

```
// Define constants for ball colours and prices

const BALL_COLORS = {

  RED: { min: 1, max: 13 },

  YELLOW: { min: 14, max: 25 },

  GREEN: { min: 26, max: 37 },

  BLUE: { min: 38, max: 52 }

};


const PRICES = {

  LOTTO_BOARD: 5,

  LOTTO_PLUS_1: 2.5,

  LOTTO_PLUS_2: 2.5

};


// Define functions for generating random numbers

function getRandomNumber(min, max) {

  return Math.floor(Math.random() * (max - min + 1)) + min;

}


function generateRandomNumbers(count, min, max) {

  const numbers = [];

  while (numbers.length < count) {

    const randomNumber = getRandomNumber(min, max);

    if (!numbers.includes(randomNumber)) {

      numbers.push(randomNumber);

    }

  }

  return numbers.sort((a, b) => a - b);

}


// Define functions for managing user data
```

```
function getUserData() {  
  return JSON.parse(localStorage.getItem('userData')) || [];  
}
```

```
function saveUserData(userData) {  
  localStorage.setItem('userData', JSON.stringify(userData));  
}
```

```
// Define functions for managing draw data
```

```
function getDrawData() {  
  return JSON.parse(localStorage.getItem('drawData')) || [];  
}
```

```
function saveDrawData(drawData) {  
  localStorage.setItem('drawData', JSON.stringify(drawData));  
}
```

```
// Define functions for managing winning tickets
```

```
function getWinningTickets() {  
  return JSON.parse(localStorage.getItem('winningTickets')) || [];  
}
```

```
function saveWinningTickets(winningTickets) {  
  localStorage.setItem('winningTickets', JSON.stringify(winningTickets));  
}
```

```
// Define functions for managing user interface
```

```
function renderUserInterface() {  
  // Render the user interface based on the current state  
}
```

```
function handleUserInput(event) {  
  // Handle user input based on the event  
}  
  
// Define functions for managing the draw simulation  
function simulateDraw() {  
  // Generate the winning numbers for each game  
  const winningNumbers = {  
    LOTTO: generateRandomNumbers(6, 1, 52),  
    LOTTO_PLUS_1: generateRandomNumbers(6, 1, 52),  
    LOTTO_PLUS_2: generateRandomNumbers(6, 1, 52)  
  };  
  
  // Get the user data and draw data from localStorage  
  const userData = getUserData();  
  const drawData = getDrawData();  
  
  // Create a new draw entry  
  const drawEntry = {  
    date: new Date().toLocaleString(),  
    winningNumbers  
  };  
  
  // Save the draw entry to localStorage  
  drawData.push(drawEntry);  
  saveDrawData(drawData);  
  
  // Check for winning tickets  
  const winningTickets = [];  
  userData.forEach(user => {  
    user.tickets.forEach(ticket => {
```

```
const matchedNumbers = {  
  LOTTO: 0,  
  LOTTO_PLUS_1: 0,  
  LOTTO_PLUS_2: 0  
};
```

```
ticket.boards.forEach(board => {  
  Object.keys(winningNumbers).forEach(game => {  
    const matchedNumbers = board[game].filter(number =>  
      winningNumbers[game].includes(number)  
    ).length;  
    ticket[game].matchedNumbers += matchedNumbers;  
  });  
});
```

```
if (ticket.LOTTO.matchedNumbers >= 3 || ticket.LOTTO_PLUS_1.matchedNumbers >= 3 ||  
ticket.LOTTO_PLUS_2.matchedNumbers >= 3) {  
  winningTickets.push(ticket);  
}  
});  
});
```

```
// Save the winning tickets to localStorage
```

```
saveWinningTickets(winningTickets);
```

```
// Alert users and admins about winning tickets
```

```
winningTickets.forEach(ticket => {  
  const user = userData.find(u => u.tickets.includes(ticket));
```

```
  // Alert the user about their winning ticket
```

```
  alert(`Congratulations! You have a winning ticket with ${ticket.LOTTO.matchedNumbers} matched  
numbers in the Lotto game.`);
```

```
  // Alert the admin about the winning ticket
```

```

    alert(`Admin: A user has won with ticket ${ticket.id}.`);
  });
}

// Define the main application logic
function main() {
  // Render the user interface
  renderUserInterface();

  // Handle user input
  document.addEventListener('input', handleUserInput);

  // Simulate draw event (Admin-only function)
  document.getElementById('simulateDrawBtn').addEventListener('click', () => {
    const isAdminLoggedIn = localStorage.getItem('adminLoggedIn') === 'true';
    if (!isAdminLoggedIn) {
      alert('Only admins can simulate draws.');
```

return;

```

    }
    simulateDraw();
  });

  // User login function
  document.getElementById('userLoginBtn').addEventListener('click', () => {
    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    // Example of user authentication (replace with actual authentication logic)
    if (username === 'user' && password === 'password') {
      // Set user logged in status
      localStorage.setItem('userLoggedIn', 'true');
```

```
    alert('Logged in as User.');
```

```
  } else {
```

```
    alert('Invalid username or password.');
```

```
  }
```

```
});
```

```
// Admin login function
```

```
document.getElementById('adminLoginBtn').addEventListener('click', () => {
```

```
  const username = document.getElementById('adminUsername').value;
```

```
  const password = document.getElementById('adminPassword').value;
```

```
  // Example of admin authentication (replace with actual authentication logic)
```

```
  if (username === 'admin' && password === 'adminpassword') {
```

```
    // Set admin logged in status
```

```
    localStorage.setItem('adminLoggedIn', 'true');
```

```
    alert('Logged in as Admin.');
```

```
  } else {
```

```
    alert('Invalid admin credentials.');
```

```
  }
```

```
});
```

```
// Check for winning tickets when the user logs in
```

```
document.addEventListener('login', () => {
```

```
  const winningTickets = getWinningTickets();
```

```
  const userLoggedIn = localStorage.getItem('userLoggedIn') === 'true';
```

```
  const adminLoggedIn = localStorage.getItem('adminLoggedIn') === 'true';
```

```
  if (userLoggedIn) {
```

```
    const user = getUserData().find(u => u.loggedIn);
```

```
    winningTickets.forEach(ticket => {
```

```
      if (user.tickets.includes(ticket)) {
```

```
        // Alert the user about their winning ticket

        alert(`Congratulations! You have a winning ticket with ${ticket.LOTTO.matchedNumbers}
matched numbers in the Lotto game.`);

    }

});

}

if (adminLoggedIn) {

    // Admin alert about winning tickets

    winningTickets.forEach(ticket => {

        alert(`Admin: A user has won with ticket ${ticket.id}.`);

    });

}

});

}

// Run the main application logic

main();
```