Table of Contents

[User stories for project 3 – Memorize Game 2](#_Toc2591267)

[Epic: 2](#_Toc2591268)

[End User Stories: 2](#_Toc2591269)

[Development Stories: 2](#_Toc2591270)

[Features left to implement: 3](#_Toc2591271)

[Mockups: 4](#_Toc2591272)

[Bugs encountered throughout 19](#_Toc2591273)

[Testing 20](#_Toc2591274)

[Unit tests 20](#_Toc2591275)

[Screenshot: 20](#_Toc2591276)

[Database information 22](#_Toc2591277)

[Collection ‘tblUsers’: 22](#_Toc2591278)

[Collection ‘tblBeginner’: 22](#_Toc2591279)

[Collection ‘tblExpert’: 23](#_Toc2591280)

[Collection ‘tblGamesPlayed’: 23](#_Toc2591281)

[Collection ‘tblAvatar’: 23](#_Toc2591282)

# User stories for project 3 – Memorize Game

#### User stories, development stories based on them, and a summary of features left to implement to enhance the experience overall. Minimum Viable Product has been reached (all development stories are checked off).

## Epic:

As an end user, I want to be able to play a memory game

### End User Stories:

1. As an end user, I want to have a unique username which I can only use (own login)
   1. I want to be able to log into my account
   2. I want to be able to log out of my account
   3. I want that my password is kept safe and not stored In human readable form
2. As an end user, I want to be able to play the game on a mobile device
3. As an end user, I don’t want to have to input text as much as possible
4. As an end user, I want to personalize my account with a picture in order to be recognizable
5. As and end user, I want that scores can be submitted in order to see how I perform versus other players.

## Development Stories:

1. User Story 1: **√**
   1. Create a welcome page, where users can login, or sign up
   2. Provide some basic explanation and description
   3. Create user registration page
      1. Allow user to go back to main page or sign in page
      2. Create logic to ensure usernames are unique
      3. Create logic for passwords
   4. Create user login page
      1. Allow user to go back to main page or registration page
      2. Ensure messages show mistakes made by the user
   5. Create user homepage/ Game Page
      1. Require logged in user for the page
         1. Return 401 page on tries to open page without login
2. User Story 2: **√**
   1. Personalise the navbar for the user **√**
      1. Show username in the navbar
   2. Create a logout button for the user **√**
   3. Create Leaderboard page **√**
3. User story 3: **√**
   1. Create a dropdown list to chose the type of game **√**
      1. 4 by 4 creates a 4 by 4 board
      2. 6 by 6 creates a 6 by 6 board
   2. Show the amount of moves needed **√**
   3. Show Time after finishing the game **√**
   4. Show Submit button for score to the leaderboards **√**

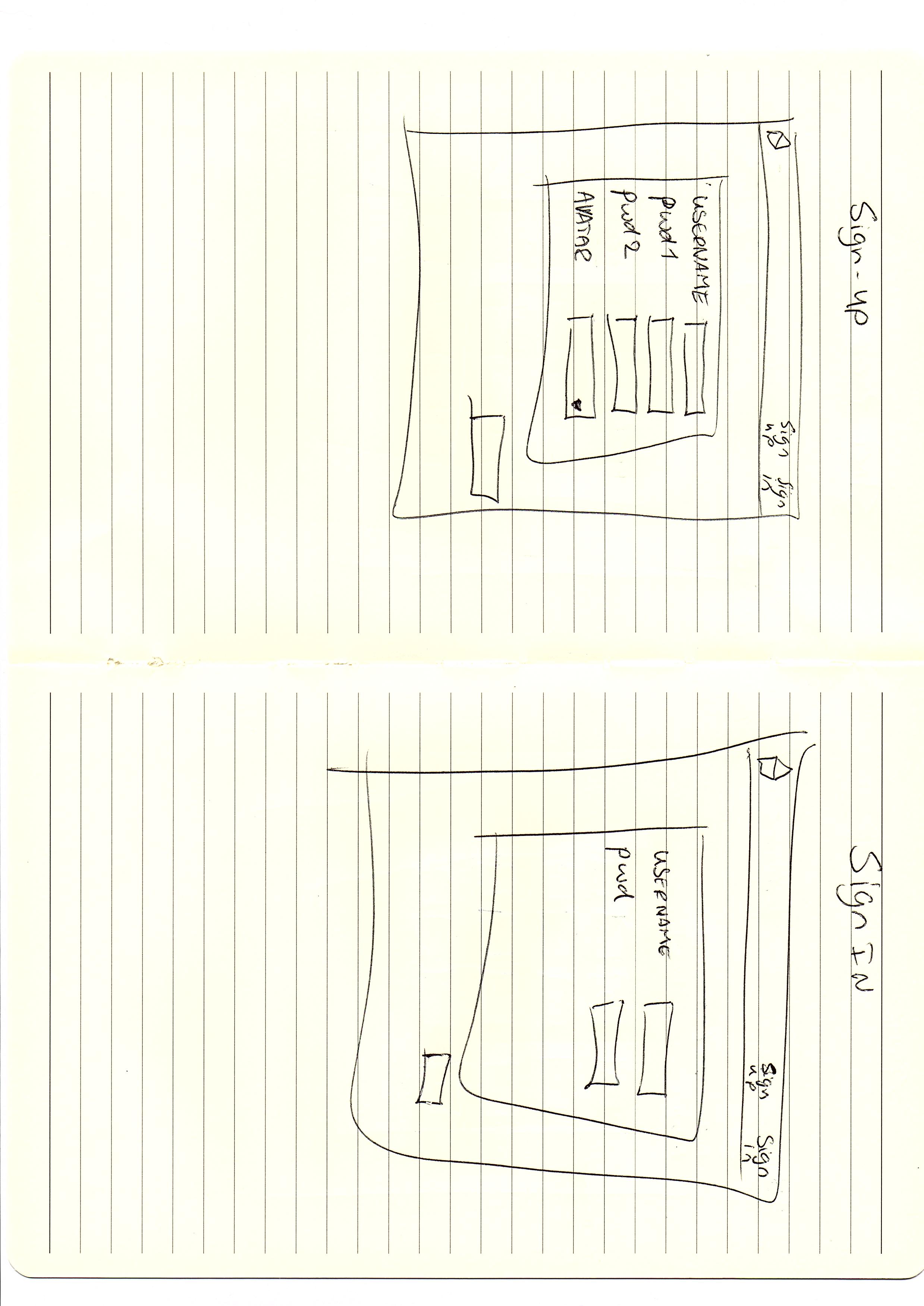
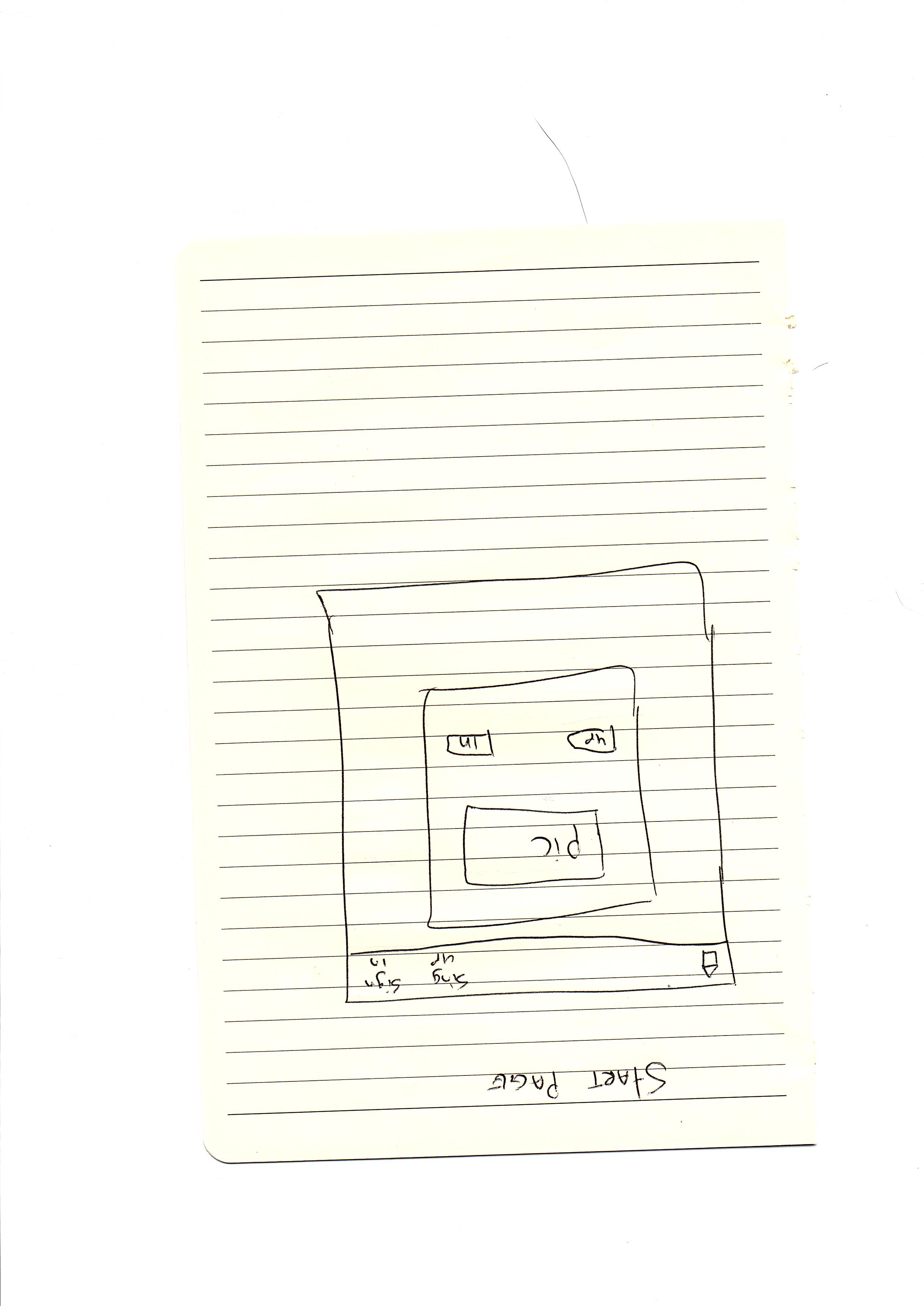
## Features left to implement:

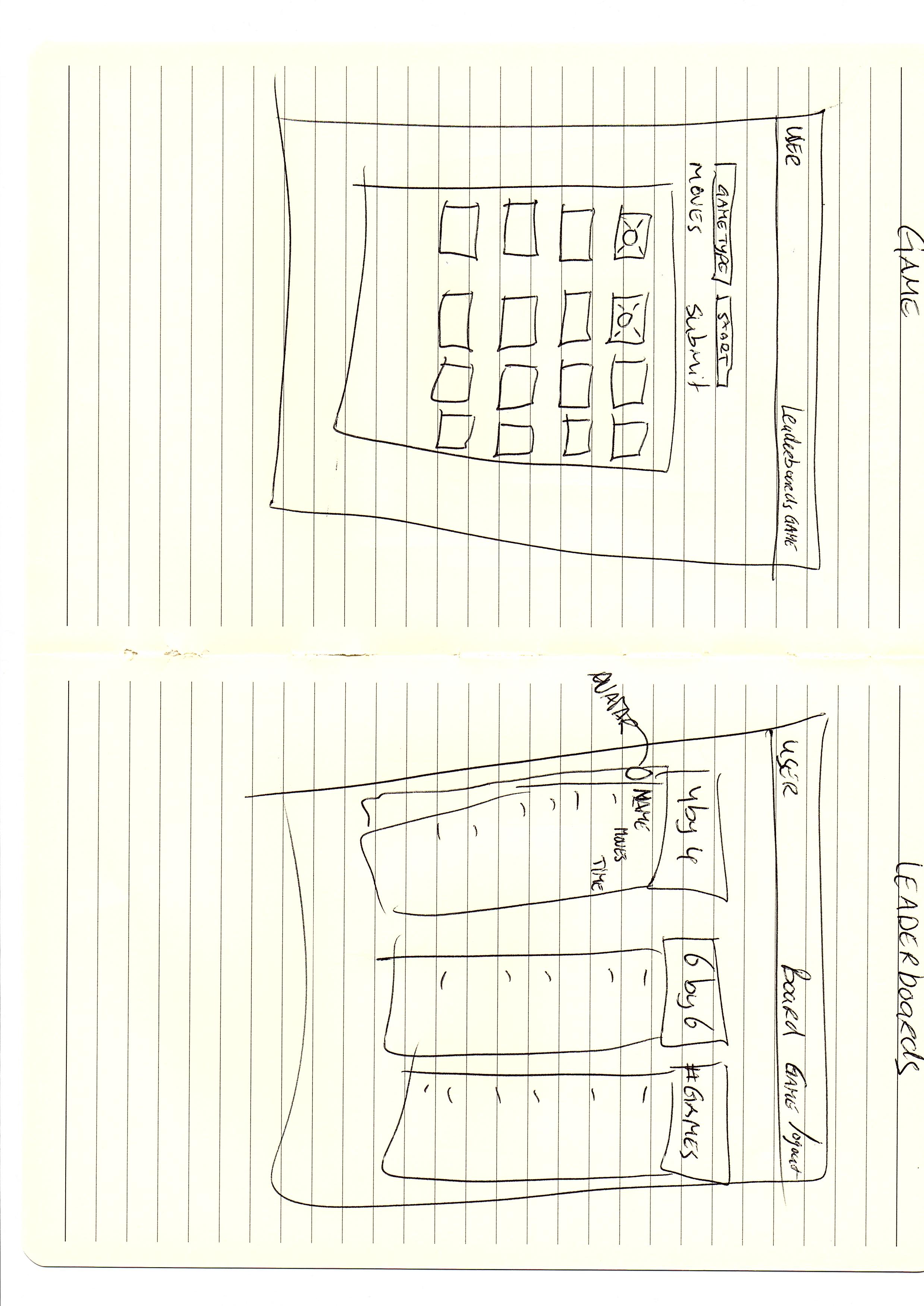
1. Create multiple icon sets so people can choose with what icons they want to play the game
2. Possibility to change the chosen avatar
3. create more game features like:
   1. match 4 instead of 2
   2. match 3 instead of 2
   3. submit score only when it's a new high score

