**Task 4**

**Title:** Chess board

**Description:** Develop a "Chess" board game with redux when one with white other is black colour of squre while clicking on white one colour change to yellow and black to red.

**Project Overview:**

This project implements a simple chessboard-like grid using React and Redux Toolkit. The board consists of 64 squares arranged in an 8x8 grid, where each square can change its colour based on user interaction. The application allows users to toggle the colours of the squares between white, black, yellow, and red, providing a visual representation of state changes on the board.

**Installation:**

**npm install @reduxjs/toolkit react-redux**

**Key Features:**

* **Dynamic Colour Toggling**: Users can click on any square to change its colour in a cycle: white → yellow → black → red → back to white.
* **Redux State Management**: The application utilizes Redux Toolkit for managing the state of the board, making it easy to track and update the colours of the squares.
* **Responsive Design**: The board is designed to be responsive and can be easily integrated into a larger application.

**Technical Details:**

* **Frameworks and Libraries**: The project is built using React for the UI and Redux Toolkit for state management.
* **State Structure**: The board's state is initialized as a two-dimensional array representing the 8x8 grid, with each square having an **id** and a **colour**.
* **Reducers**: The main reducer, **toggleColour**, updates the color of a square based on its current state and the action dispatched.

**Code Structure:**

* **initialBoardState**: Defines the initial state of the board as an 8x8 grid with alternating colours.
* **boardSlice**: Contains the slice of state related to the board, including the initial state and reducers for colour toggling.
* **store**: Configures the Redux store with the board reducer.

**Example Code Snippet:**

const initialBoardState = Array(8)

  .fill(null)

  .map((\_, row) =>

    Array(8)

      .fill(null)

      .map((\_, col) => ({

        id: `${row}-${col}`,

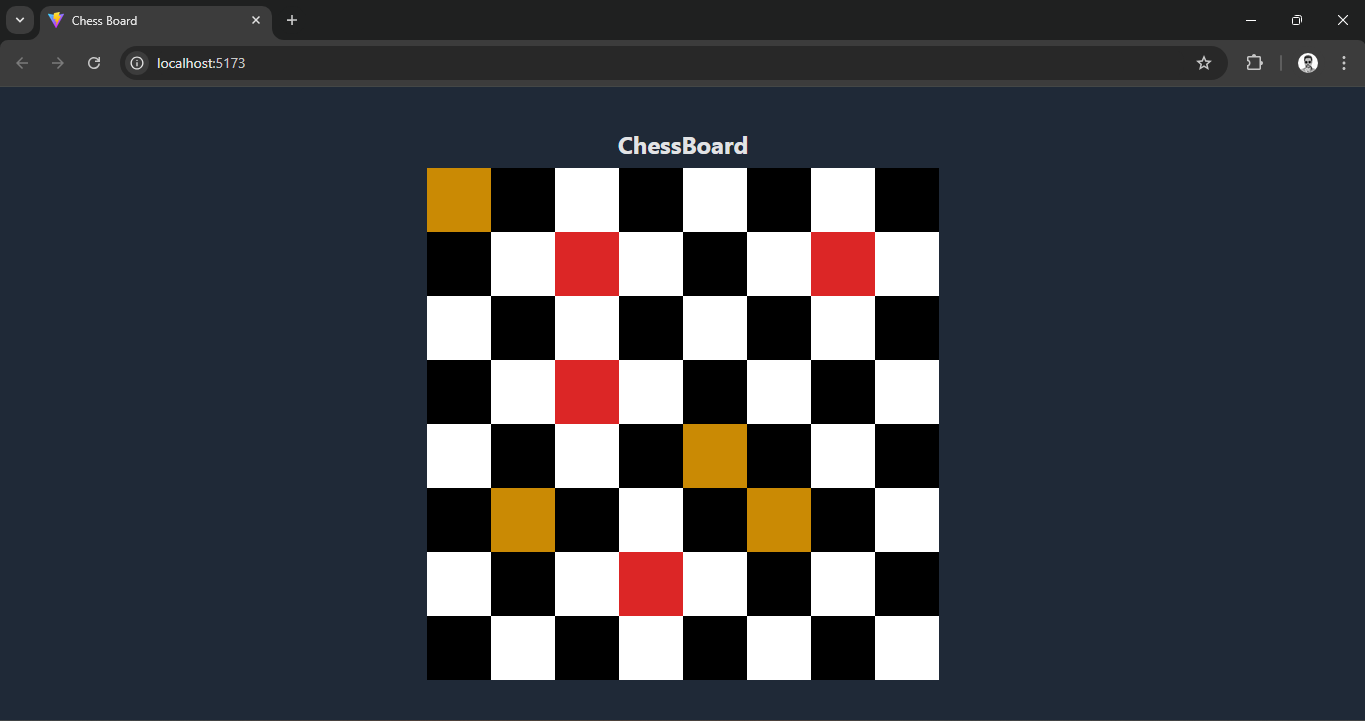
        color: (row + col) % 2 === 0 ? 'white' : 'black',

