Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektonsko poslovanje

https://elab.fon.bg.ac.rs/wp-content/uploads/2018/05/elab-logo.png

Seminarski rad iz predmeta Internet tehnologije  
  
Tema:**E-apoteka kao primena aplikacije e-zdravstva**

Mentor: Studenti:  
Dušan Barać Milica Milinković 426/15

Katarina Jovičić 74/15

Beograd, februar 2020.

Sadržaj

[Korisnički zahtev 1](#_Toc32149476)

[Slučajevi korišćenja 2](#_Toc32149477)

[SK1 – Registracija kupca 3](#_Toc32149478)

[SK2 – Prijava na sistem 4](#_Toc32149479)

[SK3 – Kupovina leka 5](#_Toc32149480)

[SK4 – Pretraga lekova 6](#_Toc32149481)

[SK5 – Dodavanje novog leka 7](#_Toc32149482)

[SK6 – Odobravanje zahteva za kupovinu 8](#_Toc32149483)

[SK7 – Brisanje zahteva za kupovinu 9](#_Toc32149484)

[Dijagrami sekvenci za slučajeve korišćenja 10](#_Toc32149485)

[DS1 – Registracija kupca 10](#_Toc32149486)

[DS2 – Prijava na sistem 10](#_Toc32149487)

[DS3 – Kupovina leka 11](#_Toc32149488)

[DS4 – Pretraga lekova 11](#_Toc32149489)

[DS5 – Dodavanje novog leka 12](#_Toc32149490)

[Model podataka 13](#_Toc32149491)

[Konceptualni dijagram klasa 13](#_Toc32149492)

[Struktura baze 13](#_Toc32149493)

[Arhitektura aplikacije 14](#_Toc32149494)

[REST API dokumentacija 15](#_Toc32149495)

[Opis korišćenih tehnologija 16](#_Toc32149496)

[HTML 16](#_Toc32149497)

[CSS 16](#_Toc32149498)

[JavaScript 17](#_Toc32149499)

[PHP 18](#_Toc32149500)

[AJAX 19](#_Toc32149501)

[SQL 20](#_Toc32149502)

[Korisničko uputstvo – E-Apoteka 21](#_Toc32149503)

[Literatura 26](#_Toc32149504)

# Korisnički zahtev

Potrebno je projektovati i implementirati veb aplikaciju apoteke **E-Apoteka***,* koja će korisnicima omogućiti pregled i mogućnost online kupovine svih proizvoda u asortimanu iste. Korisnici takođe mogu pristupiti aktuelnim vestima iz oblasti zdravlja i informisati se o kratkom istorijatu apoteke E-Apoteka.

Platformi mogu pristupati 3 različite role: **administrator**, **kupac** i **posetilac**.

U sklopu veb aplikacije, **kupci** će moći da:

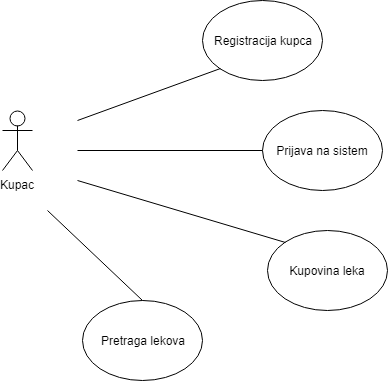
1. Imaju pregled svih dostupnih lekova u E-Apoteci.
2. Prave svoj lični nalog putem forme za registraciju.
3. Uloguju se na aplikaciju ukoliko su prethodno registrovani.
4. Vrše kupovinu lekova koje žele da poseduju.
5. Čitaju aktuelne vesti iz oblasti zdravlja.
6. Vrše pretragu lekova po svim relevantnim kriterijumima.

**Administrator** ima mogućnost da:

1. Pregleda sve dostupne lekove u E-Apoteci.
2. Uloguje se na svoj lični nalog.
3. Čita aktuelne vesti iz oblasti zdravlja.
4. Dodaje nove lekove sa svim relevantnim informacijama.
5. Odobri pristigle zahteve za kupovinu.
6. Obriše pristigle zahteve za kupovinu.
7. Pregleda statistiku cene lekova.