## Mockups:

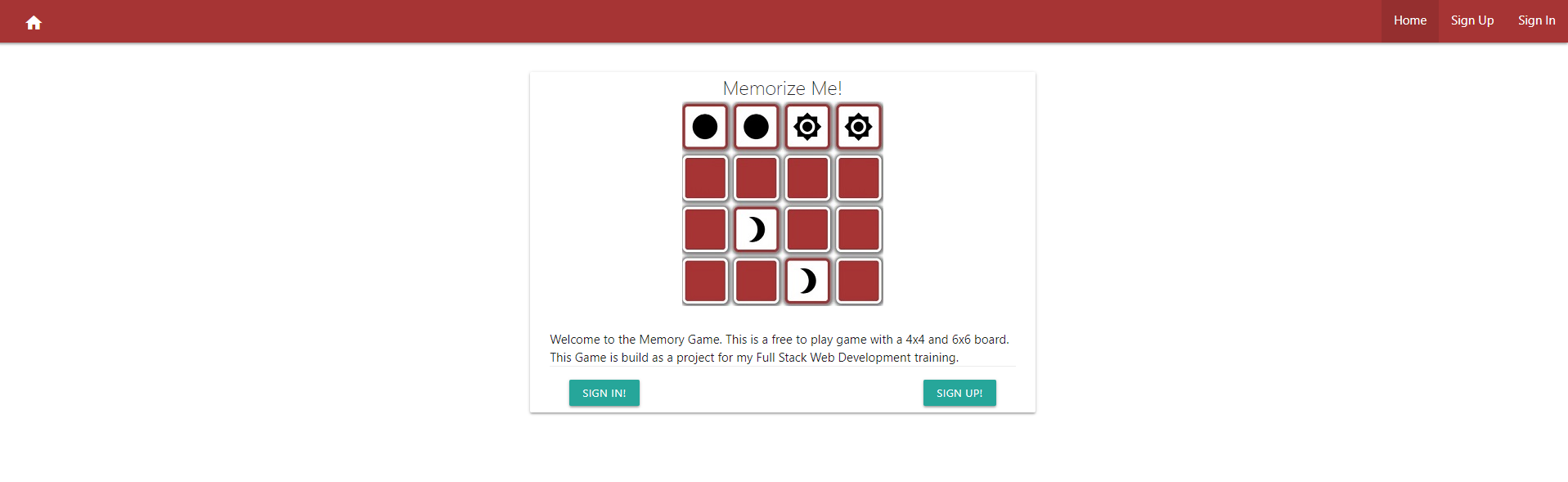
Below images were global thoughts on how it should look.

Underneath are screenshots of the end result.

