

Fig1 : Homepage

Home Page :

The home page function in a **Managing sports League** web application renders the home.html template when a request is made. It takes the request object as a parameter and returns the rendered template. This function serves to display the home page of the web application. Non-authenticated users would only see "Login" and "Register" links. This approach simplifies the menu by treating all logged-in users the same, with differentiating between regular users and staff members. It ensures that all authenticated users have access to the same features, streamlining the user interface

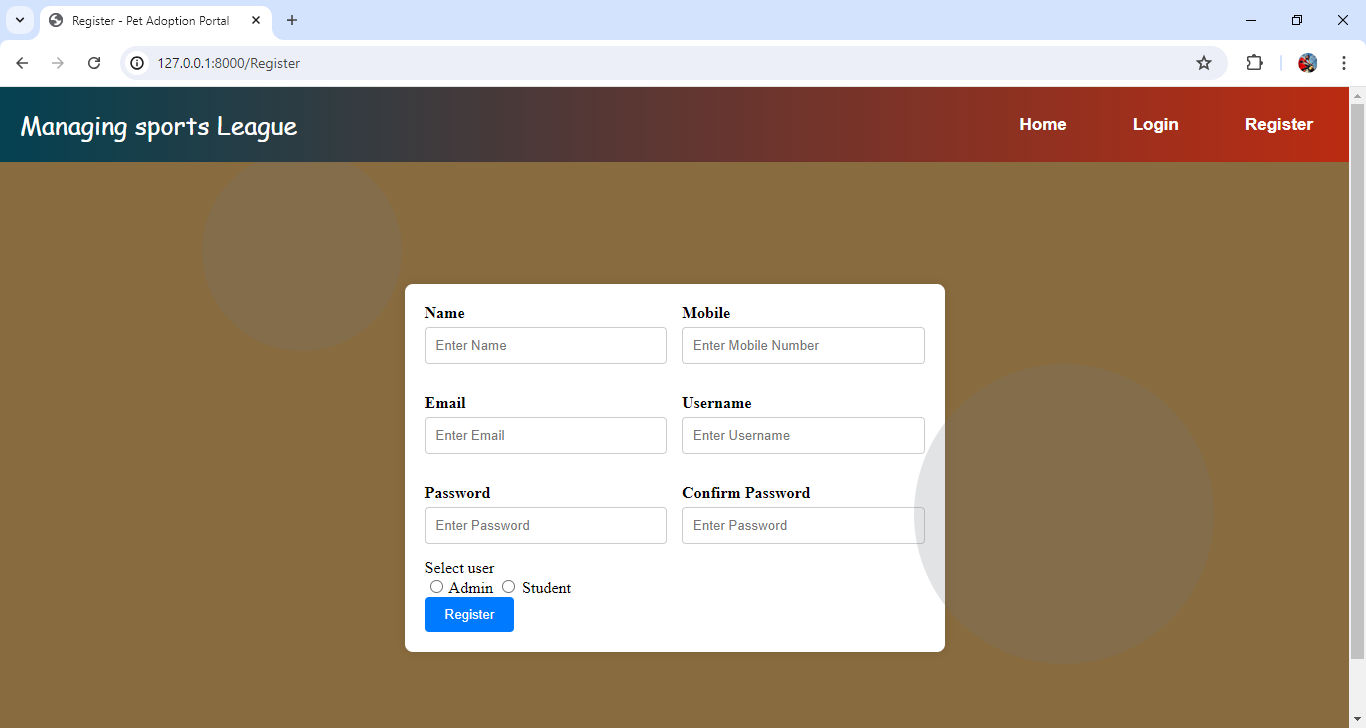


Fig 2 : Registration Page

REGISTER :

The register function handles user registration in a **Managing sports League** web application. When a POST request is made, it retrieves user details from the form, including name, email, username, password, confirmation password, and user type (admin or regular). It checks if the passwords match and whether the username already exists. If the username is unique and passwords match, a new user is created with the provided details, including setting the user as staff if selected. On success, it redirects to the login page with a success message. If there are errors, appropriate error messages are displayed, and the user is redirected back to the registration page. For GET requests, it renders the registration form.

