



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

Relazione Reti di Calcolatori

Università degli Studi di Urbino Carlo Bo

Dipartimento di Scienze Pure e Applicate

Facoltà di Informatica Applicata

Anno Accademico 2023/2024

Docente: Antonio Della Selva

Studente: Miriana Sette

Matricola: 313206

Indice

1	Titolo	2
2	Introduzione	2
2.1	Struttura del Progetto	2
2.2	Configurazione progetto	3
2.3	Architettura	3
3	Discussione critica della tematica	4
3.1	Sessioni Django	4
3.2	Database	4
3.3	Creazione Utente e Autenticazione	4
3.4	RSS Feed	5
3.5	Web Scraping	6
3.6	Test ed esecuzione	8
4	Conclusione	10

1 Titolo

Sviluppo di un'applicazione web basata su architettura client server che implementa un lettore Rss feed permettendo agli utenti di visualizzare una lista di canali inerenti al mondo del cinema e dello streaming. L'applicazione, oltre alle funzionalità di lettore, deve predisporre interfacce personali, garantendo una gestione privata dei canali per ogni utente.

2 Introduzione

L'obiettivo di questo progetto è quello di creare un'applicazione web che consenta all'utente di visualizzare dapprima i canali riguardanti le news sul cinema disponibili e che permetta l'iscrizione e/o disiscrizione solo dopo la creazione di un account e autenticazione al server di riferimento, così che possa essere modellato uno spazio di contenuti personali nel quale saranno esposte le ultime notizie appartenenti ai canali ai quali l'utente si è iscritto.

L'utente, dopo aver scelto il canale di interesse, potrà accedere alla pagina di login ed eventualmente creare il proprio account, in modo tale da ottenere i requisiti per la gestione del proprio spazio personale. Una volta fatto ciò, l'utente sarà in grado di selezionare e aggiungere qualsiasi canale presente alle proprie preferenze, in modo che, nella propria sezione personale, potrà visualizzare sempre gli ultimi 5 aggiornamenti degli Rss feed prendendo visione dei contenuti in maniera molto semplice e diretta.

2.1 Struttura del Progetto

Il progetto è sviluppato utilizzando il framework Django, noto per la sua robustezza e flessibilità nella creazione di applicazioni web. Di seguito vengono elencate le principali tecnologie e dipendenze utilizzate:

- Django 5.0.6: framework principale per lo sviluppo backend.
- SQLite: database utilizzato per la gestione dei dati.
- BeautifulSoup: libreria utilizzata per il web scraping.

Il progetto è strutturato in diversi file e directory, tra cui:

- MovieHub: contiene l'intero progetto e la configurazione del progetto Django.
- MovieNews: include i file python per la vera applicazione.
- Templates: contiene tutti i file HTML necessari per definire le pagine e il frontend dell'applicazione.
- Static: directory per i file CSS e le immagini utilizzate all'interno dei template html.

2.2 Configurazione progetto

Una volta installato il progetto ad esempio tramite link del repository github è necessario eseguire diversi comandi per l'esecuzione in locale dell'applicazione web:

1. `cd path\locale\repository`
2. `python3 -m venv ambiente`
3. `source ambiente\bin\activate` [da Unix/MacOS]
4. `.\ambiente\Scripts\activate` [da Windows]
5. `pip install -r requirements.txt`
6. `cd MovieHub`
7. `python manage.py migrate`
8. `python manage.py runserver`

2.3 Architettura

Django segue il pattern architetturale Model-View-Template (MVT), che è simile al pattern Model-View-Controller (MVC). La struttura del progetto è organizzata in modo da separare chiaramente i vari componenti che compongono un'applicazione di questo tipo:

- **Model:** permette di definire la logica dei dati e l'interazione con essi attraverso la gestione del database; SQLite è il DB utilizzato per la gestione di una quantità minima di dati con semplici interazioni.
- **View:** rappresenta la logica di business dell'applicazione, definendo il comportamento delle funzionalità presenti.
- **Template:** gestisce la presentazione dei dati all'utente finale, contenendo quindi le pagine html.

