

---

# EUROZONE

---

## Progetto Basi di Dati II

### **Autori**

Giuseppe Pio Sorrentino - 0522501965 - g.sorrentino101@studenti.unisa.it

Pasquale Sorrentino - 0522501954 - p.sorrentino47@studenti.unisa.it

Vito Piegari - 0522502037 - v.piegari2@studenti.unisa.it

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Preprocessing dei dati</b>	<b>3</b>
2.1	Struttura del Dataset . . . . .	3
2.2	Match . . . . .	4
2.3	Giocatori . . . . .	5
2.4	Allenatori . . . . .	6
<b>3</b>	<b>Architettura Back-End</b>	<b>6</b>
3.1	Database NoSQL: MongoDB . . . . .	6
3.2	Server-Side Logic: Python e Flask . . . . .	7
3.3	Struttura Modulare del Progetto . . . . .	7
3.4	Gestione degli Eventi Dinamici . . . . .	8
3.5	Funzionalità Derivate e Aggregazioni . . . . .	8
3.6	Sicurezza e Robustezza . . . . .	8
<b>4</b>	<b>Architettura Front-End</b>	<b>9</b>
4.1	Interfaccia utente . . . . .	9
4.1.1	Homepage . . . . .	9
4.1.2	Dettagli Competizione . . . . .	10
4.1.3	Dettaglio partita . . . . .	11
4.1.4	Dettaglio Giocatore . . . . .	14
4.1.5	Dettaglio Allenatore . . . . .	15

# 1 Introduzione

Il presente documento illustra nel dettaglio le scelte progettuali, tecnologiche e architetture adottate nello sviluppo della piattaforma *extbfEurozone*, un'applicazione web interattiva finalizzata alla *extbf*gestione, consultazione e analisi dei dati storici relativi al Campionato Europeo di calcio (UEFA Euro).

L'obiettivo principale del progetto è quello di offrire un sistema strutturato, flessibile e scalabile che consenta all'utente finale di accedere, in maniera intuitiva e interattiva, alle informazioni sulle partite, sui giocatori, sugli allenatori e sugli eventi significativi accaduti durante le varie edizioni del torneo, dal 1960 fino ai giorni nostri.

Il progetto si articola in tre componenti fondamentali:

- *extbfPreprocessing* dei dati: individuazione, analisi e pulizia di un dataset pubblico contenente informazioni eterogenee riguardanti le competizioni europee;
- *extbfBack-end*: realizzazione di una infrastruttura software basata su *extitFlask* e *extitMongoDB Atlas*, che si occupa della gestione e interrogazione dei dati;
- *extbfFront-end*: sviluppo di un'interfaccia web orientata all'utente, volta alla visualizzazione delle informazioni in maniera chiara, dinamica e navigabile.

L'intero sistema è stato concepito per garantire una *extbf*modularità elevata, promuovere il riuso del codice, e facilitare eventuali evoluzioni future del progetto, come l'integrazione con API esterne o l'estensione a ulteriori competizioni calcistiche.

## 2 Preprocessing dei dati

Per alimentare l'applicazione *Eurozone*, si è fatto uso di un dataset open-source reperito sulla piattaforma **Kaggle**, intitolato *"Football / Soccer: UEFA Euro 1960–2024"*.

Il dataset, fornito in formato **CSV**, contiene informazioni dettagliate su tutte le edizioni degli Europei: partite, squadre, risultati, eventi (goal, sostituzioni, sanzioni), allenatori e formazioni.

Il preprocessing ha comportato un attento lavoro di esplorazione, *extbf*pulizia e *extbf*selezione delle feature più rilevanti, al fine di garantire una corretta coerenza semantica tra i dati e il modello logico dell'applicazione.

### 2.1 Struttura del Dataset

Il dataset originale è articolato in quattro file principali:

- `match.csv`: informazioni dettagliate relative alle singole partite;
- `coaches.csv`: elenco degli allenatori associati ai match;
- `players.csv`: anagrafiche e presenze dei calciatori.

## 2.2 Match

I dati delle partite coprono tutte le edizioni dell'Europeo disputate, dal 1960 al 2024.

Ogni partita è rappresentata da una riga nel file `match.csv`, che presenta inizialmente 52 colonne, molte delle quali ridondanti o contenenti informazioni scarsamente rilevanti.

