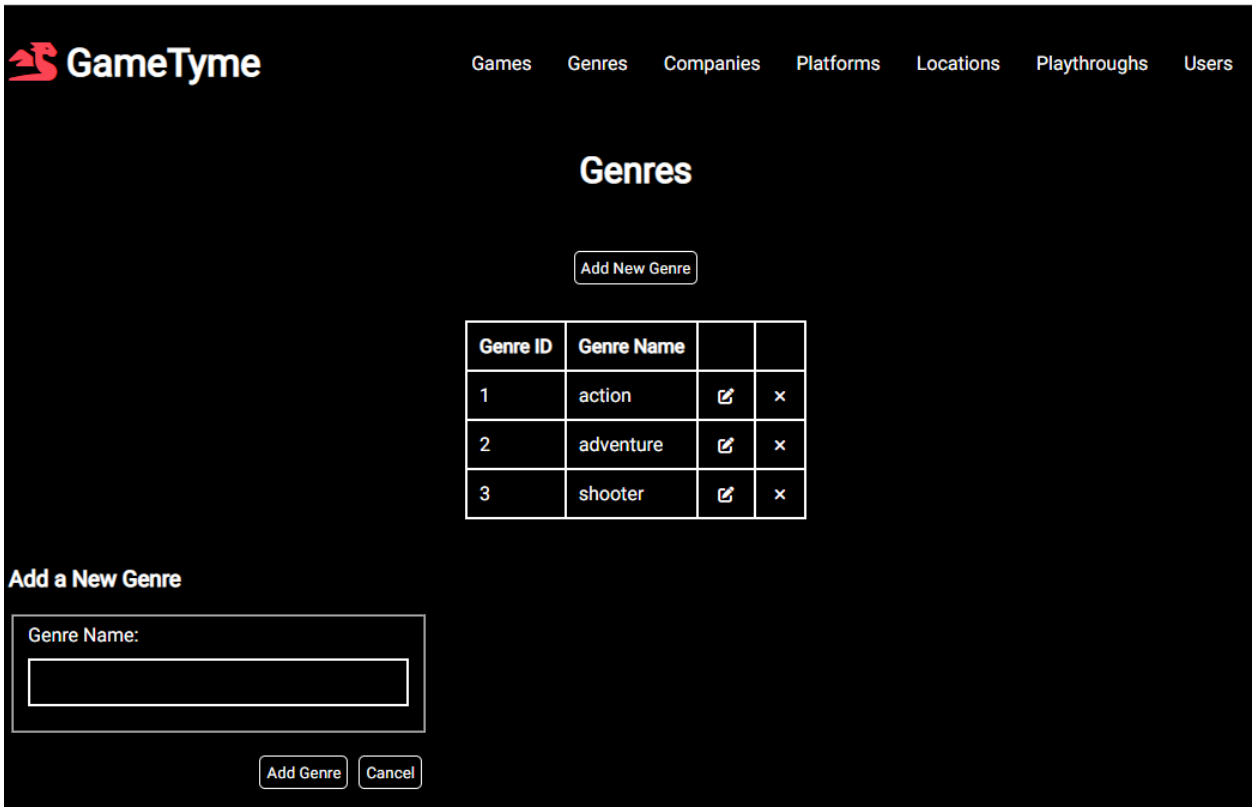


Genres

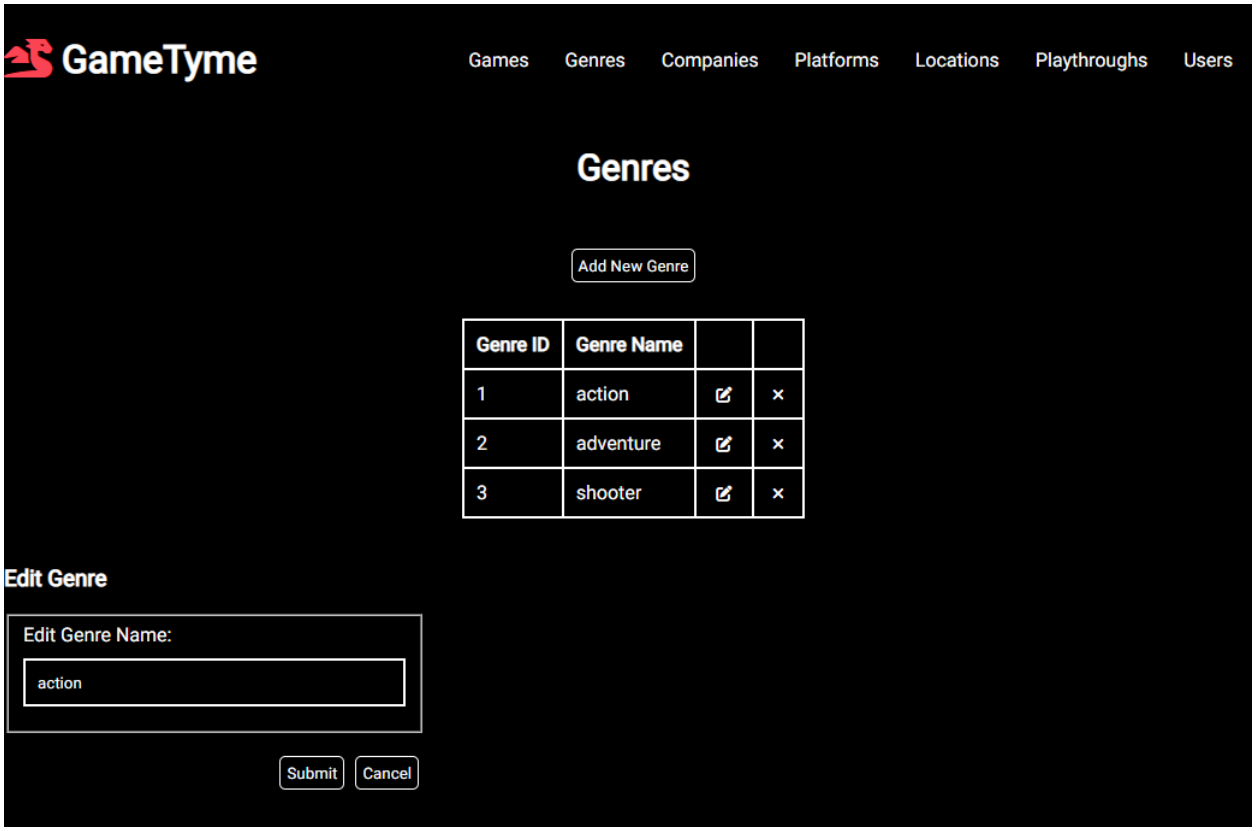
READ/BROWSE/DISPLAY Genres Page



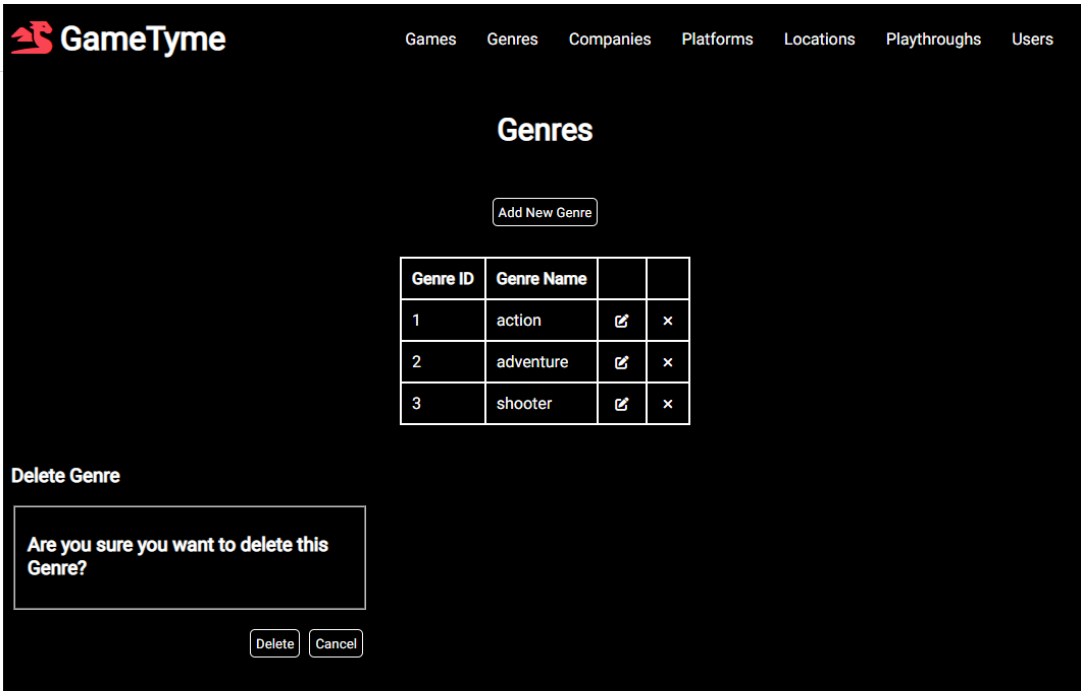
CREATE/INSERT/ADD NEW Genre Page



EDIT/UPDATE Genre Page

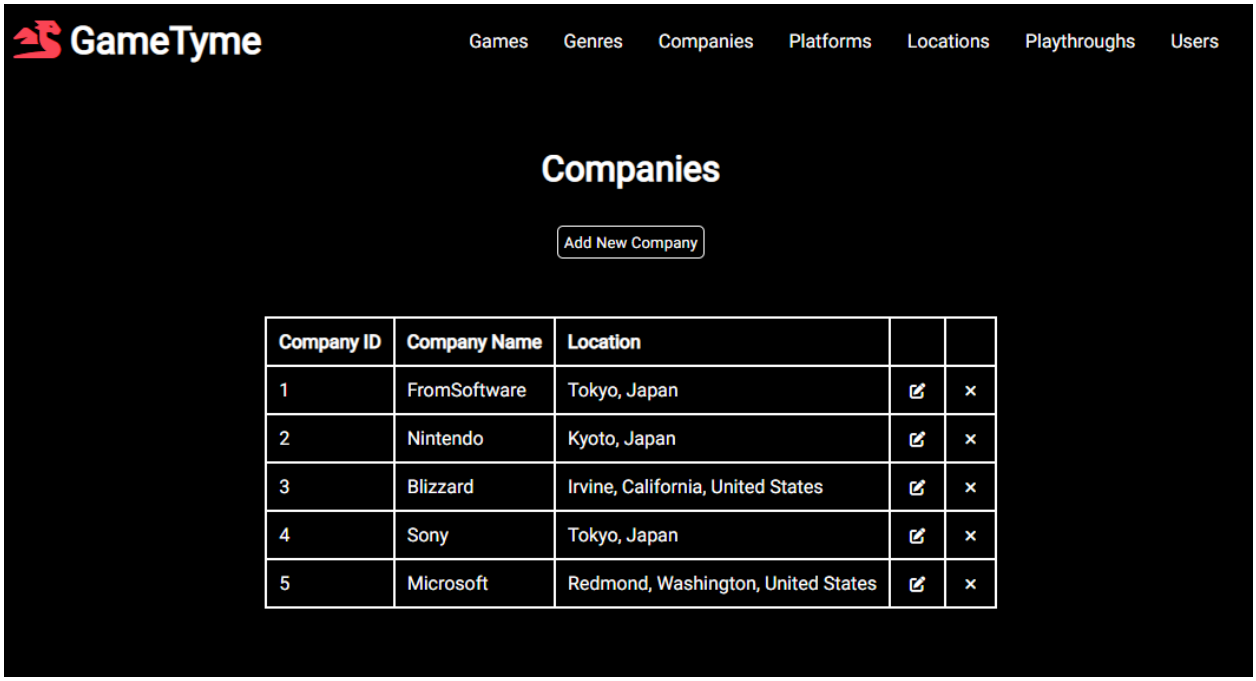


DELETE Genre Page

