

Final Project Report

Michael Wood, Deston Willis, Chris Hinckley

Project Description: Inventory Management

Our project depicts an inventory management system for a video game retail store. It consists of a web browser, a NodeJS Web server, and a MySQL database server that are all connected. The intended function is for users, who would typically be the staff of the store, to log in on the browser. Once the login information is verified, the user gains access to the inventory menu displaying all of the current games the store has in stock along with their individual information (id, name, price, etc.). From this page, they have the ability to search for a specific game, add new games to the inventory, update a game's information, or delete games from the inventory. Any and all of these changes made in the browser would update the data in the MySQL Database. Once a user is finished editing or viewing the inventory, they have the ability to logout and redirect the browser back to the login page.

ER MODEL



RM MODEL

games(game_id, gname, price, publisher, player_type)

platforms(platform_id, pname, manufacturer)

inventory(inventory_id, game_id, platform_id)

staff(staff_id, staff_name)

login(username, password)

login_staff(username, staff_id)

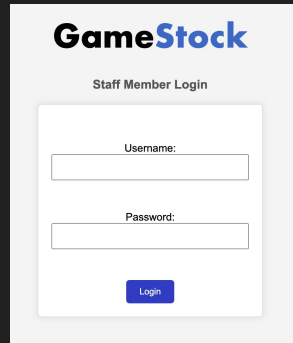
Notes/Assumptions:

- The platforms table was not used in the final version of the project.
- We assume that the store has separate entities for their current inventory and all games.
- The ER Model file is also in our FinalProject repository as DB_ER_Model.mwb
- The actual creation and insertion statements in the SQL database can be found in the repository as DB_Schema_Creation.sql and Initial_Insert_Statements.sql respectively as well.

Login Page/Functionality

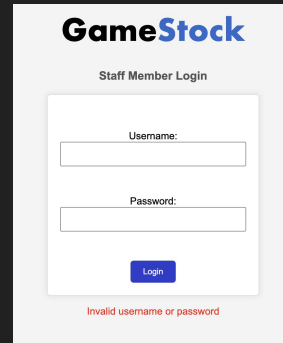
The first page that users see is Staff_Login.html. Users login by typing in their information. If the information does not match the information in the login table in the database, the login fails and an error message is displayed. If it does match, the login succeeds and the page is directed to the inventory menu. The server does this by using a SELECT SQL Operation that finds any login information that matches the information entered into the username and password boxes on the web browser.

Login box:



The screenshot shows the 'GameStock' logo at the top, followed by the text 'Staff Member Login'. Below this is a white box containing two input fields: 'Username:' and 'Password:'. A blue 'Login' button is positioned at the bottom of the white box.

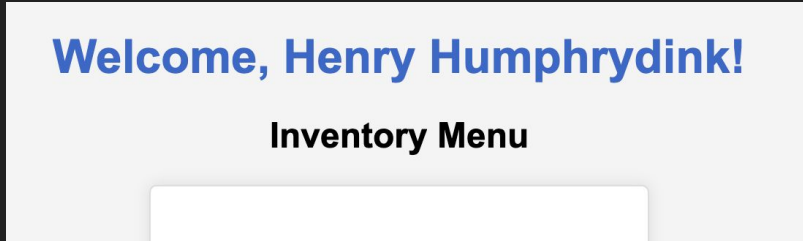
Error message:



The screenshot shows the 'GameStock' logo at the top, followed by the text 'Staff Member Login'. Below this is a white box containing two input fields: 'Username:' and 'Password:'. A blue 'Login' button is positioned at the bottom of the white box. Below the white box, the text 'Invalid username or password' is displayed in red.

