

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Žan Fras

Razvoj spletne aplikacije za vpisovanje izkušenj z boleznijo ALS

Diplomsko delo

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Žan Fras

Mentor: izr. prof. dr. Aleš Žiberna

Razvoj spletne aplikacije za vpisovanje izkušenj z boleznijo ALS

Diplomsko delo

Ljubljana, 2017

Zahvaljujem se družini, ki me je spodbujala in podpirala tekom študija. Za pomoč pri izdelavi diplomskega dela se zahvaljujem vsem, ki so mi pomagali, posebna zahvala pa gre mentorju izr. prof. dr. Alešu Žiberni za vse nasvete ter strokovno pomoč pri pisanju.

Razvoj spletne aplikacije za vpisovanje izkušenj z boleznijo ALS

Cilj diplomske naloge je razviti spletno aplikacijo za vpisovanje izkušenj z boleznijo imenovano amiotrofična lateralna skleroza ali krajše ALS. V nalogi so opisane vse tehnologije in orodja, ki so bila skozi proces razvoja uporabljena, kakor tudi načrtovanje in funkcionalnosti aplikacije. Spletna aplikacija je rezultat diplomske naloge, ki je bil dosežen z uporabo programskega jezika Python s pomočjo tehnologij HTML, CSS, JavaScript, Bootstrap in SQLite podatkovne baze. Aplikacija registriranim uporabnikom omogoča vpisovanje novih izkušenj z boleznijo ALS preko strukturiranega spletnega obrazca, poleg tega pa omogoča vsem uporabnikom tudi pregled vnosov izkušenj in uspešnosti kategorij vnosov skozi vizualizacijo podatkov. Vsi uporabniki imajo tudi dostop do opisa same bolezni, kakor tudi do seznama uporabnih telefonskih števil. Ugotovljeno je bilo tudi, kako oblikovati spletni obrazec uporabniško prijazen in intuitiven, da bi uporabniki znali vnesti svoje izkušnje, medtem ko je bilo za prikaz podatkov ugotovljeno, kateri so tisti primerni podatki za prikaz ter kako jih prikazati, da bodo uporabni in razumljivi.

Ključne besede: spletna aplikacija, Python, podatkovna baza, spletni obrazec, vizualizacija.

Development of an online application for the registration of ALS experience

The goal of the diploma is to develop a web application where users can post their experiences with disease named amyotrophic lateral sclerosis or shorter ALS. Diploma describes all the technologies and tools I used though the development process, as well as design and functionality of application. Web application is the result of diploma, which was achieved using the Python programming language alongside HTML, CSS, JavaScript, Bootstrap and SQLite database. The application allows registered users to enter new ALS experience through a web form, and also allows all users to review the entries of experiences and the performance of the categories of entries through the visualization of the data. All users also have access to description of disease and list of useful telephone numbers. Through diploma I tried to figure out how to design a web form that is user-friendly and intuitive, so users would know how to post their experiences, while I also tried to figure out what data is appropriate and how to display it to be useful and understandable.

Key words: web application, Python, database, web form, visualization.

KAZALO

1 Uvod	7
2 Predstavitev bolezni ALS	9
3 Uporabljene tehnologije in orodja	11
3.1 Spletne tehnologije	11
3.1.1 HTML	11
3.1.2 CSS	12
3.1.3 Bootstrap	13
3.1.4 JavaScript.....	13
3.1.4 Python	14
3.1.5 SQLite	14
3.2 Razvojna orodja	15
3.2.1 Visual Studio Code	15
4 Ideja aplikacije	16
5 Izdelava in predstavitev aplikacije	18
5.1 Načrt in potek dela.....	18
5.1.1 Načrt ogrodja	18
5.1.2 Načrt podatkovne baze	18
5.2 Razvoj aplikacije	19
5.2.1 Knjižnice in postavitve delovnega okolja znotraj projekta	20
5.2.2 Inicializacija podatkovne baze.....	21
5.2.3 Registracija in prijava uporabnika	22
5.2.4 Spletni obrazec za vnos izkušenj	23
5.2.5 Izpis vseh izkušenj	29
5.2.6 Uspešnost kategorij.....	30
6 Sklepne ugotovitve	35
7 Literatura	36
Priloga A: Povezava za dostop do repozitorija kode aplikacije	38

SEZNAM SLIK

SLIKA 3.1: OSNOVNO HTML5 OGRODJE	12
SLIKA 3.2: CSS PRAVILO ZA OBLIKO NAVIGACIJE	13
SLIKA 3.3: JAVASCRIPT FUNKCIJA ZA PREKLAPLJANJE OB KLIKU NA GUMB	14
SLIKA 3.4: RAZVOJNO ORODJE VISUAL STUDIO CODE	15
SLIKA 5.1: OSNUTEK STRUKTURE ZAČETNE STRANI	18
SLIKA 5.2: RELACIJSKI MODEL PODATKOVNE BAZE	19
SLIKA 5.3: STRUKTURA PROJEKTA	21
SLIKA 5.4: PYTHON KODA ZA KREACIJO NOVIH TABEL V SQLITE PODATKOVNI BAZI	21
SLIKA 5.5: REGISTRACIJSKI OBRAZEC	22
SLIKA 5.6: PRIJAVNI OBRAZEC	22
SLIKA 5.7: UPORABA TEHNIKE SLEDENJE OČEM NA SPLETNEM OBRAZCU	26
SLIKA 5.8: SPLETNI OBRAZEC ZA VPISOVANJE IZKUŠENJ Z BOLEZNIJO ALS	28
SLIKA 5.9: ZAČETNA STRAN S TABELO VSEH VNOSOV	29
SLIKA 5.10: STRUKTURNI STOLPCI	32
SLIKA 5.11: STRUKTURNI KROG	33
SLIKA 5.12: PRIKAZ DELEŽEV SPOLOV IN NJIHOVE POVPREČNE STAROSTI	34

1 Uvod

Ljudje se dnevno srečujemo z boleznimi, ki nas prizadenejo na takšen ali drugačen način. Prizadene nas lahko že enostaven prehlad, ob katerem pa tudi že iz predhodnih izkušenj vemo kako reagirati nanj. Večji problem nastane v primerih, ko oseba zboli za hujšim obolenjem in ne ve kako ukrepati. Prvi korak je zagotovo zdravniška pomoč, sledi iskanje informacij preko različnih medijev.

Živimo v svetu, kjer smo obdani z nepredstavljivimi količinami podatkov. Največji proizvajalec podatkov je internet, s katerim pa se srečujemo dnevno ter na vsakem koraku. Uporabljamo ga za zabavo, izobraževanje, nakupovanje, iskanje informacij in mnogo več. Iskanje informacij je z uporabo interneta naredilo korak naprej in sam proces je postal enostavnejši in hitrejši. Z uporabo interneta so nam tako informacije glede bolezni, njenih simptomov, posledic ter zdravljenja na dosegu rok.

Problem nastane ob odkritju bolezni, ki je iz strani stroke poznana kot neozdravljiva. Ljudje kot bitja pa imamo seveda naravni instinkt preživetja in bomo skušali narediti vse, da bi tako imenovano neozdravljivo bolezen premagali. Internet z uporabo različnih forumov, blogov ter spletnih aplikacij lahko v primeru iskanja informacij glede bolezni predstavlja pravo zakladnico informacij ter mnenj, ki si jih ljudje delijo med seboj.

Glavni cilj diplomske naloge je razvoj spletne aplikacije, ki omogoča registriranim uporabnikom deljenje svojih izkušenj in spoznanj o eni izmed mnogih neozdravljivih bolezni, ALS. Cilj aplikacije je v osnovi omogočiti uporabnikom vnos svojih izkušenj z boleznijo ALS, s katero so se na tak ali drugačen način srečali v življenju. V vmesniku aplikacije je glavni vir informacij seznam vnosov izkušenj, ki jih uporabniki preko strukturiranega spletnega obrazca vpišejo ter jih delijo z drugimi. Pri oblikovanju spletnega obrazca me bo predvsem zanimalo, kako oblikovati spletni obrazec za vnos podatkov, da bo uporabniško prijazen in intuitiven.

Poleg izpisa vseh vnešenih izkušenj pa bodo imeli uporabniki aplikacije na voljo tudi ogled osnovnih statistik kategorij izkušenj, kjer pa me bo predvsem zanimalo kako te statistike predstaviti, da bodo primerne, razumljive in uporabne.

Postavil sem tudi sledečo hipotezo in raziskovalni vprašanji:

- Python v kombinaciji z ogrodjem Flask je primeren programski jezik za razvoj spletnih aplikacij, ki temeljijo na relacijski podatkovni bazi.
- Kako oblikovati spletno formo za vnos podatkov, da bo uporabniško prijazna in intuitivna?
- Kako predstaviti osnovne statistike, tako da bodo primerne in uporabne za uporabnike?

V prvem delu diplomske naloge bom na kratko opisal bolezen ALS. Gre namreč za zelo napreduvalno bolezen, kjer na določeni točki ljudje ne vedo, kaj bi še lahko naredili in zato je moj cilj razviti spletno aplikacijo za deljenje izkušenj z boleznijo.

V drugem delu diplomske naloge bom podrobneje opisal vse uporabljene metode ter orodja, ki jih bom skozi proces razvoja spletne aplikacije uporabljal. Sama aplikacija bo temeljila na treh delih, kjer bo prvi del tisti, s katerim bo uporabnik spletne aplikacije v interakciji in bo narejen s pomočjo označevalskih jezikov HTML in CSS. Drugi del bo ozadje aplikacije, kjer poteka celotna logika, le-ta pa bo napisana v programskem jeziku Python z uporabo ogrodja Flask. Vse skupaj bo povezovala tudi podatkovna baza, kjer bodo shranjeni vsi vnosi izkušenj. Skozi diplomsko nalogo bom tako tudi ugotavljal, ali je programski jezik Python primeren jezik za razvoj spletnih aplikacij, ki temeljijo na relacijski podatkovni bazi.

V tretjem delu diplomske naloge bom opisal celotno idejo aplikacije. Tu bom predvsem izpostavil uporabo aplikacije, komu je namenjena ter zakaj se lahko uporablja.