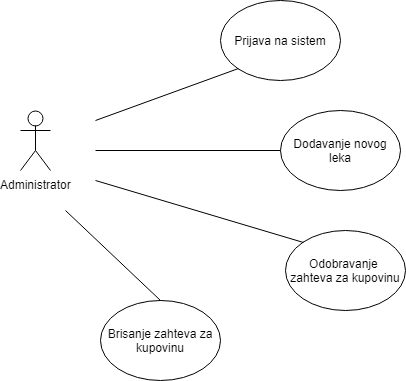
**Posetilac** će moći da vidi početnu stranu veb aplikacije, čita interaktivne vesti i pristupi informacijama o samoj E-Apoteci. Pored toga, posetilac ima uvid u sve lekove koje E-Apoteka nudi i može se registrovati po želji.

# Slučajevi korišćenja



Slučajevi korišćenja za **kupca**:

1. SK1 – Registracija kupca
2. SK2 – Prijava na sistem
3. SK3 – Kupovina leka
4. SK4 – Pretraga lekova



Slučajevi korišćenja za **administratora**:

1. SK5 – Prijava na sistem
2. SK6 – Dodavanje novog leka
3. SK7 – Odobravanje zahteva za kupovinu
4. SK8 – Brisanje zahteva za kupovinu

## SK1 – Registracija kupca

**Naziv SK**

Registracijakupca

**Aktori SK**

Kupac

**Učesnici SK**

Kupac i sistem (program)

*Preduslov*: Sistem je uključen i prikazuje formu za registrovanje kupca.

**Osnovni scenario SK**

1. Kupac se registruje sa potrebnim podacima. (APUSO)

2. Kupac proverava da li je uneo ispravne podatke. (ANSO)

3. Kupac poziva sistem da zapamti podatke. (APSO)

4. Sistem pamti podatke. (SO)

5. Sistem prikazuje kupcu poruku: „Sistem je uspešno zapamtio Vaše podatke“. (IA)

**Alternativna scenarija**

* 1. Ukoliko sistem ne može da kreira korisnika prikazuje poruku “Sistem ne može da kreira korisnika na osnovu unetih vrednosti”. (IA)

## SK2 – Prijava na sistem

**Naziv SK**

Prijavljivanje korisnika

**Aktori SK**

Korisnik (kupac i administrator)

**Učesnici SK**

Korisnik i sistem (program)

*Preduslov*: Sistem je uključen i prikazuje formu za prijavljivanje na sistem.

**Osnovni scenario SK**

1. Korisnik unosi podatke kojima se prijavljuje na sistem. (APUSO)

2. Korisnik poziva sistem da pronađe korisnika sa zadatim podacima.(APSO)

3. Sistem pretražuje korisnike. (SO)

4. Sistem prikazuje poruku „Uspešno ste se prijavili na sistem“. (IA)

**Alternativna scenarija**

* 1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Sistem ne može da nađe korisnika na osnovu unetih vrednosti”. (IA)

## SK3 – Kupovina leka

**Naziv SK**

Kupovina leka

**Aktori SK**

Kupac

**Učesnici SK**

Kupac i sistem (program)

*Preduslov*: Sistem je uključen i prikazuje formu za kupovinu lekova. (Učitana je lista lekova)

**Osnovni scenario SK**

1. Kupac bira lek koji želi da kupi. (APUSO)

2. Kupac proverava da li je ispravno izabrao željeni termin . (ANSO)

3. Kupac poziva sistem da zapamti podatke o porudžbini. (APSO)

4. Sistem pamti podatke o porudžbini. (SO)

5. Sistem prikazuje kupcu poruku: „Sistem je uspešno zapamtio podatke o porudžbini“. (IA)

**Alternativna scenarija**

* 1. Ukoliko sistem ne može da zapamti podatke o porudžbini prikazuje poruku “Sistem ne može da zapamti željenu kupovinu”. (IA)

## SK4 – Pretraga lekova

**Naziv SK*:*** Pretraga lekova

**Aktori SK*:***Kupac

**Učesnici SK*:***Kupac i sistem

*Preduslov****:*** Sistem je pokrenut. Kupac je ulogovan pod svojom šifrom. Sistem prikazuje formu za rad sa lekovima. Učitana je lista lekova.

**Osnovni scenario*:***

1. Kupac unosi vrednost po kojoj pretražuje lekove. (APUSO)

2. Kupac poziva sistem da nađe lekove po zadatoj vrednosti. (APSO)

3. Sistem traži lekove po zadatoj vrednosti. (SO)

4. Sistem prikazujekupcu podatke o lekovima I poruku “Sistem je našao lekove po zadatoj vrednosti”. (IA)

***Alternativni scenario:***

4.1 Ukoliko sistem ne može da nađe lekove, on prikazuje kupcu poruku “Sistem ne može da nađe lekove po zadatoj vrednosti”. (IA)

## SK5 – Dodavanje novog leka

**Naziv SK*:***Dodavanje novogleka

**Aktori SK*:***Administrator

**Učesnici SK*:***Administrator i sistem

*Preduslov****:*** Sistem je pokrenut. Administrator je ulogovan pod svojom šifrom. Sistem prikazuje formu za rad sa lekovima. Učitana je lista kategorije lekova.

