
RELAZIONE PROGETTO TECNOLOGIE WEB

Anno accademico 2022/2023

Portale del campionato
di Formula 1 2023



Membri del gruppo:

Enrico Marras – 152336

Lorenzo Colli – 153063

Funzionalità del progetto

Struttura del portale:

- **Homepage:** contiene un resoconto sull'attuale stato del campionato, le news e i risultati dell'ultima gara disputata.
- **Highlight:** sintesi dei momenti di maggior interesse.
- **Sessioni:** risultati delle sessioni disputate.
- **Circuiti:** dati tecnici sui circuiti del campionato.
- **Scuderie:** resoconto tecnico/storico dei team e del loro attuale stato nel campionato.
- **Store:** vendita dei biglietti delle sessioni.
- **Profilo:** dati sull'utente attualmente loggato.
- **Carrello:** lista d'acquisto per l'utente regolare attualmente loggato.

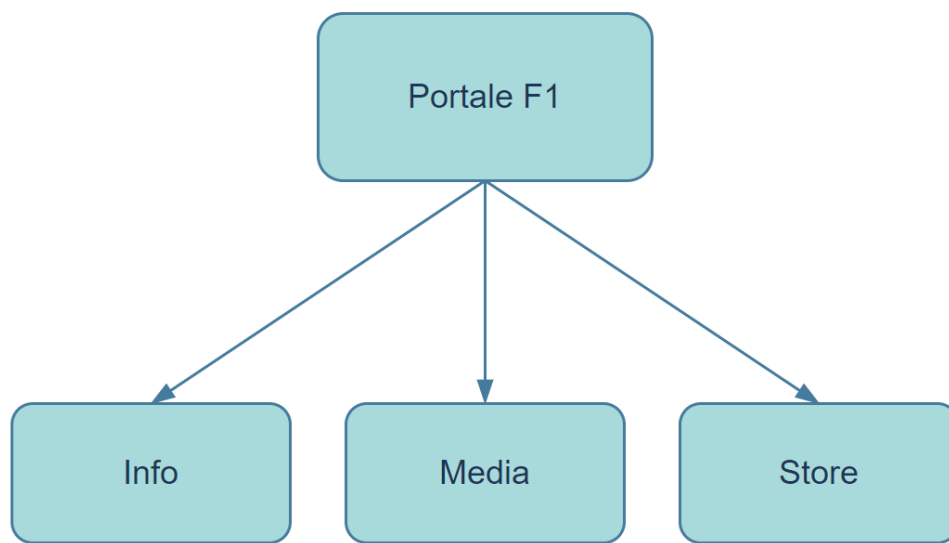
Aspetti d'uso del portale:

- **Tutti i visitatori** al sito possono visualizzare i contenuti multimediali (**Highlight**), lo stato del campionato (**Sessioni**), informazioni su scuderie e circuiti (**Scuderie – Circuiti**) e la disponibilità dei biglietti in vendita (**Store**).
- Ogni utente può registrarsi come **utente regolare** o **gestore di un circuito**.
- Gli **utenti regolari** possono aggiungere e acquistare i biglietti nel carrello, seguire delle particolari scuderie e accedere a una pagina dedicata (**Profilo**) nella quale possono modificare i propri dati, visualizzare i propri ordini e i propri biglietti.
- I **gestori di circuiti** possono inserire nuovi biglietti per assistere alle sessioni all'interno dello Store. Essi hanno accesso a una pagina dedicata (**Profilo**) nella quale possono visionare e cambiare i propri dati, gestire i biglietti, visualizzare lo stato degli ordini e il resoconto delle vendite per settore nell'ultima settimana.
- Il **carrello** permette a un utente regolare di aggiungere ed eventualmente acquistare più biglietti contemporaneamente.
- Il **Recommendation System** permette agli utenti regolari loggati di ricevere news pertinenti basati sulle loro scuderie preferite.

- Un sistema di **notifiche** avverte gli utenti regolari e i gestori dei circuiti sullo stato dell'acquisto dei biglietti.
- Un sistema di **generazione automatica dei biglietti** permette di generare dinamicamente il biglietto dopo l'acquisto.