Četrto poglavje pa bo zajemalo celoten proces izdelave aplikacije, kjer bom opisal postopek razvoja, probleme, s katerimi sem se v procesu srečeval, ter poleg tega tudi predstavil, kako sama aplikacija deluje.

V zadnjem poglavju bom na podlagi predhodnih predstavil sklepne ugotovitve ter odgovore na hipotezo in raziskovalni vprašanji.

2 Predstavitev bolezni ALS

ALS ali amiotrofična lateralna skleroza je hitro napredujoča, neozdravljiva bolezen, ki prizadene motorične živčne celice v možganski skorji, možganskem deblu ter v hrbtenjači. Drugo ime, ki se pojavlja, za bolezen je bolezen motoričnega nevrona. Za ALS je značilno, da prizadene tako spodnji motorični nevron kot tudi zgornji motorični nevron, kar pa se običajno kaže z mišično oslabelostjo. Bolnike bolezen prizadene na različne načine, saj se njeni znaki pokažejo v enem izmed udov, ki pa ni nujno vedno isti. Sam vzrok bolezni ni znan. Ljudje v svojem vsakdanu uporabljamo mišice za upravljanje najrazličnejših nalog, njihovo krčenje pa upravljajo možgani, kjer pred samim gibom nastane tako ideja kot načrt gibanja. Za samo izvedbo giba nato poskrbijo živčne celice, ki pa so pri bolnikih okvarjene, zato z okvaro teh živčnih celic mišice postajajo šibke in nefunkcionalne (Inštitut za klinično nevrofiziologijo 2016). Začetek slabitve najprej zajame žarišče uda in se kasneje postopoma razširi na sosednje predele telesa, preden se razširi tudi na drugo stran telesa ter prizadene mišice, ki so odgovorne za govor, dihanje in požiranje (Kirbiš in drugi 2015).

Skozi napredovanje bolezni ALS se bolniki srečujejo z znaki, kot so »nespretnost pri rokovanju s predmeti in počasnost gibov, utrudljivost, izginjanje mišic (atrofija), spastičnost, krči, trzljaji v mišicah in drugi« (Inštitut za klinično nevrofiziologijo 2016).

Bolezen v večjem deležu prizadene starejše osebe, ki pa so v času diagnoze stare med 40 in 70 let, s povprečno starostjo 55 let. Treba pa je izpostaviti, da bolezen lahko prizadene tudi otroke in mlajše osebe. V večini pogostejše obolevajo moški, vendar pa je s povečano starostjo incidenca bolezni bolj enaka med spoloma (ALS Association 2017).

Bolniki se srečujejo z mnogimi ovirami, ki jih spremljajo iz dneva v dan. Opravljanje vsakodnevnih stvari skozi čas postaja vse težje, saj zaradi posledic mišičnih oslabelosti izgubljajo moč. V fazi, ko bolnik nima več moči, da bi samostojno opravljal vsakodnevne stvari, so na voljo številne podporne možnosti v obliki pomoči na domu, ortopedskih pripomočkov ter kirurških posegov za vstavev pripomočkov, kot je na primer cevka za prehranjevanje. Na voljo so tudi številni načini blažilnega zdravljenja, ki lahko pripomorejo predvsem h kvaliteti in udobju življenja bolnika kot tudi svojcev. Gre za

multidisciplinarni pristop, kjer se s sporočanjem diagnoze vzpostavi sodelovanje, v katerem pa so aktivni strokovnjaki številnih strok. Tu je treba izpostaviti predvsem službe, ki nudijo pomoč na domu, ter patronažne sestre, ki v kasnejših fazah bolezni obiskujejo bolnika dnevno ter jim nudijo pomoč (Inštitut za klinično nevrofiziologijo 2016). ALS bolniki so o svojem neozdravljivem stanju obveščeni takoj, lahko pa traja tudi več let, preden osebe umre, zato ima le-ta veliko časa za odločanje o svojih poslednjih fazah. Za karkoli se odloči, bo bolezen kljub temu napredovala, zato je potrebno osebo in svojce opozoriti na vse nadaljnje postopke, ki vključujejo tudi pomoč na domu. Nezmožnost premikanja bolnika je neizogibno, zato so tudi strategije za zdravljenje in pomoč na domu vnaprej predpisane. Četudi je ALS neozdravljiva bolezen lahko optimizacija oskrbe na domu poveča kvaliteto življenja ter omogoči udoben dom tako bolniku kot svojcem (Carter in drugi 1999, 707–708).

3 Uporabljene tehnologije in orodja

3.1 Spletne tehnologije

Pri procesu razvoja spletne aplikacije sem uporabil najnaprednejše spletne tehnologije, ki v kombinaciji z razvojnimi orodji omogočajo izdelavo modernih spletnih aplikacij. Za vizualni del aplikacije sem uporabil kombinacijo jezikov HTML, CSS in JavaScript, ki veljajo za standardne spletne tehnologije. Logiko aplikacije pa sem sprogramiral s pomočjo programskega jezika Python, saj je enostaven, vendar kljub temu zmogljiv. Python je sam po sebi programski jezik namenjen različnim področjem programiranja, zato sem za razvoj spletne aplikacije vanj vključil knjižnico Flask, s pomočjo katere je razvoj spletnih aplikacij poenostavljen ter razširjen z novimi možnostmi. Celotno spletno aplikacijo sem povezal tudi s podatkovno bazo SQLite, ki je kljub svoji enostavnosti dovolj zmogljiva za moje potrebe.

3.1.1 HTML

HTML predstavlja kratico za Hyper Text Markup Language in je osnova vseh spletnih strani. Gre za označevalni jezik, ki definira ogrodje spletne strani in je z obiskom spletnega mesta skozi brskalnik pretvorjen v vsebino strani. HTML jezik sestavljata znački, začetna in končna, med katerima pa je podana vsebina. Obstaja veliko različnih značk, s katerimi so določeni naslovi, slike, tabele, povezave in druge vsebine (W3Schools 2017c).

Od nastanka interneta je bilo skozi čas razvitih več verzij HTML jezika, kjer je bila ideja vsake verzije nadgraditi prejšnjo. Zadnja verzija označevalnega jezika je HTML5, ki podpira najnovejše tehnologije prikazovanja videa, zvočnih posnetkov, najnaprednejših spletnih obrazcev in veliko več (Duckett 2011, 8). Poleg samih tehnologij je ideja najnovejše verzije tudi v semantični strukturi, ki naredi pisanje HTML jezika bolj razumljivo z zagotavljanjem dodatnih informacij (Duckett 2011, 41).

Slika 3.1: Osnovno HTML5 ogrodje

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
      scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans"
      rel="stylesheet">
    <title>ALS</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <div>
      <h1>HTML5</h1>
    </div>
  </body>
</html>
```

3.1.2 CSS

CSS predstavlja kratico za Cascading Style Sheets. Gre za rešitev iz strani World Wide Web Consortiuma in je namenjen stilnemu oblikovanju spletnih strani. Pred uvedbo stilskih predlog CSS so razvijalci spletnih strani za oblikovanje uporabljali attribute in oznake HTML jezika, kot je na primer `<big>`, ki pa sam po sebi ni namenjen oblikovanju spletne strani, vendar zgolj njeni strukturi. Z možnostjo ločitve ogrodka in oblikovnih stilov z ločenimi CSS datotekami je tako proces postal enostavnejši, predvsem pa bolj organiziran (W3Schools 2017a).

S CSS stili je možno ustvarili sloge, s katerimi se določa, kako se različni HTML elementi prikazujejo na spletni strani. Tako se ustvarijo pravila za prikazovanje različnih barv, velikosti, poravnave, pozicij, robov in podobno (Duckett 2011, 227). CSS pravila se lahko povežejo z različnimi elementi HTML-ja in njihovimi izbirniki, tako da lahko z enim pravilom spremenimo obliko in izgled večih elementov na spletni strani.

Slika 3.2: CSS pravilo za obliko navigacije

```
.topnav {  
    background-color: #19323C;  
    overflow: hidden;  
    border-bottom: #2980b9 3px solid;  
    margin-bottom: 30px;  
    width: 100%;  
}  
  
.topnav a {  
    float: left;  
    display: block;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
    font-size: 16px;  
}  
  
.topnav a:hover {  
    background-color: #ddd;  
    color: black;  
}
```

3.1.3 Bootstrap

Bootstrap je najbolj popularno ogrodje, ki s pomočjo jezikov HTML, CSS in JavaScript razvijalcem spletnih strani olajša delo, saj so mnoge težave, s katerimi se pri procesu razvoja spletnih strani srečujejo, vnaprej rešene in pripravljene za uporabo. Bootstrap vsebuje veliko prednastavljenih CSS stilov, komponent ter JavaScript funkcionalnosti, ki pa so pripravljene tako, da so pravilno ter odzivno prikazane na različnih napravah (Bootstrap). Element ogrodja Bootstrap, zaradi katerega je tako popularen, je fleksibilnost uporabe mreže, s pomočjo katere elemente na spletni strani lahko razporedimo po želji in prilagodimo poglede na različnih napravah.

3.1.4 JavaScript

JavaScript je programski jezik, s pomočjo katerega spletni strani dodamo možnost interakcij ali obnašanj, ki jih s pomočjo jezikov HTML in CSS ni mogoče doseči. JavaScript je kompakten, a zelo prilagodljiv programski jezik, s pomočjo katerega je statičnost spletnih strani mogoče spremeniti v svet interakcij. Od najenostavnejših klikov na gumbe do privlačnih animacij ter bolj zahtevne logike delovanja spletne strani (Mozilla Developer Network 2017). Deluje podobno kot CSS, saj prav tako zahteva, da

se logika funkcionalnosti poveže na HTML element, nato pridobi možnost upravljanja z njim in tako je z njegovo uporabo razvijalcem na voljo implementacija najrazličnejših funkcionalnosti.

Slika 3.3: JavaScript funkcija za preklapljanje ob kliku na gumb