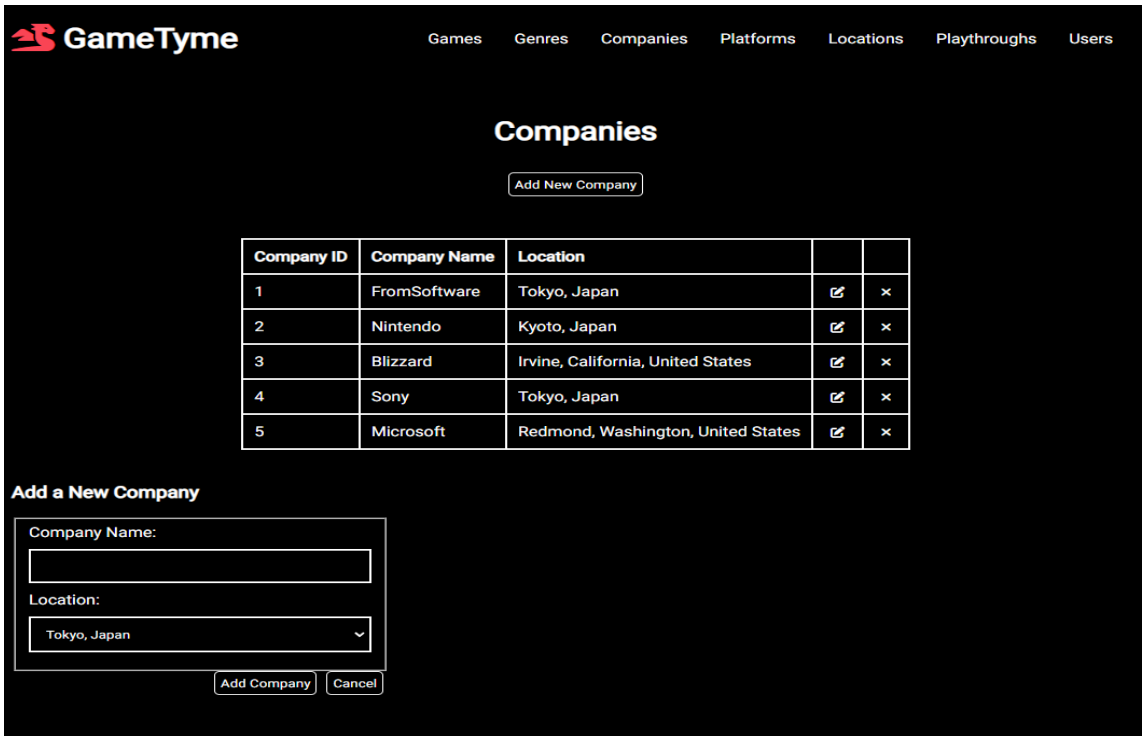


Companies

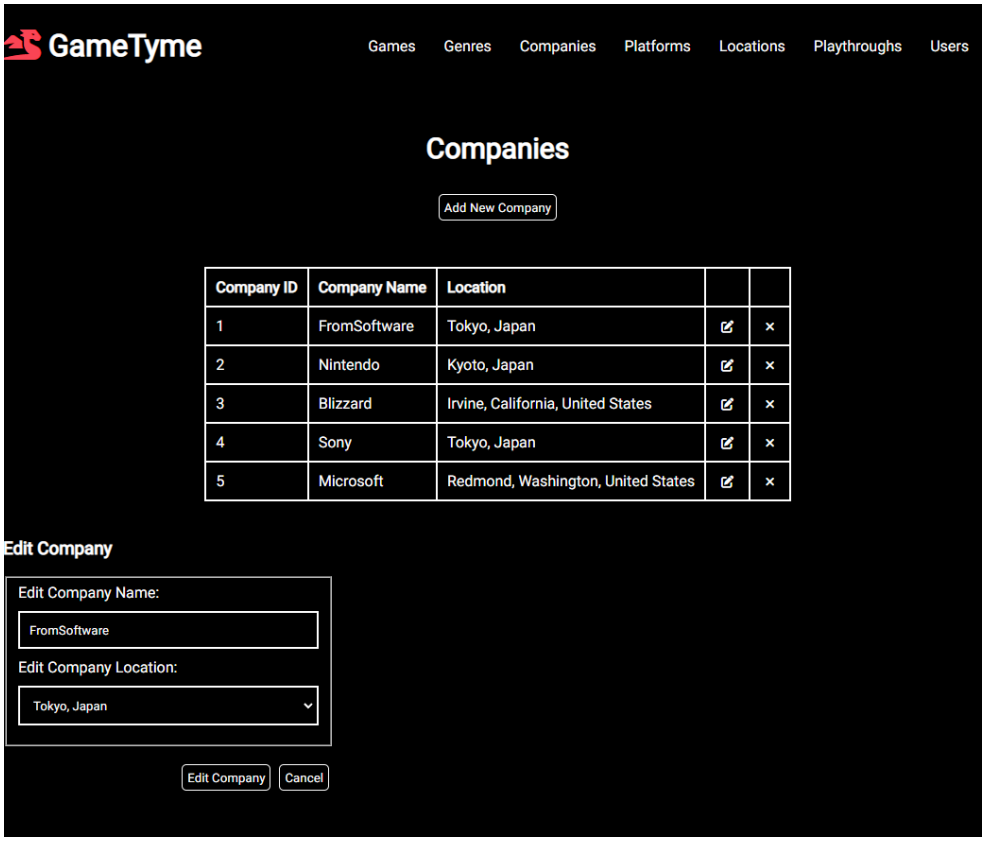
READ/BROWSE/DISPLAY Companies Page



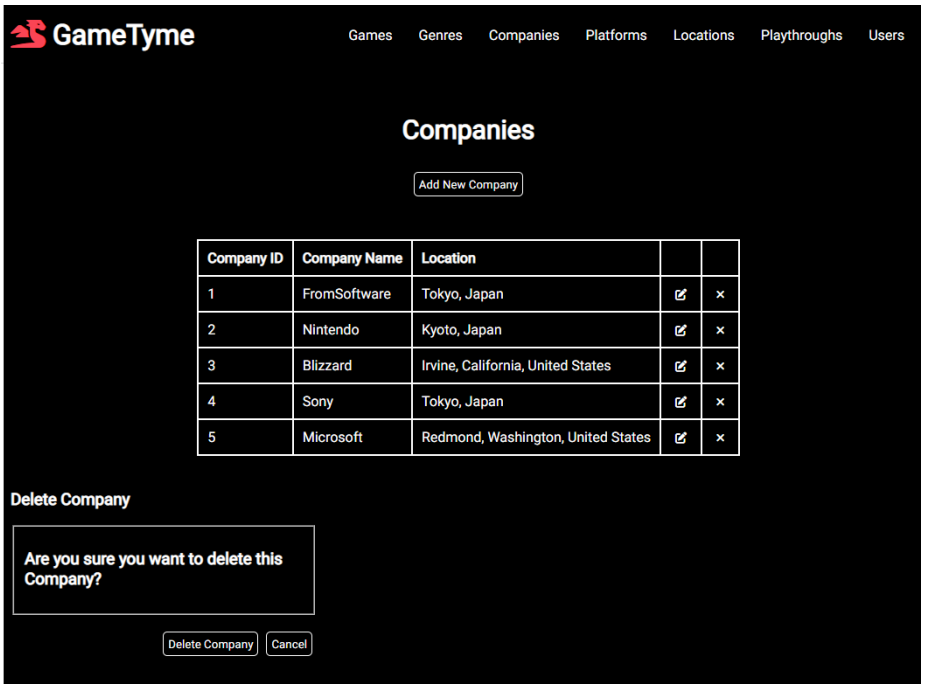
CREATE/INSERT/ADD NEW Companies Page



EDIT/UPDATE Companies Page

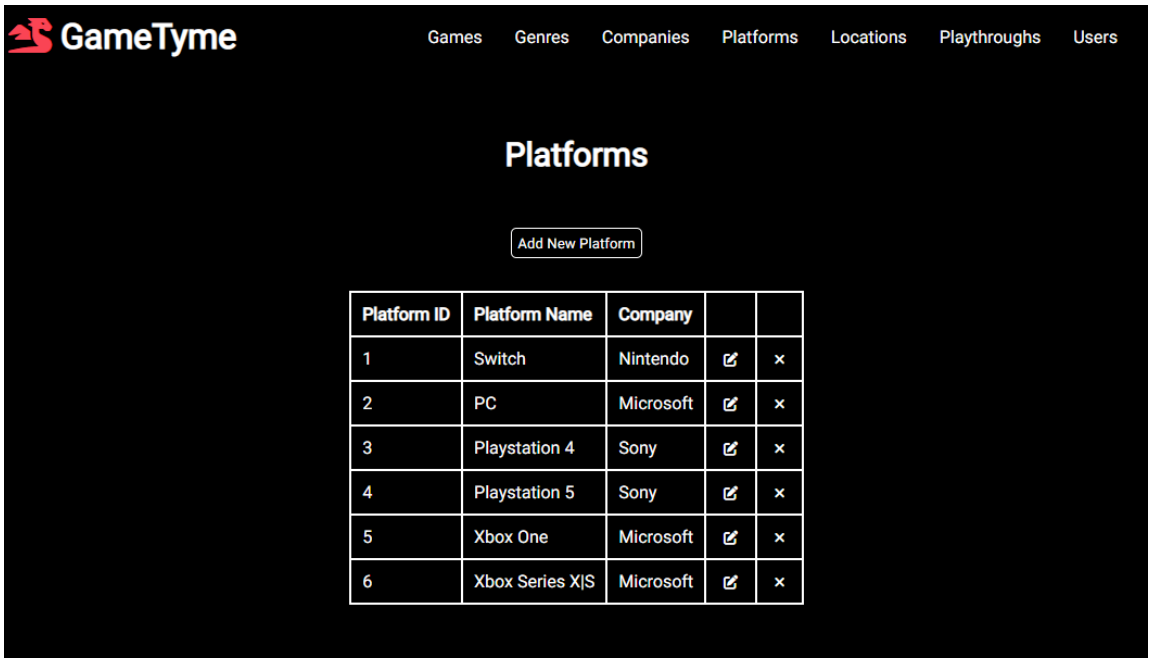


DELETE Companies Page

