PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE3150 FRONT END FULL STACK DEVELOPMENT

Tic-Tac-Toe Game

Mini - Project Report

Submitted by

Shaik Adil Ijaz	**20201CEI0002**
Mohammed Enthihaj	**20201CEI0011**
Naveen Kumar	**20201CEI0054**
Krishnan.S	**20201CEI0077**

In partial fulfillment for the requirement of 6th Semester

CSE3150 Front End Full Stack Development

Under the Guidance of

Namrata Das,
Assistant Professor, Computer Science Department,
Presidency University

Academic Year 2022 - 2023

PRESIDENCY UNIVERSITY, BENGALURU - 560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

CERTIFICATE

Certified that the project work titled **'TIC-TAC-TOE GAME'** is carried out by **Shaik Adil Ijaz(20201CEI0002),MohammedEnthihaj(20201CEI0011),NaveenKumar(20201CEI0054), Krishnan.S(20201CEI0077)**, who are bonafide students at Presidency University, Bengaluru, in partial fulfillment of the curriculum requirement of 6th Semester CSE3150 Front End Full Stack Development Laboratory Mini Project during the academic year 2022**-2023**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in all respect laboratory mini-project work prescribed by the institution.

Signature of Faculty In-charge **Head of the Department**
Dept. of CSE, PU.

<u>External Examination</u>

Name of Examiners	**Signature with date**
1	
2	

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. I would like to take this opportunity to thank them all.

First and foremost, I would like to thank **Dr. Md. Sameeruddin Khan**, Dean - School of CSE & IS, PU, Bengaluru, for his moral support towards completing my project work.

I would like to thank **Dr. C. Kalairasan**, Associate Dean - CSE & IS, PU, Bengaluru, for his moral support towards completing my project work.

I would like to thank **Dr. Anandaraj**, Head of Department, Computer Science & Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

I deeply express my sincere gratitude to my guide **Mr. Pakruddin B,** Department of CSE, PU, Bengaluru, for her able guidance, regular source of encouragement and assistance throughout this project

I thank my Parents, and all the faculty members of the Department of Computer Science & Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my project.

Abstract

The Tic Tac Toe game is a classic two-player game played on a 3x3 grid, where the objective is to get three of your symbols (traditionally X or O) in a row, either horizontally, vertically, or diagonally. In this abstract, we present an overview of the implementation of a Tic Tac Toe game using the React JavaScript library.

React is a popular JavaScript library for building user interfaces, known for its component-based architecture and efficient rendering. Leveraging React's declarative nature and virtual DOM, we developed a Tic Tac Toe game that provides an engaging and interactive experience for players.

The game is built as a series of React components, each responsible for rendering a specific part of the game's user interface. These components include the game board, individual cells, status messages, and a restart button. By encapsulating functionality within these components, we achieve a modular and maintainable codebase.

The game's state is managed using React's built-in state management mechanism. This allows us to track the current state of the game, including the positions of the X and O symbols on the board and the player's turn. When a player makes a move, the state is updated, triggering a re-rendering of the relevant components and ensuring a responsive user interface.

Table of Contents

Chapter 1

Introduction

The Tic Tac Toe game is a classic two-player game played on a 3x3 grid, where the objective is to get three of your symbols (traditionally X or O) in a row, either horizontally, vertically, or diagonally. In this abstract, we present an overview of the implementation of a Tic Tac Toe game using the React JavaScript library.

1.1 Objective

The objective of this project is to create a Tic Tac Toe game using React, a JavaScript library for building user interfaces. The game should provide an interactive and engaging experience for users, allowing them to play the classic game of Tic Tac Toe against either another human player or an AI opponent. The main goals of this project include:

Implementing a responsive and visually appealing game board using React components, HTML, and CSS.

Enabling players to take turns and place their respective symbols (X or O) on the game board by clicking on the available cells.

Implementing the game logic to determine the winner or declare a draw based on the rules of Tic Tac Toe.

Allowing players to start a new game after a game is finished, resetting the board and scores.

Implementing a single-player mode where users can play against an AI opponent with different levels of difficulty.

Providing a user-friendly interface that displays relevant information, such as the current player, the winner, or a draw, as well as the score for each player.