Fase di progettazione

Al fine di gestire al meglio la fase di sviluppo abbiamo deciso di suddividere il progetto in tre applicazioni:

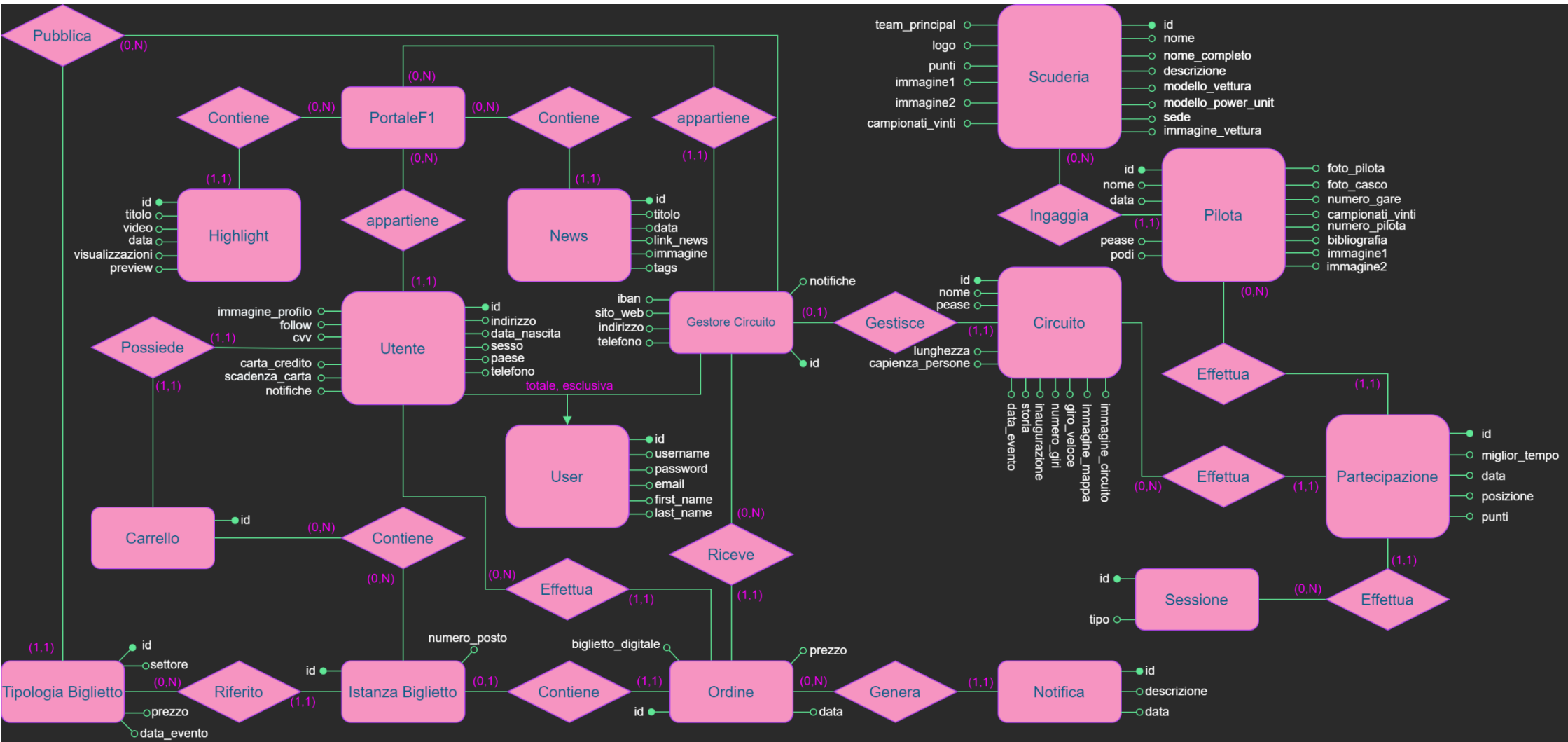


- **Info:** gestisce le parti del portale relative alle scuderie, ai piloti , ai circuiti e a tutti i risultati delle sessioni.
- **Media:** gestisce le parti del portale relative ai contenuti multimediali e l'homepage.
- **Store:** gestisce l'utenza (registrazione, profili e cambiamento dati personali), la pubblicazione di biglietti, il carrello, l'e-commerce e la generazione dei biglietti digitali.

Tecnologie usate: DBMS

- Non abbiamo reputato necessario cambiare il database standard proposto da Django, in quanto SQLite riesce sufficientemente a soddisfare le nostre necessità implementative.
- Ciononostante, per una più facile gestione dei file multimediali, abbiamo deciso di memorizzare il percorso relativo a tali file rispetto alla locazione nel progetto per le rispettive entry del database riferite a tali risorse.