**Osnovni scenario*:***

1. Administrator poziva sistem da kreira nov lek. (APSO)
2. Sistem kreira novi lek. (SO)
3. Sistem prikazuje administratoru novi lek i poruku “Sistem je kreirao novi lek.”(IA)
4. Administrator unosi podatke o novom leku. (APUSO)
5. Administrator kontroliše da li je korektno uneo podatke o novom leku. (ANSO)
6. Administrator poziva sistem da zapamti podatke o novom leku. (APSO)
7. Sistem prikazuje administratoru poruku “Zapamćen je novi lek”. (IA)

***Alternativni scenario:***

3.1. Ukoliko sistem ne može da kreira novi lek, on prikazuje administratoru poruku “Sistem ne može da kreira novi lek.”. Prekida se izvršenje scenaria. (IA)

7.1. Ukoliko sistem ne može da zapamti podatke o novom leku, on prikazuje administratoru poruku “Sistem ne može da zapamti novi lek.”. (IA)

## SK6 – Odobravanje zahteva za kupovinu

**Naziv SK*:***Odobravanje zahteva za kupovinu

**Aktori SK*:***Administrator

**Učesnici SK*:***Administrator i sistem

*Preduslov****:*** Sistem je pokrenut. Administrator je ulogovan pod svojom šifrom. Sistem prikazuje formu za rad sazahtevima. Učitana je listasvih zahteva za kupovinom.

**Osnovni scenario*:***

1. Administratorbira zahtev koji želi da odobri. (APUSO)
2. Administrator poziva sistem da učita podatke o zahtevu. (APSO)
3. Sistem učitava podatke o odabranom zahtevu. (SO)
4. Sistem obaveštava administratora o uspešnom učitavanju podataka o zahtevu porukom “Podaci o odabranom zahtevu su prikazani.” I prikazuje podatke o odabranom zahtevu. (IA)
5. Administrator poziva sistem da odobri zahtev za kupovinu. (APSO)
6. Sistem odobrava zahtev. (SO)
7. Sistem prikazuje administratoru poruku “Sistem je odobrio zahtev.”. (IA)

***Alternativni scenario:***

4.1. Ukoliko sistem ne može da nađe zahtev, on prikazuje administratoru poruku “Sistem ne može da nađe zahtev po zadatoj vrednosti.”. Prekida se izvršenje scenaria. (IA)

7.1. Ukoliko sistem ne može da odobri zahtev za kupovinom, on prikazuje administratoru poruku “Sistem ne može da odobri zahtev za kupovinom.”. (IA)

## SK7 – Brisanje zahteva za kupovinu

**Naziv SK*:***Odobravanje zahteva za kupovinu

**Aktori SK*:***Administrator

**Učesnici SK*:***Administrator i sistem

*Preduslov****:*** Sistem je pokrenut. Administrator je ulogovan pod svojom šifrom. Sistem prikazuje formu za rad sa zahtevima. Učitana je lista svih zahteva za kupovinom.

**Osnovni scenario*:***

1. Administratorbira zahtev koji želi da obriše. (APUSO)
2. Administrator poziva sistem da učita podatke o zahtevu. (APSO)
3. Sistem učitava podatke o odabranom zahtevu. (SO)
4. Sistem obaveštava administratora o uspešnom učitavanju podataka o zahtevu porukom “Podaci o odabranom zahtevu su prikazani.” i prikazuje podatke o odabranom zahtevu. (IA)
5. Administrator poziva sistem da obriše zahtev za kupovinu. (APSO)
6. Sistem briše zahtev. (SO)
7. Sistem prikazuje administratoru poruku “Sistem je obrisao zahtev.”. (IA)

***Alternativni scenario:***

4.1. Ukoliko sistem ne može da nađe zahtev, on prikazuje administratoru poruku “Sistem ne može da nađe zahtev po zadatoj vrednosti.”. Prekida se izvršenje scenaria. (IA)