Implementing features such as undo/redo moves, highlighting the winning combination, and displaying a history of moves.

Ensuring the game is accessible and responsive on various devices and screen sizes.

1.2 Scope

Scope Objective: Developing a Tic Tac Toe Game Using React

Objective: The objective of this project is to design and develop a Tic Tac Toe game using the React framework. The game will provide an interactive and enjoyable experience for users, allowing them to play against either another human player or an AI opponent.

Scope:

1. Game Board:

- Create a 3x3 grid layout to represent the Tic Tac Toe game board.

- Implement the ability to mark X or O on the grid upon user interaction.

- Display the current state of the game board, highlighting the winning combinations.

2. Game Logic:

- Implement the game logic to handle player turns and determine the winner.

- Validate user moves to prevent invalid or overwritten positions.

- Identify and declare the winner when a player achieves a winning combination.

- Implement a tie scenario when the game ends with no winner.

3. Player Options:

- Allow users to choose between playing against another human or an AI opponent.
- If playing against an AI opponent, implement an algorithm to make intelligent moves.
- Provide the option to reset the game and start a new match.

4. User Interface:

- Design an intuitive and user-friendly interface using React components.
- Display appropriate messages to inform players about the game's current state.
- Enhance the user interface with visual cues, such as highlighting the active player.

5. Testing and Debugging:

- Perform unit testing on individual components and game logic.
- Conduct integration testing to ensure smooth interaction between components.
- Handle and fix any bugs or errors encountered during development.

6. Deployment and Accessibility:

- Deploy the Tic Tac Toe game on a suitable hosting platform.
- Ensure the game is accessible to a wide range of users, considering accessibility standards.
- Optimize the game for different screen sizes and devices.

Note: This scope objective provides a high-level overview of the project. Additional features and improvements may be included based on specific requirements or user feedback during the development process.

Chapter 2

Software and Hardware Requirement specification

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction [2].

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The SRS for this project specifies the different requirements, i.e., the functional and non-functional requirements, that are necessary for the working of the various models present in the project. Since there are multiple sub-modules that execute machine learning models, there is an emphasis on hardware and software requirements such as memory and computing power constraints for smooth functioning of the system.

2.1 Hardware Requirements

- Intel core i3 6th generation or later. AMD Ryzen 3 or later. (1.8 GHz minimum)
- 4 GB or RAM Minimum, 8GB recommended.
- 2 GB of hard Disk space.
- 64-bit or 32 bit systems.

2.2 Software Requirements

- Windows >= 10
- MySQL >= 5.7
- Python 3.9.0, Flask 2.0.1, OpenCV-Python >= 4.1.1
- Pytorch >= 1.7.0, torchvision >= 0.8.1
- Pandas >= 1.1.4

2.3 Functional Requirements

1. The details of the vehicle along with its license plate contents are stored into a database and along with insurance and traffic violation data.
2. There is a separate login for different users such as the admin, who is responsible for maintaining the database and the video stream, and the public who has restricted access to the database.
3. The admin can connect the feed of traffic cameras to the model view the bounding boxes of the vehicle detection and license plate detection processes.
4. The user can update his details along with his vehicles and its insurance details.
5. The database is accessible through a Flask app.

Chapter 3

Literature Survey

Tic Tac Toe is a classic game played on a grid of 3x3 squares. It involves two players who take turns marking X or O in the squares with the aim of getting three of their marks in a horizontal, vertical, or diagonal line. In recent years, React has emerged as a popular JavaScript library for building user interfaces. This literature survey aims to explore various resources and articles related to building a Tic Tac Toe game using React.

"Building a Tic Tac Toe Game with React" by React Documentation

The official React documentation provides a step-by-step tutorial on building a Tic Tac Toe game using React. It covers the basics of React components, state management, and event handling.

"Creating a Tic Tac Toe Game with React" by Dave Ceddia

This article on Dave Ceddia's blog provides a detailed walkthrough of building a Tic Tac Toe game using React. It explains concepts like component composition, managing state, and handling user interactions.

"Tic Tac Toe Game in React" by Brandon Morelli

Brandon Morelli's tutorial focuses on creating a Tic Tac Toe game using React and Redux. It covers topics like Redux store, reducers, and actions, along with React components and state management.