Le colonne ritenute essenziali includono:

- **Id match**: identificatore numerico univoco della partita;
- **Team e Team code**: nomi delle squadre partecipanti e relative abbreviazioni, utili per l'associazione ai loghi;
- **Score e Score total**: goal segnati nei 90 minuti regolamentari e complessivi, comprensivi di supplementari e rigori;
- **Penalty**: relativo ai falli commessi dalle due squadre;
- **Winner e Winner\_reason**: squadra vincitrice e causa del risultato finale (tempo regolamentare, supplementari, rigori);
- **Year e Date**: anno e data ISO 8601 della partita;
- **Stadio, città, capienza e manto**: informazioni logistiche essenziali sull'impianto ospitante;
- **Goals, Events, Lineups**: liste strutturate di oggetti JSON che descrivono goal, eventi significativi e formazioni.
- **Coaches**: nomi degli allenatori per squadra di casa e squadra ospite, con eventuale staff di supporto (vice, preparatori);
- **Penalties\_missed**: elenco dei rigori falliti da entrambe le squadre, con giocatore, minuto e fase di gioco;
- **Penalties**: tutti i rigori assegnati durante l'incontro (segnati e sbagliati), con indicazione di esecutore, portiere, risultato e minuto;
- **Red\_cards**: sta dei cartellini rossi emessi, riportando giocatore, minuto e fase della gara.

Sono state rimosse colonne non essenziali come: `game_referees`, `condition_humidity`, `stadium_sponsor`, in quanto considerate marginali rispetto agli obiettivi informativi della piattaforma.

Abbiamo inoltre filtrato delle colonne per evitare al più possibile informazioni superflue.

- **Goal** : salva solo il nome del giocatore, il minuto del goal e il tipo;
- **Lineup**: Memorizzando solo l'id del giocatore e la posizione di partenza;
- **Coaches**: Salvando solo il nome dell'allenatore e infine i cartellini rossi e i rigori.

## 2.3 Giocatori

La tabella dei giocatori fornisce informazioni su ogni calciatore convocato per le edizioni europee.

Dopo un'opportuna pulizia, le colonne mantenute sono:

- **country\_code:** codice della nazione di appartenenza del giocatore, usato come chiave per associare la bandiera corrispondente.
- **Name:** nome completo del calciatore (nome e cognome).
- **Name\_shirt:** nome riportato sulla maglia in quella partita; circa l'8 % di valori nulli, in cui si è utilizzato *Name* come valore sostitutivo.
- **jersey\_number:** numero di maglia indossato dal giocatore in quella gara.
- **position\_national:** ruolo generico nella nazionale (defender, midfielder, striker, goal-keeper).
- **position\_field:** ruolo generico nel posizionamento di campo (stessa codifica della precedente colonna).
- **position\_field\_detailed:** ruolo specifico (es. ala); colonna scartata per l'elevata presenza di nulli (26 %) e valori non validi (UNKNOWN, 27 %).
- **country\_birth:** codice della nazione di nascita del calciatore; eliminata perché contiene il 57 % di valori nulli.
- **birth\_date:** data di nascita del giocatore in formato YYYY-MM-DD.
- **id\_match:** identificativo univoco della partita a cui si riferisce la riga.
- **id\_player:** identificativo univoco del calciatore (costante su match diversi).
- **Start:** indica se il giocatore è partito **field** (titolare) o **bench** (panchina).
- **Year:** edizione dell'Europeo in cui ricade la presenza.
- **start\_position\_x:** coordinata X di posizionamento iniziale; colonna scartata (49 % di nulli).
- **start\_position\_y:** coordinata Y di posizionamento iniziale; colonna scartata (49 % di nulli).
- **Height:** altezza del giocatore; eliminata per il 48 % di valori mancanti.
- **id\_club:** identificativo del club di appartenenza; mantenuto nonostante il 44 % di nulli (sanificati con -1).
- **weight:** peso corporeo del calciatore; scartato a causa del 50 % di valori nulli.
- **id\_national\_team:** identificatore della nazionale (ridondante rispetto a *country\_code*).

## 2.4 Allenatori

Per ciascun match sono associati uno o più allenatori per ogni squadra.

I dati mantenuti sono:

- **country\_code:** codice della nazione di provenienza dell'allenatore, usato per associare la bandiera corrispondente.
- **Country:** nome esteso della nazione di appartenenza dell'allenatore.
- **Name:** nome completo dell'allenatore (nome e cognome).
- **id\_match:** identificativo univoco della partita alla quale è associata la presenza dell'allenatore.
- **Year:** edizione dell'Europeo in cui è registrata la partecipazione dell'allenatore.

Il preprocessing dei dati si conclude con l'importazione delle entità nel database MongoDB, tramite uno script ETL centralizzato (**Extract, Transform, Load**), conforme alla struttura logica del modello dati descritto nella sezione seguente.

