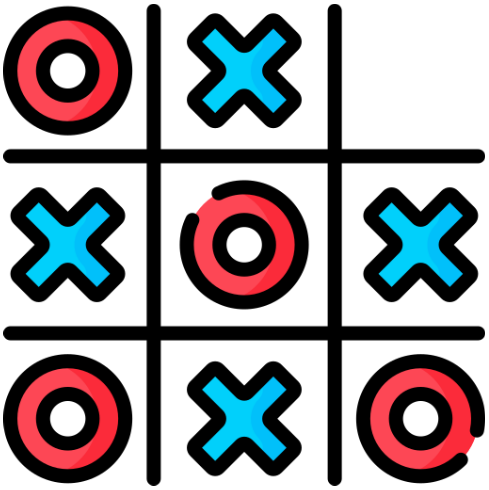
TIC TAC TOE WEB APPLICATION



1. Project Overview

The **Tic Tac Toe** web application is an interactive multiplayer game that allows two players to play in real time. It is built using **Java Servlets**, **JSP**, and the **Maven** framework for backend development, with a modern and responsive frontend design. The project emphasizes user-friendly gameplay, efficient session management, and scalable architecture.

2. Key Features of Tic Tac Toe

Core Gameplay

- **Multiplayer Mode**: Supports two players in a single game session.
- **Game Logic**: Automatically detects and declares a winner, or identifies a tie.
- **Move Validation**: Prevents invalid or repetitive moves with friendly error messages.
- **Game Reset**: Players can restart the game at any point after completion.

Innovative Enhancements

- **Real-Time Updates**: The game board refreshes dynamically after every move.
- **Responsive Design**: Works seamlessly across devices with different screen sizes.
- **Session Persistence**: Ensures all game states are preserved within a single session.

3. Technology Stack

Backend

- **Java Servlets**: For handling server-side logic and user requests.
- **JSP (JavaServer Pages)**: For rendering dynamic content.
- **JDBC**: For connecting to the SQL database to manage user and game data.

Frontend

- **HTML5 & CSS3**: For structuring and styling the web pages.
- **JavaScript & jQuery**: For enhancing interactivity and dynamic behavior.

Tools & Frameworks

- **Maven**: Manages dependencies and builds the project.
- **JUnit**: Provides robust unit testing for backend logic.

4. Functional Modules

User Authentication

- **Login and Registration**: Users can log in or register securely.
- **Session Tracking**: Ensures only authenticated users can access the game.

Game Mechanics

- **Real-Time Gameplay**: Tracks and validates moves for both players.
- **Result Computation**: Automatically determines the winner or a tie.
- **Reset Functionality**: Allows restarting the game within the same session.

Database Management

- **User Data**: Stores and retrieves user credentials.
- **Game Statistics**: (Optional) Tracks games played, wins, and losses for analysis.

4. Functional Modules

The Tic Tac Toe application consists of several functional modules that work together to deliver an engaging and seamless experience. Each module focuses on a specific aspect of the application, ensuring modularity and maintainability.

4.1 User Management

- **Registration Module**:
  - Allows new users to create accounts with basic validation for username, email, and password.
  - Prevents duplicate usernames and invalid entries.
- **Login Module**:
  - Authenticates users with secure credential validation.
  - Displays error messages for invalid login attempts.
- **Session Management**:
  - Maintains user sessions to track logged-in status and ensure secure access to gameplay features.

4.2 Game Logic Module

- Implements core game rules, such as:
  - Validating player moves.
  - Checking for winning conditions (rows, columns, diagonals).
  - Detecting ties when all cells are filled without a winner.
- Handles turn alternation between Player X and Player O.
- Provides real-time updates on the game state.

4.3 Gameplay Interface Module

- **Frontend Interaction**:
  - Displays the game board and updates it dynamically based on player moves.
  - Shows real-time feedback, such as "Player X's Turn" or "Game Over".
- **Result Handling**:
  - Notifies players of the game result (Win, Lose, Tie) at the end of the game.
  - Offers an option to reset the game board and start a new session.

4.4 Data Management Module

- **User Data Handling**:
  - Stores and retrieves user credentials securely using a database.
  - Ensures data consistency during registration and login.

    .

4.5 Error Handling and Validation Module

- Validates inputs for registration and login (e.g., non-empty fields, email format).
- Prevents invalid gameplay actions, such as:
  - Playing out of turn.
  - Clicking on already filled cells.
- Displays user-friendly error messages for smooth troubleshooting.