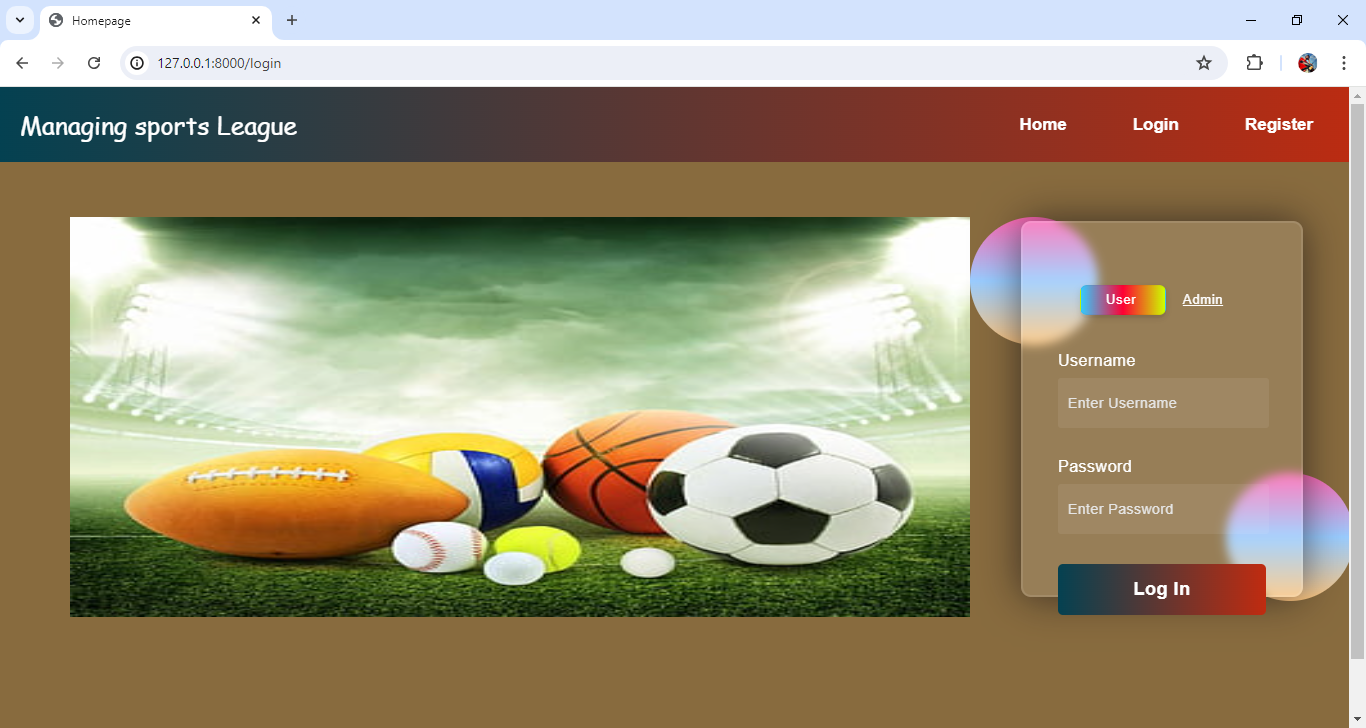


Fig 3 : Login for User and Admin

LOGIN :

The login function handles user authentication in a **Managing sports League** web application. It processes POST requests by retrieving the username and password, authenticates the user, and logs them in if the credentials are correct. On successful login, it redirects to the home page and shows a success message. If authentication fails, it redirects back to the login page with an error message. For GET requests, it renders the login page.



Fig 4: Admin Homepage After Login

Admin Home Page :

The navigation menu would display the same options for all authenticated users. Logged-in users would see links to "Create Team," "Schedule Matches," "Update Scores," and "Logout," regardless of their role or privileges. Non-authenticated users would only see "Login" and "Register" links. This approach simplifies the menu by treating all logged-in users the same, with differentiating between regular users and staff members. It ensures that all authenticated users have access to the same features, streamlining the user interface.