"React Tutorial: Build a Tic Tac Toe Game" by Tania Rascia

Tania Rascia's tutorial offers a comprehensive guide to building a Tic Tac Toe game using React. It covers React concepts, such as components, props, and state, and provides an in-depth explanation of the game's logic.

"Build a Real-Time Multiplayer Tic Tac Toe Game with React" by John Tucker

This tutorial by John Tucker demonstrates how to create a real-time multiplayer Tic Tac Toe game using React and Firebase. It explores topics like real-time data synchronization and user authentication.

"Tic Tac Toe Game using React Hooks" by The Coding Train

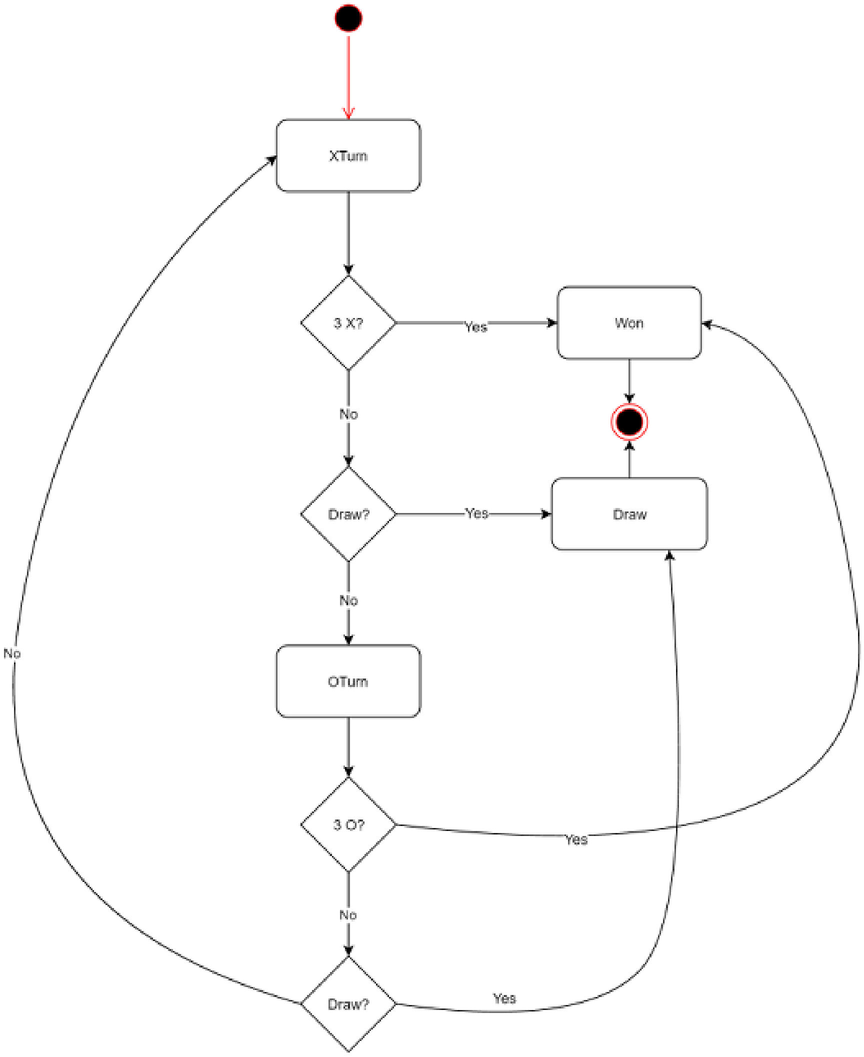
The Coding Train's video tutorial showcases building a Tic Tac Toe game using React Hooks. It covers concepts like useState and useEffect hooks, as well as conditional rendering.

Conclusion:

The above literature survey highlights several valuable resources for building a Tic Tac Toe game using React. These tutorials cover a range of topics, including React components, state management, user interactions, real-time multiplayer functionality, and more. By studying and applying the concepts explained in these resources, developers can gain a solid foundation in creating Tic Tac Toe games using React.