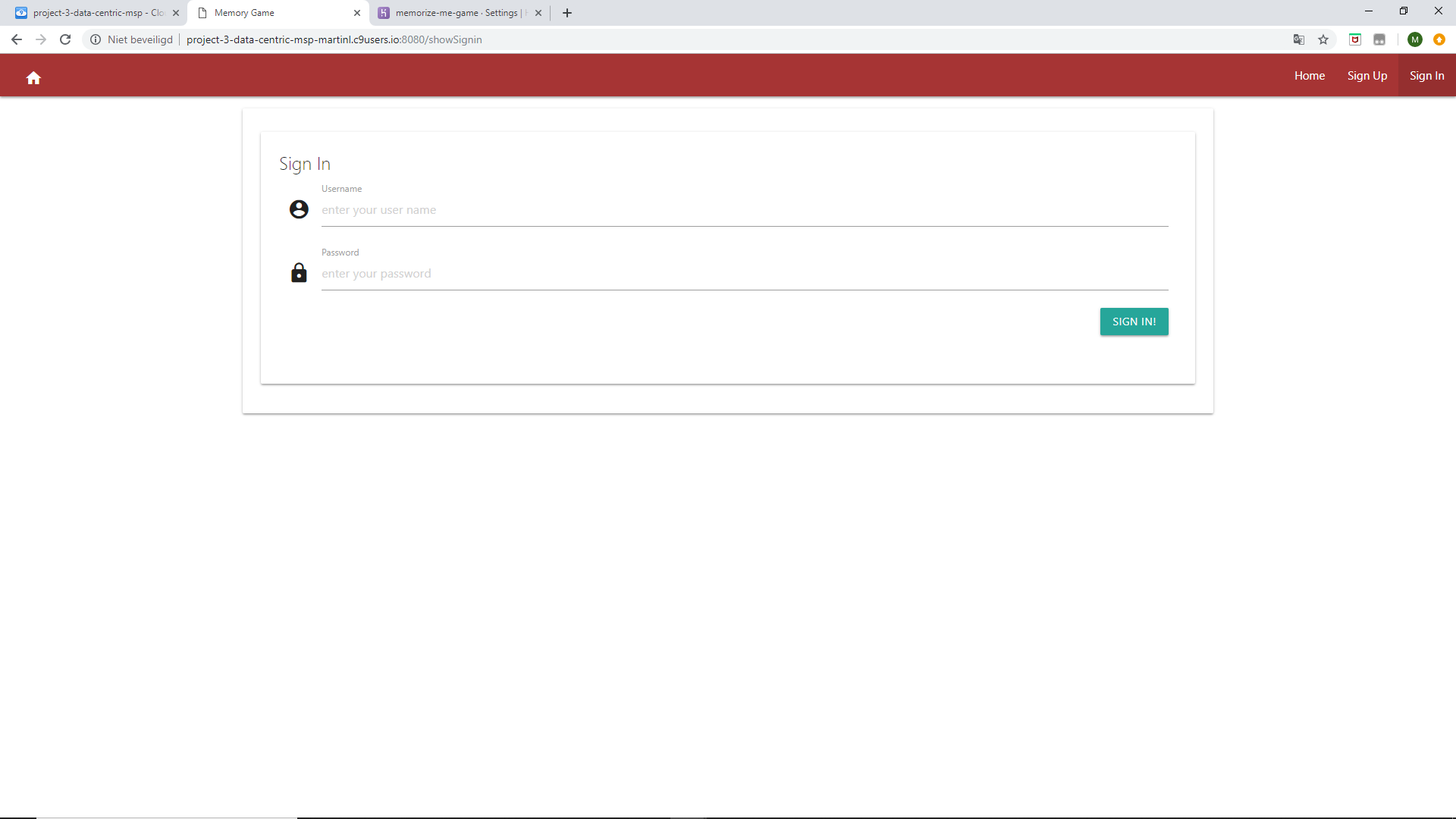




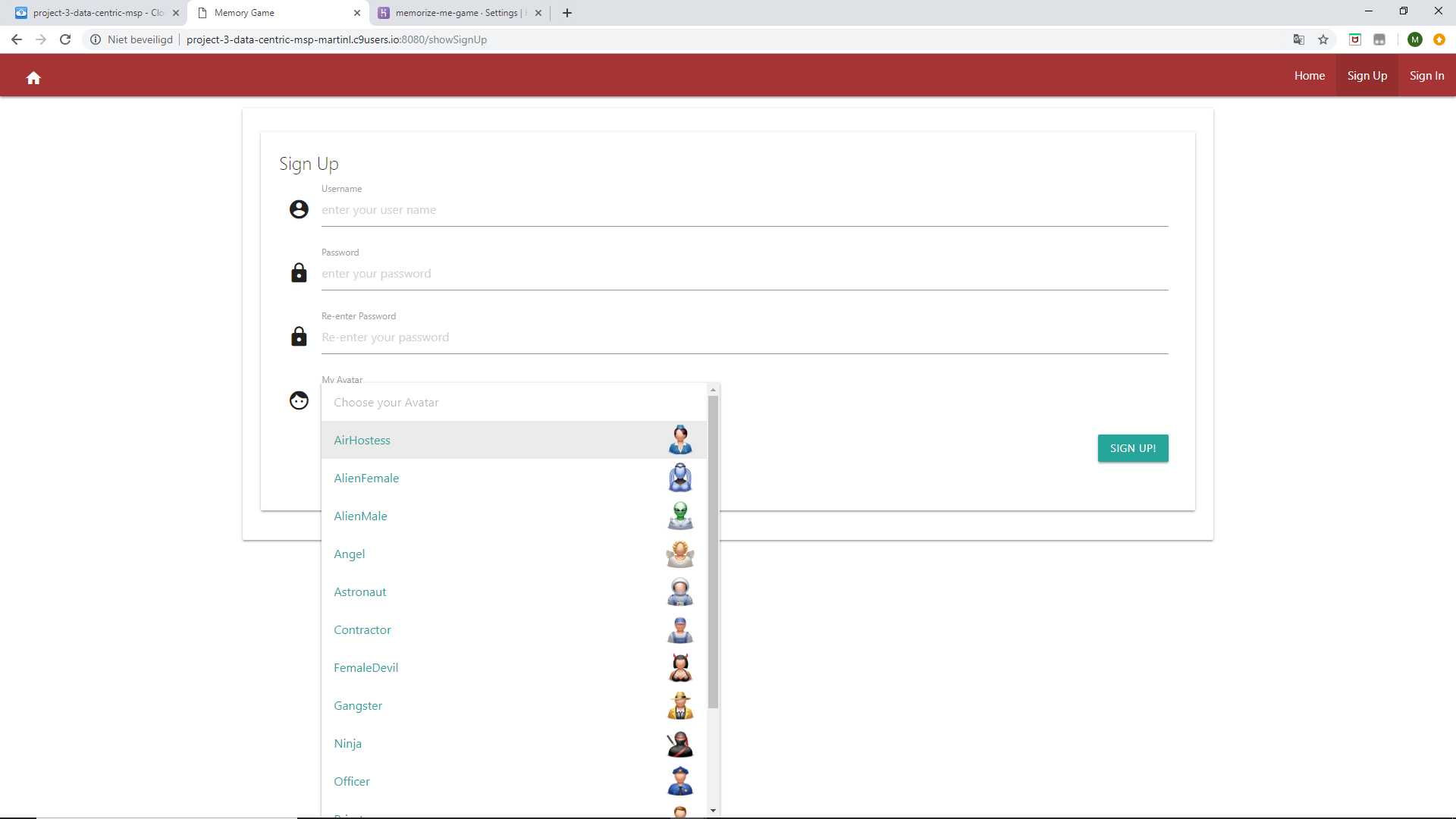
Index Page



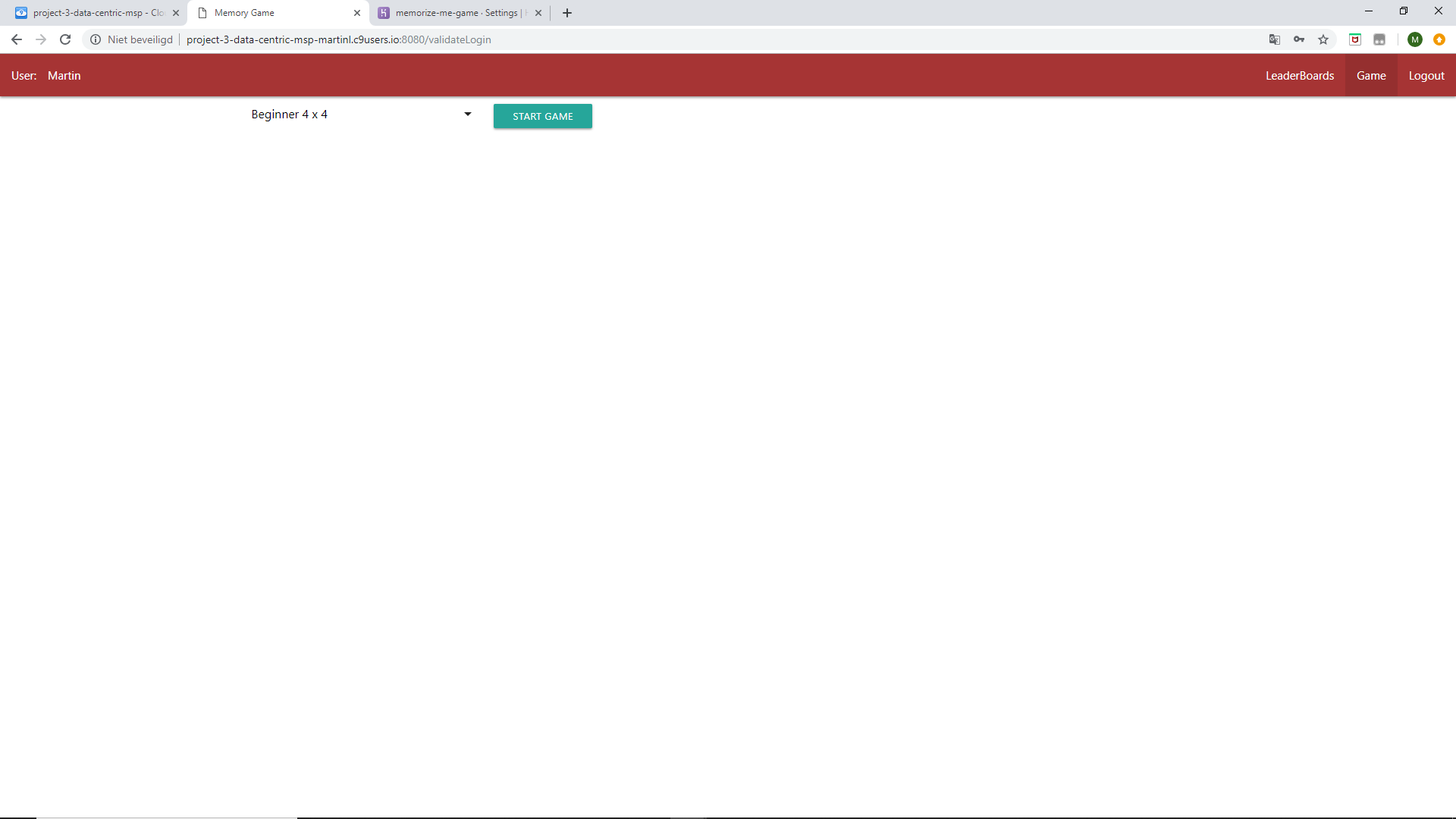
Sign In Page



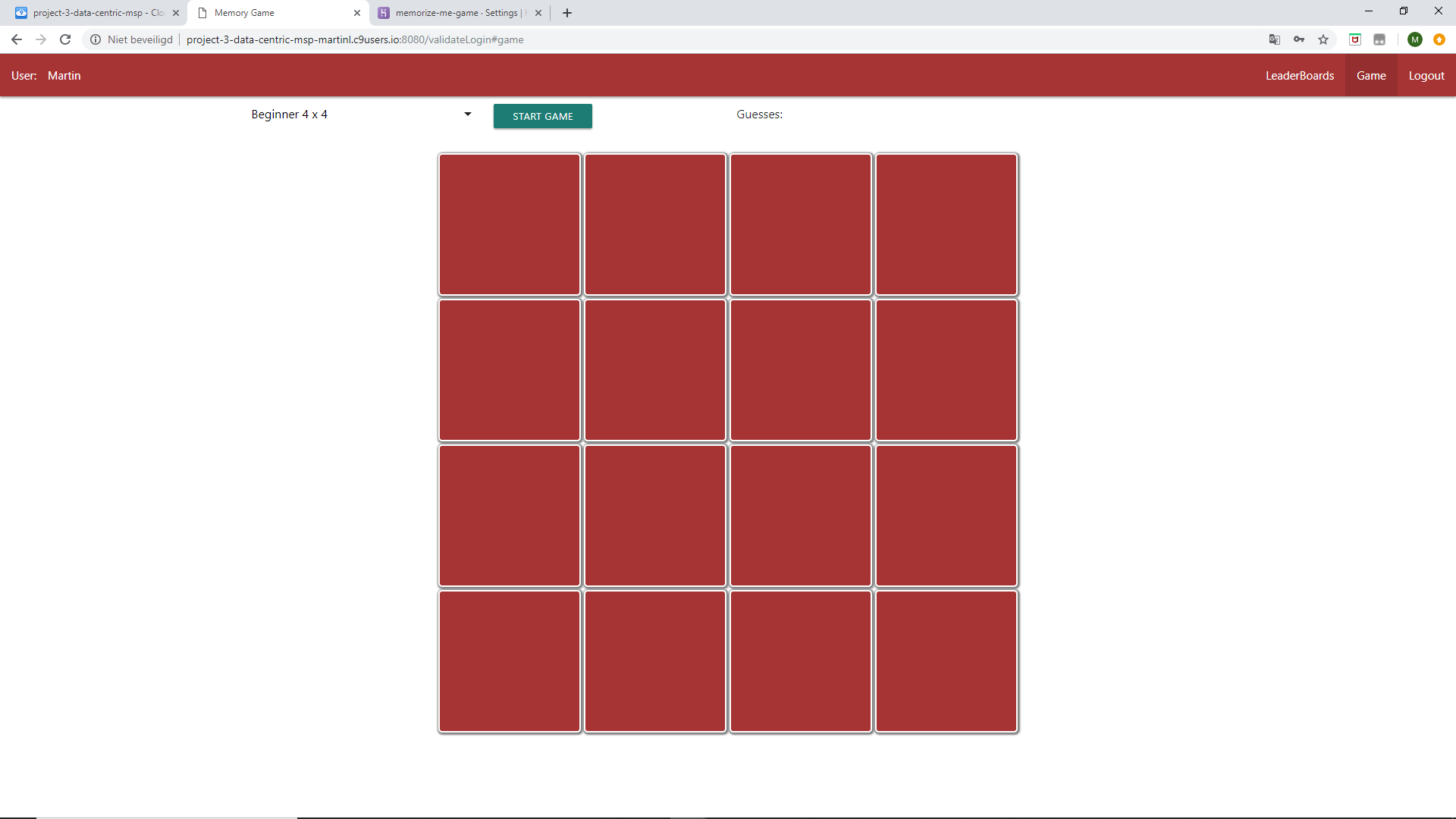
Sign Up Page



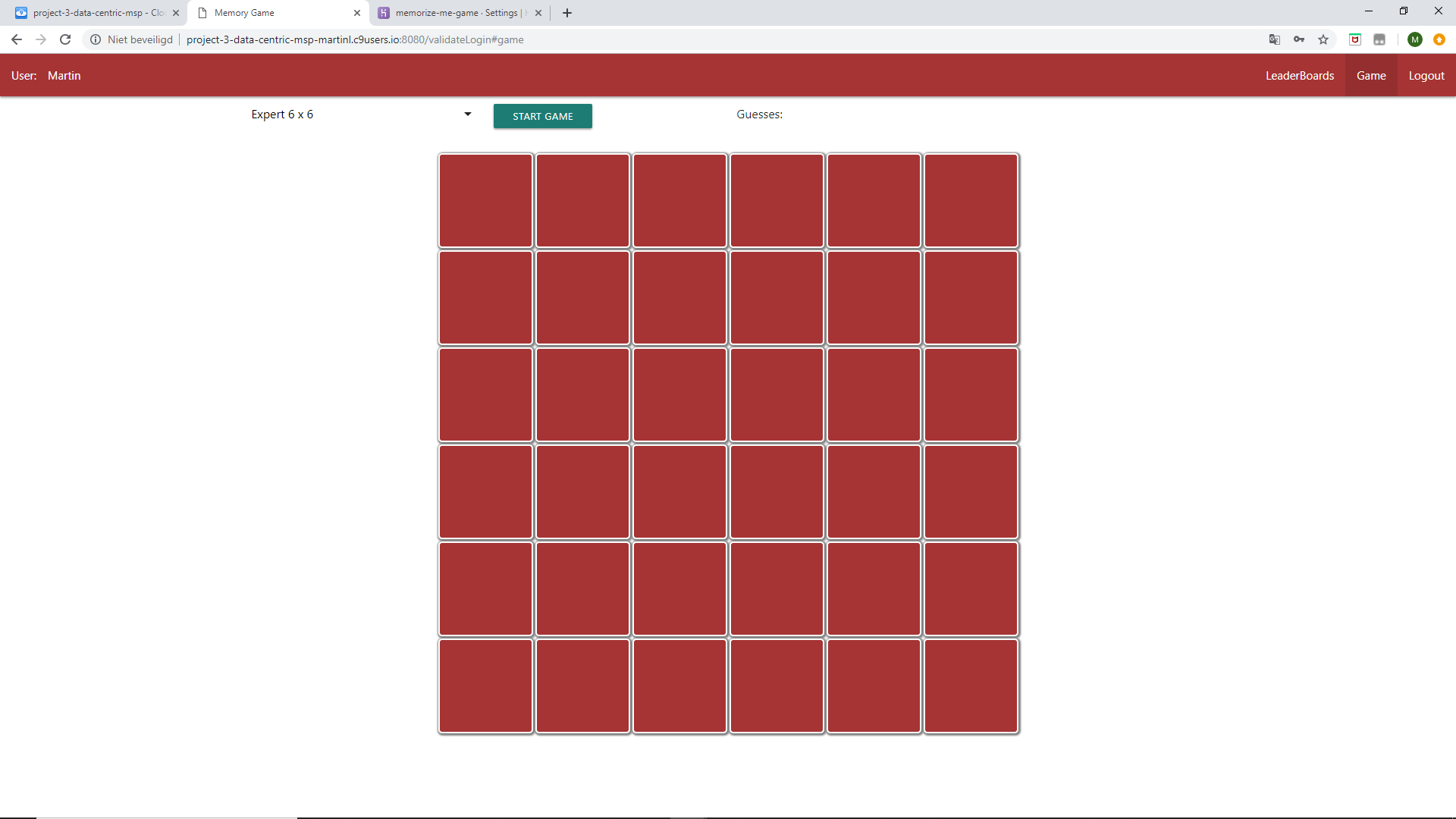
User Home Page (defaults to Game)