Schema ER



Tecnologie usate: Front-End

Per quanto riguarda il lato front-end, abbiamo fatto uso di Bootstrap (4.5.2) sia integrandolo nei Template, sia nei form di Django tramite l'utilizzo di Django-Crispy-Forms (2.0).

Abbiamo inoltre utilizzato JavaScript per implementare alcune funzionalità interagendo direttamente con il back-end (gestione delle notifiche) e la libreria Chart.js per la gestione grafica delle statistiche.

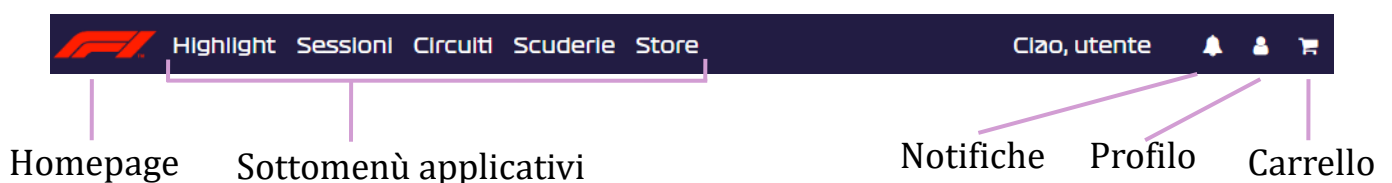
Tecnologie usate: Back-End

Nel back-end non abbiamo avuto la necessità di aggiungere applicazioni extra all'infuori di quelle pre-installate da Django, se non per poche eccezioni come `django_countries` (per l'aggiunta di ChoiceField contenenti tutti gli Stati).

Gestione Template

Al fine di evitare il più possibile riutilizzo di codice nei template è stata adottata una struttura gerarchica, dove **base.html** è il template esteso da tutti gli altri. Esso definisce il tipo di file (html), include le dipendenze, definisce lo spazio per l'estensione di questo template da parte di altri `{% block content %}`, e predispone tutti gli elementi che dovranno sempre essere mostrati in ogni singola pagina del portale:

- **Wallpaper**: sfondo renderizzato nel background con ridimensionamento adattivo basato sulla dimensione della finestra del browser (**wallpaper.js**).
- **Navbar**: barra interattiva che permette la navigazione all'interno delle pagine del portale.

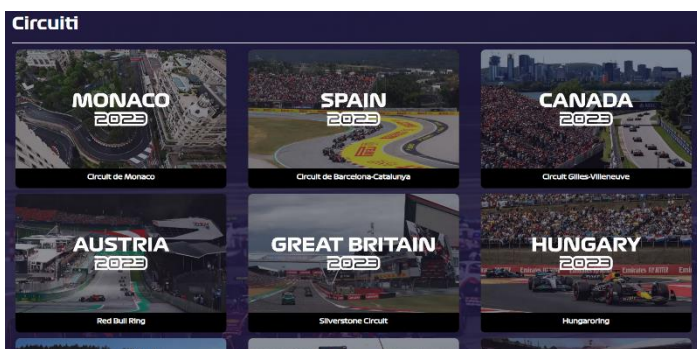


Applicazione Info

Per tutte le view dell'applicazione Info abbiamo scelto di seguire un pattern del tipo **menu di selezione** ⇒ **preciso elemento selezionato**, in quanto è sempre necessario mostrare molte scelte di navigazione all'utente.

I **circuiti** sono gestite da:

- **CircuitiView**: raggruppa in una pagina tutti i circuiti della stagione corrente.
- **CircuitoView**: fornisce tutte le specifiche informazioni riguardanti un preciso circuito.



CircuitiView - menu_circuito.html



CircuitoView - circuito.html

Le **sessioni** sono gestite da:

- **SessioniView**: elenca brevemente il risultato tutte le sessioni che si sono svolte nella stagione.
- **RisultatoSessioneView**: fornisce indicazioni sul risultato dei piloti che hanno partecipato a quella precisa sessione. Permette inoltre, per ogni evento, di selezionare la precisa sessione da visualizzare.