Ogni funzionalità all'interno dei progetti Django è definita nel suo complesso attraverso l'interazione e la sistemazione di questi tre elementi, modellando l'applicazione sotto tutti gli aspetti.

3 Discussione critica della tematica

In questa sezione si vogliono descrivere con cura tutte le scelte implementative ed eventuali passi di codice importanti per la descrizione dell'applicazione di riferimento.

3.1 Sessioni Django

All'interno del progetto sono stati utilizzati concetti relativi alle sessioni. Django utilizza quest'ultime tramite l'uso di cookie per gestire lo stato degli utenti durante le interazioni con l'applicazione. Le sessioni sono usate per mantenere all'interno del browser i dati importanti relativi agli utenti e alle loro azioni.

Quando un utente accede al sito, il framework crea delle sessioni ad hoc per quell'utente, associandone un identificativo univoco. Tutte queste informazioni sono memorizzate attraverso dei cookie, veri e propri file che agiscono inviando dati dal server al browser, memorizzandoli nella macchina locale del client.

Durante l'esecuzione, nel momento in cui occorre individuare queste informazioni, ottenerle attraverso le sessioni salvate risulta molto rapido e semplice.

3.2 Database

Per sviluppare la seguente app, è necessario l'utilizzo di un Database in modo che le informazioni possano persistere. Nel Database vengono memorizzati i dati relativi ai diversi utenti che si registrano, le iscrizioni ai diversi canali e le relative notizie.

Il Database è gestito principalmente dal framework Django, che presenta dei modelli in grado di definire le strutture che costituiscono il database e le operazioni fondamentali come l'inserimento e la cancellazione dell'entità.

Le seguenti operazioni possono essere eseguite direttamente nell'area amministrativa predisposta da Django come parte del progetto creato.

3.3 Creazione Utente e Autenticazione

La creazione di un utente garantisce la possibilità di avere uno spazio personalizzato per ogni utente. Questo deriva dalla necessità di poter creare interfacce diverse a seconda delle preferenze e delle scelte dell'utente. Le operazioni definite in merito sono:

- Registrazione: gli utenti possono registrarsi fornendo un nome, username e password, che verranno salvati nel database.
- Login: gli utenti possono autenticarsi confrontando le credenziali fornite con quelle memorizzate nel database.

Non sono state prese in considerazione ulteriori scelte di sicurezza per l'accesso privato ai contenuti, poichè non ritenute necessarie per le informazioni trattate. Tuttavia, in un sistema più evoluto, con la presenza di più dati sensibili, l'implementazione di ulteriori livelli di sicurezza sarebbe obbligatoria.

```

if request.method == 'POST':
    username = request.POST.get('username')
    password = request.POST.get('password')
    nome = request.POST.get('nome')

    user_exist = User.objects.filter(user=username).first()
    if not user_exist:
        # raccolti i dati posso creare l'utente e salvarlo in DB
        user = User.objects.create(
            name = nome,
            user = username,
            password = password
        )
        user.save()
        request.session["username"] = user.user
        return redirect("home")

```

3.4 RSS Feed

Gli RSS feed (Really Simple Syndication) sono utilizzati per ottenere e aggiornare le notizie dei vari canali a cui l'utente si iscrive.

Nel database vengono memorizzate le informazioni inerenti il titolo del canale, la descrizione e l'immagine caratterizzante, oltre alle news estratte ed associate a quel canale. Gli utenti potranno visualizzare tutti i canali disponibili, iscriversi e/o disiscriversi e successivamente visualizzarne i contenuti.

```

if "username" in request.session:
    user_log = request.session["username"]
    user = User.objects.get(user = user_log)

    if request.method == "POST":
        url = request.POST.get("url_rss")
        channel = RssFeed.objects.get(url = url)
        if not channel in user.rss_list.all():
            user.rss_list.add(channel)
        else:
            user.rss_list.remove(channel)

        user.save()
    sub_url = list(user.rss_list.all())

```

I canali Rss presenti in Database sono composti di un file xml, definendo una struttura sintattica delle notizie presenti sempre uguale e ripetitiva per ognuna. Questo aspetto ha permesso di utilizzare degli strumenti, adatti all'estrazione di tutte le informazioni, in maniera sistematica per tutte le notizie.

