Project Summary: 8 Queens Puzzle

This project implements an interactive **8 Queens Puzzle** using **HTML**, **CSS**, and **JavaScript**, where users manually place queens on a chessboard following these rules:

1. **Each row, column, and diagonal can only contain one queen.**
2. **Queens cannot be adjacent to each other (including diagonals).**

Code Overview

1. HTML Structure

- **Header**: A title (<h1>) and a status message (<p>) guide the user.
- **Chessboard**: An 8x8 grid (#chessboard) dynamically generated in JavaScript.
- **Reset Button**: A button (#resetButton) allows users to reset the board to start fresh.

2. CSS for Styling

- **Centering and Layout**: The page is centered using **Flexbox**, ensuring the chessboard and elements are aligned in the middle of the screen.
- **Chessboard Design**:
    - Each cell (.cell) has a light gray border (1px solid #ddd) for clear separation.
    - Alternate cell colors (white and light gray) mimic a chessboard.
    - A subtle drop shadow around the chessboard adds elegance.
- **Responsiveness**: The board is fixed-sized (60px squares), keeping the layout clean and uniform.

3. JavaScript for Functionality

- **Dynamic Chessboard Creation**:
  The chessboard grid is generated dynamically using a for loop in createBoard(), ensuring flexibility and ease of updates. Each square is clickable.
- **Queen Placement Rules**:
    - The handleCellClick(row, col) function allows users to place or remove queens.
    - Valid placements are checked using the isValidPlacement(row, col) function:
        1. Ensures no queen exists in the same row, column, or diagonal.
        2. Checks adjacent cells for queens to enforce the "no adjacency" rule.
    - Invalid moves are blocked, and feedback is provided via the **status message**.
- **Win Detection**:

    - The checkWin() function checks if all 8 queens are placed correctly. Upon success, a congratulatory message is displayed.

- **Reset Functionality**:
    - The resetButton clears the chessboard, removes all queens, and resets the status message.

How It Works

1. **Game Start**:
    - The chessboard is displayed, and users can click squares to place or remove queens.
    - Each click triggers the **placement rules check** to validate the move.
2. **Status Feedback**:
    - Users are informed whether their move was valid, invalid, or if they've won.
3. **Reset**:
    - Users can reset the board to start over.

Key Design Considerations

- **Interactivity**: Real-time feedback ensures an engaging experience.
- **Accessibility**: Simple and intuitive layout for users of all levels.
- **Code Modularity**: Functions are organized for clarity and easy maintenance.