```
function classToggle() {  
    var x = document.getElementById("myTopnav");  
    if (x.className === "topnav") {  
        x.className += " responsive";  
    } else {  
        x.className = "topnav";  
    }  
}
```

3.1.4 Python

Python je visoko nivojski programski jezik, ki je enostaven, a zelo priljubljen. Klasificiran je kot objektno-usmerjen programski jezik in je primeren za razvoj najrazličnejših programov, vse od skriptnih programov do programov za podatkovno analizo in kompleksnih spletnih aplikacij. Python je zelo priljubljen zaradi svoje elegantne sintakse in dinamičnega tipkanja, kar ga naredi idealnega za hiter razvoj različnih aplikacij. Programski jezik je leta 1991 razvil Guido van Rossum, poimenoval pa ga je po televizijski oddaji (Swaroop C. H 2013, 15). Razlog za širšo priljubljenost programskega jezika Python je tudi dejstvo, da gre za odprtokodni programski jezik, kar pomeni, da se razvija s pomočjo prispevkov skupnosti razvijalcev s celotnega sveta (Python). Posebna korist v tem, da gre za odprtokodni programski jezik, pa je to, da vsakdo lahko prispeva s svojim delom kode in tako se razvija veliko število knjižnic za lažji proces razvoja aplikacij. Dober primer knjižnice, ki je bila razvita iz strani skupnosti, je Flask, ki je namenjen hitremu in enostavnemu razvoju spletnih aplikacij.

3.1.5 SQLite

SQLite je odprtokodni sistem za upravljanje relacijskih baz podatkov. Bil je zasnovan tako, da je enostaven za uporabo že od namestitve naprej. Znan je predvsem po dobri prenosljivosti, enostavnosti, učinkovitosti, kompaktnosti ter zanesljivosti. Gre za vgrajeno bazo podatkov, saj soobstaja znotraj aplikacije in ne kot samostojni proces (Allen in Owens 2010, 2). Upravljanje SQLite baze poteka z SQL jezikom, ki je

standardni jezik za upravljanje podatkov v podatkovnih bazah, saj preko njega lahko podatke shranjujemo, pridobivamo ter z njimi manipuliramo.

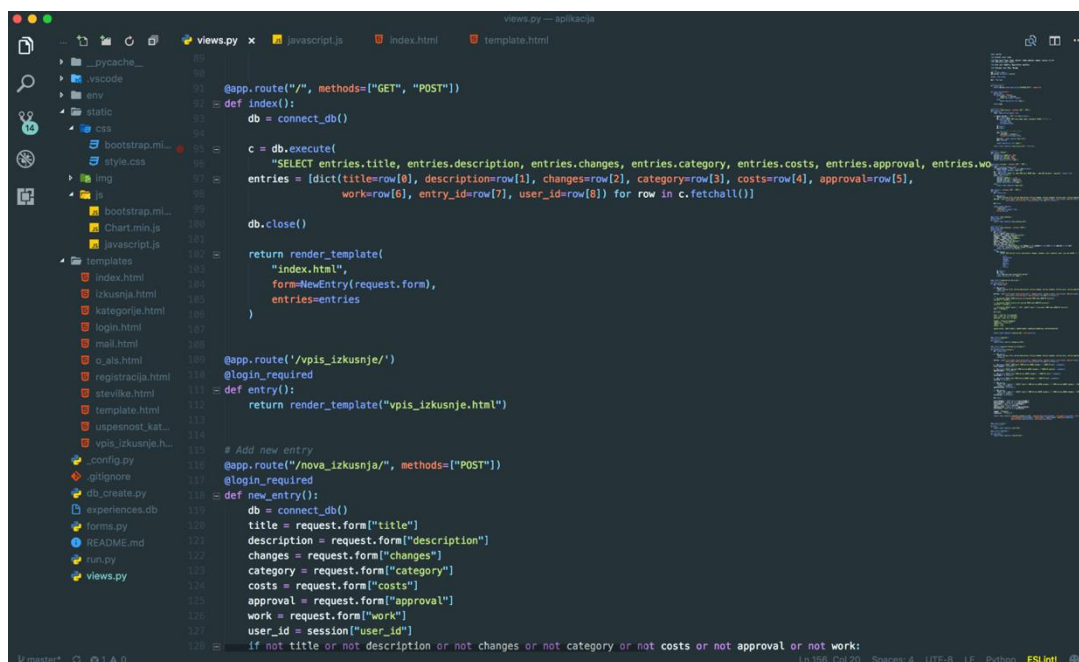
3.2 Razvojna orodja

3.2.1 Visual Studio Code

Odločil sem se za izbiro razvojnega orodja Visual Studio Code, ki je produkt podjetja Microsoft.

Visual Studio Code je lahek, a zelo učinkovit kodni urejevalnik, ki podpira vrsto operacijskih sistemov ter programskih jezikov. Njegove izrazite prednosti, zaradi katerih sem se odločil za uporabo, so hitrost delovanja, podpora pri pisanju sintakse programskega jezika, prilagodljivost ter možnost dodajanja različnih razširitev.

Slika 3.4: Razvojno orodje Visual Studio Code



4 Ideja aplikacije

Ideja in uporaba spletne aplikacije je zelo preprosta, saj gre v splošnem za aplikacijo, ki registriranim uporabnikom omogoča, da z drugimi javno delijo svoje izkušnje in spoznanja z boleznijo ALS. Vodilo do ideje je bila lastna izkušnja z bitko proti bolezni v družini, saj se v poznejših fazah te neozdravljive bolezni ljudje soočajo s težavo, ker ne vedo, na koga bi se lahko obrnili, kaj bi lahko še poizkusili ter kako bi lahko bolniku omogočili boljšo kvaliteto življenja. Podobnih spletnih aplikacij, z izjemo spletnih forumov, v splošnem primanjkuje, saj bi si z njimi ljudje med seboj lahko delili izkušnje in ideje ter tako morda komu pomagali.

Osnovne funkcije aplikacije so omogočene vsem uporabnikom, za objavljanje novih izkušenj pa je potrebna registracija in prijava.

Vpis nove izkušnje poteka preko strukturiranega spletnega obrazca, kjer se nato ob kliku na gumb za oddajo izkušnje v ozadju izvede funkcija, ki z uporabo SQL jezika v podatkovno bazo zapiše izkušnjo, le-ta pa se tudi doda na seznam vseh izkušenj na začetni strani spletne aplikacije.

Na začetni strani so v obliki seznama z glavnimi lastnostmi prikazane vse izkušnje, ki so zapisane v podatkovni bazi, s klikom na gumb »Preberi več« pa lahko vsak uporabnik dostopi do strani, kjer je zapisana celotna izkušnja in dobi vpogled do vseh vpisanih lastnosti. Seznam omogoča tudi določitev števila vrstic, ki se prikažejo na strani, pod tabelo pa je dodana tudi paginacija, kjer lahko uporabnik dostopa do vseh vnosov na drugih straneh. Za pomoč iskanja specifičnih vnosov je uporabniku na voljo tudi iskalno vnosno polje.

Spletna aplikacija ima tudi več podstrani in vsaka služi svojemu namenu. Podstrani so:

- *vnosi*, kjer so v tabeli izpisani vsi vnosi shranjeni v podatkovni bazi,
- *uspešnost*, kjer lahko uporabniki pregledujejo osnovne statistike kategorij izkušenj,
- *o ALS*, kjer je na voljo nekaj informacij o sami bolezni, ter
- *pomembne številke*, kjer so na voljo razne telefonske številke, kamor lahko uporabniki pokličejo za več informacij o sami bolezni ali pomoč.

Spletna aplikacija omogoča pregled osnovnih statistik, ki pa so deljene po kategorijah, ki so na voljo uporabniku ob vnašanju nove izkušnje. Tako je mogoče videti, katera kategorija vnosov je bolj uspešna in katera je manj uspešna. Skozi diplomsko nalogo bom ugotavljal kako te statistike predstaviti, da bodo razumljive in uporabne.

Pri celotnem oblikovanju spletne aplikacije je bila pozornost tudi na upoštevanju trendov uporabe različnih elektronskih naprav. Še ne tako dolgo nazaj so bile spletne strani oblikovane s statično širino, zato se je tudi predvidevalo, da bodo vsi uporabniki imeli podobno uporabniško izkušnjo na spletni strani. S prihodom pametnih telefonov, ur in podobnih naprav se je pristop k razvoju spletnih strani spremenil in začel se je uporabljati tako imenovan odziven spletni način. Z uporabo tehnologij, kot sta HTML5 in CSS3, se spletna stran prilagodi vsaki napravi, na kateri je tako zagotovljena tudi optimalna uporabniška izkušnja (Frain 2012, 7). Z uporabo sodobnih spletnih tehnologij sem spletno aplikacijo oblikoval za optimalen prikaz na vseh napravah z različnimi širinami zaslonov. V splošnem bi lahko rekel, da gre za spletno aplikacijo, ki tvori skupnost vseh ljudi, ki so se ali se srečujejo z boleznijo ALS, zato sem samo aplikacijo poimenoval ALSkupnost.