5. Testing

The Tic Tac Toe application underwent comprehensive testing to ensure functionality, reliability, and a seamless user experience.

5.1 Unit Testing

- Tested game logic methods such as move validation, win detection, and tie conditions.
- Verified session management to maintain game state during gameplay.
- Example: Ensured the checkWinner() method correctly identifies winning patterns or a tie.

5.2 Service Layer Testing

- Validated database operations for user registration, login, and session handling.
- Tested the JDBC connection for robustness and appropriate error handling.
- Example: Ensured login functionality only authenticates valid credentials.

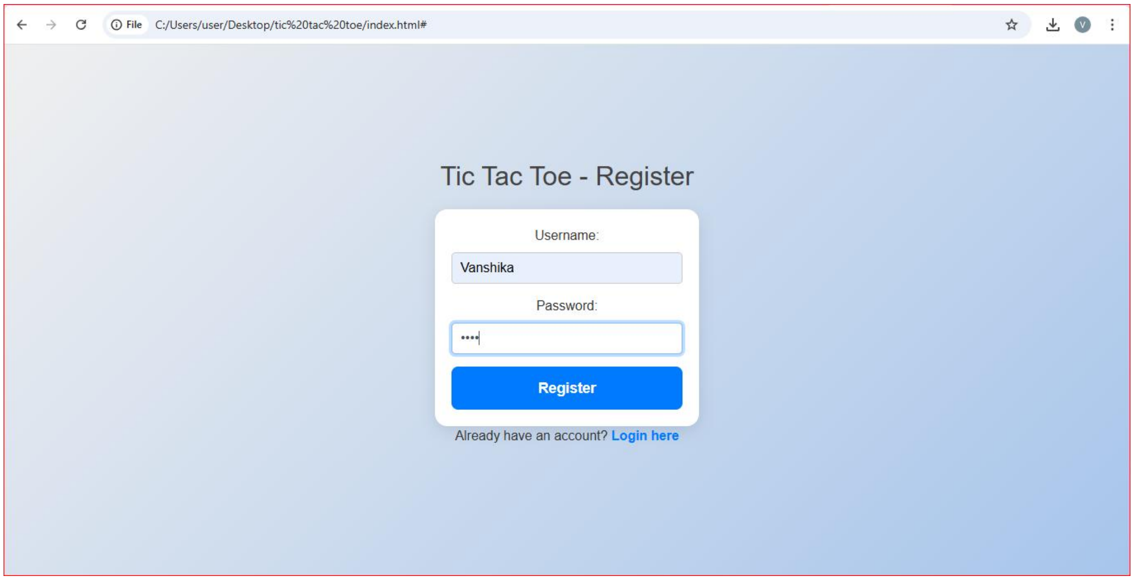
5.3 Integration Testing

- Verified end-to-end workflows, such as user login, gameplay updates, and result display.
- Ensured smooth interaction between the frontend (JSP) and backend (Servlets).
- Example: Checked if user actions (e.g., making a move) correctly update the game state in real time.

5.4 UI/UX Testing

- Ensured the interface is responsive and functional across various devices and browsers.
- Verified error messages for clarity and accuracy (e.g., invalid login or duplicate moves).

