CHOMPER

GAME WEBSITE

GAME WEBSITE REPORT

By Syed Nasiruddin

Game Overview

The game is Chomper, like Pacman, the user controls a yellow orb that eats pellets, but instead there are no walls within the map, and the ghosts bounce across the walls randomly.

It is a player vs computer game, and the goal is to accumulate the highest score possible while dodging the ghosts.

Chomper can be moved to upwards, downwards, left, and right when the user holds down the keyboard's "W," "A," "S," and "D" keys, respectively.

The player can eat the power up in the middle of the map, which will allow them to eat ghosts for a brief period.

The map is reset with pellets and the power up once again in case the player clears it, allowing the user to continue playing and contribute to their accumulated score. Each time the map is cleared, the ghosts start to bounce faster.

The player has two extra lives, and upon collision with the ghosts these lives are deducted until the player has no more and the game ends.

Scoring

Each pellet is worth ten points, when the player is powered-up and eats the ghosts, each ghost is worth a hundred points.

Objects

The main objects in the game include the boundary, pellets, power ups, Chomper and the ghosts.

Scripts

There are three main scripts:

- game.js
  It uses classes from ghostClass.js, boundaryClass.js, chomperClass.js, pelletClass.js, and powerUpClass.js. It is the script that runs the game.

- leaderboard.js
  This script is used to populate the leaderboard table.

- signupLoginLogout.js
  Used for the sign up, login and logout processes.

How it Works

General

The user can navigate the website using the navigation bar at the top, which includes links to the “About”, “Game”, “Leaderboard”, “Sign Up” and “Login” pages.

The user clicks the “enter” key to start the game on the “game” page.

The user can play the game without having to sign up and/or login but will be considered as a “guest” and their score will not be recorded for display in the leaderboard.

If the user does login, their name will be displayed below the game/canvas nest to “player:” and their score will be recorded to be displayed in the leaderboard, and next to “hi—score:” in the game page.

Sign Up, Login, and Logout

When the user signs up successfully, the values from all the input fields in the “Sign Up” page are stored in the local storage inside a “users” array.

When the user logs in, a “check” object is made and stored in session storage, this is used to know whether the user is currently logged in and if so, what is their username.

Also, the “Sign Up” and “Login” links are hidden, and the logout link is displayed after the user is logged in.

When logging out, this “check” object is removed, and links for “login” and “sign up” are shown once again.

Rankings / Leaderboard

The leaderboard page consists of a table that shows the position, username and scores of the users, sorted from highest to lowest.

Not only the top five or top ten scores are shown, but all the users are displayed, and the user can scroll through the table.

Once the game ends, the “highScore” property of the user in the “users” array is updated with the new score the user has achieved, only if it is higher than the currently stored value.

The table is then populated using the fillLeaderboardTable function in the leaderboard.js file, which gets all users from the “users” array in the local storage and creates a row using their username and highscore which it then adds to the innerHTML of the table.

Input Validation / Error Handling

Sign Up

The below are validated using regular expressions:

- For “Sign Up”, the user cannot enter special characters (e.g. [,] {, }, (,), &, #, $) for their first name, last name and username.
- Underscores and numbers are accepted in username, but not in first name and last name.
- Passwords can include lower and uppercase letters, numbers, dollar sign, ampersand, and period.

The user must choose a username that is not already taken/stored.

The confirm password and password values must match.

All the fields must be filled out before submission, username and password length must be greater than three and eight characters respectively.

If any of the above conditions are not met, an error message is displayed, and the sign-up process cannot continue.

Login

The user first must fill out both the fields to continue the login process.

The value from the username field is taken and is used to check whether the user exists within the “users” array in the local storage.

If not, an error is displayed, else it moves on and checks whether the entered password matches the stored password.

If so, the user has successfully logged in otherwise an error is shown.

Game

If the user clicks logout in the middle of the game, the game is cancelled and it says, “logged out”, and whatever score the user had (if greater than the stored highscore) is stored.

Reflection

This coursework allowed me to refine different concepts of html, css and javascript.

I was also able to revisit a few mathematical concepts while handling collision detection.

I learned how to:

- Animate different objects and shapes using canvas.
- Store user data in local and session storage.
- Display stored data in a table format.
- Detect collision between different shapes.
- Validate input and handle errors.