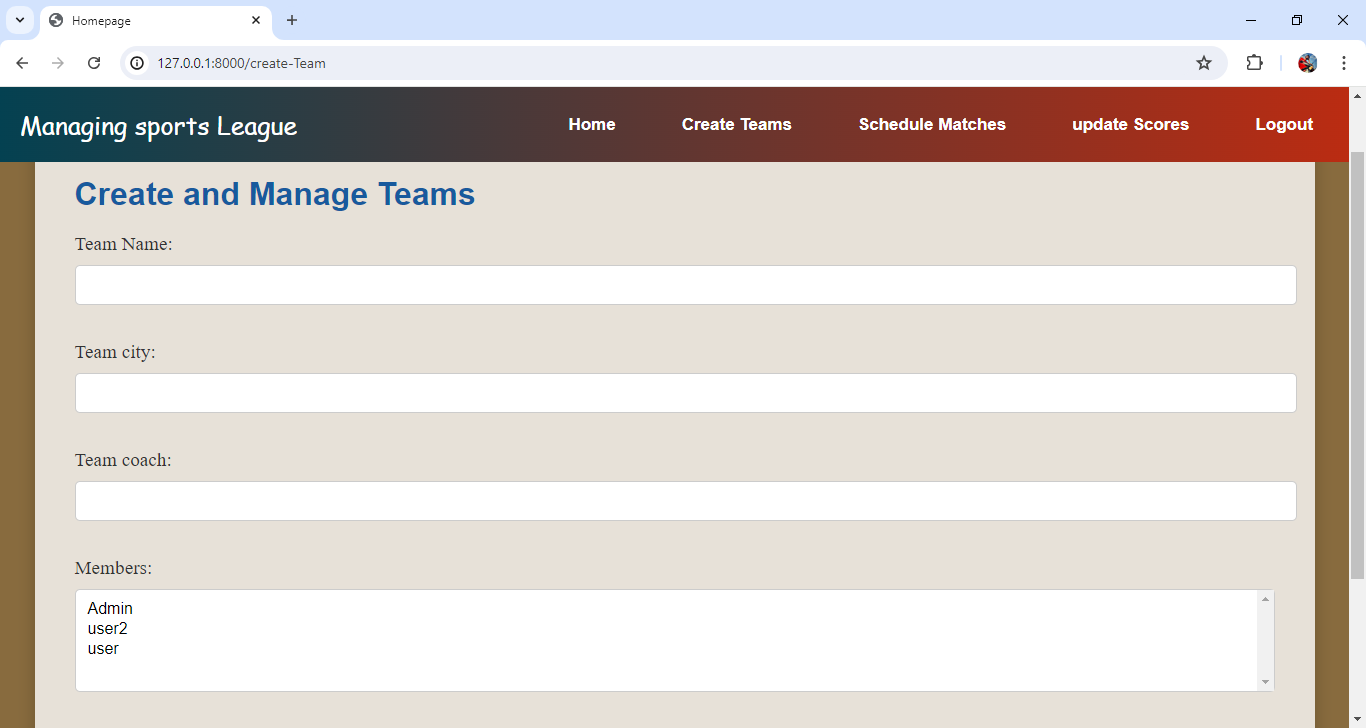


Fig 5: Create Team Page for Creating New Teams

The Team create function is a **Managing sports League** view that handles creating new team entries via a form. It checks if the request method is POST to process the submitted form data; if valid, it saves the form and redirects to the home page. If the request method is not POST, it displays an empty Team Create Form. The function renders the team html template, passing the form to it for user interaction. This allows users to create new teams efficiently through a form interface.