REGISTRATION



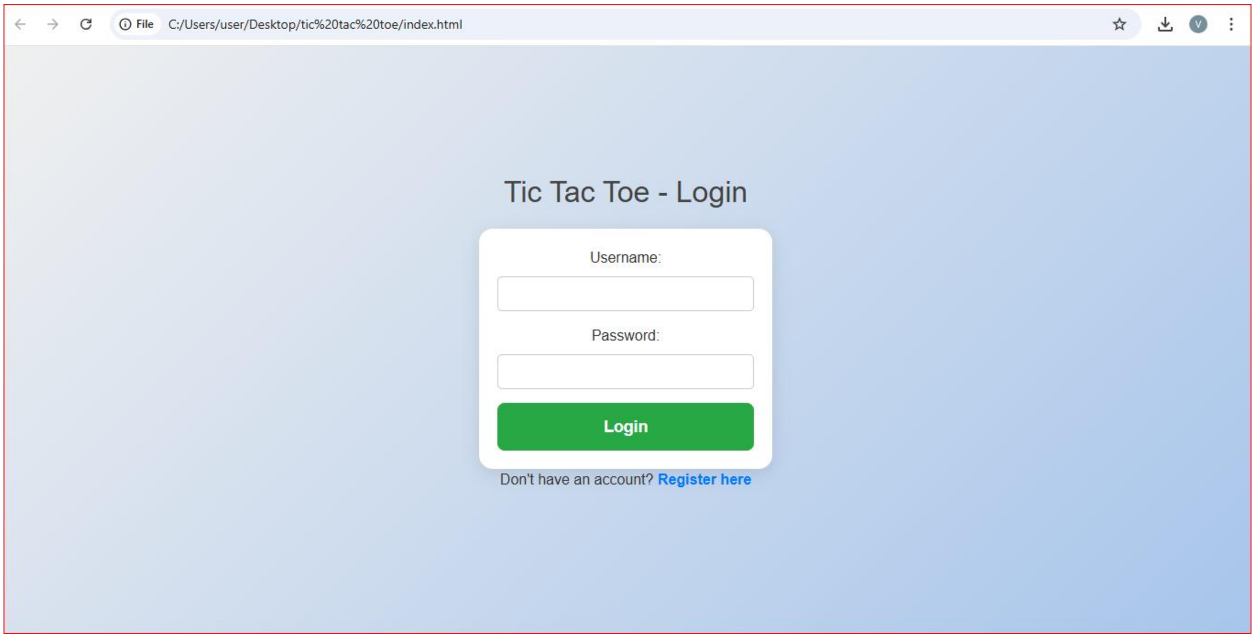
Tic Tac Toe - Register

Username: Vanshika

Password: ••••

Already have an account? Login here

LOGIN



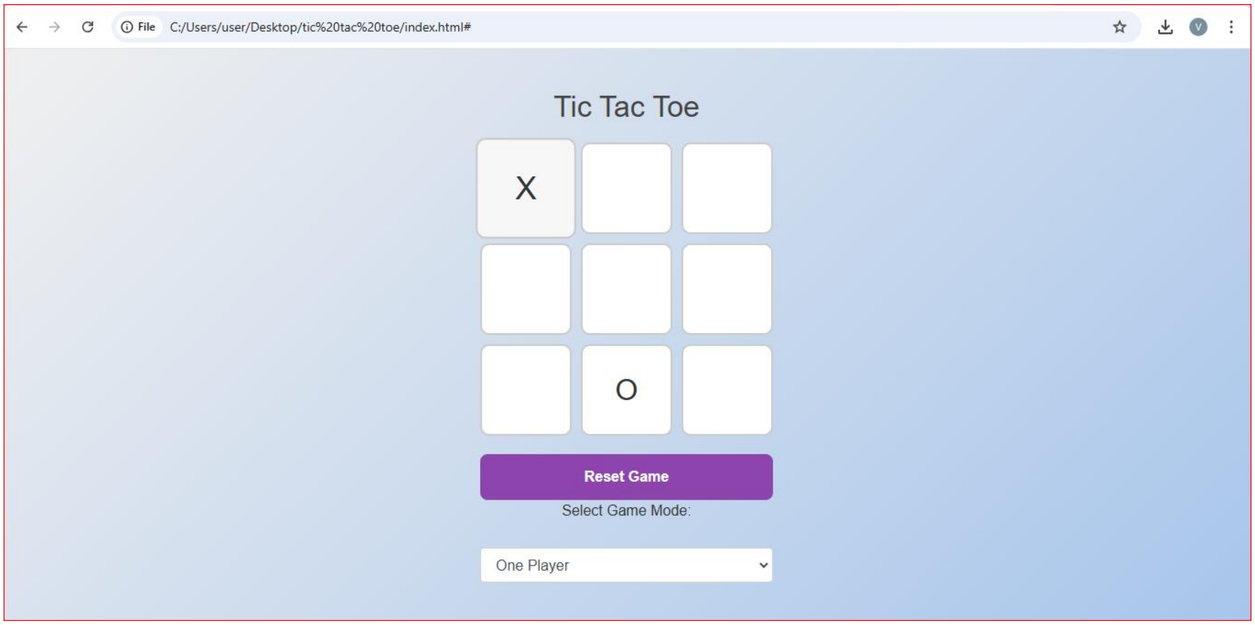
Tic Tac Toe - Login

Username:

Password:

Don't have an account? Register here

ONE PLAYER MODE



C:/Users/user/Desktop/tic%20tac%20toe/index.html#

Tic Tac Toe

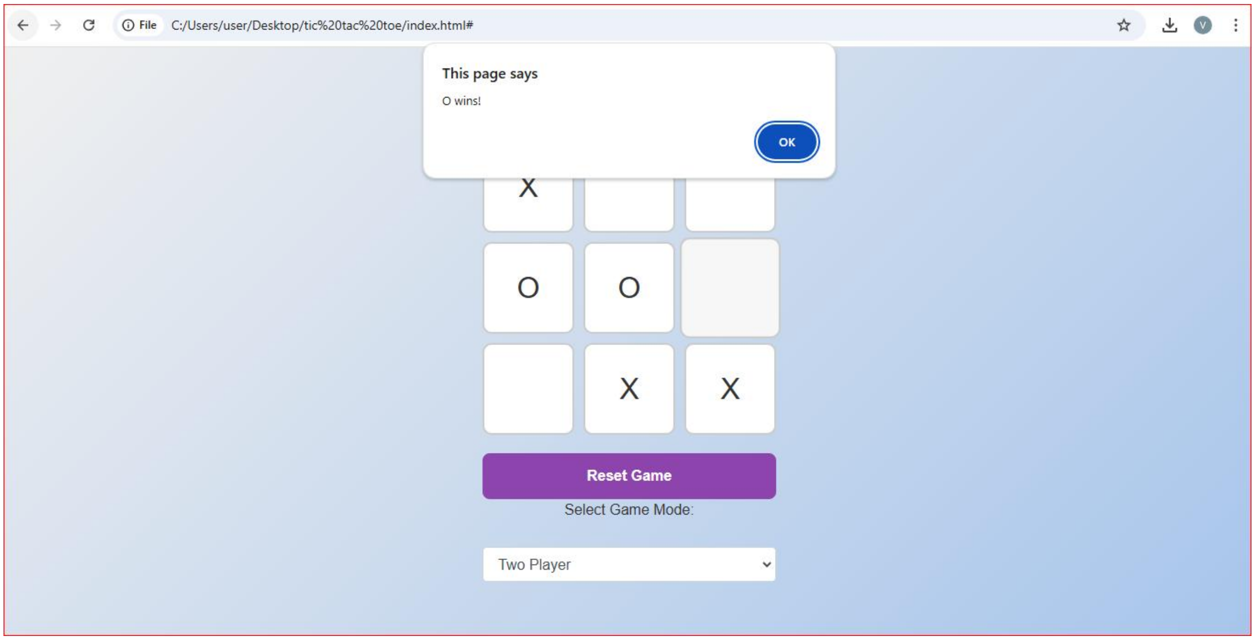
X		
	O	

Reset Game

Select Game Mode:

One Player

WINNER



C:/Users/user/Desktop/tic%20tac%20toe/index.html#

This page says

O wins!

OK

X		
O	O	
	X	X

Reset Game

Select Game Mode:

Two Player

6. Login Page

Innovative Features

- **User-Friendly Design**: Clean interface with intuitive fields for login and registration.
- **Error Handling**: Displays specific messages for invalid credentials or failed logins.
- **Data Validation**: Enforces secure and sanitized inputs to prevent SQL injection.

7. Challenges and Solutions

Challenge 1: Real-Time Gameplay Synchronization

- **Problem**: Synchronizing moves between two players in real time.
- **Solution**: Implemented session-based state management and AJAX-based requests for smooth updates.

Challenge 2: Handling Game State Across Sessions

- **Problem**: Preserving game state when sessions expire or reset.
- **Solution**: Designed a scalable architecture to handle state persistence using session attributes.

Challenge 3: Cross-Browser Compatibility

- **Problem**: Ensuring consistent UI across different browsers.
- **Solution**: Adopted CSS normalization and thorough testing on major browsers like Chrome, Firefox, and Safari.

8. Conclusion and Future Scope

Conclusion

The Tic Tac Toe project successfully demonstrates the integration of **Java Servlets**, **JSP**, and **Maven** for developing interactive web applications. It provides a user-friendly platform for gameplay with efficient session management and clean design.

Future Scope

- **Enhanced Multiplayer Features**: Enable remote multiplayer support using WebSockets.
- **AI Integration**: Add a single-player mode with an AI opponent.
- **Analytics Dashboard**: Introduce game statistics like win/loss ratios for users.
- **Mobile App Development**: Expand the project into a cross-platform mobile application using frameworks like Flutter or React Native.