Development Issues

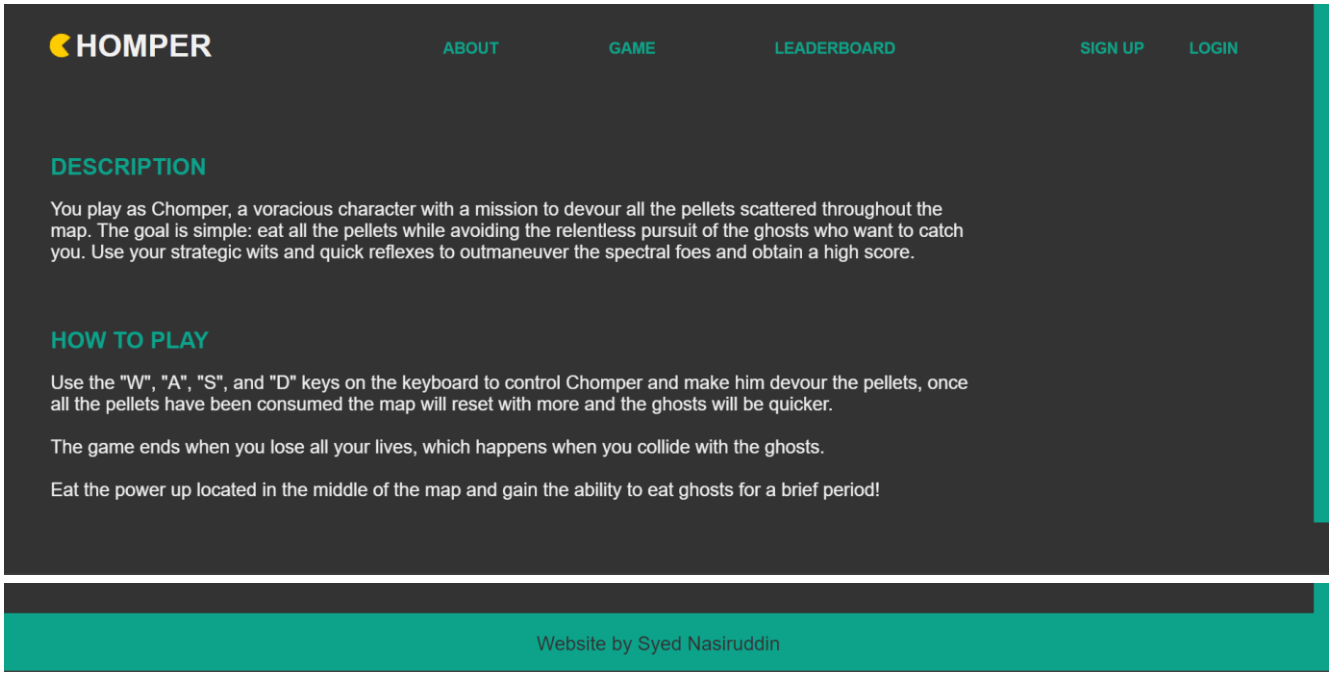
- **Collision Detection**
  After much trial and error, I was able to handle collisions with minimal error but to get to that point it involved a lot of testing, trying to convert the ghosts from circle to rectangles so I could handle it as rectangular-rectangular collision but that didn't work out, in the end I kept the ghosts as they are with their circular hitbox and worked around it to handle collision with the boundary.

- **Respawning Eaten Ghosts**
  This is being done using a setTimeout function, but it depends on a variable (chomper.movemnetRestricted) who's value changes in another setTimeout function, and this was causing the eaten ghosts to spawn and move even though chomper could not move.

Screenshots

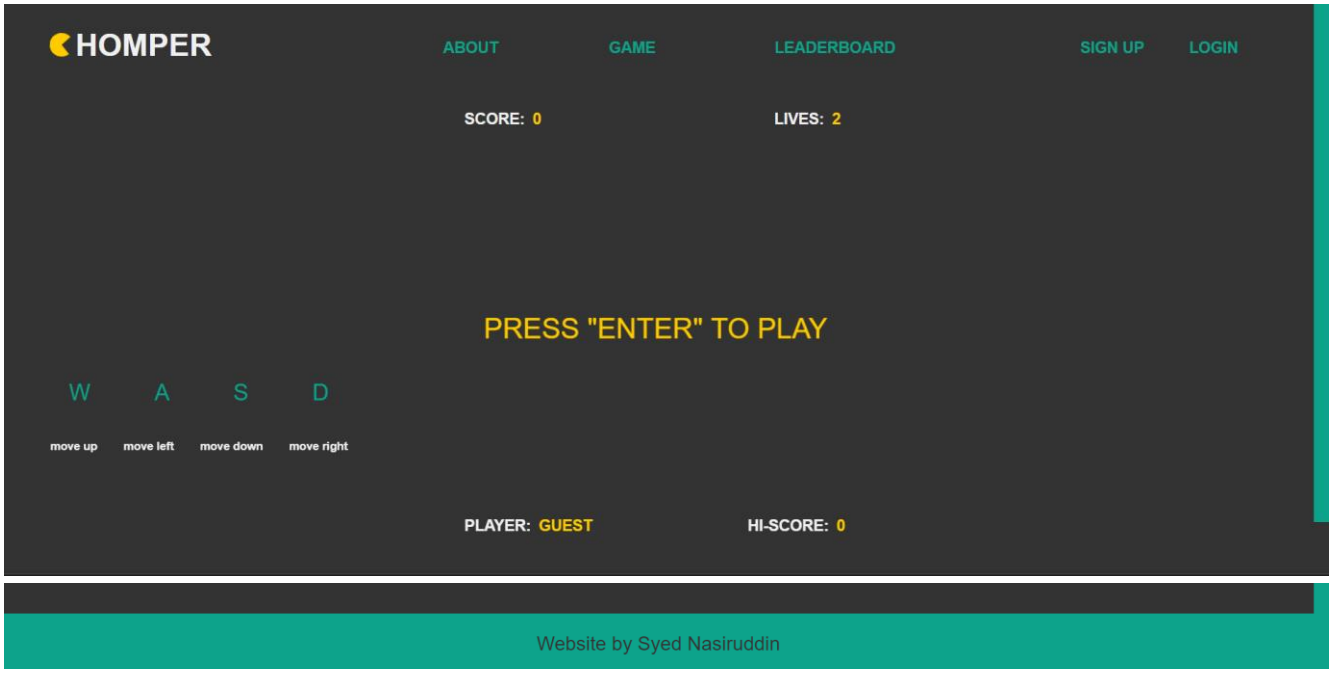
“About” Page

Gives the user a brief description about the game as well as instructions.