Session Tracking

The server(server.js) is able to track the session when a user successfully logs in. The menu page displays the current user that has logged in:



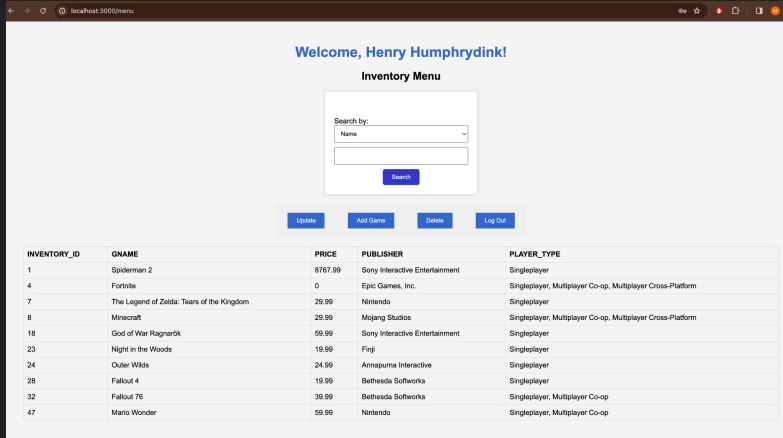
Once the user makes the needed changes to the inventory. There is a Log Out button. This button takes the user back to the login page and the menu page cannot be accessed until another user has logged in again.



Menu Page/Search Function

Once a user has successfully signed in, the inventory menu page is displayed. This page displays every inventory item the store has including that item's inventory id and the game information that comes with that item. The user can also search for specific games by Name, Price, and Publisher, which takes them to the new_search page that displays all items, if any, that match with the search criteria and description. If nothing is entered in the search box and the user presses the search button, then the new search simply displays all of the inventory items again by default.

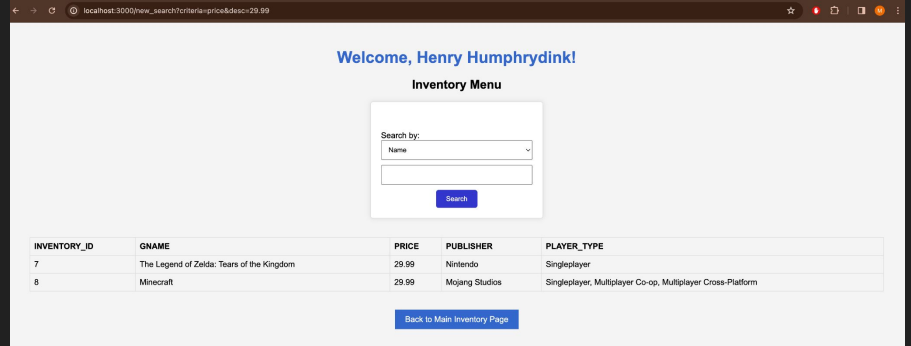
The Initial Menu Page Display



A screenshot of a web browser displaying the 'Inventory Menu' page. The page has a dark header with the text 'Welcome, Henry Humphrydink!' and 'Inventory Menu'. Below the header is a search bar with a dropdown menu set to 'Name' and a 'Search' button. Below the search bar are four buttons: 'Update', 'Add Game', 'Delete', and 'Log Out'. Below these buttons is a table with five columns: 'INVENTORY_ID', 'GNAME', 'PRICE', 'PUBLISHER', and 'PLAYER_TYPE'. The table contains 15 rows of game data.

| INVENTORY_ID | GNAME | PRICE | PUBLISHER | PLAYER_TYPE |
|--------------|---|----------|--------------------------------|---|
| 1 | Spiderman 2 | \$767.99 | Sony Interactive Entertainment | Singleplayer |
| 4 | Fortnite | 0 | Epic Games, Inc. | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |
| 7 | The Legend of Zelda: Tears of the Kingdom | 29.99 | Nintendo | Singleplayer |
| 8 | Minecraft | 29.99 | Mojang Studios | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |
| 18 | God of War Ragnarök | 59.99 | Sony Interactive Entertainment | Singleplayer |
| 23 | Night in the Woods | 19.99 | Frog | Singleplayer |
| 24 | Outer Wilds | 24.99 | Annapurna Interactive | Singleplayer |
| 28 | Fallout 4 | 19.99 | Bethesda Softworks | Singleplayer |
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |

The New Menu Display After Searching the Price 29.99



A screenshot of a web browser displaying the 'Inventory Menu' page after a search for the price 29.99. The page has a dark header with the text 'Welcome, Henry Humphrydink!' and 'Inventory Menu'. Below the header is a search bar with a dropdown menu set to 'Name' and a 'Search' button. Below the search bar is a button labeled 'Back to Main Inventory Page'. Below the button is a table with five columns: 'INVENTORY_ID', 'GNAME', 'PRICE', 'PUBLISHER', and 'PLAYER_TYPE'. The table contains 2 rows of game data.