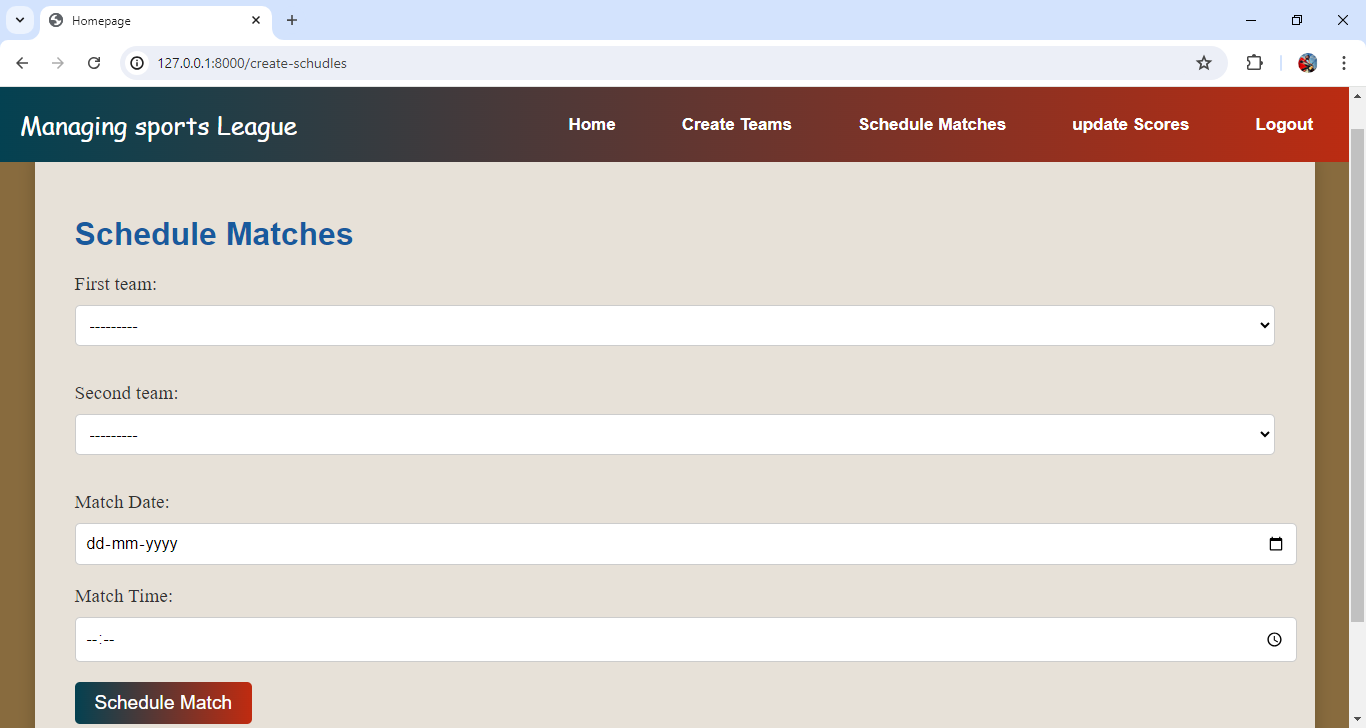


Fig 6: schedule page for match between the teams

The New schedules function in **Managing sports League** handles creating new match schedules. For POST requests, it processes and validates form data, extracting match date, match time, First team, and Second team to create and save a schedule match object. If the form is valid, it redirects to the home page. For GET requests, it displays an empty schedule match Create Form. The function renders the schedule.html template with the form, allowing users to input match details. This facilitates the creation and scheduling of new matches through a user-friendly interface.