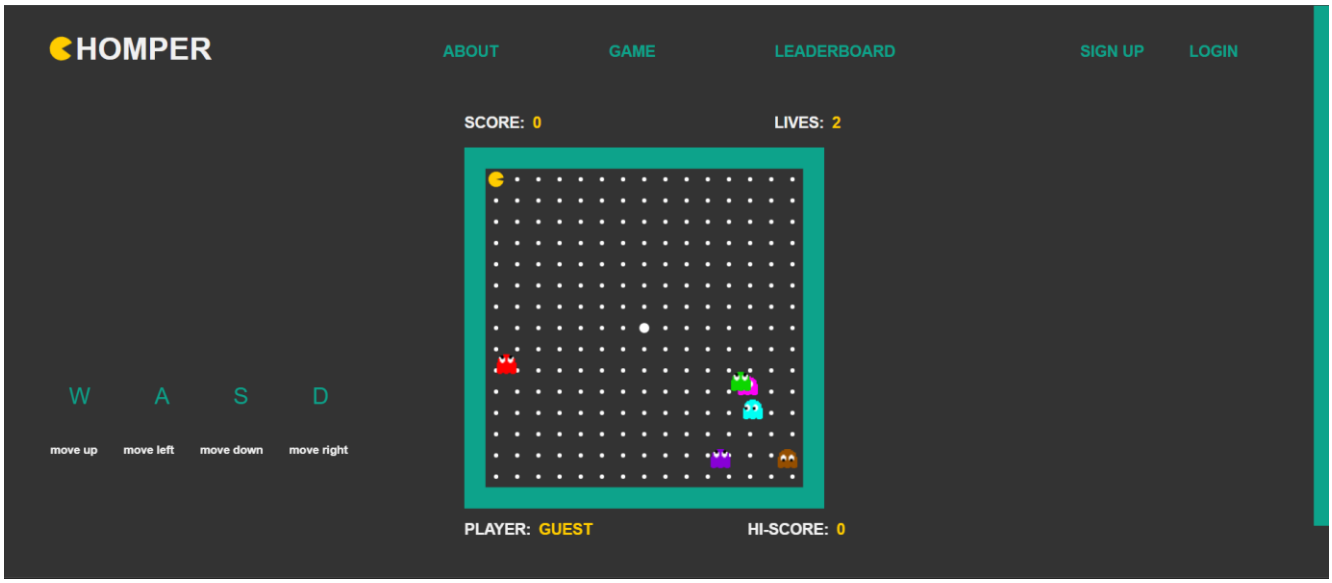


“Game” Page

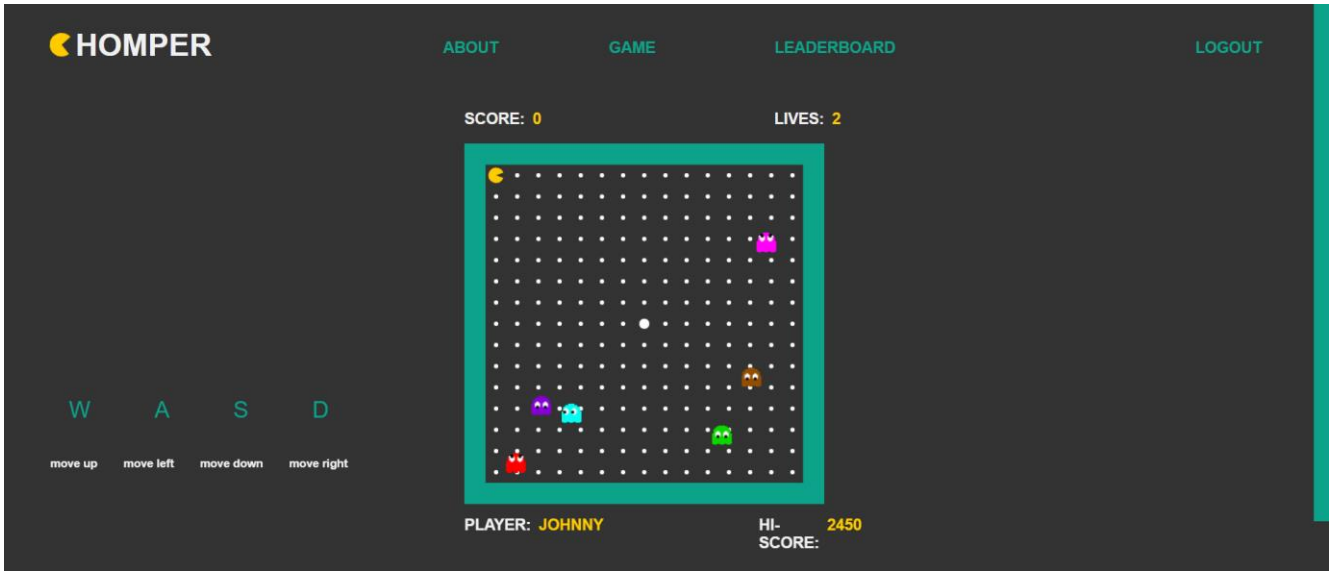
Starts the game after the user presses enter (note that at the under the canvas at the bottom left it says “player:guest” because the user is not logged in).



