Full Stack Coding Challenge - Esteban Rueda Martínez

Introduction:

This project was developed as part of a Full Stack Coding Challenge. The goal was to implement both frontend and backend components of a memory card game, since I have experience with web3 apps and Blockchain I went ahead and coded the web3 option too.

🧩 **Task Breakdown by Role (Ordered by Importance):**

Frontend Developer:

- Style the login page to be visually appealing and responsive.
- Create a modal dialog for level selection (Easy, Medium, Hard).

Backend Developer:

- Implement a simple API endpoint to save game results.
- Create a route to fetch the game result history (no need for complex authentication).

Full Stack Developer:

- Complete both the Frontend and Backend tasks.
- Integrate the API to display the game result history on a new page.

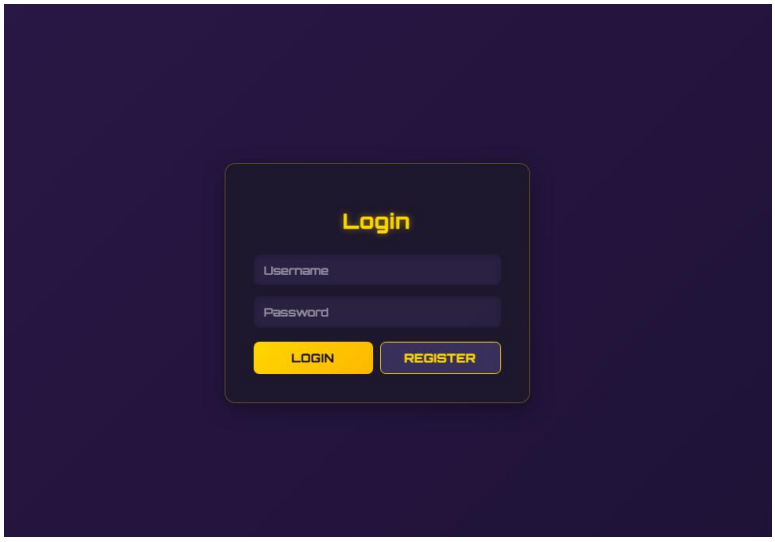
Web3 Developer:

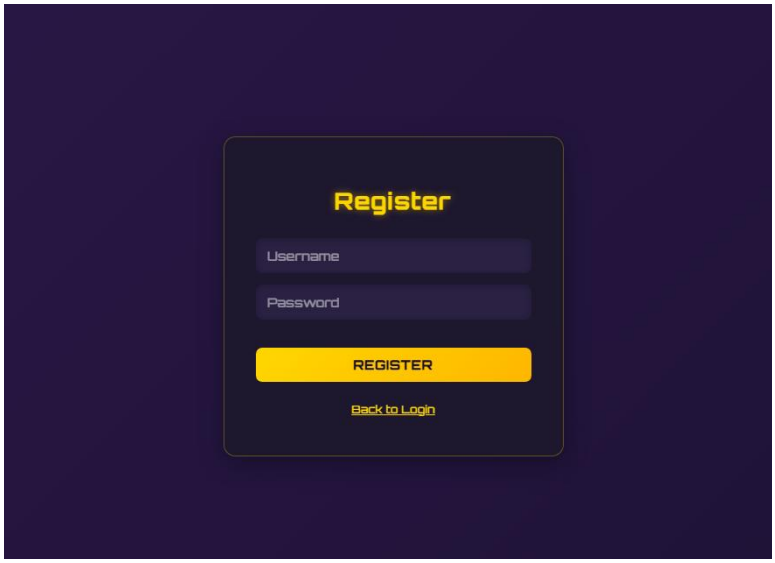
- Implement MetaMask wallet connection.
- Show a message displaying the connected wallet address.