La maggior parte dei siti web utilizza Rss feed per aggiornare frequentemente i contenuti dei loro siti; i canali selezionati non sono da meno e sono stati selezionati canali che hanno in comune gli elementi che compongono il loro file xml.

Questo consente una buona scalabilità nell'inserimento futuro di nuovi canali, infatti ognuno di essi potrà essere analizzato analogamente ad un altro. Gli articoli nei file dei canali presenti sono così composti:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss xmlns:atom="http://www.w3.org/2005/Atom" version="2.0">
  <channel>
    <title>Xeud.it - Serie TV & Film</title>
    <link>https://xeud.it/category/3</link>
    <description>Feed RSS per la categoria Serie TV & Film</description>
    <item>
      <title>Fantastic Four: cosa sappiamo finora?</title>
      <link>https://xeud.it/article/fantastic-4-cosa-sappiamo-finora</link>
      <description>Il 14 Febbraio di quest'anno è stata annunciata la data di uscita di Fantastic 4, film già annuncia-
sin da allora, le aspettative del film sono sempre state alte da parte dei fan. La data di uscita programmata pe-
momento questo progetto è avvolto ancora in molte ombre, ma proviamo a fare chiarezza! Il cast principale Il 14
anche un'immagine dei quattro protagonisti con i loro attori, ed abbiamo: Pedro Pascal (The Last of Us, The Mand
Joseph Quinn (Stranger Things) nel ruolo di Johnny Storm/La Torcia Umana, Vanessa Kirby (Mission Impossible 7) n
Bachrach (The Bear) nel ruolo di Ben Grimm/La Cosa. Il casting ha sorpreso i fan, in particolare la scelta di Eb
un po' di stucco la scelta di Pedro Pascal come Reed Richards, dopo la presenza di John Krasinski in Multiverse
ciò le aspettative s... Leggi l'articolo completo su https://xeud.it/article/fantastic-4-cosa-sappiamo-finora</d
      <author>Alberto Dini</author>
      <enclosure url="https://xeud.it/image/50a05158-0b10-41c0-a94b-d21cc67707f9.webp" type="image/jpeg"/>
      <pubDate>Mon, 27 May 2024 08:55:00 +0100</pubDate>
    </item>
  </channel>
</rss>
```

3.5 Web Scraping

Per estrarre le informazioni Rss Feed in modo tale da poter essere organizzate e gestite all'interno della pagina web html e renderle quindi disponibili in un formato leggibile all'utente, è necessario adottare tecniche di web scraping.

Il web scraping è utilizzato per estrarre informazioni dal web, attraverso la conoscenza del documento che si sta visionando. Comunemente, il web scraping viene effettuato su pagine html o, in alternativa, su documenti xml, che rappresentano l'estensione degli Rss feed.

La libreria BeautifulSoup, presente in Python, facilita l'analisi del contenuto a tale scopo. Una volta che l'utente è all'interno di una sessione, sarà possibile accedere ai canali a cui è iscritto, reperire l'url di riferimento al rispettivo Rss feed e procedere con una specifica analisi del documento xml presente.

In particolare la scelta implementativa è stata quella di salvare le informazioni estratte dal documento nel Database così da renderle disponibili per ulteriori controlli e richieste senza dover eseguire nuovamente lo scraping.

L'aggiornamento delle informazioni è stato impostato su base oraria, in linea con le frequenze di aggiornamento dei canali testati.

