

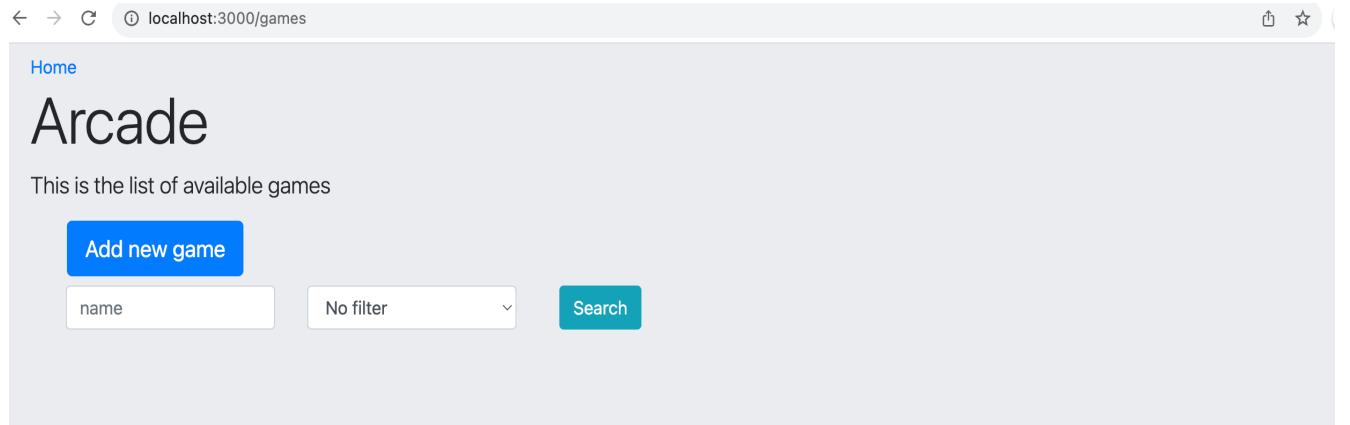
CS 6314.002 - Assignment 3

Due date: November 12, 2023 11:59PM.

Create an application by using Node.js and Express.js.

1. Your application should include all CRUD operations (Create, Read, Update, Delete) for manipulating *games* collection in Arcade Database within MongoDB. You can choose to use Single Page Application approach (consume the API using AJAX calls) or utilize ejs files. For either approach, it's important to implement all the below functions:

- localhost:3000/ should redirect to localhost:3000/games automatically (If you choose to use a different port, mention it or capture it in the screenshot)
- List all games with thumbnail images (game details in ArcadeGames.json and images are uploaded in public/images with the zip file along with the description).
 - Get the image URLs from MongoDB after uploading the json file.
 - Read the image from the fetched path and display in a grid along with game title as a hyper link to open the game details page.



