

Documentazione Web APP Fantacalcio Manageriale

21 novembre 2025

Sommario

Questo documento fornisce una panoramica completa dell'architettura tecnica e dello stack tecnologico dell'applicazione web, inclusa la configurazione del backend Flask, lo schema del database Supabase e le procedure di deployment su Render. Lo scopo è supportare la manutenzione, l'onboarding di nuovi sviluppatori e la gestione operativa del sistema.

Indice

1 Architettura del Sistema e Stack Tecnologico	2
1.1 Panoramica dello Stack	2
1.2 Flusso di Dati Chiave	2
2 Schema del Database Supabase	2
2.1 Modelli di Dati	2
2.2 Regole RLS (Row Level Security)	2
3 Riferimento del Backend (Flask Endpoints)	2
3.1 Dettaglio degli Endpoint	2
4 Procedure di Deployment e Operazioni	2
4.1 Setup Locale per Sviluppatori	3
4.1.1 Prerequisiti e Installazione	3
4.1.2 Configurazione Variabili d'Ambiente	3
4.2 Gestione della Piattaforma Render	3
4.2.1 Configurazione del Deployment	3
4.2.2 Variabili d'Ambiente di Produzione e Debugging	3
5 Guida Utente	3
5.1 Accesso e Autenticazione	3
5.2 Funzionalità Principali: La Pagina Mercato	3
6 Changelog	3

1 Architettura del Sistema e Stack Tecnologico

Questa sezione descrive la composizione fondamentale dell'applicazione, specificando il ruolo di ogni componente e la loro interconnessione.

1.1 Panoramica dello Stack

La Web App si basa su una architettura modulare, con una chiara separazione dei compiti tra frontend, backend e servizi di persistenza.

Tabella 1: Panoramica dello Stack Tecnologico

Componente	Ruolo	Tecnologia
Backend	Logica Applicativa	Python / Flask
Frontend	Interfaccia Utente	HTML/Jinja + CSS/Bootstrap
Database	Persistenza Dati	Supabase (PostgreSQL)
Autenticazione	Gestione Utenti	Supabase Auth
Hosting	Deployment Continuo	Render.com

1.2 Flusso di Dati Chiave

Analisi di una transazione critica: l'interazione tra i componenti durante una richiesta autenticata.

2 Schema del Database Supabase

Dettagli tecnici del modello dati e delle politiche di sicurezza implementate.

2.1 Modelli di Dati

Tabella sintetica delle entità principali.

Tabella 2: Schema delle Tabelle Principali

Tabella	Colonna	Tipo	Descrizione
squadre	id	uuid	PK, ID univoco della squadra.
squadre	crediti	int	Crediti attuali disponibili.
scambi	scambio_id	uuid	PK, ID univoco dello scambio.
scambi	stato	text	Stato dello scambio (<code>in_attesa</code> , <code>accettato</code> , <code>annullato</code>).
scambi	squadra_proponente	text	Nome squadra che propone (FK).
scambi	squadra_destinataria	text	Nome squadra destinataria (FK).

2.2 Regole RLS (Row Level Security)

Descrizione di come le politiche di sicurezza su Supabase limitano l'accesso ai dati (e.g., solo l'utente proprietario può leggere/scrivere i propri dati di squadra).

3 Riferimento del Backend (Flask Endpoints)

Elenco dettagliato delle rotte esposte dal server Flask, essenziale per la manutenzione e la comprensione del flusso di controllo.

3.1 Dettaglio degli Endpoint

4 Procedure di Deployment e Operazioni

Istruzioni passo-passo per la gestione dell'infrastruttura di produzione e il setup di sviluppo.

Tabella 3: Mappatura degli Endpoint Flask

Route	Metodo	Endpoint	Funzione	Input (Dati attesi)	Output (Risposta)
Home	GET	/	Visualizza la dashboard principale. Richiede autenticazione.	Nessuno.	Render home.html
Mercato (View)	GET	/mercato/<nome_squadra>	Visualizza tutti i scambi per la squadra.	<nome_squadra> (da URL)	Render mercato.html con dati scambi.
Mercato (Azione)	POST	/mercato/<nome_squadra>	Propone una proposta o accetta/rifiuta quella dello scambio.	<nome_squadra> (da URL)	Redirect all'endpoint GET /mercato/<nome_squadra>
Nuovo Scambio (GET)	GET	/mercato/nuovo/<nome_squadra>	Form per creare una nuova proposta.	Nessuno.	Render nuovo_scambio.html
Nuovo Scambio (POST)	POST	/mercato/nuovo/<nome_squadra>	Invia la nuova proposta di scambio a Supabase.	Campi del form (Giocatori/Crediti Offerti/Richiesti, Destinatario)	Redirect all'endpoint GET /mercato/<nome_squadra>.
Autenticazione (GET)	GET	/login	Visualizza la pagina di login.	Nessuno.	Render login.html
Autenticazione (POST)	POST	/login	Tenta il login tramite Supabase Auth.	email, password (Form data)	Redirect su successo. Render login.html con errore.
Logout	POST	/logout	Esegue il logout utente.	Nessuno.	Redirect su /login.

4.1 Setup Locale per Sviluppatori

4.1.1 Prerequisiti e Installazione

Descrizione dei requisiti software (Python, pip) e dei comandi per clonare il repository, creare l'ambiente virtuale e installare le dipendenze da `requirements.txt`.

4.1.2 Configurazione Variabili d'Ambiente

Dettagli sulla creazione del file `.env` con le chiavi `SUPABASE_URL`, `SUPABASE_KEY`, `SECRET_KEY`, ecc.

4.2 Gestione della Piattaforma Render

4.2.1 Configurazione del Deployment

Dettagli sull'integrazione di Render con GitHub/GitLab, l'uso di un potenziale `Procfile` e il comando di avvio per Flask.

4.2.2 Variabili d'Ambiente di Produzione e Debugging

Procedura per l'inserimento sicuro delle chiavi di Supabase e della `SECRET_KEY` nella dashboard di Render e come accedere ai log per il debugging.

5 Guida Utente

Istruzioni chiare e non tecniche per l'utente finale.

5.1 Accesso e Autenticazione

Come completare la registrazione e il login, gestendo i flussi di Supabase Auth (e.g., reset password).

5.2 Funzionalità Principali: La Pagina Mercato

Descrizione dell'interfaccia `mercato.html`:

- Visualizzazione Scambi Proposti (Attive).
- Visualizzazione Scambi Ricevuti (Gestione: Accetta/Rifiuta).
- Creazione di Nuove Proposte di Scambio.
- Visualizzazione Scambi Conclusi e Annullati.

6 Changelog

Storico delle versioni e delle modifiche significative.