Progetto relativo al corso di

“Linguaggi Dinamici”

Interfaccia Web di un campionato di calcio

Realizzato da Mattia Desiderio e Giorgia Gibellini

Panoramica

Il progetto sviluppato consiste in un’interfaccia web per la gestione di un campionato di calcio. Dal lato utente le azioni possibili sono :

- Visualizzazione di tutte le squadre iscritte al campionato tramite il pulsante “Squadre” nel menù situato nell’header
- Per ogni squadra visualizzazione di tutte le partite giocate con relativi risultati cliccando sull’icona della squadra richiesta
- Per ogni squadra visualizzazione di dati statistici come : numero partite giocate, vinte, pareggiate e perse, gol realizzati e subiti e differenza reti (ovvero gol realizzati - gol subiti)
- Per ogni squadra visualizzazione dei punti in classifica, tenendo conto che nel calcio una vittoria corrisponde a 3 punti in classifica, un pareggio a 1 punto in classifica e una sconfitta non aggiunge punti in classifica
- Visualizzazione di tutte le partite esistenti nel campionato rappresentate con : numero giornata, data, squadra locale e squadra ospite tramite il pulsante “Calendario” nel menù situato nell’header

- Visualizzazione di tutte le partite, e relativi risultati, appartenenti a una giornata selezionata tramite il pulsante “Risultati” nel menù situato nell’header e poi cliccando sulla giornata richiesta
- Visualizzazione di tutte le schedine relative a una singola giornata di campionato tramite il pulsante “Risultati” nel menù situato nell’header cliccando sulla giornata richiesta e selezionando il pulsante “Mostra schedina” in alto a destra. Per schedina si intende il valore 1, 2 o X a seconda che abbia vinto la squadra locale, ospite o che sia avvenuto un pareggio
- Visualizzazione della classifica del campionato tramite il pulsante “Classifica” nel menù situato nell’header

Dal lato amministratore le azioni possibili sono :

- Aggiungere una squadra
- Eliminare una squadra
- Modificare una squadra
- Aggiungere nuovi risultati
- Eliminare risultati
- Modificare risultati

Annotazioni : L’eliminazione di una squadra dal campionato implica l’eliminazione di tutte le partite in cui essa è presente e di conseguenza l’annullamento dei punti in classifica e gol che ne derivano per le altre squadre coinvolte

Utilizzo del software

- Avvio dell’applicazione

  Aprire il terminale dentro la cartella del progetto e scrivere il seguente comando “python manage.py runserver” quindi copiare e incollare nel proprio browser web il link che compare sul terminale. Non serve effettuare ulteriori operazioni in quanto è già stato precaricato un campionato

- Caricamento di dati

  Il caricamento dei dati nel database può essere fatto sia manualmente, dal lato admin dell'interfaccia web, sia tramite file json. Un file json debitamente strutturato può essere caricato nel database tramite il comando da terminale "python manage.py loaddata nome_file.json"

- Modifiche alla struttura del database

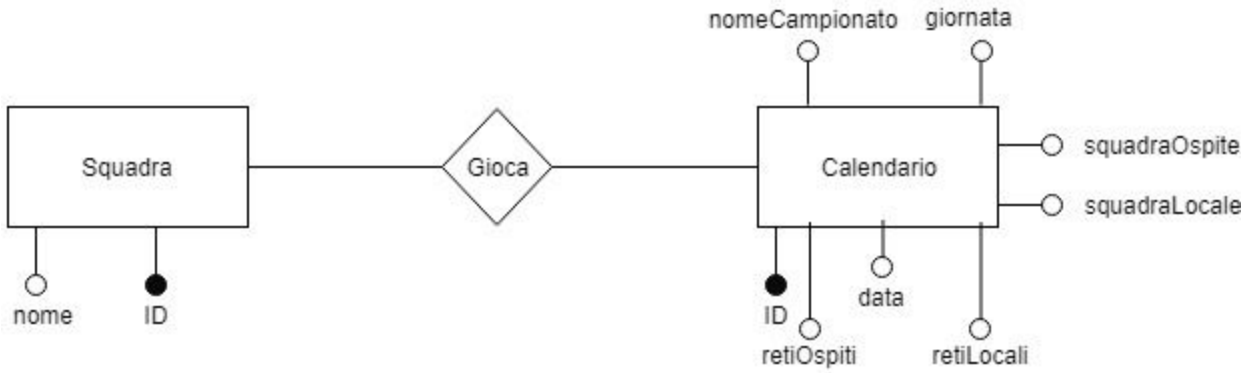
  Nel caso vengano effettuate modifiche alla struttura del database dal file models.py è necessario aggiornare la cartella migrations del progetto tramite il comando da terminale "python manage.py makemigrations nome_package"

Specifiche tecniche

Il progetto è stato realizzato tramite il framework Django, il query language adottato è SQLite3 e il database è locale.