5 Izdelava in predstavitev aplikacije

5.1 Načrt in potek dela

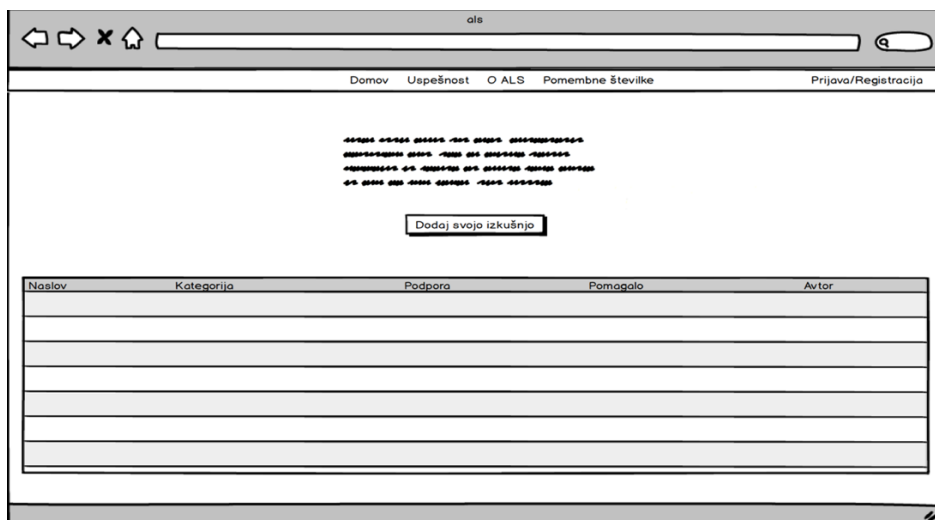
Razvoj spletne aplikacije se je odvijal na operacijskem sistemu macOS, ki je s kombinacijo uporabe komandne konzole in razvojnega orodja za pisanje kode odlično okolje za razvijanje.

Proces razvoja se je delil na razvoj uporabniškega vmesnika spletne aplikacije (front-end) in kode na strežniku, ki skrbi za logiko same aplikacije (back-end).

5.1.1 Načrt ogrodja

Pred samim začetkom pisanja kode spletne aplikacije sem s pomočjo orodja Balsamiq zrisal preprost osnutek izgleda aplikacije. S tem sem si lahko v nadaljevanju pomagal predvsem pri razvijanju uporabniškega vmesnika in si tako vnaprej pripravil strukturo elementov aplikacije.

Slika 5.1: Osnutek strukture začetne strani



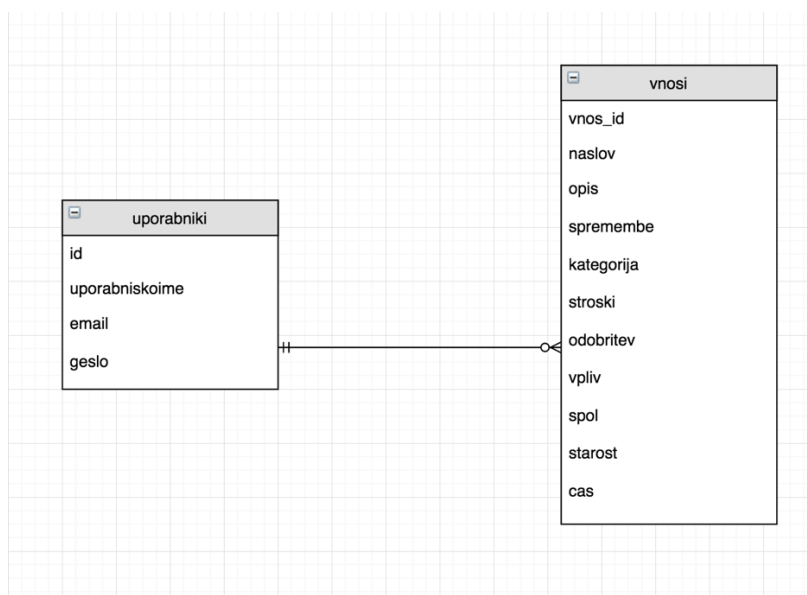
5.1.2 Načrt podatkovne baze

Podatkovna baza je najbolj osnoven sestavni del ideje celotne spletne aplikacije in je bil zato prvi korak v proces njenega razvoja. Kljub temu da gre za zelo preprosto strukturo podatkovne baze, sem na začetku faze načrtovanje najprej pregledal druge spletne aplikacije, ki imajo podobno strukturo delovanja. Na osnovi identifikacije podatkov, ki

jih želim v podatkovno bazi zbirati, sem naredil preprost model podatkovne baze, ki ima zgolj 2 entiteti, ti pa sta v povezavi 1:N. Ti entiteti sta:

- uporabniki in
- vnosi.

Slika 5.2: Relacijski model podatkovne baze



Entiteta »uporabniki« opisuje in shranjuje registrirane uporabnike spletne aplikacije. Ob registraciji uporabnik vnese svoje uporabniško ime, elektronski naslov ter geslo, nato pa se podatki shranijo v tabelo in uporabniku je omogočena prijava. Sama entiteta se povezuje z entiteto »vnosi«, s katero je v razmerju 1:N, kar pomeni, da ima lahko en uporabnik nič ali več vnosov. V entiteti "vnosi" se shranjujejo vsi podatki, ki jih prijavljeni uporabniki vnesejo v strukturiran spletni obrazec, ki se nato preko strežnika, z uporabo SQL jezika, zapišejo v tabelo. Posamezni vnos definirajo podatki, ki jih opisujem v drugem poglavju, ti pa so: naslov, opis, opažene spremembe, kategorija, stroški, odobritev medicine, vpliv na počutje, spol in starost bolnika ter čas vnosa.

5.2 Razvoj aplikacije

Razvoj spletne aplikacije, ki je potekal v orodju Visual Studio Code, se je začel s kreacijo novega projekta, katerega sem poimenoval *ALSizkusnje*. Vsa logika aplikacije je napisana v programskem jeziku Python, ki odlično podpira vgradnjo poljubnih knjižnic potrebnih za razvoj. Python knjižnice so lahko nameščene globalno na sistemski ravni, vendar sem

jih za boljše upravljanje z njimi s pomočjo uporabe Pythonovega virtualnega okolja namestil na raven projekta in so tako na voljo za uporabo znotraj projekta. Posamezni projekti tako postanejo izolirani in ne vplivajo en na drugega (Python 2017).

5.2.1 Knjižnice in postavitve delovnega okolja znotraj projekta

Ob prednaloženem programskem jeziku Python na operacijskem sistemu je na voljo tudi upravljalac paketov PIP. Gre za funkcijo, ki nam omogoča enostavno razširitev Python aplikacije z dodajanjem poljubnih modulov oziroma knjižnic. Za razvoj aplikacije sem uporabil knjižnico Flask, ki sem jo v projekt aplikacije namestil z ukazom *pip install Flask*. Flask je preprosta knjižnica za razvoj spletnih aplikacij s pomočjo programskega jezika Python, s katerim lahko preko strežnika prikazujemo HTML datoteke, se povežemo s podatkovno bazo, prikazujemo dinamično vsebino in še veliko več.

Po uspešni namestitvi je potrebno knjižnico tudi uvoziti v samo datoteko projekta, to pa sem dosegel z ukazom *from flask import Flask*. Uvoz knjižnice v Python datoteko, ki sem jo poimenoval *views.py*, je bil tudi prvi korak za uspešen zagon lokalnega strežnika, preko katerega je omogočen razvoj spletne aplikacije brez potrebe po gostujočem strežniku. Datoteka *views.py* predstavlja celotno logiko aplikacije. Vanjo so uvožene vse knjižnice, zadolžena pa je tudi za omogočanje dostopa do podatkovne baze, usmerjanja internetnih zahtev ter prikazovanja datotek.

Za lažjo organizacijo dela sem projekt strukturiral po datotekah, med katerimi ima vsaka svoj namen, ter jih umestil v primerno mapo. Tako so poleg datoteke *views.py* tu še datoteka *_config.py*, v kateri so shranjene sistemske nastavitve, *db_create.py*, s pomočjo katere se kreira podatkovna baza, *forms.py*, v kateri so shranjeni razredi vseh spletnih obrazcev, ter *run.py*, ki omogoča postavitve lokalnega razvojnega strežnika. Vse HTML datoteke za prikaz strukture spletne aplikacije so shranjene v mapi *templates*, medtem ko so vse slike, CSS stili in JavaScript funkcije shranjene v mapi *static*.

Slika 5.3: Struktura projekta



5.2.2 Inicializacija podatkovne baze

Pri razvoju aplikacij je na voljo veliko različnih vrst podatkovnih baz za shranjevanje uporabnikov in njihovih vnosov, pri svoji aplikaciji pa sem se odločil za uporabo podatkovne baze SQLite. Gre za podatkovno bazo, ki je enostavna za uporabo ter predhodno tudi integrirana v samo namestitev Pythona na računalniku.

Narejen načrt podatkovne baze še ni dovolj za delovanje z njo, saj je treba ustvariti potrebne tabele z načrtovanimi stolpci. Upravljanje SQLite podatkovne baze poteka preko SQL jezika, tega pa lahko enostavno zaženemo tudi preko Python datoteke.

V svoji *db_create.py* datoteki sem najprej uvozil SQLite modul, ki nam omogoča dostop in interakcijo z SQLite podatkovno bazo. Iz datoteke *_config.py* sem nato uvozil pot do lokalne podatkovne baze in z njo ustvaril povezavo. Ob ustvarjeni povezavi s podatkovno bazo lahko poljubno vstavljamo, brišemo ali spreminjamo vnose.

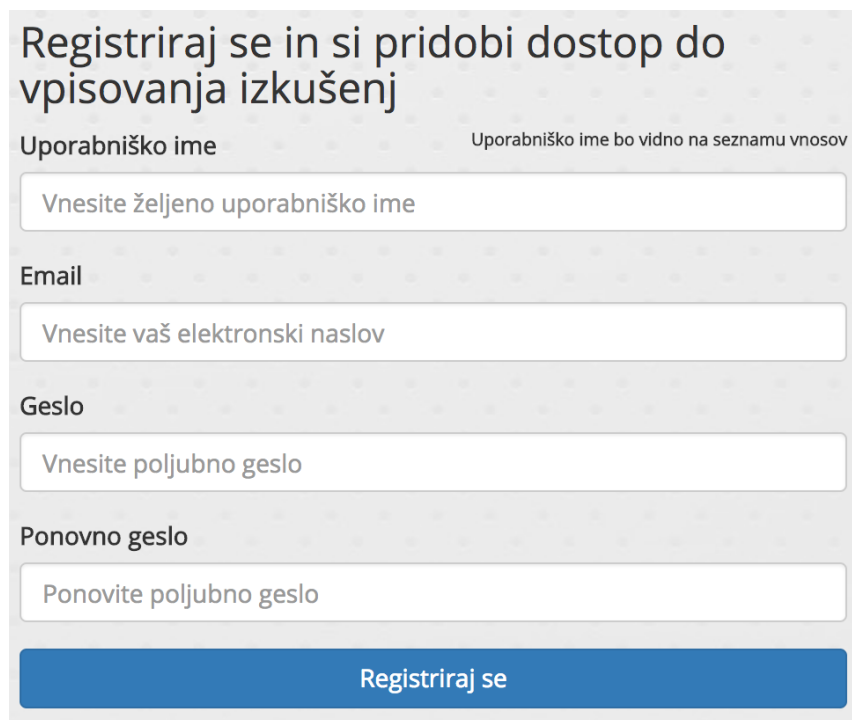
Slika 5.4: Python koda za kreacijo novih tabel v SQLite podatkovni bazi