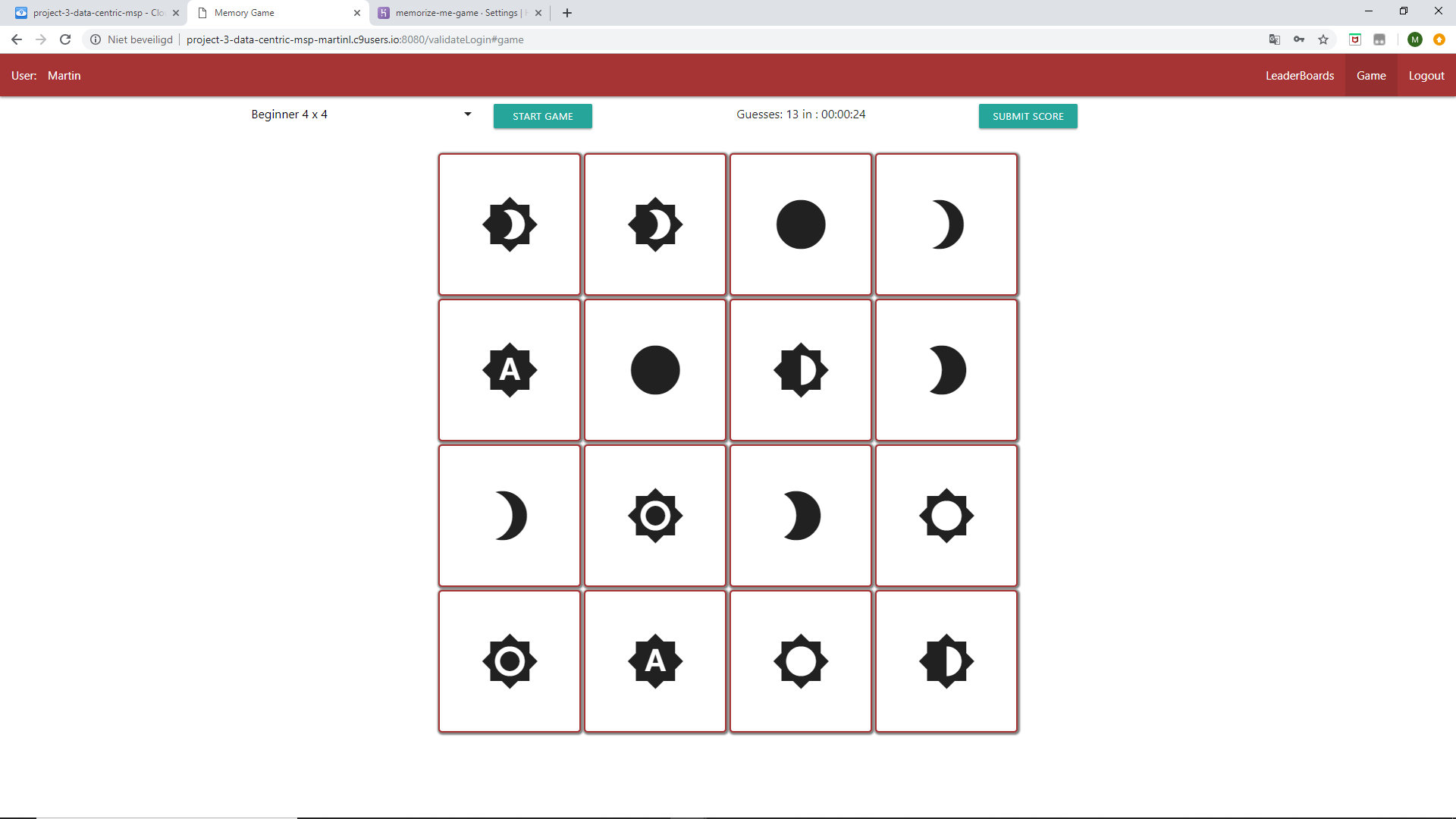
4 by 4 game



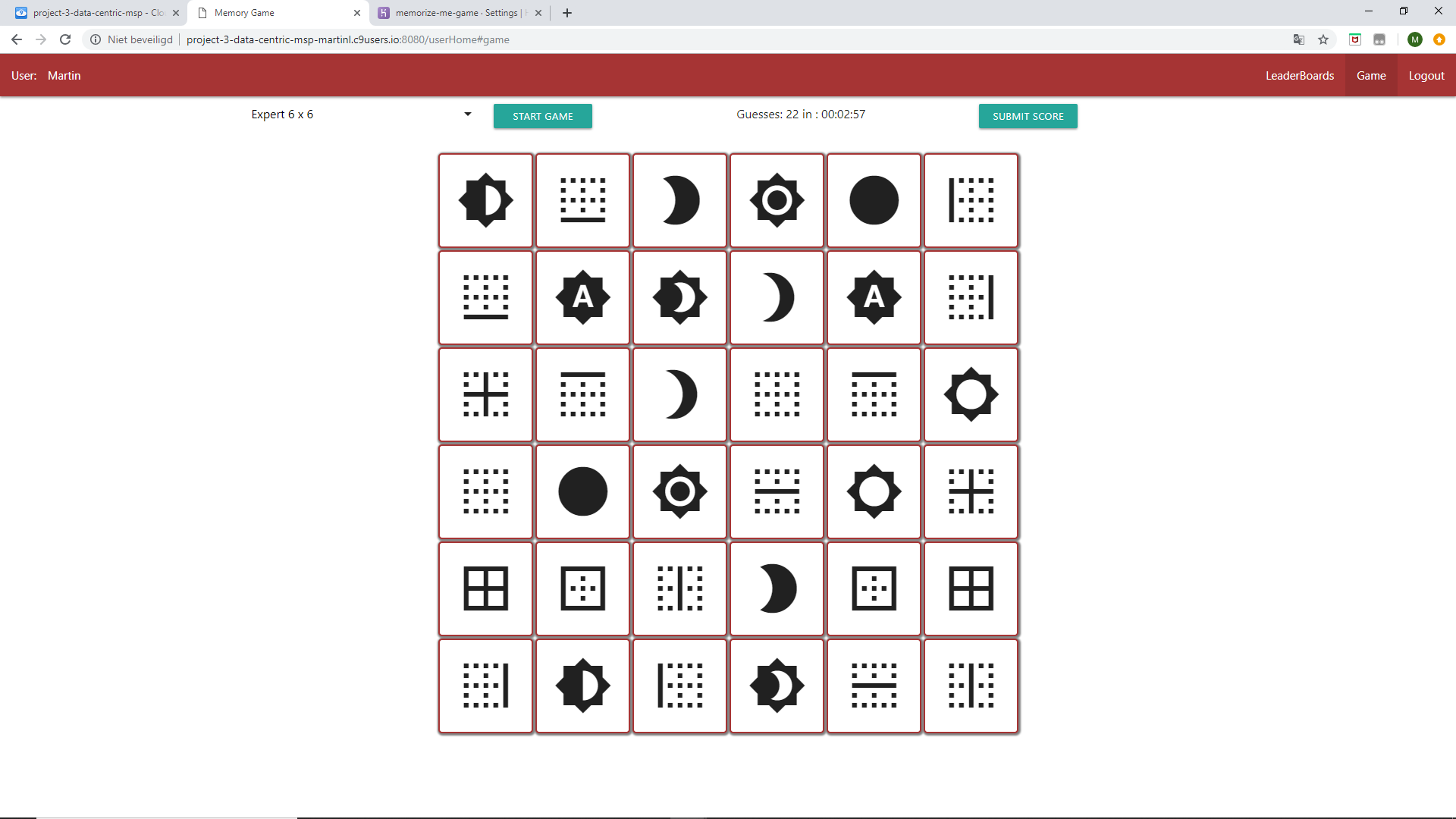
6 by 6 game



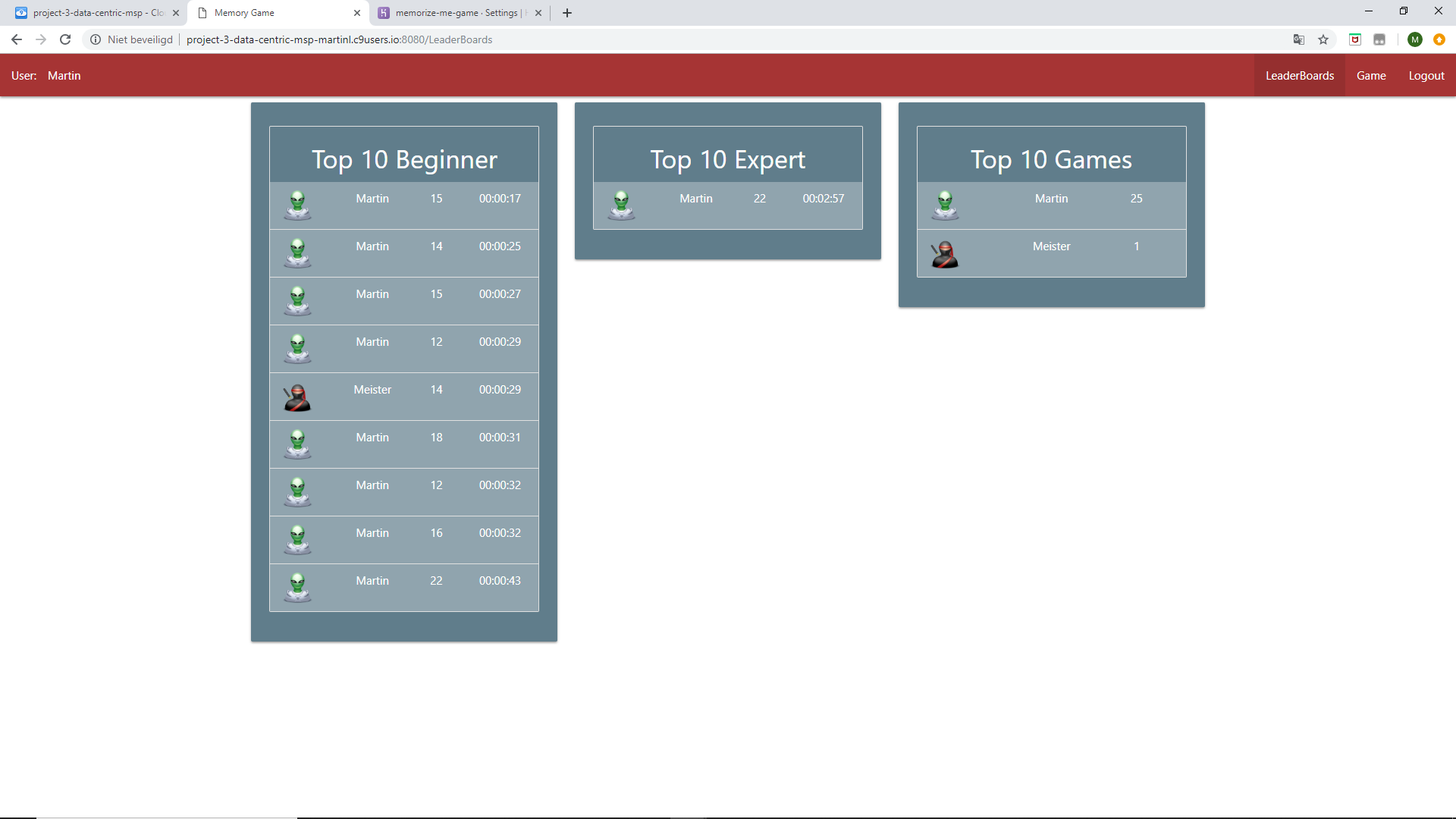
Finished 4 by 4 game



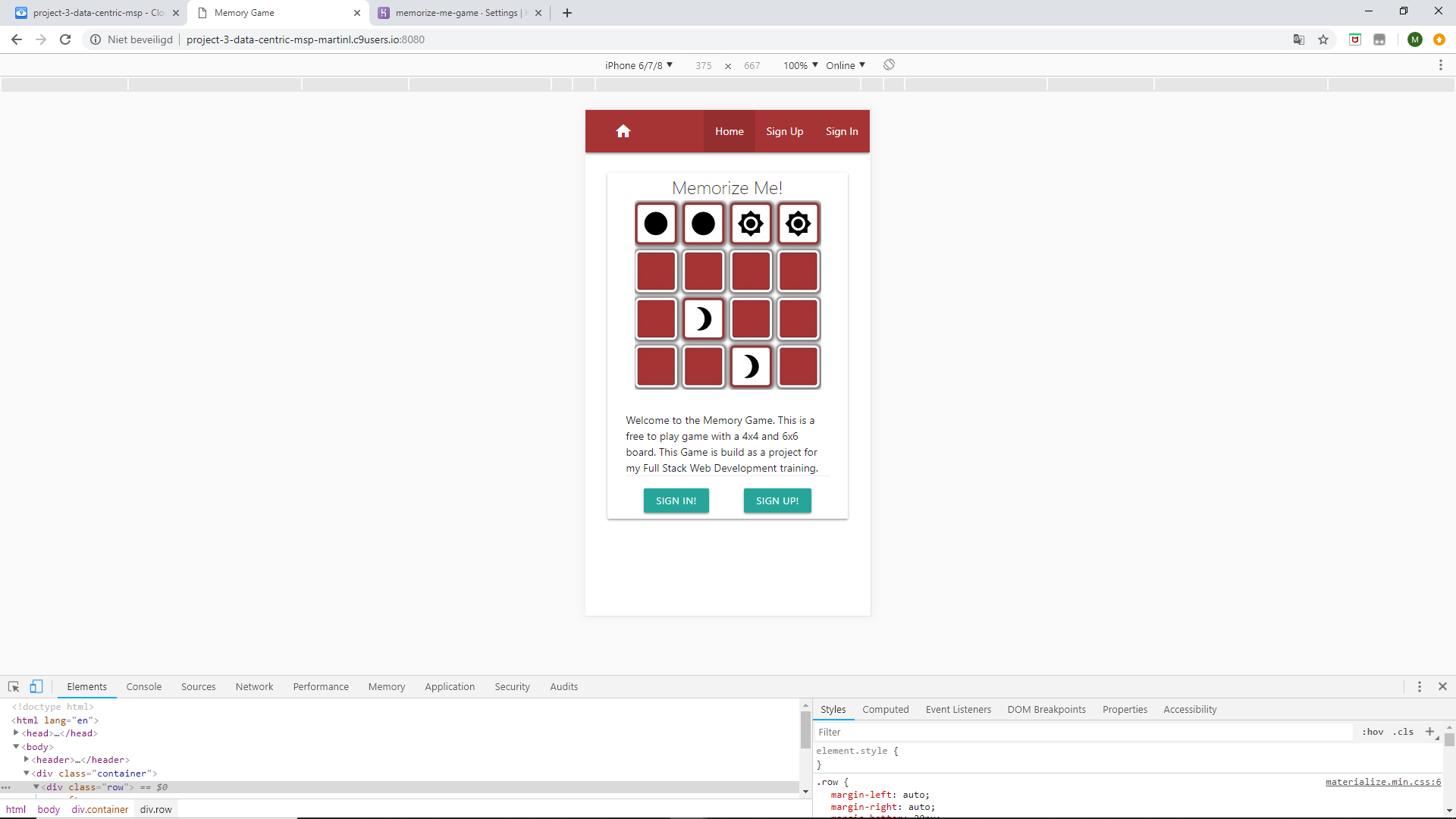
Finished 6 by 6:

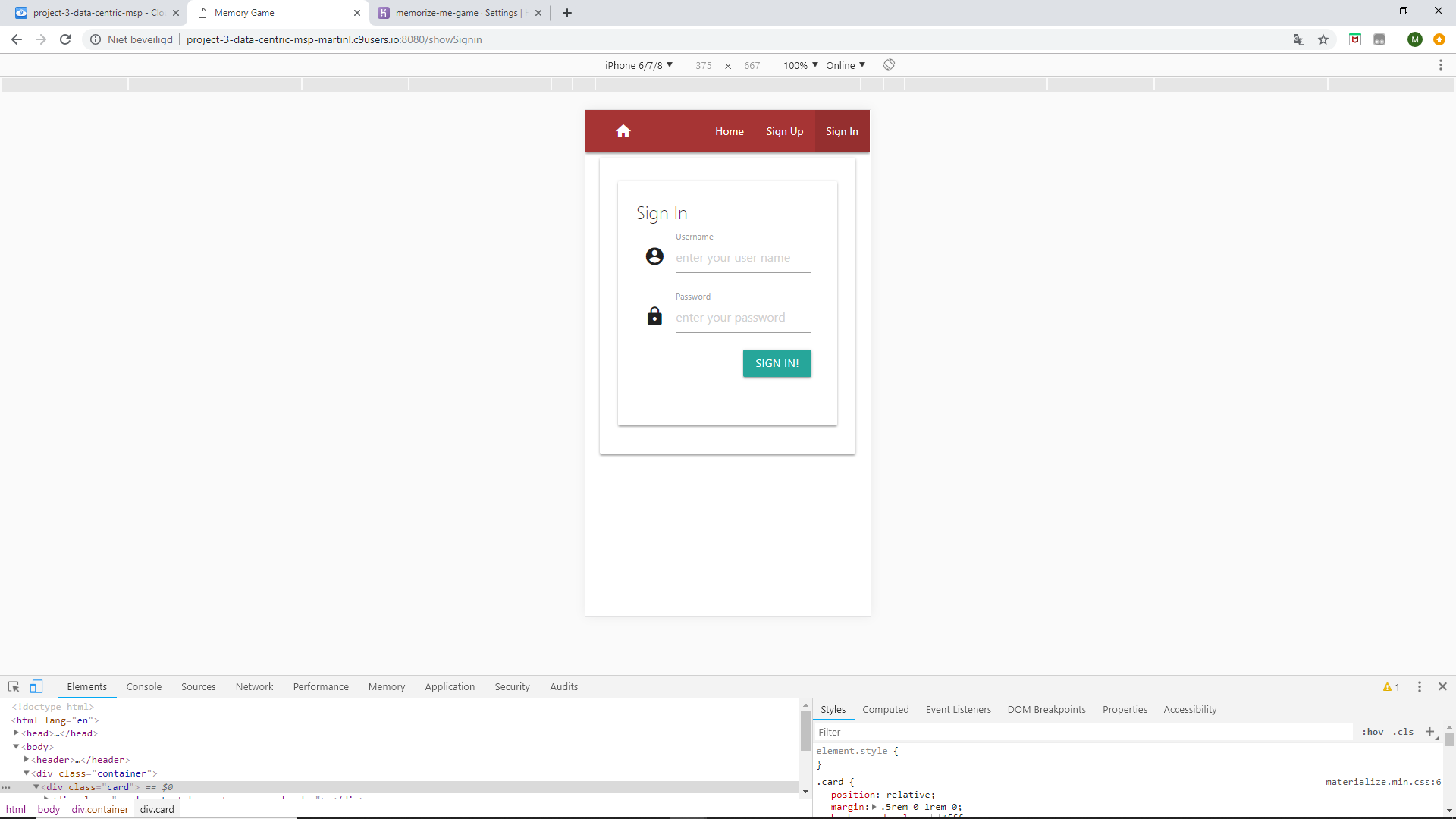


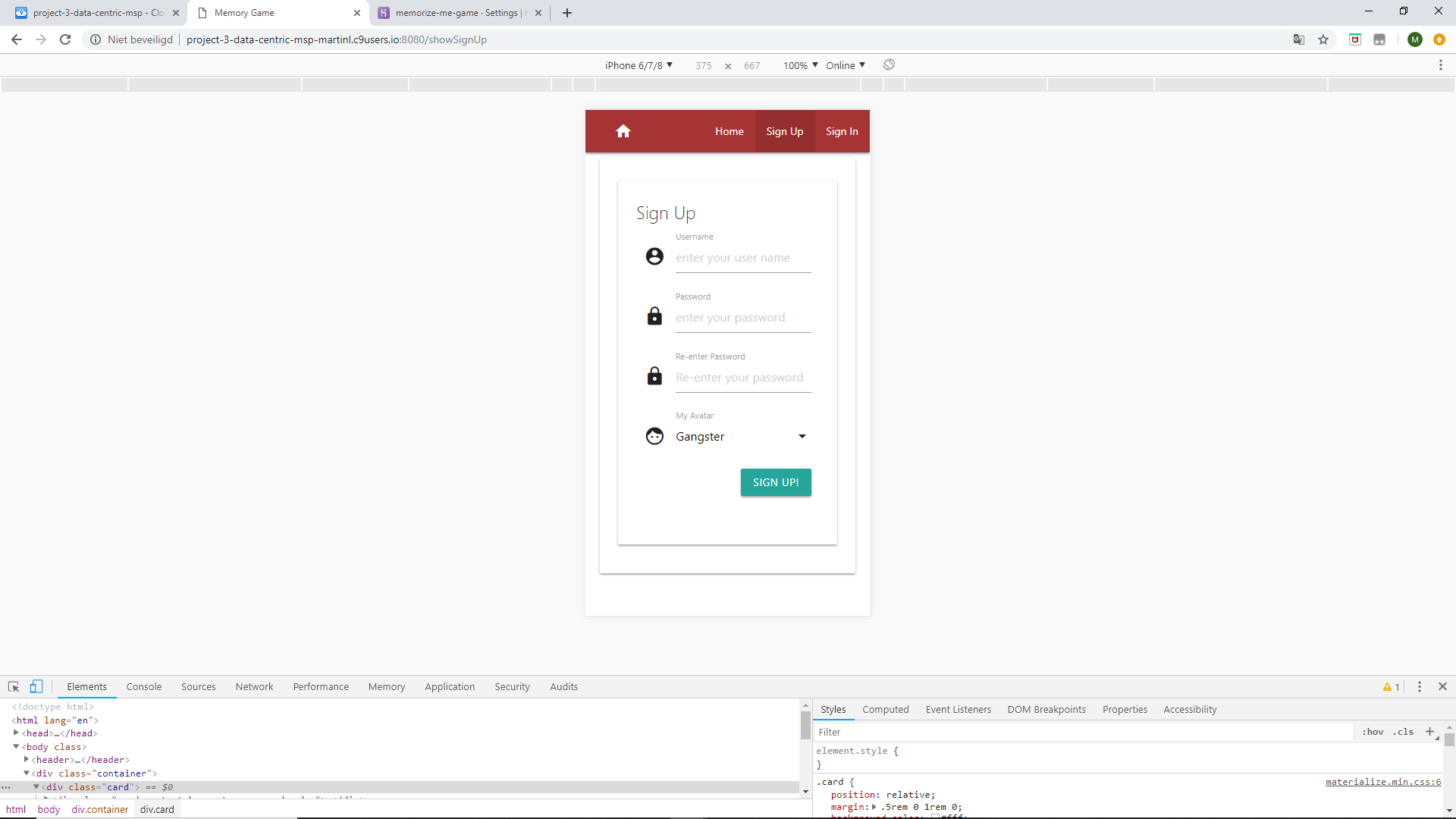
Leaderboards

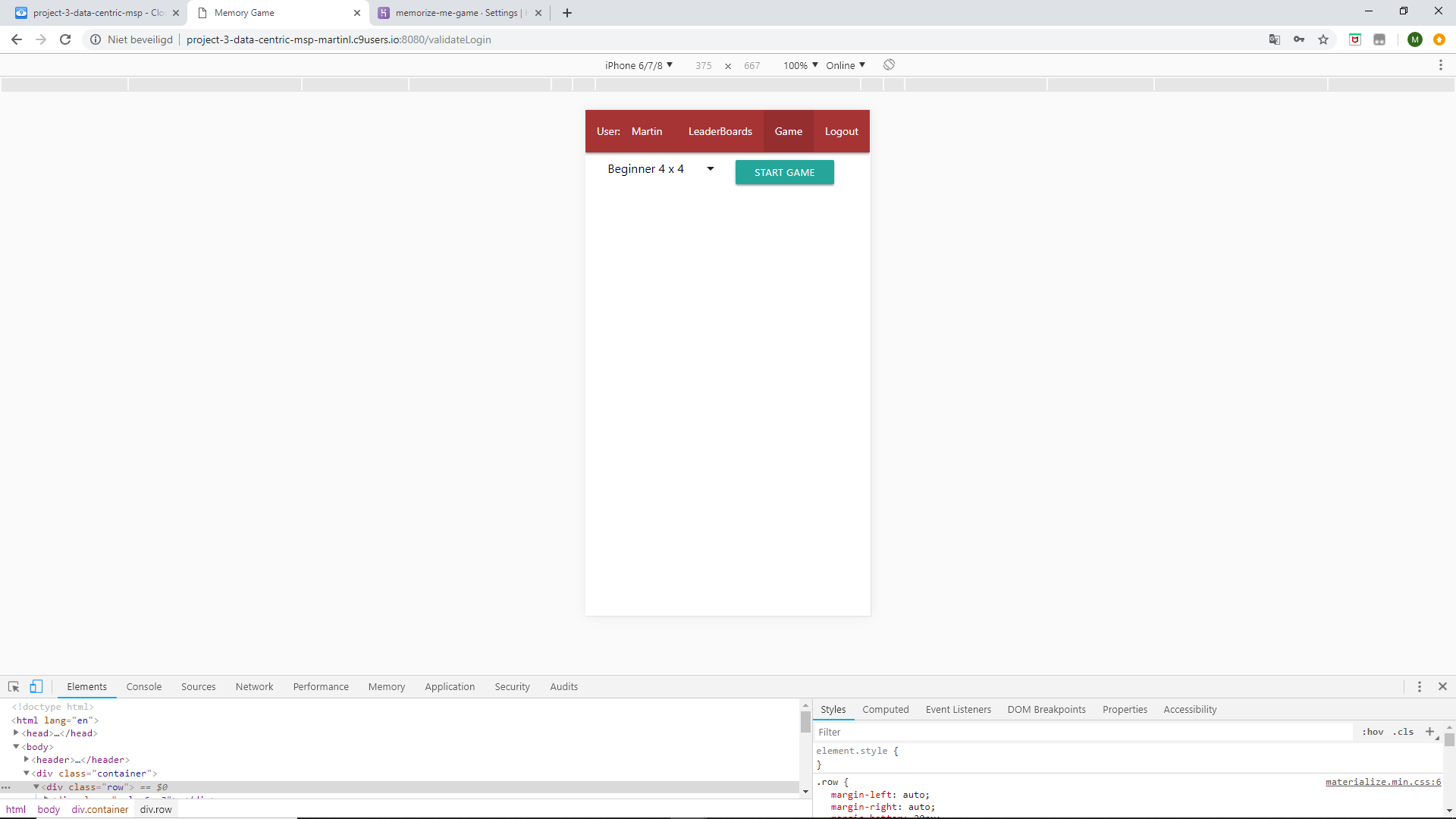


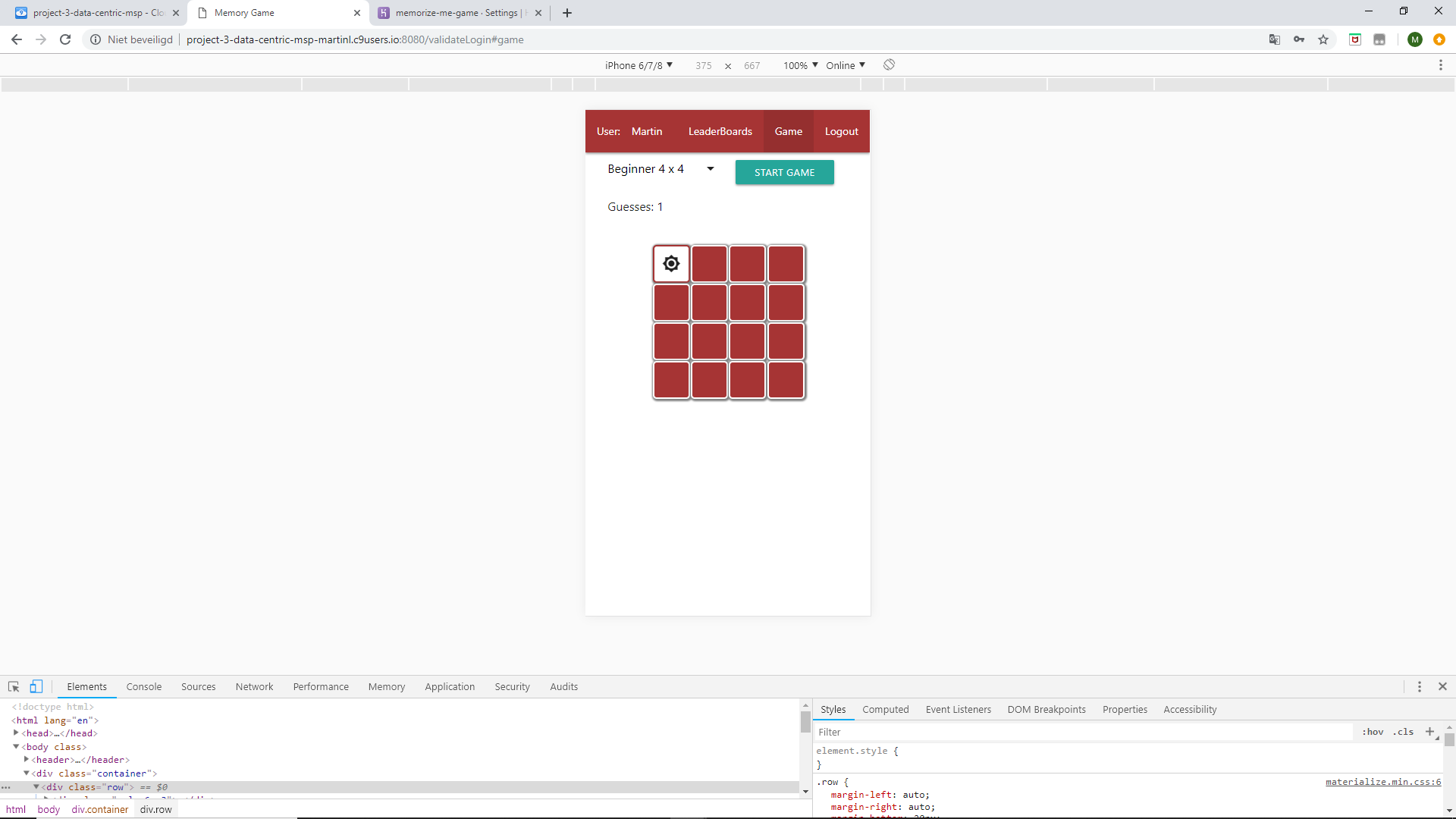
Mobile images (simulated with Chrome developer tools) (iphone 6/7/8)