Sessioni				
GRAN PREMIO	DATA	VINCITORE	SCUDERIA	GIRI
Hungaroring	July 23, 2023	Max Verstappen	Red Bull	70
Silverstone Circuit	July 9, 2023	Max Verstappen	Red Bull	52
Red Bull Ring	July 2, 2023	Max Verstappen	Red Bull	71
Circuit Gilles-Villeneuve	June 18, 2023	Max Verstappen	Red Bull	70
Circuit de Barcelona-Catalunya	June 4, 2023	Max Verstappen	Red Bull	66
Circuit de Monaco	May 28, 2023	Max Verstappen	Red Bull	78

SessioniView - sessioni.html

Risultati gara - Circuit de Barcelona-Catalunya				
POSIZIONE	PILOTA	SCUDERIA	PUNTI	GIRO VELOCE
1	Max Verstappen	Red Bull	26	1:16.330
2	Lewis Hamilton	Mercedes	18	1:16.676

RisultatoSessioneView - risultato_sessione.html

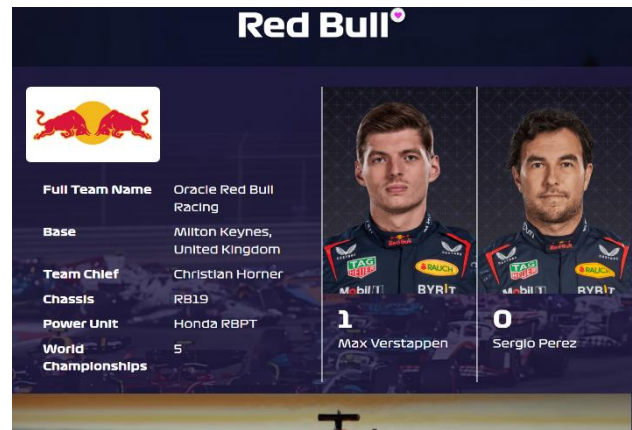
Le **scuderie** sono gestite da:

- **ScuderieView**: elenca tutte le scuderie partecipanti alla stagione corrente ordinate per numero di punti ottenuti.

- **TeamView**: fornisce informazioni riguardanti la scuderia selezionata.



ScuderieView - [scuderia.html](#)



TeamView - [team.html](#)

- **PilotaView**: similmente alla view della singola scuderia, fornisce informazioni riguardanti il pilota selezionato.

Seguire una scuderia

Se un utente regolare è loggato avrà la possibilità di seguire una determinata scuderia tramite un apposito pulsante.



Questo permette all'utente di ricevere in homepage articoli pertinenti basati sui propri interessi (vedasi Recommendation System↴).

A livello implementativo, è stata definita una view **follow** che aggiunge la scuderia in questione all'elenco delle scuderie seguite dall'utente.

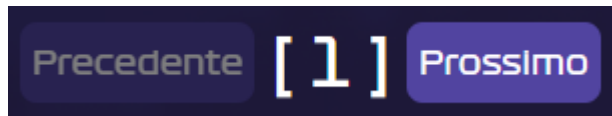
È importante sottolineare come solamente un utente regolare possa seguire una scuderia e conseguentemente, come questa funzionalità vada vietata non solo nel front-end, nascondendo semplicemente il pulsante, ma anche nel back-end tramite controlli sullo stato dell'autenticazione e sul tipo di utente.

Applicazione Media

Gli **Highlight** sono gestiti da:

- **HighlightPageView**: menu di selezione che elenca tutti gli Highlight presenti nel portale su più pagine.
Per gestire un numero elevato di elementi, è stato utilizzato il modulo **Paginator** di Django, che permette di suddividere equamente i risultati delle query in più pagine.
Tramite controlli effettuati con DTL viene permessa la navigazione tra le pagine dei risultati offerti all'utente: i bottoni saranno premibili qualora si

possa procedere in altre pagine e saranno disabilitati quando ci si troverà nella prima e nell'ultima pagina di ricerca.



