

# Web aplikacija za procjenu rizika obolijevanja i praćenje tijeka bolesti probavnog sustava primjenom analize podataka

---

**Vuica, Petra**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:200:706864>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-12-27**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni prijediplomski studij Računarstvo**

**WEB APLIKACIJA ZA PROCJENU RIZIKA  
OBOLIJEVANJA I PRAĆENJE TIJEKA BOLESTI  
PROBAVNOG SUSTAVA PRIMJENOM ANALIZE  
PODATAKA**

**Završni rad**

**Petra Vuica**

**Osijek, 2024.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

<b>Ime i prezime pristupnika:</b>	Petra Vuica
<b>Studij, smjer:</b>	Sveučilišni prijediplomski studij Računarstvo
<b>Mat. br. pristupnika, god.</b>	R4736, 29.07.2021.
<b>JMBAG:</b>	0165092129
<b>Mentor:</b>	prof. dr. sc. Goran Martinović
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Web aplikacija za procjenu rizika obolijevanja i praćenje tijeka bolesti probavnog sustava primjenom analize podataka
<b>Znanstvena grana završnog rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rada:</b>	U teorijskom dijelu rada treba analizirati probleme, izazove i postupke utvrđivanja rizika obolijevanja i praćenja tijeka bolesti probavnog sustava. Na temelju analize stanja u području i postojećih rješenja, potrebno je definirati čimbenike rizika, te kriterije procjene rizika i praćenja tijeka bolesti. Također, treba predložiti i pripremiti za ugradnju u web aplikaciju odgovarajuće postupke analize podataka s ciljem procjene rizika i praćenja tijeka bolesti, kao i postupak stvaranja preporuka u skladu sa stanjem pacijenta. Nadalje, treba definirati funkcionalne i nefunkcionalne zahtjeve, model, arhitekturu i dizajn web
<b>Datum prijedloga ocjene završnog rada od strane mentora:</b>	04.09.2024.
<b>Prijedlog ocjene završnog rada od strane mentora:</b>	Izvrstan (5)
<b>Datum potvrde ocjene završnog rada od strane Odbora:</b>	11.09.2024.
<b>Ocjena završnog rada nakon obrane:</b>	Izvrstan (5)
<b>Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:</b>	12.09.2024.



**FERIT**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

## IZJAVA O IZVORNOSTI RADA

Osijek, 12.09.2024.

**Ime i prezime Pristupnika:**

Petra Vuica

**Studij:**

Sveučilišni prijediplomski studij Računarstvo

**Mat. br. Pristupnika, godina upisa:**

R4736, 29.07.2021.

**Turnitin podudaranje [%]:**

14

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za procjenu rizika obolijevanja i praćenje tijeka bolesti probavnog sustava primjenom analize podataka**

izrađen pod vodstvom mentora prof. dr. sc. Goran Martinović

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
<b>1.1. Zadatak završnog rada</b> .....	<b>1</b>
<b>2. IZAZOVI U LIJEČENJU BOLESTI PROBAVNOG SUSTAVA I STANJE U PODRUČJU</b> .....	<b>3</b>
<b>2.1. Probavni sustav</b> .....	<b>3</b>
<b>2.2. Bolesti probavnog sustava</b> .....	<b>3</b>
2.2.1. Gastroezofagealna refluksna bolest (GERB) .....	3
2.2.2. Gastritis .....	4
2.2.3. Dispepsija .....	5
2.2.4. Crohnova bolest.....	5
2.2.5. Sindrom iritabilnog crijeva.....	6
2.2.6. Celijakija .....	7
<b>2.3. Stanje u području praćenja bolesti probavnog sustava</b> .....	<b>8</b>
<b>2.4. Postojeća rješenja</b> .....	<b>8</b>
2.4.1. Digestive Health Risk Assessments.....	8
2.4.2. Symptomate.....	11
2.4.3. Nerva.....	12
2.4.4. My IBD Care.....	13
2.4.5. Bowelle – The IBS Tracker.....	14
<b>3. MODEL, GRADA I POSTUPCI ANALIZE PODATAKA WEB APLIKACIJE</b> .....	<b>15</b>
<b>3.1. Funkcionalni zahtjevi na web aplikaciju</b> .....	<b>15</b>
3.3.1. Prijava korisnika .....	15
3.3.2. Registracija korisnika.....	15
3.3.3. Upitnik o simptomima korisnika .....	16
3.3.4. Dodjela preporuka s obzirom na odgovore u upitniku .....	16
3.3.5. Pregled pacijenata.....	17
3.3.6. Odjava korisnika.....	18
<b>3.2. Nefunkcionalni zahtjevi web aplikacije</b> .....	<b>18</b>
<b>3.3. Izbor značajki modela</b> .....	<b>18</b>
<b>3.4. Korišteni postupak za analizu podataka</b> .....	<b>19</b>
<b>3.5. Razvoj modela analize podataka</b> .....	<b>20</b>
<b>4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE</b> .....	<b>22</b>
<b>4. 1. Korištene programske tehnologije, jezici i razvojna okolina</b> .....	<b>22</b>
4.1.1. Django .....	22
4.1.2. MVT arhitektura .....	22

4.1.3. Python .....	23
4.1.3. HTML .....	23
4.1.4. CSS .....	23
4.1.5. SQLite .....	23
4.1.6. Visual Studio Code .....	24
<b>4.2. Programsko rješenje na strani korisnika .....</b>	<b>24</b>
4.2.1. Programsko rješenje registracije korisnika .....	24
4.2.2. Programsko rješenje prijave i odjave korisnika.....	26
4.2.3. Programsko rješenje upitnika .....	26
4.2.4. Programsko rješenje zdravstvenog kartona pacijenta .....	30
<b>4.3. Programsko rješenje na strani poslužitelja .....</b>	<b>32</b>
4.3.1. Prikaz preporuka.....	32
4.3.2. Baza podataka.....	34
<b>5. PRIKAZ RADA WEB APLIKACIJE; ANALIZA REZULTATA .....</b>	<b>35</b>
<b>5.1. Prikaz rada web aplikacije .....</b>	<b>35</b>
5.1.1. Prikaz postupka za registriranje korisnika .....	35
5.1.2. Prikaz postupka za prijavu korisnika .....	36
5.1.3. Prikaz postupka za ispunjavanje upitnika .....	37
5.1.4. Prikaz preporuka.....	37
5.1.5. Prikaz postupka za zdravstveni karton pacijenta.....	38
<b>5.2. Primjeri korištenja s analizom rezultata.....</b>	<b>39</b>
5.2.1. Prvi primjer korištenja aplikacije .....	40
5.2.2. Drugi primjer korištenja aplikacije .....	41
5.2.3. Treći primjer korištenja aplikacije.....	42
<b>5. ZAKLJUČAK.....</b>	<b>44</b>
<b>LITERATURA .....</b>	<b>45</b>
<b>SAŽETAK .....</b>	<b>48</b>
<b>ABSTRACT .....</b>	<b>49</b>
<b>ŽIVOTOPIS .....</b>	<b>50</b>
<b>PRILOZI.....</b>	<b>51</b>

## 1. UVOD

Probavni sustav je složeni skup organa i procesa koji igra ključnu ulogu u održavanju zdravlja čovjeka. On je odgovoran za unos hrane, njezinu razgradnju u hranjive tvari, apsorpciju tih tvari u krvotok te izbacivanje neprobavljenih tvari iz tijela. Problemi s probavnim sustavom mogu imati ozbiljne posljedice na kvalitetu života i opće zdravlje, uključujući simptome kao što su mučnina, povraćanje, bolovi u trbuhu i druge smetnje koje mogu ukazivati na prisutnost bolesti. Zdrav probavni sustav je neophodan za cjelokupno zdravlje organizma, no mnogo ljudi zanemaruju simptome poremećaja ili ih ne prepoznaju na vrijeme. Redoviti pregledi i konzultacije s liječnikom ključni su za učinkovito liječenje. Razvoj aplikacije za praćenje simptoma probavnog sustava pomaže pacijentima da bolje razumiju svoje zdravstveno stanje i omogućuje im da pravovremeno zatraže liječničku pomoć.

Glavni cilj završnog rada je istražiti i analizirati pravila i tehnike analize podataka za potrebe razvoja web aplikacije za procjenu rizika obolijevanja i praćenja tijeka bolesti probavnog sustava. Na temelju tih pravila i tehnika, bit će izrađeno programsko rješenje koje omogućuje praćenje simptoma i generiranje preporuka. Web aplikacija, pod nazivom *GastroRisk*, izrađena je u programskom okviru Django i programskom jeziku Python. Glavna funkcionalnost web aplikacije su ispitivanje simptoma pacijenta kroz ispunjavanje upitnika, a zatim se pomoću logističke regresije stvaraju odgovarajuće preporuke za korisnike. Spomenute funkcionalnosti, postupci i tehnologije su detaljno opisani u radu.

U drugom poglavlju rada bit će navedene najčešće bolesti probavnog sustava. Uz to analizira probleme i izazove u liječenju bolesti probavnog sustava te pruža pregled postojećih rješenja i tehnologija u tom području. Treće poglavlje, opisuje funkcionalne i nefunkcionalne zahtjeve web aplikacije te postupke analize podatka korištene za procjenu rizika obolijevanja od određene bolesti. U četverom će poglavlju biti detaljno opisane korištene tehnologije, programski jezici i alati za razvoj aplikacije, zajedno s implementacijom korisničkog sučelja i funkcionalnosti, na strani poslužitelja, te uz prikaz arhitekturi aplikacije. U petom poglavlju rada prikazano je korištenje aplikacije i njena analiza kroz prikladne slučajeve korištenja.

### 1.1. Zadatak završnog rada

U teorijskom dijelu rada potrebno je analizirati probleme, izazove i postupke utvrđivanja rizika obolijevanja i praćenja tijeka bolesti probavnog sustava. Na temelju analize stanja u području i

postojećih rješenja, potrebno je definirati čimbenike rizika, te kriterije procjene rizika i praćenja tijeka bolesti. Također, potrebno je predložiti i pripremiti za ugradnju u web aplikaciju odgovarajuće postupke analize podataka s ciljem procjene rizika i praćenja tijeka bolesti, kao i postupak stvaranja preporuka u skladu sa stanjem pacijenta. Nadalje, definirat će se funkcionalni i nefunkcionalni zahtjevi modela, arhitektura i dizajn web aplikacije na strani korisnika i poslužitelja. Web aplikacija treba omogućiti registriranje i prijavu korisnika (liječnika i pacijenta), izradu osobnog profila pacijenta, unos i pohranu podataka, određivanje rizika obolijevanja, praćenje tijeka bolesti u vremenskoj domeni te stvaranje i prikaz preporuka pacijenta usmjerenih načinu života i prehrani. Web aplikaciju treba ostvariti koristeći prikladne programske jezike, tehnologije i razvojne okvire te je ispitati na odgovarajućim ulaznim podacima i slučajevima korištenja.



## **2. IZAZOVI U LIJEČENJU BOLESTI PROBAVNOG SUSTAVA I STANJE U PODRUČJU**

U ovom poglavlju objasnit će se probavni sustav, bolesti koje ga pogađaju, stanje u području praćenja bolesti probavnog sustava te će biti navedena moguća rješenja.

### **2.1. Probavni sustav**

Probavni sustav [1] obavlja važan posao za tijelo. To je usklađeni sustav ljudskog tijela u kojem se odvija proces probavljanja hrane. Njegova uloga je kemijski i mehanički razgraditi hranu na komponente koje se apsorbiraju u krvotok, kako bi putem krvi dospjele do svake stanice u tijelu. Duljina probavnog sustava može varirati od osobe do osobe. Prema [2], prosječna duljina probavnog sustava je između 7,5 i 8,5 metara, dok kod nekih ljudi može biti čak oko 9 metara.

Probavni sustav sastoji se od gastrointestinalnog trakta [3] – koji se još naziva GI trakt ili probavni trakt, te jetre, gušterače i žučnog mjehura. GI trakt je niz šupljih organa spojenih u dugu, zavojitu cijev od usta do anusa. Jetra, gušterača i žučni mjehur su čvrsti organi probavnog sustava. Ovaj sustav započinje u ustima, gdje se hrana žvače na male komadiće. Žlijezde u obrazima i ispod jezika proizvode slinu koja oblaže hranu, olakšavajući žvakanje i gutanje. Nakon što se hrana prožvače i formira bolus, ona ulazi u jednjak, cijev koja prenosi hranu iz usta u želudac. U želucu se hrana miješa s želučanom kiselinom koja je razgrađuje i priprema za daljnju probavu i apsorpciju vitamina i minerala. Iz želuca, hrana odlazi u tanko crijevo, koje je kod odrasle osobe dugačko oko 5 metara i sastoji se od mnogih zavoja i nabora. Velika površina tankog crijeva omogućuje učinkovitu apsorpciju proteina, masnih kiselina, šećera, vitamina i minerala u krv. Preostali nepotrebn materijal zatim ulazi u debelo crijevo. Neprobavljena vlakna se miješaju sa sluzi i bakterijama koje pomažu u njihovoj razgradnji, održavajući zdravlje debelog crijeva. Čvrsti izmet ili stolica formira se u posljednjem dijelu debelog crijeva, rektumu, prije nego što napusti tijelo kroz anus tijekom pražnjenja crijeva [4].

### **2.2. Bolesti probavnog sustava**

U sljedećem dijelu rada, objasnit će se bolesti koje pogađaju probavni sustav, a to su gastroezofagealna refluksna bolest, gastritis, dispepsija, Chronova bolest, sindrom iritabilnog crijeva te celijakija.

#### **2.2.1. Gastroezofagealna refluksna bolest (GERB)**

Gastroezofagealna refluksna bolest (GERB) je stanje u kojem dolazi do povratka želučanog sadržaja u jednjak i usta, a kod nekih bolesnika i u gornje dišne puteve [5]. Povratak želučanog

sadržaja događa se zbog oslabljenog funkcionalnog donjeg ezofagealnog sfinktera (LES), ventila koji obično sprječava povratak želučanog sadržaja u jednjak.

Glavni simptom GERB-a je žgaravica, koja se opisuje kao vatrenu osjećaj u prsima, uz vraćanje kisele tekućine u grlo ili usta. Kombinacija žgaravice i regurgitacije toliko je karakteristična za GERB da nekad posebni testiranje nije potrebno. Osim žgaravice, simptomi GERB-a mogu uključivati nepekuću bol u prsima, koja se obično nalazi u sredini prsa i širi se prema leđima te poteškoće pri gutanju (disfagiju). Također, mogu biti prisutni atipični simptomi povezani s grlom, grkljanom ili plućima, uključujući grlobolju, kašalj, povećano slinjenje i kratkoća daha [6].

### **2.2.2. Gastritis**

Gastritis [7] se javlja kada se želučana sluznica upali ili natekne. Može biti izazvan infekcijom bakterije *Helicobacter pylori*, konzumiranjem alkohola, uzimanjem određenih lijekova, osobito nesteroidnih antireumatika. Također, može biti uzrokovan stresom i autoimunim poremećajima. Prema [8], dijeli se na dvije grupe prema histološkom nalazu: akutni i kronični. Akutni gastritis pojavljuje se iznenada i traje kratko, dok se kronični gastritis razvija postepeno i može trajati dug vremenski period. Ako se akutni gastritis ne liječi na vrijeme može postati kroničan. Također, još se može podijeliti i prema stupnju oštećenja želučane sluznice. Kada je želučana sluznica prekrivena erozijama ili čirevima riječ je o erozivnom gastritisu. Kod neerozivnog gastritisa, želučana sluznica je upaljena, ali nije prikrivena erozijama ili čirevima. Ovo vrsta gastritisa može biti kronična i često je povezana s infekcijom *Helicobacter pylori*.

Gastritis ne mora uzrokovati nikakve vidljive simptome. Ako se simptomi pojave, to može ukazivati na ozbiljniji i duži problem. Simptomi se mogu pojaviti kada je sluznica želuca dovoljno istrošena da se više nije u stanju braniti od vlastitih kiselina i enzima. Kiseline mogu izazvati simptome probavnih smetnji i mogu izazvati čir na želucu, koji može boljeti i krvariti. Simptomi gastritisa uključuju gubitak apetita ili osjeća sitosti nakon jela, bolove u trbuhu, nadutost te mučninu i povraćanje. Ako je prisutan čir, moguće je primijetiti crnu krv u stolici ili crnu krv u povraćanju.

Liječenje gastritisa razlikuje se od osobe do osobe i ovisi o uzorku. Ako je gastritis uzrokovan infekcijom *Helicobacter pylori*, preporučuje se uzimanje jednog ili više antibiotika. Također, preporučuje se izbjegavanje masne, začinjene i kisele hrane što može pomoću u smanjenju iritacije želučane sluznice te uzimanjem više manjih obroka tijekom dana. Savjetuje se potpuni prestanak pušenja i konzumiranje alkohola.

