TRICKY TAC TOE

Overview

An unbeatable Tic Tac Toe game powered by AI is built for Microsoft's Engage 2020. Minimax algorithm is implemented in javascript to determine the best move. Users can create accounts and log in to play the game. Highscores are displayed on the leaderboard. The frontend uses hbs view engine. The backend is developed using NodeJS, ExpressJS and MongoDB Atlas.

Table of Contents

Tic Tac Toe

The general rules of tic tac toe are as follows :

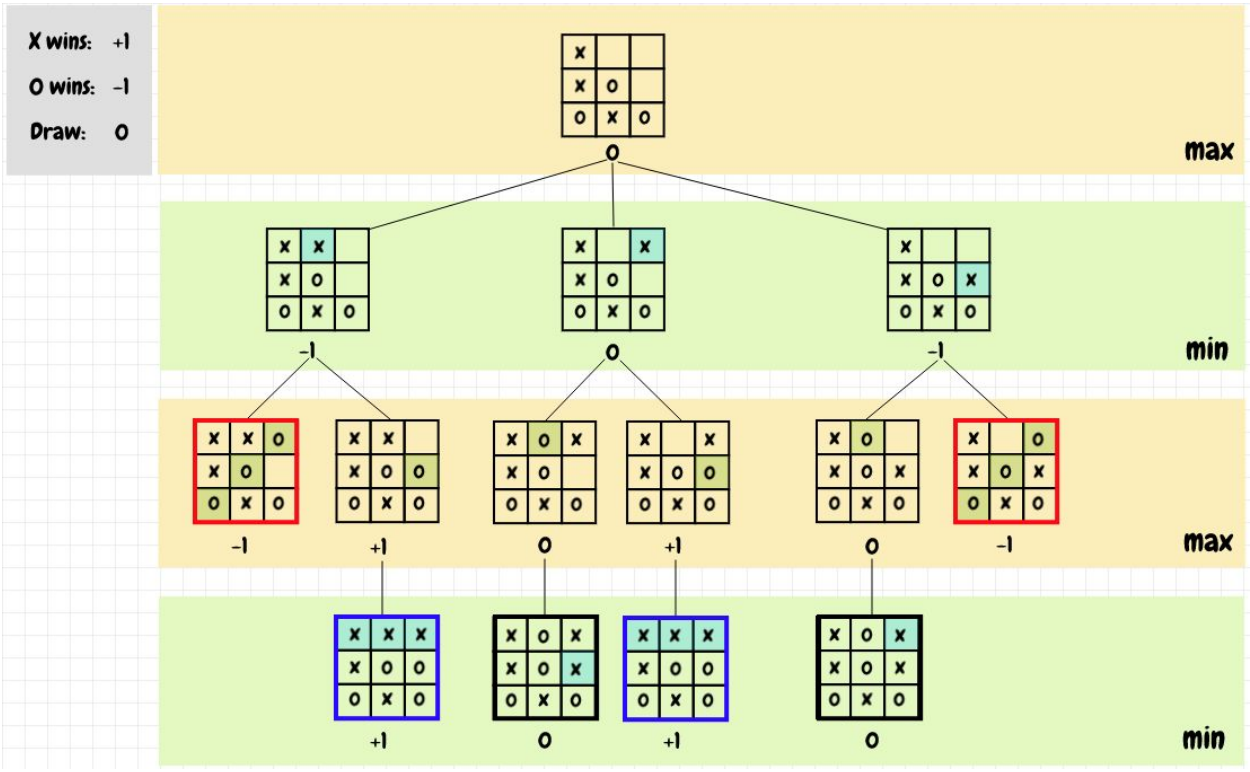
1. The game is played on a 3 x 3 grid.
2. Two players, 'X' and 'O' take turns marking the empty squares in the grid.
3. The first player to succeed in placing three of their marks in a horizontal, vertical or diagonal row is the winner.
4. The game ends in a tie when all the nine squares are filled but no player could get three marks in a row.

Here, the user (first player) competes with AI(second player).

