

Szegedi Tudományegyetem
Informatikai Intézet

**Fogászati rendelő webalkalmazás
implementálása Django keretrendszerben**

Szakedolgozat

Készítette:

Sándor Márton

programtervező informatikus BSc
szakos hallgató

Témavezető:

Antal Gábor

egyetemi docens

Szeged

2025

Feladatkiírás

A szakdolgozat célja, hogy a hallgató megismerkedjen az egyik legismertebb Python alapú webes keretrendszerrel, a Django-val. A megszerzett ismeretei segítségével a hallgató egy fogászati szakrendelőnek tervezett időpontfoglalási rendszert épít, amelyben elérhető a PayPal fiókos, és a bankkártyás fizetés, és az időpontfoglalás. A rendelő orvosainak lehetőségük van a saját munkaidejük bevitelére, a páciensek kezeléstörténetének megtekintésére, és a magukhoz foglalt időpontok megtekintésére. Továbbá bármelyik páciens adatait és elérhetőségeit megtekinthetik. A rendszer emellett küld email-es értesítéseket is, hogy a felhasználók és az orvosok is kapjanak információkat a foglalásokról. Az alkalmazás tartalmaz admin felületet is, amivel a felhasználónak joga van új orvosokat hozzáadni a rendszerhez, a felhasználóknak új jelszót adni, és az adatbázisban tárolt kezeléseket létrehozni, szerkeszteni, vagy törölni. A felhasznált adatbázis kezelő rendszer a Django beépített rendszere, ami sqlite3 adatbázist használ. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat

quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Tartalmi összefoglaló

A téma megnevezése:

Fogászati rendelő webalkalmazás implementálása Django keretrendszerben

A megadott feladat megfogalmazása:

A Django keretrendszer által nyújtott előnyök megismerése, felhasználása a fogászati rendelő időpontfoglalási rendszerének fejlesztéséhez. Az alkalmazásnak rendelkeznie kell felhasználói, orvosi, és adminisztrátori felülettel. Meg kell valósítani az online időpontfoglalás, és az online fizetés lehetőségét. Az orvosoknak meg kell valósítani egy felületet amivel vissza tudják nézni a páciens kezeléstörténetét. Az adatokat pedig FHIR szabvány szerint kell tárolni, hogy bármilyen más egészségügyi rendszerrel kompatibilis legyen.

A megoldási mód:

Megismertem a Django keretrendszerrel, annak felépítésével. Megterveztem a rendelő webalkalmazásának adatbázis-struktúráját, amely tartalmazza a páciensek, orvosok, kezelések és időpontfoglalások moduljait. Az alkalmazás reszponszív felhasználói felületét modern CSS megoldásokkal építettem ki, míg a backend részben a Django admin és REST API-k biztosítják az adatok hatékony kezelését. Emellett integráltam a PayPal fizetési rendszert az online fizetések lebonyolításához. A fejlesztés során a projektet GitHubon verzióztam, biztosítva ezzel a kód stabilitását és könnyű karbantarthatóságát.

Alkalmazott eszközök, módszerek:

Git, Github, Django, SQLite, HTML, JavaScript, CSS, Python

Elért eredmények:

Megismertem a Django-t, és az általa nyújtott lehetőségeket, a PayPal integrációt, a Python nyelvet, az FHIR szabványt, és a webfejlesztést. Sikeresen működik az általam integrált időpontfoglalási és fizetési rendszer.

Kulcsszavak:

Django, SQLite, HTML, JavaScript, CSS, Python, ORM

Tartalomjegyzék

Feladatkiírás	1
Tartalmi összefoglaló	3
Tartalomjegyzék	5
1. A Django keretrendszer	6
1.1. Az MVT programszervezési minta	6
1.2. Model	6
1.3. View	7
1.4. Template	7
2. Alkalmazás struktúrája	9
3. Összefoglaló	12
Nyilatkozat	14
Köszönetnyilvánítás	15
Irodalomjegyzék	16

1. fejezet

A Django keretrendszer

A Django egy magas szintű Python webkeretrendszer, amely támogatja a gyors fejlesztést és az egyszerű, jól átgondolt megoldásokat. Tapasztalt fejlesztők által készített, így számos webfejlesztési nehézséget megold, és lehetővé teszi, hogy a fejlesztő alkalmazás írására koncentráljon, anélkül, hogy újra fel kellene találnia a kereket. További pozitívuma, hogy ingyenes, és nyílt forráskódú.[1]

1.1. Az MVT programszervezési minta

Django projekt lévén az alkalmazás az MVT (Model View Template) design pattern alapelveit kell, hogy kövesse. Ez áll a model-ből, ahol az adatbázis struktúráját építjük fel, a view-ből, ami lényegében a projekt azon része, ahol a háttérfolyamatok futnak, és a template-ből, ami a felhasználói felületet tartalmazza. Ez a kapcsolata a felhasználónak az alkalmazással.