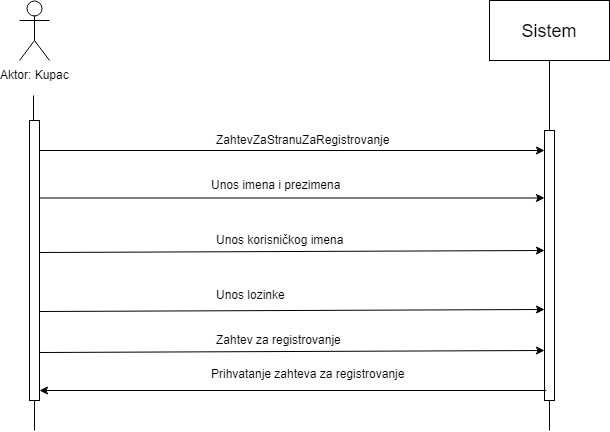
7.1. Ukoliko sistem ne može da obriše zahtev za kupovinu, on prikazuje administratoru poruku “Sistem ne može da obriše zahtev za kupovinu.”. (IA)

# Dijagrami sekvenci za slučajeve korišćenja

## DS1 – Registracija kupca

**Osnovni scenario:**

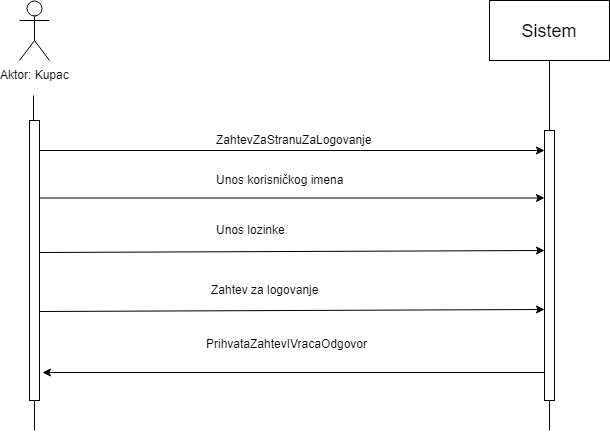
1. Korisnik navigira do stranice za registrovanje
2. U odgovarajuća polja unosi ime, prezime, korisničko ime i lozinku i potvrđuje unos pritiskom na dugme
3. Sistem prihvata prijavu korisnika i dozvoljava mu pristup stranama



## DS2 – Prijava na sistem

**Osnovni scenario**:

1. Korisnik navigira do stranice za logovanje
2. U odgovarajuća polja unosi korisničko ime i lozinku i potvrđuje unos pritiskom na dugme
3. Sistem prihvata prijavu korisnika i dozvoljava mu pristup stranama

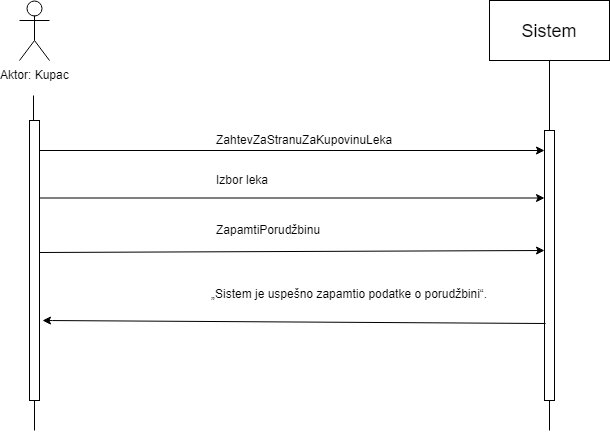


## DS3 – Kupovina leka

**Osnovni scenario:**

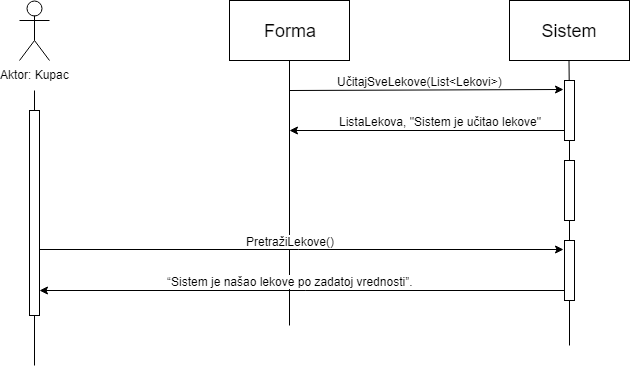
1. Korisnik navigira do stranice za kupovinu leka
2. Iz liste lekova korisnik bira lek koji želi da kupi
3. Sistem prihvata prijavu korisnika i dozvoljava mu da obavi kupovinu