Minimax Algorithm

Minimax is a recursive algorithm which is used to choose an optimal move for a player assuming that the opponent is also playing optimally. The goal of the algorithm is to

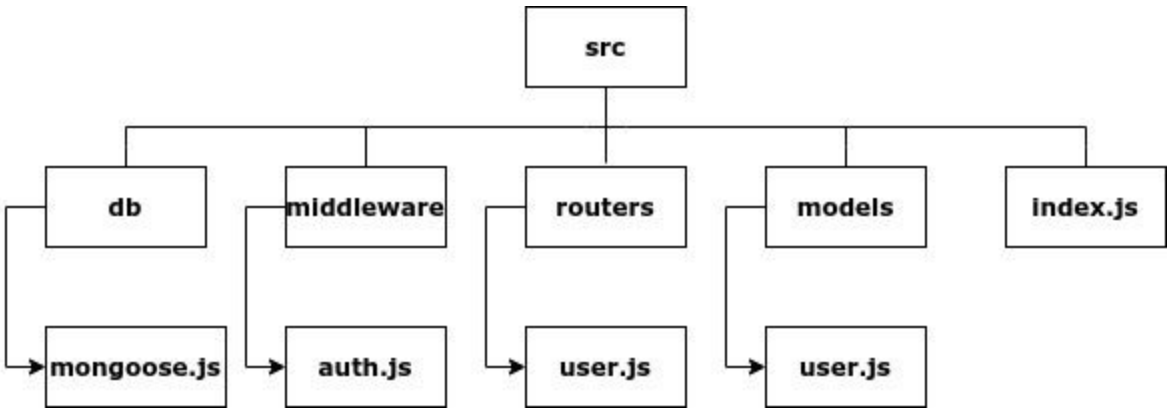
minimise the maximum loss. After the player makes a move, the Minimax algorithm is used to calculate all possible moves for the computer and determine the best move. We explore the game from a given starting position until we reach all possible end-of-game states. We can represent this as a tree, with each level of the tree showing the possible board positions for a given player's turn. When we reach an end-of-game state, there is no choice to be made, so the value is the game result, that is 1 if X won, -1 if O won, and 0 if it is a draw. We can see that in a perfect match, only a draw is possible, but if the players play at random, the computer always wins.



Backend

The backend is developed using Node.js and Express.js.

File structure



REST API

- **User registration**
    - URL : /register
    - Method : POST
    - URL params : None
    - Data params :
        - username = [string]
        - password = [string]
        - email = [email]
    - Success response :
        - Code : 200
    - Error response
        - Code : 400

- **Registration page**
    - URL : /register
    - Method : GET
    - URL params : None

    - Data params : None
    - Success response :
        - Code : 200
    - Error response:
        - Code: 400

- **Login page**

    - URL : {{url}}/
    - Method : GET
    - URL params : None
    - Data params : None
    - Success response
        - Code : 200
    - Error response

- **User login**

    - URL : {{url}}/
    - Method : POST
    - URL params : None
    - Data params :
        - username = [string]
        - Password = [string]
    - Success response
        - Code : 200
    - Error response
        - Code : 400

- **Logout**

    - URL : /logout
    - Method : POST
    - URL params : None
    - Data params :
        - Token = [string]
    - Success response
        - Code : 200

    - Error response
        - Code : 500

- **Leaderboard**

    - URL : /leaderboard
    - Method : GET
    - URL params : None
    - Data params : None
    - Success response
        - Code : 200
    - Error response
        - Code : 500

- **Update highscore**

    - URL : /updatehighscore
    - Method : PATCH
    - URL params : None
    - Data params :
        - Score = [integer]
    - Success response
        - Code : 200
    - Error response
        - Code : 400

- **Dashboard**

    - URL : /dashboard
    - Method : GET
    - URL params : None
    - Data params : None
    - Success response
        - Code : 200
    - Error response
        - Code : 500

- **Game page**
    - URL : /game
    - Method : GET
    - URL params : None
    - Data params : None
    - Success response
        - Code : 200
    - Error response
        - Code : 500

Node.js

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside a web browser.

Express.js

Express.js is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.

Features

Hashing

Passwords are hashed using bcrypt.js. Bcrypt is a password hashing function based on Blowfish cipher.

Authentication

Cookies are used for authentication. The authentication system is implemented using Express.js's middleware functionality.

