

# Edward Savin

contact@edw.dev | +40 720-056-079

[portfolio](#) | [github](#) | [linkedin](#)

## SKILLS

**Programming Languages** TypeScript | JavaScript | Go | Python  
**Technologies** Next.js | React | Node.js | CSS | Prisma | MySQL | Clerk

## PROJECTS

### Moodvie

<https://moodvie.edwardcs.com/>

*Full Stack Developer*

- Built an open-source web application called Moodvie, which recommends movies based on the user's mood and music taste. The app retrieves users' recently listened to songs from Spotify and generates movie suggestions with the help of AI using OpenAI's GPT-3.5 Turbo.
- Frontend and backend development using the T3 stack, which includes Next.js (React.js), tRPC, Tailwind CSS, TypeScript, and Prisma. The frontend is built with Next.js, Tailwind CSS, and shadcn-ui, while the backend uses tRPC to provide typesafe APIs for the frontend. The app also uses Clerk for authentication and the APIs from Spotify, OpenAI (gpt-3.5-turbo), and TMDB.
- Implemented rate limiting using Upstash Redis to protect the application from potential abuse and ensure fair usage. This helps maintain the performance and availability of the app for all users.
- Integrated Clerk for secure user authentication and convenient identity management. Clerk provides user authentication and management functionality, allowing users to sign in with their Spotify account and receive personalized movie recommendations based on their mood and music taste.

### Battleship 44

<https://edwardsavin.github.io/battleship/>

*Full Stack Developer*

- Developed a Battleship game where the player competes against a computer AI. The AI's attack intelligence is based on the hunt and parity algorithm, making the game more challenging and engaging for the player.
- The game is built using only TypeScript without any framework + CSS for the frontend and Node.js for the backend. The frontend handles user interactions and game state updates, while the backend manages the AI logic and game state persistence.
- Implemented the hunt and parity algorithm for the computer AI, which improves the AI's attack strategy by taking into account the parity of the board and targeting adjacent cells after hitting a ship. This makes the computer AI a challenging and intelligent opponent for the player.
- Applied test-driven development using Vitest to ensure the quality and reliability of the Battleship game's code.
- Designed a user interface that allows players to easily interact with the game board, place their ships, and plan their attacks. The game provides visual feedback on hits, misses, and sunk ships to enhance the player's experience.

### Weather 44

<https://edwardsavin.github.io/weather-app/>

*Full Stack Developer*

- Developed a Weather App that allows users to search for weather information by city, state, or country. The application retrieves real-time weather data from OpenWeatherMap's Weather API and displays it in an intuitive user interface.
- The project was built using TypeScript, HTML, and CSS, with a focus on modular and maintainable code. The TypeScript code is organized into modules that handle different functionalities, such as fetching weather data and rendering content on the page.
- Implemented error handling and user input validation, ensuring that the application can handle invalid or incomplete search queries, as well as API errors. This provides a seamless user experience and ensures that the application remains functional in various scenarios.
- Designed a responsive user-friendly interface that displays weather information, including temperature, humidity, and weather conditions, in an easily understandable format. The app also features a search bar for users to quickly find the desired location and obtain weather information for it.

## EDUCATION

### Harvard University CS50P

[View Certificate](#)

*Harvard University*

**Oct 2019 - Jun 2022**

### Sociology

*Babes-Bolyai University*

*Cluj-Napoca, RO*