| INVENTORY_ID | GNAME | PRICE | PUBLISHER | PLAYER_TYPE |
|--------------|---|-------|----------------|---|
| 7 | The Legend of Zelda: Tears of the Kingdom | 29.99 | Nintendo | Singleplayer |
| 8 | Minecraft | 29.99 | Mojang Studios | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |

Sorting Function

Each of the column headings in the display table can be clicked on. Clicking on the headings sort the data in either ascending or descending order according to that specific column.

| INVENTORY_ID | GNAME ↓ | PRICE | PUBLISHER | PLAYER_TYPE |
|--------------|---|---------|--------------------------------|---|
| 28 | Fallout 4 | 19.99 | Bethesda Softworks | Singleplayer |
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 4 | Fortnite | 0 | Epic Games, Inc. | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |
| 18 | God of War Ragnarök | 59.99 | Sony Interactive Entertainment | Singleplayer |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| 8 | Minecraft | 29.99 | Mojang Studios | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |
| 23 | Night in the Woods | 19.99 | Finji | Singleplayer |
| 24 | Outer Wilds | 24.99 | Annapurna Interactive | Singleplayer |
| 1 | Spiderman 2 | 8767.99 | Sony Interactive Entertainment | Singleplayer |
| 7 | The Legend of Zelda: Tears of the Kingdom | 29.99 | Nintendo | Singleplayer |

The table display
sorted by game
name in
ascending order.

| INVENTORY_ID | GNAME | PRICE ↑ | PUBLISHER ↑ | PLAYER_TYPE |
|--------------|---|---------|--------------------------------|---|
| 1 | Spiderman 2 | 8767.99 | Sony Interactive Entertainment | Singleplayer |
| 18 | God of War Ragnarök | 59.99 | Sony Interactive Entertainment | Singleplayer |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 7 | The Legend of Zelda: Tears of the Kingdom | 29.99 | Nintendo | Singleplayer |
| 8 | Minecraft | 29.99 | Mojang Studios | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |
| 24 | Outer Wilds | 24.99 | Annapurna Interactive | Singleplayer |
| 23 | Night in the Woods | 19.99 | Finji | Singleplayer |
| 28 | Fallout 4 | 19.99 | Bethesda Softworks | Singleplayer |
| 4 | Fortnite | 0 | Epic Games, Inc. | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |

The table display
sorted by
publisher name
in descending
order.

Add Function

On the menu page, there is an Add button. Clicking on this button takes the user to the addGame.html page. The user may enter the new game's information into the designated boxes. If all of the information is entered in correctly, the server uses the INSERT INTO SQL operation to insert a new inventory item and a new item in the games table if the new information is unique.

The Add Box

Add a Game

Name:

Test Game 92

Price (in the form of 00.00):

9.99

Publisher:

Pizza Studios

**Player Types (i.e.
"1 = Singleplayer,
2 = Multiplayer Co-op,
3 = Multiplayer Cross-Platform"):**

2

Add

[Back to Main Menu](#)

The New Item Added to the Data Table

| | | | | |
|----|--------------|-------|--------------------|---------------------------------|
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| 51 | Test Game 92 | 9.99 | Pizza Studios | Singleplayer, Multiplayer Co-op |
| | | | | |

Add Function (SQL Database)

The new Item is added in the inventory Table

```
1 select *
2 from inventory;
```

0% 16:2

Result Grid Filter Rows:

| inventory_id | game_id | platform_... |
|--------------|---------|--------------|
| 1 | 1 | 3 |
| 4 | 2 | 3 |
| 7 | 3 | 4 |
| 8 | 4 | 1 |
| 18 | 6 | 3 |
| 23 | 7 | 7 |
| 24 | 8 | 1 |
| 28 | 9 | 1 |
| 32 | 10 | 2 |
| 47 | 39 | NULL |
| 51 | 41 | NULL |
| NULL | NULL | NULL |