## 3 Architettura Back-End

L'architettura back-end della piattaforma **Eurozone** costituisce il nucleo logico dell'intero sistema.

Essa si occupa della gestione dei dati, della loro manipolazione e della loro esposizione attraverso un insieme di API RESTful, che rappresentano l'interfaccia tra il server e il front-end.

Il back-end è progettato secondo il paradigma *modulare* e *manutenibile*, con separazione esplicita delle responsabilità tra livelli applicativi.

### 3.1 Database NoSQL: MongoDB

Per la persistenza dei dati è stato adottato **MongoDB**.

Tale scelta è stata motivata da caratteristiche architetturali ben allineate con la natura dei dati da trattare:

- **Modello flessibile a documenti:** l'eterogeneità delle informazioni relative a partite, giocatori, eventi e allenatori trova una rappresentazione naturale nel formato BSON (Binary JSON), che consente l'inserimento di documenti nidificati, array e campi opzionali;
- **Scalabilità orizzontale:** la piattaforma supporta meccanismi di *sharding* e replica, rendendola adatta a una futura estensione ad applicazioni con grandi volumi di accessi simultanei;
- **Supporto nativo a query complesse:** MongoDB permette operazioni di *aggregation*, *lookup*, e *pipeline* su strutture annidate, fondamentali per l'analisi di dati sportivi complessi.

Il database logico, denominato **EurozoneDB**, è articolato in tre **collezioni** principali:

- **matches**: contiene tutti i metadati di ogni partita, incluse formazioni, eventi cronologici, dati anagrafici e tecnici;
- **players**: anagrafica dei calciatori, con informazioni di carriera aggregata e metadati associati alle presenze;
- **coaches**: informazioni sugli allenatori e sulla loro partecipazione alle varie edizioni del torneo.

### 3.2 Server-Side Logic: Python e Flask

La logica server-side è implementata in **Python**, un linguaggio largamente impiegato nel contesto scientifico e data-driven, in combinazione con **Flask**, micro-framework particolarmente indicato per applicazioni web RESTful leggere e modulari.

Il suddetto framework è stato scelto per la sua semplicità, estendibilità e per la disponibilità di un ricco ecosistema di librerie.

**Flask** agisce da orchestratore delle rotte, offrendo un meccanismo semplice per l'esposizione di endpoint HTTP.

Ciascun endpoint è associato a una funzione Python che esegue operazioni di interrogazione, aggiornamento o inserimento dati tramite un'interfaccia dedicata al *Data Access Layer*.

### 3.3 Struttura Modulare del Progetto

L'organizzazione logica del progetto segue un'architettura stratificata, ispirata al principio di *Separation of Concerns*, al fine di garantire **testabilità**, **estensibilità** e **riutilizzo** del codice.

Il progetto è strutturato secondo le seguenti componenti:

- **app.py**: **punto di ingresso** dell'applicazione. Si occupa della creazione dell'istanza Flask, della configurazione dell'URI di connessione al cluster MongoDB Atlas, della registrazione delle rotte REST e della gestione della sessione;
- **model/**: contiene le **classi dominio** che modellano entità come **Match**, **Player**, **Goal**, **Red\_Card**, ciascuna dotata di costruttore, metodi di serializzazione/deserializzazione (**from\_dict()** e **to\_dict()**) e controlli di consistenza. Queste classi costituiscono l'astrazione logica dei documenti presenti in database.
- **repository/**: implementa il **pattern Repository**, incaricato dell'accesso ai dati. Questa astrazione consente di isolare la logica di accesso al database dalla logica applicativa. Sono presenti:
  - **MatchRepository**: interroga la collezione **matches**, esponendo metodi per ricerca per anno o ID, inserimento di eventi (goal, rigori, corner), e aggregazioni statistiche;
  - **PlayerRepository**: interroga la collezione **players**, permettendo la ricerca per ID o nome, e il calcolo di presenze, assist, goal e corner battuti;

- **CoachRepository**: interfaccia per estrarre informazioni anagrafiche e storiche degli allenatori.
- **data/**: contiene gli script di **pre-elaborazione e caricamento dati** (ETL). In particolare, `load_data.py` si occupa di caricare nel database i documenti puliti e convertiti da sorgenti CSV in formato JSON, validando gli schemi e le dipendenze tra le entità.

### 3.4 Gestione degli Eventi Dinamici

Una delle componenti più dinamiche dell'applicazione è la gestione in tempo reale degli eventi di partita (goal, cartellini, sostituzioni, calci d'angolo).

