

Proiect ASO

Problematica propusa

Se va implementa un site in python utilizand frameworkul django. Site-ul va avea urmatoarele functionalitati:

- utilizatorii se pot loga
- utilizatorii pot posta mesaje si vizualiza mesajele postate. Mesajele se vor salva in baza de date (deocamdată SQLite), folosind ORM-ul din Django5. Paginile vor fi realizate cu ajutorul template-urilor.
- Pe langa text, utilizatorii pot posta si imagini. Imaginile postate se vor salva ca fisiere media, nu in baza de date.
- Exista mai multe camere de chat. Un utilizator poate crea camere de chat in care sa invite alti utilizatori.
- Mesaje private criptate: fiecare utilizator are o cheie publica si o cheie privata. Cheia privata se tine local, nu pe server. Cine transmite mesajul il cripteaza cu cheia publica a destinatarului, iar acesta si-l decripteaza cu cheia privata.
- Atunci cand un utilizator incepe sa tasteze, pe chat apare informat, ca "utilizatorul X scrie". - Daca un utilizator nu este on-line cand i se transmite un mesaj, primeste un e-mail.

Librarii propuse spre utilizare

Channels library este o librerie python care a fost creat pentru a extinde capacitatea lui Django la nivelul următor. Django nu suporta asincron și canale și conexiune prin WebSockets pentru a crea aplicații în timp real. Channels extinde capacitatea Django dincolo de HTTP și îl fac să funcționeze cu WebSockets, protocoale de chat, protocoale IoT și multe altele. Este construit pe suport ASGI, care înseamnă Asynchronous Server Gateway Interface. ASGI este succesorul WSGI care oferă o interfață între async și python. Canalele oferă funcționalitatea ASGI, prin extinderea WSGI la acesta și oferă suport ASGI cu WSGI. Canalele îmbină, de asemenea, arhitectura bazată pe evenimente cu straturile de canale, un sistem care vă permite să comunicați cu ușurință între procese și să vă separați proiectul în diferite procese.

Mod de instalare:

```
python -m pip install -U channels
```

Biblioteca Twilio Python Helper facilitează interacțiunea cu API-ul Twilio din aplicația dvs. Python. Cea mai recentă versiune a bibliotecii poate fi găsită pe PyPi. Biblioteca Twilio Python Helper acceptă aplicații Python scrise în Python 3.6 și versiuni ulterioare.

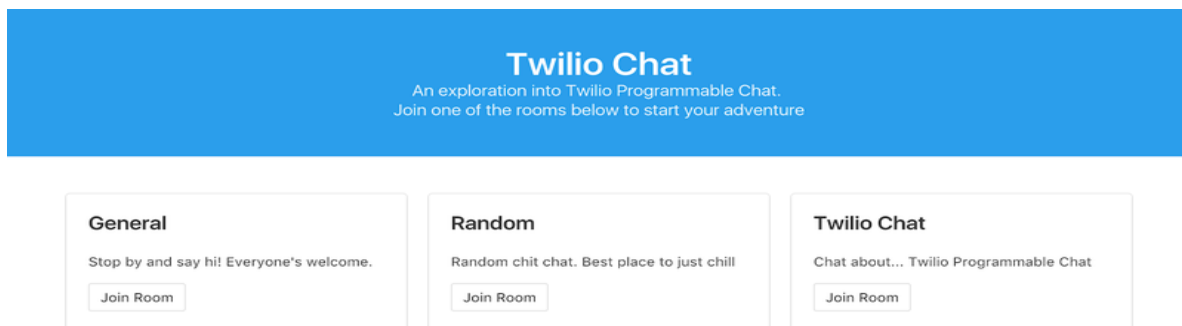
Instalare Twilio

```
pip install django==1.11.10 twilio==6.10.1
```

```
pip install python-dotenv
```

```
# chatroom_app/settings.py
```

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'chat.apps.ChatConfig',  
]
```



Tehnologii propuse

- Python
- TCP/IP
- WEB Sockets
- JavaScript
- HTTP
- Docker

Docker poate împacheta o aplicație și dependențele acesteia într-un container virtual care poate rula pe orice computer Linux, Windows sau macOS. Acest lucru permite aplicației să ruleze într-o varietate de locații, cum ar fi on-premises , în public (vezi calcul descentralizat , calcul distribuit și cloud computing) sau cloud privat . Când rulează pe Linux, Docker folosește caracteristicile de izolare a resurselor kernel-ului Linux (cum ar fi cgroups și spații de nume kernel) și un sistem de fișiere capabil de unire (cum ar fi OverlayFS) pentru a permite containerelor să ruleze într-o singură instanță Linux, evitând suprasarcina de pornire și întreținere a mașinilor virtuale . Docker pe macOS folosește o mașină virtuală Linux pentru a rula containerele