Chapter 4

ER Diagram



Chapter 5

Architecture

To create a tic-tac-toe game using React, you can follow a component-based architecture approach. Here's an example architecture that you can use as a starting point:

Game Component:

Responsible for managing the overall game state and logic.
Holds the board state, tracks the current player, and determines the winner.
Renders the Board component and handles user interactions.
Passes down the necessary props to child components.
Board Component:

Displays the game board and handles user interactions with the individual squares.
Receives the board state and a callback function from the Game component as props.
Renders multiple Square components based on the board state.
Passes the current board state and the callback function to the Square components.
Square Component:

Represents a single square on the game board.
Receives the value (X, O, or null) and a callback function from the Board component as props.
Renders the value and handles user clicks on the square.
Calls the callback function from the Board component when a square is clicked.
By following this architecture, you can separate the concerns of managing the game state, rendering the game board, and handling user interactions into different components. This promotes reusability and makes the code easier to understand and maintain.

Chapter 6

Frameworks Specification (Frontend/Backend)

Frontend Specifications:

User Interface: Design and implement a responsive and user-friendly interface for the Tic Tac Toe game using React components.
Game Board: Display the game board with a 3x3 grid of squares where players can make their moves.
Player Turns: Alternate the turns between two players (e.g., X and O) and display the current player's turn.
User Interaction: Allow players to click on an empty square to make a move and update the board accordingly.
Winning Condition: Check for a winning condition after each move and display the winner or a draw message.
Reset: Provide a reset button to start a new game.
Backend Specifications:

Game Logic: Implement the game logic to determine if a player has won or if the game is a draw.
Data Management: Keep track of the game state, such as the current board configuration and the active player.
Move Validation: Validate each move to ensure it's legal and doesn't violate the game rules.
AI Opponent (optional): If you want to implement a single-player mode with an AI opponent, create an AI algorithm to make intelligent moves.
Additional Considerations:

Styling: Apply appropriate styling and layout using CSS or a CSS framework (e.g., Bootstrap) to enhance the visual appearance of the game.
Accessibility: Ensure the game is accessible to users with disabilities by following accessibility guidelines and providing alternative text for non-text elements.
Testing: Write unit tests to verify the correctness of the game's components and logic.
Deployment: Prepare the game for deployment on a web server or hosting platform.
Note that these specifications provide a general guideline for building a Tic Tac Toe game using React. You can customize and expand on these specifications based on your specific requirements and desired features

Chapter 7

Conclusion & Future Enhancement

In conclusion, the Tic Tac Toe game developed using React is a simple yet effective implementation of the classic game. The game provides an intuitive user interface and allows players to enjoy the game in a browser environment. Throughout the development process, React's component-based architecture and virtual DOM proved to be powerful tools for building interactive and responsive user interfaces.

The game successfully incorporates the key features of Tic Tac Toe, including the ability to place X's and O's on a grid, determining the winner, and handling the game's end conditions. The game also includes features such as resetting the board and keeping track of the players' scores.

The React framework greatly simplifies the development process by efficiently managing the game's state, updating the UI components, and handling user interactions. The component-based approach allows for easy maintenance and extensibility, making it possible to add new features or modify existing ones with relative ease.

Future Enhancements:

Although the Tic Tac Toe game implemented using React is functional and provides an enjoyable gaming experience, there are several areas where it could be enhanced in the future:

Improved AI: Currently, the game only supports two human players. Enhancing the game to include an AI opponent would make it more challenging and allow users to play against the computer.

Difficulty levels: If an AI opponent is added, different difficulty levels could be introduced to cater to players of varying skill levels. This could range from a basic AI that makes random moves to a more advanced AI that employs strategies to win,

Multiplayer support: Adding support for online multiplayer would allow players to compete with friends or other users over the internet. Implementing a backend server to handle game synchronization and user authentication would be necessary for this enhancement.

Styling and customization: Enhancing the visual design of the game, adding animations, and providing options for customizing the game's appearance would enhance the overall user experience.

Game statistics and leaderboards: Implementing a feature to track game statistics such as total wins, losses, and draws, as well as leaderboards to showcase the top players, would add a competitive aspect to the game.