```
import sqlite3
from _config import DATABASE_PATH
with sqlite3.connect(DATABASE_PATH) as connection:
    c = connection.cursor()
    c.execute("""CREATE TABLE uporabniki(id INTEGER PRIMARY KEY AUTOINCREMENT,
        uporabniskoime TEXT NOT NULL UNIQUE, email TEXT NOT NULL UNIQUE, geslo TEXT NOT NULL)""")
    c.execute("""CREATE TABLE vnosi(vnos_id INTEGER PRIMARY KEY AUTOINCREMENT,
        naslov TEXT NOT NULL, opis TEXT NOT NULL, spremembe TEXT NOT NULL, kategorija TEXT NOT NULL,
        stroški TEXT NOT NULL, odobritev TEXT NOT NULL, vpliv TEXT NOT NULL, spol TEXT NOT NULL, starost INTEGER NOT NULL,
        uporabnik_id INTEGER NOT NULL REFERENCES uporabniki(id), cas TEXT NOT NULL)""")
```

5.2.3 Registracija in prijava uporabnika

Registracija omogoči uporabniku prijavo v spletno stran, s katero uporabnik pridobi možnost objavljanja novih izkušenj. V registracijski obrazec mora uporabnik vnesti željeno uporabniško ime, elektronski naslov ter geslo, katerega mora tudi ponoviti.

Slika 5.5: Registracijski obrazec



Registriraj se in si pridobi dostop do vpisovanja izkušenj

Uporabniško ime Uporabniško ime bo vidno na seznamu vnosov

Email

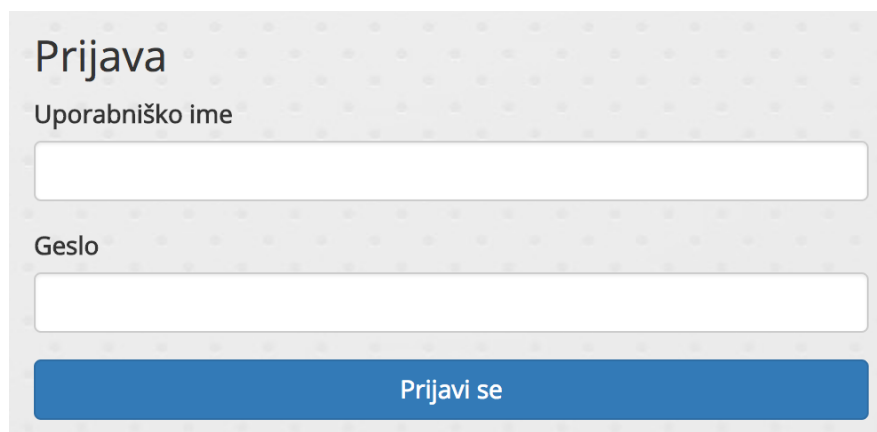
Geslo

Ponovno geslo

Registriraj se

Uporabnik mora za prijavo vnesti svoje izbrano uporabniško ime ter geslo. Ob uspešni prijavi se mu omogoči dostop do obrazca za vnos nove izkušnje ter dostop do možnosti odstranjevanja svojih vnesenih izkušenj.

Slika 5.6: Prijavni obrazec



Prijava

Uporabniško ime

Geslo

Prijavi se

5.2.4 Spletni obrazec za vnos izkušenj

Najpomembnejši gradnik spletne aplikacije je strukturiran spletni obrazec, preko katerega registrirani uporabniki lahko vnašajo svoje izkušnje z boleznijo ALS, ki se nato shranijo v podatkovno bazo.

Gre za vrsto interakcije med uporabnikom in spletno aplikacijo, preko katere uporabnik pošlje podatke vnešene v polja obrazca. Polja spletnega obrazca so lahko različnih tipov, odvisno od podatkov, ki jih želimo zajeti (W3Schools 2017b).

Spletni obrazec sem strukturiral tako, da vnos izkušnje vsebuje potrebne podatke, ki lahko vsem uporabnikom, ki berejo vnose ter iščejo pomoč, pomagajo pri razumevanju vnosa ter posledično pri morebitni uporabi izkušnje v svojem življenju.

Spletni obrazec za vnos nove izkušnje z boleznijo ALS predstavlja srce celotne aplikacije, zato me je pri oblikovanju le-tega zanimalo kako ga naredi, da bo uporabniško prijazen in intuitiven. Pri raziskovanju vprašanja uporabniške izkušnje spletnih obrazcev se nisem poslužil izvajanja testov uporabnosti, vendar sem upošteval nasvete, ki sem jih zasledil ob pregledu literature, ter se zgledoval po dobrih praksah oblikovanja intuitivnih spletnih obrazcev.

Kakšen pravzaprav je dober spletni obrazec? Po Jarrett idr. (2008, xi) mora biti dober obrazec »popolnoma jasen in popolnoma samoumeven«. Osebno mislim, da je oblikovanje takšnih obrazcev zelo pomembno, saj se z njimi srečujemo skoraj dnevno, pa naj bodo to spletni obrazci za registracijo, obrazci za pošiljanje pošte ali podobno. Jarrett idr. (2008) razdelijo svojo teorijo dobrih obrazcev na tri plasti: odnos, pogovor in videz. Pri odnosu gre predvsem za odnos med tistim, ki sprašuje, in tistim, ki odgovarja. Plast pogovora opredeljuje vprašanja, navodila za izpolnjevanje ter kako so vprašanja deljena. Videz obrazca pa opredeljuje, kako obrazec izgleda, kakšne barve so uporabljene ter kakšnega tipa vnos je (Jarrett in drugi 2008, 6). Sam sem se osredotočil predvsem na plast pogovora in videza, saj me je zanimalo kako oblikovati svoj spletni obrazec za vpisovanje izkušenj tako, da bo ta razumljiv.

Pri obrazcih gre za obliko komunikacije, kjer obrazec sprašuje uporabnika vprašanja, na katera želimo odgovore. Hitro pa so lahko obrazci tudi polni nepotrebnih elementov, ki ga v celoti naredijo nepreglednega in tako morda uporabnika tudi odvrnejo od izpolnjevanja. Poleg oblike obrazcev pa je za prijazno uporabniško izkušnjo pomembna tudi dolžina obrazca, saj lahko predolg obrazec, ki vsebuje veliko potencialno nepomembnih in nerelevantnih vprašanj, prav tako odvrne uporabnika od izpolnjevanja. V svoj obrazec sem tako vključil zgolj vprašanja, ki jih potrebujemo, da lahko z odgovori na njih nekomu pomagamo, ter dve dodatni vprašanji za splošno statistiko. Držal pa sem se tudi pravila, ki pravi, da je optimiziran obrazec »tako dolg, da postavlja ustrezna vprašanja, da je dovolj kratek, da minimizira prizadevanje uporabnika, in da so vprašanja postavljena v kontekstu celotnega odnosa« (Jarrett in drugi 2008, 24).

Prvo vprašanje, na katerega sem moral odgovoriti glede oblike obrazca, je, katero vrsto kontrole obrazca bom uporabil. Gre za vprašanje tipa vnosa podatka, ki pa ni vedno enak. Za enostaven zajem tekstovnega tipa podatkov sem uporabil vnosno vrsto kontrole. V HTML jeziku vnosno vrsto kontrole dobimo z uporabo značke `<input type="text">`. Gre za polje, ki od uporabnika spletne aplikacije pričakuje vnos besedila. Za ostale podatke, ki jih želim zajemati, sem uporabil izbirno vrsto kontrole, ki ob kliku uporabniku prikaže izbrane vrednosti kot odgovore. Okno z možnimi odgovori v spustnem meniju dobimo z uporabno HTML značko `<select>`, kjer znotraj nje vsak možen odgovor vpišemo v značko `<option>`. Tako lahko uporabnik ob vnosu nove izkušnje pri vprašanjih, ki od njega pričakujejo izbiro med preddefiniranimi možnostmi, enostavno izbere možnost, ki predstavlja njegov odgovor, ne da bi moral sam vnašati odgovor.

Obrazec za vnos izkušnje vsebuje 9 vprašanj, izmed katerih so 4 vprašanja tekstovnega tipa vnosa odgovora, 5 vprašanj pa ima izbirni vnos odgovora. Prvo tekstovno vnosno polje predstavlja naslov izkušnje, s katerim lahko ostali uporabniki lažje identificirajo vsebino samega vnosa izkušnje med brskanjem po seznamu vseh izkušenj. Sledita dva tekstovna vnosna polja za besedilo, kjer uporabnik vnese opis celotne izkušnje ter opažene spremembe, ki jih je zaznal pri bolniku. Oblikovno gre za polja, ki sta nekoliko večja kot polje za vnos naslova, saj se od uporabnika pričakuje nekoliko daljši vnos.

Pri nadaljevanju vnosa podatkov v spletni obrazec se uporabniki srečajo z vprašanji, ki od njih zahtevajo izbiro preddefiniranega odgovora. Prvo izbirno polje predstavlja izbiro

kategorije, kjer so preddefinirane kategorije vnosov, po katerih se vnosi izkušenj kategorizirajo ter prikazujejo svojo osnovno statistiko na podstrani »uspešnost«. Kategorije so definirane po področjih lastnih izkušenj, s katerimi sem se srečal, te pa so:

- prehrana,
- fizična aktivnost,
- uporaba pripomočkov,
- terapija,
- zdravila in
- drugo.