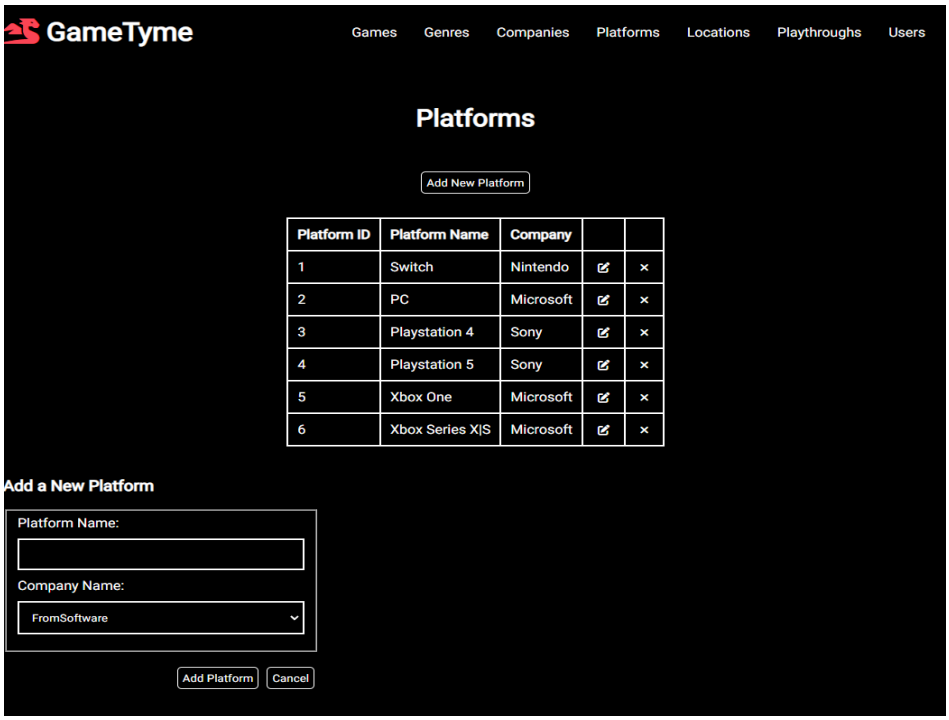


Platforms

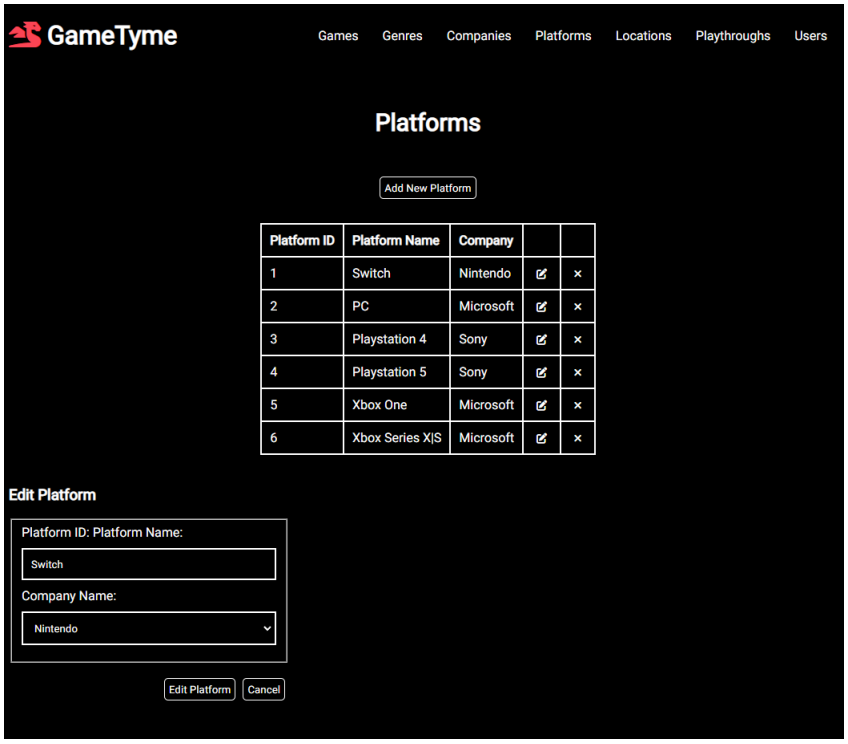
READ/BROWSE/DISPLAY Platforms Page



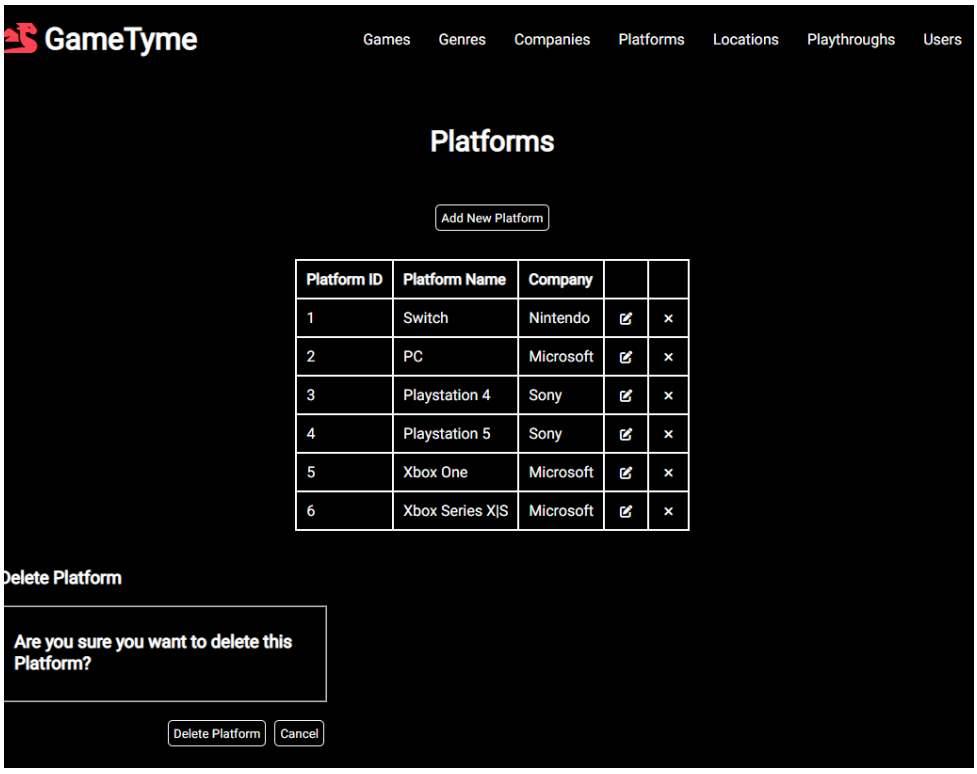
CREATE/INSERT/ADD NEW Platforms Page



EDIT/UPDATE Platforms Page

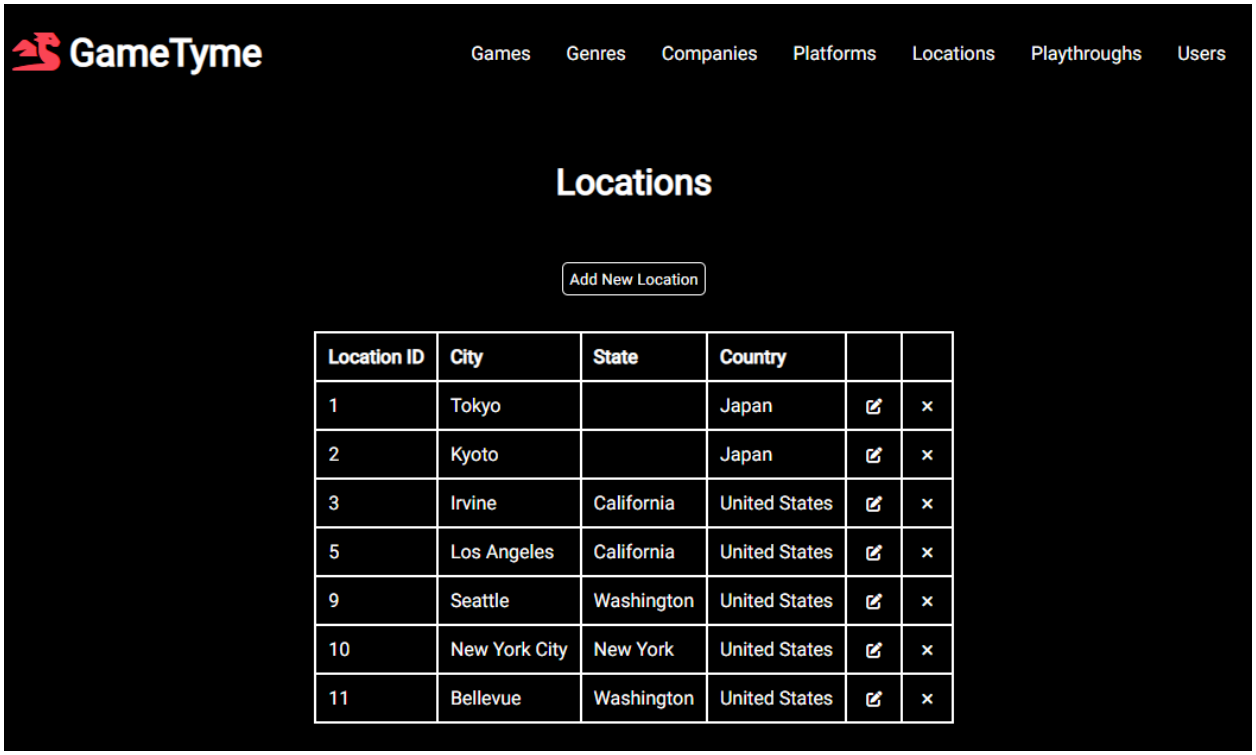


DELETE Platforms Page

