

Progetto d'esame

Tecnologie Web

Nicholas Lo Piccolo

Indice

1. [Idea](#)
2. [Struttura del codice](#)
3. [Utenti anonimi e funzionalità di ricerca](#)
4. [Utenti registrati](#)
 - a. [Autenticazione](#)
 - b. [Forms di inserimento](#)
 - c. [Datatables](#)
 - d. [Dettaglio pasto](#)
 - e. [Interazioni sociali fra gli utenti](#)
 - f. [Dashborad Utente](#)
5. [Notifiche](#)
 - a. [Notifiche a schermo](#)
 - b. [Notifiche email](#)
6. [Utenti admin](#)
 - a. [Dashboard admin](#)
 - b. [Azioni nella dashboard](#)

Idea

L'idea generale è quella di dare un luogo alle persone con disturbi alimentari o che semplicemente vogliono tener traccia dei propri consumi calorici e in cui poter cercare l'apporto calorico di un alimento o pasto in base a nome. Gli utenti registrati inoltre avranno la possibilità di inserire i consumi calorici dei propri pasti tramite un form apposito, ma non solo infatti essi potranno anche inserire il proprio peso per tenerne traccia nel corso del tempo.

Struttura del codice

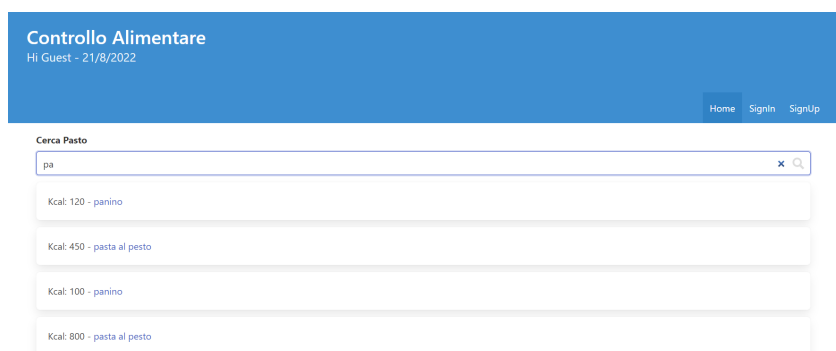
Il modulo principale del progetto è il modulo core che al suo interno ha il file settings.py (contenente le configurazioni del progetto django), la cartella dei templates e la cartella dei file statici.

Le altre due applicazioni del progetto sono:

- L'applicazione del **peso** in cui sono stati creati i file models.py per la definizione del modello, il file forms.py in cui vi è il form dell'oggetto peso e il file ajax_datatable_views.py che tramite la libreria django-ajax-datable definisce la classe che si occuperà della gestione del datatable del peso.
- L'applicazione del **pasto** in cui sono stati creati i file models.py per la definizione dei modelli: pasto, commento, likeCommento e likePasto. Vi è anche il file forms.py in cui è stato definito il form del pasto e infine il file ajax_datatable_views.py che gestisce il datatable del pasto.

Utenti anonimi e funzionalità di ricerca

L'applicazione permette di sfruttare la funzionalità di ricerca a tutti gli utenti quindi anche ad utenti anonimi o guest. Questa feature permette agli utenti di ricercare un pasto in base al nome e in particolare ad ogni typing da parte dell'utente viene inviata una richiesta di tipo post e django risponde con il queryset corrispondente. L'output ottenuto è una lista di elementi che riporta quantitativo in kcal e la descrizione del pasto come hyperlink che porta l'utente al dettaglio del pasto.



The screenshot shows the 'Controllo Alimentare' web application. At the top, there is a blue header with the text 'Controllo Alimentare' and 'Hi Guest - 21/8/2022'. On the right side of the header, there are links for 'Home', 'SignIn', and 'SignUp'. Below the header, there is a search bar labeled 'Cerca Pasto' with the text 'pa' entered. Below the search bar, there is a list of food items with their calorie counts and descriptions:

- Kcal: 120 - panino
- Kcal: 450 - pasta al pesto
- Kcal: 100 - panino
- Kcal: 800 - pasta al pesto


Una volta atterrati nella pagina di dettaglio gli utenti di tipo guest vedranno in modo offuscato il nome dell'autore e la sezione dei commenti. Se l'utente anonimo prova ad inserire un commento verrà reindirizzato alla pagina di login.

Controllo Alimentare
 Hi Guest - 21/8/2022

[Home](#)
[SignIn](#)
[SignUp](#)

Autore: anonimo

Foto:



Descrizione:
Panino con la mortadella


Kcal:
480


Tipo:
pranzo

Data:
21 Agosto 2022

0

Sezione commenti disabilitata per gli utenti guest.