### **2.2.3. Dispepsija**

Dispepsija [9] je medicinski termin koji se koristi za opis boli u gornjem dijelu trbuha. Javlja se češće kod odraslih osoba i nije isto što i žgaravica. Dispepsija može biti uzrokovana različitim zdravstvenim problemima, stilovima života ili uzimanjem određenih lijekova. Zdravstveni problemi uključuju razna stanja kao što su čirevi ili ranice u želucu ili tankom crijevu, gastritis, gastroezofagealna refluksna bolest, bakterijska infekcija u želucu uzrokovana bakterijom *Helicobacter pylori* i upala žučnog mjehura. Loša probava može biti rezultat loših navika u stilu života. To uključuje prekomjerno konzumiranje alkohola, pušenje, prebrzo jedenje, prejedanje, konzumiranje masne i teške hrane te hrane bogate vlaknima, kao i izloženost stresu. Osim toga, određeni lijekovi mogu utjecati na probavni sustav. Pojedini antibiotici koji se koriste za liječenje bakterijskih infekcija, aspirin i lijekovi protiv bolova i vrućica mogu uzrokovati različite probavne smetnje.

Simptomi dispepsije mogu se razlikovati kod svake osobe. Neki od najčešćih simptoma su osjećaj sitosti prerano tijekom jela, osjećaj boli, žarenja i nelagode u gornjem dijelu trbuha, osjećaj nadutosti, povraćanje te proljev.

Kako bi se izliječila dispepsija moguće je poduzeti različite mjere. Potrebno je prilagoditi prehranu, odnosno izbjegavati hranu koja je začinjena i masna te smanjiti unos hrane koja uzrokuje nadutost. Važno se pridržavati redovitih obroka tijekom dana i ograničiti unos kofeina, gaziranih pića i alkohola. Promjena u načinu života mogu doprinijeti poboljšanju probavnog sustava. Redovna tjelesna aktivnost je ključna u poticanju probave i održavanju zdravlja.

### **2.2.4. Crohnova bolest**

Crohnova bolest je vrsta kronične upalne bolesti crijeva koja uzrokuje oticanje i nadraženost probavnog trakta. To je autoimuni poremećaj, što zapravo znači da imunološki sustav tijela greškom napada zdravo tkivo u svom tijelu. Crohnova bolest većinom se javlja u ranoj životnoj dobi, otprilike jedna šestina pacijenata razvije simptome prije 15. godine. Uzroke ove bolesti nije poznat, ali liječnici sumnjaju na genetski utjecaj, budući da su većinom pogođeni članovi iste obitelji [10]. Gen koji se smatra odgovorni za razvoj Crohnove bolesti naziva se NOD2/CARD15 i smješten je na 16. kromosomu [11].

Najčešći simptomi Crohnove bolesti uključuju grčeve u truhu (obično lokalizirani u donjem desnom dijelu trbuha), proljev, zakašnjeni rast (kod mlađe djece), koji se često javlja prije početka gastrointestinalnih simptoma, gubitak težine, vrućica, anemija, rane na ustima, umor te pojava groznica. Ovisno koji je dio gastrointestinalnog trakta zahvaćen, postoje različiti tipovi Crohnove

bolesti. Svaki podtip bolesti ima specifične probleme. Pacijenti ponekad mogu imati simptome koji su problematičniji od problema sa crijevima. Tu su simptomi koji uključuju kronični artritis, bubrežne kamence i komplikacije kod mokraćnog sustava.

Trenutno ne postoji lijek za Crohnovu bolest. Postoje različite terapije, ali ni jedna nije učinkovita za sve. Potrebno se savjetovati s liječnikom kako bi se pronašao pravi tretman kako bi život s Crohnovom bolesti bio podnošljiviji. Liječenje Crohnove bolesti uključuje lijekove, kao što su naprimjer kortikosteroidi, koji smanjuju upalu sluznice crijeva. Uz uzimanje određenih lijekova, prehrana igra važnu ulogu u održavanju i smirivanju bolesti. Važno je piti puno tekućine zbog proljeva, preporučuje se jesti pet manji obroka tijekom dana i smanjiti unos masne i pržene hrane. Također, potrebno je smanjiti unos mlijeka i mliječnih prerađevina te vlaknaste hrane poput orašastih plodova, sjemenki, kukuruza i kokica, jer mogu uzrokovati grčeve i proljev. Poželjno je izbjegavanje alkohola, kofeina, gaziranih pića i složenih šećera te povrća koje iritira sluznicu kao što su brokula, karfiol, poriluk, kupus i kelj.

### **2.2.5. Sindrom iritabilnog crijeva**

Sindrom iritabilnog crijeva (IBS) je kronični poremećaj koji zahvaća debelo crijevo [12]. Sindrom iritabilnog crijeva je čest. Pogađa otprilike dvostruko više žena nego muškaraca, a najčešće se pojavljuje kod osoba mladih od 45 godina.

Točan uzrok sindroma nije poznat, a smatra se kao posljedica višestrukih vanjskih i unutarnjih čimbenika. Mnoga istraživanja otkrila su da je ovaj sindrom mnogo kompleksniji nego što se ranije mislio te da ni jedna teorija o njegovom nastanku nije isključena, već se međusobno nadopunjuju. Kao i kod većine bolesti, genetska predispozicija igra važnu ulogu, ali nije jedini faktor u nastanku IBS-a. Glavni mehanizmi koji doprinose razvoju ovoga sindroma uključuju nekoliko važnih aspekata. Prvi je poremećaj odnosa središnjeg i crijevnog živčanog sustava. Bilo kakav disbalans ovih sustava može dovesti do problema. Drugi mehanizam je poremećaj proizvodnje neurotransmitera u crijevima, koji su ključni za pravilnu komunikaciju između živčanih stanica i koordinaciju crijevnih funkcija. Kronična upala sluznice, također igra značajnu ulogu. Ova upala može narušiti normalnu funkciju crijeva i pridonijeti pojavi simptoma. Psihološki faktori, kao što su stres i anksioznost, također su prepoznati kao važni faktori u razvoju i pogoršanju ovog sindroma. Na kraju, poremećaj crijevnog mikrobioma, odnosno neravnoteža bakterija u crijevima, može dodatno komplicirati stanje i doprinijeti simptomima. Glavni simptom sindroma iritabilnog crijeva je bol u trbuhu ili nelagoda povezana s promjenom crijevnih navika. Pacijenti s ovim sindromom mogu opisati ovu nelagodu na različite načine, kao što su oštra bol u trbuhu, nadutost,

osjećaj punoće ili čak peckanje. Osim boli, drugi simptomi IBS-a uključuje pojavu sluzi u stolici, osjećaj hitnosti za odlazak na toalet te osjećaj nepotpune evakuacije nakon pražnjenja crijeva. Ovi simptomi mogu značajno utjecati na svakodnevni život i kvalitetu života pacijenta. Osobe koje boluje od ovoga sindroma također mogu doživjeti simptome koji nisu direktno povezani s crijevima. Među te sindrome spadaju migrenske glavobolje, poremećaji spavanje, anksioznost i kronična bol u zdjelici. Ovi simptomi često pridonose složenosti stanja i zahtijevaju sveobuhvatan pristup liječenju kako bi se ublažili neugodni simptomi i poboljšala kvaliteta života pacijenta. Pretjeran unos masnoća, prejedanje, neredovito i brzo konzumiranje hrane te prekomjeren unos alkohola, kave i gaziranih pića utječu na razvoj bolesti i pogoršanje simptoma [13].

Cilj liječenja sindroma iritabilnog crijeva je olakšavanje simptoma, a pristup se prilagođava vrsti i težini simptoma pacijenta. Uspjeh liječenja ovisi o dobrom razumijevanju IBS-a i otvorenoj komunikaciji s liječnikom. Liječenje IBS-a može uključivati promjene u prehrani, lijekove, psihoterapiju i alternativne terapije. Promjene u prehrani mogu pomoći kod nadutosti, bolova u trbuhu i proljeva, smanjenjem unosa kofeina i masne hrane, te prilagodbom unosa ugljikohidrata. Lijekovi za liječenje IBS-a uključuju relaksante glatkih mišića, lijekove protiv proljeva, antibiotike i niske doze antidepresiva. Psihoterapija može pomoći u upravljanju simptomima povezanim s stresom i emocionalnim stanjima.

### **2.2.6. Celijakija**

Celijakija [14] je autoimuni poremećaj zbog kojeg tijelo ne može tolerirati gluten, protein koji se nalazi u pšenici, ječmu i raži. Procjenjuje se da pogađa 1% svjetske populacije i može se javiti kod bilo koga, bez obzira na dob. Kada osobe s celijakijom konzumiraju gluten, dolazi do oštećenja tankog crijeva. To može dovesti do povraćanja, mučnina, plinova, bolova u trbuhu i drugih simptoma. Određeni faktori kao što su previše stresa, trudnoća, kirurški zahtjevi, tjelesne ozljede, infekcije i porođaj mogu potaknuti pojavu simptoma. Celijakija pogađa ljude na različite načine. Neki pokazuje simptome još kao djeca, dok ih drugi tek mogu razviti u odrasloj dobi. Simptomi celijakije mogu biti raznoliki, neki od njih su kronični proljev ili zatvor, gubitak težine, nadutost ili plinovi.

Uobičajeni znakovi celijakije uključuju blijedu stolicu s neugodnim mirisom, anemiju koja uzrokuje umor, izostanak menstruacije, neplodnost, osteoporoza u mladosti ili lomljive kosti, promjene u boji zuba ili gubitak cakline. Djeca s celijakijom mogu zaostati u rastu. Čovjek može imati celijakiju, ali da nema nikakve simptome. To je zato što je jedan dio tankog crijeva nije ozlijeđen i još uvijek može primiti dovoljno hranjivih tvari.

Ako je pacijentu ima celijakiju, potrebno je odmah prestati jesti gluten. Konzumiranje glutena dodatno oštećuje tanko crijevo. U većini slučajeva izbacivanjem glutena iz prehrane zaustavit će simptome ove bolesti i svako oštećenje tankog crijeva će se izliječiti. Nakon prestanka konzumiranja hrane s glutenom, simptomi će se poboljšati za nekoliko dana do tjedana, a tanko crijevo bi trebalo zacijeliti za 3 do 6 mjeseci. U nekim slučajevima može proći dulje za ozdravljenje. Ako je pacijent starije životne dobi, može proći i do 2 godine da se tijelo opрави.

### **2.3. Stanje u području praćenja bolesti probavnog sustava**

Praćenje bolesti probavnog sustava postalo je ključno u modernoj medicini, jer rana detekcija i precizna prognoza uvelike može poboljšati ishode liječenja i pozitivno utjecati na kvalitetu života pacijenta. Korištenjem digitalnih tehnologija, rad medicinskog osoblja je olakšan, štednjom vremena na preglede i povećanjem efikasnosti. Umjetna inteligencija i algoritmi strojnog učenja postaju sve važniji alat u predikciji i upravljanju gastroenterološkim bolestima. Prema [15], primjenom algoritama umjetne inteligencije razvijaju se sklopovsko-programska rješenja za predviđanje i praćenja gastroenteroloških bolesti. Posebna pažnja posvećena je algoritmima kao što su *k-Nearest Neighbors* (k-NN), *Support Vector Machine* (SVM), umjetne neuronske mreže i *Random Forest*, koji se primjenjuju na podatke pacijenta. Njihova primjena omogućuje točnu dijagnozu i prognozu, smanjujući mogućnost grešaka u dijagnostičkom postupku. Upotrebom digitalnih tehnologija, prethodno navedenih algoritama i modernih programskih jezika poput Pythona omogućuju medicinskom osoblju da u kratkom vremenu postave sveobuhvatnu dijagnozu.

Prema [16], duboko učenje kao grana umjetne inteligencije nudi revolucionarne mogućnosti u dijagnostici i liječenju bolesti probavnog sustava. Duboko učenje može poboljšati dijagnostiku bolesti kao što su polipi, čirevi i karcinomi, kroz analizu endoskopskih slika, te automatizirati histopatološke procese, smanjujući mogućnost grešaka u dijagnozi.

## **2.4. Postojeća rješenja**

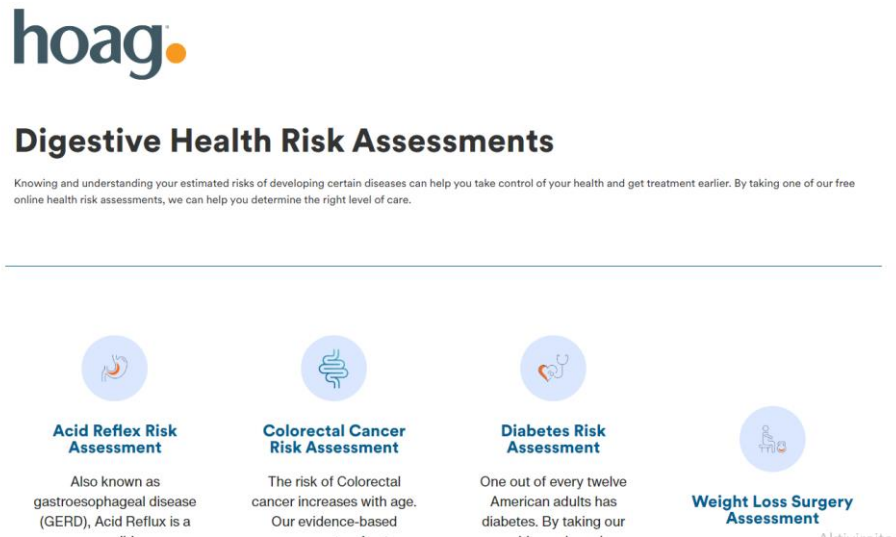
### **2.4.1. Digestive Health Risk Assessments**

Digestive Health Risk Assessments su aplikacije koje pomažu korisnicima da procjene rizik od različitih bolesti probavnog sustava na temelju njihovih simptoma i zdravstvenih navika.

Aplikacije su razvijene od strane Hoag Medical Group (HMG). HMG[17] je grupacija liječnika sa specijalizacijom u različitim područjima.

Njihovi stručnjaci koriste znanstvene studije i klinička istraživanja kako bi razvili algoritme i modele koji mogu analizirati podatke o simptomima i prepoznati obrasce povezane s određenim probavnim bolestima.

Pristupom u aplikaciju, korisniku se prikazuje početna stranica koja nudi četiri dostupne procjene rizika za različita oboljenja. Početna stranica aplikacije prikazana je na slici 2.1. Na raspolaganju su procjene rizika za GERB-a, raka debelog crijeva, dijabetesa i anoreksije.



Slika 2.1. Prikaz početne stranice *Digestive Health Risk Assessments* aplikacije

Kada korisnik odabere željenu procjenu, u ovom slučaju je odabrana opcija *Acid Reflex Risk Assessment*, otvara se stranica za unos osobnih podataka, uključujući godine, spol, visinu, težinu i email adresu. Nakon popunjavanja svojih podataka, sustav postavlja još dva pitanja, vezana za narodnost i o trajanju pušenja (ili nepušenja). Također, korisnik će biti upitan je mu ikad dijagnosticiran GERB i kada je zadnji put posjetio liječnika. Zatim kreću pitanja o simptomima. Pitanja se odnose imali li korisnik simptome, poput žgaravice, mučnine, bolove u gornjem dijelu stomaka i slično. Sustav postavlja 10 pitanja o simptomima. Na kraju upitnika, postavljaju se dva pitanja vezana za životne navike korisnika. Na slici 2.2 prikazana su primjeri pitanja iz aplikacije.

How many times per week does this occur?

Pain in the middle of the upper stomach area

0 days

1 day

2 or 3 days

4 to 7 days

Nausea

0 days

1 day

2 or 3 days

4 to 7 days

*Slika 2.2 Prikaz pitanja u upitniku za ispitivanje simptoma u aplikaciji Digestive Health Risk Assessments*

Odgovaranjem na sva pitanja, korisnik je dužan upisati svoje ime, prezime, poštanski broj i broj telefona. Nakon toga, njegov nalaz će mu biti poslan na adresu e-pošte, a postoji i opcija za kontaktiranje od strane Hoag-a. Prikaz prve stranice nalaza prikazan je slici 2.3.

**hoag** **Acid Reflux Assessment**

Name: \_\_\_\_\_ 8-27-2024  
PAGE 1 OF 4

**YOUR SYMPTOM ASSESSMENT**

**Ask a Doctor About Your Symptoms**

**SYMPTOMS & RISK FACTORS (REPORTED)**

**3 Common Symptoms**

**1 Risk Factor**

**Based on the frequency of your symptoms, acid reflux disease is not likely.**

- Abdominal pain is not a typical symptom of acid reflux disease. Talk to a doctor about the intensity and frequency of this symptom.
- Consider trying more of the healthy habits listed on page two of this report. Reaching a healthy weight may also help reduce or prevent symptoms.

---

**Common Symptoms of Acid Reflux Disease**

The symptoms listed in the table to the right are some of the most common symptoms of acid reflux disease. However, this list does not include all possible symptoms.