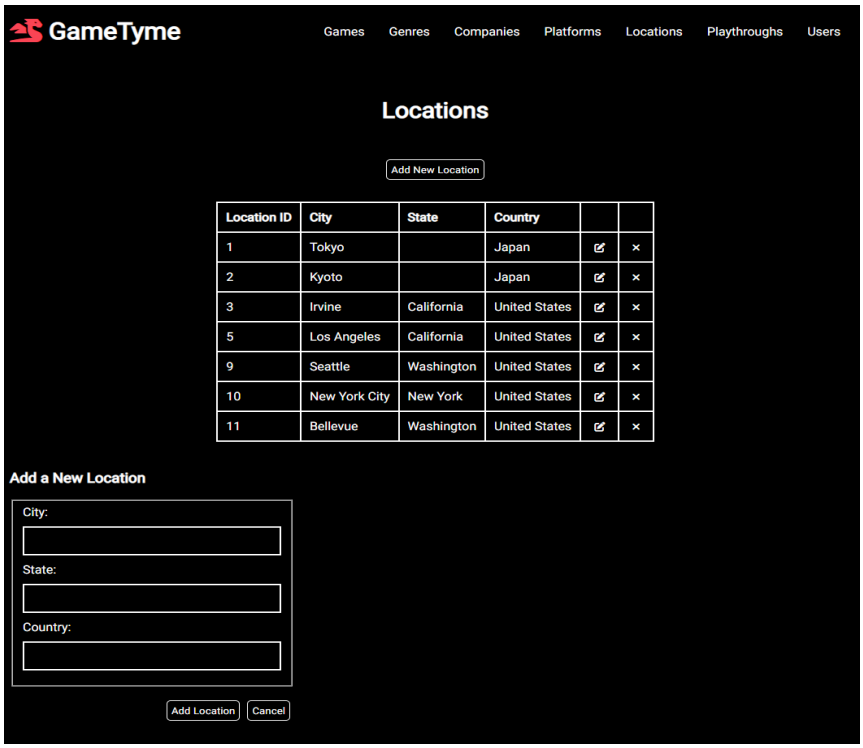


Locations

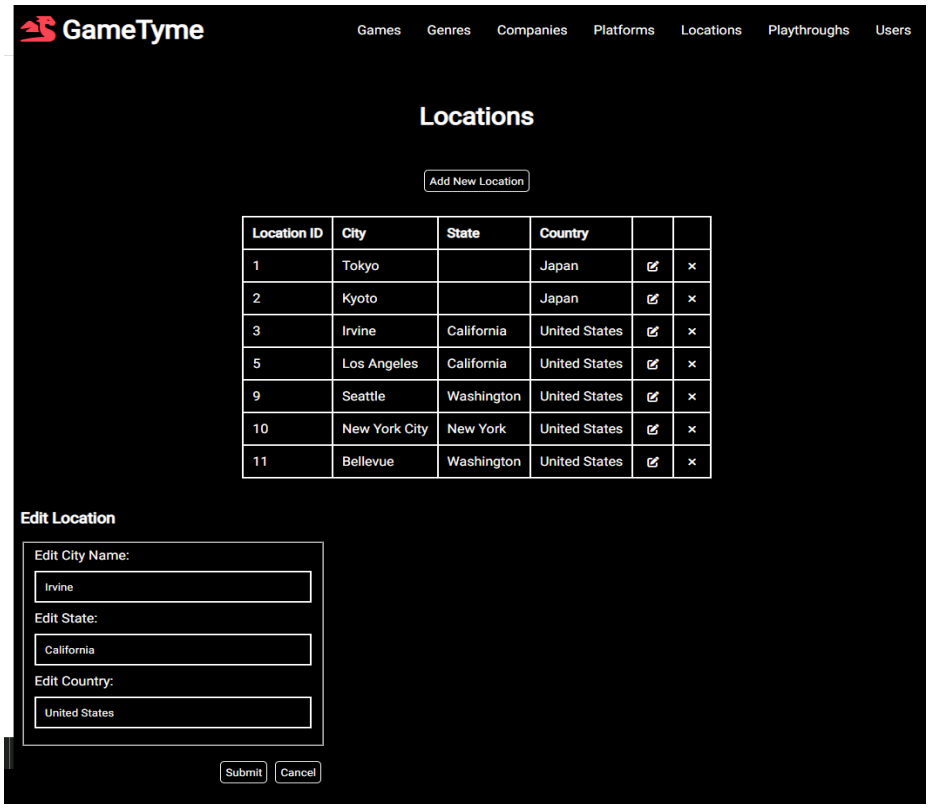
READ/BROWSE/DISPLAY Locations Page



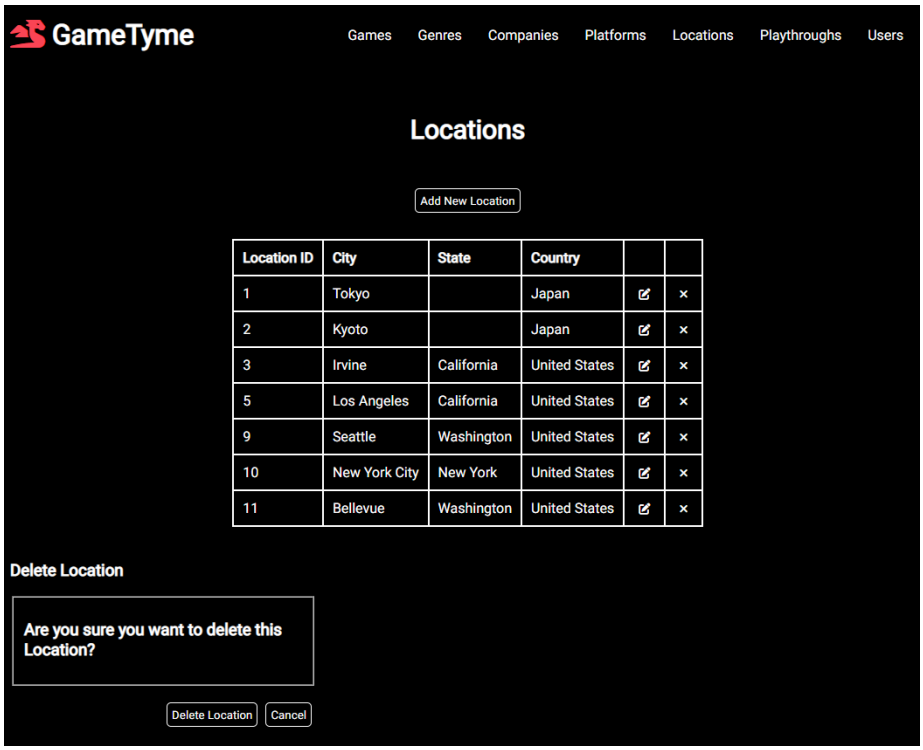
CREATE/INSERT/ADD NEW Locations Page



EDIT/UPDATE Locations Page

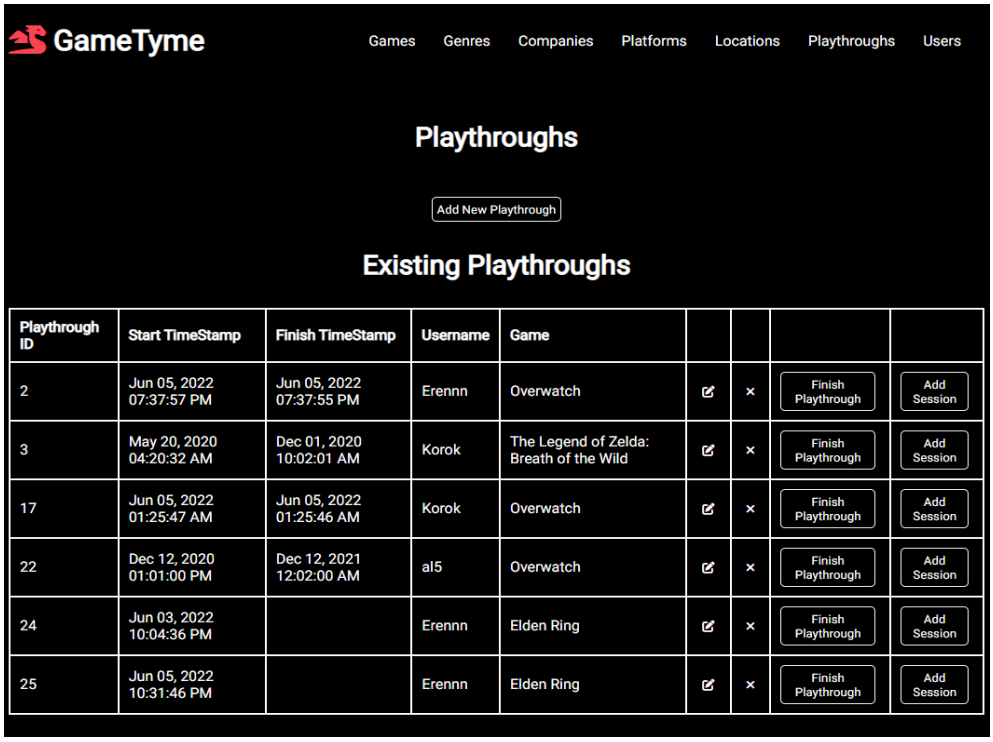


DELETE Locations Page