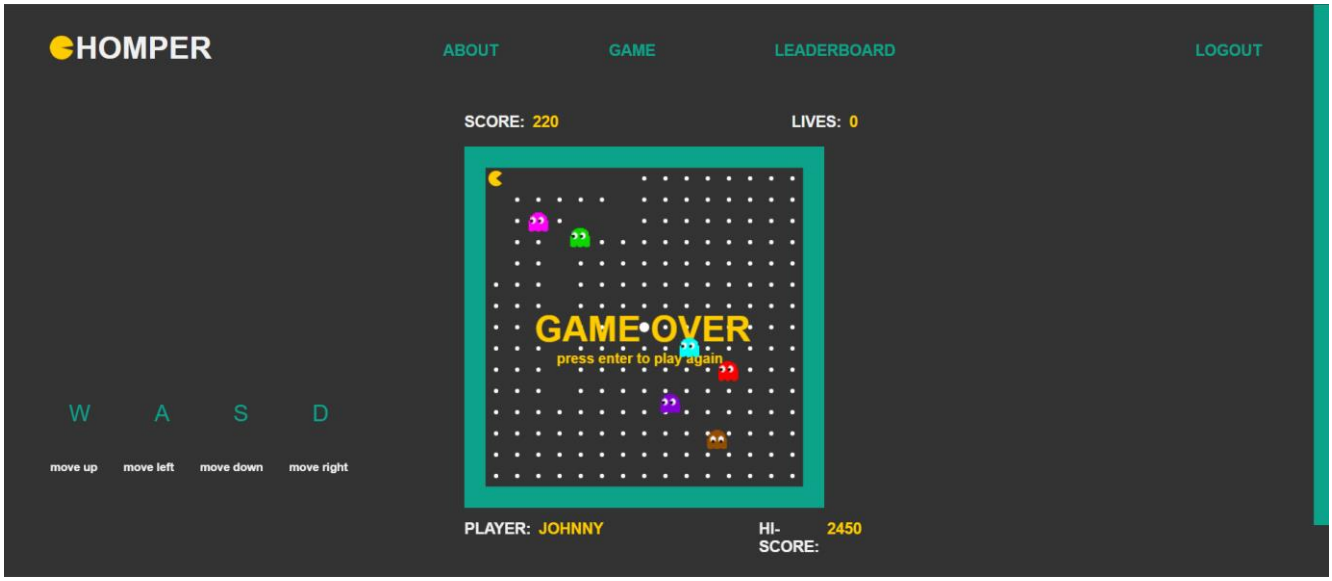
After the user presses enter.



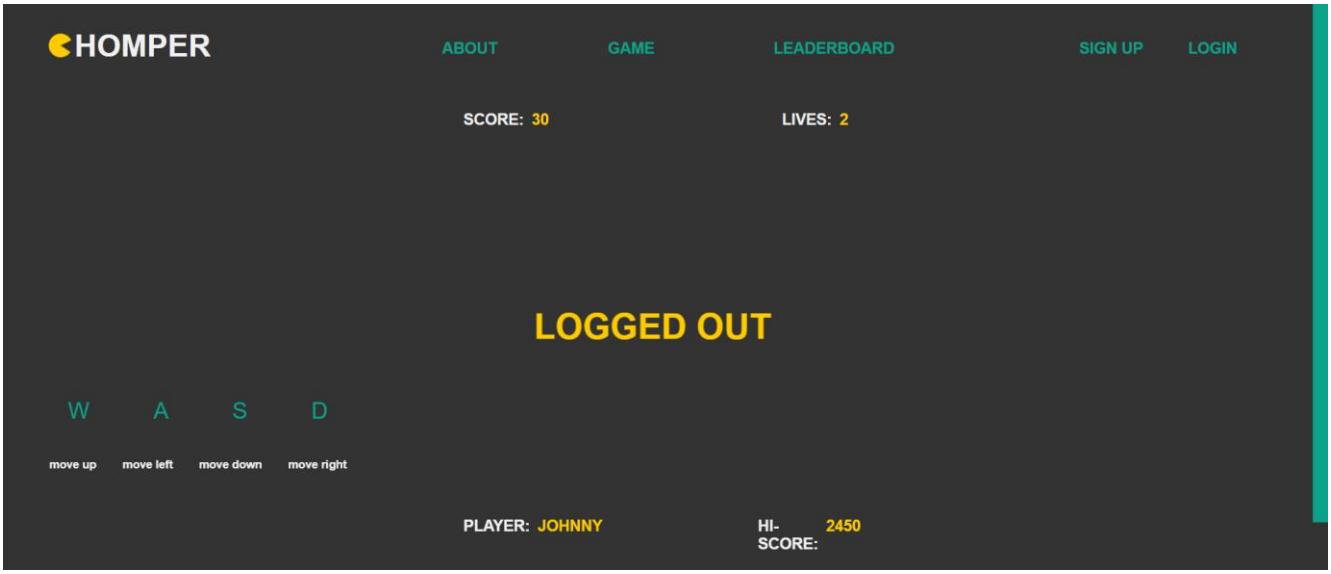
When the user is logged in (under canvas, “player: playername” and “hi-score: stored highscore”).



Game over screen.