Come database pre-esistente sono stati considerati i due file json "2015-16.json" e "2016-17-json" presenti su GitHub al seguente link "https://github.com/Linguaggi-Dinamici-2018-Modulo-Python/Jupyter-notebooks".
I file sono stati modificati nella struttura per poter essere caricati sul database tramite il comando da terminale "python manage.py loaddata nome_file". Nei file inoltre erano presenti incongruenze nella trascrizione dei risultati che sono stati corretti per rispecchiare i dati reali dei campionati di calcio svoltosi negli anni 2015-16 e 2016-17.

Modello ER



La struttura delle tabelle nel database rispecchia il modello ER rappresentato. Abbiamo usato come primary key per entrambe le entità gli identificatori auto-incrementali forniti di default.

L'entità squadra oltre al suddetto ID che è anche PK ha come unico altro attributo il nome della squadra.

L'entità calendario invece è più complessa e ogni sua istanza rappresenta una partita del campionato. Gli attributi di calendario sono oltre a ID che è anche PK, la data di svolgimento del match, la giornata di campionato all'interno della quale si svolge la partita, il nome del campionato in corso, il numero di gol realizzati dalla squadra locale e dalla squadra ospite e i nomi delle suddette squadre. Abbiamo considerato "squadraOspite" e "squadraLocale" come FK per collegare le tabelle "squadra" e "calendario" ed evitare incongruenze nei dati.

Implementazione delle funzioni principali

- *schedina*, funzione presente nel file models.py, permette di calcolare la schedina relativa a un match ovvero restituisce 1, 2 o X a seconda della vittoria della squadra di casa, ospite o di un pareggio
- *statistics*, funzione presente nel file views.py, ritorna il numero di partite vinte, perse, pareggiate e il numero di punti in classifica di una determinata squadra, inoltre ritorna

anche il numero di gol realizzati, subiti e la differenza reti. Come parametro in ingresso viene passato l'id della squadra di cui si desiderano le statistiche.

In "partite_squadra" vengono salvate tutte le partite in cui la squadra compare o come "squadraLocale" o come "squadraOspite", ciò è possibile tramite la funzione OR implementata in Django dalla libreria Q. Dopo di chè si analizza una partita alla volta tramite il for, se l'id passato come parametro alla funzione statistics corrisponde all'id ospite di quella partita e il numero di gol degli ospiti è maggiore di quello dei locali viene assegnata la vittoria, stessa cosa nel caso contrario. Lo stesso procedimento viene applicato per le partite perse e pareggiate. Il calcolo dei punti in classifica viene fatto direttamente nella return moltiplicando le partite vinte per tre e sommando il numero di partite pareggiate.

Il conto dei gol realizzati e subiti avviene in due step, prima si contano i gol realizzati/subiti in casa poi quelli in trasferta e successivamente vengono sommati. I gol realizzati in casa vengono calcolati prima filtrando tutte le partite in cui la squadraLocale ha l'id richiesto e poi facendo una sum sulle retiLocali, il risultato di questa operazione viene salvato in "goal_fatti_casa". Lo stesso procedimento viene ripetuto per gol realizzati in trasferta e gol subiti.

Quando si passano i parametri alla render però non è possibile usare "goal_fatti_casa" in quanto è un tipo di dato particolare e per accedere al suo valore numerico è necessario scrivere : goal_fatti_casa["retiLocali__sum"]

- *calendar*, funzione presente nel file views.py, permette la visualizzazione di tutte le partite del campionato ordinate per data
- *risultati,* funzione presente nel file views.py, permette la visualizzazione dei nomi di tutte le giornate presenti nel campionato senza ripetizioni
- *risultati_giornata,* funzione presente nel file views.py, permette la visualizzazione di tutte le partite del campionato relative ad una certa giornata passata come parametro d'ingresso alla funzione

- *schedina*, funzione presente nel file views.py, permette la visualizzazione di tutte le schedine delle partite del campionato relative a una certa giornata specificata come parametro d'ingresso
- *classifica*, funzione presente nel file views.py, permette di calcolare dinamicamente la classifica attuale del campionato. Il funzionamento è simile a quello della funzione statistics ma anziché restituire con una return i punti in classifica essi vengono salvati nel dizionario test che ha come chiavi i nomi delle squadre e come valori i punti in classifica relativi ad ognuna. Il dizionario viene poi ordinato in maniera decrescente nella return così da fornire la classifica ordinata dalla squadra con maggior numero di punti a quella con minor numero di punti