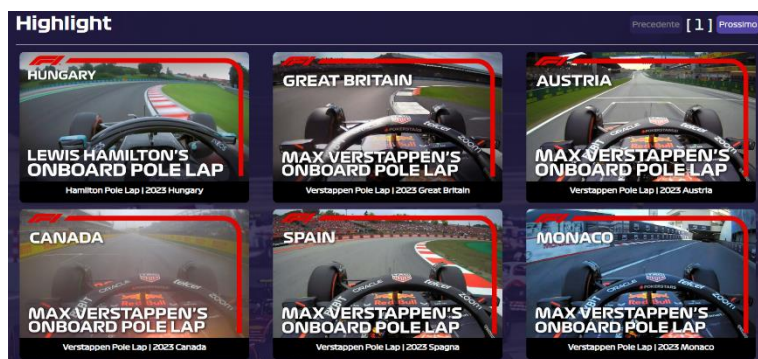
Prima pagina di ricerca



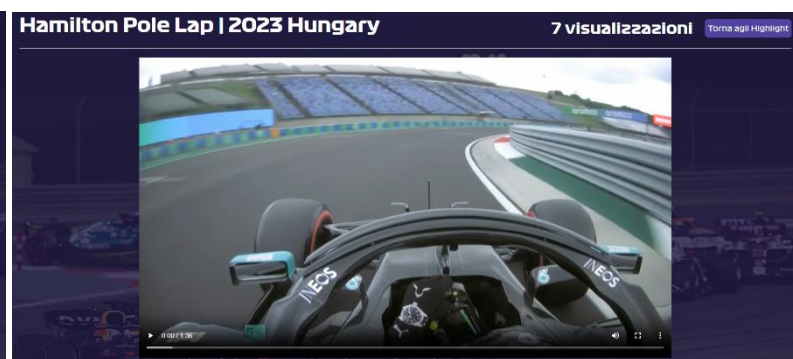
Ultima pagina di ricerca

Per la navigazione nelle pagine viene utilizzato un argomento passato nell'url (esempio: [/highlight/?page=2](#)). Qualora l'utente decida di cercare una pagina inesistente (esempio: [/highlight/?page=12345678](#)), verrà reindirizzato alla pagina "Nothing here" che permette di informarlo sull'esito della sua ricerca senza necessariamente mostrare un errore 404.

- **VideoHighlightPageView**: gestisce il video selezionato dall'utente e aggiorna il contatore delle visualizzazioni per quel preciso Highlight.



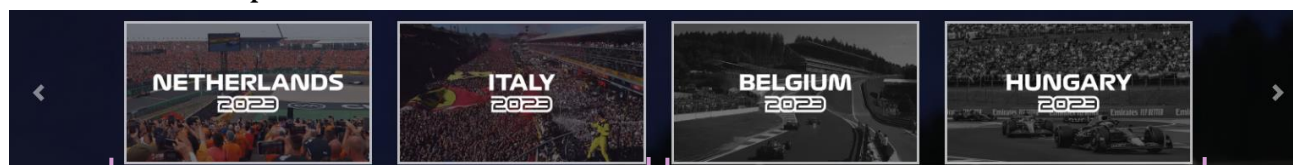
HighlightPageView – highlight.html



VideoHighlightPageView – video_highlight.html

L'homepage è gestita dalla **HomePageView**:

- Fornisce le rispettive liste dei gran premi già svolti e non ancora svolti ordinate per data.



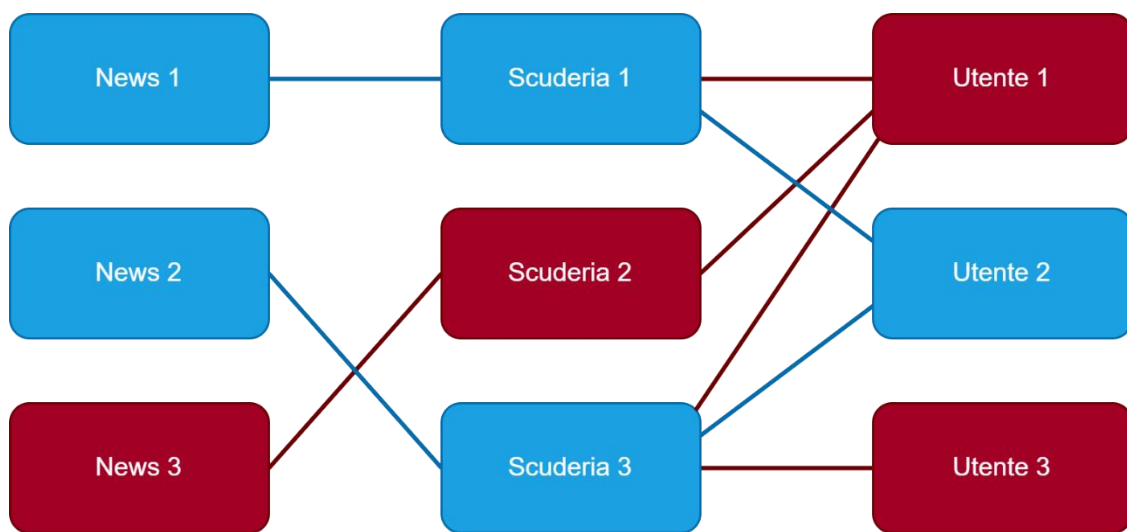
Gran premi non ancora svolti

Gran premi già svolti

- Fornisce i risultati per tutti i piloti che hanno partecipato all'ultima sessione.
- Implementa un **Recommendation System per le News** con lo scopo di proporre a un utente regolare IDONEO news pertinenti ai propri interessi.