Database

MongoDB Atlas

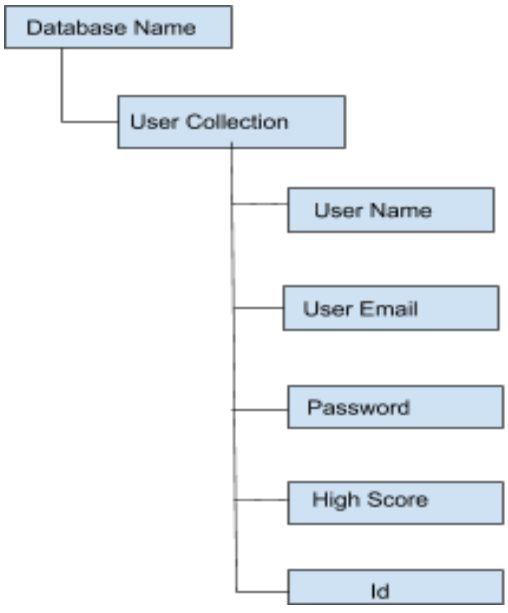
MongoDB Atlas is a cloud database as a service created by the experts who design and engineer MongoDB. Atlas provides all of the features of MongoDB while automating database administration tasks such as database configuration, infrastructure provisioning, patches, scaling events, backups, and more.

Database Storage

In MongoDB, data is stored as documents. These documents are stored in MongoDB in JSON (JavaScript Object Notation) format. JSON documents support embedded fields, so related data and lists of data can be stored with the document instead of an external table.

JSON is formatted as name/value pairs. In JSON documents, fieldnames and values are separated by a colon, field name and value pairs are separated by commas, and sets of fields are encapsulated in “curly braces” ({}).

Database Structure



Frontend

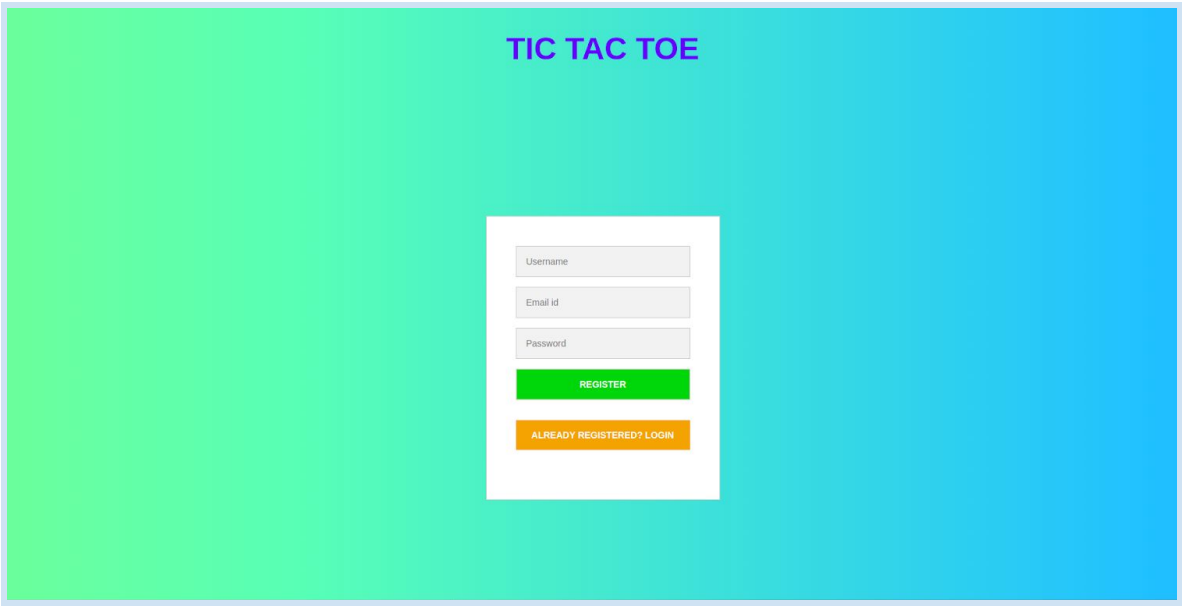
View engine

The hbs view engine is used to render templates. Handlebars provides the power necessary to build semantic templates effectively. It compiles templates into JavaScript functions. This makes the template execution faster than most other template engines.

