🧪 Testing Plan

1. Authentication Tests

- **Login View Test**
  - **Input Valid Data**: Test with a registered user and correct credentials.
  - **Invalid Username/Password**: Verify error messages when incorrect username or password is provided.
  - **Form Errors**: Submit incomplete data and check if error messages display correctly.
  - **Redirection**: Verify redirection to the home page on successful login.
- **Register View Test**
  - **Valid Data**: Test successful registration flow with unique username, email, and password.
  - **Duplicate Username**: Attempt to register with an already-taken username to confirm error handling.
  - **Field Validation**: Submit invalid emails, weak passwords, and other faulty inputs to test validation messages.
- **Logout Test**
  - Ensure logout functionality works and users are redirected to the login page afterward.

2. Game Functionality Tests

- **Game View Tests**
  - Verify that the game view renders correctly for authenticated users.
  - **AI Model Selection**: Test human vs. AI game with various AI models (e.g., “gemini,” “claude”).
  - **Human vs. Human**: Ensure the human vs. human game view is accessible.
- **Async Game Simulation Tests**
  - **Human vs. AI Game**: Test if asynchronous gameplay with the AI models logs the game outcome correctly.
  - **AI vs. AI Game**: Run AI vs. AI game simulations and verify outcome is stored as expected.
- **Outcome Storage Test**
  - Verify that game outcomes are correctly stored in the `GameResult` model.

3. Leaderboard View Tests

- **Data Sorting**: Confirm that AI models are sorted by win count and displayed correctly.
- **Draw Display**: Ensure draw counts are visible and accurate.

4. Summary & Statistics Views

- **Data Ordering**: Confirm that game results are shown in descending order based on `game_date`.
- **Statistics**: Verify that all game outcomes are included in the statistics view.

5. Error Handling & Edge Cases

- **Invalid Game Model**: Test invalid AI model names to confirm appropriate error handling.
- **Missing Fields in Requests**: Test API requests with missing fields to ensure the app handles them gracefully.
- **Concurrency Test**: Verify application stability when multiple users are playing or accessing views concurrently.

---

6. Asynchronous Function Tests

- **Game Simulation**: Ensure the asynchronous game simulation handles different AI combinations and saves logs without blocking main execution.
- **Player Type Switching**: Confirm seamless switching between human and AI players.

7. User Interface Tests

- **Responsive Design**: Test UI across devices (desktop, mobile, tablet).
- **Error Messages**: Ensure error messages are clear, concise, and placed appropriately.

---

8. Performance Testing

- **Load Testing**: Simulate high traffic conditions to see how many concurrent users the server can handle.
- **Database Performance**: Evaluate database query times and optimize where needed.