## 

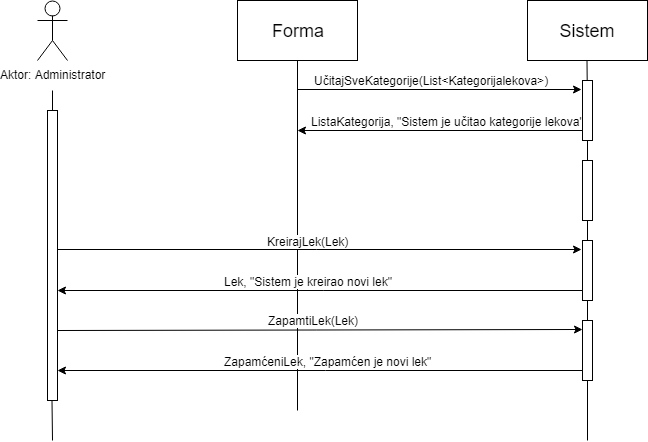


## DS4 – Pretraga lekova

## 

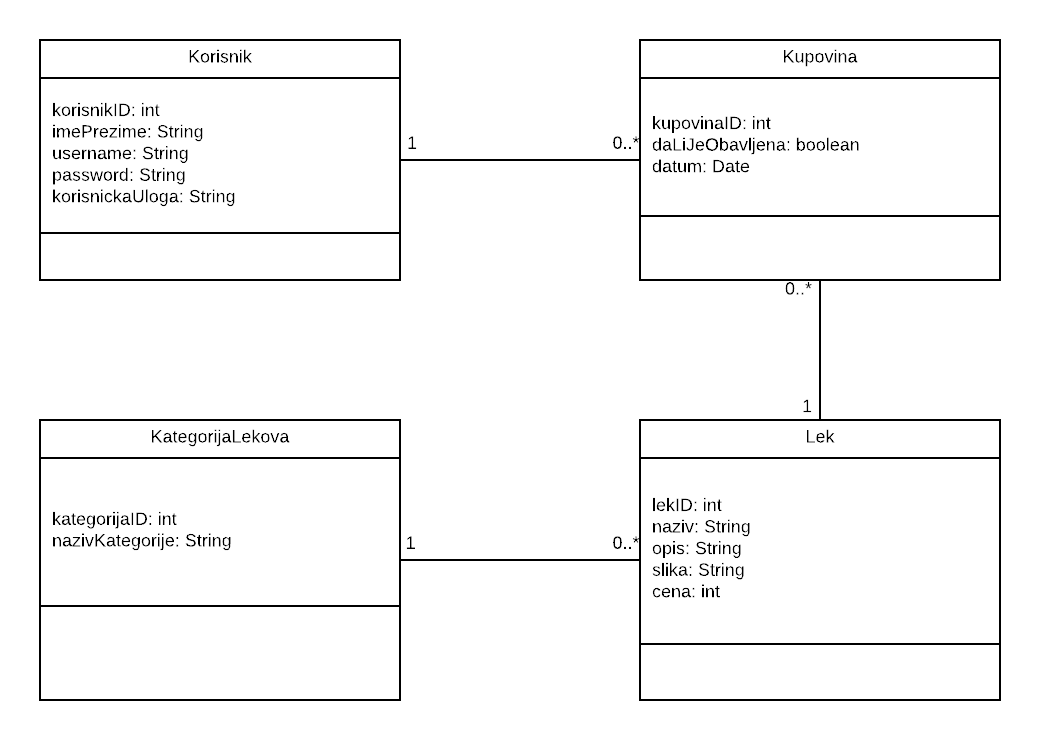


## DS5 – Dodavanje novog leka

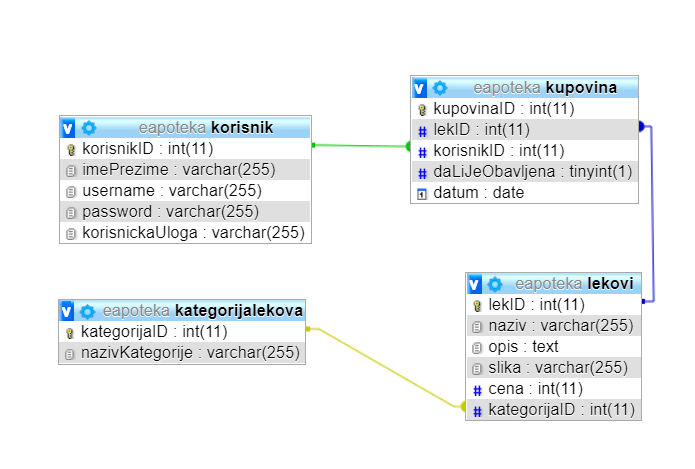


# Model podataka

## Konceptualni dijagram klasa



## Struktura baze

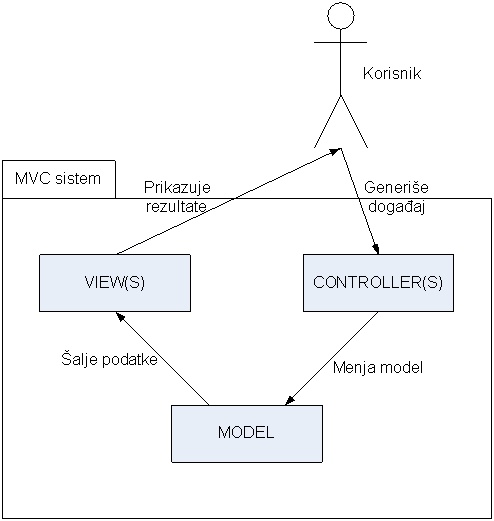


# Arhitektura aplikacije

**Model** – Predstavlja strukture podataka. Klase u model sloju uglavnom služe za čitanje, dodavanje i ažuriranje podataka u bazi podataka.

**View** – Informacije prezentovane korisniku. View je obična web strana, ali u CodeIgniter-u to može i biti deo strane,na primer zaglavlje.

**Controller** – Služi kao posrednik između model-a, view-a  i ostalih resursa potrebnih za procesiranje HTTP zahteva.



# REST API dokumentacija

Veb servis je napravljen poštujući REST arhitekturu i sastoji se od sledecih funkcija:

* **GET lekoviKupovina.json**
* **GET kategorije.json**
* **GET lekovi/@lekID.json**
* **GET lekovi/@naziv.json**
* **GET lekovi/@cena.json**
* **POST ubaciLek.json**
* **POST kupiLek.json**
* **POST ubaciKorisnika.json**
* **PUT kupovina/@daLiJeObavljena**
* **DELETE kupovina/@kupovinaID**

Sve funkcije su smeštene u rest/index.php i napravljene su sledeće rute:

* **Flight::route('GET /kategorije.json', function()**
* **Flight::route('GET /lekoviKupovina.json', function()**
* **Flight::route('POST /ubaciLek.json', function()**
* **Flight::route('POST /ubaciKorisnika.json', function()**
* **Flight::route('POST /kupiLek.json', function()**
* **Flight::route('GET /lekovi/@lekID.json', function($lekID)**