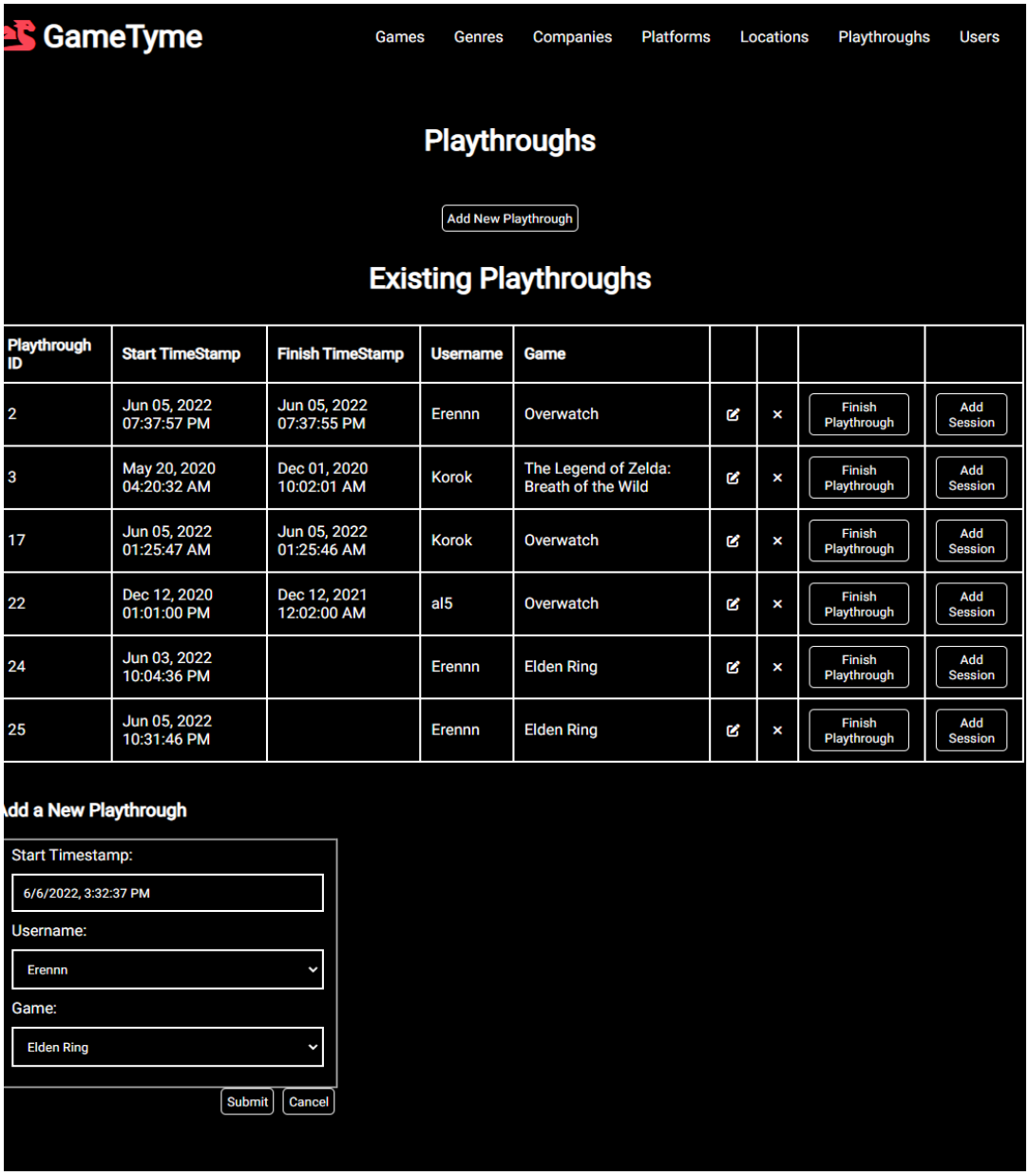


Playthroughs

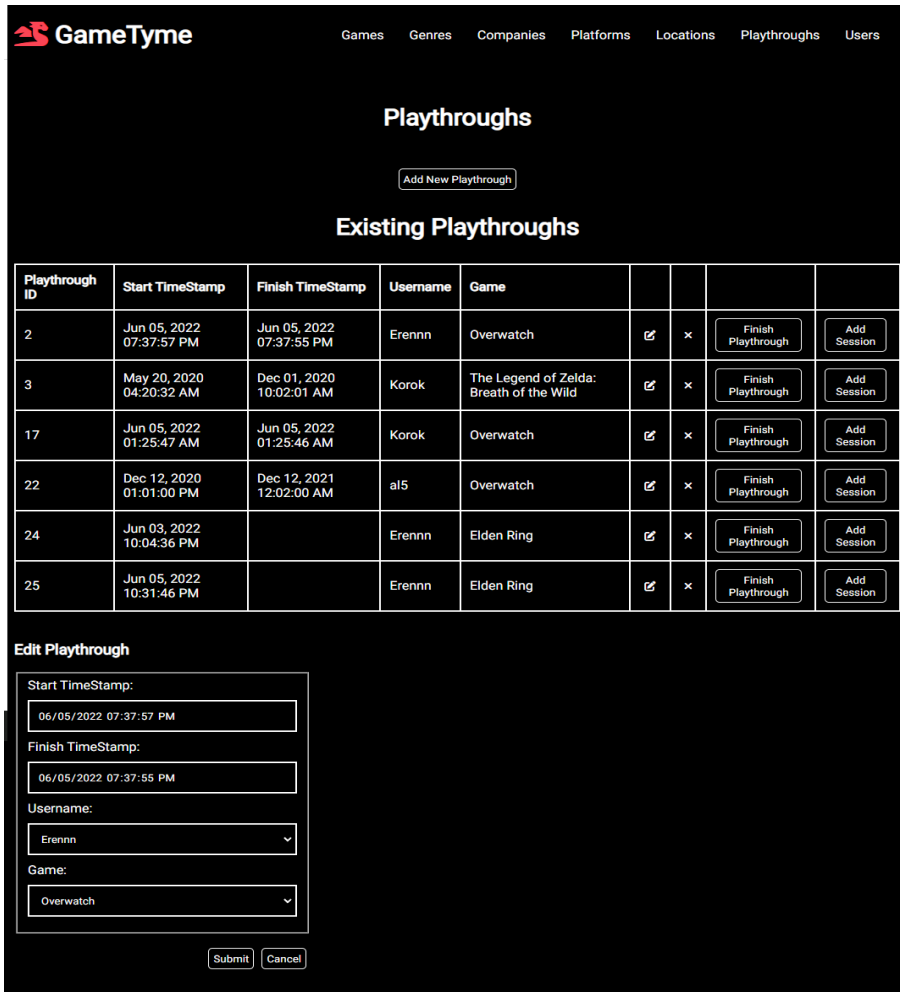
READ/BROWSE/DISPLAY Playthroughs Page

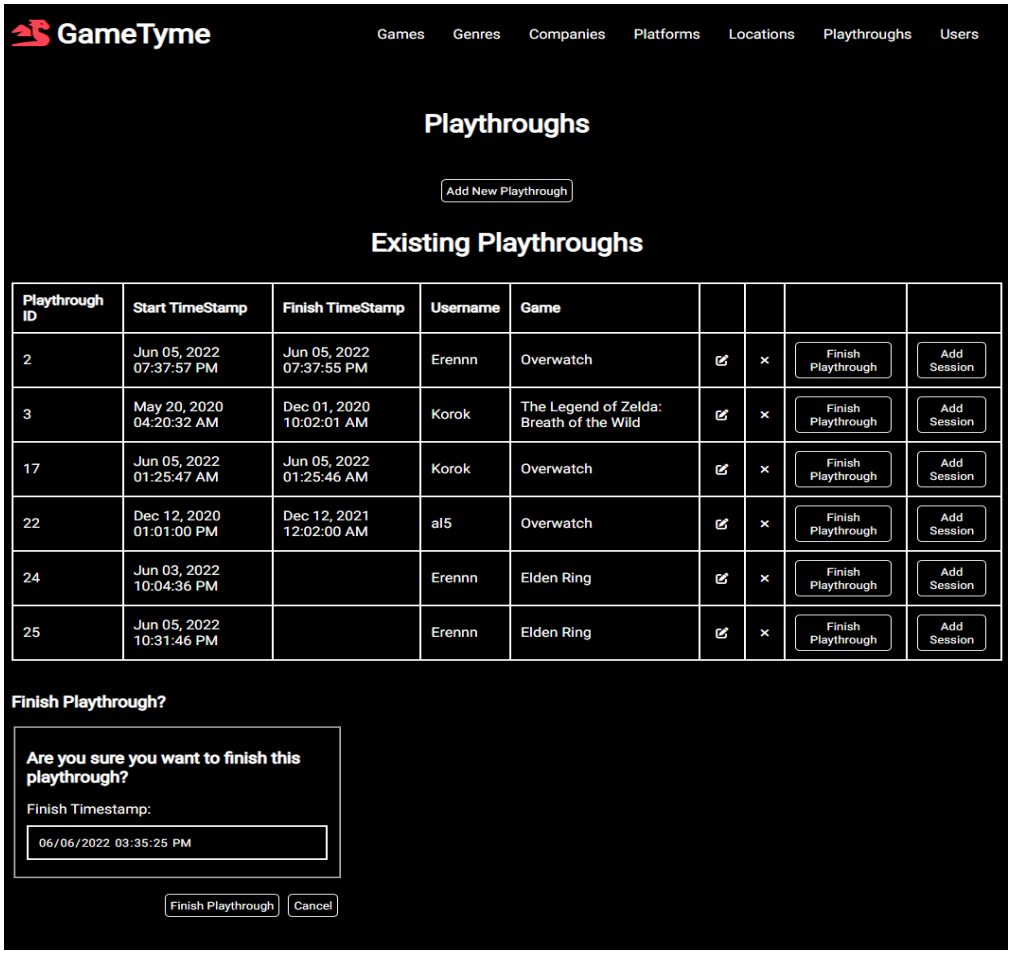


CREATE/INSERT/ADD NEW Playthroughs Page

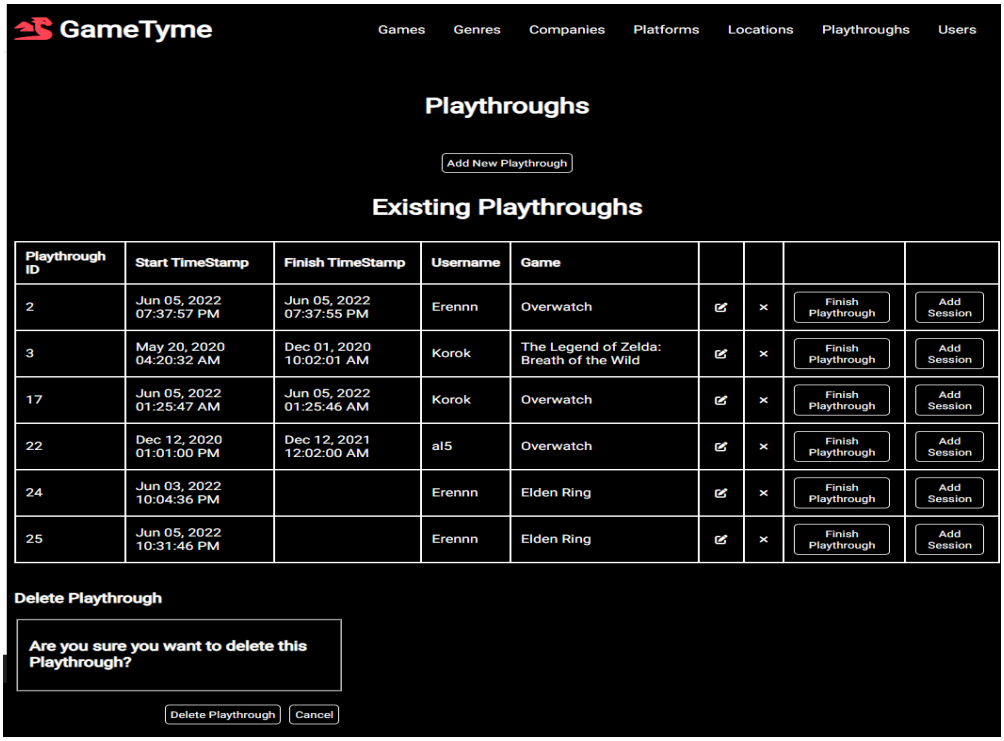


EDIT/UPDATE Playthroughs Page