Views

Registration page will ask for our username, password and email id and all the details will be stored in the database. The next time the user opens the website, he / she will be asked to login using the username and password. After logging in, the user will be directed to the dashboard. From the dashboard the user can navigate to the leaderboard or the game page. The leaderboard lists the players in descending order of highscores. We have used the Handlebars view engine for the frontend.

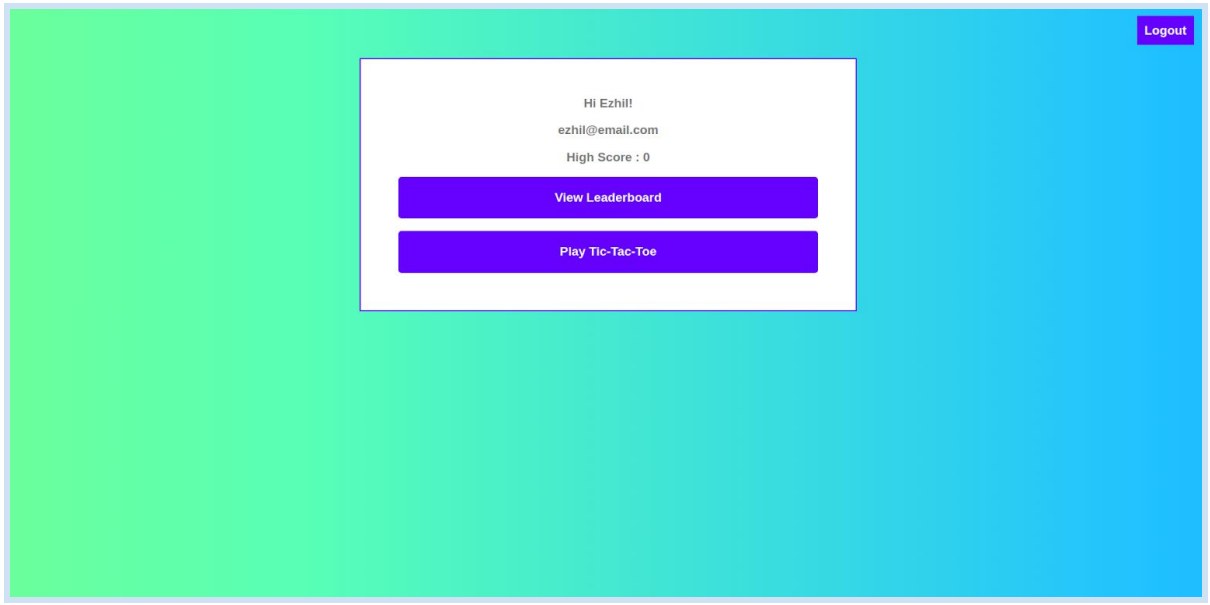
Registration Page



Login Page

Dashboard

The dashboard welcomes the player to the game. It shows the username, email and high score of the player. It has buttons to navigate to the game page and the leaderboard. It contains a button to logout from the game.