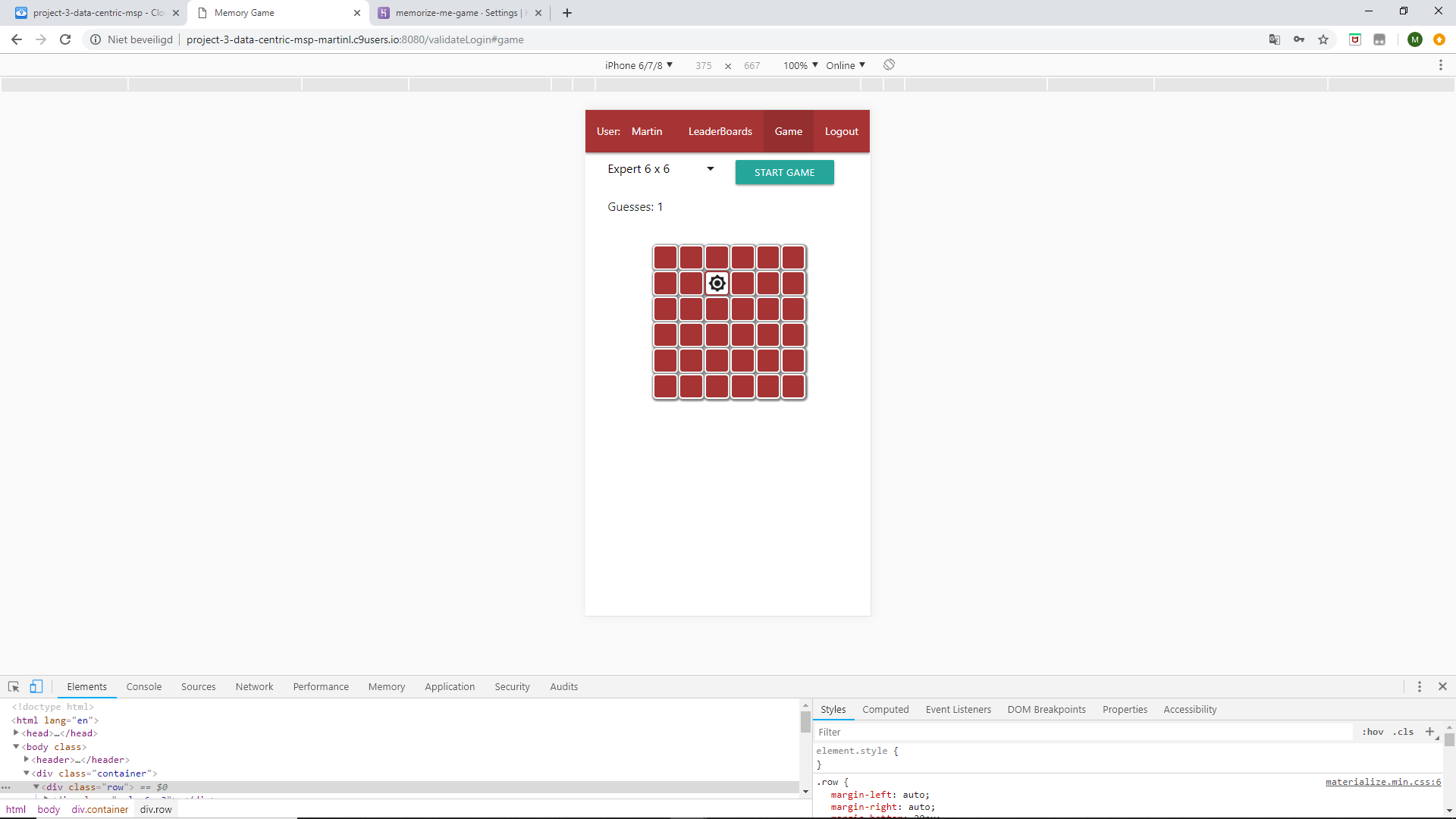


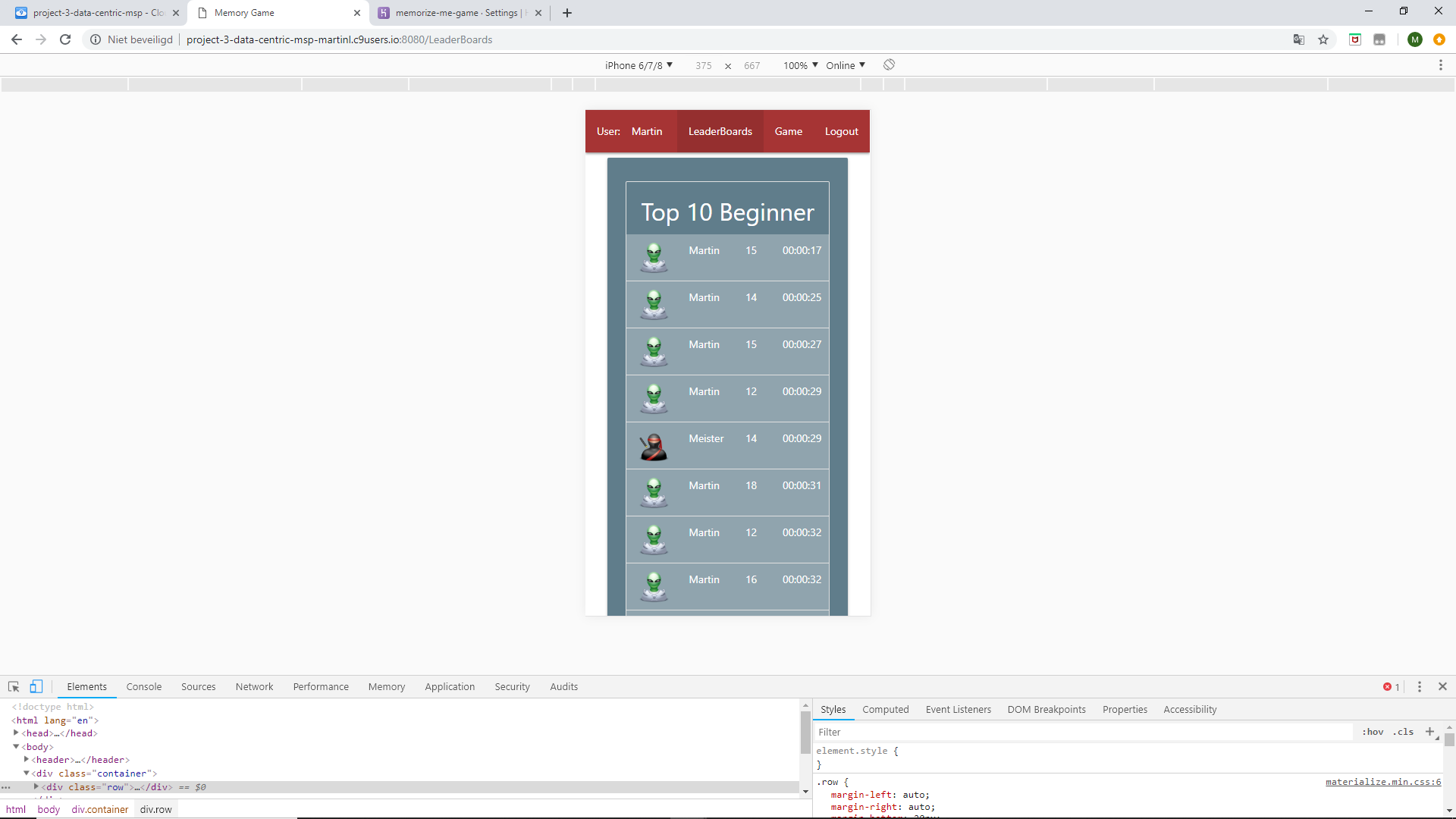




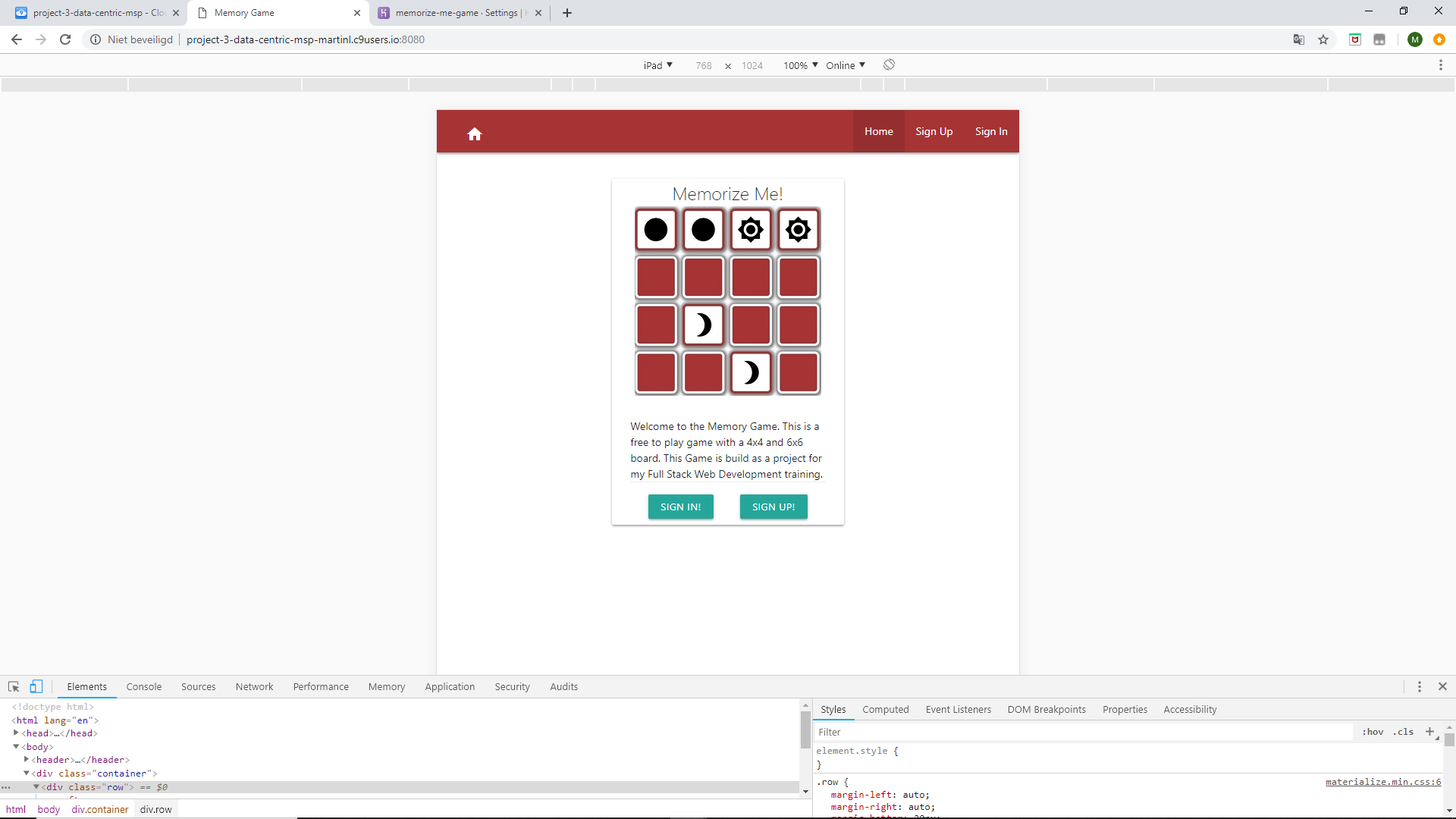


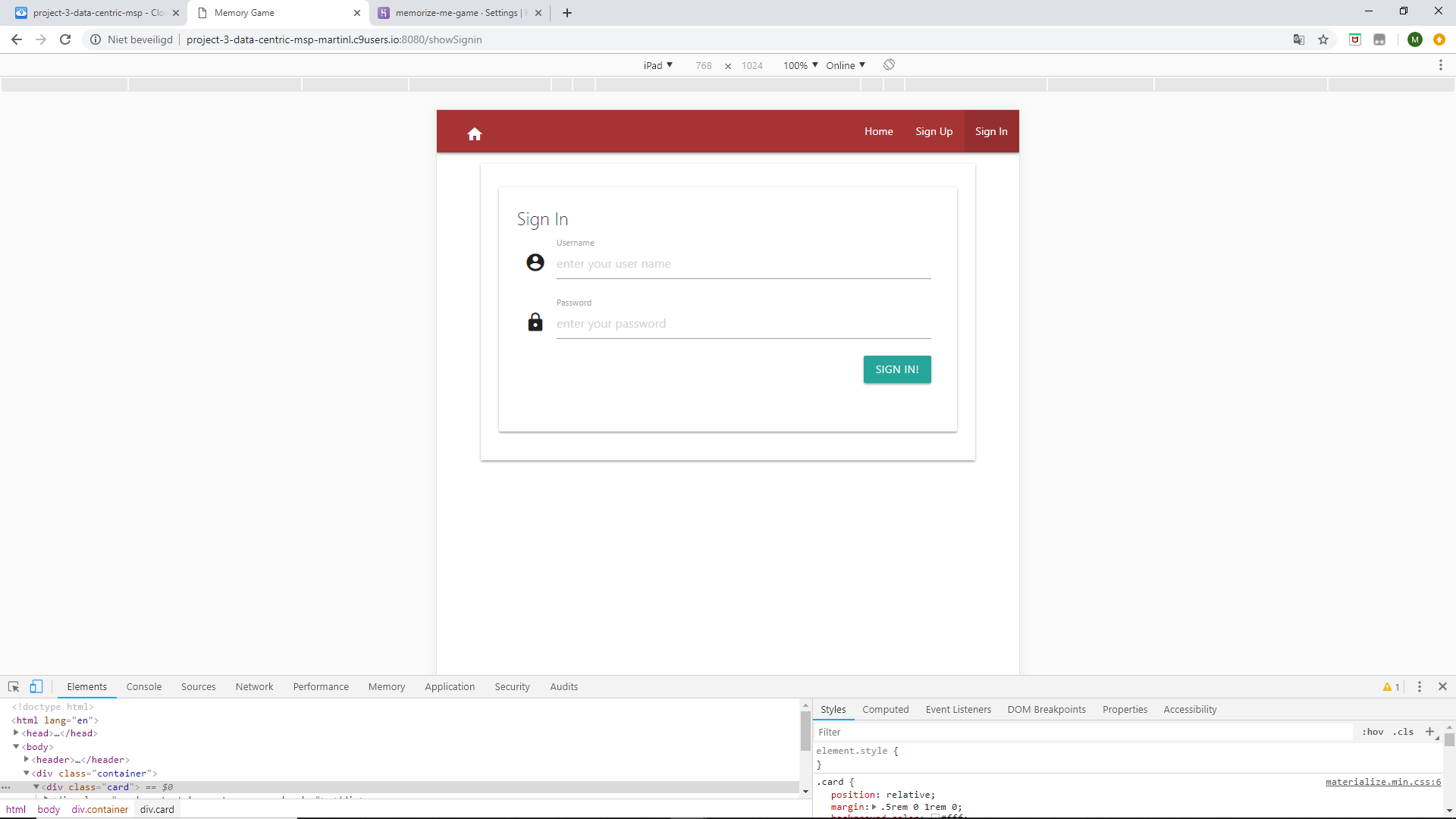