Questo è reso possibile da:

- **Inserimento dinamico**: tramite endpoint POST su `/add_event`, l'utente può aggiungere un evento. Il server effettua controlli su ID validi, correttezza temporale, tipo di evento e validazione semantica, per poi persistere l'oggetto nella lista eventi del match corrispondente;
- **Cancellazione**: tramite endpoint POST su `/match/{match_id}/delete_event/{event_id}`, è possibile rimuovere eventi errati. La rimozione aggiorna immediatamente la visualizzazione front-end attraverso ricaricamento asincrono o redirect alla schermata di dettaglio;
- **Ordinamento temporale**: nella pagina di dettaglio partita, gli eventi sono ordinati cronologicamente (con opzione di reverse order) utilizzando i campi `time_minute` e `time_second`, con fallback su valori nulli.

### 3.5 Funzionalità Derivate e Aggregazioni

Il back-end è responsabile anche del calcolo di statistiche aggregate, tra cui:

- Conteggio dei goal, assist e corner per ciascun giocatore;
- Calcolo del numero di presenze e presenze da titolare;
- Estrazione delle partite giocate e allenate per ogni edizione;
- Identificazione delle edizioni europee a cui un dato player o coach ha partecipato.

Tali operazioni si appoggiano ai meccanismi di aggregazione di MongoDB o a funzioni Python interne nei repository.

### 3.6 Sicurezza e Robustezza

Il back-end implementa alcune funzionalità minime di robustezza:

- **Gestione degli errori**: tramite blocchi `try-except` e codici di risposta HTTP coerenti, viene garantita la gestione ordinata di fallimenti di inserimento, ricerche su ID inesistenti o errori di validazione semantica;



- **Sanificazione dell'input:** tutti i campi testuali o numerici ricevuti dalle chiamate HTTP sono validati, convertiti ai rispettivi tipi e verificati in termini di coerenza logica;
- **Isolamento della logica di accesso ai dati:** ogni accesso a MongoDB è delegato al livello `repository`, evitando chiamate dirette all'interno delle funzioni controller.

## 4 Architettura Front-End

L'interfaccia di front-end è progettata come una `extbfSingle Page Application (SPA)`, sviluppata per offrire un'esperienza utente fluida, reattiva e moderna.

Questa architettura prevede che la logica di presentazione e interazione sia interamente gestita dal client (browser), che comunica con il back-end tramite chiamate HTTP asincrone alle API RESTful esposte da Flask.

Il progetto prevede l'utilizzo di Flask anche come motore di rendering per pagine HTML lato server.

L'interfaccia è costruita secondo i principi della responsività, semplicità e chiarezza, mediante l'utilizzo combinato di HTML5, CSS3 e librerie di supporto. Il codice front-end è organizzato in due sottodirectory principali:

- `static/`: Contiene tutte le risorse statiche (come immagini dei loghi e icone);
- `templates/`: Include i file HTML renderizzati da Flask. Ogni template rappresenta una schermata o componente della piattaforma.

### 4.1 Interfaccia utente

L'interfaccia utente della piattaforma **Eurozone** è articolata in cinque schermate principali, progettate per guidare l'utente in modo intuitivo attraverso i dati storici degli Europei di calcio.

Le sezioni seguenti descrivono ciascuna schermata sia dal punto di vista funzionale che grafico, con l'integrazione di screenshot significativi per ciascuna.

#### 4.1.1 Homepage

La schermata principale dell'applicazione, mostrata in Figura 1, rappresenta l'interfaccia iniziale della piattaforma *Eurozone*, dedicata alla consultazione e gestione dei dati storici del Campionato Europeo di calcio UEFA.

Essa fornisce una panoramica visiva e ordinata di tutte le edizioni del torneo, indicizzate cronologicamente in ordine decrescente (dalla più recente alla più remota).

Ogni edizione è rappresentata da una card contenente:

- il logo ufficiale dell'evento;
- l'etichetta testuale con il nome dell'edizione (es. Euro 2024).

Dal punto di vista funzionale, ciascuna card è interattiva: un click reindirizza l'utente alla schermata di dettaglio della relativa competizione, in cui vengono visualizzati i risultati

dei match, divisi per fase del torneo (es. gironi, quarti, semifinale, finale, ecc.). L'interfaccia è progettata secondo i principi dell'usabilità visiva: sfondo neutro, spaziatura uniforme tra le card, utilizzo di colori istituzionali UEFA (come il blu) e immagini ad alta risoluzione per favorire l'immediata riconoscibilità dell'edizione.