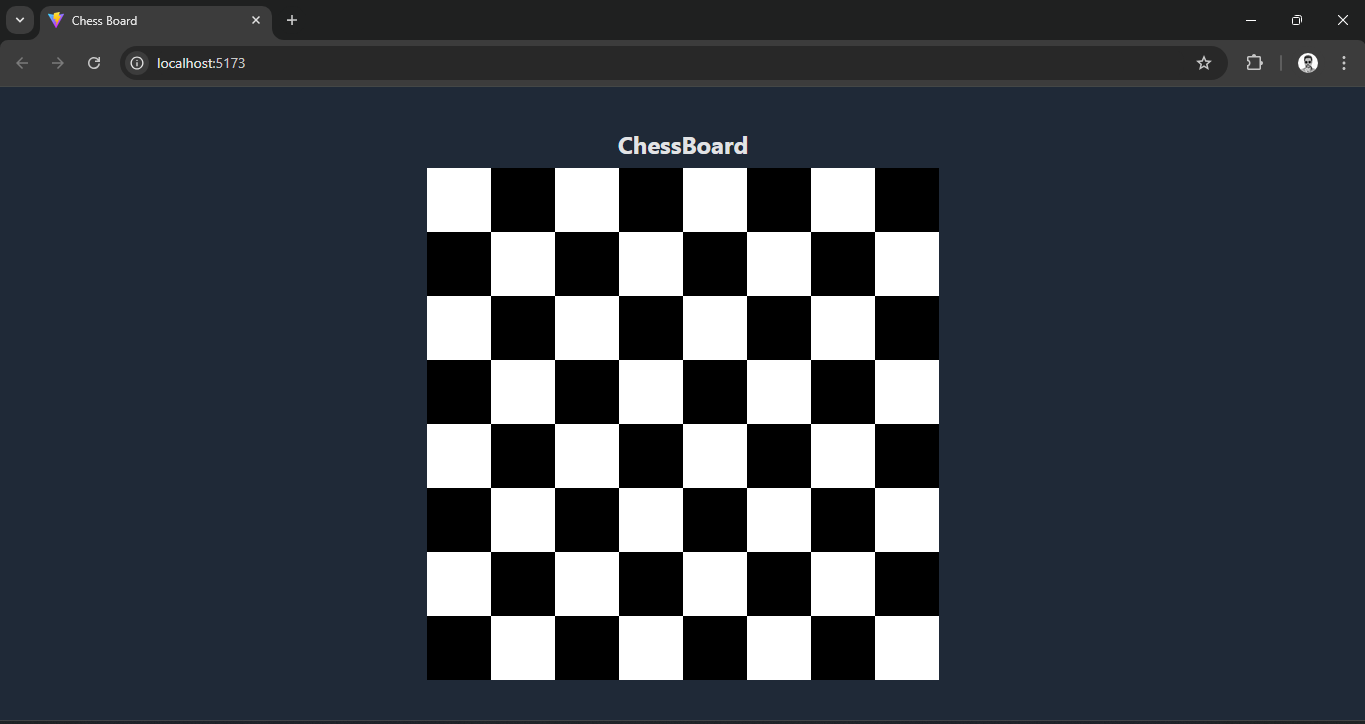
      }))

  );

this code is a two-dimensional array (an array of arrays) that represents the chessboard. Each element of the outer array corresponds to a row, and each element of the inner arrays corresponds to a square on that row. Each square is represented as an object with an **id** and a **colour** property.

**Screenshots:**

****

****

**Conclusion:**

This project successfully demonstrates the use of React and Redux Toolkit to create an interactive chessboard application.