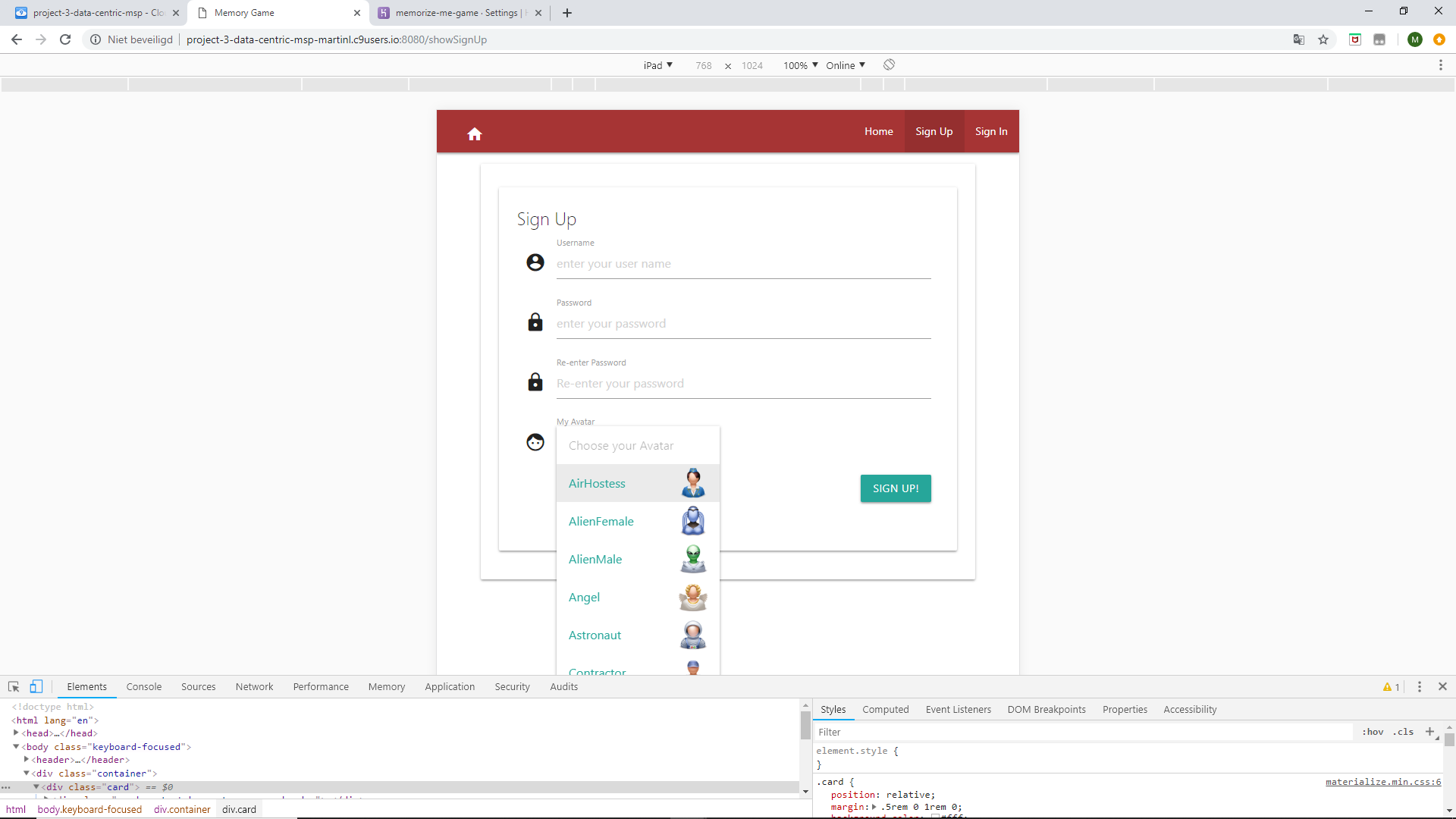


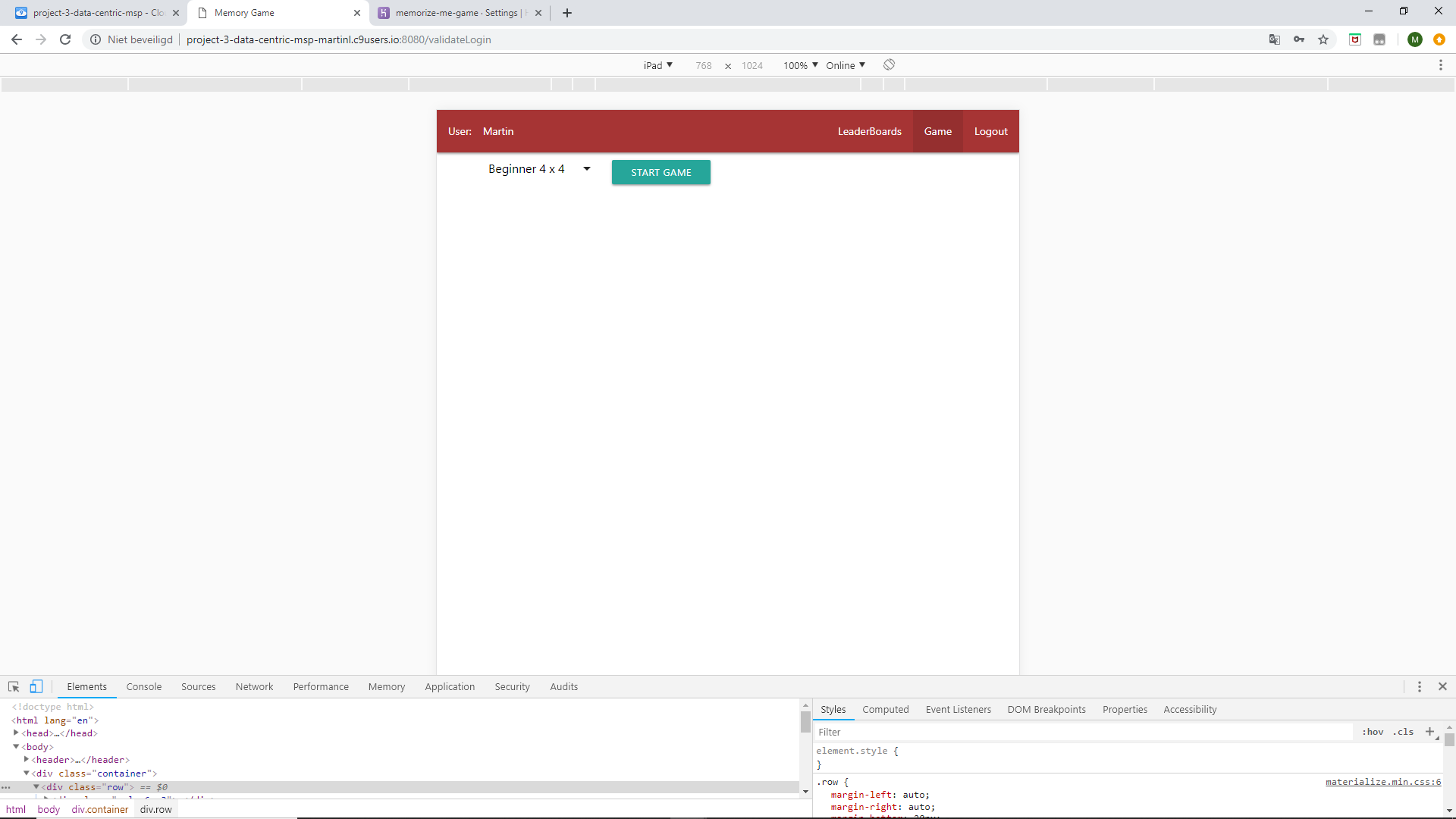


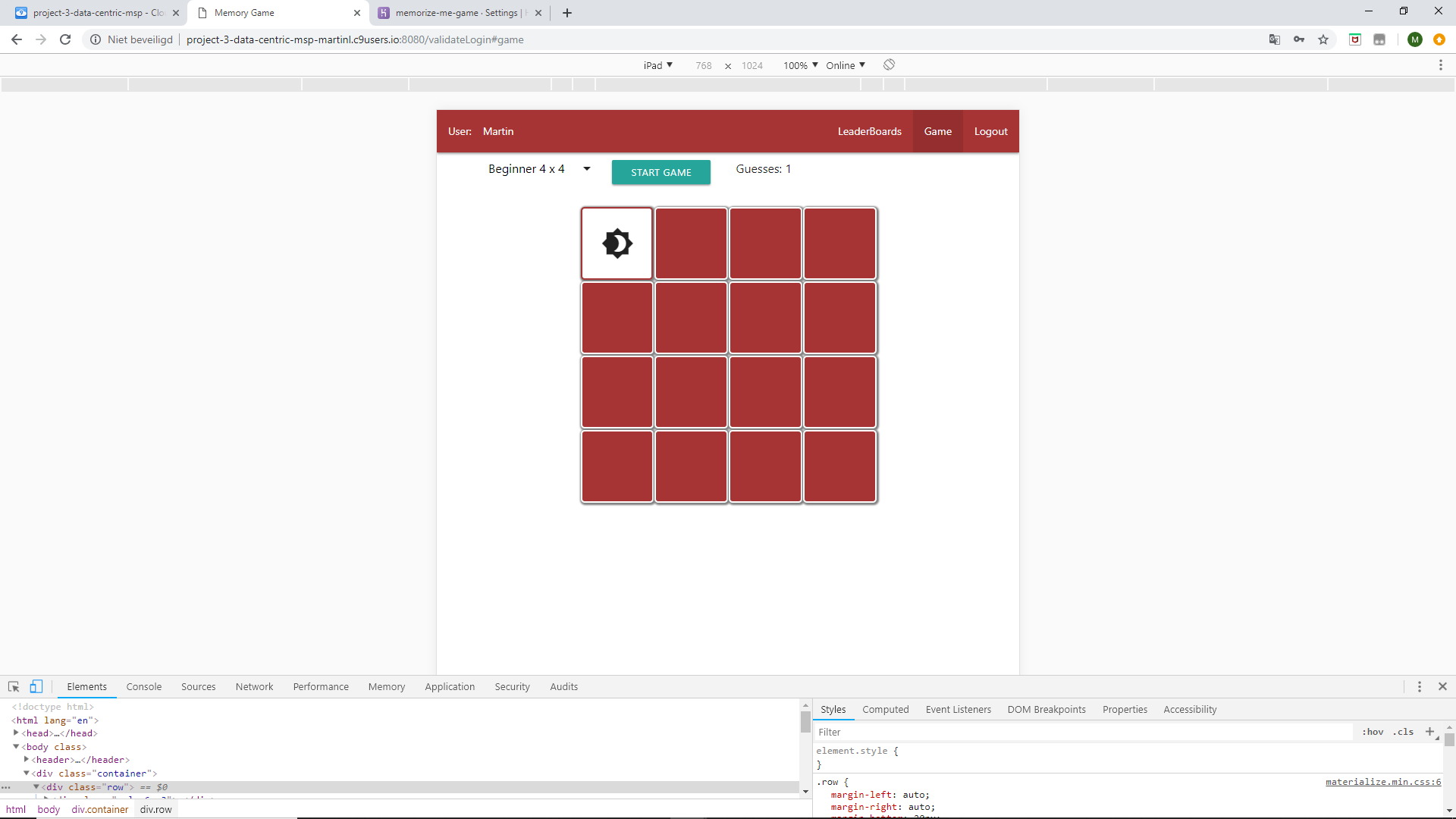
Mobile images (simulated with Chrome developer tools) (iPad)

