Django Sports League Ranking Table

Thank you for your interest in our company. As part of the hiring process, we would like you to implement a small web application using Django that displays the ranking table for a sports league.

The Problem

Your task is to implement a Django web application that allows the user to upload a CSV file containing the results of games in the sports league and displays the ranking table based on the uploaded data. The CSV file should have the following format: `team_1 name, team_1 score, team_2 name, team_2 score`. The ranking table should be generated based on the sample output below.

In the web application, the user should also be able to add, edit, and delete games from the list through the web interface.

A draw (tie) is worth 1 point and a win is worth 3 points. A loss is worth 0 points. If two or more teams have the same number of points, they should be ranked based on their points, and teams with the same number of points should be ordered alphabetically.

You can expect that the input file will be well-formed. There is no need to add special handling for malformed input files.

Sample Input

```
 Crazy Ones, 3, Rebels, 3
Fantastics, 1, FC Super, 0
Crazy Ones, 1, FC Super, 1
Fantastics, 3, Rebels, 1
Crazy Ones, 4, Misfits, 0
```

Sample Output

Ranking	Team	Points
1	Fantastics	6
2	Crazy Ones	5
3	FC Super	1
4	Rebels	1
5	Misfits	0

Requirements

- Implement the web application using Django.
- Allow the user to upload a CSV file containing the results of the games and display the ranking table based on the uploaded data.
- Allow the user to add, edit, and delete games from the list through the web interface.
- Include unit tests for the Django models and views.
- Document any steps necessary to run the web application and the tests.
- Use the following minimum versions:
  - Python: 3.9
  - Django: 3.2

The Extra Mile

If you have extra time and want to go above and beyond, you can consider implementing authentication and authorization in the web application. This would allow only authenticated users to upload and modify the data.

Evaluation Criteria

We will be looking at the quality and professionalism of your work. In particular, we look for clean, well-designed, maintainable code. Although this is a small task, it should be approached as you would an actual task for a customer.

We would like to see some architecture, but only as much as reasonable. Your code should be easily extensible without burdensome abstractions.

Submission

Please commit your code to a Github repository and provide us with the link to the repository. We would like to see your knowledge of Git and version control.

Platform Support

This will be run in a unix-ish environment (OS X).