Post comment

Utenti registrati

Passiamo ora agli utenti veri e propri del sito che potranno utilizzare a pieno tutte le funzionalità della piattaforma. Questi utenti non sono altro che gli User standard in django (`from django.contrib.auth.models import User`).

Autenticazione

Gli utenti potranno registrarsi alla piattaforma con il form di SignUp inserendo uno username che dovrà essere univoco, un'email e una password ripetuta due volte. Questo form è stato ridefinito sulla base di quello originale (`from django.contrib.auth.forms import UserCreationForm`) ed è stato aggiunto il campo email. Si trova nel file `core/forms.py`.

Username

Email

password

re-password

- Evita password comuni
- Lunghezza minima 8 caratteri

Registrati

Username
nicholas

Password

- Evita password comuni
- Lunghezza minima 8 caratteri

Login

L'utente con account potrà autenticarsi grazie all'apposito form di login in cui basterà inserire username e la password. Il form utilizzato è quello default di django

```
(from django.contrib.auth.forms import
AuthenticationForm).
```

Controllo Alimentare

Hi nicholas - 22/8/2022

L'utente una volta autenticato potrà capire con quale account ha eseguito il login tramite la navbar presente in ogni pagina.

Forms di inserimento

Come già detto i dati che possono essere inseriti dagli utenti sono prevalentemente due il peso ed il pasto.

Il **peso** è l'inserimento più semplice fra i due poichè richiede solo di inserire il peso corporeo dell'utente espresso in kilogrammi ed una data indicativa.

Es. ho fatto una vacanza di due settimane e all'inizio pesavo 75 kg, sono tornato oggi e ne peso 80. Mell'applicazione eseguirò due inserimenti uno con data pregressa dell'inizio della mia vacanza ed uno con la data odierna.

Peso:

Data:

gg/mm/aaaa


Invia

Il **pasto** invece ha un form più articolato poichè necessità di più dettagli, ecco la lista:

- Descrizione: testo di lunghezza massima 1024 caratteri in cui si può descrivere dettagliatamente il pasto
- Kcal: serve per specificare il numero di calorie totale del pasto
- Tipo: indica la tipologia di pasto che può avere 4 valori predefiniti
 - colazione

- pranzo
- snack
- cena
- Data: rappresenta la data in cui si è consumato il pasto
- Immagine: opzionalmente si può inserire un'immagine del pasto


In aggiunta al form vi è anche la barra di ricerca che può essere un utile spunto per l'utente durante l'inserimento del proprio pasto.

Cerca Pasto


Descrizione:

Kcal:

Tipo:

Data:


Inserisci una foto del tuo pasto:

Datatables

Oltre ai form è necessario mostrare agli utenti i propri inserimenti che potranno **visualizzare** nel dettaglio (solo per i pasti), **modificare** o **eliminare** tali dati tramite le apposite actions a destra di ogni riga dei datable.

Show entries

Search:

	Date	Descrizione	Kcal	Tipo	Edit
	<input type="text"/>	<input type="text"/>		<input type="text"/>	
+ -	22/08/2022	Pasta al pesto con scaglie di grana	480	pranzo	i ✎ ✖
+ -	22/08/2022	Waffle con la nutella	220	snack	i ✎ ✖
+ -	22/08/2022	Piadina con il cotto	499	cena	i ✎ ✖
+ -	19/08/2022	panino	120	colazione	i ✎ ✖

Showing 1 to 4 of 4 entries

Previous
1
Next

Come mostrato nell'immagine nel datatable è possibile eseguire una ricerca per data, descrizione o tipo di pasto. Oltre ad una piccola descrizione è possibile estendere il singolo elemento e visionare tutti i dati salvati nel database.

Dettaglio pasto


Ogni pasto ha una pagina di dettaglio dedicata per una miglior consultazione e per le azioni social, che verranno illustrate in dettaglio nel prossimo paragrafo.

Controllo Alimentare

Hi nicholas - 22/8/2022

[Home](#)
[Pasto](#)
[Peso](#)
[Logout](#)

Autore: nicholas

Foto:



Descrizione:
Pasta al pesto con scaglie di grana

Kcal:
480

Tipo:
pranzo

Data:
22 Agosto 2022

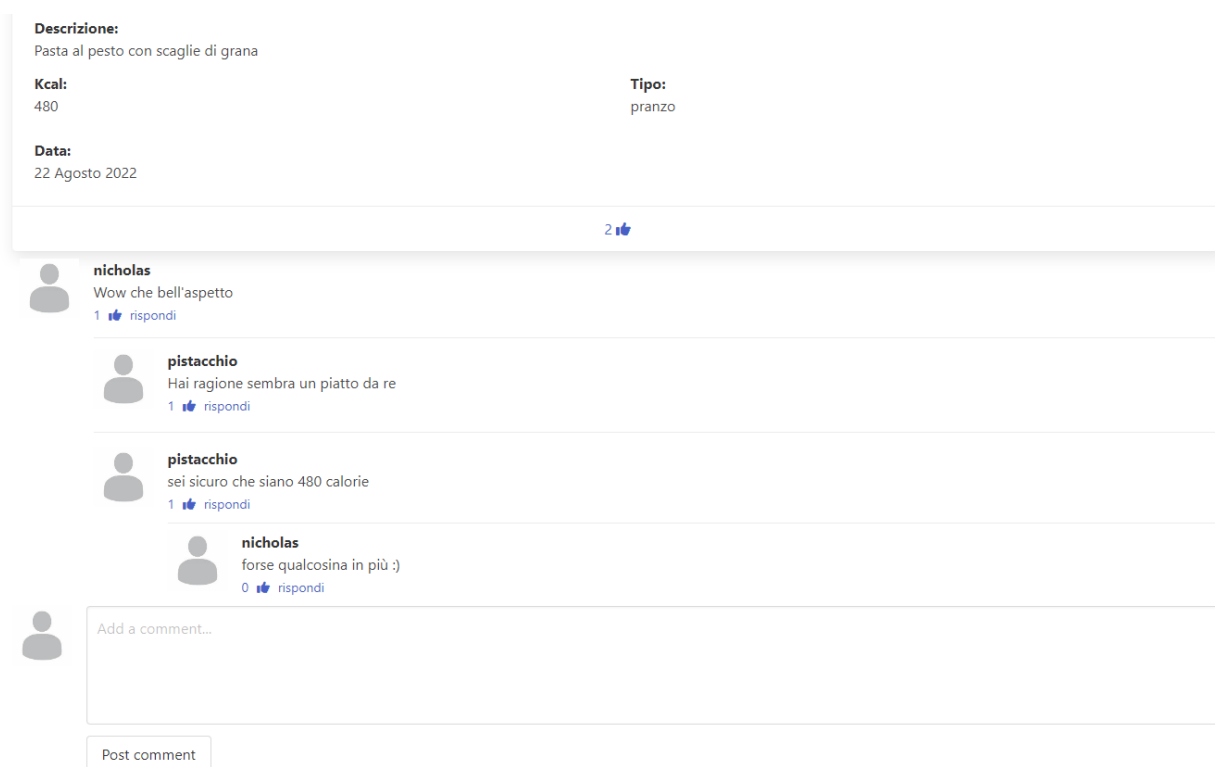
0 [👍](#)


Add a comment...

Post comment

Interazioni social fra gli utenti

Come visto nell'immagine precedente l'utente può commentare i pasti altrui tramite un'apposita sezione e può anche rispondere a commenti degli altri utenti. Un'ulteriore interazione social è quella di poter aggiungere like a commenti e pasti.



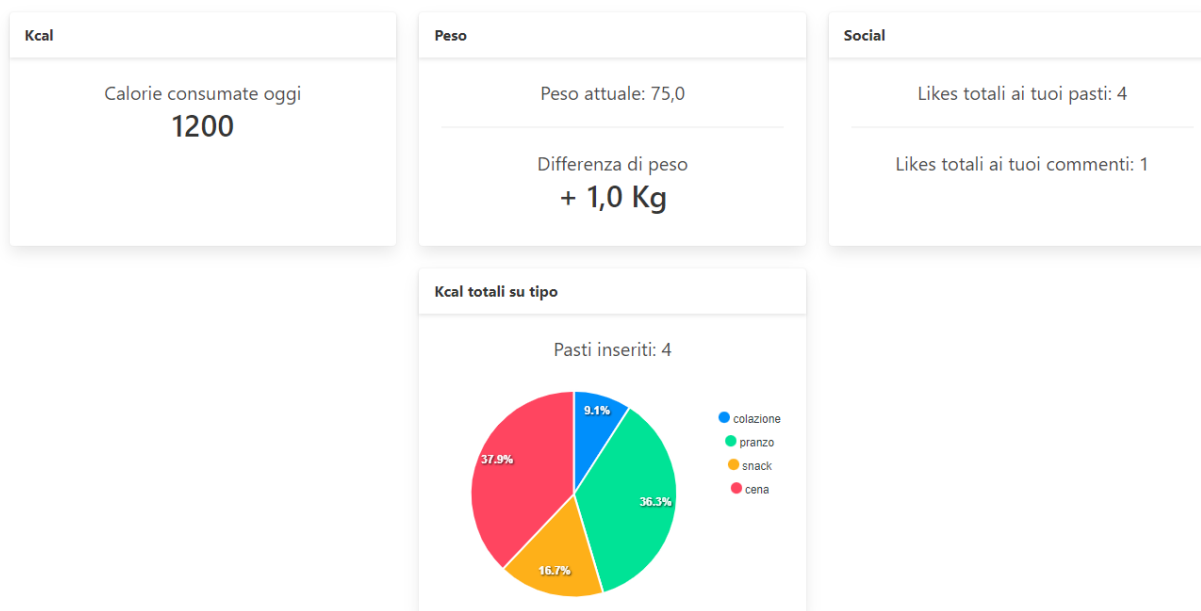
Dashboard utente

Questa dashboard è visibile nella pagina home di ogni utente e mostra alcuni summary che potrebbero essere utili.