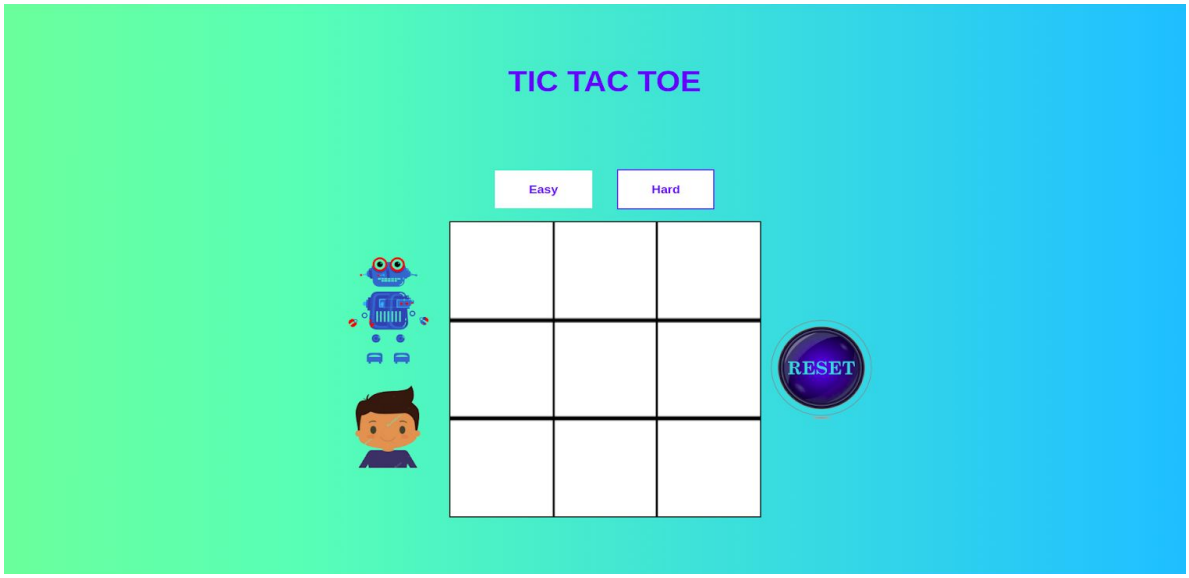


Leaderboard

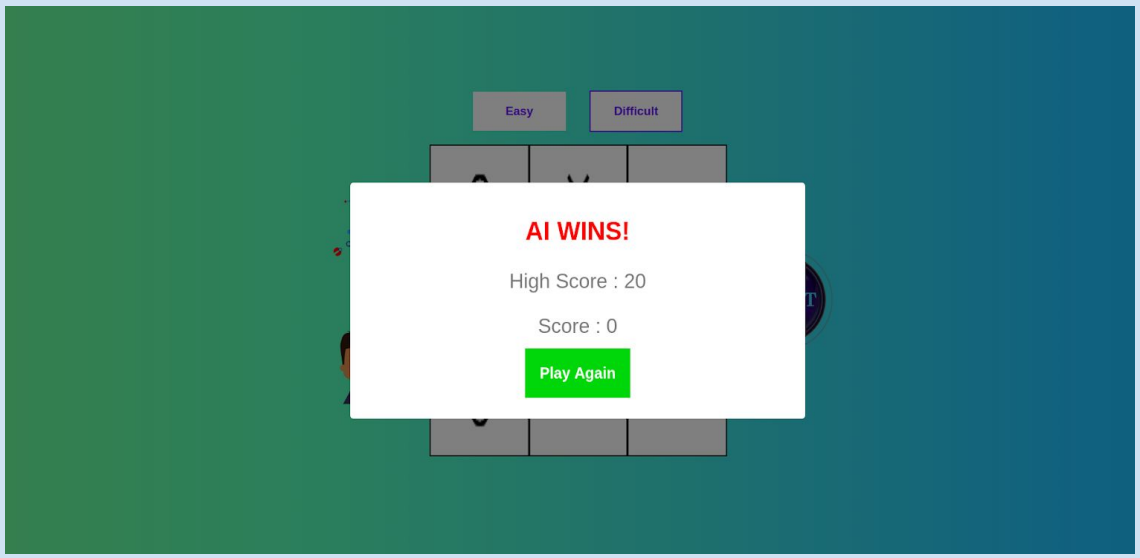
Username	Highscore
test1	30
ad	30
Ezhil	30
Fruity	30
Paavai	20
abc	5
Alice	0
Sarah	0
Debbie	0

Game Page

The game page has a canvas element containing a 3 X 3 grid. It allows the player to mark X's, while the computer marks O's alternatively. The restart button allows the user to restart the game when clicked. A pop-up occurs when the user or computer or neither of them wins the game. The pop-up contains the current score and the high score of the player. The toggle button on the game page allows the player to switch to different difficulty levels and restart the game.



Game Page Pop-up



Deployment

The tricky tac toe web application has been successfully deployed in Heroku.
Heroku URL : https://guarded-escarpment-13236.herokuapp.com/
Github URL :https://github.com/Smrithi23/Tricky_Tac_Toe