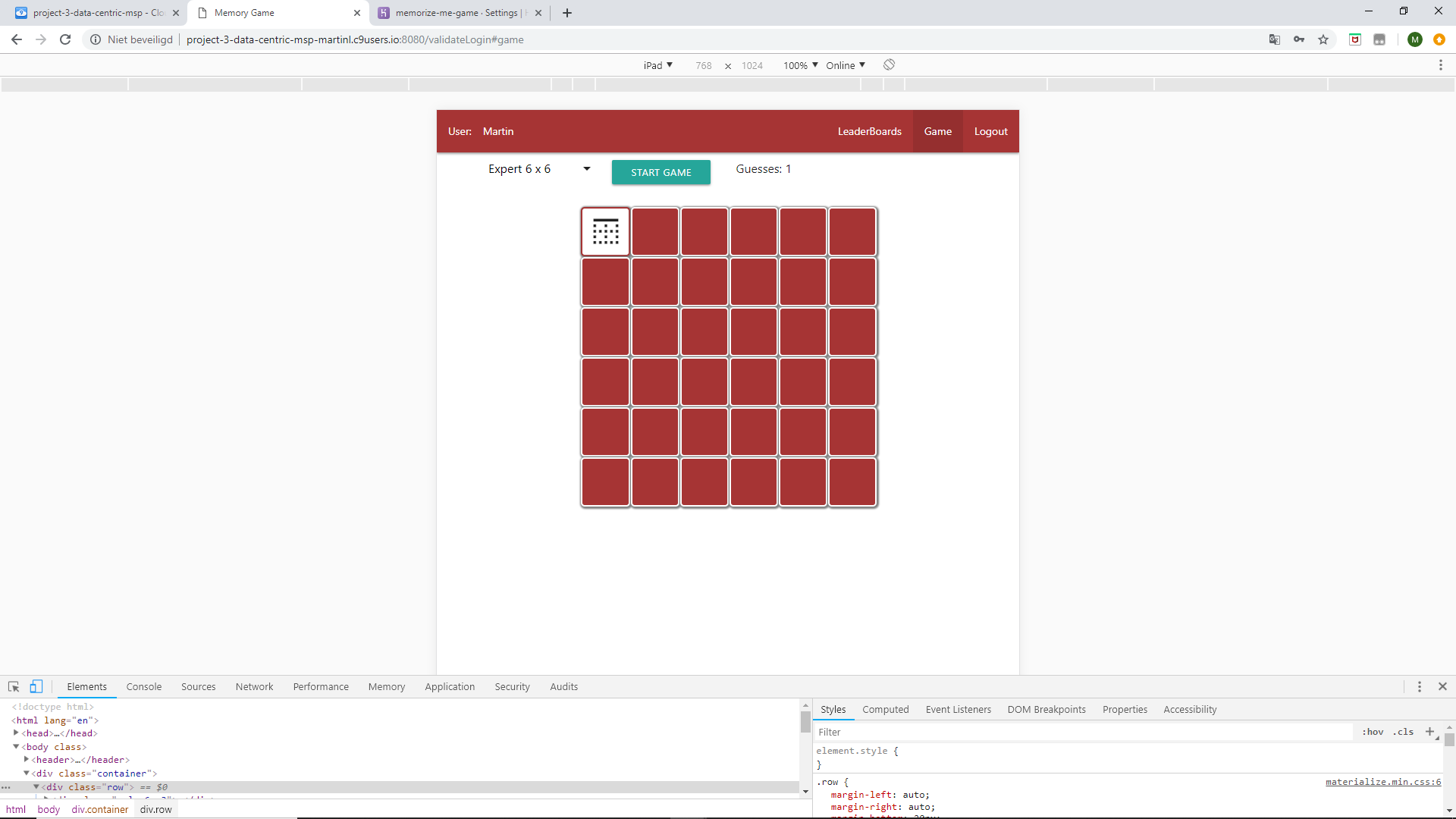


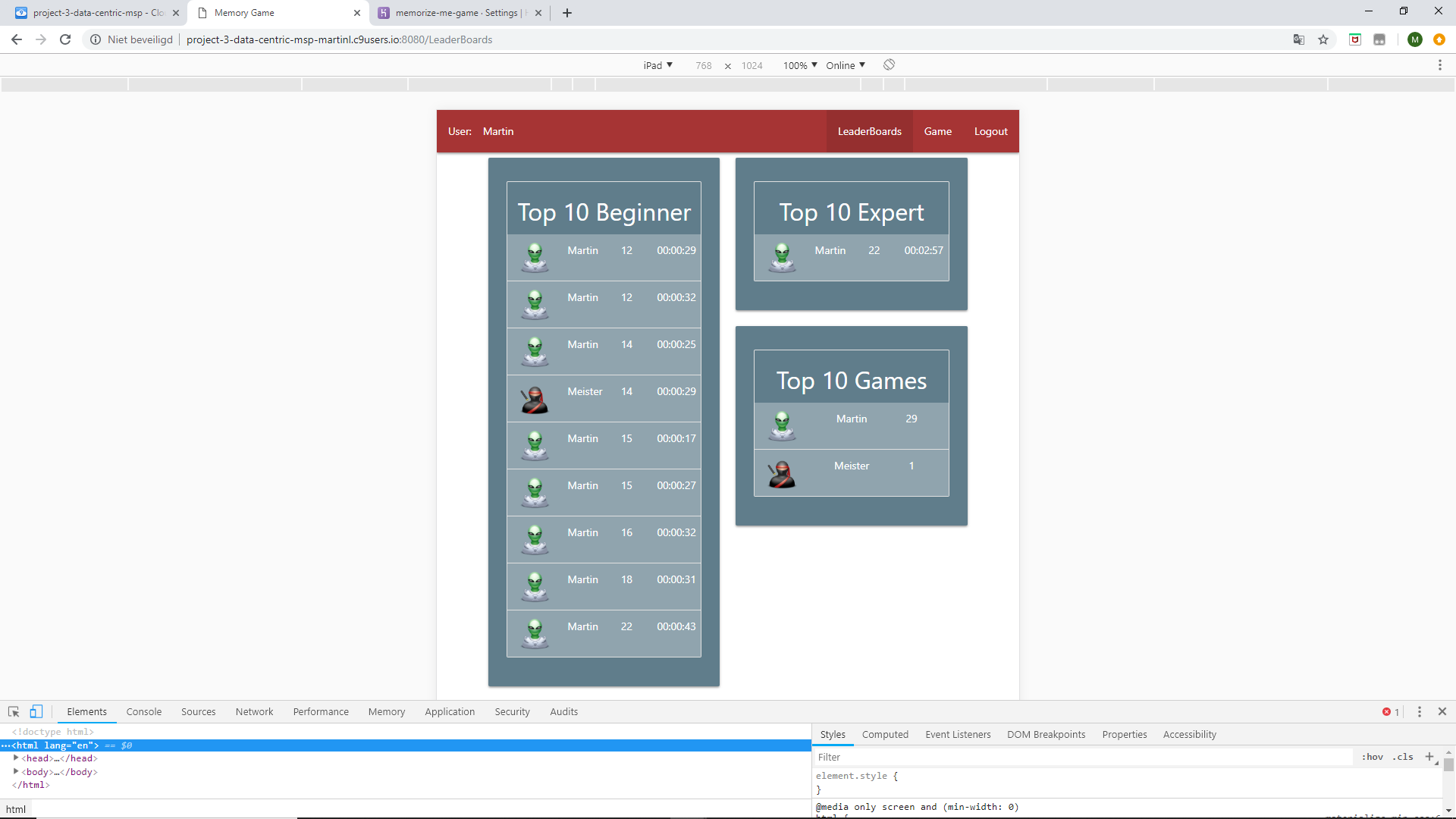












.

## Bugs encountered throughout

Throughout, small bugs were encountered while creating the site.

|  |  |
| --- | --- |
| Tiles didn’t flip back after starting a game after solving a previous game | Counter correct was not reset |
| Board hide itself after starting a new game | Callback function clear board wasn’t in place, so after the board was cleared it hide itself instead of showing a fresh new board. |
| Site on Heroku wasn’t working, missing secret key | Used capital SECRET\_KEY versus lower\_case secret\_key, which was clearly not very clever |
|  |  |
|  |  |

Other than the above, bugs encountered usually revolved around incorrect indentation, or around modifications to HTML leading to unclosed tags screwing the layout up entirely.

None worth mentioning aside from the above.

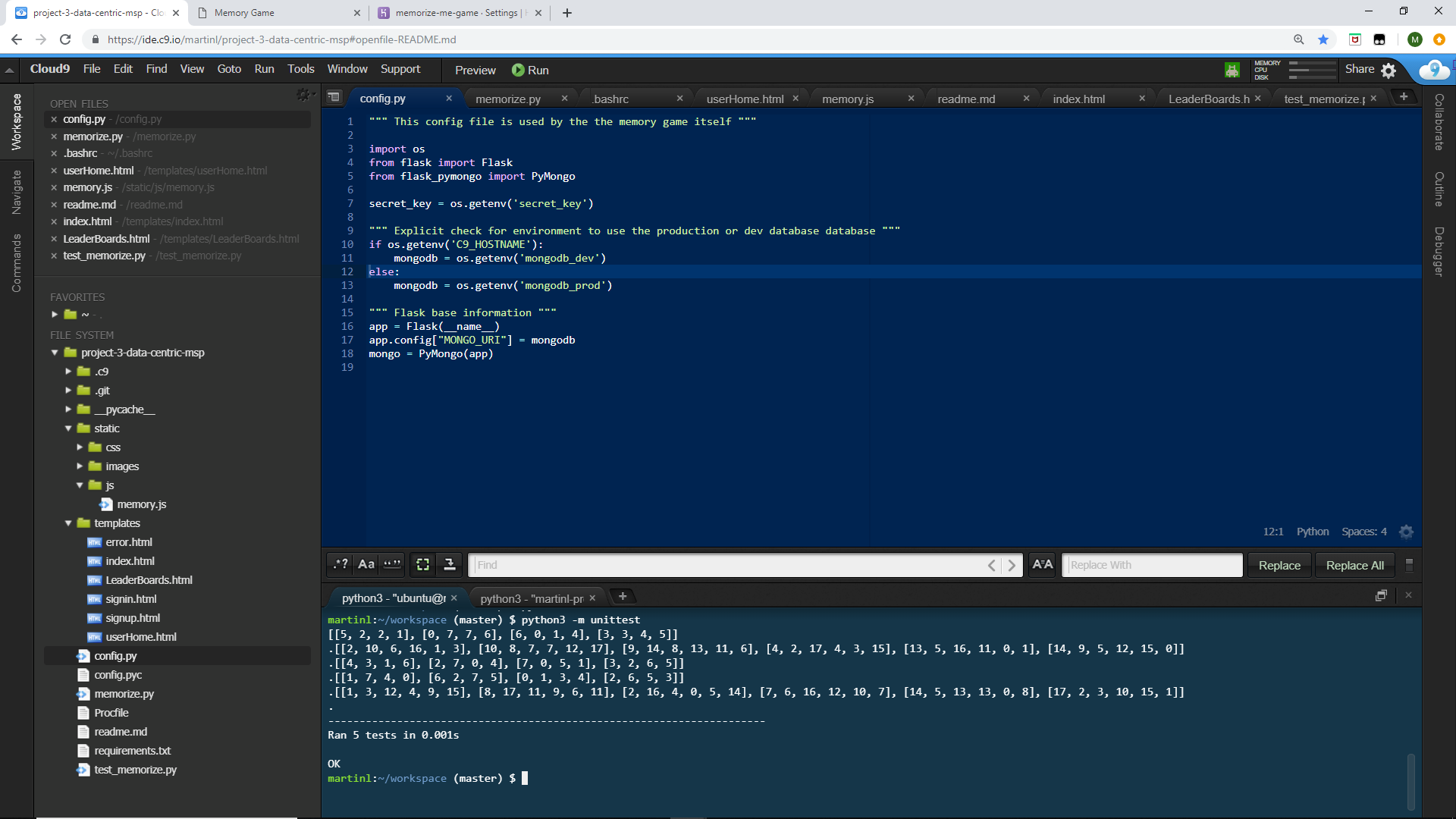
## Testing

|  |  |
| --- | --- |
| Initial Page | * On first installation, a base.html was made, followed by a simple index page. |
| Add mongo to the site | * User registration and user login basics tested. * User stored in mongo, test login functionality. * Check password hashing for ‘saltiness’ by creating multiple users with the same password and comparing the hashes. |
| Expand user creation logic to test for uniqueness and minimal requirements | * Create a duplicate username and receive an error * Try mismatching passwords and receive error * Check message returned on any of the above * When correct, check addition into database * When correct, check correct re-routing to user home page |
| Try to enter a page without logging in | * Confirm 401 error with Black Widow image * Confirm for user home, user settings, admin page, add comics, read comics, add all comics |
| Start a New Game | * Check if the board is 4 by 4 (visual) or 6 by 6 |
| Check moves count | * Manual count the moves on paper and compare with the counter on the screem |
| Check time | * Start external stopwatch at the moment you press start new game, and stop it after flipping the last tile |
| Game working | * Play after every modification a game to test if all is still working |
| Leaderboard sortation | * Manipulate scores and times in the database in order to see if there are 10 results shown * Manipulate scores and times in the database order to see if there sortation is working (least amount of moves and if equal shortest time) |
|  |  |

## Unit tests

Unit tests were created to perform some automated tests alongside the visual testing by hand.

### Screenshot:



## Database information

Note: while different databases, development and production have the exact same composition of data in the collections.

As described in the README.md, there are 5 collections to the database:

* tblUsers
* tblBeginner
* tblExpert
* tblAvatar
* tblGamesPlayed

Below examples of documents in each collection are given, along with reasoning:

### Collection ‘tblUsers’:

Example:

|  |
| --- |
| {  "\_id": {  "$oid": "5c766b19208ccd14123f4ab5"  },  "username": "Martin",  "pwd": "pbkdf2:sha256:50000$zn3QS7LQ$80bde3a3ce8af327ee73e9fd2bce59508c12c9563a54379dda3022cfee6f1e3c",  "avatar": "/static/images/alien-male-icon.png"  } |

A user will with the chosen username.

As can be seen, passwords are stored hashed. This is an added bit of security to avoid plaintext passwords being available.

Avatar is registered to personalize the account for the leaderboards

### Collection ‘tblBeginner’:

Example:

|  |
| --- |
| {  "\_id": {  "$oid": "5c78019d208ccd23d7256601"  },  "Username": "Martin",  "Moves": 15,  "Avatar": "/static/images/alien-male-icon.png",  "Time": "00:00:17",  "Size": "4"  } |

Contains the scores of users whoch played the 4 by 4 board.

Username, the time it took them to solve the game, amount of moves they needed, the avatar of the player is added for the the leaderboard.

### Collection ‘tblExpert’:

Example:

|  |
| --- |
| {  "\_id": {  "$oid": "5c78019d208ccd23d7256601"  },  "Username": "Martin",  "Moves":62,  "Avatar": "/static/images/alien-male-icon.png",  "Time": "00:50:17",  "Size": "6"  } |

Contains the scores of users whoch played the 6 by 6 board.

Username, the time it took them to solve the game, amount of moves they needed, the avatar of the player is added for the the leaderboard.

### Collection ‘tblGamesPlayed’:

Example:

|  |
| --- |
| {  "\_id": {  "$oid": "5c780172208ccd23d7256600"  },  "Games": 29,  "Avatar": "/static/images/alien-male-icon.png",  "Username": "Martin"  } |

After every click on start new game, the entry of the player in this table is updated. The Amount of Games is added with 1. I made the decision to make no difference in type (4by4 or 6by6) of games that is played.

### Collection ‘tblAvatar’:

Example:

|  |
| --- |
| {  "\_id": {  "$oid": "5c7667b1e7179a3e36de18de"  },  "avatar\_id": "2",  "link": "/static/images/airhostess-woman-icon.png"  } |

Contains the available avatars which a player can chose from. It contains the unique avatar ID with the link to the image file.