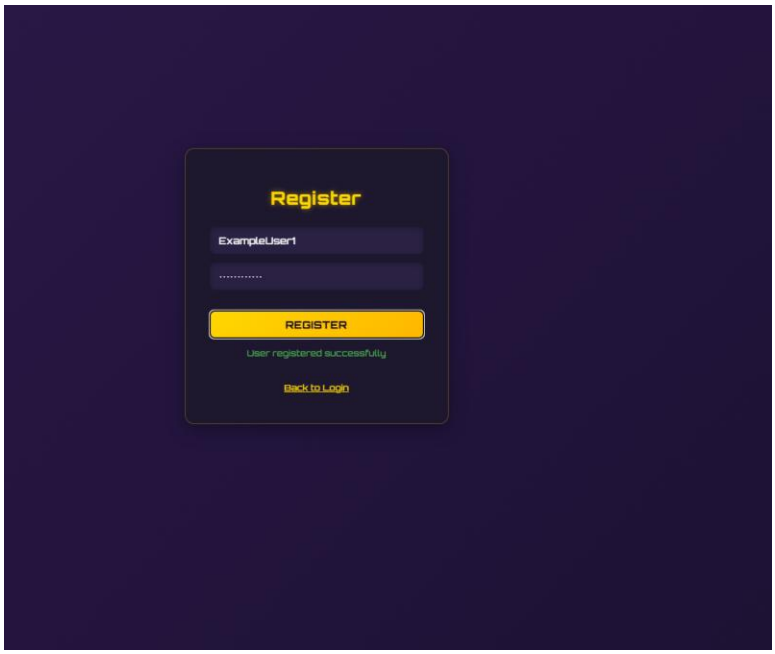
Frontend Developer

STYLE THE LOGIN PAGE TO BE VISUALLY APPEALING AND RESPONSIVE

The login and registration pages were designed to be both visually appealing and responsive following the same style as the rest of the game pages.


Login
Username
Password
LOGIN