È importante sottolineare come per utente regolare idoneo si intenda un utente che abbia messo almeno un “follow” a una scuderia. Pertanto, un utente non loggato, un utente gestore di circuito e un utente regolare loggato ma senza nessuna scuderia seguita riceveranno le stesse news in homepage ordinate cronologicamente. Se un utente regolare è loggato e segue una o più scuderia, le prime news che vedrà in homepage fino a un massimo di 5, saranno pertinenti alle scuderie che segue, mentre le altre saranno non direttamente pertinenti.

Esempio di funzionamento Recommendation System



Nella figura, si può vedere lo stato di un ipotetico database riguardante le preferenze di tre utenti. Le connessioni indicano rispettivamente che:

- La news riguarda una determinata scuderia
- L'utente segue gli aggiornamenti di una scuderia.

Prendendo in considerazione l'utente 2 e supponendo che a un numero più alto di news corrisponda a una news più recente, si otterranno i seguenti ordinamenti:



News per utente senza Recommendation System



News per Utente 2

Applicazione Store

In questa applicazione, per semplificare la gestione di quantità elevate di biglietti, abbiamo effettuato una normalizzazione per la singola entità biglietto la quale è stata suddivisa in:

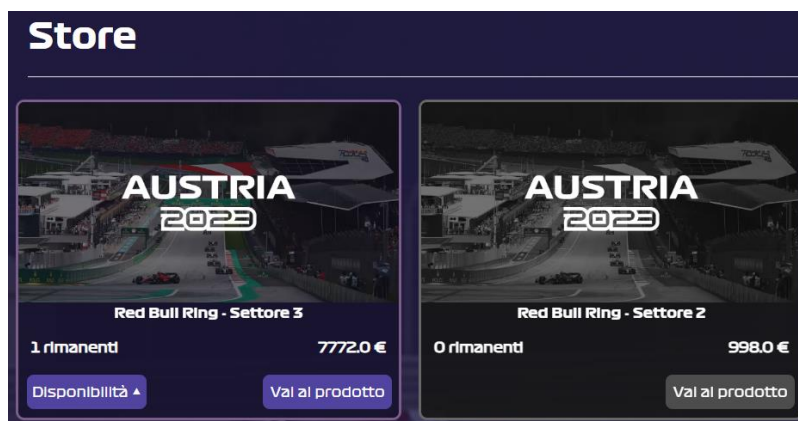
- **Tipologia Biglietto:** si riferisce a un particolare settore di un circuito e stabilisce il prezzo di tutti i biglietti per i posti che ne fanno parte.
- **Istanza Biglietto:** si riferisce a una particolare tipologia di biglietto e specifica il numero del posto ad esso associato.

Le funzionalità dell'e-commerce sono divise fra:

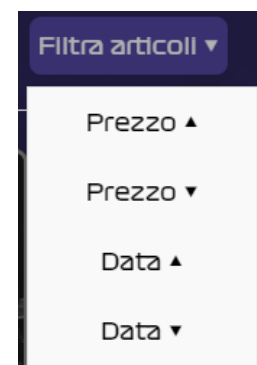
- **StoreView:** gestisce il menu di selezione per le tipologie biglietti da acquistare.

Come per gli Highlight, i vari prodotti in vendita sono paginati all'interno dello store tramite un Paginator, con la differenza che l'utente è ora in grado di interagire con l'ordinamento dei prodotti per mezzo dei filtri. Per mettere facilmente in evidenza la disponibilità di un prodotto ancor prima di selezionarlo, sono presenti:

- Un contatore delle istanze di biglietti rimanenti
- Un drop-down menu per verificare quali posti sono disponibili



StoreView – store.html



Filtri dello Store

- **ProductView:** popola il form TicketForm con le istanze di una specifica tipologia di biglietto, permettendo all'attuale utente regolare di aggiungerle al proprio carrello.

ProductView – TicketForm

Il carrello è gestito interamente dalla view **CartView**, la quale ha il compito di:

- Elencare tutti gli articoli nel carrello dell'utente regolare.
- Mostrare i dati dell'utente (nominativo e dati di pagamento).
- Calcolare il costo totale di tutti gli articoli presenti nel carrello con l'aggiunta delle tasse.

Nel momento in cui un utente decide di acquistare gli articoli nel suo carrello, invia una richiesta gestita dalla funzione post di CartView, che inizialmente si occupa di:

- Controllare che l'utente abbia fornito tutte le **informazioni valide** per il pagamento (nome, cognome, codice e scadenza carta di credito e cvv) tramite controlli selettivi su ogni parametro.
- Assicurarsi che il **carrello contenga articoli**.