* **Flight::route('GET /lekovi/@naziv.json', function($naziv)**
* **Flight::route('GET /lekovi/@cena.json', function($cena)**
* **Flight::route('PUT /kupovina/@daLiJeObavljena, function($daLiJeObavljena)**
* **Flight::route('DELETE /kupovina/@kupovinaID', function($kupovinaID) {**

# Opis korišćenih tehnologija

## HTML

HTML, HyperText Markup Language, je standarizovani jezik koji se koristi pri strukturiranju tekstova, medija i ugrađenih objekata u web stranice i elektronsku poštu. Kao modifikovanu i pojednostavljenu verziju SGML jezika, HTML standarizuje i održava World Wide Web Consortium (W3C). Originalnu verziju HTML-a kreirao je Tim Berners-Lee, a prva zvanična verzija je izašla juna 1993. godine.

HTML jezik je sačinjen od običnog teksta i tagova. Iako se HTML znakovi (tags) često nazivaju kodom, tehnički HTML nije kod jer kompjuterski kod su instrukcije koje od računara traže da izvrši određenu operaciju. Svrha HTML znakova je da se se struktura dokumenta "označi" tako da bi korisnikov agent (user agent), tj. internet pretraživač mogao da prepozna strukturu dokumenta i ispravno je prikaže u prozoru internet pretraževača kojeg korisnik koristi.

Za dodatno modifikovanje web stranica, do sada su izašli neki dodaci:

* CSS ili Cascading Style Sheets za izgled i položaj prezentacije
* Skriptni jezici (Javascript, VBScript) za omogućavanje dinamičnosti i interaktivnosti na web prezentacijama,
* DOM ili Document Object Model, koji označava vezu između skripte i elementa na stranici
* Gornji dodaci, u paketu sa HTML jezikom, ponekad se nazivaju DHTML ili Dinamični HTML.