Izbirna vrsta kontrole lahko predstavlja težavo, če na seznamu možnih odgovorov ni odgovora, ki bi ga uporabnik želel izbrati, zato se to lahko enostavno reši z dodatnim odgovorom »drugo« (Jarrett in drugi 2008, 92). Poleg tega sem dodal za primarno izbran odgovor prednastavljeno navodilo uporabniku, ki sporoča, da je potrebno izbrati ustrezen odgovor. Tako se lahko, če uporabnik polje preskoči, izognemo napačnemu vnašanju izkušenj. V primeru, da uporabnik polje preskoči in želi objaviti izkušnjo, se prikaže opozorilo, da je izpustil obvezno polje.

Izbira stroškov je naslednje polje, kjer uporabnik izbere med preddefiniranimi kategorijami stroškov, ki so v določenem razponu. Na voljo ima tudi izbiro možnosti »ni stroškov«, saj ni nujno, da je moral za svojo izkušnjo plačati. Prvotna ideja je bila uporabiti definicijo mesečnih stroškov, vendar bi nastal problem ob vnašanju enkratnih stroškov. Tako lahko izbira stroškov vključuje oboje.

Vsaka izkušnja je nekako odobrena iz strani medicine ali pa tudi ne, zato naslednje polje sprašuje uporabnika, ali je za svojo izkušnjo dobil odobritev oziroma podporo medicine ali ne, kjer mu je na voljo tudi odgovor »ne vem«, v primeru da se z osebjem medicine ni pogovarjal o tem.

Vsaka deljena izkušnja potrebuje tudi splošen odziv na to, kako je izkušnja vplivala na počutje, zato sem v obrazec dodal tudi to vprašanje, kjer so na voljo odgovori v obliki Likartove lestvice z možnimi odgovori:

- zelo negativno,
- negativno,

- niti negativno niti pozitivno,
- pozitivno in
- zelo pozitivno.

Na koncu obrazca pa sta polja, ki sprašujeta po spolu ter starosti bolnika, s katerim je bila izkušnja pridobljena. Polje za definicijo spola bolnika je izbirne vrste kontrole, polje za starost pa je tekstovno vnosne vrste kontrole, kjer se lahko vnese samo številko.

Po uspešno izpolnjenem spletnem obrazcu mora uporabnik poslati svojo izkušnjo tudi na strežnik, to pa stori s klikom na gumb »Dodaj izkušnjo«, ki je postavljen na sam konec spletnega obrazca. Ob kliku na gumb se spletni obrazec preko HTML značke `<form>` z metodo POST pošlje na strežnik, kjer Python funkcija, namenjena usmerjanju HTTP zahtev, vse podatke pridobljene iz vnosnih polj obrazca s pomočjo SQL jezika vnese v podatkovno bazo.

Po vprašanju uporabe vrste kontrol me je zanimalo, kam postaviti vprašanje, ki se navezuje na specifično polje za odgovor. Z uporabo tehnike sledenja očem so ugotavljali, kako se uporabniki premikajo po strani, kjer je obrazec. Ob izpolnjevanju spletnih obrazcev se uporabniki osredotočijo predvsem na oznake in polja, tako da spregledajo drugo večino vsebine (Jarrett in drugi 2008, 125).

Slika 5.7: Uporaba tehnike sledenje očem na spletnem obrazcu



Vir: Jarrett in drugi (2008, 125).

Iz slike je razvidno tudi, da se uporabnik redko kdaj z očmi premika po desni strani, zato je oznake oziroma vprašanja najbolje postavljati levo od polj ali nad njimi (Jarrett in drugi 2008, 126). Postavitev oznak levo od polj je dobro zaradi manjše porabe strani in tako posledično manjšega obrazca, vendar postavitev oznak nad polji omili skakanje oči med elementi (Jarrett in drugi 2008, 127). Pri svojem obrazcu sem se odločil za postavitev vprašanj nad polji, saj so vprašanja daljša in bi leva postavitev porabila preveč prostora. Zgornja postavitev v večini tudi zmanjša čas, ki je potreben za izpolnitev obrazca. Ker sta tako oznaka oziroma vprašanje ter polje izredno blizu, povezovanje le-teh zahteva malo napora. V celoti se obrazec izpolnjuje v eni smeri, dol, zato taka postavitev ohrani hitro pot izpolnjevanja (Wroblewski 2008, 89).

Obrazec vsebuje vprašanja, ki tvorijo celoto zgolj tako, da se odgovori na vsa vprašanja, zato sem vsa polja, razen dveh, ki sprašujeta po spolu in starosti bolnika, označil za obvezna. Za indikacijo obveznih polj je več možnosti, a zaradi splošne spletne uporabnosti je dobro priporočilo pri obveznih poljih vstaviti zvezdico ter na vrhu obrazca dodati pojasnilo, da zvezdica označuje obvezna polja (Jarrett in drugi 2008, 134). Vedno obstaja verjetnost, da uporabnik ne bo upošteval navodil, zato sem ob morebitnih nepopolno izpolnjenih obrazcih dodal prikaz sporočila, ki uporabniku ob osvežitvi obrazca sporoči, da ni izpolnil vseh obveznih polj.

Vsebinsko se obrazec deli na dva dela in sicer prvi del sprašuje po sami izkušnji, drugi del pa po lastnostih bolnika. Enostavna vsebinska ločitev vprašanj poenostavi izpolnjevanje obrazcev spletnim uporabnikom (Jarrett in drugi 2008, 158). Med obema deloma sem dodal nekaj prostora, tako da ju je lažje ločiti.

Dodatna pomoč pri izpolnjevanju obrazca je lahko tudi *placeholder* atribut v HTML znački, s katerim v tekstovno vnosno polje dodamo dodatna navodila, kaj točno se v polje vpisuje. Oznake oziroma vprašanja bi morala sama po sebi biti dovolj jasna, da uporabnik zna izpolniti obrazec, vendar dostikrat ni tako (Wroblewski 2008, 161). Tako z uporabo dodatne pomoči začasnega besedila v poljih lahko pomagamo uporabniku uspešno izpolniti vsa obvezna in neobvezna polja.

Za dodatno orientacijo med izpolnjevanjem spletnega obrazca sem dodal efekt, ki obarva zunanji del polja, kjer se uporabnik trenutno nahaja. Navigacija po samem obrazcu je tako nekoliko olajšana.

Slika 5.8: Spletni obrazec za vpisovanje izkušenj z boleznijo ALS

Vpis nove izkušnje

Polja označena z * so obvezna

Naslov izkušnje *

Opis izkušnje *

Vnesite celoten opis vaše izkušnje

Opažene spremembe *

Vnesite spremembe, ki ste jih opazili med izkušnjo

Kategorija *

Izberi kategorijo

Stroški *

Izberi vsoto stroškov

Ali ste za izkušnjo imeli podporo oziroma odobritev medicine? *

Izberi odgovor

Kako se vam zdi, da je izkušnja vplivala na počutje bolnika? *

Izberi počutje

Spol bolnika

Izberi spol

Starost bolnika

Vnesite starost bolnika

Dodaj izkušnjo

5.2.5 Izpis vseh izkušenj

Začetna stran spletne aplikacije ponuja uporabniku takojšen pogled na vse objavljene izkušnje, tako na svoje kot na izkušnje drugih uporabnikov, ter dostop do spletnega obrazca preko klika na gumb. Izpis vseh izkušenj shranjenih v podatkovni bazi je predstavljen v obliki tabele, kjer vsak stolpec predstavlja vsebino vprašanja iz obrazca, vsaka vrstica pa predstavlja svoj vnos. Izpis vseh vnosov je dinamičen, saj preko SQL poizvedbe na strani strežnika pridobim vse vnose, ki jih nato strežnik pošlje naprej do HTML datoteke, ki skrbi za prikaz strani. V HTML datoteki tako s pomočjo dodatkov, ki jih omogoča Python knjižnica Flask, preko uporabe zanke izpiše vse vnose v tabelo. V sami tabeli pa niso predstavljeni vsi podatki vnosov, saj so opisi izkušenj ter opažene spremembe lahko enostavno predolga besedila, zato sem na koncu tabele k vsakemu vnosu dodal gumb »preberi več«, ki uporabnika usmeri na podstran vnosa, kjer mu je na voljo popoln vpogled v vse vnešene podatke posameznega vnosa. Poleg tega gumba pa je tudi gumb za odstranitev vnosa, ki ga vidi zgolj uporabnik, ki je avtor objavljenega vnosa. Za izbris svojega vnosa je potrebna dodatna potrditev.