localhost:3000/games

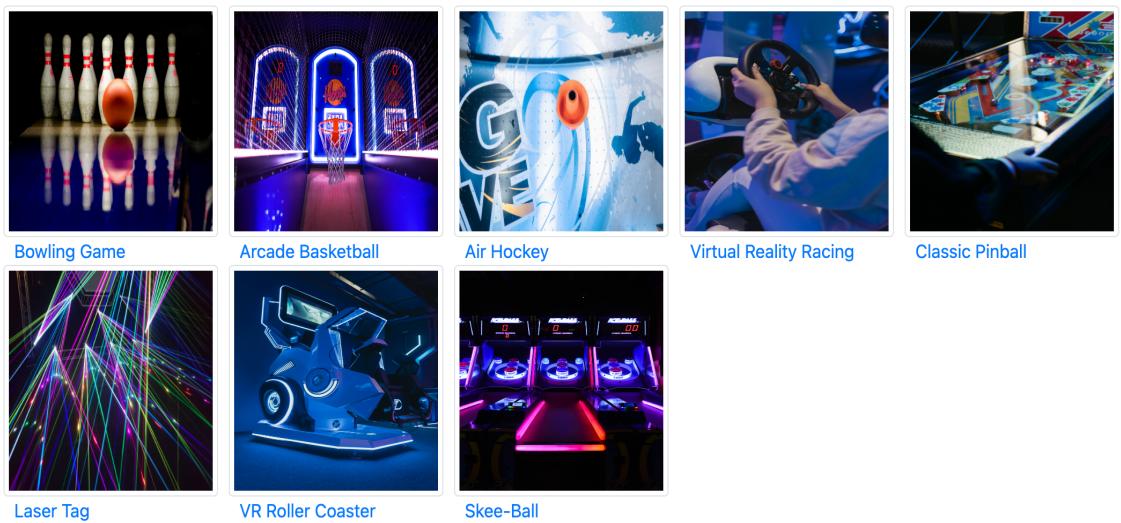
Home

Arcade

This is the list of available games

Add new game

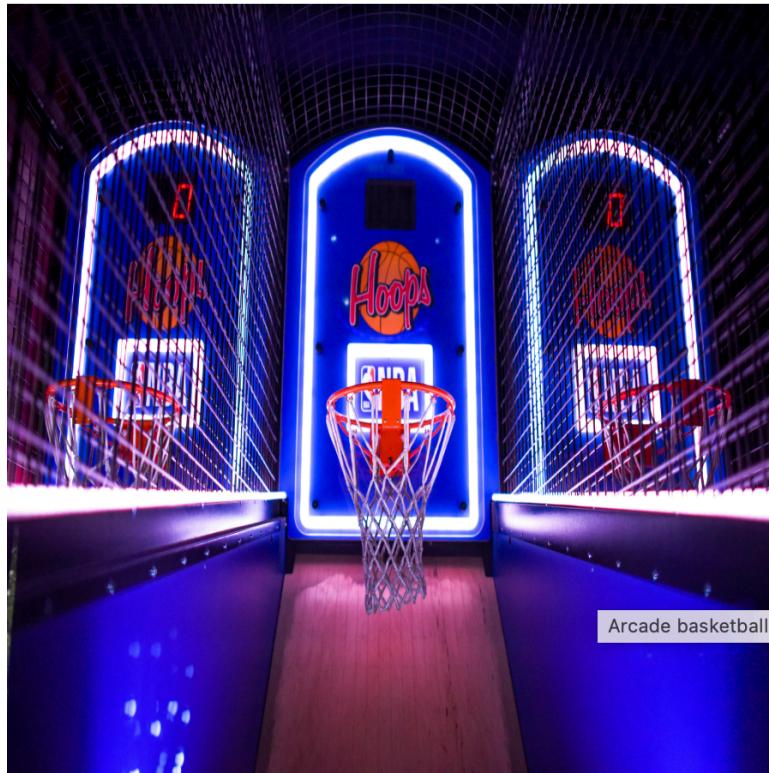
name No filter Search



- show an individual game details page and display the image description (you read from Mongodb) as a tooltip when you hover over the image.

← → ⌂ ⓘ localhost:3000/games/6542d33eac9fe7515c1abc59

Game Details



Name: Arcade Basketball

Description: Test your basketball skills and aim for the high score!

Type: Physical

Minimum Age: 10

Pricing(Hourly): \$25.00

Pricing(Per Game): \$5.00

Edit
game

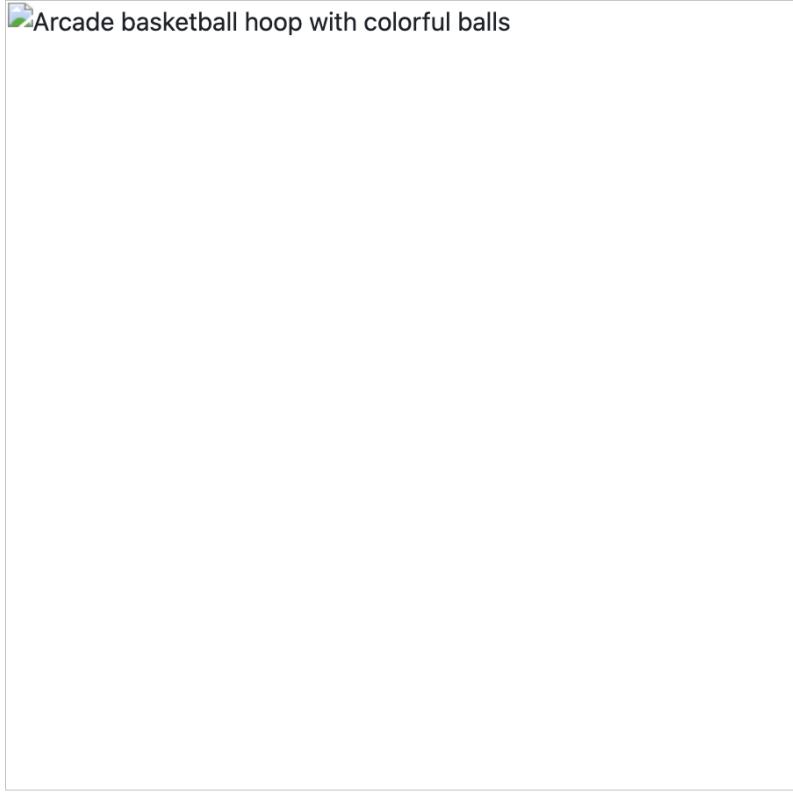
Delete game

[Back to List](#)

- In case the application is not able to fetch the image in the previous screen, alt text should be loaded with the image description as shown below.

← → ⌂ ⓘ localhost:3000/games/6542d33eac9fe7515c1abc59

Game Details



Name: Arcade Basketball

Description: Test your basketball skills and aim for the high score!

Type: Physical

Minimum Age: 10

Pricing(Hourly): \$25.00

Pricing(Per Game): \$5.00

[Edit game](#) [Delete game](#)

[Back to List](#)

- Add a new game (For the purpose of this assignment, you are not expected to upload an image from file explorer – image path and alt text are input text fields).

← → ⌂ ⓘ localhost:3000/games/new

New Game

title

description

type

minimum age

Pricing(Hourly)

Pricing(Per Game)

image path

image alt text

Save

[Cancel](#)

- Edit and update an existing game.