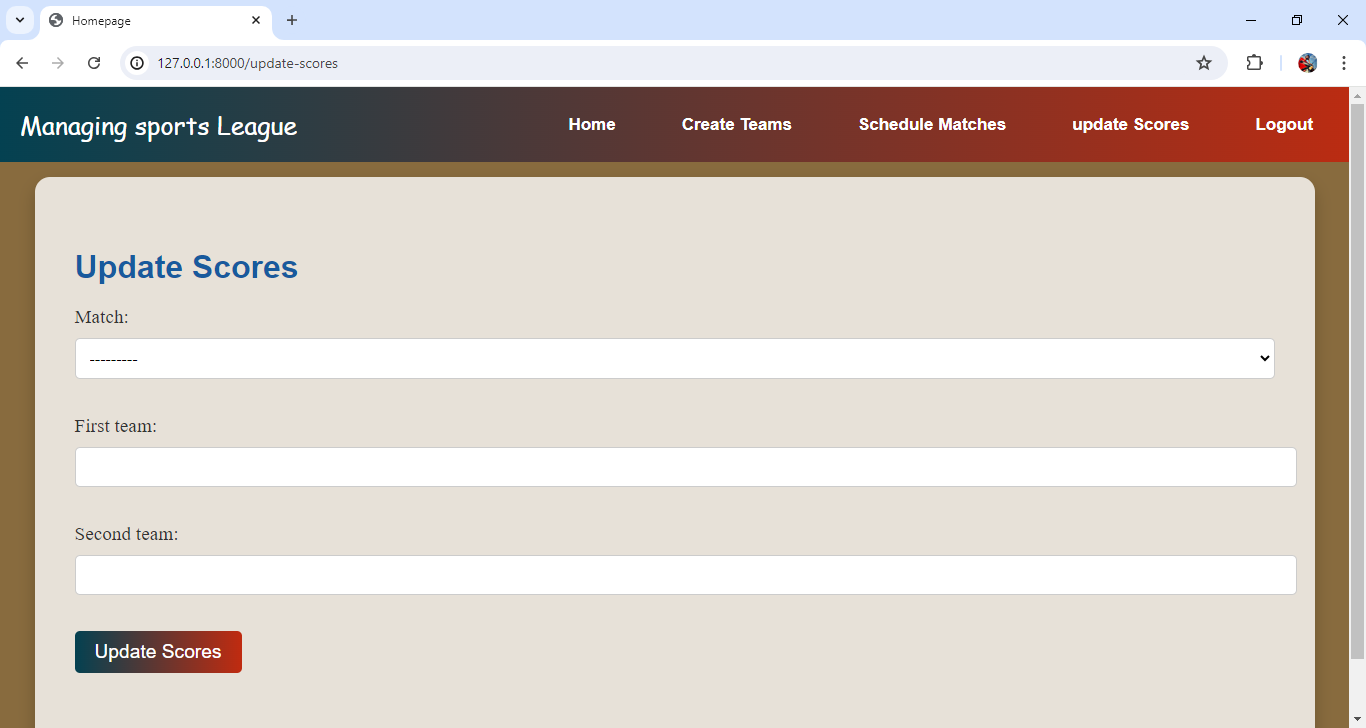


Fig 7: Update Score Page For Updating Score Latest

The Update scores function in **Managing sports League** handles updating or creating match scores. For POST requests, it processes and validates form data, extracting match and team scores. If a score entry for the match exists, it updates the scores; otherwise, it creates a new score entry. The function then redirects to the home page. For GET requests, it displays an empty Score Create Form. The function renders the scores html template with the form, enabling users to input and update match scores efficiently. This ensures accurate and up-to-date score management.



Fig 8 : User homepage After Login

User Home Page :-

The navigation menu would display the same options for all authenticated users. Logged-in users would see links to "View Team," "Check schedule," "Track Scores," "contact," and "Logout," regardless of their role or privileges. Non-authenticated users would only see "Login" and "Register" links. This approach simplifies the menu by treating all logged-in users the same, with differentiating between regular users and staff members. It ensures that all authenticated users have access to the same features, streamlining the user interface.