```

for channel in rss_list:
    first_news = channel.news.first()
    if first_news:
        last_update = first_news.date
        current_time = datetime.now(pytz.utc)
        difference = current_time - last_update
        hour = timedelta(hours=1)
        # se per quel canale, l'ultima notizia in DB ha meno di un ora allora
        # per quel canale si hanno già le news aggiornate
        if difference >= hour:
            scrape = True

    if not first_news or scrape:
        scraping(channel, channel.news.all())

```

Questo permette al nostro sistema di essere molto veloce nella gestione delle richieste degli utenti e sempre disponibile, evitando sovraccarichi inutili per richieste che di fatto non necessitano di una nuova estrazione dei dati. La visualizzazione è limitata alle ultime 5 notizie per canale, ottimizzando la presentazione nell'area personale. La funzione di scraping, visualizzabile di seguito, esegue un normale download della pagine web, analizza il contenuto estraendo, secondo la logica dettata, le informazioni di interesse e aggiorna il database con le nuove informazioni.

In particolare si nota che le notizie reperibili sono limitate a 5 per praticità di sviluppo e l'orario viene impostato ad un contesto di zona UTC per evitare problemi di sincronizzazione con diversi contenuti.

```

page = requests.get(channel.url)
soup = BeautifulSoup(page.content, features = 'xml')
# si prendono solo i primi 5 item
articles = soup.findAll(name='item', limit = 5)