The new game is added in the games Table

```
1 select *
2 from games;
```

0% 12:2

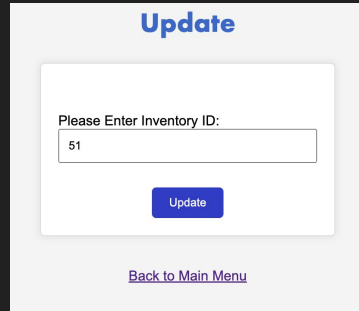
Result Grid Filter Rows: Search Edit: Export/Import:

| g... | gname | price | publisher | player_type |
|------|-----------------------|-----------|------------------|-------------|
| 25 | God of War (2018) | 39.99 | Sony | 1 |
| 26 | Test Game | 99999.... | test Pub | 2 |
| 27 | test5 | 56.97 | me | 1 |
| 28 | Super Mario World | 49.99 | Ninentdo | 1 |
| 29 | Test Game 2 | 32.99 | Test Game 2 | 3 |
| 30 | Test jndf;ajsn | 32.22 | test ublishdf 3 | 1 |
| 31 | Test game 4 | 23.30 | dsadas | 2 |
| 32 | Cult of the Lamb | 19.99 | Lamb productions | 1 |
| 33 | Test22 | 10.00 | me | 2 |
| 34 | Test Game 4 | 1.34 | Test Publi | 3 |
| 35 | 11232 | 23.00 | 123123 | 2 |
| 36 | sgsdg | 23.00 | dfdf | 2 |
| 37 | Minecraft | 29.43 | sdiobvpgiuabds | 2 |
| 38 | tes | 3.00 | as21 | 1 |
| 39 | Mario Wonder | 59.99 | Nintendo | 2 |
| 40 | Test GameIkABDVKJDSAB | 36.23 | Test | 2 |
| 41 | Test Game 92 | 9.99 | Pizza Studios | 2 |
| | NULL | NULL | NULL | NULL |

Update Function

On the menu page, there is an Update button. Clicking on this button takes the user to the updateGame.html page. The user first enters the inventory item that needs to be updated with the correct inventory id. The user then makes any changes to the item that needs to be done. If all of the information is entered in correctly, the server uses the UPDATE SQL operation to update the item's information.

Initial Update Box



The image shows a web form titled "Update" in blue text. Below the title is a white box containing the text "Please Enter Inventory ID:" followed by a text input field with the value "51". Below the input field is a blue button with the text "Update". At the bottom of the form is a blue link that says "Back to Main Menu".

Update

Please Enter Inventory ID:

51

Update

[Back to Main Menu](#)

Update Function (cont.)

Update Game Info

Game ID(Can't be changed):
41

Name:
Test Game 92

Price:
9.99

Publisher:
Pizza Studios

Player Type:
2

Update

[Back to Main Menu](#)

The game's price and player type are updated from 9.99 and 2 to 4.99 and 3 respectively. The updated information is then displayed in the menu page.

Update Game Info

Game ID(Can't be changed):
41

Name:
Test Game 92

Price:
4.99

Publisher:
Pizza Studios

Player Type:
3

Update

[Back to Main Menu](#)

Before

| | | | | |
|----|--------------|-------|--------------------|---------------------------------|
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| 51 | Test Game 92 | 9.99 | Pizza Studios | Singleplayer, Multiplayer Co-op |

After

| | | | | |
|----|--------------|-------|---------------|---|
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| 51 | Test Game 92 | 4.99 | Pizza Studios | Singleplayer, Multiplayer Co-op, Multiplayer Cross-Platform |

Update Function (SQL Database)

The item is updated in the games table with the connected game id in the inventory table.

1 • select *
2 from inventory;

0% 16:2

Result Grid Filter Rows:

| inventory_id | game_id | platform_... |
|--------------|---------|--------------|
| 1 | 1 | 3 |
| 4 | 2 | 3 |
| 7 | 3 | 4 |
| 8 | 4 | 1 |
| 18 | 6 | 3 |
| 23 | 7 | 7 |
| 24 | 8 | 1 |
| 28 | 9 | 1 |
| 32 | 10 | 2 |
| 47 | 39 | NULL |
| 51 | 41 | NULL |
| NULL | NULL | NULL |

Before

1 • select *
2 from games;

