**Tic-Tac-Toe problem using full stack development**

**Backend Development (Java, Spring Boot):**

1. Project Structure:

**tic-tac-toe-backend/**

|-- src/

| |-- main/

| | |-- java/

| | | |-- com/

| | | |-- example/

| | | |-- tic\_tac\_toe/

| | | |-- TicTacToeApplication.java

| | | |-- controller/

| | | | |-- GameController.java

| | | |-- model/

| | | | |-- Game.java

| | | | |-- Player.java

| | | |-- repository/

| | | | |-- GameRepository.java

| | | |-- service/

| | | |-- GameService.java

|-- pom.xml

**Code Snippets:**

Game.java (Model class for the Tic Tac Toe game)

@Entity

public class Game {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Enumerated(EnumType.STRING)

private GameState state;

@OneToMany(mappedBy = "game", cascade = CascadeType.ALL)

private List<Move> moves;

// Constructors, getters, setters

}

GameController.java (RESTful API endpoints)

@RestController

@RequestMapping("/api/game")

public class GameController {

@Autowired

private GameService gameService;

@PostMapping("/start")

public ResponseEntity<Game> startGame() {

Game game = gameService.startGame();

return ResponseEntity.ok(game);

}

// Other endpoints for making moves, getting game state, etc.

**}**

**Frontend Development (React JS):**

**1. Project Structure:**

tic-tac-toe-frontend/

|-- src/

| |-- components/

| | |-- GameBoard.js

| | |-- Square.js

| |-- App.js

| |-- index.js

**2. Code Snippets:**

GameBoard.js (Component for displaying the game board)

import React from 'react';

import Square from './Square';

const GameBoard = ({ squares, onClick }) => {

return (

<div className="game-board">

{squares.map((square, index) => (

<Square key={index} value={square} onClick={() => onClick(index)} />

))}

</div>

);

};

export default GameBoard;

**Square.js (Component for individual squares in the game board)**

import React from 'react';

const Square = ({ value, onClick }) => {

return (

<button className="square" onClick={onClick}>

{value}

</button>

);

};

export default Square;

**App.js (Main component to orchestrate the game)**

import React, { useState } from 'react';

import GameBoard from './components/GameBoard';

const App = () => {

const [squares, setSquares] = useState(Array(9).fill(null));

const [xIsNext, setXIsNext] = useState(true);

const handleClick = (index) => {

const squaresCopy = [...squares];

if (calculateWinner(squaresCopy) || squaresCopy[index]) {

return;

}

squaresCopy[index] = xIsNext ? 'X' : 'O';

setSquares(squaresCopy);

setXIsNext(!xIsNext);

};

const winner = calculateWinner(squares);

const status = winner ? `Winner: ${winner}` : `Next player: ${xIsNext ? 'X' : 'O'}`;

return (

<div className="game">

<div className="game-board">

<GameBoard squares={squares} onClick={handleClick} />

</div>

<div className="game-info">

<div>{status}</div>

</div>

</div>

);

};

function calculateWinner(squares) {

// Logic to determine the winner

}

export default App;

**This is a basic setup for a Tic Tac Toe game using full-stack development with Spring Boot for the backend and React JS for the frontend.**