← → ⌂ ⓘ localhost:3000/games/6542d33eac9fe7515c1abc59/edit

Edit Game

Title

Description

Type

Minimum Age

Pricing(Hourly)

Pricing(Per Game)

Image Path

Image Alt text

Save

[Cancel](#)

- delete a game – Delete the game from mongodb and display the latest list of games on home screen.

2. Include search (partial search on “title”) and filtering (filter by “type”) functions.

Integrate search and filter together (if a type is selected, search for games under selected type). Search and filtering should be available on home page. You can hardcode the list of types for drop down list that match entries in your mongodb.

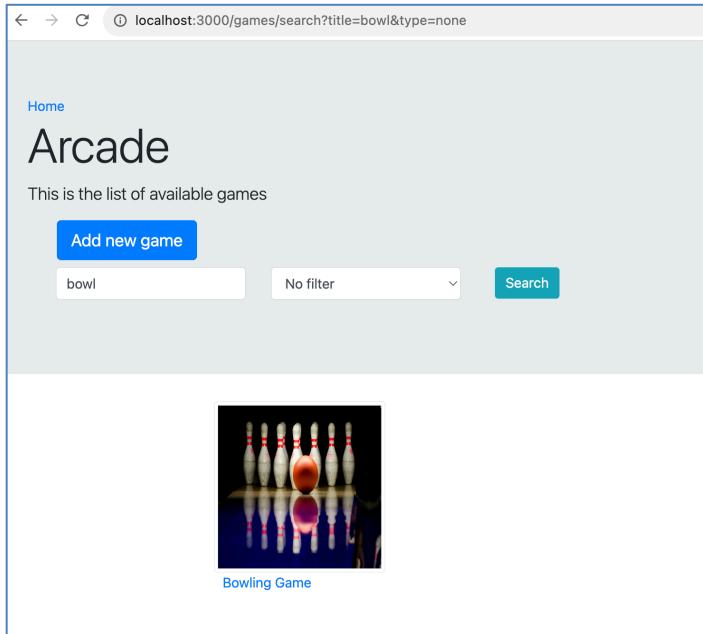


Fig: Search with Title only

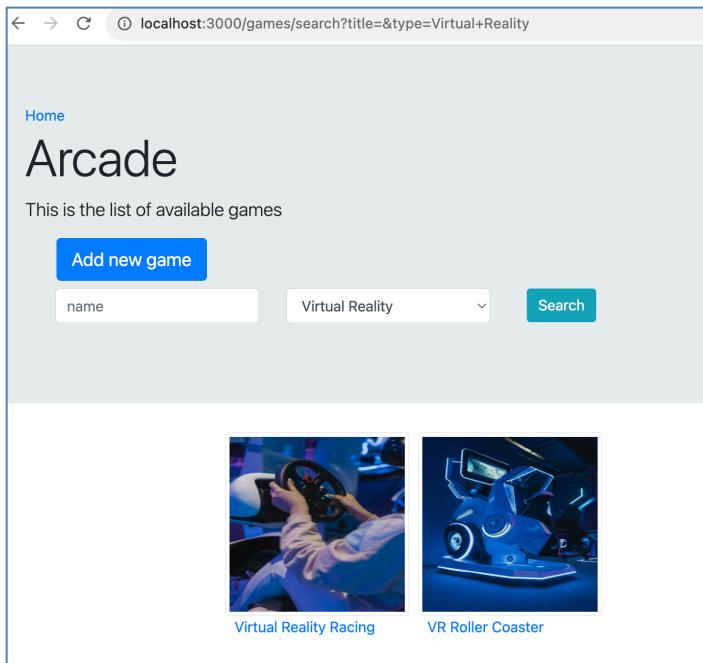


Fig: Filter by Type only

← → ⌂ ⓘ localhost:3000/games/search?title=racing&type=Virtual+Reality

Home

Arcade

This is the list of available games

Add new game



[Virtual Reality Racing](#)

Fig: Search by title and filter by Type

RESTful Routes

Name	Path	HTTP Verb	Purpose
Index	/games	GET	List all games
New	/games/new	GET	Show new game form
Create	/games	POST	Create new game
Show	/games/:id	GET	Show information about one specific game
Edit	/games/:id/edit	GET	Show edit form for one game
Update	/games/:id	PUT	Update a particular game
Destroy	/games/:id	DELETE	Delete a particular game

Resources:

<https://www.geeksforgeeks.org/express-js-res-redirect-function/>

<https://www.mongodb.com/docs/manual/reference/method/db.collection.find/>

<https://www.mongodb.com/docs/manual/reference/method/db.collection.findOne/>

<https://www.mongodb.com/docs/manual/reference/operator/query/regex/>

<https://www.geeksforgeeks.org/express-js-req-query-property/#>

Hands-on examples done in class.

Deliverables:

1. Zipped application folder (do not include *node_modules* folder)
2. Screenshots for game listing page
 - Show all games.
 - Show results of search and filtering (individually and combined)
3. Screenshot for adding a new game
4. Screenshot for game edit page
5. Screenshot for the result of game update