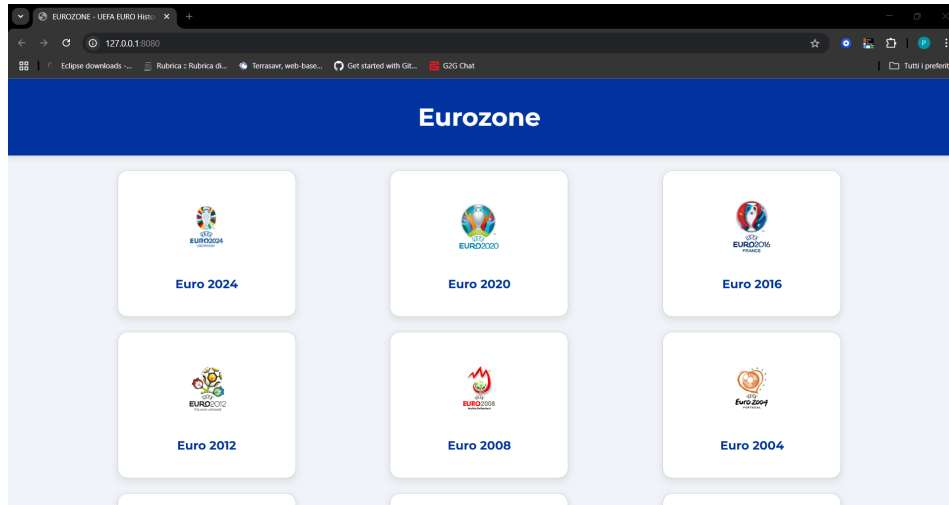


Figure 1: Homepage dell'applicazione *Eurozone*, punto di accesso alle edizioni del Campionato Europeo UEFA.

#### 4.1.2 Dettagli Competizione

La schermata dei dettagli competizione (Figura 2) permette all'utente di visualizzare la struttura e i risultati di una specifica edizione del Campionato Europeo.

I match sono ordinati per fase (es. Quarti di finale, Semifinale, Finale) e all'interno di ciascuna fase vengono elencate le partite con le rispettive squadre e punteggi.

È presente una barra laterale per la **navigazione rapida tra le edizioni**, che consente una fruizione fluida dei contenuti storici.

Il layout, semplice ma informativo, favorisce l'analisi comparativa dei percorsi delle squadre, supportando potenziali studi statistici retrospettivi sul rendimento nazionale.

Risultati Euro 2020			
Partite dell'edizione 2020			
<b>Finale</b>			
Italy (ITA)	(3) 4 - 2 (2)	England (ENG)	
<b>Semifinale</b>			
England (ENG)	1 - 2	Denmark (DEN)	
Italy (ITA)	(4) 5 - 3 (2)	Spain (ESP)	
<b>Quarti di finale</b>			
Ukraine (UKR)	0 - 3	England (ENG)	
Czechia (CZE)	1 - 2	Denmark (DEN)	
Belgium (BEL)	1 - 2	Italy (ITA)	
Switzerland (CHE)	(4) 3 - 3 (3)	Spain (ESP)	

Figure 2: Schermata dedicata alla visualizzazione dettagliata di una competizione europea. I risultati sono suddivisi per fase del torneo e accompagnati dalle bandiere e sigle internazionali delle squadre.

#### 4.1.3 Dettaglio partita

La schermata di dettaglio rappresenta uno dei punti centrali dell'interfaccia utente, consentendo un'esplorazione approfondita di una singola partita all'interno del torneo selezionato. Tale vista è articolata su due tab principali: **Riassunto** e **Formazioni**, e ogni elemento è stato progettato per garantire chiarezza informativa, supporto all'interazione e funzionalità di manipolazione del dato.

Nel tab "Riassunto" (Figura 3), l'utente ha accesso a una cronologia completa degli eventi accaduti durante il match, ordinabili in senso cronologico o inverso.

Ogni evento è rappresentato da un'icona contestuale (goal, cartellino, corner, ecc.) e da una descrizione testuale che include i nomi dei calciatori coinvolti, con possibilità di accedere ai rispettivi profili **cliccando sul nome**.

Una funzionalità rilevante è la **gestione dinamica degli eventi**: l'utente può cliccare su "Aggiungi" per avviare l'inserimento manuale di un evento, tramite un *form guidato* che prevede la selezione del tipo di evento, del calciatore coinvolto, del minuto di gioco e, ove previsto, dell'assist-man (Figura 4).

Alla conferma, l'evento viene salvato nel database e visualizzato nell'interfaccia (Figura 5). Parallelamente, ogni evento è accompagnato da un pulsante "Cancella" che consente la rimozione dell'evento stesso previa conferma da parte dell'utente, tramite una *modale* di dialogo dedicata (Figura 6).

Il secondo tab "Formazioni" (Figura 7) presenta una **visualizzazione interattiva** della disposizione in campo delle due squadre.