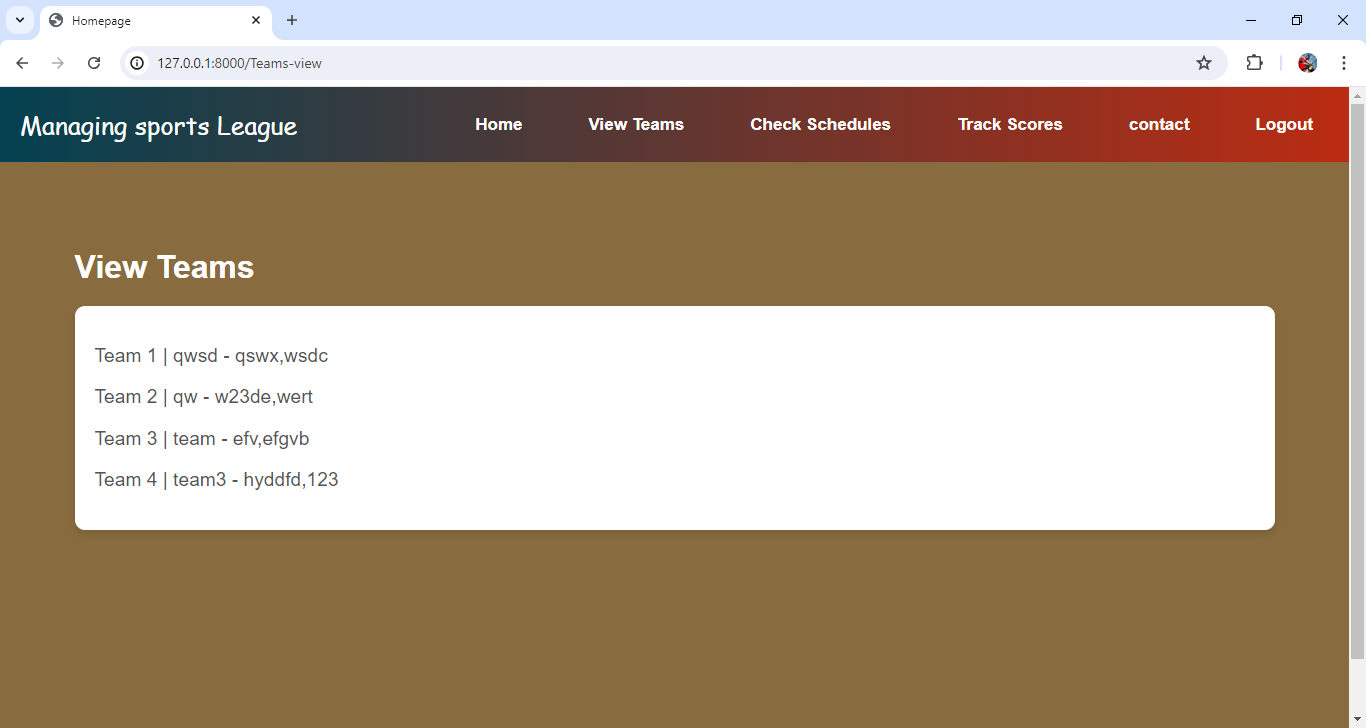


Fig 9: View team page

The view Team function in Retrieves all Team objects from the database and renders them in the useractions.html template. The data is passed to the template under the context variable Teams, allowing users to view a list of all teams. This provides a straightforward way to display team information.