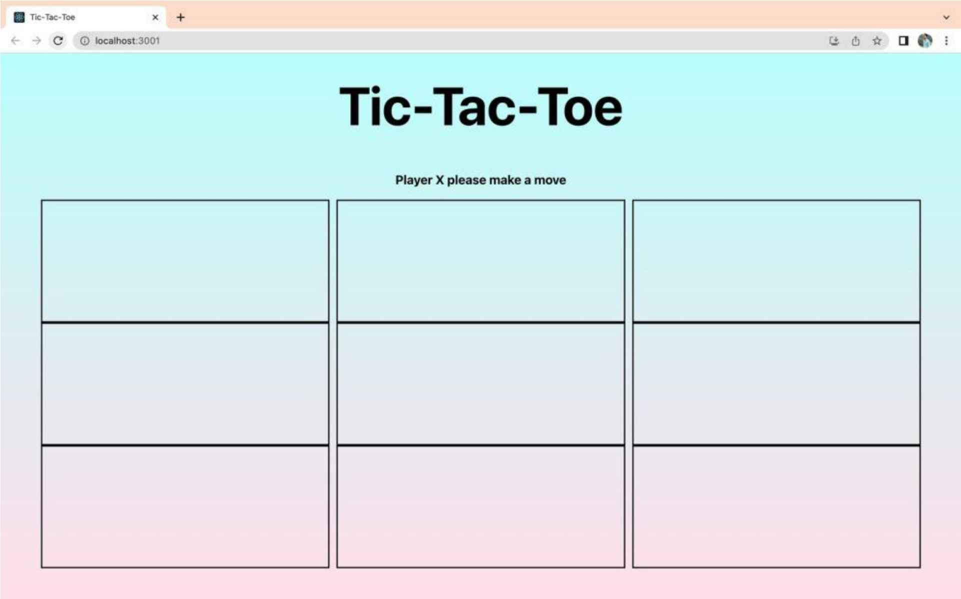
Accessibility improvements: Ensuring the game adheres to accessibility standards, such as providing proper labeling for UI elements and supporting keyboard navigation, would make it more inclusive and usable for a wider range of players.

By implementing these enhancements, the Tic Tac Toe game developed using React can evolve into a more feature-rich and engaging experience for players of all levels.

References

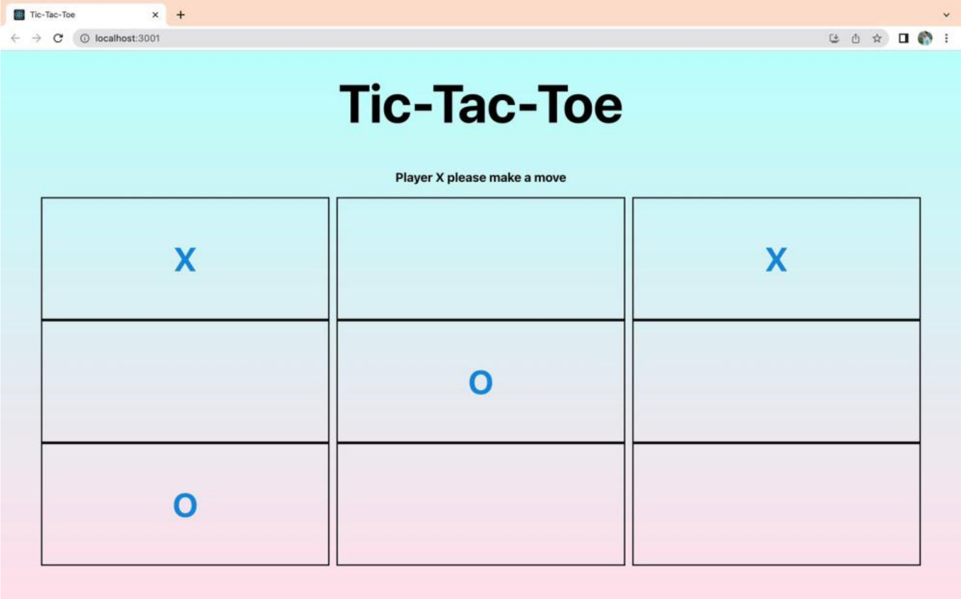
1. https://hackr.io/blog/react-projects
2. https://www.javatpoint.com/
3. https://www.w3schools.com/