## CSS

CSS je u osnovi skup instrukcija koje definišu kako neki HTML dokument treba da se prikaže. Najbolja stvar kod CSS-a je da se on može definisati u spoljašnjoj datoteci, ili listu (sheet). Pretraživač onda može da pročita ovaj spoljašnji list i da primeni sadržaj na određeni sadržaj datoteke. Na taj način se stil i formatiranje strane odvajaju od samog sadržaja. Ovo je vrlo bitan koncept kod dizajna web strana. Ovo Vam pruža moćan i detaljan način da ažurirate svoj web sajt. U datoteci sa stilovima možete definisati da se za tekst primenjuje font Arial, da je veličina slova 12 piksela, da su slova crvene boje, da su masna, da je tekst levo poravnat i da je pozadina plava. Ako ovu datoteku pridružite svim stranama na svom sajtu, tekst će se uvek prikazivati na taj način. Ako promenite mišljenje, možete da promenite jednu datoteku i da zadate da font bude Times, da veličina slova bude 28 piksela, da su zelene boje, pisana kurzovim i desno poravnata. Upamtite tu novu datoteku i ceo web sajt se automatski ažurira. CSS takođe omogućava da radite neke stvari koje u HTML-u nisu moguće. Za svoju stranu možete definisati nivo kroz koji ćete se kretati klizačem. Na taj način možete da imate prozor u sredini strane sa sadržajem kroz koji se možete kretati uz pomoć klizača. Za svaku ivicu u okviru tabele možete definisati različitu boju. Možete podesiti da se kod oznaka za nabrajanje koristi korisnički definisana grafika, ili možete obrisati podvlačenje hiperlinkova. Premda se sa svakom novom verzijom pretraživača poboljšava podrška koju oni imaju za CSS, ipak neće svi pretraživači prikazati sve elemente CSS-a, niti će sve elemente prikazati na isti način. Uvek je dobro da svoju stranu testirate u različitim pretraživačima. Jedna od bitnih ideja u vezi CSS-a je da ako pretraživač ne podržava neki stil, informacije su i dalje tu, ali se ne koriste stilovi koji nedostaju. IE 5 i Netscape 6 i novije verzije uglavnom prikazuju većinu CSS elemenata. Izvesna ograničenja postoje u verziji Netscape 4. CSS Vam daje neverovatnu kontrolu nad stranama. W3C sada preporučuje da za formatiranje strana koristite CSS. Trebalo bi da pažljivo razmotrite mogućnosti primene CSS-a kod dizajna.

## JavaScript

Java Script pripada grupi jezika za skriptovanje, pre svega klijentske strane, mada se može izvršavati i na serveru (runat=“server“). Java Script je najpopularniji jezik na Internetu, koji je dizajniran da poveća interaktivnost HTML strana.

Razvila ga je kompanija Netscape Communications i uvela u Netscape Navigator, počevši od verzije 2.0, uporedo sa uvođenjem podrške za Javu. JavaScript je uveden radi povećanja interaktivnosti veb strana. Veb čitači koji podržavaju JavaScript interpretiraju kod.

JavaScript dodaje interaktivnost veb prezentaciji na strani klijenta. Na primer, korišćenjem JavaScripta je moguće odgovarati na akcije korisnika u samom veb čitaču. Bitno je naglasiti da JavaScript program može da se izvrašava samo u okviru veb čitača i nigde drugde. Pri kreiranju JavaScript koda može se koristiti bilo koji ASCII editor.

Polazni naziv JavaScript jezika bilo je LiveScript, ali u saradnji Netscape-a i Sun-a, ime je izmenjeno u JavaScript, zbog namere da Java i JavaScript budu jezici Interneta i internet programiranja.

JavaScript programi se koriste da bi detektovali i reagovali na događaje koji su inicirani od strane korisnika, kao npr. prelazak mišem preko linka i grafičkog objekta na strani. JavaScript kod može da poboljša kvalitet sajta navigacionim pomagalima, skrolovanim porukama, dialog prozorima, dinamičnim slikama, koliicima za kupovinu (shopping cart), itd. JavaScript omogućava korisnicima da kontrolišu izled veb strane dok se dokument „tumači“ od strane čitača. Uz pomoć JavaScripta, možemo da proverimo tačnost podataka koje je korisnik ukucao u formu, pre nego što se ti podaci prebace na server. JavaScript testira brauzaer korisnika da vidi da li ima neophodne dodatke (plug inns) za učitavanje tražene strane i ako ih nema, JavaScript usmerava korisnika na drugi sajt da ih nabavi. JavaScript služi kao programski jezik. Njegovo jezgro sadrži osnovne elemente programskih jezika kao što su promenljive, tipovi podataka, kontrolne petlje, if/else izjave, switch izjave, funkcije i objekte. JavaScript se koristi za aritmetičke operacije, manipulacije datumom i vremenom, nizovima, stringovima i objektima. On takođe čita i upisuje vrednosti kuki fajlova i dinamički kreira HTML na osnovu vrednosti kukija.