# aggiorno il DB
for a in articles:
    descr = a.find('description').text
    short_split = descr.split()[:10]
    pubDate = a.find('pubDate').text
    date = datetime.strptime(pubDate, _format='%a, %d %b %Y %H:%M:%S %Z')
    feed = News.objects.create(
        news_title = a.find('title').text,
        description = descr,
        short_description = ' '.join(short_split),
        imagine = a.find('enclosure').get('url'),
        date = date,
        rss_model = channel
    )
    feed.save()

```

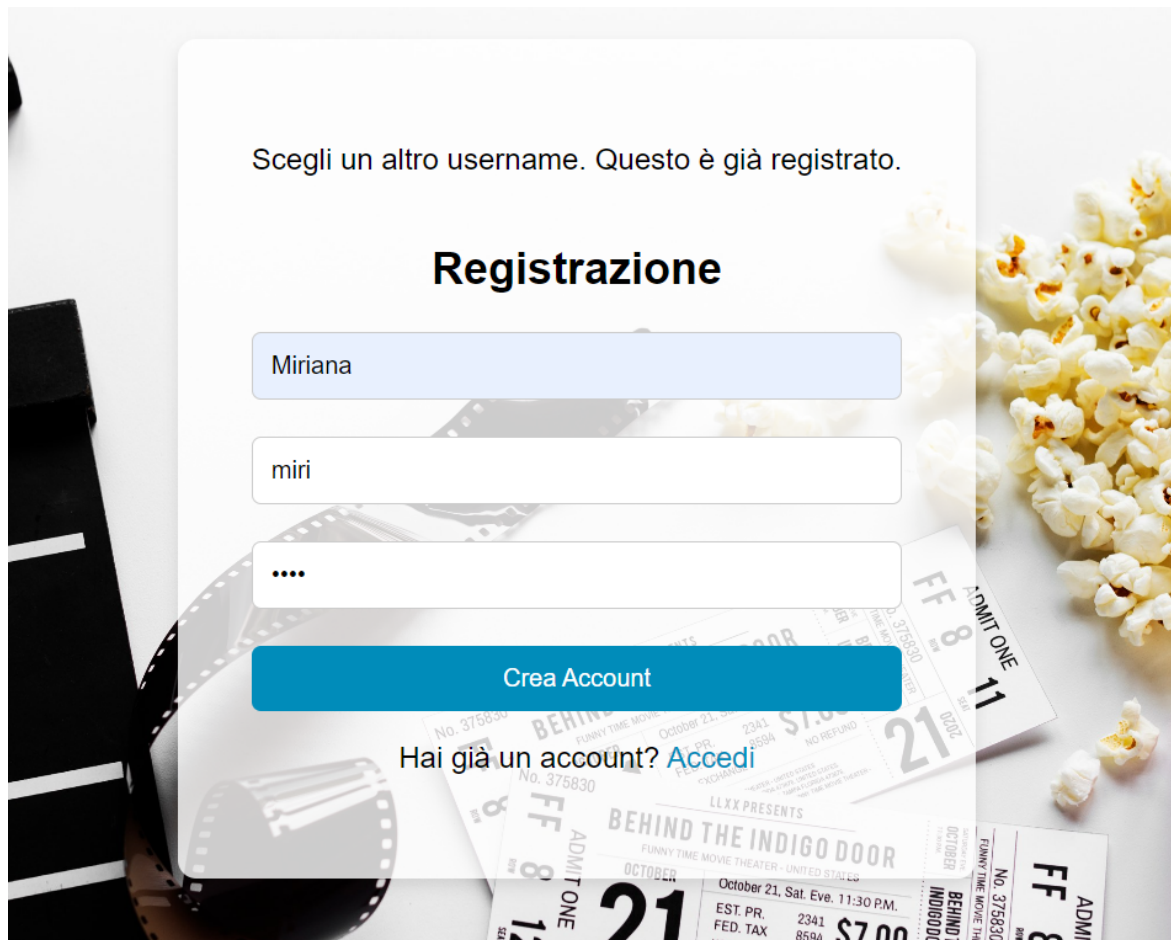

3.6 Test ed esecuzione

Le diverse fasi del progetto spiegate nella parte introduttiva della relazione, sono state testate a sufficienza, e di seguito sono mostrate le principali funzionalità dell'app.

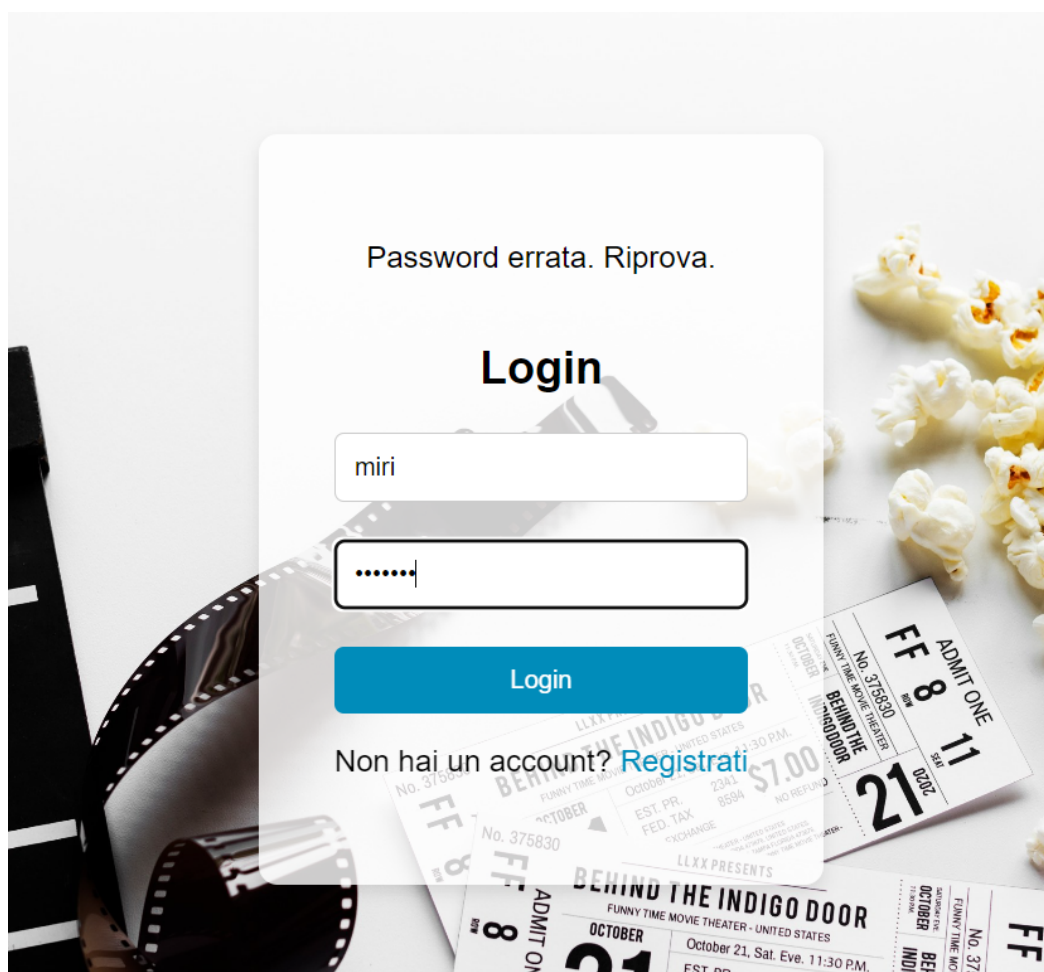
REGISTRAZIONE

La pagina di seguito viene visualizzata anche come pagina principale all'avvio dell'applicazione, consentendo la visione dei canali presenti.

Nel momento in cui però, tale pagina viene visitata senza essere all'interno del proprio account si è impossibilitati nel compiere operazioni di iscrizione o disiscrizione che lasciano tracce di dati per un determinato user, che quindi è richiesto.



LOGIN



HOMEPAGE

Ciao, miri Logout I tuoi contenuti



Xeud - Serie TV & Film

Se siete amanti delle serie tv e film allora questo è sicuramente il canale adatto a voi, con recensioni, interviste, articoli, guide e molto altro.

[Disiscriviti](#)



Xeud - Anime & Manga

Se sei un appassionato di Anime e Manga ecco a te notizie, articoli, interviste, recensioni e molto altro per la categoria!

[Iscriviti](#)

AREA PERSONALE

Di seguito viene mostrato un esempio di area personale con le notizie ai canali seguiti visibili. Questo però non sarà possibile se non si dispone di un account e comunque se non si seguono opportunamente dei canali.



Fantastic Four: cosa sappiamo finora?