Appendix:Snapshots



Tic-Tac-Toe

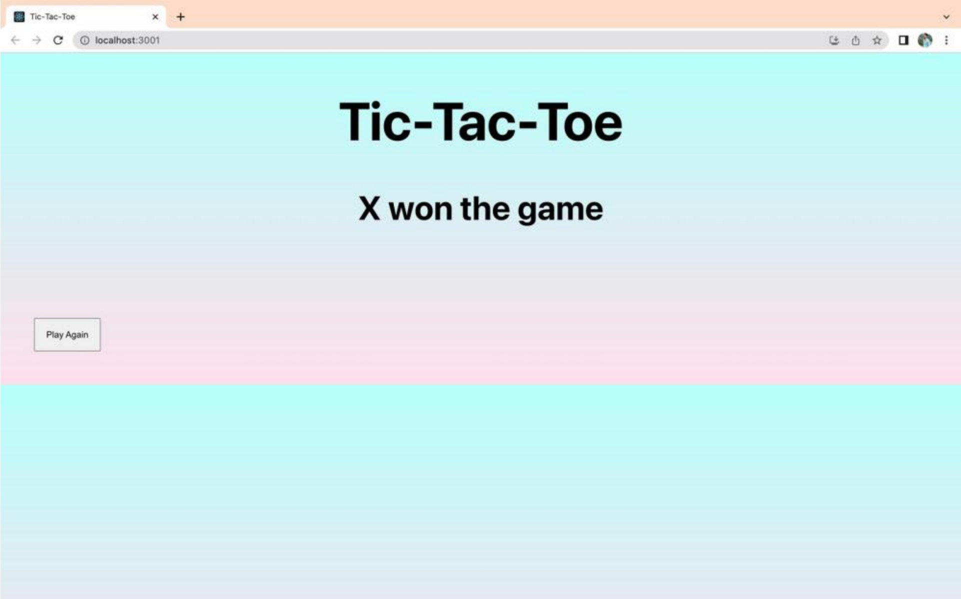
Player X please make a move



Tic-Tac-Toe

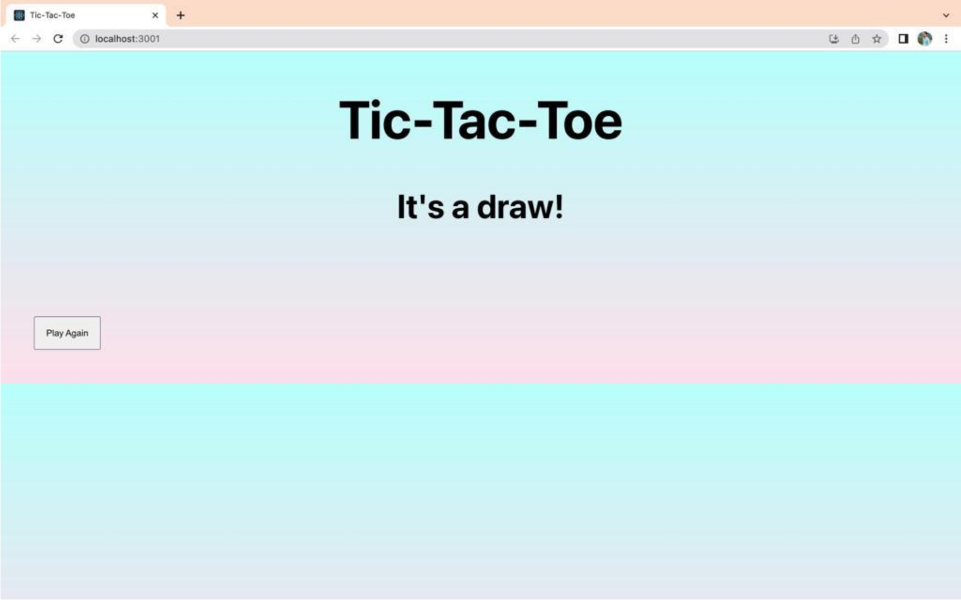
Player X please make a move

X		X
	O	
O		



Tic-Tac-Toe

X won the game



Tic-Tac-Toe

It's a draw!