In caso vengano superati tutti i controlli sopra menzionati:

- Per ogni istanza di biglietto nel carrello crea un **ordine**.
- Per ogni istanza di biglietto nel carrello genera il **biglietto digitale**.
- Per ogni ordine generato, crea una **notifica**.
- Infine, rimuove tutti i prodotti appena acquistati dal carrello di ogni utente regolare della piattaforma.

Il tuo carrello Torna allo Store

Hai 3 articoli nel tuo carrello

Articolo	Prezzo
Red Bull Ring - Settore 3 Posto 625	7772.0 €
Red Bull Ring - Settore 2 Posto 229	998.0 €
Circuit de Spa-Francorchamps - Settore 1 Posto 569	59.99 €

Dettagli del pagamento

Intestatario carta: Enrico Rossi
Numero carta: 9999999999999999
Scadenza carta: Oct. 24, 2031
CVV: ***

Totale prodotti: 8829.99 €
Tasse: 1942.60 €
Totale (Iva Inclusa): 10772.59 €

[Checkout](#)

CartView – cart.html



Transazione conclusa con successo


[Chiudi](#)

Modale successo acquisto

La sezione dedicata al **profilo** viene gestita dalla view **UserProfile**, la quale adotta un comportamento selettivo in base al tipo di utente che vi accede.

Nel caso di un utente regolare:

- **Mostrare** tutti i **dati personali**.
- **Elencare** gli **ordini** passati con e i corrispondenti **biglietti digitali**.
- Gestire la **modifica** dei **dati personali**.
- Permettere all'utente di **cambiare immagine** del **profilo**.



utentecarrello

Ultimo accesso Aug. 14, 2023, 3.36 p.m.

[Modifica profilo](#)

Nome Enrico

Cognome Rossi

Indirizzo Email aaa.com

Data di nascita Feb. 15, 2023

Sesso Maschio

Paese Azerbaijan

Telefono 3333333332

Carta di credito 999999999999998

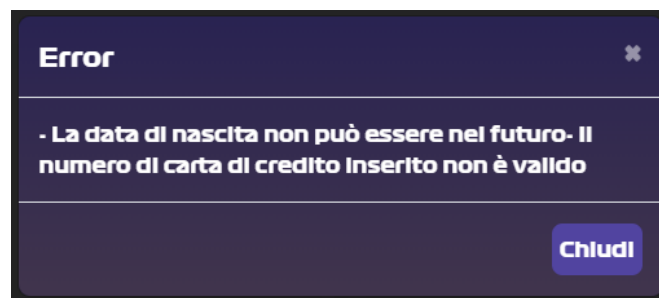
I tuoi ordini

Id ordine	Data acquisto	Prezzo	Download
Ordine 24	Aug. 17, 2023	73.19 €	📄
Ordine 23	Aug. 17, 2023	121756 €	📄
Ordine 22	Aug. 17, 2023	9481.84 €	📄
Ordine 21	Aug. 11, 2023	121756 €	📄
Ordine 20	Aug. 11, 2023	121756 €	📄
Ordine 19	Aug. 11, 2023	121756 €	📄
Ordine 18	Aug. 11, 2023	61.00 €	📄
Ordine 17	Aug. 11, 2023	61.00 €	📄
Ordine 16	Aug. 11, 2023	270.84 €	📄

UserProfile – profile.html – Profilo lato utente regolare

La **modifica dei dati personali** dell'utente viene attuata tramite la view **UtenteProfileDataChangeViewUpdate** tramite il form **UtenteProfileFormData**. Tramite funzioni di controllo selettivo, si occupa di verificare che ogni campo inserito nel form sia consono.

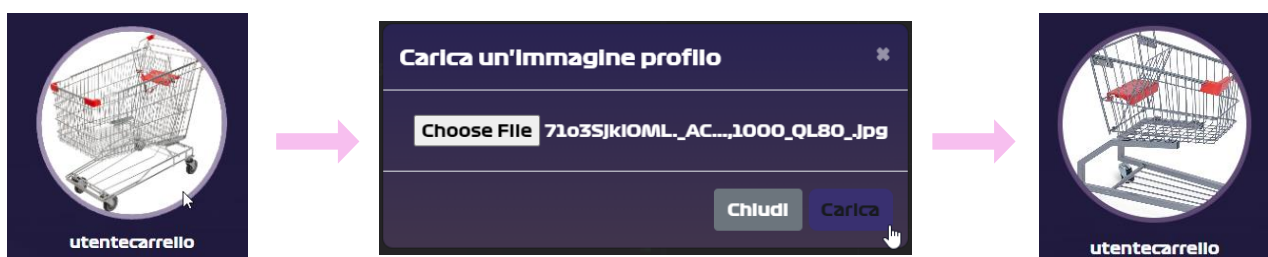
Per ogni campo che non rispetta le regole specificate, un modale di errore viene popolato con ogni motivo per il quale la modifica non ha avuto successo.