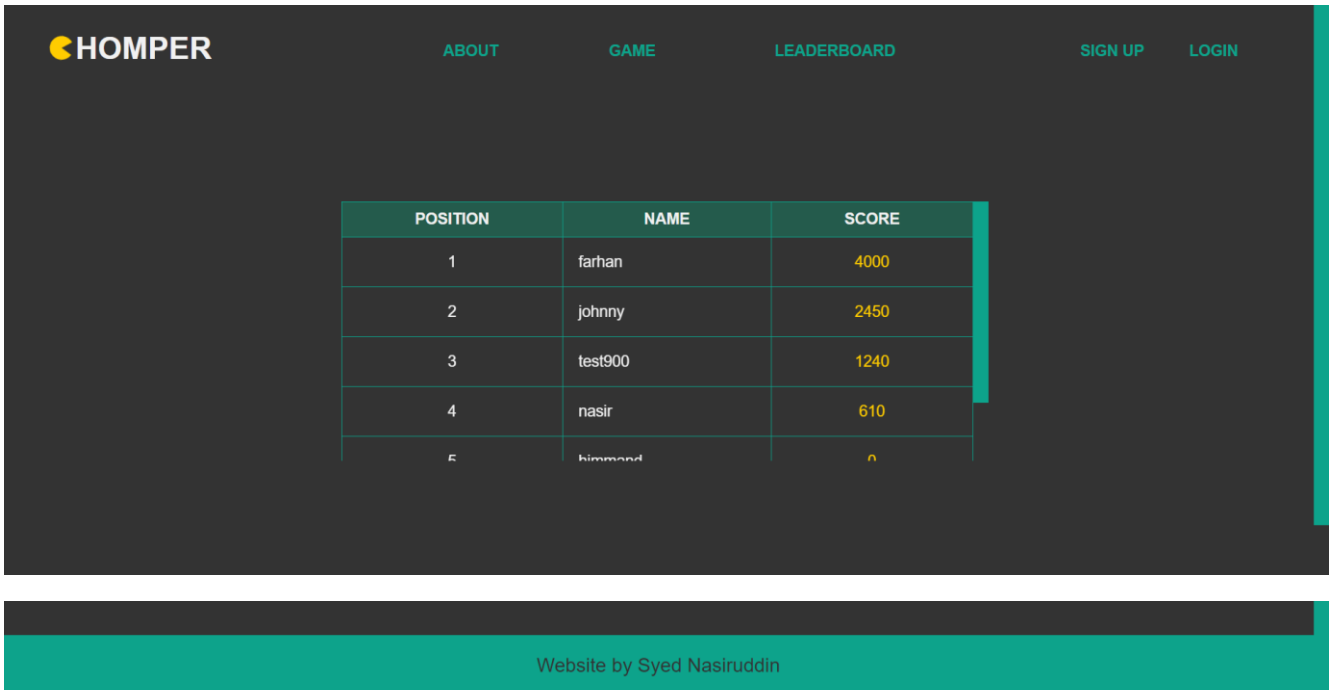


If the user logs out in the middle of the game.