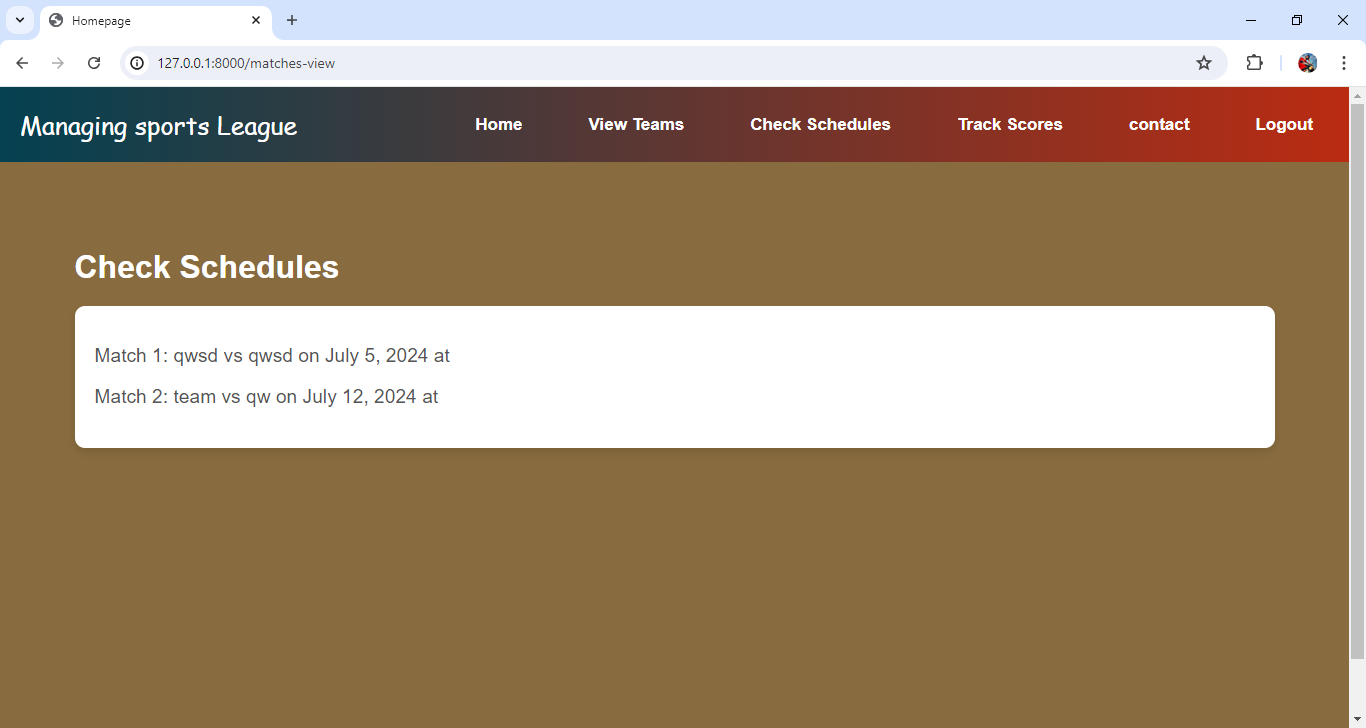


Fig 10: Check match schedules page

The match schedules function in **Managing sports League** retrieves all schedule match objects from the database and renders them in the user actions html template. The data is passed to the template under the context variable matches, allowing users to view a list of all scheduled matches. This facilitates the display of match schedules.Top of FormBottom of Form