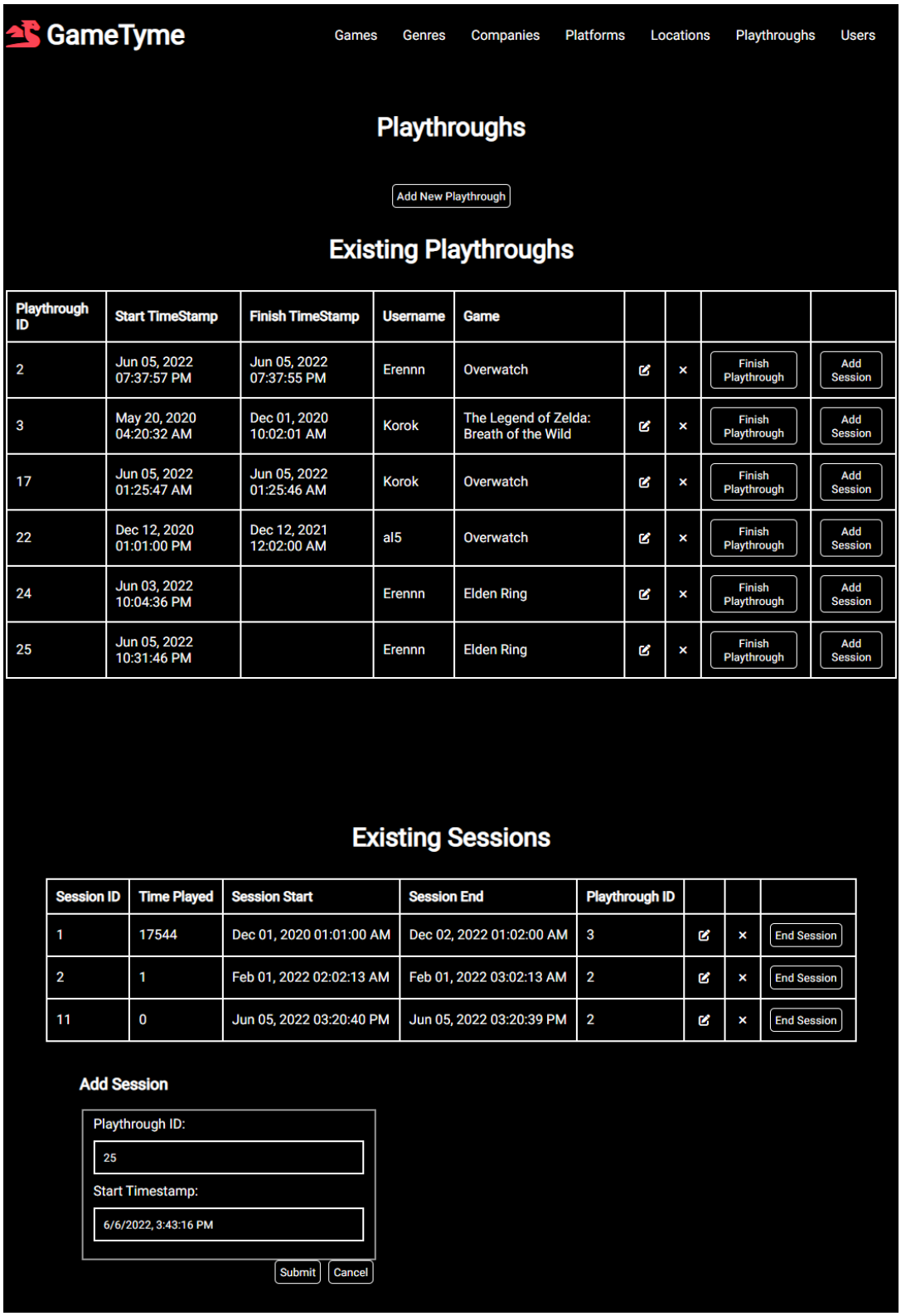
DELETE Playthroughs Page



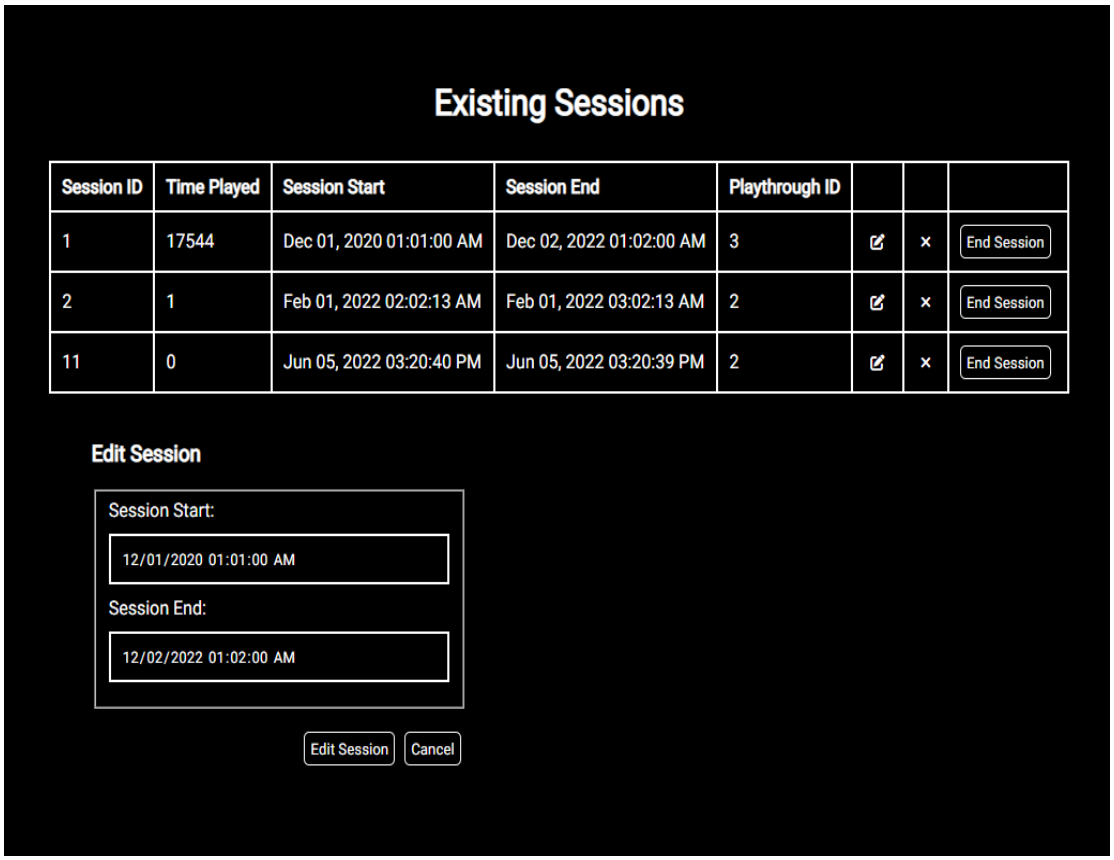
Sessions

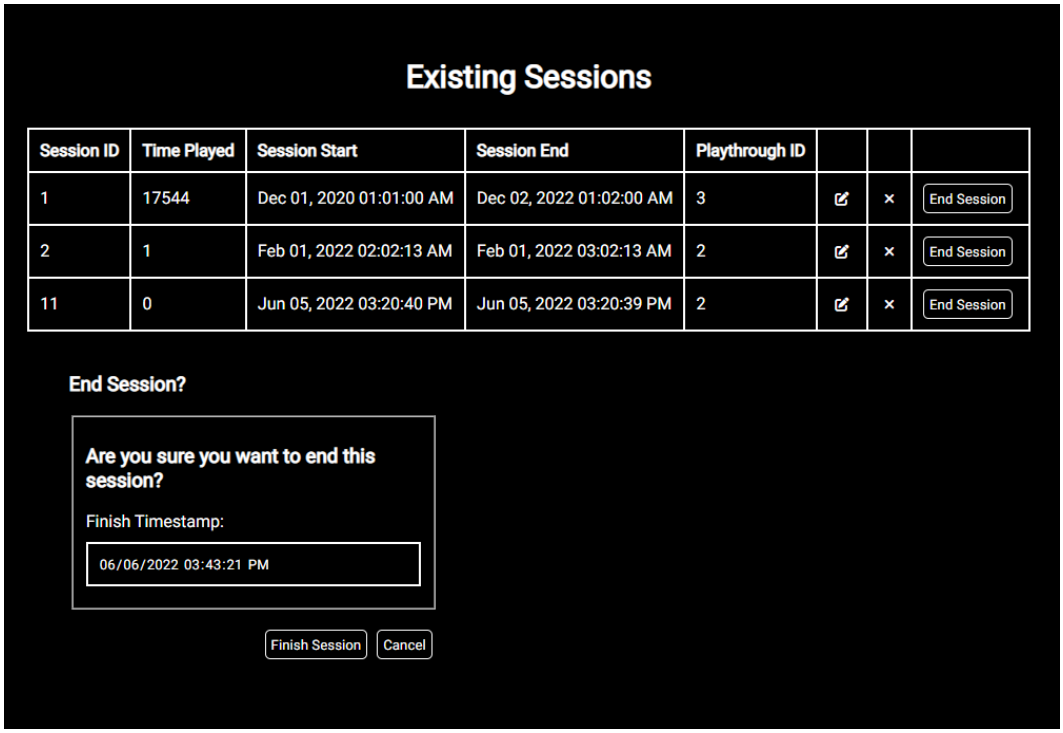
READ/BROWSE/DISPLAY Sessions Page

CREATE/INSERT/ADD NEW Sessions Page

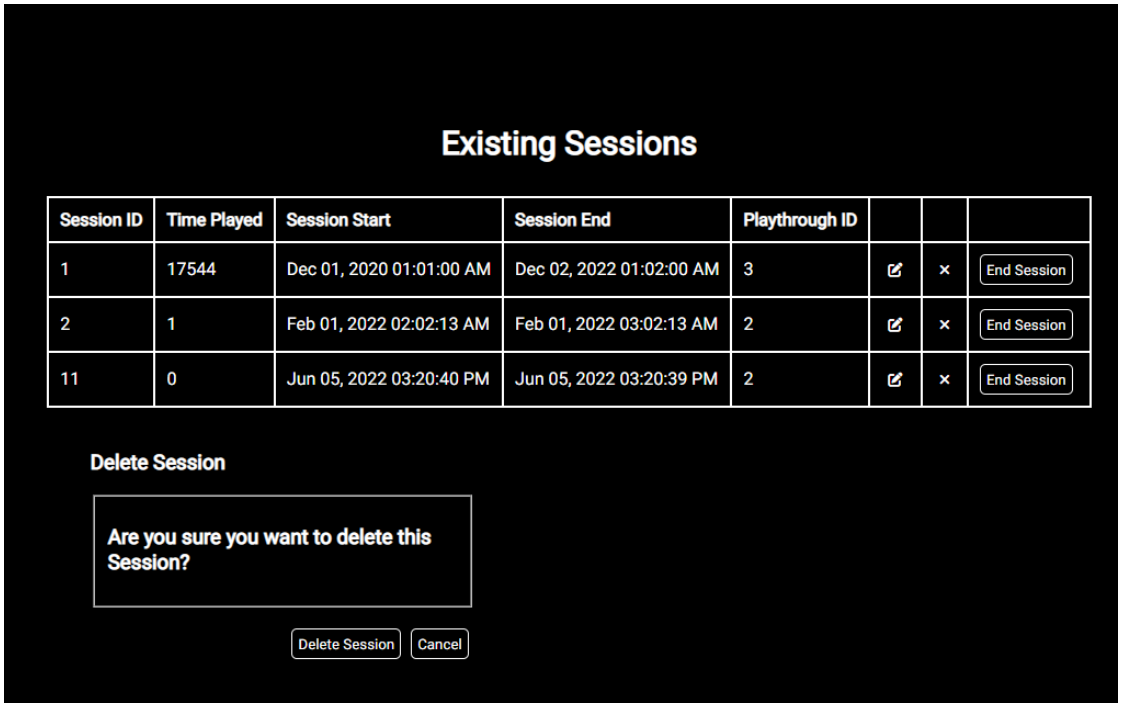


EDIT/UPDATE Sessions Page