1.2. Model

A Model-ek a Django alkalmazáson belüli adatszerkezet kezelését és interakcióját irányítják, így a Django alkalmazások alapját képezik, mivel az adatok kritikus szerepet játszanak. A Django Model-ek egy erőteljes, Objektum-Relációs Leképezést (ORM: Object-Relational Mapping) megvalósító funkciót használnak, amely áthidalja a szakadékot a relációs adatbázis és a Python kód között. Ez a leképezés a Python objektumokat (osz-

tályokat) adatbázis táblákká alakítja, az osztályok attribútumait oszlopokká, és az egyes példányokat a táblák soraivá. Az ORM egyik nagy előnye, hogy lehetővé teszi az adatbázissal való interakciót Python objektumokon keresztül, anélkül, hogy SQL lekérdezéseket kellene írunk. A Django Model-ek összegzik az összes adatbázissal kapcsolatos logikát és meghatározzák az adatbázis szerkezetét, mint egy tervrajzot annak, hogy milyen adatokat szeretnénk tárolni.[2]

1.3. View

Ha az MVC modellhez szeretnénk hasonlítani, akkor az MVT modellben a View hasonló, mint az MVC-ben a Controller. A Django view-k felelősek a felhasználói kérések feldolgozásáért és a válaszok visszaküldéséért. Híd szerepét töltik be a Model és a Template között: Adatokat gyűjtenek a Model-ből, logikai műveleteket (például bizonyos kritériumok alapján végzett lekérdezéseket) hajtanak végre rajtuk, majd az eredményeket átadják a Template-nek a megjelenítéshez. A View-kat függvényekként vagy osztály alapú View-ként is megírhatjuk, attól függően, hogy az alkalmazásunk komplexitása és követelményei melyik megközelítést igénylik.[2]

1.4. Template

A Django Template-ek feladata, hogy a böngészőben megjelenítendő végső HTML kimenetet rendereljék. Meghatározzák, miként kell az adatokat bemutatni, HTML és a Django sablonnyelvének kombinációjával. A Django sablonnyelv template tageket (A sablonok továbbá formázhatók CSS-sel, illetve bármely kedvelt CSS keretrendszerrel, hogy a felhasználói felület még szebb legyen. Emellett animálhatók is JS segítségével.[2]

Az 1.1. ábra, és az 1.1. ábra.

Ez még nem készült..

TODO: A mondat befejezése

1.1. ábra. Egy nagyon szép kép



2. fejezet

Alkalmazás struktúrája

Egy Django projekt esetében a projekt felépítése modulárisan, egy vagy több alkalmazásból (app) áll, melyek mindegyike egy adott funkcionális területért felel. A szakdolgozatomban esetében a "rendelo" mappa tartalmazza a teljes webalkalmazás forráskódját.

A "rendelo" mappa a következő részekből áll:

- Gyökérszint:
 - manage.py: A Django projekt parancssori kezelője, amely a fejlesztési feladatok (például migrációk futtatása, szerver indítása) végrehajtását segíti.
 - db.sqlite3: Az alapértelmezett, fejlesztési környezetben használt SQLite adatbázis fájlja.
- Projekt főkönyvtára ("rendelo"): Itt találhatóak a projekt globális beállításait és konfigurációs fájljait, mint például a settings.py, urls.py, wsgi.py és asgi.py. Ezek a fájlok felelősek az alkalmazás működésének alapvető paramétereinek meghatározásáért, az útvonalak kezeléséért és a szerverrel való kommunikációért.
- Alkalmazás könyvtára ("rendeloweboldal"): Ez a rész tartalmazza a rendszer egyes moduljait, amelyek a következő fő komponensekből állnak:
 - models.py: Az adatbázis szerkezetét definiáló model-ek, melyek meghatározzák a páciensek, orvosok, kezelések, időpontfoglalások és az időpont foglalások fizetési státuszának struktúráját.