If you're worried about whether you have acid reflux disease, or if you have other symptoms that are bothering you, talk with a health care professional about what's going on.

Number of Days Per Week You Experience Symptoms:	
Burning feeling behind the breastbone	1 Day
Stomach contents moving up to the throat or mouth	0 Days
Pain in the middle of the upper stomach area	1 Day
Nausea	0 Days
Trouble getting a good night's sleep because of symptoms	0 Days
Need for over-the-counter medicine for symptoms	1 Day
<b>Total Symptoms Reported:</b>	<b>3</b>

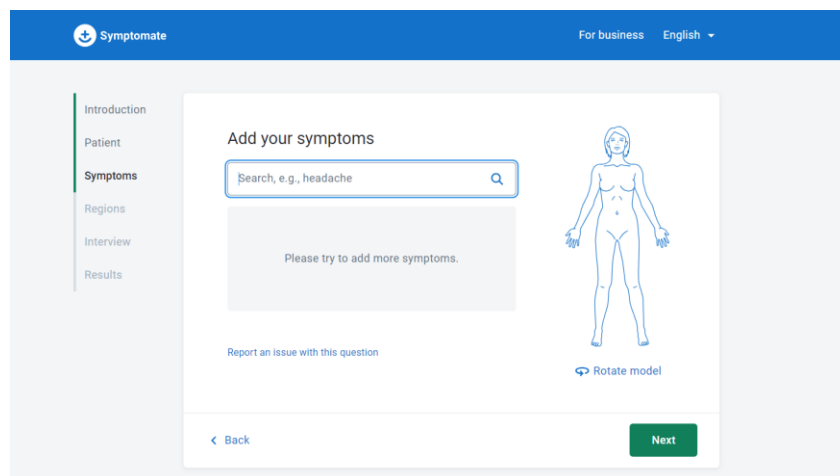
*Slika 2.3 Prikaz nalaza kojeg je generirao sustav Digestive Health Risk Assessments*



## 2.4.2. Symptomate

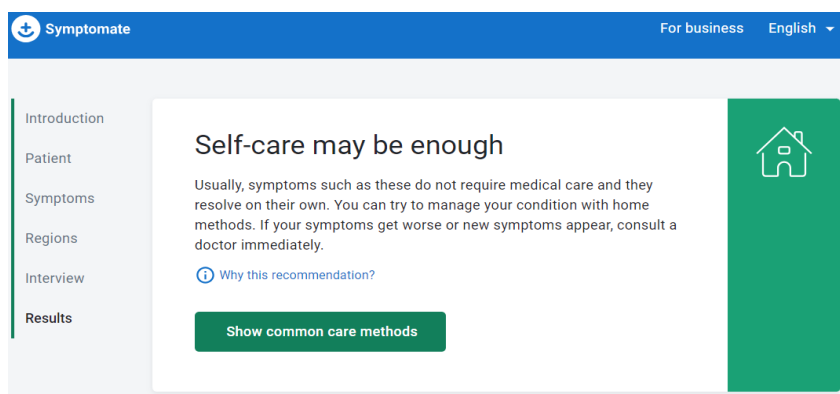
Symptomate [18] je napredni alat za procjenu simptoma koristeći umjetnu inteligenciju za prilagodbu ankete prema specifičnim simptomima korisnika. Razvijen je 2012. godine od strane tvrtke Infermedica. Symptomate kombinira sofisticirani AI motor za donošenje odluke s opsežnom medicinskom bazom podataka, omogućavajući korisnicima da preciznije identificiraju moguće uzroke svojih simptoma. Potpuno je besplatan, s naglaskom na privatnost i sigurnost korisnički podataka.

Prilikom pristupa korisnika njihovom upitniku, prvo im se postavljaju opća pitanja vezana za dob, spol i godinu rođenja te pitanja o osobnom zdravlju, povijesti bolesti i životnim navikama. Nakon toga, korisnici imaju mogućnost dodati simptome koje trenutno osjeća kao što je prikazano na slici 2.4.



Slika 2.4 Prikaz mogućnosti za dodavanje simptoma u aplikaciji Symptomate

Nakon dodavanja simptoma, bit će mu postavljena detaljnija pitanja oko simptoma. Na kraju se kreira preporuka, prikazana na slici 2.5.



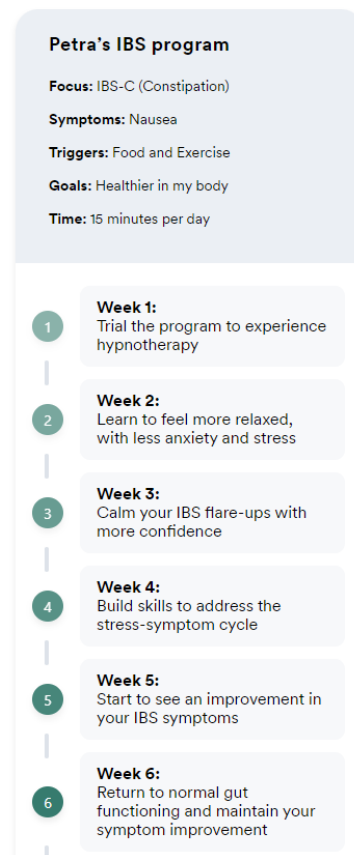
Slika 2.5 Prikaz preporuke koju je generirala aplikacija Symptomate

### 2.4.3. Nerva

Nerva [19] je digitalna aplikacija namijenjena osobama s iritabilnim crijevima, koja primjenjuje dokaznu hipnoterapiju. Kroz aplikaciju, korisnici usvajaju tehnike za upravljanje stresom, što može poboljšati komunikaciju između crijeva i mozga te olakšati simptome. Program je osmislila dr. Simone Peters, koja je provela kliničko istraživanje u suradnji sa Sveučilištem Monash i otkrila da njen pristup učinkovito pomaže ljudima da kvalitetnije žive sa sindromom iritabilnog crijeva.

Nerva funkcionira kroz nekoliko jednostavnih koraka. Prvo, korisnici ispunjavaju kratki kviz koji će procijeniti je li Nerva pravi izbor za njih. Zatim, slušaju hipnoterapije koju su razvili stručnjaci i istraživači. Kroz ovaj proces, uče tehnike za obuku mozga da ignorira lažne signale iz crijeva koje hrana i stres mogu izazvati. Na kraju, program podučava korisnike kako dugo upravljati simptomima iritabilnog crijeva, poboljšavajući kvalitetu života i smanjujući učestalost neugodnih epizoda.

Na slici 2.6 prikazan je šesto tjedni program za upravljanje sindromom iritabilnog crijeva. Program je nastao kao rezultat ispunjavanja kviza.

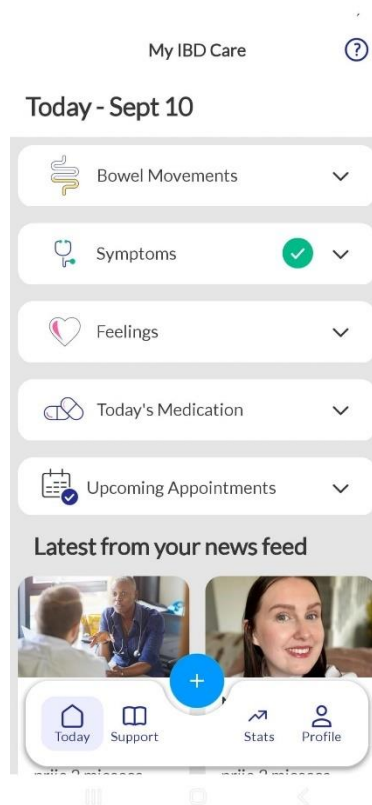


Slika 2.6 Prikaz programa za upravljanje sindroma iritabilnog crijeva

#### 2.4.4. My IBD Care

My IBD Care je mobilna aplikacija osmišljena za osobe koje boluju od upalnih bolesti crijeva, kao što su Chronova bolest i ulcerozni kolitis. Aplikaciju je razvila tvrtka Ampersand u suradnji s vodećim medicinskim stručnjacima [20].

Korisnici aplikacije mogu bilježiti svoje simptome, uključujući bol, umor, promjene u probavi i raspoloženju, što im omogućuje bolje razumijevanje tijeka bolesti i uzroka pogoršanja. Aplikacija nudi podsjetnike za uzimanje lijekova, čime omogućava da korisnici prate svoje planove liječenja. Također, pruža korisne informacije i savjete u vezi s prehranom, načinom života i upravljanjem stresom kako bi pomogla u ublažavanju simptoma. Korisnici mogu dijeliti podatke o svom zdravstvenom stanju s liječnicima ili medicinskim timom. Aplikacija omogućava i postavljanje osobnih zdravstvenih ciljeva te praćenje napretka u njihovom postizanju. Na slici 2.7 prikazana je početna stranica aplikacije.

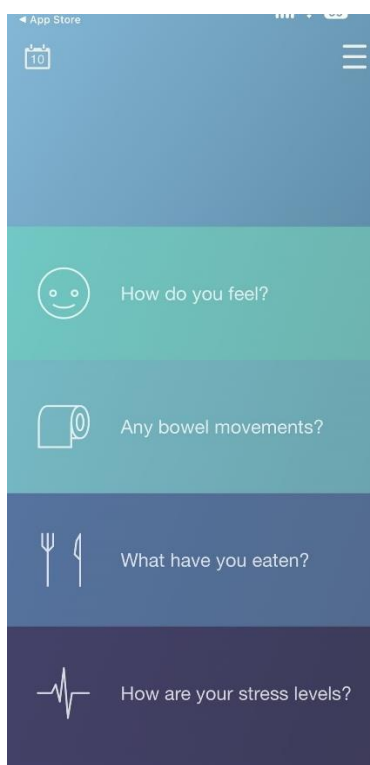


Slika 2.7 Prikaz početne stranice My IBD Care aplikacije

#### 2.4.5. Bowelle – The IBS Tracker

Bowelle – The IBS Tracker [21] je aplikacija koja je namijenjena osobama koje pate od sindroma iritabilnog crijeva i drugih probavnih tegoba. Razvijena je os strane tvrtke Knotable. Glavna svrha aplikacije je omogućiti korisnicima praćenje simptoma povezanih s probavnim sustavom, što može pomoći u identificiranju uzoraka, potencijalnih okidača simptoma i poboljšanju upravljanja bolešću.

Bowelle aplikacija nudi brojne značajke koje olakšavaju praćenje probavnog zdravlja. Početna stranica aplikacije prikazana je na slici 2.8.



Slika 2.8 Prikaz početne stranice Bowelle – The IBS Tracker aplikacije

Aplikacija omogućava detaljno bilježenje simptoma poput nadutosti, boli u trbuhu, zatvora i proljeva, a korisnici mogu zabilježiti kada se i koliko često javljaju. Također, omogućava praćenje intenziteta simptoma tijekom vremena. Druga važna značajka je praćenje unosa hrane kroz dnevnik prehrane, što korisnicima pomaže identificirati namirnice koje izazivaju simptome. Aplikacija generira i grafičke prikaze te statistiku prikupljenih podataka, omogućujući korisnicima da uoče poveznice između prehrane i simptoma, dok personalizirani uvidi pružaju korisne preporuke za bolje upravljanje sindromom iritabilnog crijeva. Dodatno, aplikacija omogućava bilježenje specifičnih događaja vezanih uz zdravlje, poput stres i aktivnosti.

### **3. MODEL, GRAĐA I POSTUPCI ANALIZE PODATAKA WEB APLIKACIJE**

U ovome će poglavlju biti prikazani funkcionalni i nefunkcionalni zahtjevi web aplikacije te postupci analize podataka.

#### **3.1. Funkcionalni zahtjevi na web aplikaciju**

Korisnicima je osiguran jednostavan pristup aplikaciji putem početne stranice koja sadrži dva gumba: jedan za prijavu, a drugi za registraciju. Ako korisnik nema račun, treba prvo ispuniti formu za registraciju i unijeti svoje osobne podatke. Ako posjeduje korisnički račun, prijavljuje se unosom korisničkog imena i lozinke. Ukoliko se korisnik prijavi kao pacijent, početna stranica će mu prikazati tipku za pristup upitniku koja sadrži unaprijed definirana pitanja o simptomima. Korisnik koji se prijavi kao liječnik moći će pregledati zdravstvene kartone svih pacijenata. Svaki zdravstveni karton uključuje osobne podatke pacijenta, sve prethodne odgovore iz upitnika, preporuke koje su proizašle iz tih upitnika te graf koji prikazuje promjene vjerojatnosti bolesti kroz vrijeme. Osim toga, liječnik ima mogućnost slanja vlastitih preporuka svakom pacijentu.

##### **3.3.1. Prijava korisnika**

Prilikom odabiranja tipke za prijavu, treba se prikazati stranica aplikacije koja se sastoji od polja za unos korisničkog imena i lozinke te tipke za prijavu. Ispod navedene tipke nalaziti će se tipka *Ne posjedujete korisnički račun?*, a odabirom te tipke otvarati se stranica na kojoj se nalazi forma za registraciju. Korisnik se može prijaviti u aplikaciju samo ako je prethodno već izradio račun.

##### **3.3.2. Registracija korisnika**

Kada korisnik prvi puta pristupa aplikaciji, potrebno je kreirati korisnički račun za daljnje korištenje. Odabirom tipke za registracije na početnoj stranici ili navigacijskoj traci, otvoriti će se forma za registraciju, koja se sastoji od polja za unos dodatnih informacija kao što su ime, prezime, korisničko ime, zaporke, datuma rođenja, spola i uloge korisnika (liječnik ili pacijent). Ako su sva polja ispravno popunjena, registracija će biti uspješna. Na kraju forme nalazi se tipka za potvrdu, pritiskom te tipke podaci se spremaju u bazu podataka. Ako korisnik odabere opciju *Jeste li liječnik?*, nakon registracije automatski će biti preusmjeren na stranicu početnu stranicu aplikacije s tipkom *Zdravstveni kartoni*, a ako je korisnik pacijent, bit će preusmjeren na početnu stranicu aplikacije i s tipkom *Ispunite upitnik*.

### **3.3.3. Upitnik o simptomima korisnika**

Nakon prijave ili registracije korisnika, otvara se početna stranica. Na početnoj stranici nalazi se gumb *Ispunite upitnik*. pritiskom na taj gumb otvara se upitnik. Svaki korisnik koji je prijavljen kao pacijent, može pristupiti upitniku u bilo kojem trenutku, bez ograničavanja u broju pokušaja. Svaki put kada korisnik preda odgovore, podaci se spremaju u bazu podataka, kao i datum predaje. Korisnik mora odgovoriti na svako pitanje kako bi podaci bili uspješno spremljeni u bazu podataka. Rezultat upitnika bit će prikazan korisniku u dva oblika. Prvi je prikazan u obliku kreirane preporuke, a drugi je prikazan u grafičkom obliku.

### **3.3.4. Dodjela preporuka s obzirom na odgovore u upitniku**

Prema [22], sustav preporuka pomaže korisnicima u pronalaženju njihovih interesa na internetu pružajući im relevantne prijedloge. Osnovna ideja sustava preporuka je povezati korisnike s člancima te odabrati najprikladniji sadržaj za svakog korisnika. Postoje četiri metode kojima sustavi preporuka generiraju popis preporuka za korisnike, a to su: sadržajno filtriranje, kolaborativno filtriranje, demografsko filtriranje i hibridno filtriranje. Sadržajno filtriranje isključivo koristi preferencije korisnika za preporuku sadržaja, uspoređujući profil korisnika s karakteristikama sadržaja članaka koje je korisnik već pogledao, ocijenio ili ispunio. Kolaborativno filtriranje prikuplja i analizira velike količine podataka o ponašanju, aktivnostima i preferencijama korisnika te iskorištava sličnosti u njihovim ocjenama i izborima za preporučivanje sadržaja. Demografsko filtriranje koristi podatke iz korisničkog profila, poput dobi, demografskih informacija, obrazovanje, interesa i komentara, kako bi identificirali korisnike s sličnim interesima te prilagođava preporuke prema dobnoj skupini i životnom prostoru korisnika. Hibridno filtriranje funkcionira kombiniranjem različitih tipova sustava preporuka kako bi postigao što učinkovitije rezultate u nekim slučajevima.

Za izradu ove aplikacije korišteno je sadržajno filtriranje, odnosno preporuke se generiraju na temelju parametara koje je korisnik ispunio. U ovom slučaju, to su odgovori iz upitnika i simptomi koje pacijent prijavljuje.

Nakon ispunjavanja upitnika, odgovori se pohranjuju u bazu podataka. Nakon toga generiraju se preporuke i identificiraju tri najvjerojatnije bolesti. Zatim se korisniku prikazuje preporuka koja mu savjetuje treba li posjetit liječnika s obzirom na njegove simptome. Moguća su tri slučaja koje prikazuje tablica 3.1.