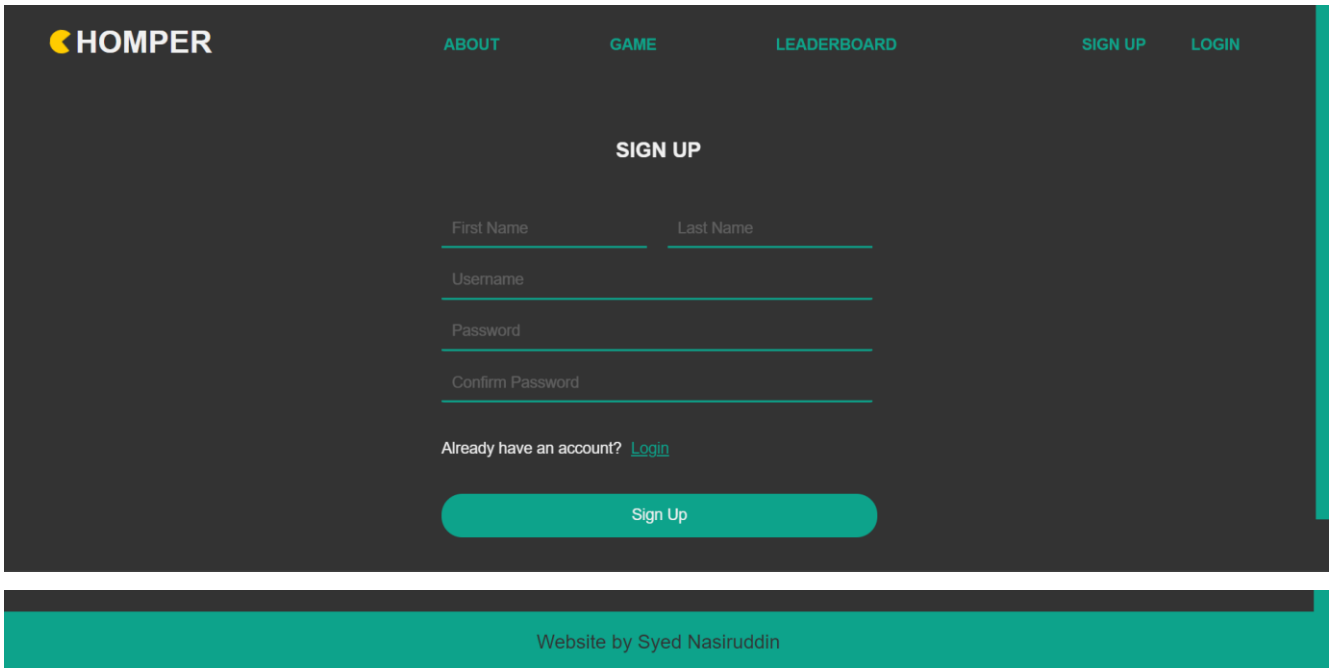


“Leaderboard” Page

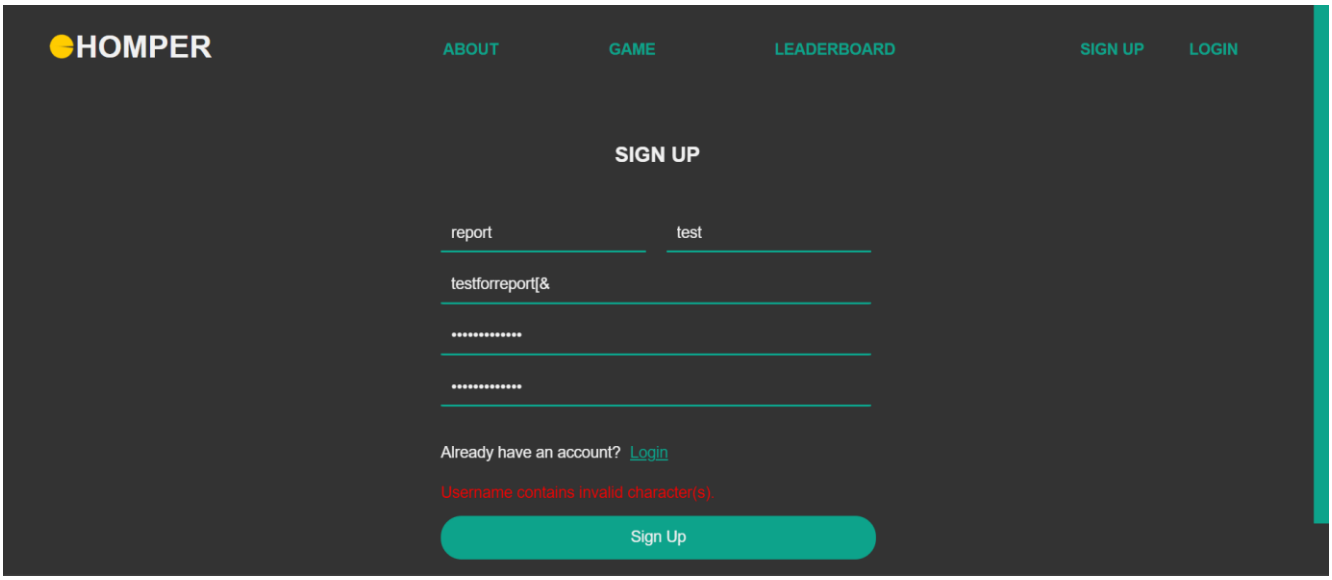
All stored users along with their scores displayed in descending order.



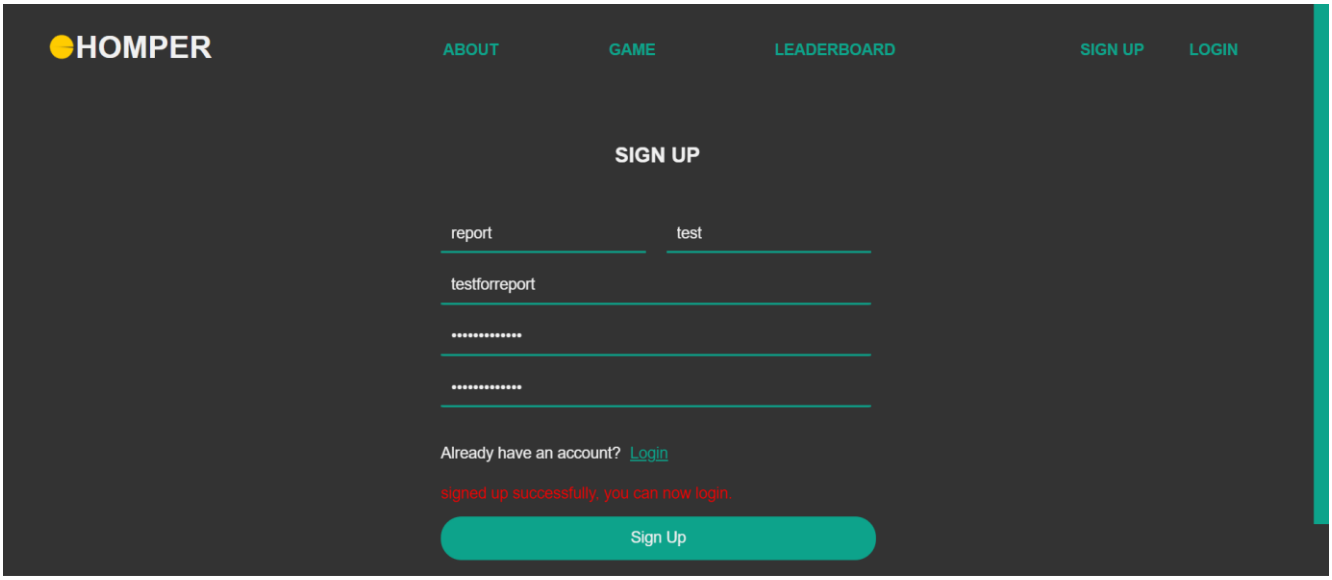
“Sign Up” Page



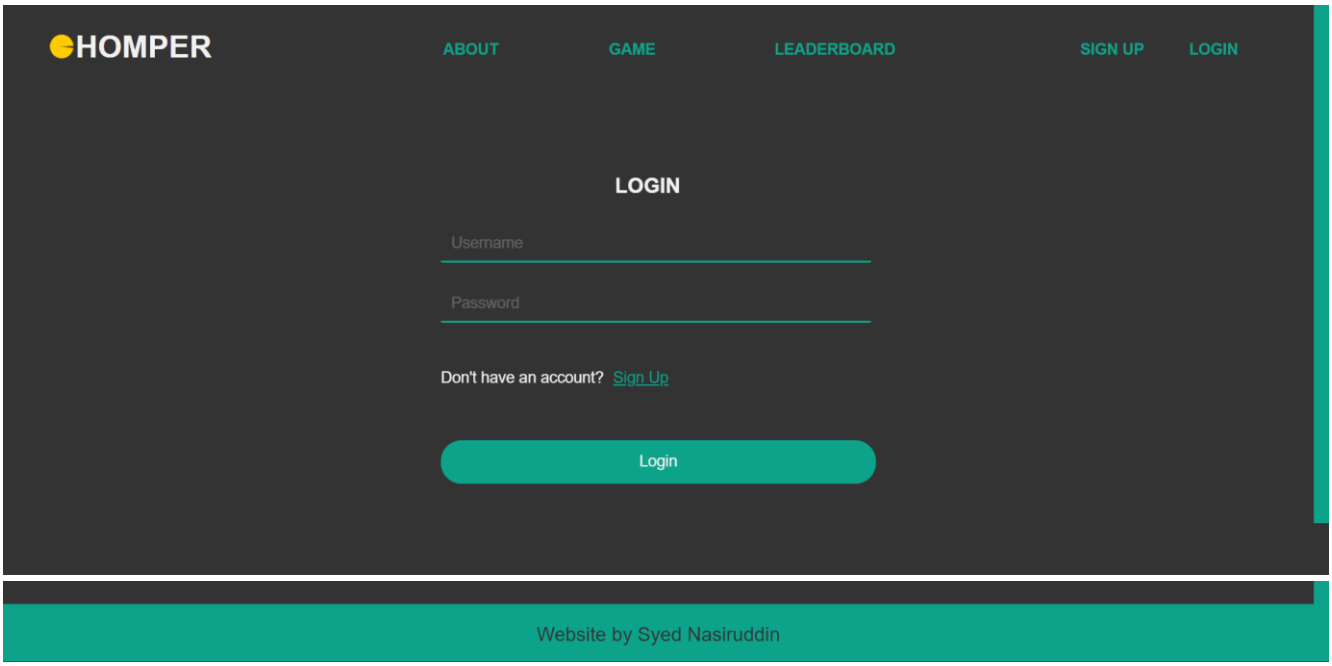
If there is an error.