I giocatori sono rappresentati come cerchi colorati (rossi per la squadra di casa, blu per la squadra ospite), posizionati in base al ruolo e allo schieramento tattico e cliccabili per accedere alla pagina del singolo giocatore.

Sono inoltre presenti i nomi degli allenatori, anch'essi cliccabili per accedere alla pagina del singolo coach.

L'utente può anche visualizzare la lista completa dei titolari e delle riserve, con il ruolo

assegnato.

Infine, la *sidebar* a sinistra (visibile in tutte le sottoschermate) consente una rapida navigazione tra le altre partite del torneo corrente, garantendo coerenza nel flusso di esplorazione.

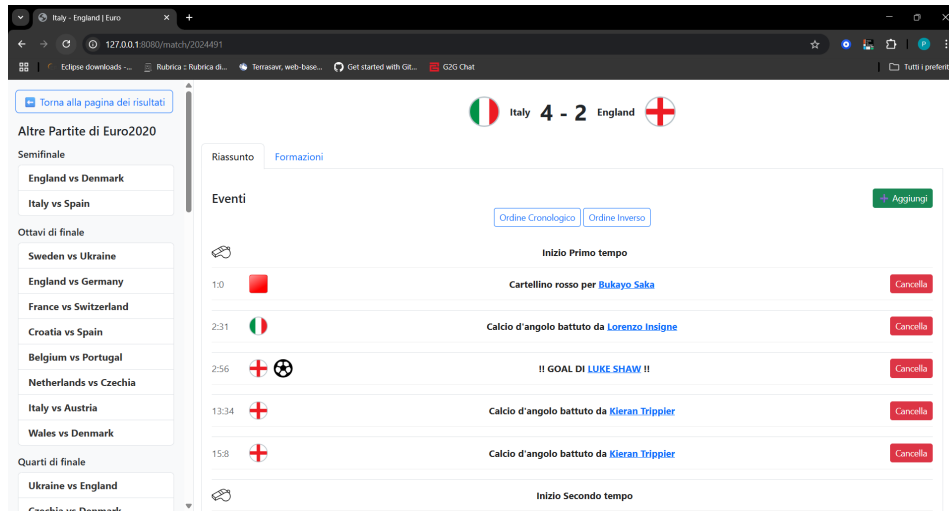


Figure 3: Scheda riassuntiva di una partita, con eventi ordinabili e funzionalità di gestione.

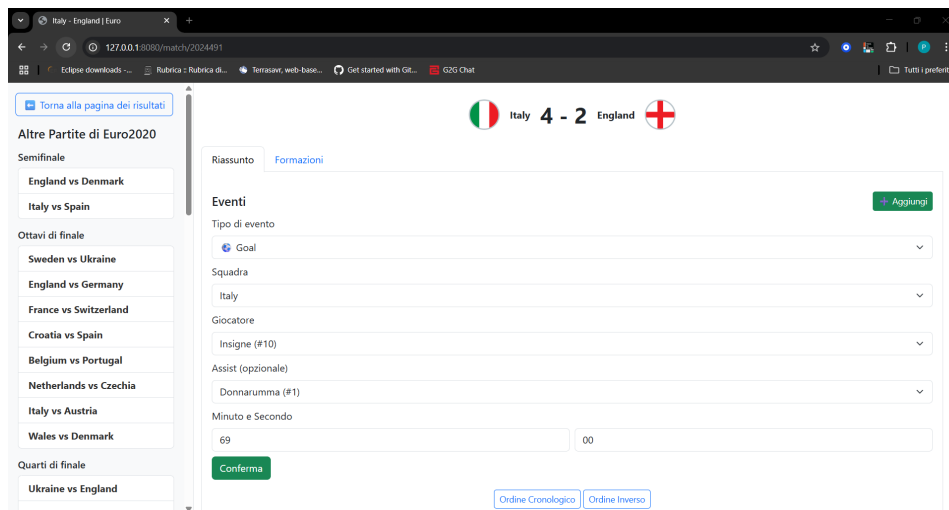


Figure 4: Form di inserimento di un nuovo evento durante la partita.