Tablica 3.1. Prikaz preporuka na temelju vjerojatnosti

Postotak	Preporuka
0-39%	Za sada ne morate posjetiti liječnika.
40-79%	Bilo bi dobro da svoje stanje provjerite s liječnikom.
80-100%	Preporučujemo da hitno posjetite liječnika.

Uz te preporuke, sustav prikazuje tri najvjerojatnije bolesti. Važno je napomenuti da sustav ne može sa sigurnošću tvrditi da pacijent boluje od određene bolesti, budući da sustav nije zamjena za liječničku dijagnozu. Pacijentu se također prikazuju i preporuke za način života i prehrane, usmjerene na bolest s najvećom vjerojatnosti. Preporuke za svaku bolest prikazane su u tablici 3.2.

Tablica 3.2. Prikaz preporuka na temelju bolesti

BOLEST	PREPORUKA
GERB	Izbjegavajte masnu i začinjenu hranu. Jedite manje obroke češće umjesto velikih obroka. Izbjegavajte ležanje odmah nakon jela.
Gastritis	Izbjegavajte začinjenu hranu. Jedite manje obroke tijekom dana. Uzmite lijekove koje vam liječnik preporučuje, uključujući antibiotike ako je gastritis uzrokovan bakterijom <i>Helicobacter pylori</i> .
Dispepsija	Izbjegavajte masnu i tešku hranu, ograničite unos alkohola, kave i gaziranih pića. Jedite redovito i izbjegavajte prejedanje. Povećajte unos vlakana u prehrani.
Chronova bolest	Jedite pet manjih obroka tijekom dana, smanjite unos masne i pržene hrane. Izbjegavajte mlijeko i mliječne prerađevine, kao i vlaknaste namirnice. Pijte puno tekućine i izbjegavajte alkohol i gazirana pića.
Sindrom iritabilnog crijeva	Izbjegavajte hranu koja uzrokuje nadutost, kao što su grah i kupus. Uvedite redovite obroke i jedite polako. Razmislite o promjeni prehrane prema FODMAP prehrani.

### 3.3.5. Pregled pacijenata

Kada se korisnik prijavi kao liječnik, otvara mu se početna stranica aplikacije. Na toj se stranici nalazi gumb *Zdravstveni kartoni*, odabirom tog gumba prikazuje se popis pacijenata poredanih abecednim redom po prezimenu. Klikom na određenog pacijenta otvara se njegov zdravstveni karton, u kojem liječnik može vidjeti sve detalje o pacijentu, prethodne ispunjene upitnike,

preporuke koje je sustav generirao, kao i graf koji prikazuje vjerojatnosti svake bolesti kroz vrijeme.

### **3.3.6. Odjava korisnika**

Svaki korisnik se može odjaviti sa svog korisničkog računa pritiskom na tipku za odjavu koja je smještena na navigacijskoj traci. Nakon što se korisnik odjavi, prikazuje se početna stranica aplikacije.

## **3.2. Nefunkcionalni zahtjevi web aplikacije**

Nefunkcionalni zahtjevi [23] definiraju kvalitete i karakteristike programskog sustava, poput performansi, skalabilnosti, upotrebljivosti i sigurnosti. Ova kategorija uključuje zahtjeve kao što su brzo vrijeme odziva, sposobnost upravljanja većim brojem korisnika, intuitivno korisničko sučelje i snažne sigurnosne mjere.

Performanse se odnose na brzinu obrade podataka i vrijeme odziva aplikacije. U ovom slučaju, aplikacija osigurava odziv korisničkih upita unutar jedne sekunde, čak i uz visoko opterećivanje sustava. Skalabilnost sustava omogućuje prilagođavanje većem broju korisnika bez pada u performansama. To se postiže korištenjem optimiziranih algoritama za obradu velikih količina podataka, kao što je generiranje personaliziranih preporuka. Upotrebljivost aplikacije vrlo je jednostavna, zahvaljujući intuitivnim korisničkim sučeljem koje omogućava jednostavno popunjavanje upitnika i dobivanje jasnih povratnih informacija korisnicima. Pouzdanost aplikacije znači da će sustav biti dostupan 99,9% vremena i da se neće gubiti podatke ni u slučaju neočekivanih kvarova. Sigurnost je također ključna komponenta nefunkcionalnih zahtjeva. Aplikacija osigurava zaštitu korisničkih podataka kroz implementaciju snažnih mehanizama autentifikacije i enkripcije podataka. Korisnici se moraju prijaviti putem sigurnosno certificiranog sustava za autentifikaciju kako bi pristupili svojim informacijama, a svi podaci prenose se putem šifriranih kanala.

## **3.3. Izbor značajki modela**

Korisnici koji se prijave kao pacijenti trebaju ispuniti upitnik koja služi za ispitivanja simptoma i životnih navika pacijenta. Pitanja su organizirana u nekoliko kategorija, a svaki odgovor doprinosi sveukupnom razumijevanju zdravstvenog stanja pacijenta. Sva pitanja prikazana su u tablici 3.3.



Tablica 3.3. Prikaz pitanja s ponuđenim odgovorima

PITANJA	PONUĐENI ODGOVRI
Konзумirate li cigarete?	Da / Ne
Konзумirate li alkohol?	Nikad, Rijetko, Ponekad, Često
Imate li osjećaj žgaravice?	Nikad, Rijetko, Ponekad, Često
Imate li neprekidnu bol u sredini prsa koja se širi prema leđima?	Da / Ne
Imate li poteškoće tijekom gutanja (disfagiju)?	Da / Ne
Jeste li ikad imali dijagnozu infekcije bakterije <i>Helicobacter pylori</i> ?	Da / Ne
Koristite li često nesteroidne antireumatike (npr. aspirin, ibuprofen)?	Da / Ne
Imate li bolove u truhu ili nadutost?	Nikad, Rijetko, Ponekad, Često
Imate li osjećaj mučnine i povraćate li?	Nikad, Rijetko, Ponekad, Često
Osjećate li bol u gornjem dijelu truha nakon jela?	Da / Ne
Patite li od proljeva?	Nikad, Rijetko, Ponekad, Često
Osjećate li grčeve u truhu, posebno u donjem desnom dijelu?	Nikad, Rijetko, Ponekad, Često
Jeste li umorni i imate li anemiju?	Da / Ne
Imate li osjećaj hitnosti za odlazak na toalet?	Da / Ne
Gubite li težinu bez ikakvog razloga?	Da / Ne
Jeste li pod stresom?	Nikad, Rijetko, Ponekad, Često
Imate li osjećaj gubitka apetita i punoće?	Nikad, Rijetko, Ponekad, Često

Odgovori na pitanja u upitniku omogućavaju procjenu simptoma i pomažu u stvaranju preporuka.

### 3.4. Korišteni postupak za analizu podataka

Regresija [24] je tehnika koja se koristi u dvije važne svrhe. Prvo, često se primjenjuje za prognozu i predviđanje, a njena primjena česta je u području strojnog učenja. Drugo, regresija može, u određenim slučajevima, pomoći u utvrđivanju odnosa između nezavisnih i zavisnih varijabli. Treba napomenuti da regresija sama po sebi prikazuje odnose između zavisne varijable i skupa različitih varijabli u skupu podataka. Za analizu podataka, najčešće se koriste linearna i logistička regresija. Linearna se regresija koristi za kontinuirane ishode, dok se logistička regresija koristi za binarne ishode. Razlika je u tome što linearna regresija modelira linearni odnos između varijabli, dok logistička regresija modelira logistički odnos.

Linearna regresija je algoritam nadzirnog učenja koji se koristi u prediktivnom modeliranju unutar strojnog učenja. Model predviđa vrijednosti na temelju neovisnih varijabli i pomaže u otkrivanju odnosa između tih varijabli. Analiziranjem linearne regresije, možemo identificirati kako promjene u jednoj ili više nezavisnih varijabli utječu na zavisnu varijablu, što je korisno u

donošenju odluka i razumijevanja obrazaca u podacima. Postoje dvije vrste regresije: jednostavna i višestruka regresija. Jednostavna linearna regresija je model koji ispituje odnos između jedne nezavisne i jedne zavisne varijable, pri čemu se taj odnos prikazuje kao pravac. Višestruka linearna regresija je model koji se koristi za analiziranje odnosa između dvije ili više nezavisnih varijabli i jedne zavisne varijable[25].

Logistička regresija[26] procjenjuje vjerojatnost da će se neki određeni događaj dogoditi, koristeći zadani skup podataka nezavisnih varijabli. Statistički je model te se često primjenjuje u klasifikaciji i prediktivnoj analitici. S obzirom na to da je ishod izražen kao vjerojatnost, zavisna varijabla se kreće unutar raspona od 0 do 1. Matematički izraz za logističku regresiju je:

$$p(y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \quad (3-1)$$

Objašnjenje simbola:

$p$  – vjerojatnost da se dogodi događaj  $Y$

$e$  – baza prirodnog algoritma

$\beta_0$  – konstanta(intercept) u modelu

$\beta_0, \beta_1, \dots, \beta_n$  – koeficijenti regresije koji predstavljaju utjecaj svake neovisne varijable  $X_0, X_1, \dots, X_n$  na ishod

$X_0, X_1, \dots, X_n$  – neovisne varijable koji utječu na zavisnu varijablu  $Y$

### 3.5. Razvoj modela analize podataka

U poglavlju 2 opisano je šest najčešćih bolesti probavnog sustava prema literaturi, a svakom simptomu unutar tih bolesti dodijeljene su odgovarajuće težinske vrijednosti. Manje vrijednosti dodijeljene su simptomima koji su manje specifični za određenu bolest, a veće vrijednosti su dodijeljene onim simptomima koji su izrazitije povezani s određenim bolestima. Sve težinske vrijednosti su konstantne i ne mijenjaju se tijekom korištenja aplikacije. Potrebno je svakom odgovoru iz upitnika dodijeliti određenu vrijednost kako bi se mogao koristiti u matematičkom izrazu za logističku regresiju. U upitniku se koriste dvije vrste odgovora, svaki s pripadajućom vrijednosti. Prvoj vrsti odgovora uključuje opcije *Nikad*, *Rijetko*, *Ponekad*, *Često* dodijeljene su vrijednosti 1, 2, 4 i 5, kako bi se održavala učestalost simptoma, druga vrsta odgovora se sastoji od odgovora *Da* i *Ne*, pri čemu su im dodijeljene vrijednosti 5 i 0, što omogućava jasno razlikovanje prisutnosti ili odsutnosti simptoma. Sustav koristi logističku regresiju za procjenu

vjerojatnosti svake bolesti na temelju težinskih vrijednosti simptoma specifičnih za svaku bolest. Na temelju tih izračuna, sustav može utvrditi koja je bolest najvjerojatnija i koja je preporuka najprikladnija za korisnika. Za najvjerojatniju bolest, sustav generira preporuke koje uključuju savjete za liječenje i nove životne navike. Osim toga, sustav pruža preporuku o potrebi za medicinskom pomoći na temelju visine vjerojatnosti. Sve preporuke, zajedno s odgovorima iz upitnika pohranjuju se u bazu podataka.

## 4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE

U ovome poglavlju objasnit će se i opisati korišteni programski tehnologije, jezici i razvojna okolina te programska rješenja na strani korisnika i poslužitelja.

### 4. 1. Korištene programske tehnologije, jezici i razvojna okolina

#### 4.1.1. Django

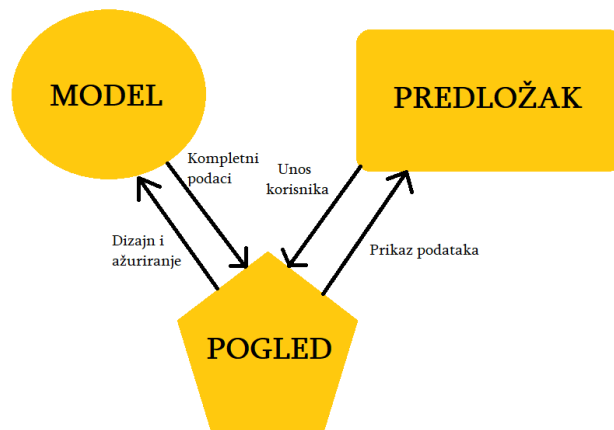
Django [27] je Python web okvir visoke razine koji omogućuje kreiranje web aplikacija, besplatan je i otvorenog koda. Tim iskusnih programera, koji su zaduženi za razvoj i održavanje Django-a nazivaju se Django Software Foundation. Django rješava mnoge izazove u web razvoju kao što je upravljanje bazom podataka i autentifikacija korisnika. Također je poznat po svojoj brzini, sigurnosti i skalabilnosti, zbog čega je vrlo popularan izbor za sve vrste web projekata, od malih stranica do velikih web aplikacija. Temelji se na MVT (Model-View-Template) arhitekturi. U ovom završnom radu za izradu aplikacije korištena je inačica verzija Django 4.2.

#### 4.1.2. MVT arhitektura

MVT je programski obrazac koji se koristi za razvoj web aplikacija. Struktura MVT arhitekture se sastoji od tri dijela:

- Model
- Pogled (engl. *view*)
- Predložak (engl. *template*)

Modeli predstavljaju ključni dio web okvira Django i omogućuju definiranje strukture podataka za aplikaciju. Django modeli pojednostavljaju zadatke i organiziraju tablice u modele. Svaki model preslikava se u pojedinačnu tablicu u bazi podataka. Pogled je korisničko sučelje, preciznije pogled predstavlja ono što se vidi u pregledniku kada se prikazuje web mjesto. Predstavljaju ga HTML, CSS, JavaScript i slične datoteke. Predlošci su treći i najvažniji dio MVT arhitekture. Uglavnom su pisani u HTML-u, CSS-u i JavaScriptu te koriste određenu sintaksu Django template jezika za prikaz dinamičkog sadržaja. Olakšavaju razvoj i održavanje web aplikacije jer omogućuju odvajanje logike prikaza od poslužiteljskog koda [28].



*Slika 4.1. Prikaz strukture MVT arhitekture*

#### 4.1.3. Python

Python [29] je objektno orijentirani, interpretirani programski jezik visoke razine s dinamičkom semantikom. Ima jednostavnu sintaksu, koju je lako naučiti i stoga smanjuje troškove održavanja programa. Python podržava module i pakete, što potiče modularnost programa i ponovnu upotrebu koda. U ovom radu korištena je inačica Python 3.11.9.

#### 4.1.3. HTML

HyperText Markup Language (HTML) [30] je jezik za označavanje koji se koristi pri izradi web stranice. Osim što je jednostavan za korištenje i učenje, HTML je od početka bio besplatan i dostupan svima za korištenje, što je razlog njegove velike popularnosti. Pomoću HTML-a ne možemo obavljati nikakve zadatke, pa ni osnovne operacije kao što su zbrajanje i oduzimanje stoga HTML nije programski jezik. On isključivo služi za opisivanje hipertekstualnih dokumenata. HTML datoteke su ustvari tekstualne datoteke s ekstenzijom .html. Svaka HTML dokument sastoji se od oznaka (*engl.* tag) koji određuju kako će se sadržaj prikazati na web pregledniku.

#### 4.1.4. CSS

Cascading Style Sheets (CSS) [31] je jezik koji se koristi pri oblikovanju web stranice. Pomoću njega, možemo kontrolirati izgled i raspored elemenata unutar HTML dokumenata. Boje, fontove, margine, razmake, poravnanje elemenata i mnoge druge aspekte vizualnog izgleda stranice možemo definirati korištenjem CSS-a. On također omogućava alternativno oblikovanje odnosno prilagodljivost različitim uređajima.

#### 4.1.5. SQLite

SQLite je zadana baza podataka u Django-u, prepoznatljiva je po svojoj maloj veličini i jednostavnosti. Idealna je za razvoj manjih aplikacija jer ne zahtijeva posebnog poslužitelja već

podatke sprema na lokalni disk što omogućuje laku prenosivost i brzu implementaciju. Također, osigurava prijenos koda na veće baze podataka kao što su PostgreSQL ili Oracle.

#### 4.1.6. Visual Studio Code

Visual Studio Code [32] je besplatno razvojno okruženje za uređivanje koda. Raspoloživ je na Windowsima, Linuxu i macOS-u. Savršen je za svakodnevnu upotrebu jer ima brz uređivač izvornog teksta. VS Code, uz podršku za stotine jezika, omogućava visoku produktivnost zahvaljujući isticanju sintakse, automatskim uvlačenjem, podudaranju zagrada i drugim funkcionalnostima. Za ozbiljno kodiranje, nekad je potrebno imati alate koji bolje razumiju kod od samih tekstualnih blokova. VS Code uključuje ugrađeni IntelliSense za dovršavanje koda, semantičko razumijevanje koda i navigaciju te refaktoriranje koda. Također, nudi i interaktivni alat za otklanjanje pogrešaka.