REGISTER


Register
Username
Password
REGISTER
Back to Login



CREATE A MODAL DIALOG FOR LEVEL SELECTION (EASY, MEDIUM, HARD)

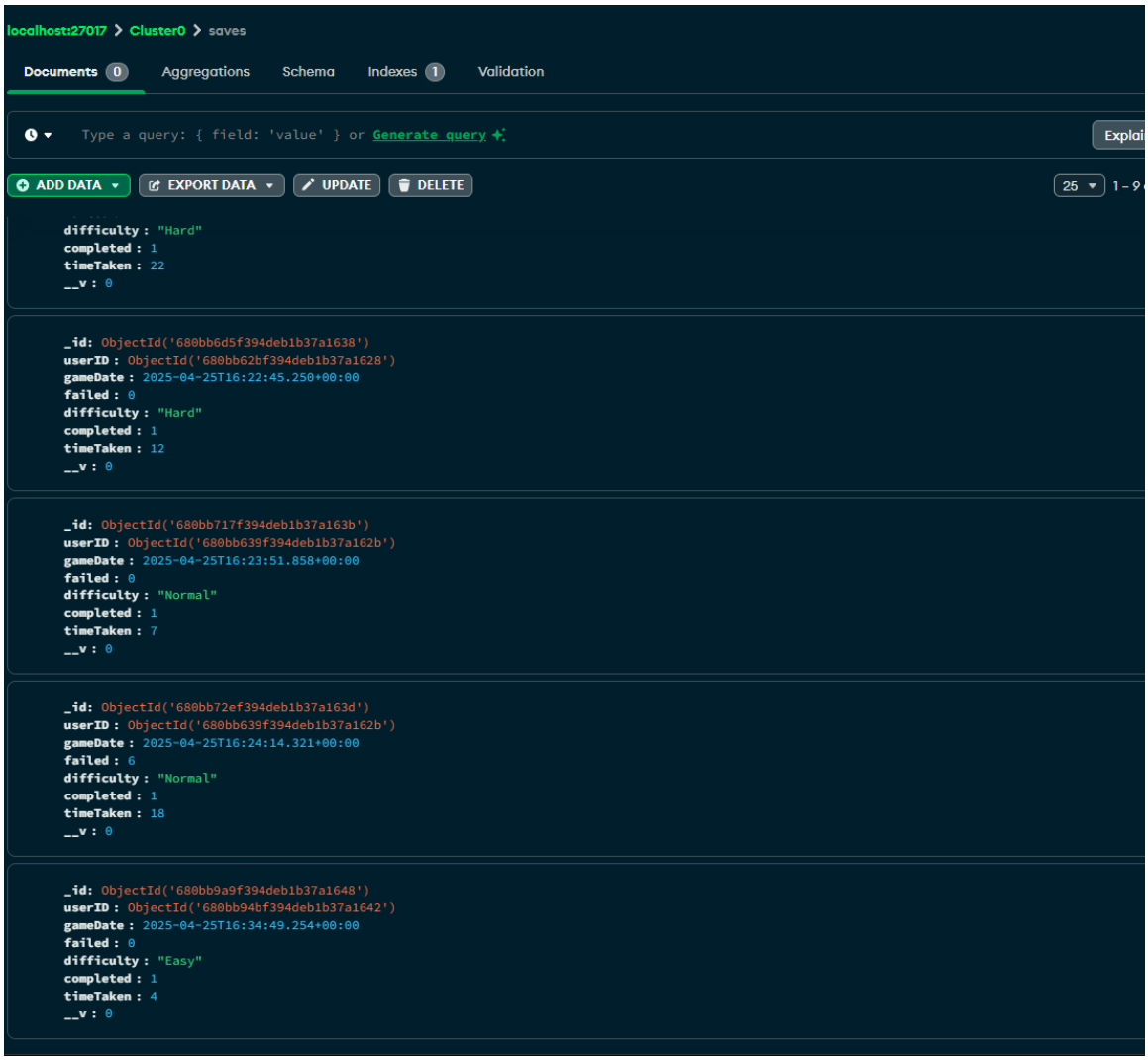
This part was already implemented in the starter project, so I didn't have to modify or recreate it. However, I made sure it worked correctly.