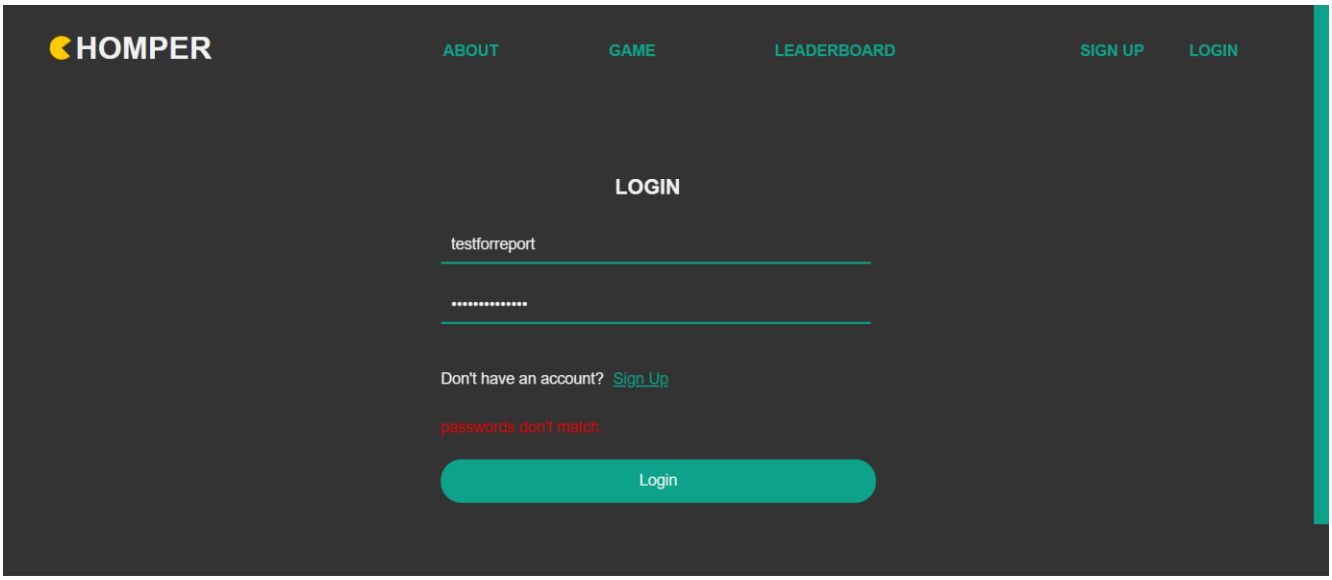
If no errors and signed up successfully.