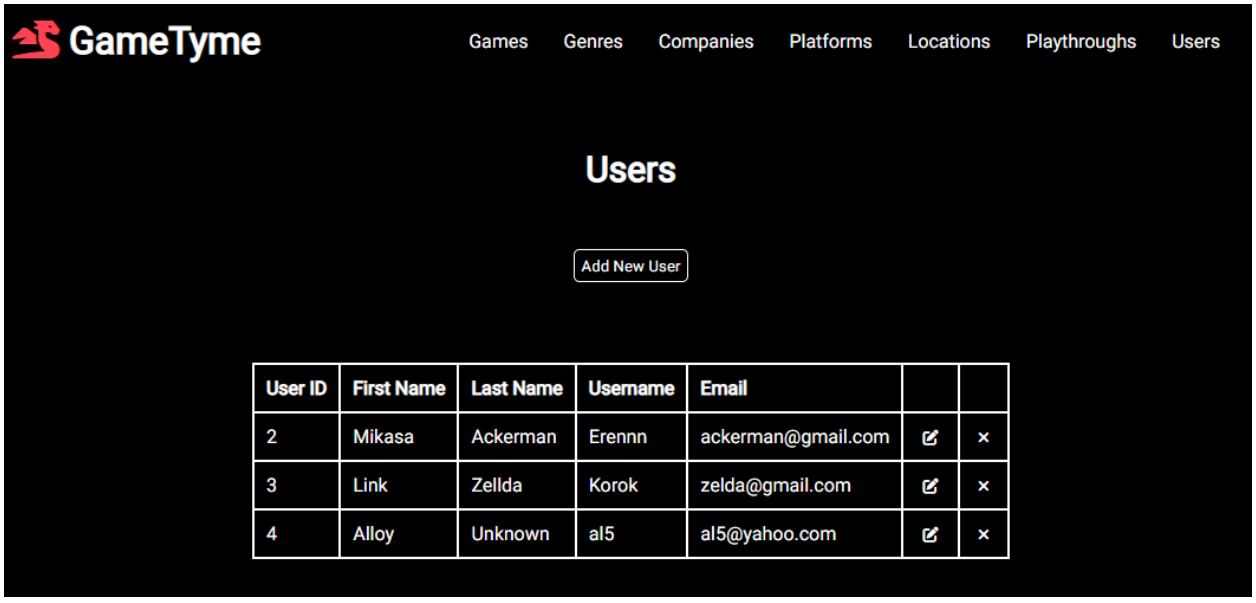


DELETE Sessions Page

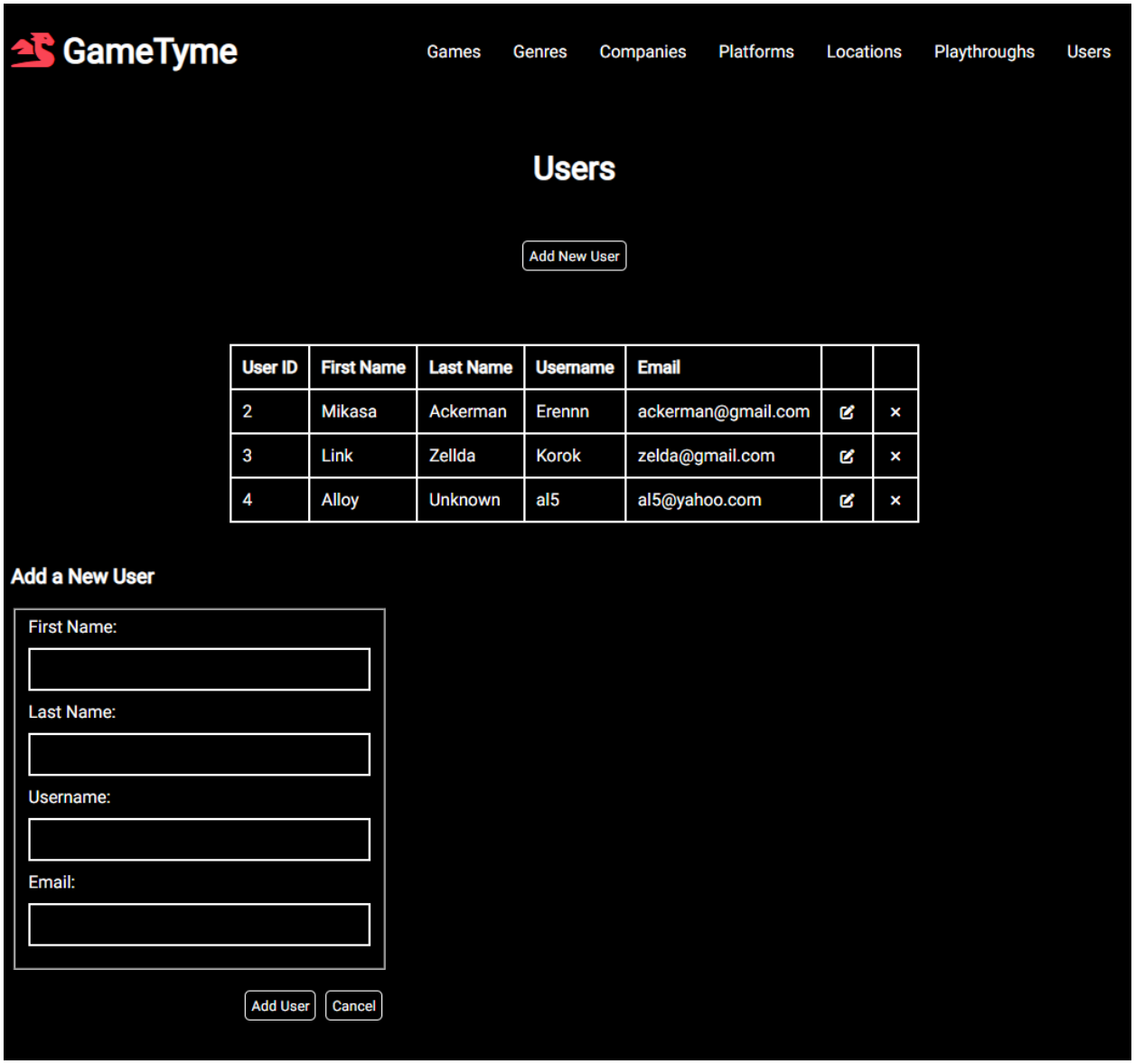


Users

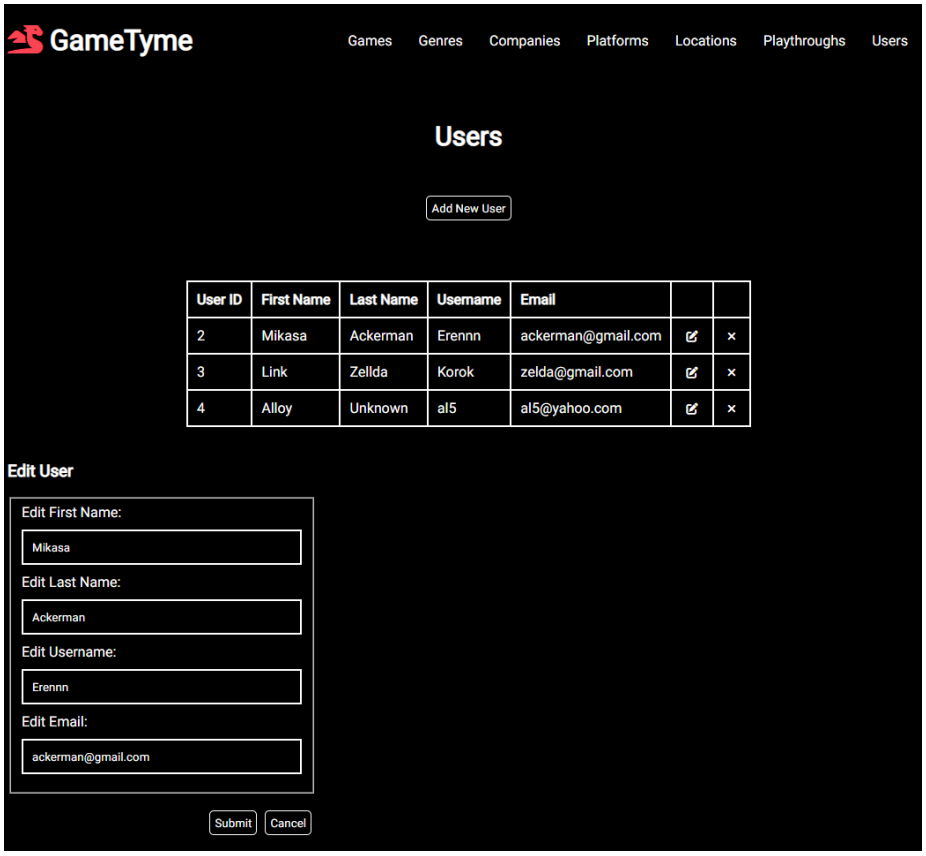
READ/BROWSE/DISPLAY Users Page



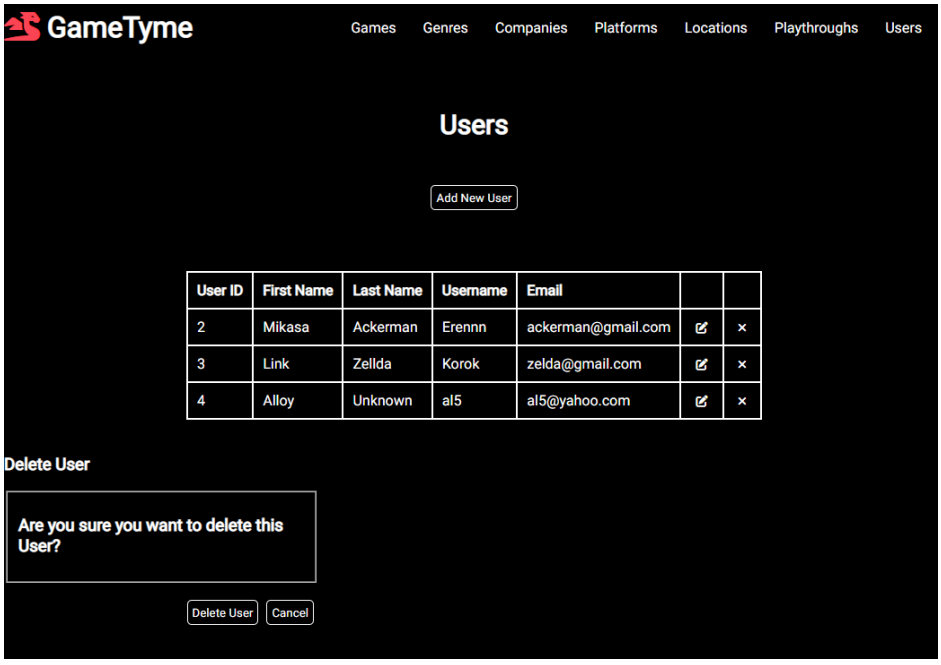
CREATE/INSERT/ADD NEW Users Page



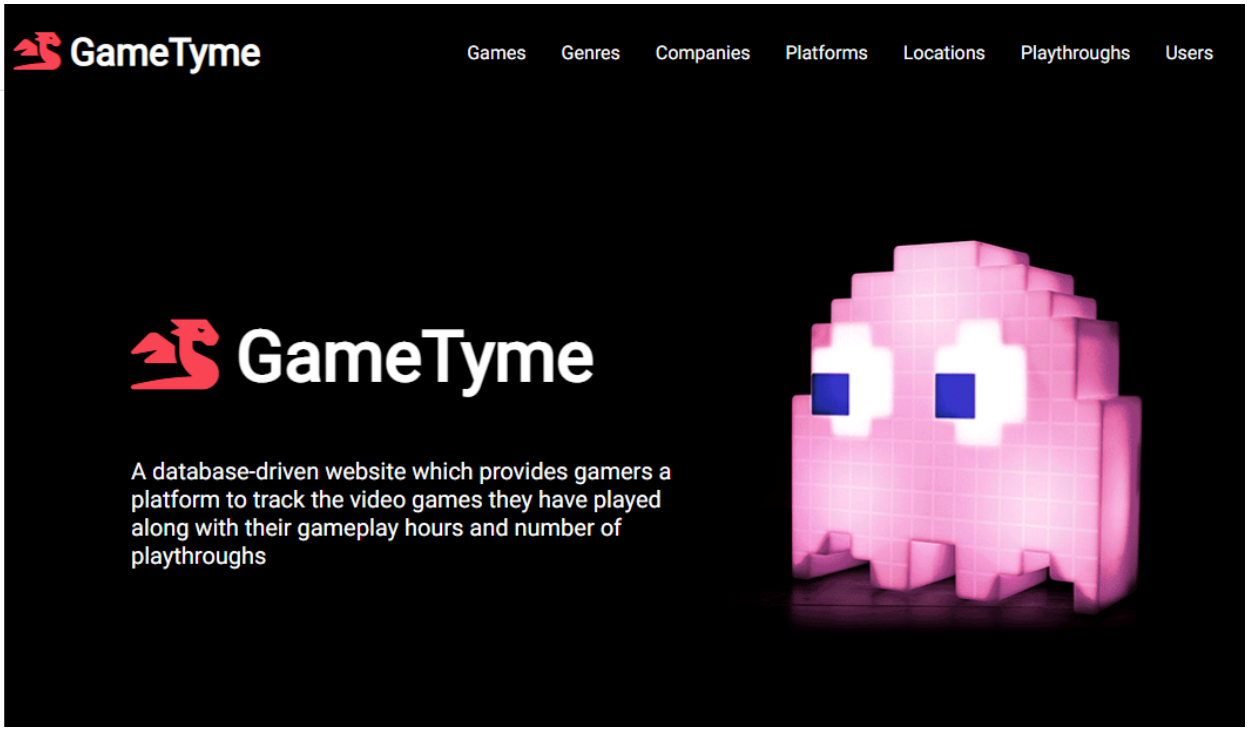
EDIT/UPDATE Users Page



DELETE Users Page



HOME


GameTyme
Games
Genres
Companies
Platforms
Locations
Playthroughs
Users
GameTyme
A database-driven website which provides gamers a platform to track the video games they have played along with their gameplay hours and number of playthroughs