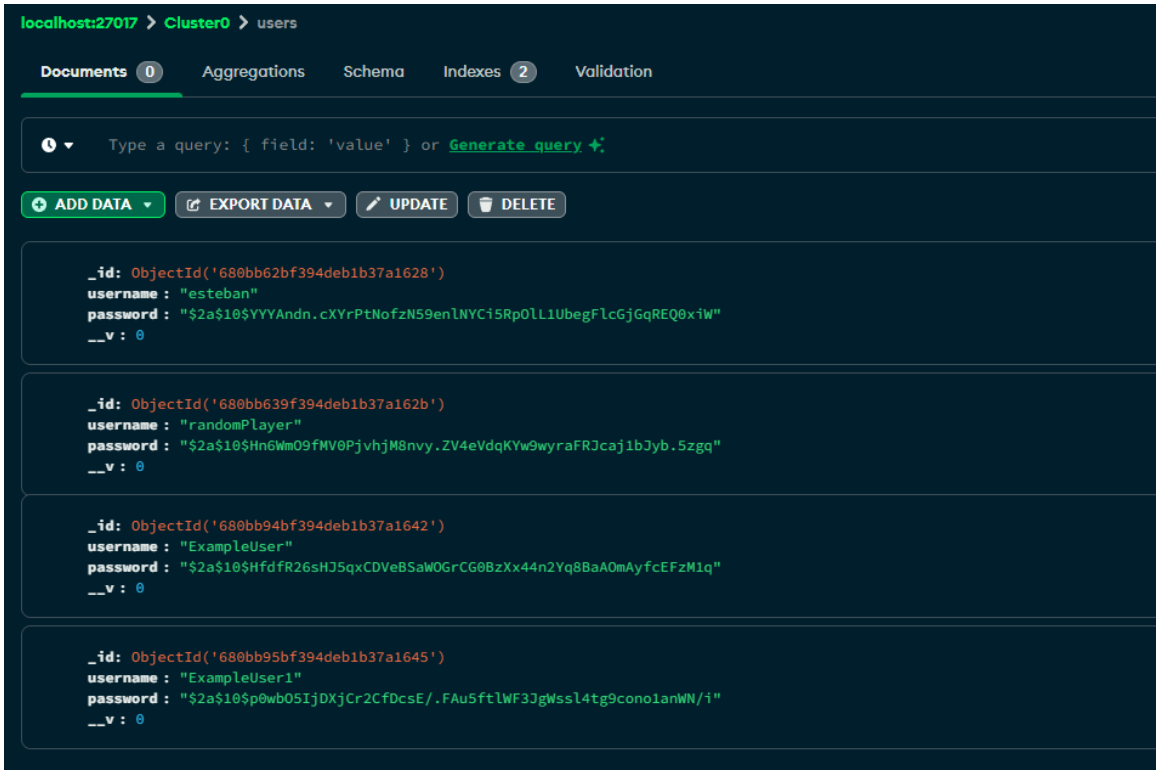
Backend Developer

IMPLEMENT A SIMPLE API ENDPOINT TO SAVE GAME RESULTS

This part was already implemented in the starter project, so I didn't have to modify or recreate it. However, I made sure it also worked correctly.

```
Received data to save: {
  userID: '680bb94bf394deb1b37a1642',
  gameDate: '2025-04-25T16:34:49.254Z',
  failed: 0,
  difficulty: 'Easy',
  completed: 1,
  timeTaken: 4
}
```


localhost:27017 > Cluster0 > saves
Documents 0
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
difficulty : "Hard"
completed : 1
timeTaken : 22
__v : 0
_id: ObjectId('680bb6d5f394deb1b37a1638')
userID : ObjectId('680bb62bf394deb1b37a1628')
gameDate : 2025-04-25T16:22:45.250+00:00
failed : 0
difficulty : "Hard"
completed : 1
timeTaken : 12
__v : 0
_id: ObjectId('680bb717f394deb1b37a163b')
userID : ObjectId('680bb639f394deb1b37a162b')
gameDate : 2025-04-25T16:23:51.858+00:00
failed : 0
difficulty : "Normal"
completed : 1
timeTaken : 7
__v : 0
_id: ObjectId('680bb72ef394deb1b37a163d')
userID : ObjectId('680bb639f394deb1b37a162b')
gameDate : 2025-04-25T16:24:14.321+00:00
failed : 6
difficulty : "Normal"
completed : 1
timeTaken : 18
__v : 0
_id: ObjectId('680bb9a9f394deb1b37a1648')
userID : ObjectId('680bb94bf394deb1b37a1642')
gameDate : 2025-04-25T16:34:49.254+00:00
failed : 0
difficulty : "Easy"
completed : 1
timeTaken : 4
__v : 0



CREATE A ROUTE TO FETCH THE GAME RESULT HISTORY (NO NEED FOR COMPLEX AUTHENTICATION)

I implemented a GET endpoint to retrieve the game history. This allows the frontend to fetch all previous results and display them on a dedicated "Game History" page.

The game history page updates dynamically based on the data returned from the backend, providing a full-stack demonstration of state synchronization between client and server.

To show this functionality, I added a button in the main interface labeled "Game History" that redirects users to the new page where all results are displayed in a clean format.

Full Stack Developer

COMPLETE BOTH THE FRONTEND AND BACKEND TASKS.

Done and explained in the previous sections.

INTEGRATE THE API TO DISPLAY THE GAME RESULT HISTORY ON A NEW PAGE.

As described in the previous Backend section, I implemented a new Game History page that displays past game results by integrating the newly created API.