## 4.2. Programsko rješenje na strani korisnika

### 4.2.1. Programsko rješenje registracije korisnika

Registracija korisnika ključna je funkcionalnost koja omogućuje korisnicima stvaranje računa i pristup personaliziranim sadržajima u aplikaciji. Proces registracije obuhvaća unos osobnih podataka korisnika, uključujući informaciju o tome je li korisnik liječnik ili pacijent. Nakon što korisnik ispuni sva polja u formi *RegistrationForm*, podaci se šalju metodi *RegisterView* koja obrađuje formu i provjerava njihovu ispravnost. Ako su svi podaci ispravni, korisnik se pohranjuje u bazu podataka, kreira se odgovarajući profil (liječnika ili pacijenta), a zatim se korisnik automatski prijavljuje pomoću funkcije *login(request, user)* te se preusmjerava na početnu stranicu aplikacije pomoću *HttpResponseRedirect(reverse('home'))*. Ukoliko dođe do pogreške, poput već postojećeg korisničkog imena ili email adrese, korisniku će biti prikazana odgovarajuća poruka o grešci na stranici registracije. Ove provjere provode metode *clean\_username* i *clean\_email* unutar *RegistrationForm*, koji dodaje greške u slučaju da korisničko ime ili email već u uporabi. Cijela *RegistrationForm* nalazi se na slici 4.2.

```

class RegistrationForm(UserCreationForm):
    firstName = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'First name'}))
    lastName = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Last name'}))
    username = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Username'}))
    email = forms.EmailField(widget=forms.EmailInput(attrs={'class': 'form-control', 'placeholder': 'Email'}))
    password1 = forms.CharField(widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': 'Password'}))
    password2 = forms.CharField(widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': 'Confirm password'}))
    gender = forms.ChoiceField(choices=[('M', 'Muški'), ('F', 'Ženski')], widget=forms.Radioselect)
    dateOfBirth = forms.DateField(widget=forms.DateInput(attrs={'class': 'form-control', 'type': 'date'}))
    isDoctor = forms.BooleanField(widget=forms.CheckboxInput(attrs={'class': 'form-check-input', 'type': 'checkbox'}), required=False)

Codeium: Refactor | Explain
class Meta:
    model = User
    fields = ('username', 'firstName', 'lastName', 'email', 'password1', 'password2', 'gender', 'dateOfBirth', 'isDoctor')

Codeium: Refactor | Explain | Generate Docstring | X
def __init__(self, *args, **kwargs):
    self.request = kwargs.pop('request', None)
    super().__init__(*args, **kwargs)

Codeium: Refactor | Explain | Generate Docstring | X
def clean_username(self):
    username = self.cleaned_data.get("username")
    if User.objects.filter(username=username).exists():
        self.add_error(None, "This username is already taken.")
    return username

Codeium: Refactor | Explain | Generate Docstring | X
def clean_email(self):
    email = self.cleaned_data.get("email")
    if User.objects.filter(email=email).exists():
        self.add_error(None, "This email address is already taken.")
    return email

```

Slika 4.2 Prikaz forme za unos osobnih podataka tijekom registracije korisnika

Slika 4.3 prikazuje isječak programskog koda kako se korisnički podaci prikupljeni putem forme obrađuju i spremaju u bazu podataka. Važno je napomenuti kako se lozinka nikad ne sprema u obliku običnog teksta, već pomoću metode `set_password` lozinka se enkriptira prije pohrane. Na taj način osigurana je visoka razina sigurnosti aplikacije.

```

def RegisterView(request):
    context = {}

    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.first_name = form.cleaned_data['firstName']
            user.last_name = form.cleaned_data['lastName']
            username = form.cleaned_data['username']
            raw_password = form.cleaned_data['password1']
            user.set_password(raw_password)
            user.save()

            # Create user profile
            gender = form.cleaned_data['gender']
            dateOfBirth = form.cleaned_data['dateOfBirth']
            isDoctor = form.cleaned_data['isDoctor']
            if isDoctor:
                Doctor.objects.create(user=user, name=username, gender=gender, dateOfBirth=dateOfBirth)
            else:
                Patient.objects.create(user=user, name=username, gender=gender, dateOfBirth=dateOfBirth)
            # Authenticate and login user
            login(request, user)
            return HttpResponseRedirect(reverse('home'))
        else:
            if not form.errors:
                form.add_error(field=None, error='Passwords do not match.')
    else:
        form = RegistrationForm()

    context['form'] = form
    return render(request, 'registration/register.html', context)

```

Slika 4.3 Funkcija `RegisterView` u `views.py`

### 4.2.2. Programsko rješenje prijave i odjave korisnika

Prijava korisnika omogućava autentifikaciju korisnika koji pristupaju sustavu, osiguravajući da samo ovlaštene korisnici mogu pristupiti zaštićenim resursima. Ova aplikacija koristi Django-ove ugrađene alate i funkcionalnosti kako bi omogućila jednostavnu i sigurnu prijavu korisnika.

U datoteci `urls.py` aplikacije, potrebno je uključiti Django-ove URL obrasce za autentifikaciju. Ovo se može postići korištenjem `django.contrib.auth.urls`, kao što je prikazano na slici 4.4. Time se automatski pruža putanje za prijavu ('`login`') i odjavu ('`logout`').

```
from django.contrib import admin
from django.urls import path, include
from app import views
from django.conf import settings
from django.conf.urls.static import static
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.home, name="home"),
    path('accounts/register', views.RegisterView, name='register'),
    path("accounts/", include("django.contrib.auth.urls")),
    path('survey/', views.SurveyView, name='survey'),
    path('result/', views.SurveyView, name='result'),
    path('patient_profile/', views.patient_profile, name='patient_profile'),
    path('patients/', views.patient_list, name='patient_list'),
    path('patients/<int:patient_id>', views.PatientDetail, name='patient_detail'),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Slika 4.4 Prikaz uključivanja Django-vih url obrazaca za autentifikaciju

Ove URL putanje omogućuju jednostavno upravljanje autentifikacijom bez potrebe za pisanjem vlastitih obrazaca i funkcionalnosti za prijavu i odjavu.

### 4.2.3. Programsko rješenje upitnika

Na slici 4.5 prikazan je `SurveyResponse` model, odnosno klasa koja predstavlja strukturirane podatke prikupljene putem upitnika. Model uključuje različita polja s pripadajućim izborima. Osim unosa simptoma, model također sadrži polje `recommendation`, koje služi za pohranu generiranih preporuka nakon obrade podataka.



```

class SurveyResponse(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    date = models.DateField(auto_now=True)
    smoking = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    alcohol = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    heartburn = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    chest_pain = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    dysphagia = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    h_pylori = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    nsaid = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    abdominal_pain = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    nausea_vomiting = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    postprandial_pain = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    diarrhea = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    cramps = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    fatigue_anemia = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    urgency = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    weight_loss = models.CharField(max_length=3, choices=[('Da', 'Da'), ('Ne', 'Ne')])
    stress = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    appetite_loss = models.CharField(max_length=7, choices=[('Često', 'Često'), ('Ponekad', 'Ponekad'), ('Rijetko', 'Rijetko'), ('Nikad', 'Nikad')])
    recommendation = models.TextField(null=True, blank=True)

    def __str__(self):
        return f"SurveyResponse {self.id} for {self.user.username} on {self.date}"

```

Slika 4.5 Programski kod za spremanja rezultata upitnika o zdravstvenom stanju pacijenta

*SurveyForm*, prikazan na slici 4.6 povezuje model *SurveyResponse* s HTML formom prikazanom korisniku. U ovom slučaju, sva polja upitnika prikazuju se korisnicima kao radio tipke, omogućujući da odaberu jednu od unaprijed definiranih opcija. Uloga *SurveyForm* je ključna jer osigurava da se korisnički unosi ispravni i ispravno pohrane u model *SurveyResponse*.

```

class SurveyForm(forms.ModelForm):
    class Meta:
        model = SurveyResponse
        fields = ['smoking', 'alcohol', 'heartburn', 'chest_pain', 'dysphagia', 'h_pylori', 'nsaid', 'abdominal_pain', 'nausea_vomiting', 'postprandial_pain', 'diarrhea', 'cramps', 'fatigue_anemia', 'urgency', 'weight_loss', 'stress', 'appetite_loss']
        widgets = {
            'smoking': forms.RadioSelect,
            'alcohol': forms.RadioSelect,
            'heartburn': forms.RadioSelect,
            'chest_pain': forms.RadioSelect,
            'dysphagia': forms.RadioSelect,
            'h_pylori': forms.RadioSelect,
            'nsaid': forms.RadioSelect,
            'abdominal_pain': forms.RadioSelect,
            'nausea_vomiting': forms.RadioSelect,
            'postprandial_pain': forms.RadioSelect,
            'diarrhea': forms.RadioSelect,
            'cramps': forms.RadioSelect,
            'fatigue_anemia': forms.RadioSelect,
            'urgency': forms.RadioSelect,
            'weight_loss': forms.RadioSelect,
            'stress': forms.RadioSelect,
            'appetite_loss': forms.RadioSelect,
        }
        labels = {
            'smoking': 'Konzumirate li cigarete?',
            'alcohol': 'Konzumirate li alkohol?',
            'heartburn': 'Imate li osjećaj žgaravice (osjećaj vraćanja kisele tekućine u grlo ili usta)?',
            'chest_pain': 'Imate li neprekidnu bol u sredini prsa koja se širi prema leđima?',
            'dysphagia': 'Imate li poteškoće tijekom gutanja (disfagiju)?',
            'h_pylori': 'Jeste li ikad imali dijagnozu infekcije bakterije Helicobacter pylori?',
            'nsaid': 'Koristite li često nesteroidne antireumatike (npr. aspirin, ibuprofen)?',
            'abdominal_pain': 'Imate li bolove u truhu ili nadutost?',
            'nausea_vomiting': 'Imate li osjećaj mučnine i povraćate li?',
            'postprandial_pain': 'Osjećate li bol u gornjem dijelu trbuha nakon jela?'
        }

```

Slika 4.6 Programski kod za unos rezultata upitnika o zdravstvenom stanju korisnika

Funkcija *SurveyView*, prikazana na slici 4.7, obrađuje HTTP zahtjeve povezane s upitnikom. Kada korisnik ispuni upitnik, *SurveyView* prima POST zahtjev i provjerava valjanost forme koristeći metodu *is\_valid()*. Ukoliko je forma ispravna, kreira se novi *SurveyResponse* objekt koji se povezuje s trenutno prijavljenim korisnikom. Nakon toga, koristi se funkcija *generate\_top\_n\_recommendations*, prikazana na slici 4.8.

```
def SurveyView(request):
    if request.method == 'POST':
        form = SurveyForm(request.POST)
        if form.is_valid():
            survey_response = form.save(commit=False)
            survey_response.user = request.user

            survey_response.save()
            # Generiraj preporuke i popis najvjerojatnijih bolesti
            recommendations, top_diseases = generate_top_n_recommendations(survey_response, n=3)

            survey_response.recommendation = '\n'.join(recommendations)
            survey_response.save()

            return render(request, 'app/recommendation.html', {
                'recommendations': recommendations,
                'top_diseases': top_diseases,
            })
        else:
            form = SurveyForm()
            return render(request, 'app/survey.html', {'form': form})
```

Slika 4.7 Programski kod za pogled upitnika

```
def generate_top_n_recommendations(survey_response, n=3):
    values = map_answers_to_values(survey_response)
    probabilities = generate_probabilities(values)

    sorted_diseases = sorted(probabilities, key=probabilities.get, reverse=True)

    highest_probability_disease = sorted_diseases[0]
    highest_probability = probabilities[highest_probability_disease]

    urgency_recommendation = generate_disease_recommendation(highest_probability)

    top_diseases = [f"{disease} s vjerojatnošću {probabilities[disease]:.4f}." for disease in sorted_diseases[:n]]

    recommendations = [urgency_recommendation]
    recommendations.append(f"Najvjerojatnija bolest: {highest_probability_disease} s vjerojatnošću {highest_probability:.4f}.")
    recommendations.extend(get_recommendations_for_condition(highest_probability_disease))

    return recommendations, top_diseases
```

Slika 4.8 Programski kod za stvaranje preporuka

Ona koristi unesene podatke kako bi generirala preporuke i identificirala najvjerojatnije bolesti. Te se preporuke zatim pohranjuju u *recommendation* polje unutar objekta *SurveyResponse*. Funkcija *generate\_top\_n\_recommendations* koristi mapiranje korisničkih odgovora u numeričke vrijednosti koje omogućuju daljnje izračune, a ovaj postupak se provodi putem funkcije *map\_answers\_to\_values*. Funkcija je prikazana na slici 4.9.

```

VALUE_MAP = {
    'Često': 5,
    'Ponekad': 4,
    'Rijetko': 2,
    'Nikad': 1,
    'Da': 5,
    'Ne': 0,
}

Codeium: Refactor | Explain | Generate Docstring | X
def map_answers_to_values(survey_response):
    values = {
        'smoking': VALUE_MAP.get(survey_response.smoking, 0),
        'alcohol': VALUE_MAP.get(survey_response.alcohol, 0),
        'heartburn': VALUE_MAP.get(survey_response.heartburn, 0),
        'chest_pain': VALUE_MAP.get(survey_response.chest_pain, 0),
        'dysphagia': VALUE_MAP.get(survey_response.dysphagia, 0),
        'h_pylori': VALUE_MAP.get(survey_response.h_pylori, 0),
        'nsaids': VALUE_MAP.get(survey_response.nsaids, 0),
        'abdominal_pain': VALUE_MAP.get(survey_response.abdominal_pain, 0),
        'nausea_vomiting': VALUE_MAP.get(survey_response.nausea_vomiting, 0),
        'postprandial_pain': VALUE_MAP.get(survey_response.postprandial_pain, 0),
        'diarrhea': VALUE_MAP.get(survey_response.diarrhea, 0),
        'cramps': VALUE_MAP.get(survey_response.cramps, 0),
        'fatigue_anemia': VALUE_MAP.get(survey_response.fatigue_anemia, 0),
        'urgency': VALUE_MAP.get(survey_response.urgency, 0),
        'weight_loss': VALUE_MAP.get(survey_response.weight_loss, 0),
        'stress': VALUE_MAP.get(survey_response.stress, 0),
        'appetite_loss': VALUE_MAP.get(survey_response.appetite_loss, 0),
    }
    return values

```

Slika 4.9 Prikaz funkcije koja pretvara odgovore u numeričke vrijednosti

Nakon mapiranja odgovora, funkcija koristi unaprijed definirane težine povezane sa povezani s različitim simptomima i bolestima. Te težine, su definirana na temelju medicinskog znanja i pomažu u procjeni koliko svaki simptom utječe na vjerojatnost određene bolesti. Dio dodijeljenih težina prikazano je na slici 4.10.

```

DISEASE_SYMPTOM_WEIGHTS = {
    'GERB': {
        'smoking': 0.2,
        'alcohol': 0.2,
        'heartburn': 6.0,
        'chest_pain': 5.0,
        'dysphagia': 5.0,
        'h_pylori': 0.0,
        'nsaids': 0.2,
        'abdominal_pain': 1.0,
        'nausea_vomiting': 1.0,
        'postprandial_pain': 0.5,
        'diarrhea': 0.1,
        'cramps': 0.3,
        'fatigue_anemia': 0.4,
        'urgency': 0.1,
        'weight_loss': 0.1,
        'stress': 1.0,
        'appetite_loss': 0.4,
    },
    'Gastritis': {
        'smoking': 1.2,
        'alcohol': 1.1,
        'heartburn': 1.3,
        'chest_pain': 0.9,
        'dysphagia': 0.4,
        'h_pylori': 2.5,
        'nsaids': 2.2,
        'abdominal_pain': 3.5,
        'nausea_vomiting': 2.3,
        'postprandial_pain': 1.4,
        'diarrhea': 0.7,
        'cramps': 0.6
    }
}

```

Slika 4.10 Dodjela simptomima definirane težine

Zatim funkcija `generate_probabilities` prolazi kroz sve potencijalne bolesti i računa vjerojatnost za svaku od njih koristeći mapirane numeričke odgovore i odgovarajuće unaprijed definirane težine simptoma za svaku bolest. Ovo se ostvaruje pozivanjem funkcije `calculate_probability`. nakon izračunavanja vjerojatnosti za sve bolesti, funkcija vraća rječnik `probabilities`, koja sadrži vjerojatnosti za svaku bolest.

Funkcija `calculate_probability` izračunava ukupnu vrijednost  $z$ . Za svaki simptom, njegova se težina množi s odgovarajućom vrijednošću iz korisničkog unosa, a svi se rezultati zbrajaju. Nakon što se izračuna zbroj, provodi se prilagodba vrijednosti  $z$ , kako bi se rezultati zadržali u razumljivim granicama. Nakon toga, prilagođena vrijednost  $z$  se prosljeđuje funkciji logističke regresije koja vraća vjerojatnost pojave bolesti na temelju simptoma. Funkcija za računanje vjerojatnosti i funkcija logističke regresije su prikazane na slici 4.11.