Slika 5.9: Začetna stran s tabelo vseh vnosov

ALSkupnost		Vnosi	Uspešnost	O ALS	Pomembne številke	frasza, pozdravljen!	Odjava
Imate izkušnje z boleznijo ALS? Želite svoje izkušnje deliti z drugimi? Mislite, da lahko s tem komu pomagate?							
Dodaj svojo izkušnjo							
Vnosi		Išči:					
Prikaži 10 zapisov	Naslov	Kategorija	Stroški	Podpora / odobritev medicine	Vpliv na počutje	Avtor	Čas vnosa
	Konopljinna semena	Prehrana	50 - 100 €	Ne	Zelo pozitivno	janezb	27-08-2017 08:09
	Joga	Fizična aktivnost	Ni stroškov	Da	Zelo pozitivno	janezb	27-08-2017 08:08
	Jutranji sprehodi	Fizična aktivnost	Ni stroškov	Da	Pozitivno	tinika	27-08-2017 08:07
	Prehranjevanje z borovnicami in aronijo	Prehrana	50 - 100 €	Ne vem	Zelo pozitivno	frasza	27-08-2017 08:05
	Popoldanska telovadba	Fizična aktivnost	10 - 50 €	Ne vem	Zelo pozitivno	frasza	27-08-2017 08:05
	Vstavitve cevke za hranjenje	Uporaba pripomočkov	50 - 100 €	Da	Pozitivno	frasza	27-08-2017 08:04
Prikazujem 1 do 6 od 6 zapisov		Pred. 1 Nasl.					

Nad samo tabelo vnosov pa je pozicioniran gumb, s katerim registrirani in prijavljeni uporabniki lahko dostopajo do spletnega obrazca za vnašanje nove izkušnje. V primeru, da uporabnik ni prijavljen, ga klik na gumb preusmeri na stran s prijavnim obrazcem.

5.2.6 Uspešnost kategorij

Uporabniki imajo na voljo pregledovanje vseh obstoječih vnosov izkušenj kot tudi dodajanje novih, vendar želim uporabnikom dodati tudi vpogled v splošno uspešnost teh vnosov, zato sem v aplikacijo dodal kategorije izkušenj, po katerih lahko s pomočjo SQL jezika računam statistiko in na podlagi nje prikažem celotno uspešnost te kategorije in tako morda s širšim pogledom na kategorijo vnosov uporabniku ponudim dodatne informacije o samih vnosih.

Prikaz vseh potrebnih podatkov bi lahko dosegel z uporabo tabel, vendar bi z njimi dosegel učinek statičnosti in nerazumevanja, saj ni nujno, da vsi uporabniki znajo prebrati statistiko iz tabel. Živimo v času, ko so podatki vse povsod in smo lahko tako tudi podvrženi preobremenjenosti z njimi (Murray 2013, 1). Na srečo pa smo ljudje vizualna bitja, zato sem se odločil podatke prikazati v obliki grafike. Vizualizacija podatkov je najhitrejši način komuniciranja s podatki, saj lahko le določeni zaznajo vzorce med vrsticami številke v tabelah, medtem ko lahko na primer stolpčni diagram razberejo že otroci in iz njega povzamejo ugotovitve (Murray 2013, 1). Skozi uporabo grafičnega prikaza podatkov sprejemamo ključne informacije na vizualen način, kakor tudi na jedrnat in hiter način (Paniello in drugi 2011, 886). V namen raziskovanja uspešnosti kategorij bi bil navaden grafični prikaz podatkov zadosten, vendar s spletno tehnologijo, ki nam je na voljo, lahko dosežemo tudi interaktiven grafični prikaz. Z uporabo dinamičnih, interaktivnih grafov lahko uporabniku damo možnost boljšega raziskovanja podatkov (Murray 2013, 2). Grafi moje spletne aplikacije zaradi enostavnosti niti niso tako interaktivni, a uporabnik lahko z enostavnimi kliki izklaplja izbrane množice podatkov ali se s pomočjo miške orientira po grafu, zato lahko rečemo, da »gre za proces odkrivanja ter omogočanja bralcem, da se premaknejo od samo branja podatkov do dejanskega pogleda na njih« (Kirk 2012, 12).

Zaradi raznolikosti vizualizacije podatkov je proces identificiranja najbolj učinkovite in primerne rešitve eden izmed najbolj pomembnih korakov pri oblikovanju vizualizacije. Gre za ugotavljanje, kako lahko prikažemo tiste stvari, ki jih želimo povedati (Kirk 2012, 84). Prva stvar je izbira metode vizualizacije, kjer »tudi ni nujno, da že na tem mestu izberemo tudi specifičen graf ali vrsto grafa, četudi jo imamo v mislih« (Kirk 2012, 84). Mišljeno je predvsem to, da imamo včasih več možnosti izbire med vrstami grafov, pripovedovanje zgodbe pa je tisto, kar določi splošno družino oziroma zbirko teh grafov.

Z uporabo vizualizacije lahko prikažemo vrsto različnih metod, kot so primerjanje, ocenjevanje, prikazovanje sprememb skozi čas ali zgolj prikazovanje povezav (Kirk 2012, 85). Različne vrste grafov lahko uporabijo tudi več metod hkrati, bi pa vsi morali demonstrirati posamezne odgovore kot tudi skupne povzetke (Paniello in drugi 2011, 886).

Odločitev za grafični prikaz podatkov iz zapisanih izkušenj terja tudi razmislek in odločitev o uporabi grafov, s katerimi bi podatke prikazal razumljivo in smiselno.

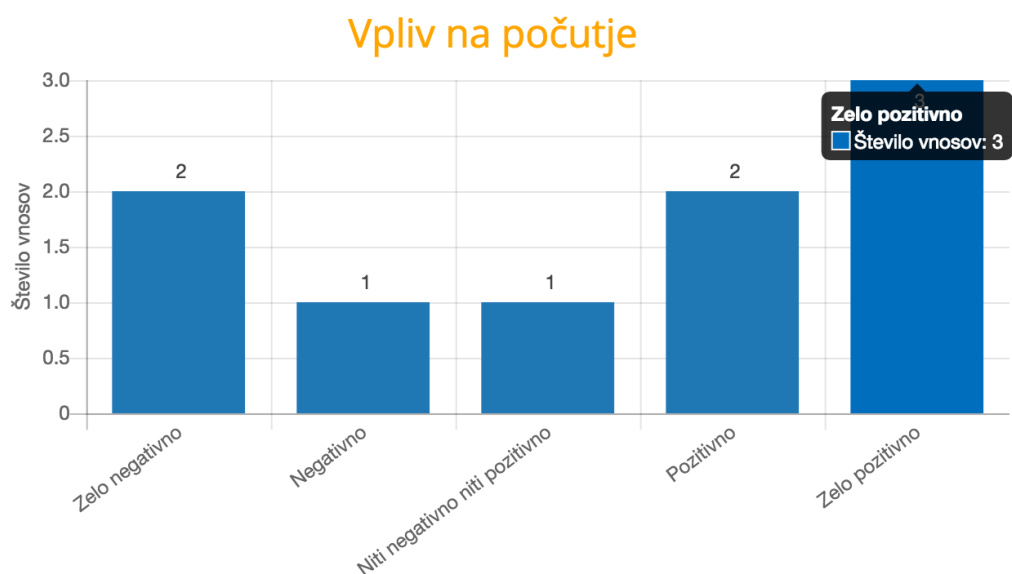
Odgovoriti sem si moral tudi na vprašanje, katero statistiko sploh želim predstaviti. Ob pregledovanju vnosnega obrazca sem tako identificiral štiri sklope podatkov, ki jih želim grafično prikazati, to pa so:

- frekvenčna porazdelitev odgovorov na vprašanje glede vpliva na počutje,
- deleže odgovorov na vprašanje glede podpore medicine,
- deleže odgovorov na vprašanje glede stroškov ter
- delitev spola in povprečje na posamezni spol.

Po identifikaciji podatkov me je predvsem zanimalo, kako jih predstaviti, da bodo uporabniki znali podatke razbrati, oziroma kateri so tisti tipi grafov, ki bi bili primerni za uporabo. Za prikaz vseh grafov pa sem uporabil Javascript knjižnico ChartJS, ki omogoča uporabo različnih vrst grafov, v katere pa je vstavev podatkov s strani strežnika enostavna in zelo prilagodljiva.

Vprašanje iz vnosnega obrazca *Kako se vam zdi, da je izkušnja vplivala na počutje bolnika* predstavlja ordinalno mersko lestvico, saj vrednosti lahko uredimo od najmanjše do največje (Ferligoj in drugi 2011, 5). Ker gre za ordinalno spremenljivko, sem frekvenčno porazdelitev prikazal s strukturnimi stolpci, saj gre za standardni prikaz tovrstnih spremenljivk. Kadar se uporablja višina oziroma širino za vizualni prikaz vrednosti, pa je pomembno, da je začetna vrednost na osi enaka nič (Kirk 2012, 123). Za lažje razumevanje grafične predstavitve, sem nad vsemi grafi umestil naslov, poleg strukturnih stolpcev pa je v grafu ob Y osi navedena oznaka vrednosti, ki jo merim, na X osi pa so navedeni odgovori danega vprašanja. Nad stolpci posameznih odgovorov je navedena tudi frekvenca odgovorov. S pomikom miške na katerikoli stolpec v graf pa se prikaže okno, kjer piše za katero vrednost gre ter kakšna je njena frekvenca.

Slika 5.10: Strukturni stolpci

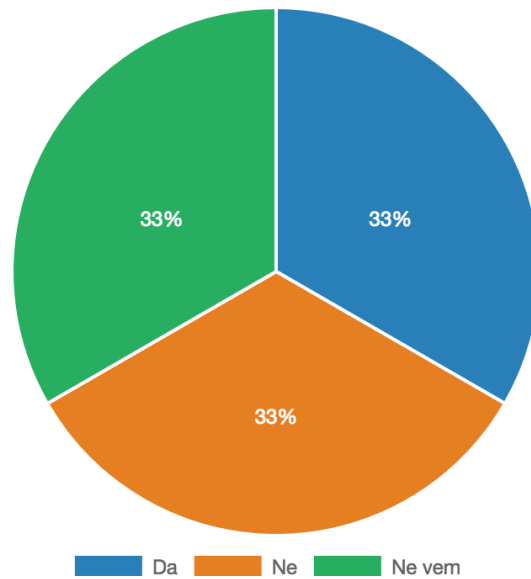


Strukturalni stolpci so dobra izbira za prikaz frekvenčne porazdelitve, vendar niso primerni za prikaz "delno v celoto" tipa, kot so na primer deleži. Za prikaz deležev se najpogosteje uporablja strukturalni krog, ki nazorno prikaže deleže posameznih vrednosti glede na celoto (Kirk 2012, 132). Strukturalni krogi v splošnem pri ekspertih niso ravno najbolj priljubljeni, saj je deleže med seboj težko primerjati, zato niso dobra izbira za primerjalne namene (de Jonge 2012, 30). V mojem primeru želim bolj primerjati deleže glede na celotno kategorijo kot pa jih med seboj primerjati, za kar pa je uporaba strukturalnega kroga priporočena.