Game History

All Games | Completed | Failed

25 abr 2025, 18:37 — FAILED
Difficulty: Level Normal
Time: 0:08

25 abr 2025, 18:37 — COMPLETED
Difficulty: Level Normal
Time: 0:06

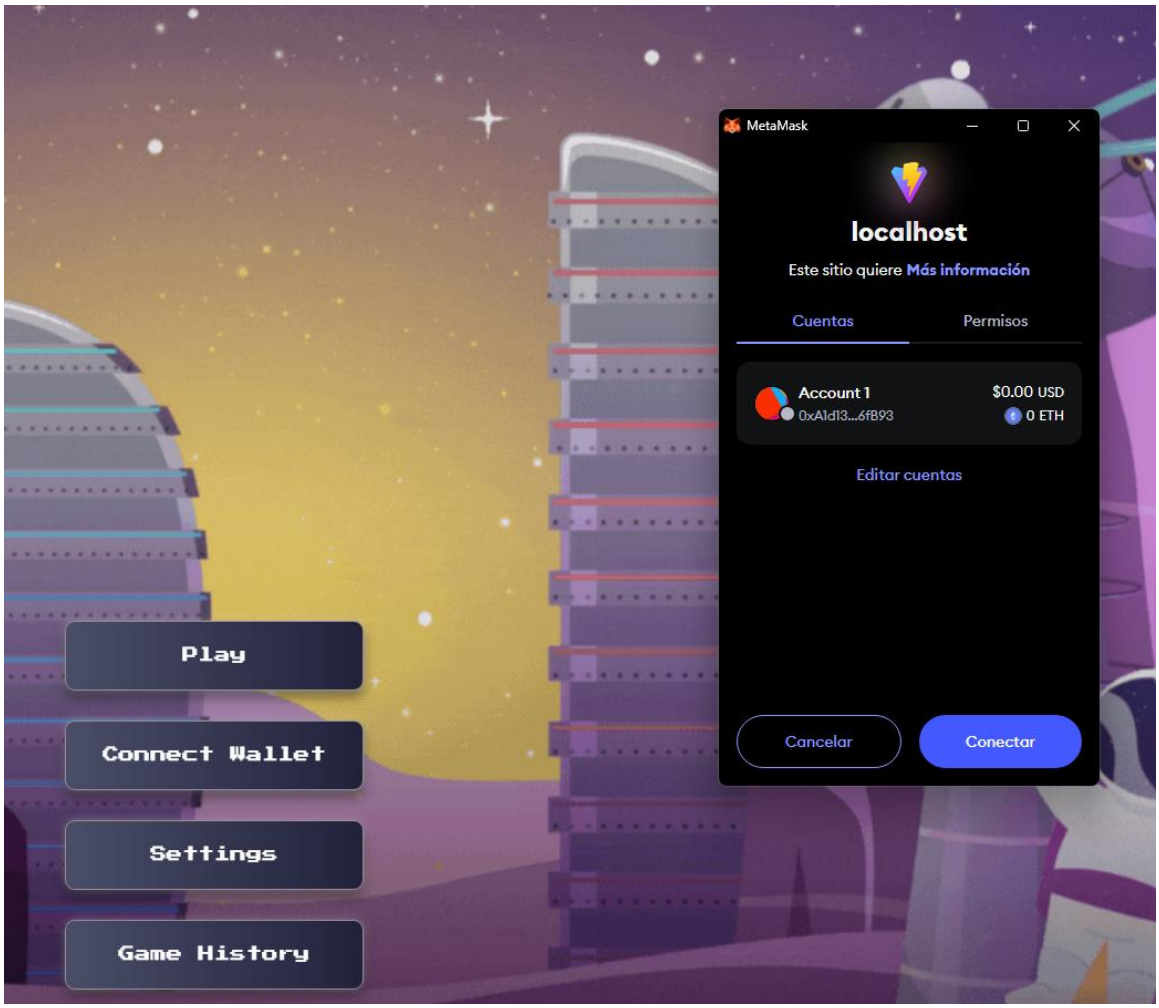
25 abr 2025, 18:34 — COMPLETED
Difficulty: Level Easy
Time: 0:04

BACK TO GAME

Web3 Developer

Implement MetaMask wallet Connection

I implemented MetaMask wallet integration and added a dedicated button to initiate the connection, as shown in the image below.



Show a message displaying the connected wallet address

Whenever a user connects their MetaMask wallet, their wallet address is displayed in the top-left corner of the main screen.


Connected wallet:
0xa1d13602091debb49
a5d5b17467f1e181c936
fb93
WonderCards
Play
Connect Wallet
Settings
Game History