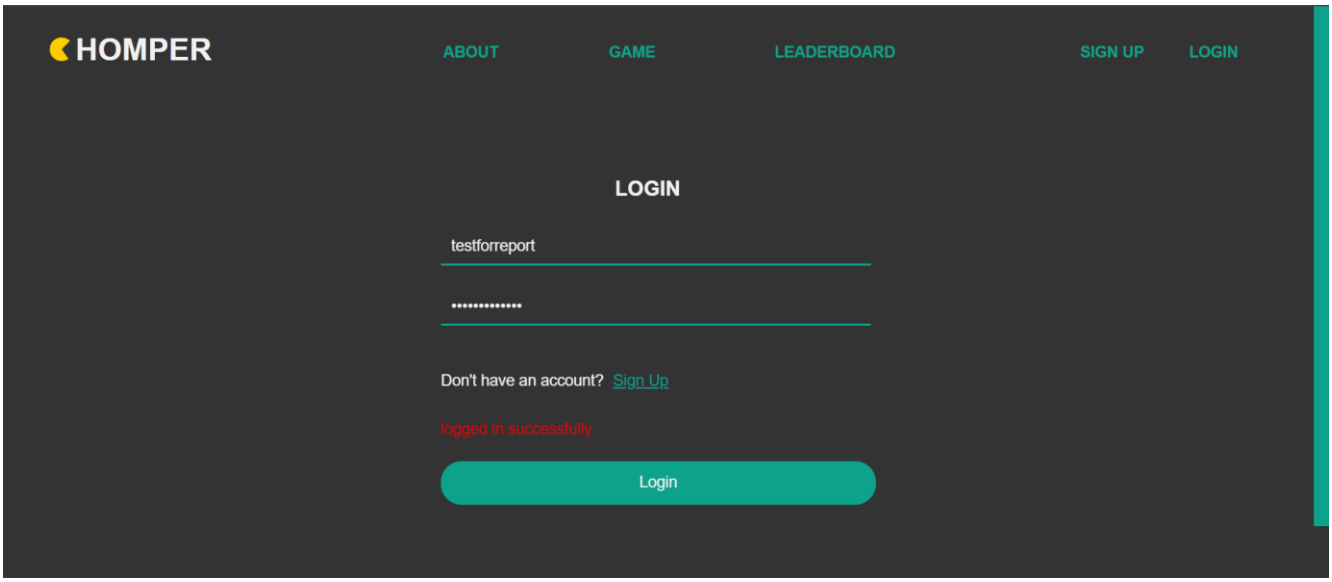
“Login” Page



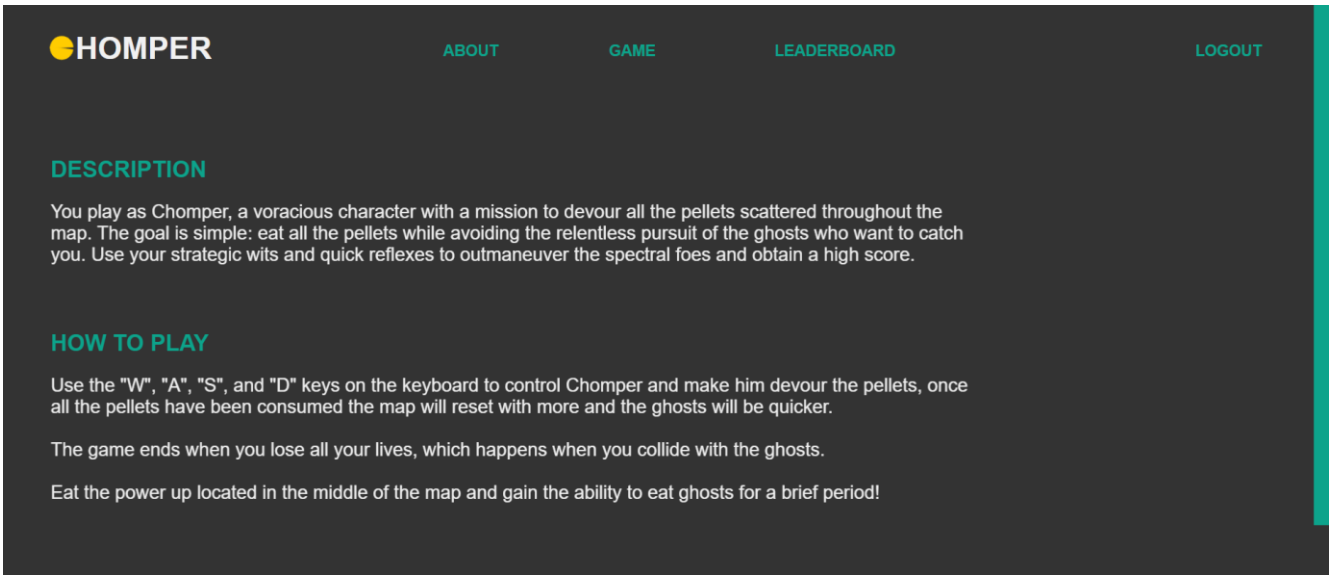
If there is an error.



If all requirements met and logged in successfully.

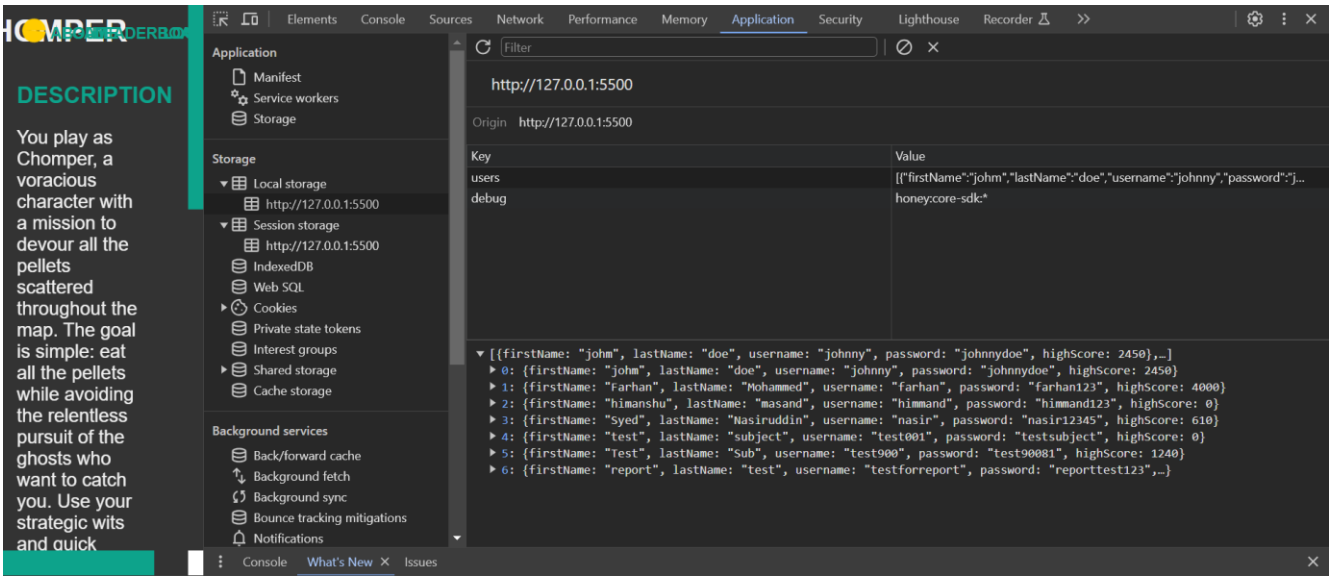


After login, the user is redirected to the “about” page and logout button appears in the top right.



Local and Session Storage

The local storage where the users are stored.



The session storage where the “check” object is stored.