La dashboard presenta 4 card e ognuna indica un elemento molto importante nella piattaforma:

- **Kcal:** rappresenta il totale delle kcal consumate nella giornata e permette all'utente di porre particolare attenzione al proprio consumo non andando ad eccedere un tetto massimo che ci si pone oppure consumandone ulteriori in caso contrario.
- **Peso:** questa card rende noti all'utente due informazioni la prima indica il peso attuale dell'utente quindi l'ultimo inserito per ordine di data e data di creazione, il secondo indica la differenza di peso dato dagli ultimi due inserimenti sempre per ordine di data e data creazione.
- **Social:** utili per tenere tracci dei like ottenuti dal punto di vista dei pasti inseriti e dai propri commenti.

- Kcal totali su tipo: questo grafico è creato sulla base di tutti i pasti inseriti nella piattaforma ed è utile per capire in che modo spendiamo le nostre calorie in una giornata tipo. Nell'immagine sottostante possiamo vedere che l'utente consuma poche calorie durante la colazione e invece ne consuma troppe in caso di snack (ovviamente queste ipotesi sono date dal tipo di dieta che l'utente sta seguendo). Il grafico è stato creato con l'utilizzo del framework web **apexcharts.js**



Notifiche

Per rendere noto agli utenti le interazioni social e l'invio con successo o meno di form ho sviluppato due sistemi di notifiche una a schermo tramite dei box colorati e uno via email.

Notifiche a schermo

Ogni azione dell'utente come l'invio di un form di login, inserimento pasto, commento, reply o like vengono notificati all'utente tramite una notifica verde. In caso di errore come il login errato o l'invio di un form non valido viene notificato con un messaggio e una notifica di colore rosso. In caso di elemento mancante o link inesistente viene notificato con una notifica gialla. Il sistema di notifiche a schermo è stato implementato tramite l'utilizzo del modulo messages messo a disposizione da django (`from django.contrib import messages`).

Login avvenuto con successo

Oggetto inesistente

Notifiche Email

Le notifiche email sono state implementate con la funzione di django send_email dopo la creazione di una email ed un'adeguata configurazione del file core/settings.py.

Per una migliore gestione interna ho creato un modulo ad hoc chiamato notification.py che implementa una funzione che prende in input un messaggio, una lista di utenti e opzionalmente un oggetto dell'email.

Le notifiche possono essere di 4 tipologie:

- Notifica per un nuovo commento sotto un proprio post
- Notifica per un nuovo commento sotto un post a cui l'utente ha risposto (in caso di reply)
- Notifica per un nuovo like ad un proprio pasto
- Notifica per un nuovo like ad un proprio commento



Utenti Admin

Gli utenti admin sono di due tipologie:

- Utente owner contraddistinto da una stellina rossa (flag `is_superuser True`)
- Utente staff contraddistinto da una stellina gialla (flag `is_staff True`)

Entrambi eseguono il login nello stesso modo degli utenti standard.

Dashboard admin









Questi utenti hanno a disposizione una dashboard aggiuntiva in cui è possibile vedere le informazioni degli utenti ed eseguire alcune azioni su di essi. Un'altra peculiarità di questo utente è la possibilità di visionare il tag admin nei pasti di loro creazione.

Azioni nella dashboard

Nella dashboard sono previste due azioni possibili su ogni utente in base al loro ruolo:

- Utenti owner: possono disattivare o attivare l'account di un utente non permettendo loro di eseguire il login nella piattaforma e possono anche dare o rimuovere i privilegi di staff
- Utenti staff: possono solo disattivare o attivare l'account di un utente

Dashboard Admin

Lista utenti:				
Utenti totali: 4				
Tipologia d'utente	Username	email	Ultimo accesso	Azioni
★ owner	admin	admin@admin.admin.it	Martedì 23 Agosto 2022 15:20	 
★ staff	mageox98	giokkaaaa@gmail.com	Martedì 23 Agosto 2022 15:20	 
utente	pistacchio	nicholas.lopiccio1998@gmail.com	Martedì 23 Agosto 2022 14:12	 
utente	nicholas	269374@studenti.unimore.it	Lunedì 22 Agosto 2022 14:46	 

Le azioni nella dashboard potrebbero non essere chiare quindi è stata aggiunta la keyword title nei tag relativi che permette di capire meglio a cosa serve la determinata azione.