Il 14 Febbraio di quest'anno è stata annunciata la data di uscita di Fantastic 4, film già annunciato tra gli annunci del San Diego Comic Con del 2022, e sin da allora, le aspettative del film sono sempre state alte da parte dei fan. La data di uscita programmata per il film è il 25 Luglio dell'anno prossimo, e al momento questo progetto è avvolto ancora in molte ombre, ma proviamo a fare chiarezza! Il cast principale Il 14 Febbraio, oltre alla data d'uscita, è stata rivelata anche un'immagine dei quattro protagonisti con i loro attori, ed abbiamo: Pedro Pascal (The Last of Us, The Mandalorian) nel ruolo di Reed Richards/Mr. Fantastic, Joseph Quinn (Stranger Things) nel ruolo di Johnny Storm/La Torcia Umana, Vanessa Kirby (Mission Impossible 7) nel ruolo di Sue Storm/La Donna Invisibile, e Ebon Ross-Bachrach (The Bear) nel ruolo di Ben Grimm/La Cosa. Il casting ha sorpreso i fan, in particolare la scelta di Ebon e Joseph nei loro rispettivi ruoli. Però ha lasciato un po' di stucco la scelta di Pedro Pascal come Reed Richards, dopo la presenza di John Krasinski in Multiverse of Madness, già più quotato come fancast, ma nonostante ciò le aspettative s... Leggi l'articolo completo su <https://xeud.it/article/fantastic-4-cosa-sappiamo-finora>



Bodkin, la serie che prende in giro il true crime

Arrivata la settimana scorsa su Netflix, Bodkin è già nella [Leggi di più](#)

4 Conclusione

Il progetto MovieHub rappresenta un'applicazione web che combina l'autenticazione utente e una gestione personalizzata dei contenuti, fungendo da lettore RSS feed per fornire un'esperienza semplice e aggiornata agli utenti, in questo caso rappresentati dagli appassionati di film.

Tramite le scelte architetture e implementative che sono state adottate, la seguente applicazione potrebbe essere espansa con nuove funzionalità in maniera molto semplice e arricchita da canali attraverso semplici inserimenti all'interno del Database.

Le nozioni acquisite durante il corso sono state fondamentali per comprendere il significato delle istruzioni presenti nel codice e il funzionamento del livello applicativo.

In futuro l'applicazione potrebbe essere adattata ad un lettore Rss universale andando ad adattare la logica e il procedimento di scraping.

Questo applicativo potrà essere sicuramente ampliato con nuove funzionalità in futuro e potrà essere pubblicato sul web per essere visibile a veri utenti finali.