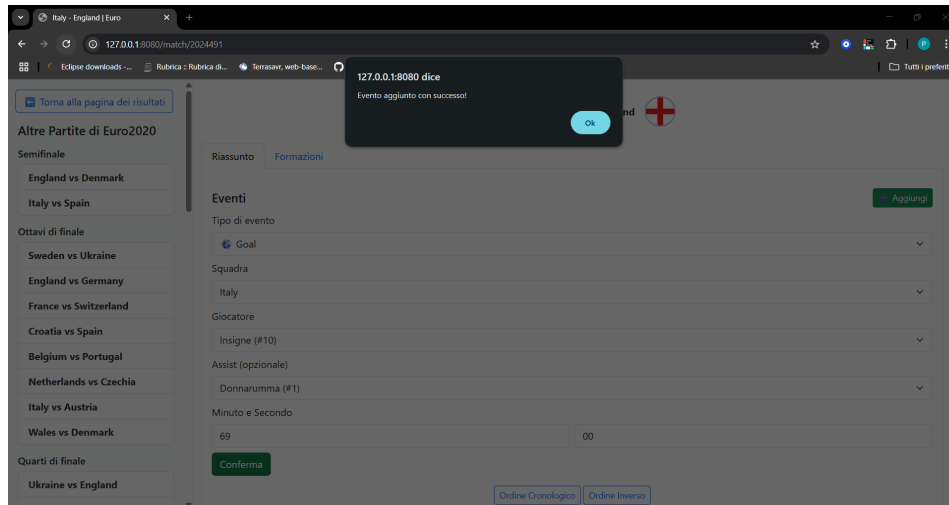


Figure 5: Notifica di conferma per l'inserimento di un evento.

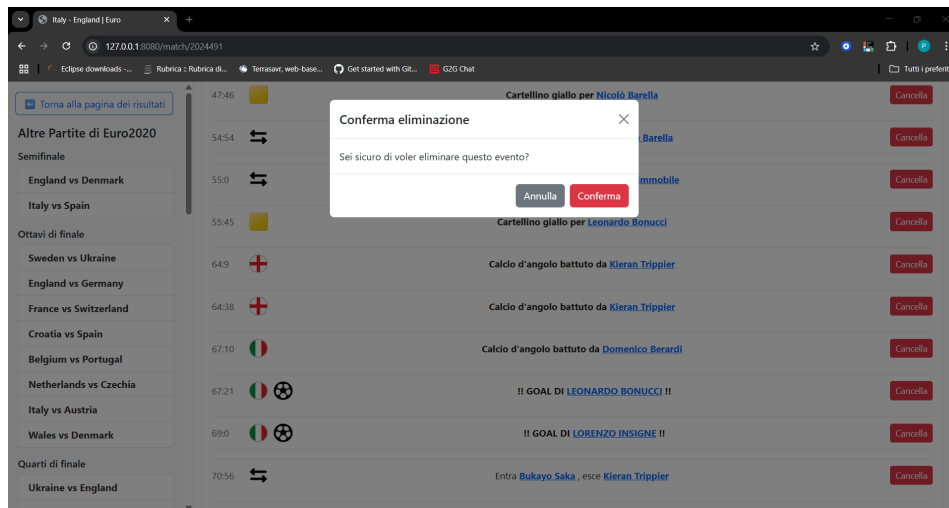


Figure 6: Finestra di conferma per l'eliminazione di un evento.



Figure 7: Visualizzazione interattiva delle formazioni in campo.

#### 4.1.4 Dettaglio Giocatore

La schermata di dettaglio giocatore, esemplificata in Figura 8, offre una vista sintetica ma informativamente completa sul profilo di un calciatore selezionato.

Essa si presenta come una card centrata e ben delimitata, all'interno della quale vengono visualizzati i principali attributi anagrafici e statistici associati al giocatore nel contesto delle fasi finali degli Europei UEFA.

L'interfaccia mostra in primo luogo il **nome** del giocatore e la relativa **bandiera nazionale**, accompagnati dal **ruolo**, **data di nascita** e **numero di maglia**.

Seguono poi le statistiche aggregate calcolate dinamicamente dalla base dati:

- **Presenze totali:** numero complessivo di partite disputate, indipendentemente dalla modalità di ingresso (titolare o subentrato);
- **Presenze da titolare:** conteggio ristretto alle partite in cui il giocatore ha iniziato dal primo minuto;
- **Goal segnati, assist forniti e corner battuti**, se presenti.

Di particolare rilievo è la sezione relativa agli **Europei partecipati**, rappresentati da una lista cliccabile di link che reindirizzano alla schermata dei risultati dell'edizione corrispondente, consentendo all'utente un'esplorazione diretta delle competizioni in cui l'atleta ha preso parte.

Infine, viene fornito un collegamento di navigazione intuitiva che permette di ritornare all'homepage, migliorando l'usabilità generale dell'applicativo.