Modale errore aggiornamento dati

Similmente al lato utente regolare, anche per le modifiche ai dati dei gestori vengono utilizzate le stesse modalità, ma con altre funzioni di controllo per via dei differenti dati personali.

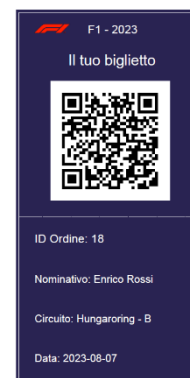
L'**aggiornamento dell'immagine profilo** dell'utente viene svolta attraverso un modale che permette di caricare un nuovo file dal proprio filesystem. Per evitare che l'utente carichi dei file non adatti, viene verificato che il file caricato sia un'immagine e che sia in uno dei formati supportati (PNG, JPG, JPEG).



Generazione dei biglietti digitali

Il singolo biglietto è generato al momento dell'acquisto per mezzo della funzione **generate_ticket**, ed è visualizzabile nella sezione profilo di ogni utente regolare, accanto a ogni ordine.

Ogni biglietto contiene varie informazioni sia in formato qrcode, sia testuale.



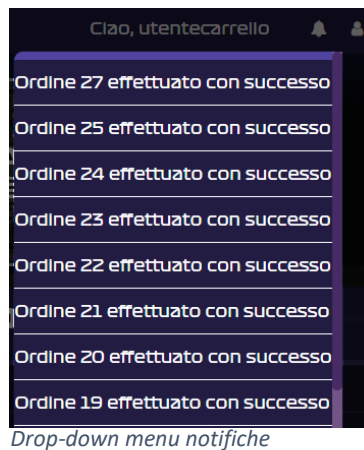
Il gestore del circuito ha accesso a funzionalità aggiuntive all'interno del suo profilo, tra cui:

- **Creazione tipologia di biglietto**, gestita dalla view **CreateTicketTypeView** dove i dati inseriti nel form **CreateTicketTypeForm** vengono controllati selettivamente verificando, in particolar modo, che non sia stato inserito un settore di un circuito già connesso a un'altra tipologia di biglietti.
- **Creazione istanza di biglietto**, gestita dalla view **CreateTicketInstanceView** dove i dati inseriti nel form **CreateTicketInstanceForm** vengono controllati selettivamente verificando, in particolar modo, che non sia stato inserito un posto già assegnato a un'altra istanza di un biglietto.
- Visualizzazione delle **statistiche di vendita** per l'ultima settimana, per permettere di identificare quale settore vende maggiormente e in quale periodo.



Notifiche

Le notifiche vengono generate quando un utente regolare acquista un'istanza di un biglietto, e hanno lo scopo di informare l'acquirente e il venditore della transazione.



Durante l'implementazione delle notifiche ci siamo imbattuti in un **problema**. Siccome le notifiche sono contenute in un drop-down menu della navbar, e la stessa navbar è contenuta in ogni template si sarebbe dovuta implementare una logica di controllo per ogni singola view.

Per risolvere il problema in maniera più elegante abbiamo scelto di:

- Creare l'endpoint [query/notifications/](#) collegata alla view **get_notifications**, la quale si occupa di recuperare le notifiche per l'attuale utente restituendole in formato Json.
- Attivare un event listener (**navbar.js**), il quale alla pressione del pulsante delle notifiche invia una richiesta e gestisce la risposta per popolare gli elementi nel drop-down menu.

Unit Test

I test sono stati effettuati in due punti dell'applicazione:

- Sulla view HighlightPageView, ci siamo concentrati sul controllo degli status code e sul corretto funzionamento della reindirizzazione nel caso di richiesta di pagine in formato non corretto o inesistenti.
- Sulle funzioni di controllo selettivo per la modifica dei dati nel profilo dell'utente, abbiamo testato ogni funzione attraverso l'inserimento di parametri volutamente scorretti per assicurarsi che l'errore mostrato all'utente sia coerente con quanto inserito.