Ingatlan közvetítő weblap

Készítette: Vilmányi Gréta, Major Ákos, Sebestyén László

2025

Tartalomjegyzék

⦁ 1. Bevezetés

⦁ 2. Telepítés és indítás

- 2.1 Telepítés lépései

- 2.2 Szükséges eszközök és szoftverek

- 2.3 Program használata

- 2.4 Admin felhasználó lehetőségei

⦁ 3. Fejlesztői dokumentáció

- 3.1 Fejlesztőkörnyezet

- 3.2 Algoritmusok

- 3.3 Tesztelés

⦁ 4. Fejlesztési lehetőségek

⦁ 1 Bevezetés

Az elkészült program egy ingatlan közvetítő vállalkozás weboldalát mutatja be, mely letisztult megjelenésével könnyen kezelhetőbb az internetes oldalakat kevésbé ismerő felhasználók részére is. Lehetőséget biztosít ingatlanok keresésére, megtekintésére és kapcsolatfelvételre az ingatlan tulajdonosaival.

Célunk, hogy egy modern, reszponzív és felhasználóbarát weboldalt hozzunk létre, amely segíti a vállalkozást az online jelenlétének javításában és a potenciális vásárlókhoz való hatékonyabb eljuttatásában. Emellett a weboldalt későbbi rendeléses oldalakhoz is fel lehet használni, vagy akár azokra integrálható.

Ez a dokumentáció bemutatja a weboldal tervezési és kivitelezési folyamatát, valamint annak potenciális hasznosulási lehetőségeit a vállalkozás számára. Tárgyalja a fejlesztés lépéseit, az alkalmazott technológiákat és a felmerülő kihívásokat.

⦁ 2 Telepítés és indítás

-2.1 Telepítés lépései:

* git clone <url> segítségével lemásoljuk a repository-t
* megnyitjuk visual studio code-ban a projektet tartalmazó mappát
* Ezt követően a következő terminál parancsok kiadása szükséges:

Virtuális környezet aktiválása:

.\env\Scripts\activate (Windows operációs rendszer esetén)

source env/bin/activate (Linux/Mac operációs rendszer esetén)

Csomagok telepítése:

pip install –r requirements.txt

Backend szerver indítása:

cd backend

python manage.py runserver

Egy újabb terminálban elindítjuk a frontend szervert:

cd frontend

npm install

npm run dev

-2.2 Szükséges eszközök és szoftverek

A weblap használatához csak internetkapcsolatra és internet kapcsolattal rendelkező eszközre van szükség. (számítógép, mobiltelefon, tablet )

- 2.3 Program használata

A program használata egyszerű: a felhasználók böngészhetnek az ingatlanok között, választhatnak különböző feltételek szerint. Amennyiben szeretnének, regisztrálhatnak is, így a későbbi keresések mentésével időt takaríthatnak meg.

- 2.4 Admin felhasználó lehetőségei

Az admin-admin felhasználónév-jelszó megadása után a felhasználónak lehetőségük van az összes rendelés kezelésére, felhasználói profilok módosítására és új hirdetések hozzáadására.

⦁ 3 Fejlesztői dokumentáció

- 3.1 Fejlesztőkörnyezet

A weblap fejlesztése során a következő eszközöket és technológiákat használtuk:

• Hardver: Számítógép, laptop

• Programozási technologiák: HTML, CSS, JavaScript, Python, Mysql, Django

• Szoftverek:Visual studio code,

Hardver: Számítógép, laptop erőforrások:

Minimum rendszerkövetelmény:

⦁ Processzor:1 gigahertzes (GHz) vagy gyorsabb processzor vagy SoC

⦁ RAM:1 gigabájt (GB) 32 bites rendszerhez vagy 2 GB 64 bites rendszerhez

⦁ Merevlemez-terület:16 GB 32 bites operációs rendszerhez vagy 20 GB 64 bites operációs rendszerhez

⦁ Videokártya:A DirectX 9-es vagy újabb verziójával kompatibilis, WDDM 1.0 illesztőprogrammal

⦁ Kijelző:800x600

Ajánlott rendszerkövetelmény:

⦁ Processzor: 1 GHz-es vagy gyorsabb, legalább 2 magos, kompatibilis, 64 bites processzor vagy egylapkás rendszer (SoC)

⦁ Memória: 4 GB RAM

⦁ Tárhely: legalább 64 GB

⦁ Videokártya: DirectX 12-kompatibilis vagy újabb, WDDM 2.0 típusú illesztőprogrammal

⦁ Képernyő: Legalább 9" képátlójú, színcsatornánként 8 bites, nagy felbontású (720p) kijelző

- 3.2 Algoritmusok

Az alkalmazás érdekesebb algoritmusai közé tartoznak:

- Felhasználók ellenőrzése:

A projektben a rest keretrendszerben elérhető simplejwt autentikációt használtunk, JWT (JSON Web Token) autentikációs osztály került beállításra.

Ez azt jelenti, hogy az API végpontokhoz történő hozzáférés JWT tokenekkel történik.

django backend mappában szereplő settings.py INSTALLED\_APPS kódrészlete:

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'rest\_framework',

    'rest\_framework\_simplejwt',

    'corsheaders',

    'rentApp',

]

REST\_FRAMEWORK = {

    'DEFAULT\_AUTHENTICATION\_CLASSES': (

        'rest\_framework\_simplejwt.authentication.JWTAuthentication',

    ),

}

backend ->rentApp mappában views.py tartalma:

from django.shortcuts import render

from rest\_framework.views import APIView

from rest\_framework.response import Response

from rest\_framework.permissions import IsAuthenticated

from rest\_framework import generics

from django.contrib.auth.models import User

from .serializers import UserSerializer, ApartmentSerializer

from .models import Apartment

class RegisterUser(APIView):

    def post(self, request):

        serializer = UserSerializer(data=request.data)

        if serializer.is\_valid():

            serializer.save()

            return Response({'message': 'User created successfully'})

        return Response(serializer.errors, status=400)

class ApartmentListCreate(generics.ListCreateAPIView):

    queryset = Apartment.objects.all()

    serializer\_class = ApartmentSerializer

    #permission\_classes = [IsAuthenticated]

A RegisterUser view egy felhasználó regisztrációját végző API végpont, amely a UserSerializer segítségével dolgozza fel az adatokat.

Az ApartmentListCreate view egy olyan API nézet, amely lehetővé teszi az apartmanok listázását és létrehozását.

- 3.3 Tesztelés

Az alkalmazás tesztelése során a következő körülményeket vizsgáltuk: • Különböző böngészők (Chrome, Firefox, Safari) • Különböző eszközök (számítógép, mobiltelefon, tablet) • Adatbázis hibakezelés

⦁ 4 Az alkalmazás továbbfejlesztésének néhány lehetséges területe:

Felhasználói élmény javítása: A felhasználói felület továbbfejlesztése és optimalizálása. ⦁ További funkciók hozzáadása: Új funkciók bevezetése, például automatikus mentés, promóciók kezelése stb. ⦁ Teljesítményoptimalizálás: Az adatbázis lekérdezések és az alkalmazás teljesítményének optimalizálása. ⦁ Kiterjesztett támogatás: Több nyelv támogatása, visszahívás kérése.