0% 12:2

Result Grid Filter Rows: Search Edit: Export/Import:

| g... gname | price | publisher | player_type |
|--------------------------|----------|------------------|-------------|
| 25 God of War (2018) | 39.99 | Sony | 1 |
| 26 Test Game | 99999... | test Pub | 2 |
| 27 test5 | 56.97 | me | 1 |
| 28 Super Mario World | 49.99 | Nintendo | 1 |
| 29 Test Game 2 | 32.99 | Test Game 2 | 1 |
| 30 Test jndf,ajsf | 32.22 | test ublishdf 3 | 1 |
| 31 Test game 4 | 23.30 | dsadas | 2 |
| 32 Cult of the Lamb | 19.99 | Lamb productions | 1 |
| 33 Test22 | 10.00 | me | 2 |
| 34 Test Game 4 | 1.34 | Test Publi | 3 |
| 35 11232 | 23.00 | 123123 | 2 |
| 36 sgsgd | 23.00 | dfdf | 2 |
| 37 Minecraft | 29.43 | sdiobvpgiuabds | 2 |
| 38 tes | 3.00 | as21 | 1 |
| 39 Mario Wonder | 59.99 | Nintendo | 2 |
| 40 Test GameikABDVKJDSAB | 36.23 | Test | 2 |
| 41 Test Game 92 | 9.99 | Pizza Studios | 2 |
| NULL NULL | NULL | NULL | NULL |

After

1 • select *
2 from games;

10% 12:2

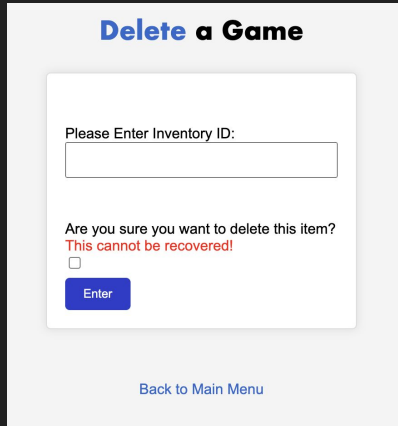
Result Grid Filter Rows: Search Edit: Export/Import:

| g... gname | price | publisher | player_type |
|--------------------------|----------|------------------|-------------|
| 25 God of War (2018) | 39.99 | Sony | 1 |
| 26 Test Game | 99999... | test Pub | 2 |
| 27 test5 | 56.97 | me | 1 |
| 28 Super Mario World | 49.99 | Nintendo | 1 |
| 29 Test Game 2 | 32.99 | Test Game 2 | 3 |
| 30 Test jndf,ajsf | 32.22 | test ublishdf 3 | 1 |
| 31 Test game 4 | 23.30 | dsadas | 2 |
| 32 Cult of the Lamb | 19.99 | Lamb productions | 1 |
| 33 Test22 | 10.00 | me | 2 |
| 34 Test Game 4 | 1.34 | Test Publi | 3 |
| 35 11232 | 23.00 | 123123 | 2 |
| 36 sgsgd | 23.00 | dfdf | 2 |
| 37 Minecraft | 29.43 | sdiobvpgiuabds | 2 |
| 38 tes | 3.00 | as21 | 1 |
| 39 Mario Wonder | 59.99 | Nintendo | 2 |
| 40 Test GameikABDVKJDSAB | 36.23 | Test | 2 |
| 41 Test Game 92 | 4.99 | Pizza Studios | 3 |
| NULL NULL | NULL | NULL | NULL |

Delete Function

On the menu page, there is an Delete button. Clicking on this button takes the user to the deleteGame.html page. The user may enter the inventory item that needs to be deleted with the correct inventory id. There is a check box that must be clicked to ensure that the user wants to delete the item from the inventory. Once that is done and the inventory id is correctly entered, the game is deleted from the data using the DELETE SQL operation in the server.

The Delete Box



Delete a Game

Please Enter Inventory ID:

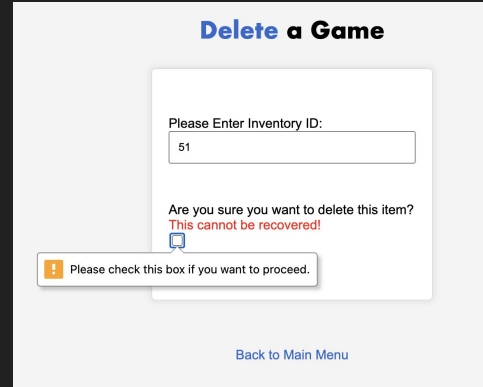
Are you sure you want to delete this item?
This cannot be recovered!