- views.py: A felhasználói kérések feldolgozásáért és az üzleti logika megvalósításáért felelős fájl, amely összeköti a modelleket a sablonokkal.
- forms.py: Az űrlapok és azok validációs szabályainak definíciója, melyek révén az adatbevitel és ellenőrzés történik.
- urls.py: Az alkalmazás specifikus URL-konfigurációja, amely a különböző view-k elérését biztosítja.
- admin.py: A Django beépített adminisztrátori felület konfigurációját tartalmazza. Hozzá kell adni az összes model-t az adatbázisból, amit elérhetővé szeretnénk tenni rajta.
- migrations/: Az adatbázis változásait követő migrációs fájlokat tartalmazza, dokumentálva a model-ek módosításait.
- static/ és templates/: A statikus fájlokat (CSS, JavaScript, és az alkalmazás designjához tartozó képek) illetve a HTML template-eket rendszerezi, amelyek a felhasználói felület megjelenítéséért felelősek.

A projekt kialakítása moduláris és átlátható, mely lehetővé teszi a fejlesztés, karbantartás és bővítés egyszerű kezelését. Emellett a projekt verziókezelése a GitHubon történik, így könnyen nyomon követhető az egész alkalmazás fejlesztése.

Kódrészlet 2.1. Hogyan ne írjunk kódot

```
szoveg = "valamiszoveg"
if szoveg[:3] == "val":
    print("igaz")
else:
    print("hamis")
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean

faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kódrészlet 2.2. Miből lehet probléma

```
szoveg = "valamiszoveg"
if szoveg[:3] == "valami":
    print("igaz")
else:
    print("hamis")
```

A 2.1. kódrészlet és a 2.1. kódrészlet.

3. fejezet

Összefoglaló

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

Maecenas dui. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetur lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetur eget, vulputate sit amet, erat.

Donec vitae velit. Suspendisse porta fermentum mauris. Ut vel nunc non mauris pharetra varius. Duis consequat libero quis urna. Maecenas at ante. Vivamus varius, wisi sed egestas tristique, odio wisi luctus nulla, lobortis dictum dolor ligula in lacus. Vivamus aliquam, urna sed interdum porttitor, metus orci interdum odio, sit amet euismod lectus felis et leo. Praesent ac wisi. Nam suscipit vestibulum sem. Praesent eu ipsum vitae pede cursus venenatis. Duis sed odio. Vestibulum eleifend. Nulla ut massa. Proin rutrum mattis sapien. Curabitur dictum gravida ante.

Phasellus placerat vulputate quam. Maecenas at tellus. Pellentesque neque diam, dignissim ac, venenatis vitae, consequat ut, lacus. Nam nibh. Vestibulum fringilla arcu mollis arcu. Sed et turpis. Donec sem tellus, volutpat et, varius eu, commodo sed, lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque enim arcu, suscipit nec, tempus at, imperdiet vel, metus. Morbi volutpat purus at erat. Donec dignissim, sem id semper tempus, nibh massa eleifend turpis, sed pellentesque wisi purus sed libero. Nullam lobortis tortor vel risus. Pellentesque consequat nulla eu tellus. Donec velit. Aliquam fermentum, wisi ac rhoncus iaculis, tellus nunc malesuada orci, quis volutpat dui magna id mi. Nunc vel ante. Duis vitae lacus. Cras nec ipsum.

Nyilatkozat

Alulírott Sándor Márton programtervező informatikus BSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Szoftverfejlesztés Tanszékén készítettem, programtervező informatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Informatikai Intézet könyvtárában, a helyben olvasható könyvek között helyezik el.

Szeged, 2025. május 9.

.....
aláírás

Köszönetnyilvánítás

Morbi sem. Nulla facilisi. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla facilisi. Morbi sagittis ultrices libero. Praesent eu ligula sed sapien auctor sagittis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Donec vel nunc. Nunc fermentum, lacus id aliquam porta, dui tortor euismod eros, vel molestie ipsum purus eu lacus. Vivamus pede arcu, euismod ac, tempus id, pretium et, lacus. Curabitur sodales dapibus urna. Nunc eu sapien. Donec eget nunc a pede dictum pretium. Proin mauris. Vivamus luctus libero vel nibh.

Irodalomjegyzék

- [1] Django: The web framework for perfectionists with deadlines . <https://www.djangoproject.com>.
- [2] How Django's MVT Architecture Works: A Deep Dive into Models, Views, and Templates. <https://www.freecodecamp.org/news/how-django-mvt-architecture-works/>.