Strukturalni krog sem uporabil za prikaz deležev odgovorov na vprašanji glede stroškov izkušnje ter podpore medicine glede izkušnje. Pod vsakim grafom pa je navedena tudi legenda, ki z barvo označuje odgovor, ki ga prezentira na grafu. Ob kliku na odgovor v legendi, lahko uporabnik interaktivno upravlja z grafom, saj tako skrrije željen delež v prikazu.

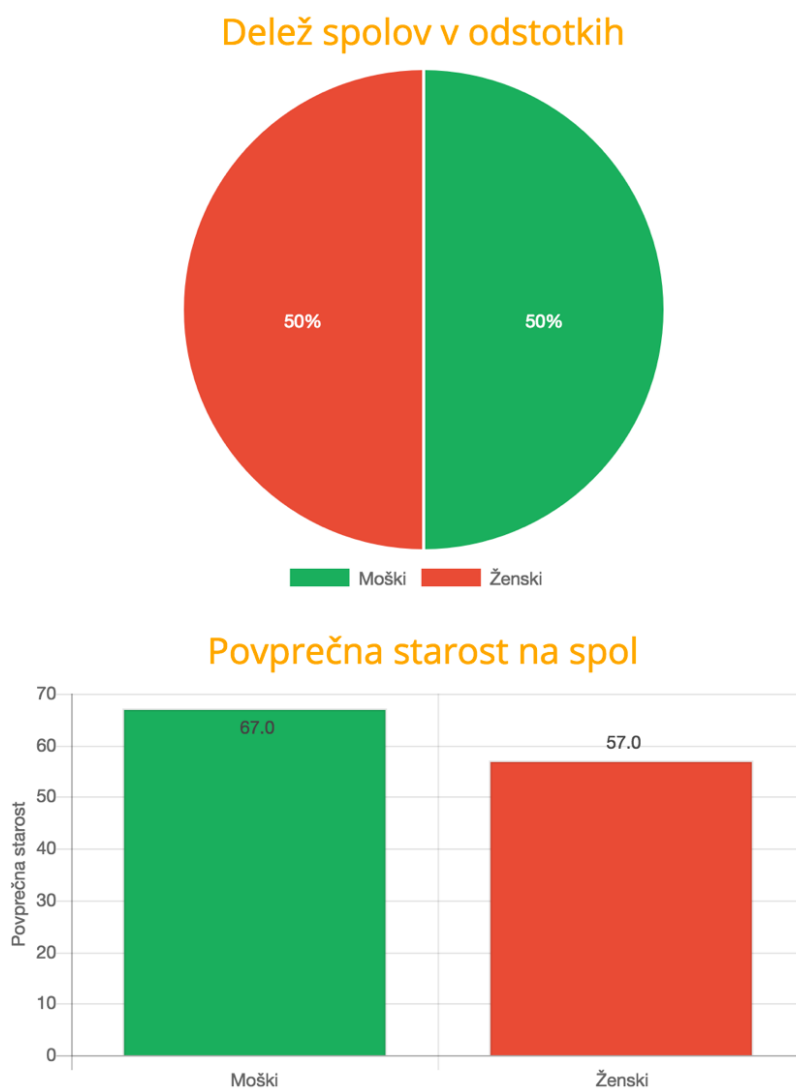
Slika 5.11: Strukturni krog

Delež podpore medicine v odstotkih



Grafični prikaz spola bi lahko ponazoril na različne načine, saj bi lahko prikazal delež spolov ali število vnosov po spolu. Vloga spola je z vidika prikaza podobna kot prikaz stroškov, zato sem deleže spolov prav tako prikazal s strukturnim krogom. Poleg strukturnega grafa, ki prikazuje deleže spolov v kategoriji, pa sem umestil tudi graf s strukturnimi stolpci, ki prikazuje povprečno starost glede na spol.

Slika 5.12: Prikaz deležev spolov in njihove povprečne starosti



S pregledom grafičnih prikazov statistike kategorij lahko tako uporabnik med seboj primerja različne kategorije. Na vsako stran kategorije pa je umeščen tudi gumb, s klikom na katerega se odpre tabela vseh vnosov, ki spadajo pod izbrano kategorijo.

6 Sklepne ugotovitve

V prvem delu diplomske naloge sem na kratko opisal bolezen ALS, in sicer sem opisal za kakšno vrsto bolezni gre ter s čim se bolniki in njihovi svojci srečujejo. V drugem delu sem opisal vso tehnologijo, ki sem jo skozi proces razvoja spletne aplikacije uporabljal, nato pa sem v tretjem delu na kratko predstavil vse njene funkcionalnosti ter podal razloge, zakaj bi lahko bila aplikacija uporabljena ter čemu bi lahko služila.

Skozi proces razvoja spletne aplikacije sem ugotovil, da je programski jezik Python, s pomočjo knjižnice Flask, odlično orodje za hiter in enostaven razvoj spletnih aplikacij, ki delujejo v povezavi z relacijskimi podatkovnimi bazami, kot je SQLite. Ta je že sama po sebi vgrajena v nastavek Pythona na računalniku, zato je uporaba in povezljivost med njima enostavna.

Celotna ideja spletne aplikacije temelji na vpisovanju izkušenj preko strukturiranega spletnega obrazca, le-ti pa so vsepovsod, zato me je tu predvsem zanimalo, kako oblikovati spletni obrazec, ki bo uporabniško prijazen in intuitiven. Skozi pregled literature sem ugotovil, da dober obrazec definira jasnost in samoumevnost. Poleg teh dveh kriterijev pa je za dober obrazec tudi pomembno, da ni predolg, če to od njega ni zahtevano. Zelo pomembno vprašanje uporabniške prijaznosti obrazcev je postavitve vprašanj, saj le-ta vpliva na potek branja in izpolnjevanja. Vprašanja je najbolj priporočljivo postaviti levo od vnosnih polj ali nad njimi, za kar sem se tudi odločil, saj je tako potek izpolnjevanja bolj naraven in tudi hitrejši. K enostavnosti izpolnjevanja obrazca pripomoreta tudi označitev obveznih polj ter vsebinska ločitev.

Zanimalo me je tudi, kako lahko prikažem osnovne statistike kategorij vnosov tako, da bodo primerne in uporabne. Skozi pregled literature sem ugotovil, da je najbolj primerna predstavitev z uporabo vizualizacije. Prikazovati želim enostavne frekvenčne porazdelitve, zato sem se za absolutne frekvence poslužil strukturnih stolpcev, za prikaz relativnih frekvenc pa strukturnih krogov, s katerimi se dobro prikaže deleže glede na celoto. Grafi so tudi ustrezno dopolnjeni z naslovi, vrednostmi ter poimenovanji osi.

Spletno aplikacijo bi lahko v nadaljevanju še dopolnil in izboljšal, saj bi lahko dodal več funkcij, kot je na primer komentiranje izkušenj.

7 Literatura

1. Allen, Grant in Mike Owens. 2010. *The Definitive Guide to SQLite*. New York: Apress.
2. ALS Association. 2017. *About ALS*. Dostopno prek: <http://www.alsa.org/about-als> (11. julij 2017).
3. *Bootstrap*. Dostopno prek: <http://getbootstrap.com> (25. julij 2017).
4. Carter, Gregory T., Lisa M. Bednar-Butler, R. Ted Abresch in Viviane O. Ugalde. 1999. Expanding the Role of Hospice Care in Amyotrophic Lateral Sclerosis. *American Journal of Hospice and Palliative Medicine* 16 (6): 707–710.
5. de Jonge, Edwin. 2012. *Data Visualisation*. Hague: Statistics Netherlands.
6. Duckett, Jon. 2011. *HTML and CSS: Design and Build Websites*. Indianapolis: John Wiley & Sons.
7. Ferligoj, Anuška, Katja Lozar Manfreda in Aleš Žiberna. *Osnove statistike na prosojnicah : študijsko gradivo pri predmetu Statistika*. Ljubljana: Fakulteta za družbene vede.
8. Frain, Ben. 2012. *Responsive Web Design with HTML5 and CSS3*. Birmingham: Packt Publishing Ltd.
9. Inštitut za klinično nevrofiziologijo. 2016. *Amiotrofična lateralna skleroza (ALS)*. Dostopno prek: <http://www2.kclj.si/ikn/INFO/ALS/ALS.htm> (10. julij 2017).
10. Jarrett, Caroline, Gerry Gaffney in Steve Krug. 2008. *Forms That Work: Designing Web Forms for Usability*. Amsterdam: Morgan Kaufmann.
11. Kirbiš, Mojca, Blaž Koritnik, Lea Leonardis, Leja Dolenc Grošenj, Polona Klinar, Stanka Ristić Kovačič in Janez Zidar. 2015. Amyotrophic Lateral Sclerosis in Slovenia – Analysis of a Patient Cohort at the Ljubljana Institute of Clinical Neurophysiology. *Zdravniški Vestnik* 84 (7): 528–35.
12. Kirk, Andy. 2012. *Data Visualization: A Successful Design Process*. Birmingham: Packt Publishing.
13. Mozilla Developer Network. 2017. *JavaScript basics*. Dostopno prek https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics (5. avgust 2017).
14. Murray, Scott. 2013. *Interactive Data Visualization for the Web: An Introduction to Designing with D3*. Sebastopol: O'Reilly Media.

15. Paniello, Randal C., J. Gail Neely, Jason T. Rich, Eric L. Slattery in Courtney C. J. Voelker. 2011. Practical Guide to Choosing an Appropriate Data Display. *Otolaryngology-Head and Neck Surgery* 145 (6): 886–94.
16. Python. Dostopno prek: <https://www.python.org> (8. avgust 2017).
17. Python. 2017. *Virtual Environments and Packages*. Dostopno prek: <https://docs.python.org/3/tutorial/venv.html> (8. avgust 2017).
18. Swaroop, C. H. 2013. *A Byte of Python*. ebsshelf Inc. Dostopno prek: <https://python.swaroopch.com/> (26. avgust 2017).
19. W3Schools. 2017a. *CSS Introduction*. Dostopno prek: https://www.w3schools.com/css/css_intro.asp (24. julij 2017).
20. --- 2017b. *HTML Forms*. Dostopno prek: https://www.w3schools.com/html/html_forms.asp (4. avgust 2017).
21. --- 2017c. *HTML Introduction*. Dostopno prek: https://www.w3schools.com/html/html_intro.asp (18. julij 2017).
22. Wroblewski, Luke. 2008. *Web Form Design: Filling in the Blanks*. Brooklyn: Rosenfeld Media.

Priloga A: Povezava za dostop do repozitorija kode aplikacije

<https://github.com/frasza/diploma>