☐

[Enter](#)

[Back to Main Menu](#)

Check Box Must be Checked



Delete a Game

Please Enter Inventory ID:

Are you sure you want to delete this item?
This cannot be recovered!

☒

[Please check this box if you want to proceed.](#)

[Back to Main Menu](#)

Delete Function (cont.)

Before the Item is Deleted

| | | | | |
|----|--------------|-------|--------------------|---------------------------------|
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| 51 | Test Game 92 | 9.99 | Pizza Studios | Singleplayer, Multiplayer Co-op |
| | | | | |

After the Item is Deleted

| | | | | |
|----|--------------|-------|--------------------|---------------------------------|
| 32 | Fallout 76 | 39.99 | Bethesda Softworks | Singleplayer, Multiplayer Co-op |
| 47 | Mario Wonder | 59.99 | Nintendo | Singleplayer, Multiplayer Co-op |
| | | | | |

Delete Function (SQL Database)

The item is deleted from the inventory table.

Before

```
1 • select *
2   from inventory;
```

0% 16:2

Result Grid Filter Rows:

| inventory_id | game_id | platform_... |
|--------------|---------|--------------|
| 1 | 1 | 3 |
| 4 | 2 | 3 |
| 7 | 3 | 4 |
| 8 | 4 | 1 |
| 18 | 6 | 3 |
| 23 | 7 | 7 |
| 24 | 8 | 1 |
| 28 | 9 | 1 |
| 32 | 10 | 2 |
| 47 | 39 | NULL |
| 51 | 41 | NULL |
| NULL | NULL | NULL |

After

```
1 • select *
2   from inventory;
```

00% 16:2

Result Grid Filter Rows:

| inventory_id | game_id | platform_... |
|--------------|---------|--------------|
| 1 | 1 | 3 |
| 4 | 2 | 3 |
| 7 | 3 | 4 |
| 8 | 4 | 1 |
| 18 | 6 | 3 |
| 23 | 7 | 7 |
| 24 | 8 | 1 |
| 28 | 9 | 1 |
| 32 | 10 | 2 |
| 47 | 39 | NULL |
| NULL | NULL | NULL |


Error Checking

Both the client and server side perform error checking to ensure that correct information is entered into the boxes. Here are some examples from the addGame page:

Empty Field

Add a Game

Name:

 Please fill out this field.

Price (in the form of 00.00):

Publisher:

Player Types (i.e.
*1 = Singleplayer,
2 = Multiplayer Co-op,
3 = Multiplayer Cross-Platform*):

[Add](#)


[Back to Main Menu](#)

Nonsensical Data

Add a Game

Name:

Price (in the form of 00.00):

 Please match the requested format.
Please enter a valid decimal format

Publisher:

Player Types (i.e.
*1 = Singleplayer,
2 = Multiplayer Co-op,
3 = Multiplayer Cross-Platform*):

[Add](#)

[Back to Main Menu](#)


Extra Spaces

Add a Game

Name:

Price (in the form of 00.00):

Publisher:

 Please match the requested format.
Please enter a value without leading or trailing spaces

*1 = Singleplayer,
2 = Multiplayer Co-op,
3 = Multiplayer Cross-Platform*):

[Add](#)

[Back to Main Menu](#)

Different Value Type


Add a Game

Name:

Price (in the form of 00.00):

Publisher:

Player Types (i.e.
*1 = Singleplayer,
2 = Multiplayer Co-op,
3 = Multiplayer Cross-Platform*):

 Please match the requested format.
Please enter a numeric value without leading or trailing spaces

[Add](#)

[Back to Main Menu](#)

Error Checking (Server Side)

If there is an error in the server side, then an error message is displayed. This could occur if data is not entered correctly and the client side error checking cannot catch it for whatever reason. Here is an example (if the price is not entered correctly for updating the game's information) :



Statement of Ranking

The ranking statement and individual tasks by each team member can be found in our FinalProject github repository in the ranking.pdf file and on canvas.

Github Repo Link

<https://github.com/Mic-73/FinalProject>