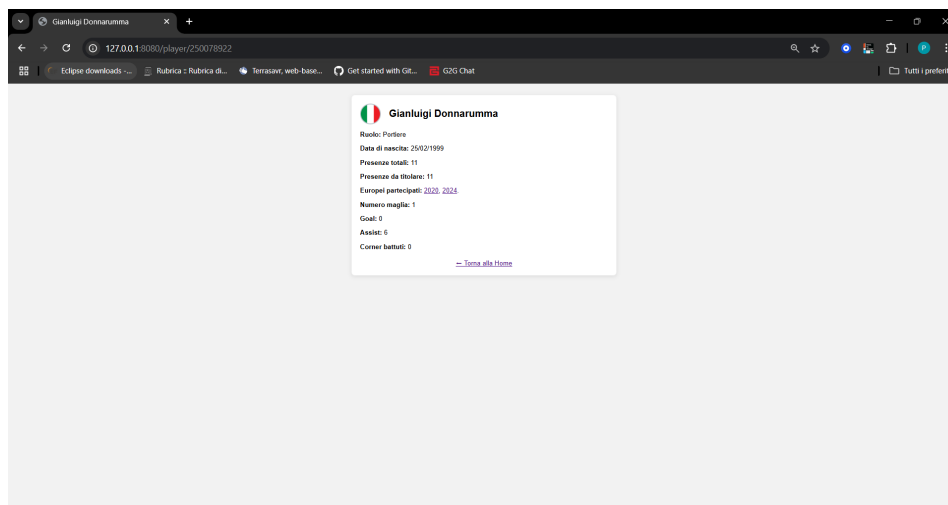


Figure 8: Vista del dettaglio di un calciatore, con dati anagrafici e statistiche aggregate.

#### 4.1.5 Dettaglio Allenatore

Analogamente alla schermata dei giocatori, l'interfaccia di dettaglio allenatore — riportata in Figura 9 — propone un riquadro informativo sintetico e visivamente centrato, che raccoglie le principali informazioni relative al tecnico selezionato.

L'interfaccia si concentra su tre attributi principali:

- La **nazionalità**, rappresentata graficamente tramite bandiera e testualmente con il nome dello Stato;
- Il numero totale di **partite allenate** nei campionati europei presenti nel database;
- L'elenco degli **Europei partecipati**, espresso come collegamenti ipertestuali cliccabili, che rimandano direttamente alle schermate dei risultati dell'edizione corrispondente.

A differenza del profilo giocatore, non sono mostrate statistiche tecnico-tattiche ma si privilegia una visione d'insieme dell'esperienza maturata nella competizione continentale. Anche in questo caso, è presente un collegamento per ritornare rapidamente all'homepage, rafforzando la navigazione bidirezionale dell'interfaccia.

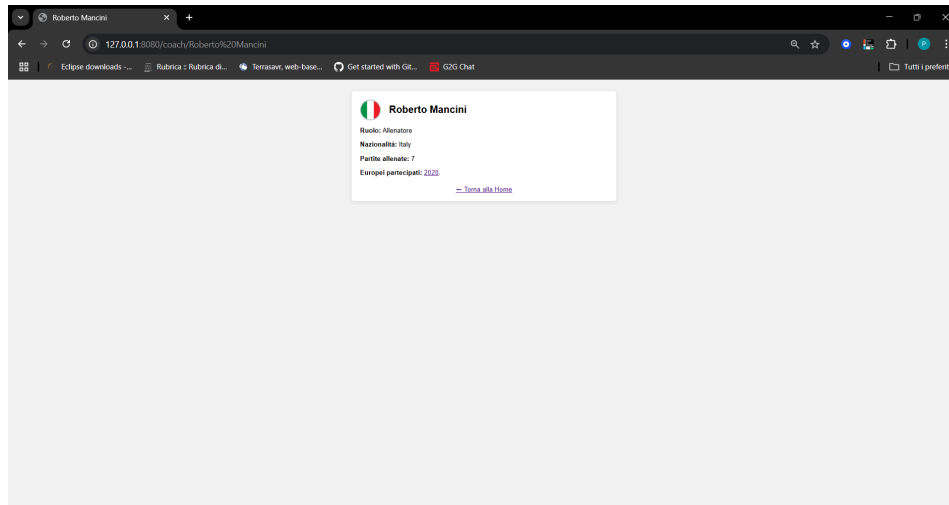


Figure 9: Vista del dettaglio allenatore. Sono mostrate le edizioni degli Europei allenate, la nazionalità e il totale di partite disputate come tecnico.

Queste schermate, combinate con un backend solido e un database ben strutturato, costituiscono lo schema informativo e funzionale della piattaforma Eurozone, offrendo un prodotto coerente, esplorabile e facilmente estendibile a nuove edizioni o nuovi tipi di dati calcistici.