Korrik 2025

Sistem menaxhimi qendre shëndetësore

Nensi Dervishhasani

**Përmbledhje projekti**

Ky projekt ka qëllim te krijojë një aplikacion web-based që menaxhon doktorët dhe pacientët e një qendre shëndetësore. Sistemi lejon pacientët të shohin oraret e vizitave të doktorëve të spitalit ndërsa doktorët mund të menaxhojnë pacientët dhe oraret e tyre të vizitave. Administratori mund tw shohw listwn e doktorwve, pacientwve si dhe tw hq pwrdorues nga sistemi. Backend-i është zhvilluar me Spring Boot ndërsa PostgreSQL është përdorur për database.

## ****Objektivat****

Krijimi I një sistemi eficient dhe të sigurt per menaxhimin e doktorëve dhe pacientëve.

I jep mundësinë doktorëve dhe pacientëve të shohin dhe ndryshojnë oraret.

I jep mundësinë pacientëve të shohin cili doktor është I lirë si dhe të shohin recetën e tyre pas vizitës.

Lehteson log-in të përdoruesit duke ndarë rolet(Doktor/Pacient).

## ****Karakteristika kryesore****

### 2.1 Log-in dhe përzgjedhja e rolit

* **Login**: Përdoruesit japin emrin, mbiemrin dhe emailin
* **Përzgjedhja e rolit**: Përdoruesi identifikohet si pacient, doktor ose admin
* **Ruajtja e te dhenave**: Të dhënat dhe rolet e përdoruesve ruhen në database(pervec adminit)

### 2.2 Menaxhimi I doktorëve

* **Profili I doktorit**: Ruan emrin mbiemrin dhe specialitetin
* **Gjendja aktuale**: Doktori del I zënë(busy) ose I lirë(free).
* **Orari**: Afati kohor tregon deri në çfarë ore do të jetë I zënë doktori.
* **Menaxhimi I pacientëve**:Doktori mund të shtojë pacientë në listë ose ti heqi ata.

### 2.3 Menaxhimi I pacientëve

* **Profili I pacientit**: Ruan emrin mbiemrin dhe emailin e pacientit.
* **Oraret**: Pacienteve u lihet nje visit me një nga doktorët në një datë dhe orë specifike.
* **Doktorët e lirë**: Pacientët kanë mundësinë të shohin cili prej doktorëve është I lirë momentalisht.
* **Recetat**: Pas vizitës pacientët mund të shohin recetën e tyre në sistem.

### 2.4 Integrimi me database

* PostgreSQL është përdorur për të dhënat.
* Tabelat përfshijnë:
  + **Users**: Përmban informacionet e log-in dhe rolet.
  + **Doctors**: Përmban detajet për doktorët dhe oraret
  + **Patients**: Përmban detajet për pacientin, orarin e vizitës, dhe informacione për doktorin që do ta vizitojë.

## ****Struktura e sistemit****

### 3.1 Paketa teknologjike

* **Backend:** Spring Boot (Java)
* **Database:** PostgreSQL
* **Build Tool:** Maven
* **IDE:** IntelliJ IDEA
* **ORM:** Spring Data JPA (Hibernate)

3.2 Struktura e paketave

* **entity**: Klasat JPA për hartimin e tabelave të bazës së të dhënave.
* **repository**: Interface që zgjerojnë JpaRepository për operacionet CRUD.
* **service**: Shtresa e logjikës së biznesit që menaxhon rrjedhat e punës dhe rregullat.
* **controller**: Pikat fundore të API-së REST që trajtojnë kërkesat dhe përgjigjet HTTP.
* **dto**: Objekte për transferimin e të dhënave (opsionale, për kapsulimin e kërkesave/përgjigjeve).
* **security**: Logjika bazë për hyrje dhe menaxhimin e roleve.

**Krijimi I Database**

Tabelat: users, doctors, patients, appointments, prescriptions dhe tabela të bashkuara

Lidhjet:

* One-to-one midis User dhe Doctor ose User dhe Patient
* Many to Many midis Doctor dhe Patient
* One to Many midis Patient dhe Appointments
* One to many midis Appointment dhe Prescriptions

**Rrjedha e përdorimit**

### 5.1 Login

* Përdoruesi jep emrin dhe passëordin
* Sistemi kontrollon të dhënat dhe jep rolin.
* Kalimi ne dashboard specifike për rolin.

### 5.2 Dashboard I doktorit

* Aksesimi I orareve të vizitave.
* Sheh dhe menaxhon pacientët.
* Update-on statusin aktual.
* Krijon dhe update-on recetat.

### 5.3 Dashboard I pacientit

* Sheh listën e doktorave të lirë.
* Cakton orare të reja vizitas.
* Sheh vizitën e rradhës.
* Sheh recetën pas vizitës.

5.4 Dashboard I administratorit

* Sheh listen e perdoruesve te programit dhe rolet perkatese
* Mund te kerkoje nje perdorues nga ID
* Krijon/Fshin nje llogari perdoruesi
* Cakton ose ndryshon rolet e perdoruesve

## Mbrojtja e programit

* Passwordet e ruajtura sigurt në database.
* Aksesi I dhënë është në varësi të rolit për të siguruar mbrojtjen e të dhënave.
* Kontroll I input-it të user-it për të parandaluar rreziqet.

## Përmirësime të mundshme

* Shtimi I një front-end UI për eksperiencë të përmirësuar.
* Lajmërime në email për të kujtuar oraret e vizitave.
* Integrimin e kalendarëve për përcaktimin e orareve.
* Dahboard analitik në mënyrë që doktori të ndjek ecurinë e pacientit.

**Përfundimi**

Ky sistem përmban një sistem backend për menaxhimin e doktorëve, pacientëve dhe vizitave të një qendre spitalore. Mbështet një rrjedhë logjike të saktë dhe ofronë mbrojtje në përdorim si dhe akses në bazë të rolit.