```
def logistic_regression(z):
    return 1 / (1 + math.exp(-z))

Codeium: Refactor | Explain | Generate Docstring | X
def calculate_probability(disease_weights, values):
    z = sum(disease_weights[symptom] * values[symptom] for symptom in values)
    z -= 55.0
    z /= len(values)

    return logistic_regression(z)
```

Slika 4.11 Programski kod za računanje vjerojatnosti bolesti

Na kraju, funkcija `generate_top_n_recommendations` vraća listu od tri najvjerojatnije bolesti i pripadajuće preporuke. Navedena lista bude prikazana pomoću HTML predloška.

#### 4.2.4. Programsko rješenje zdravstvenog kartona pacijenta

Kada se korisnik prijavi kao liječnik u aplikaciju, ima mogućnost da pregleda popis svih pacijenata. Prikazani programski kod na slici 4.12, omogućuje prikaz svih pacijenata sortiranih po prezimenu i imenu, uz osigurava da samo prijavljeni korisnici mogu vidjeti ovu listu.

```
def patient_list(request):
    patients = Patient.objects.all().order_by('user__last_name', 'name')
    return render(request, 'app/patient_list.html', {'patients': patients})
```

Slika 4.12 Prikaz svih pacijenata sortiranih abecedno po prezimenu

Klikom na ime i prezime svakoga pacijenta otvara se stranica koja prikazuje detalje o tom pacijentu. Zatim, funkcija `PatientDetail` unutar `views.py` prikazana na slici 4.13, dohvaća sve relevantne podatke o pacijentu koristeći njegov ID. Ova funkcija koristi `get_object_or_404` kako

bi pronašla objekt *Patient* koji odgovara zadanom *patient\_id*. Ako pacijent s tim id-em ne postoji, funkcija će vratiti 404 stanicu.

Nakon što je pacijent identificiran, funkcija prikuplja sve prethodne upitnike tog pacijenta koristeći *SurveyResponse.objects.filter(user=patient.user).order\_by('-date')*. Za svaki upitnik, koristi se funkcija *generate\_recommendation* kako bi generirala preporuke na temelju rezultata upitnika. Rezultati preporuka se pohranjuju u rječnik *recommendations*, gdje ključ upitnika predstavlja ID upitnika, a vrijednost tekstualnu preporuku.

Iako liječnik i pacijent vide istu stranicu, ali s jednom razlikom. Liječnik može poslati poruku pacijentu i vidjeti poruke koje je poslao, dok pacijent može samo pregledavati poruke, ali ne može slati nove. Liječnik šalje nove poruke putem obrasca *MessageForm*, prikazan na slici 4.13, koji se prikazuje na stranici. Ova forma omogućava unos sadržaja poruke i koristi *Textarea* widget za unos teksta.

```
class MessageForm(forms.ModelForm):
    class Meta:
        model = Message
        fields = ['content']
        widgets = {
            'content': forms.Textarea(attrs={'rows': 3, 'placeholder': 'Unesite poruku ovdje...'}),
        }
```

Slika 4.13 Programski kod forme za slanje poruke

Kad liječnik pošalje poruku putem POST zahtjeva, *MessageForm* se koristi za spremanje nove poruke u bazu podataka. Poruka se sprema pozivom metode *message.save()*, kao što je prikazano na slici 4.14. Nakon toga se korisnik preusmjerava natrag na stranicu s detaljima pacijenta.

```
def PatientDetail(request, patient_id):
    patient = get_object_or_404(Patient, id=patient_id)
    survey_responses = SurveyResponse.objects.filter(user=patient.user).order_by('-date')

    recommendations = {}
    for response in survey_responses:
        recommendation = generate_recommendation(response)
        recommendations[response.id] = recommendation

    messages = Message.objects.filter(sender=request.user, recipient=patient.user)
    messages = messages.order_by('timestamp')

    if request.method == 'POST':
        form = MessageForm(request.POST)
        if form.is_valid():
            message = form.save(commit=False)
            message.sender = request.user
            message.recipient = patient.user
            message.save()
            return redirect('patient_detail', patient_id=patient_id)
    else:
        form = MessageForm()
```

Slika 4.14 Programski kod pogleda za zdravstveni karton pacijenta

Uz sve prethodne informacije, stranica prikazuje i graf koji vizualizira promjene vjerojatnosti bolesti kroz vrijeme. Ovaj graf pomaže liječnicima i pacijentima da bolje razumiju kako se simptomi i vjerojatnosti bolesti mijenjaju tijekom vremena na temelju prethodnih upitnika. Programski kod za generiranje grafa prikazuje se na slici 4.15.

Za generiranje grafa koristi se nekoliko koraka. Prvo, prikupljaju se svi rezultati upitnika za pacijente i generirane preporuke potrebne za graf. Datumi upitnika i vjerojatnosti za različite bolesti prikupljaju se i organiziraju se u odgovarajuće strukture podataka. Koristeći biblioteku *matplotlib*, kreira se linijski graf koji prikazuje promjene vjerojatnosti za različite bolesti. Svaka bolest ima svoju liniju na grafu u odgovarajućoj boji. Različiti datumi predstavljeni su na x-osi, dok su vjerojatnosti prikazane na y-osi. Graf se sprema u memoriju kao slika u PNG formatu koristeći *BytesIO*. Slika se kodira u Base64 format kako bi se mogla direktno prikazati na web stranici bez potrebe za spremanjem na poslužitelj.

```
133 def PatientDetail(request, patient_id):
155
156     # Generiranje grafa
157     if survey_responses.exists():
158         dates = [response.date for response in survey_responses]
159         probabilities = {disease: [] for disease in DISEASE_SYMPTOM_WEIGHTS.keys()}
160
161         for response in survey_responses:
162             values = map_answers_to_values(response)
163             probs = generate_probabilities(values)
164             for disease in probabilities:
165                 probabilities[disease].append(probs[disease])
166
167         plt.figure(figsize=(10, 6))
168         for disease, probs in probabilities.items():
169             plt.plot(dates, probs, label=disease)
170
171         plt.xlabel('Datum')
172         plt.ylabel('Vjerojatnost')
173         plt.title('Vjerojatnosti bolesti kroz vrijeme')
174         plt.legend()
175         plt.grid(True)
176
177         buf = BytesIO()
178         plt.savefig(buf, format='png')
179         buf.seek(0)
180         img_str = base64.b64encode(buf.getvalue()).decode('utf-8')
181         buf.close()
182     else:
183         img_str = None
184
```

Slika 4.15 Programski kod za generiranje grafa

## 4.3. Programsko rješenje na strani poslužitelja

### 4.3.1. Prikaz preporuka

Kada korisnik ispuni upitnik i izračuna vjerojatnosti od obolijevanja od bolesti probavnog sustava, sustav izbacuje preporuke na temelju postotka. Na slici 4.16, prikazane su preporuke koje sustav

izbacuje. Ako je postotak vjerojatnosti 80% ili veći, sustav će preporučiti hitan posjet liječniku. Ako je postotak između 40% i 80%, sustav će sugerirati da bi bilo poželjno posjetiti liječnika, a ako je manji od 40%, sustav će preporučiti da za sada nije potrebno posjetiti liječnika.

```
def generate_disease_recommendation(probability):  
    if probability >= 0.8:  
        return "Preporučujemo da hitno posjetite liječnika."  
    elif probability >= 0.4:  
        return "Bilo bi dobro da svoje stanje provjerite s liječnikom."  
    else:  
        return "Za sada ne morate posjetiti liječnika."
```

Slika 4.16 Programski kod za stvaranja preporuka na temelju vjerojatnosti

Nakon što sustav prikaže preporuke temeljem izračunate vjerojatnosti obolijevanja, također prikazuje tri bolesti koje imaju najveću vjerojatnost da pacijent ima prema odgovorima unesenim u upitniku. Uz to, sustav daje specifične preporuke za bolest koja ima najveću vjerojatnost da pacijent ima. Na slici 4.17, prikazana je funkcija `get_recommendations_for_condition(condition)` koja koristi liječnik `recommendations` za pohranu specifičnih preporuka za svaku bolest. Funkcija vraća preporuku za bolest koja je prosljeđena kao argument.