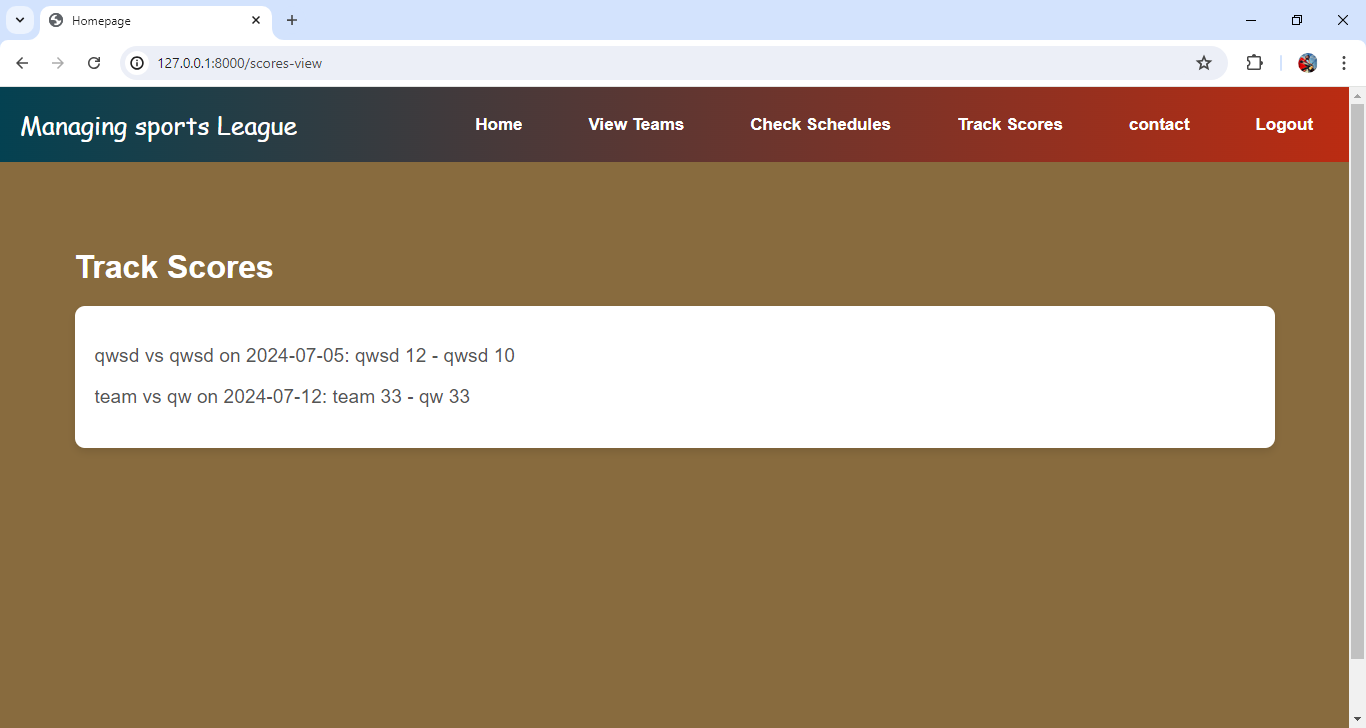


Fig 11: Tracking the updated score of the matches

The Track score function in **Managing sports League** retrieves all score objects from the database and renders them in the user actions html template. The data is passed to the template under the context variable scores, allowing users to view a list of all scores. This enables easy tracking of match scores.

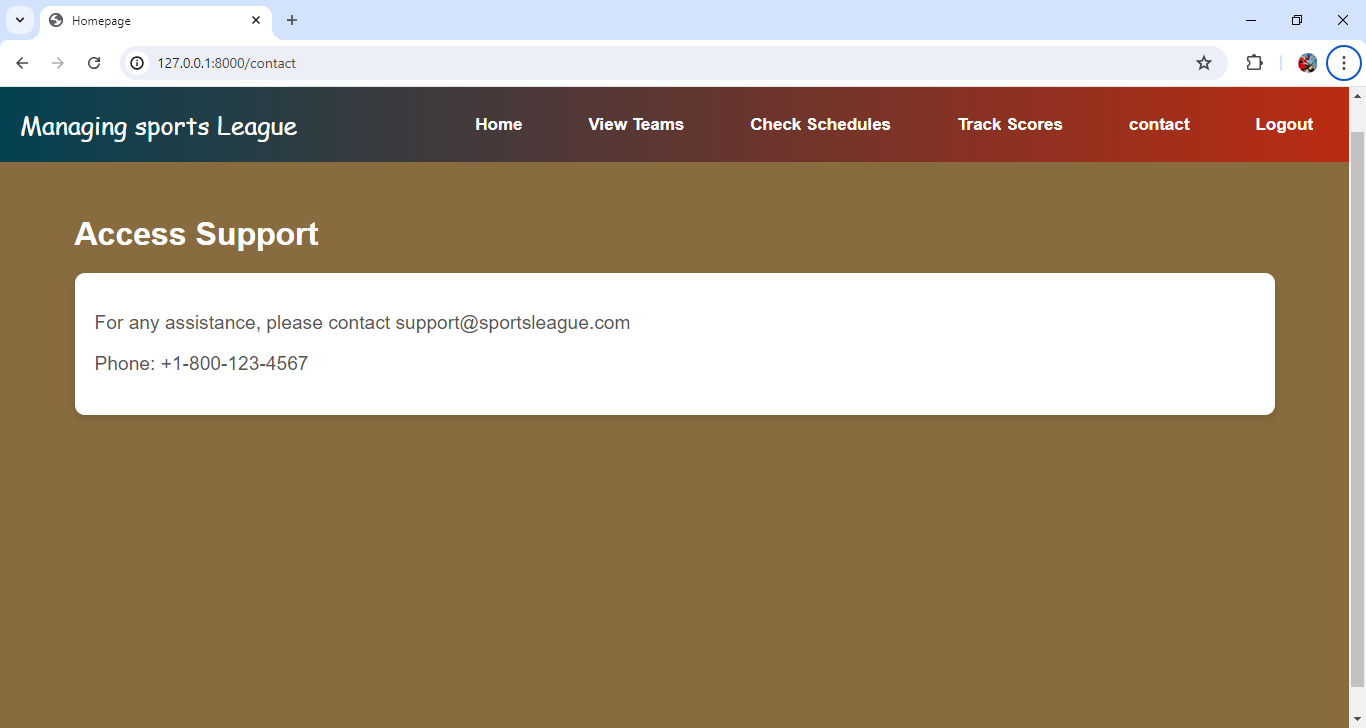


Fig 12: Contact page for user support

The contact view function in **Managing sports League** sets a variable data to True and renders the user actions html template. The variable data is passed to the template under the context variable contact, indicating that the contact information or section should be displayed. This simple function helps manage the visibility of the contact section in the template.