```
def get_recommendations_for_condition(condition):  
    recommendations = {  
        'GERB': [  
            "Izbjegavajte masnu i začinjenu hranu.",  
            "Jedite manje obroke češće umjesto velikih obroka.",  
            "Izbjegavajte ležanje odmah nakon jela.",  
        ],  
        'Gastritis': [  
            "Izbjegavajte začinjenu hranu.",  
            "Jedite manje obroke tijekom dana.",  
            "Uzmite lijekove koje vam liječnik preporuči, uključujući antibiotike ako je gastritis uzrokovan bakterijom Helicobacter pylori.",  
        ],  
        'Dispepsija': [  
            "Izbjegavajte masnu i tešku hranu, ograničite unos alkohola, kave i gaziranih pića.",  
            "Jedite redovito i izbjegavajte prejedanje.",  
            "Povećajte unos vlakana u prehrani."  
        ],  
        'Crohnova bolest': [  
            "Jedite pet manjih obroka tijekom dana, smanjite unos masne i pržene hrane.",  
            "Izbjegavajte mlijeko i mliječne preradevine, kao i vlaknaste namirnice.",  
            "Pijte puno tekućine i izbjegavajte alkohol i gazirana pića."  
        ],  
        'IBS': [  
            "Izbjegavajte hranu koja uzrokuje nadutost, kao što su grah i kupus.",  
            "Smanjite unos kave i alkohola.",  
            "Uvedite redovite obroke i jedite polako.",  
            "Razmislite o promjeni prehrane prema FODMAP prehrani."  
        ],  
        'Celijakija': [  
            "Potpuno izbjegavajte gluten u prehrani.",  
            "Pažljivo pročitajte etikete na prehrambenim proizvodima.",  
            "Koristite alternativne žitarice poput riže, kukuruza i kvinoje.",  
            "Pratite svoj nutritivni status i konzultirajte se s dijetetičarom."  
        ],  
    }  
    return recommendations.get(condition, [])
```

Slika 4.17 Programski kod stvaranja preporuka za najvjerojatniju bolest

### 4.3.2. Baza podataka

Registracijom korisnika, njegovi se podaci spremaju u bazu podataka. Slika 4.18 prikazuje popis pacijenata u administrativnom sučelju Django aplikacije.

Select patient to change

Action: ----- Go 0 of 4 selected

- PATIENT
- 4 - Ivan Ivić
- 3 - Mia Mijić
- 2 - Ante Anić
- 1 - Petra Vuica

4 patients

Slika 4.18 Prikaz pacijenata u bazi podataka

Odabirom pacijenta, prikazuju se podaci uneseni tijekom registracije kao što su spol, korisničko ime i datum rođenja. Na slici 4.19 prikazani su podaci jednog korisnika. Za spremanje imena i prezimena korisnika koristimo Django-ova polja *first\_name* i *last\_name* unutar *User* modela za pohranu imena i prezimena.

Change patient

1 - Petra Vuica

User: pero [edit] [add] [view]

Username: pero

Gender: Ženski

DateOfBirth: 2002-07-11 Today [calendar]

SAVE Save and add another Save and continue editing

Slika 4.19 Prikaz detalja o korisniku u bazi podataka

Pored podataka o korisnicima, baza podataka čuva i odgovore na upitnike koje pacijenti popunjavaju. Svaki odgovor na upitnik čuva se zajedno s datumom kada je upitnik ispunjen, što omogućava praćenje promjena u zdravstvenom stanju. Također, poruke koje liječnik može poslati pacijentu pohranjuju se u bazu podataka. Svaka poruka sadrži informaciju o pošiljatelju i primatelju poruke, kao i vremensku oznaku kada je poruka poslana.



## **5. PRIKAZ RADA WEB APLIKACIJE; ANALIZA REZULTATA**

### **5.1. Prikaz rada web aplikacije**

Pristupanjem aplikaciji, otvara se početna stranica koja prikazuje osnovne informacije o aplikaciji i njenim funkcionalnostima. Da bi mogao koristiti aplikaciju, svaki mora biti prijavljen. Na početnoj stranici aplikacije nalaze se gumbi za registraciju ili prijavu korisnika. Ako korisnik nema još svoj korisnički račun, treba odabrati gumb *Registracija* i bit će preusmjeren na stranicu s formom za registraciju. Ukoliko korisnik već posjeduje korisnički račun treba kliknuti na gumb *Prijava* te će mu se otvoriti forma za prijavu u aplikaciju. Zatim se otvara početna stranica, ovisna o ulozi korisnika (liječnik ili pacijent). Ako je korisnik liječnik, početna stranica će sadržavati gumb *Zdravstveni kartoni*, dok će pacijentima biti prikazan gumb *Ispunite upitnik*. Klikom na gumb *Zdravstveni kartoni*, liječniku se otvara popis svih pacijenta, sortiranih abecedno po prezimenu. Odabirom određenog pacijenta otvara se zdravstveni karton tog pacijenta. U zdravstvenom kartonu liječnik može pregledati osobne podatke pacijenta, prethodno ispunjene upitnike, preporuke sustava, graf koji prikazuje vjerojatnost svake bolesti kroz vrijeme te može poslati poruku pacijentu. A ukoliko je korisnik pacijent, početna stranica će sadržavati gumb *Ispunite upitnik*. Pritiskom tog gumba otvara se upitnik koji se sastoji od pitanja o životnim navikama i simptomima. Pacijent mora ispuniti sva pitanja u upitniku kako bi upitnik bio uspješno ispunjen. Nakon uspješnog popunjavanja upitnika, prikazuju se preporuke na temelju odgovora pacijenta. Pacijent također može vidjeti svoj profil, odnosno svoj zdravstvenom kartonu. Pacijent nema mogućnost slanja poruka liječniku. Svaki korisnik se može odjaviti iz aplikacije klikom na tipku *Odjava* na navigacijskoj traci.

#### **5.1.1. Prikaz postupka za registriranje korisnika**

Korisnik započinje postupak registracije kada klikne dugme *Registracija* na početnoj stranici. Nakon toga se otvara stranica koja sadrži formu za registraciju, prikazana na slici 5.1. Korisnik ispunjava osobne podatke te odabire svoju ulogu. Sva polja moraju biti ispunjena kako bi registracija bila uspješna, u suprotnom, ispisuje se odgovarajuća poruka.

The registration form is titled "Registracija". It contains the following fields and options:

- Ime (Name)
- Prezime (Surname)
- Korisničko ime (Username)
- Email
- Lozinka (Password)
- Ponovi lozinku (Repeat password)
- dd.mm.gggg. 📅 (Date of birth)
- Spol:  Muški  Ženski (Gender)
- Jeste li liječnik? (Are you a doctor?)

A blue button labeled "REGISTRIRAJ SE" is located at the bottom of the form.

Slika 5.1 Forma za registraciju korisnika

### 5.1.2. Prikaz postupka za prijavu korisnika

Ako korisnik ima korisnički račun, odabire tipku *Prijava* na početnoj stranici. Pritiskom te tipke prikazuje se stranica koja sadrži formu za prijavu korisnika, prikazanu na slici 5.2. U okviru ove forme nalazi se i tipku *Ne posjedujete korisnički račun?*, koja omogućava korisniku da započne proces registracije ako ne posjeduje korisnički račun.

The login form is titled "Prijavi se". It contains the following fields and options:

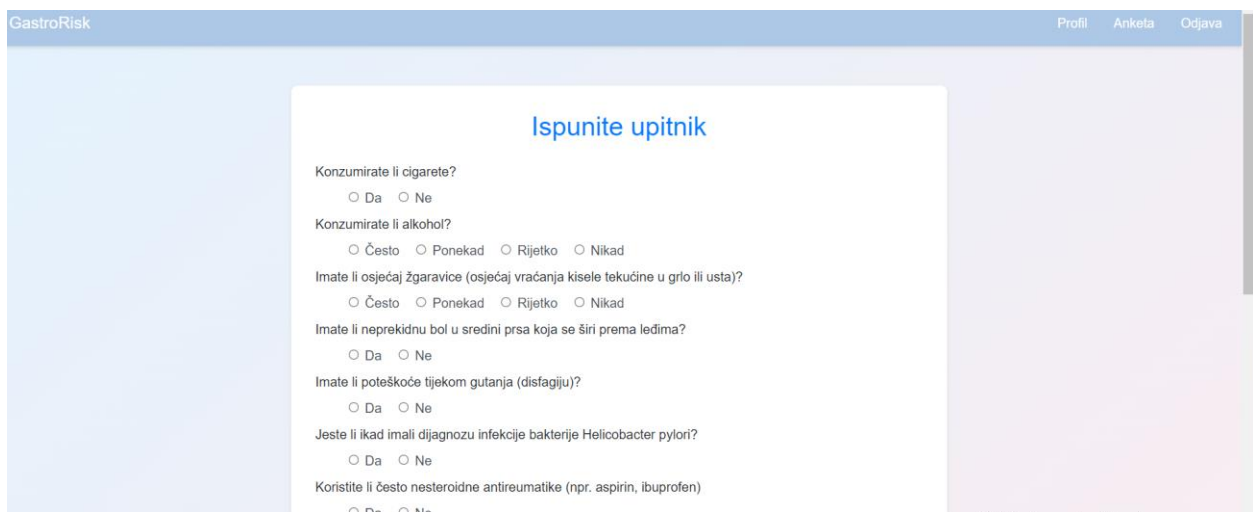
- Korisničko ime (Username)
- Lozinka (Password)

A blue button labeled "PRIJAVA" is located below the fields. Below the button is a link: "Ne posjedujete korisnički račun?"

Slika 5.2 Forma za prijavu korisnika

### 5.1.3. Prikaz postupka za ispunjavanje upitnika

Nakon prijave u aplikaciju, pacijent ima mogućnost popunjavanja upitnika. Nakon otvaranja stranica, prikazuje se forma za upitnik. Dio upitnika prikazan je na slici 5.3. Za svako pitanje, pacijent ima mogućnost odabiranja jedan od ponuđenih odgovora koristeći radio gumbe. Gumbi omogućavaju jednostavan izbor jedne od opcija, pri čemu je svako pitanje obavezno. Ako korisnik pokuša da podnese upitnik bez popunjavanja svi polja, forma će prikazati odgovarajuću poruku koja upozorava korisnika da su neka polja ostala ne popunjena. Nakon popunjavanja svih polja, pritiskom na tipku *Pošalji* koja se nalazi na dnu forme, svi se uneseni podaci šalju i čuvaju u bazi podataka te se otvara stranica koja sadrži preporuke za pacijenta.



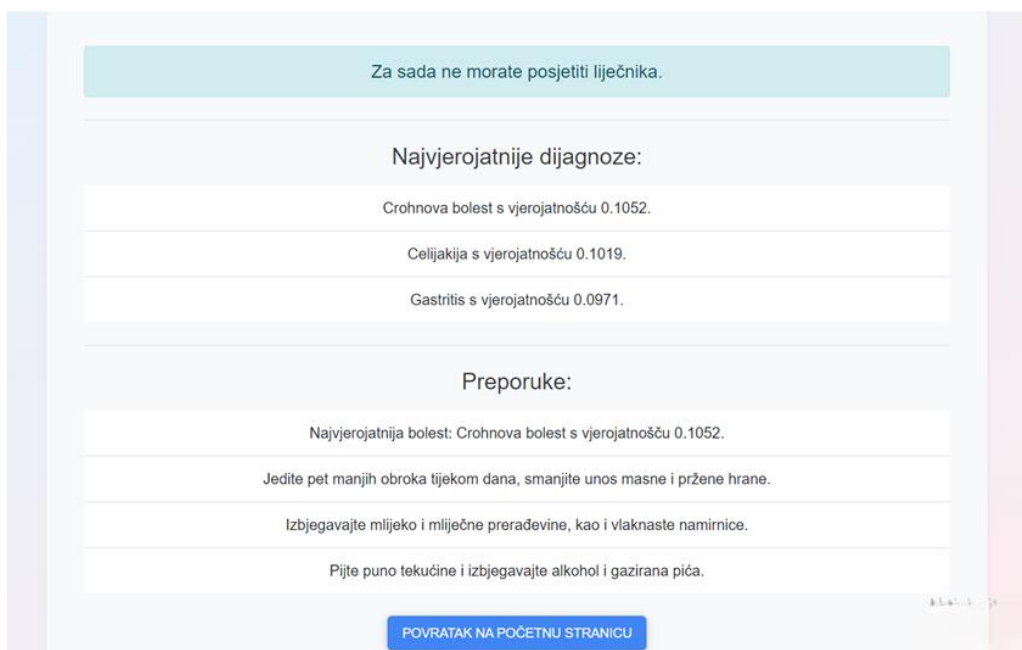
The screenshot shows a web interface for the 'GastroRisk' application. At the top, there is a navigation bar with the application name 'GastroRisk' on the left and links for 'Profil', 'Anketa', and 'Odjava' on the right. The main content area is a white box titled 'Ispunite upitnik'. It contains the following questions and options:

- Konзумirate li cigarete?  
 Da  Ne
- Konзумirate li alkohol?  
 Često  Ponekad  Rijetko  Nikad
- Imate li osjećaj žgaravice (osjećaj vraćanja kiseline tekućine u grlo ili usta)?  
 Često  Ponekad  Rijetko  Nikad
- Imate li neprekidnu bol u sredini prsa koja se širi prema leđima?  
 Da  Ne
- Imate li poteškoće tijekom gutanja (disfagiju)?  
 Da  Ne
- Jeste li ikad imali dijagnozu infekcije bakterije *Helicobacter pylori*?  
 Da  Ne
- Koristite li često nesteroidne antireumatike (npr. aspirin, ibuprofen)?  
 Da  Ne

*Slika 5.3 Prikaz dijela upitnika aplikacije GastroRisk*

### 5.1.4. Prikaz preporuka

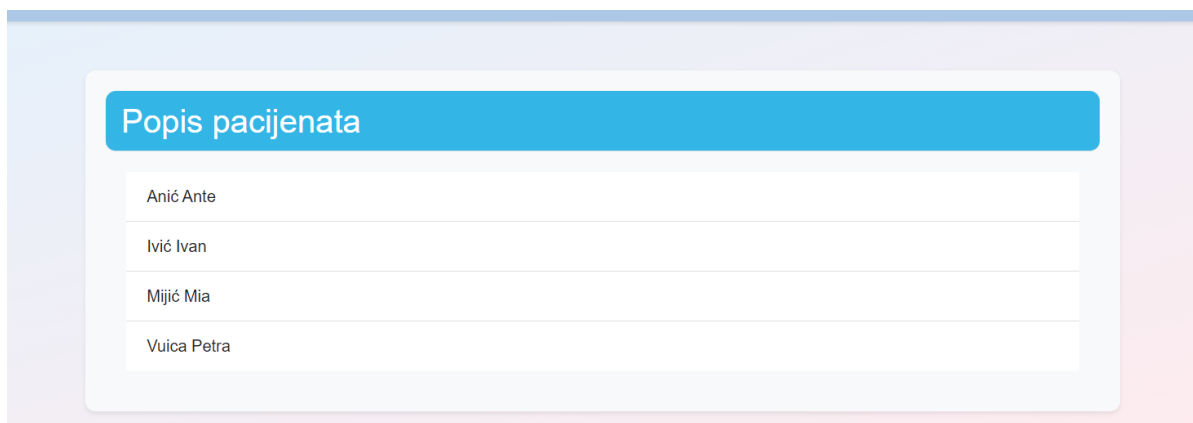
Na temelju unesenih odgovora o simptomima, sustav generira preporuke na osnovu izračunatih vjerojatnosti. Prvo se prikazuje preporuka o važnosti posjete liječniku. Zatim se prikazuju tri najvjerojatnije bolesti zajedno s njihovim vjerojatnostima. Uz to, sustav pruža i preporuke vezane uz najvjerojatniju bolest koju je izračunao. Prikaz dobivenih preporuka prikazano je na slici 5.4.



*Slika 5.4 Prikaz preporuka koje je generirala aplikacija GastoRisk*

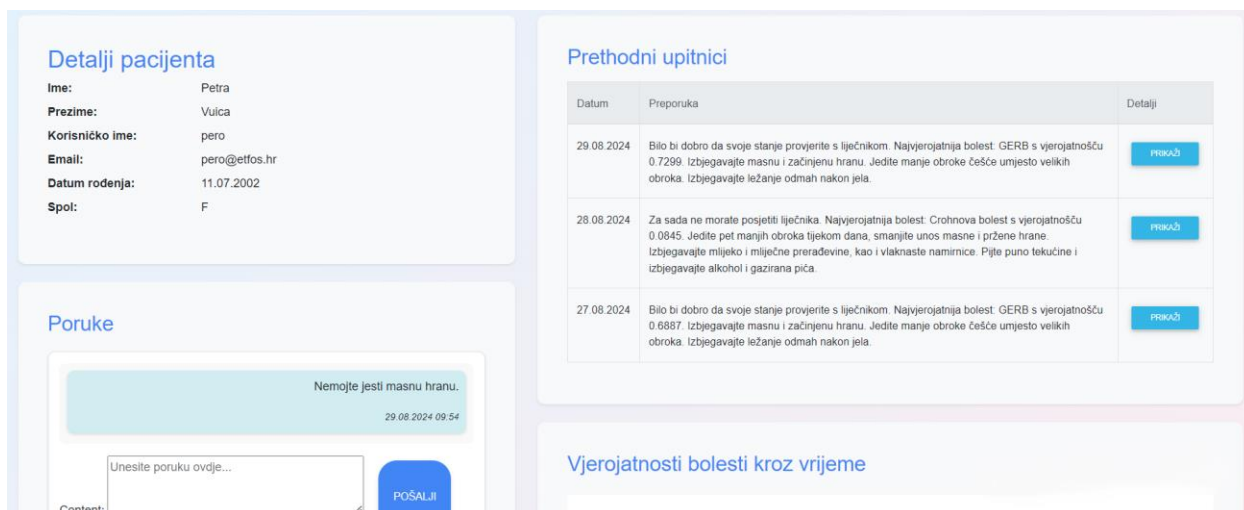
### **5.1.5. Prikaz postupka za zdravstveni karton pacijenta**

Liječnik ima mogućnost vidjeti zdravstvene kartone svih pacijenata. Pritiskom na tipku *Zdravstveni kartoni*, otvara se stranica s popisom svih pacijenata, prikazana na slici 5.5.



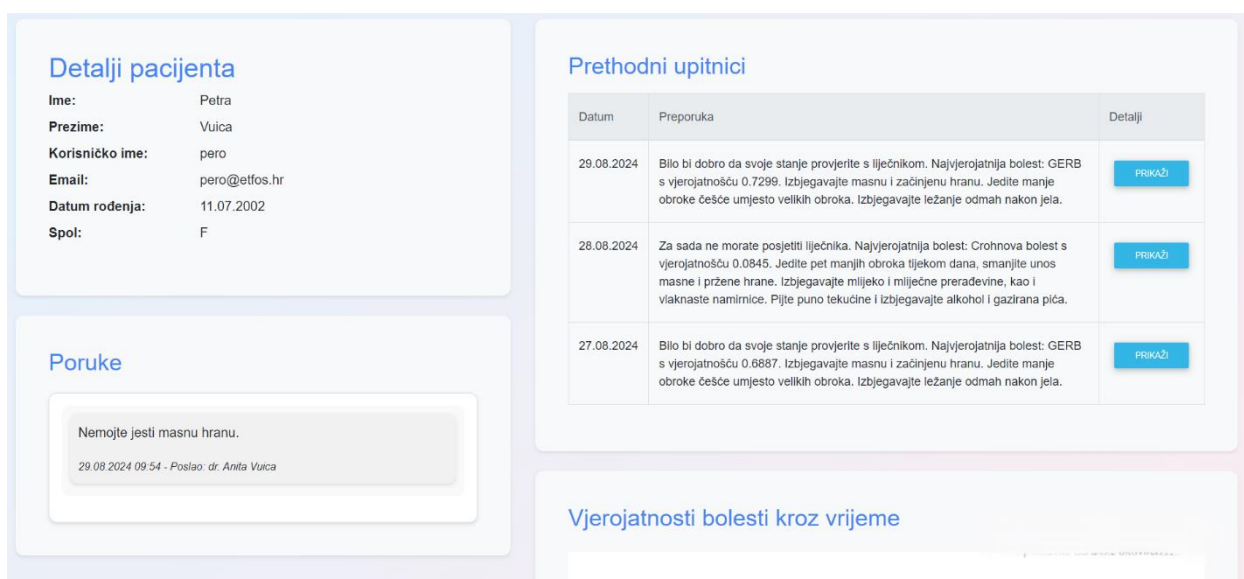
*Slika 5.5. Prikaz popisa pacijenata*

Odabiranjem određenog pacijenta otvara se stranica prikazana na slici 5.6. Liječnik ima uvid u osobne podatke i sve prethodne upitnike, također i mogućnosti slanja poruka pacijentima.



*Slika 5.6 Prikaz zdravstvenog kartona pacijenta – prijavljen liječnik*

Isto kao i zdravstveni karton, izgleda profil pacijenta. Profil pacijenta prikazan je na slici 5.7.



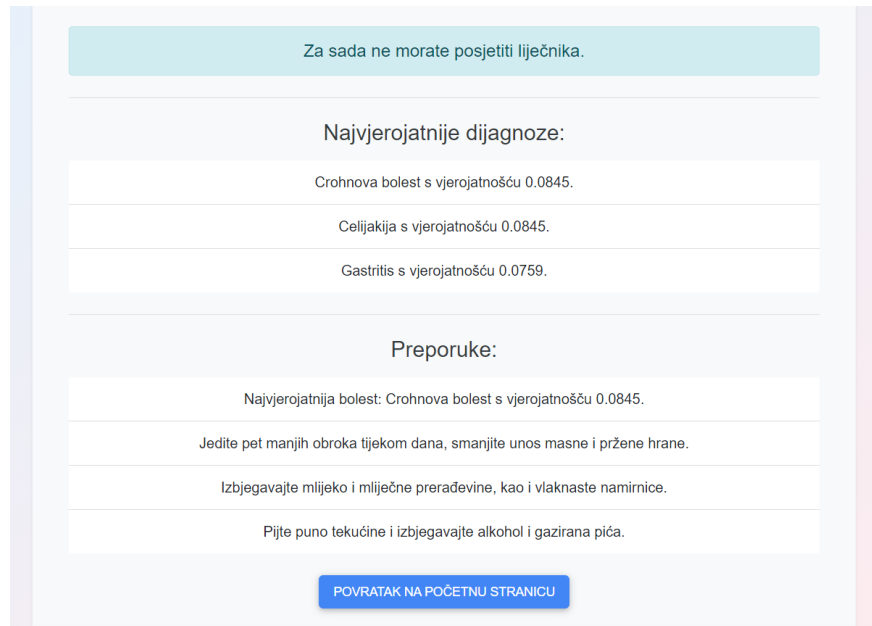
*Prikaz 5.7 Prikaz zdravstvenog kartona pacijenta – prijavljen pacijent*

## 5.2. Primjeri korištenja s analizom rezultata

U ovom će poglavlju biti opisano kako se aplikacija koristi u različitim scenarijima, ovisno o simptomima koje pacijent unosi. U svakom sljedećem primjeru bit će opisana tri različita scenarija: kada je pacijent zdrav, kada pokazuje umjerene simptome i kada ima simptome koji zahtijevaju hitan posjet liječnika.

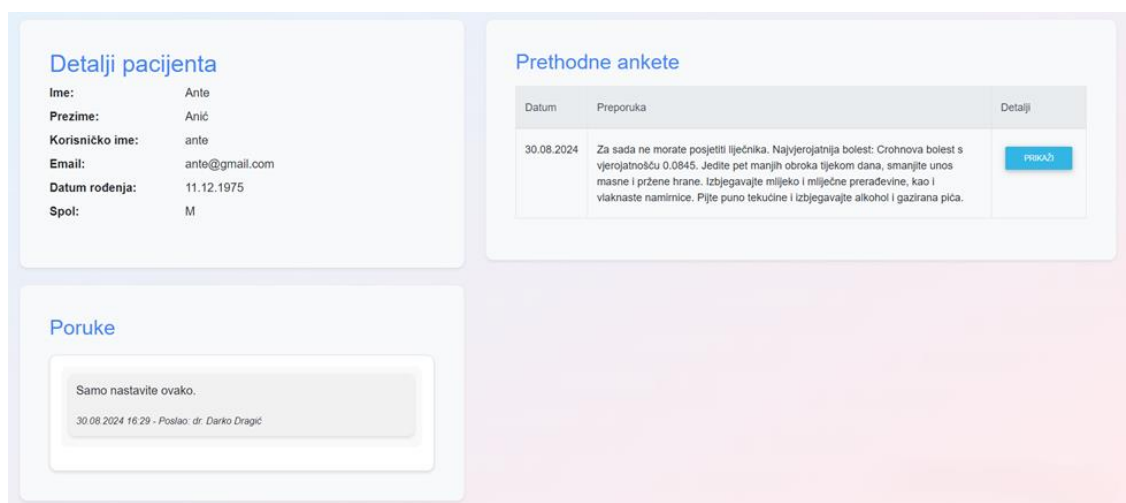
### 5.2.1. Prvi primjer korištenja aplikacije

U ovom primjeru, pacijent unosi odgovore koji ukazuju na minimalne ili nikakve simptome bolesti. Pacijent odgovara na sva pitanja u negativnom smislu odnosno sad *Ne* ili *Nikad*. Na temelju uneseni podataka, aplikacija izračunava vjerojatnosti za različite bolesti, ali su vjerojatnosti vrlo niske. Kao rezultat, aplikacija preporučuje da pacijent ne mora posjetiti liječnika jer nema značajnih zdravstvenih problema. Ovaj scenarij prikazuje kako aplikacija može potvrditi dobro zdravstveno stanje pacijenta. Prikaz preporuka koja je sustav izbacio prikazano je na slici 5.8.



Slika 5.8 Prikaz preporuka koje je generirala aplikacija GastroRisk

Na slici 5.9 prikazan korisnički profil, odnosno njegov zdravstveni karton, koji korisnik vidi nakon ispunjenog upitnika. Klikom na tipku *Prikaži*, otvara se prozor, prikazan na slici 5.10.



Slika 5.9 Prikaz profila pacijenta nakon ispunjene jednog upitnika

Detalji upitnika - 30.08.2024

Pušenje:	Ne
Alkohol:	Nikad
Žgaravica:	Nikad
Bol u prsima:	Ne
Disfagija:	Ne
Helicobacter pylori:	Ne
NSAIDs:	Ne
Bolovi u trbuhu:	Nikad
Mučnina:	Nikad
Bol nakon jela:	Ne
Proljevi:	Nikad
Grčevi:	Nikad
Umor:	Ne
Hitnost odlaska na WC:	Ne
Gubitak težine:	Ne
Stres:	Nikad
Gubitak apetita:	Nikad

ZATVORI

Slika 5.10 Prikaz detalja upitnika

### 5.2.2. Drugi primjer korištenja aplikacije

U ovom primjeru, pacijent unosi odgovore koji ukazuju na blage do umjerene simptome. Na primjer, pacijent povremeno osjeća žgaravicu i bolove u trbuhu, ali ne pokazuje ozbiljne simptome. Preporuke sustava prikazane su na slici 5.11.

Bilo bi dobro da svoje stanje provjerite s liječnikom.

Najvjerojatnije dijagnoze:

GERB s vjerojatnošću 0.7097.
Gastritis s vjerojatnošću 0.1639.
Celijakija s vjerojatnošću 0.1615.

Preporuke:

Najvjerojatnija bolest: GERB s vjerojatnošću 0.7097.
Izbjegavajte masnu i začinjenu hranu.
Jedite manje obroke češće umjesto velikih obroka.
Izbjegavajte ležanje odmah nakon jela.

POVRATAK NA POČETNU STRANICU

Slika 5.11 Prikaz preporuka koje je generirala aplikacija GastroRisk

Kada pacijent ima više od dva ispunjena upitnika, na svom profilu može vidjeti graf vjerojatnosti bolesti kroz vrijeme. Na slici 5.12 prikazan je taj graf, koji se pojavljuje nakon prvog i drugog primjera korištenja aplikacije. Graf omogućuje pacijentu praćenje promjena u vjerojatnosti pojave određene bolesti.



*Slika 5.12 Prikaz grafa nakon ispunjena dva upitnika*

### 5.2.3. Treći primjer korištenja aplikacije

U ovom primjeru, pacijent unosi odgovore koji ukazuju na ozbiljne simptome. Pacijent osjeća bol u trbuhu, mučninu, gubitak težine, gubitak apetita i anemiju. Na temelju ovih odgovora, sustav izračunava visoke vjerojatnosti, pri čemu je najvjerojatnije da pacijent boluje od Crohnove bolesti. Preporučuje se da pacijent hitno posjeti liječnika jer simptomi ukazuju na ozbiljno zdravstveno stanje koje zahtjeva medicinsku intervenciju. Ovaj primjer prikazuje kako sustav može prepoznati situacije koje zahtijevaju hitnu medicinsku pažnju. Prikaz preporuka koje ja sustav generirao prikazan je na slici 5.13.



Preporučujemo da hitno posjetite liječnika.

**Najvjerojatnije dijagnoze:**

Crohnova bolest s vjerojatnošću 0.8086.
Celijakija s vjerojatnošću 0.7946.
IBS s vjerojatnošću 0.4721.

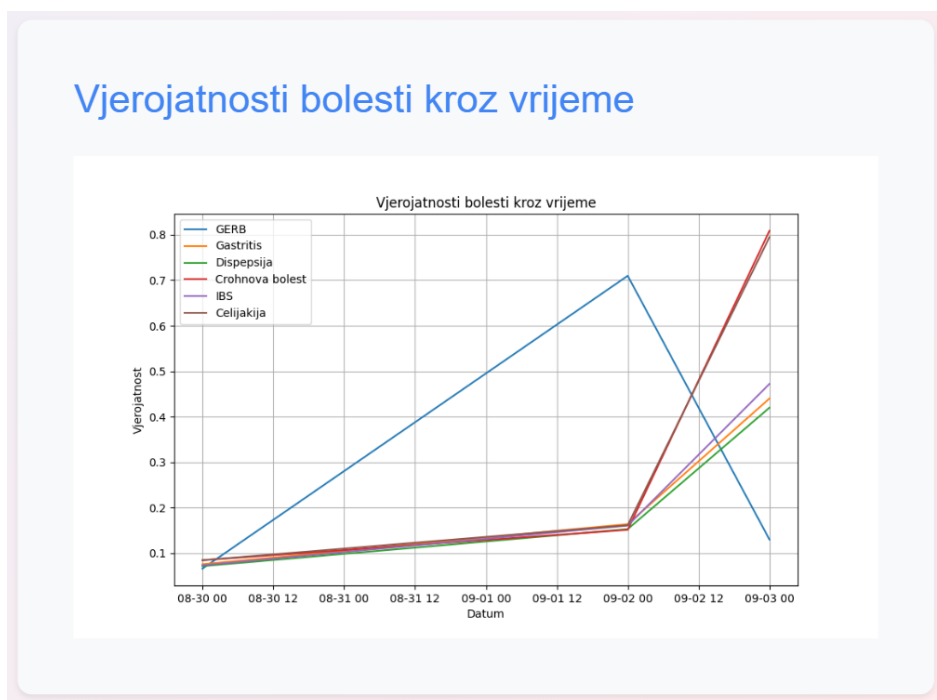
**Preporuke:**

Najvjerojatnija bolest: Crohnova bolest s vjerojatnošću 0.8086.
Jedite pet manjih obroka tijekom dana, smanjite unos masne i pržene hrane.
Izbjegavajte mlijeko i mliječne prerađevine, kao i vlaknaste namirnice.
Pijte puno tekućine i izbjegavajte alkohol i gazirana pića.

[POVRATAK NA POČETNU STRANICU](#)

*Slika 5.13 Prikaz preporuka koje je generirala aplikacija GastroRisk*

Slika 5.14 prikazuje kako se mijenja vjerojatnost za bolesti kroz vrijeme za korisnika na kojem su ispitani primjeri.



*Slika 5.14 Prikaz grafa nakon tri primjera korištenja aplikacije*

## 5. ZAKLJUČAK

Bolesti probavnog sustava su sve češće, a mnogi ljudi nisu svjesni ozbiljnosti simptoma dok bolest ne uznapreduje. Kako bi se omogućilo pravovremeno prepoznavanje i praćenje tih bolesti, važno je razvijati aplikacije i sustave koji mogu pouzdano procijeniti rizike i dati preporuke za daljnje postupanje. Cilj aplikacije *GastroRisk* je korisnicima pružiti brz i jednostavan način za praćenje simptoma i dobivanje preporuka na temelju analize podataka. Aplikacija koristi metodu logističke regresije kako bi precizno izračunala rizik razvoja određenih bolesti probavnog sustava i generirala preporuke koje pacijentima mogu pomoći u upravljanju njihovim zdravstvenim stanjem. Korisnicima je omogućeno registriranje, prijava, ispunjavanje upitnika o simptomima, pregled vlastitih podataka i praćenje preporuka, dok je liječnicima omogućen pregled i slanje preporuka pacijentima.

Ispitivanjem aplikacije provedeno je kroz tri različita slučaja korištenja s unosima koji rezultiraju različitim razinama rizika, gdje su za svaki slučaj prikazane odgovarajuće preporuke i pohranjeni podaci za buduće praćenje. Aplikacije se može dodatno unaprijediti algoritmima analize podataka o simptomima i uvođenjem naprednih algoritama analize podataka, kako bi se poboljšala preciznost i proširila mogućnost primjene u različitim medicinskim scenarijima. Namjera ove aplikacije je pomoći liječniku i pacijentu, ali ne može zamijeniti liječnika.

## LITERATURA

- [1] M. Radić, Probavni sustav [online], mandisPHARM, 2022, dostupno na: <https://mandispharmljekarne.hr/blog/probavni-sustav-50/> [20.6.2024]
- [2] A. J. Tresca, Digestive Health: Your Digestive System and How It Works [online], 2023, dostupno na: <https://www.verywellhealth.com/your-digestive-system-and-how-it-works-1941716> [20.6.2028]
- [3] Health Information: Digestive Diseases [online], National Institute of Diabetes and Digestive and Kidney Diseases, 2024, dostupno na: <https://www.niddk.nih.gov/health-information/digestive-diseases> [20.6.2024.]
- [4] Digestive system [online], Healthdirect, 2024, dostupno na: <https://www.healthdirect.gov.au/digestive-system#:~:text=The%20digestive%20system%20is%20a,as%20carbohydrates%2C%20fats%20and%20proteins> [20.6.2024]
- [5] M. Ražov Radas, Mapa tijela: GERB [online], Pliva zdravlje, 2019, dostupno na: <https://www.plivazdravlje.hr/aktualno/clanak/32156/Simptomi-i-dijagnoza-GERB-a.html> [20.6.2024.]
- [6] Gastroesophageal Reflux Disease (GERD) [online], The Johns Hopkins Medicine, 2024, dostupno na: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/gastroesophageal-reflux-disease-gerd> [21.6.2024.]
- [7] M. M. Phillips, Gastritis [online], National Library of Medicine, 2023, dostupno na: <https://medlineplus.gov/ency/article/001150.htm> [15.6.2024.]
- [8] D. Ljubas i Ž. Cvitkušić, Gastritis – simptomi, uzroci i prehrana [online], Mazalin poliklinika, 2023, dostupno na: <https://poliklinika-mazalin.hr/blog/gastritis-simptomi-uzroci-i-prehrana/> [21.6.2024.]
- [9] D. Vranešić Bender, I. Alebić, Ž. Krznarić, Dispepsija [online], 2007, dostupno na: <https://www.plivazdravlje.hr/aktualno/clanak/13090/Dispepsija.html> [21.6.2024]
- [10] Crohn's Disease [online], The Johns Hopkins Medicine, 2024, dostupno na: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/crohns-disease> [22.6.2024.]
- [11] Chronova bolest [online], Poliklinika ANALIZA, 2024, dostupno na: <https://poliklinika-analiza.hr/chronova-bolest/> [21.6.2024]
- [12] M. M. Phillips, D. C. Dugdale, Irritable bowel syndrome [online], National Library of Medicine, dostupno na: <https://medlineplus.gov/ency/article/000246.htm> [21.6.2024.]

- [13] Irritable Bowel Syndrome (IBS) [online], The Johns Hopkins Medicine, 2024, dostupno na: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/irritable-bowel-syndrome-ibs> [21.6.2024.]
- [14] S. G. Anzlovar, How to Identify and Treat Celiac Disease in Infants [online], 2022, dostupno na: <https://www.healthline.com/nutrition/celiac-disease-in-infants> [22.6.2024.]
- [15] Y. E. Rustam, M. B. Bahodir, E. E. Dilshod, K. B. Meyerbek, Evaluation of Machine Learning Algorithms for Gastroenterological Diseases Prediction, Science and Innovation International Scientific Journal, vol. 2, No. 7, str. 83-94, srpanj 2023.
- [16] K. A. Manhar, V. Roy, P. D. Patil, L. Roy, Deep Learning's Role in Advancing Gastroenterology and Digestive Health, 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG), IEEE ICTBIG-2023, str. 1- 4
- [17] New Patient Handbook [online], Hoag Medical Group, 2023. dostupno na: [12204\\_HMG\\_PatientHandbookRevise1222\\_5.pdf](https://www.hoagmedicalgroup.com/patient-handbook-revised-2022-5.pdf) (hoagmedicalgroup.com) [27.8.2024.]
- [18] Symptom Checker and Health Assistant [online], Infermedica, dostupno na: <https://symptomate.com/> [27.8.2024.]
- [19] S. Peters, Fix The Miscommunication Between Your Gut And Brain [online], Nerva, 2024., dostupno na: <https://try.nervaibs.com/hypnotherapy-nerva/> [10.9.2024]
- [20] My IBD Care [online], Ampersand, 2024., dostupno na: <https://ampersandhealth.co.uk/myibdcare/> [10.9.2024.]
- [21] Bowelle – The IBS Tracker, Bowelle [online], 2024., dostupno na: <https://bowelle.com/> [10.9.2024.]
- [22] S. Gasmi, T. Bouhadada, Survey on Recommendation Systems, Proceedings of the 10th Int. Conference on Information Systems and Technologies (ICIST '20), Kaunas, Lithuania, 2020, str. 1-7.
- [23] S. Saleem, M. N. Asim, L. Van Elst, A. Dengel, FNReq-Net: A Hybrid Computational Framework for Functional and Non-functional Requirements Classification, Journal of King Saud University – Computer and Information Sciences, vol. 35, No. 101665
- [24] D. H. Maulud, A. M. Abdulazeez, A Review on Linear Regression Comprehensive in Machine Learning, Journal of Applied Science and Technology Trends, vol. 01, No. 02, str. 140-147, prosinac 2020.
- [25] S. Mondal, Linear Regression vs Logistic Regression: Difference [online], Analytics Vidhya, 2020., dostupno na: <https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/> [24.8.2024.]

- [26] What is Logistic Regression? [online], IBM, dostupno na: <https://www.ibm.com/topics/logistic-regression> [6.8.2023.]
- [27] Meet Django [online], Django, dostupno na: <https://www.djangoproject.com/> [8.7.2024.]
- [28] Django Project MVT Structure [online], GeeksforGeeks, dostupno na: <https://www.geeksforgeeks.org/django-project-mvt-structure/> [8.7.2024.]
- [29] What is Python? Executive Summary [online], Python, dostupno na: [https://www.python.org/doc/essays/blurb/?external\\_link=true](https://www.python.org/doc/essays/blurb/?external_link=true) [8.7.2024.]
- [30] HTML: HyperText Markup Language [online], Mozilla Developer Network, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTML> [14.8.2024]
- [31] CSS: Cascading Style Sheets [online], Mozilla Developer Network, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/CSS> [14.8.2024.]
- [32] Why did we build Visual Studio Code [online], Visual Studio Code, dostupno na: <https://code.visualstudio.com/docs/editor/whyvscode> [14.6.2023.]

## SAŽETAK

U ovom završnom radu osmišljena je i razvijena web aplikacija za procjenu obolijevanja i praćenja tijeka bolesti probavnog sustava. Aplikacija je razvijena korištenjem programskog okvira Django, programskog jezika Pythona, HTML-a i CSS-a. Ova aplikacija omogućava korisnicima registriranje i prijavu s različitim razinama ovlasti te pristup funkcionalnostima prilagođenim njihovoj ulozi. Pohrana osobnih podataka i praćenje simptoma oboljenja omogućeni su kroz sigurnu bazu podataka. Analiza podatka provodi se logističkom regresijom, dok se preporuke generiraju na temelju simptoma. Testiranjem aplikacije za različite slučajeve korištenja utvrđena je njena točnost i učinkovitost u praćenju zdravstvenog stanja i generiranju relevantnih preporuka.

**Ključne riječi:** bolesti probavnog sustava, generiranje preporuka, -logistička regresija, praćenje simptoma, web aplikacija.

## **ABSTRACT**

**Title: Web application for risk assessment of gastrointestinal disease and monitoring of disease progression using data analysis**

In this paper, a web application was designed and developed for assessing the risk of digestive system diseases and monitoring their progression. The application was built using the Django framework, the Python programming language, HTML, and CSS. It allows users to register and log in with different levels of access, providing functionalities tailored to their roles. Personal data storage and symptom tracking are facilitated through a secure database. Data analysis is performed using logistic regression, while recommendations are generated based on the symptoms. Testing the application across various use cases demonstrated its accuracy and effectiveness in monitoring health conditions and generating relevant recommendations.

**Keywords:** digestive system diseases, recommendation generation, logistic regression, symptom tracking, web application.

## **ŽIVOTOPIS**

Petra Vuica rođena je 11. srpnja 2002. godine u Osijeku. Nakon završetka osnovne škole Vladimira Becića u Osijeku, upisuje I. gimnaziju u Osijeku. Nakon završetka srednje škole, upisuje preddiplomski studija računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer programsko inženjerstvo.



## **PRILOZI**

Prilog 1. Završni rad „Web aplikacija za procjenu rizika obolijevanja i praćenje tijeka bolesti probavnog sustava primjenom analize podataka“ u *.docx* formatu

Prilog 2. Završni rad „Web aplikacija za procjenu rizika obolijevanja i praćenje tijeka bolesti probavnog sustava primjenom analize podataka“ u *.pdf* formatu

Prilog 3. Git repozitorij s programskim kodom aplikacije