## PHP

Server - side skripting je web server tehnologija koja omogućava da se korisnički zahtevi obrađuju pomoću skripti koje se izvršavaju na serverskoj strani kako bi se generisale dinamičke stranice. Najčešće se koristi da bi se interaktivne web stranice povezale sa bazama podataka, radi identifikacije korisnika, ažuriranja sadržaja, stvaranja raznih diskusionih grupa i još zbog mnogo toga. Razlika između skriptovanja na serverskoj i korisničkoj strani je u tome što se kod korisničke strane skripte izvršavaju u web browser-u korisnika, najčešće pomoću JavaScript-a.

Najzastupljenije tehnike serverske strane:

* Common Gateway Interface
* Active Server Pages
* Java Servlet i Java Server Pages
* PHP

PHP je open source jezik koji se koristi za razvoj server-side aplikacija, kao i dinamičkog Web sadržaja. PHP dozvoljava interakciju sa velikim brojem relacionih baza podataka kao što su MySQL, Oracle, IBM D2, Microsoft SQL Server, PostgreSQL i SQLite. PHP radi na većini operativnih sistema današnjice, kao što su UNIX, Linux, Windows i Mac OS i može da interaguje sa većinom Web servera.

Personal Home Page Tools je predstavio PHP 1995. godine, a dve godine kasnije su razvoj nastavila dva Izraelska programera. Od 1999. godine se zasniva na Zend engine-u. PHP je prvo bio akronim za Personal Home Page Tools, da bi kasnije promenio naziv u Hypertext Preprocessor.

PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C. Tokom vremena jezik se proširivao i sticao mogućnosti za objektno orijentisano programiranje, naročito od verzije 5.0. Nalikuje jeziku C++ u smislu da dozvoljava i čisto-proceduralno programiranje ali omogućava i korišćenje klasa i drugih koncepata objektno orijentisanog programiranja (nasleđivanje, apstraktne metode, interfejsi itd.).

Danas, PHP je instaliran na više od 20 miliona sajtova i preko million web servera. Poslednja realizovana stabilna verzija je 5.2.6 iz maja 2008.god.

Neke od osnovnih odrednica PHP – a su:

* PHP je akronim za Hypertext Preprocessor
* server – side skriptni programski jezik
* skripte se izvršavaju na serveru
* podržava rad sa različitim bazama podatama (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, itd.)
* open source softver (OSS)
* radi na različitim platformama (Windows, Linux, Unix i td.)
* kompatibilan je sa skoro svim serverima koji se danas koriste (Apache, IIS, itd. )
* lak je za učenje i efikasno funkcioniše na strani servera.

***Za PHP fajlove, bitne su sledeće činjenice:***

* mogu da sadrže tekst, HTML tagove i skripte
* vraćaju se klijentskom browser – u kao HTML tekst (plain HTML)
* imaju ekstenziju: “.php”, “.php3” ili “.phtml”

## AJAX

AJAX predstavlja web razvojnu tehniku za kreiranje interaktivnih web sadržaja. AJAX tehnologija se zasniva na razmeni malih količina podataka sa serverom, kako korisnik ne bi stalno morao da osvežava stranicu. Na taj način se omogućava potpuna interaktivnost, efikasnost i poboljšava funkcionisanje stranice uopšte. AJAX nije tehnologija sam po sebi, već termin koji se odnosi na korišćenje grupe tehnologija.

AJAX obuhvata:

* prezentaciju baziranu na standardima koristeći XHTML i CSS;
* dinamički prikaz i interakciju preko Document Object Model-a;
* razmenu i manipulaciju nad podacima koristeći XML i XSLT;
* asinhrono prikupljanje podataka uz pomoć XMLHttpRequest-a koji čini jezgro;
* i JavaScript koji povezuje sve u celinu.

Ideja koja se krije iza AJAX-a jeste da se stranica na kojoj se odvija web aplikacija učita samo jednom, a da se svaka dalja komunikacija sa serverom sakrije od očiju korisnika i obavlja bez ponovnog učitavanja čitave stranice. Svaki prenos podataka između servera i klijenta (u slučaju AJAX-a to je browser) vrši se u pozadini. Jasno je da je ovo nemoguće izvesti statičkim HTML-om, pa tu na scenu stupa JavaScript. JavaScript je zadužen za komuniciranje sa serverom – slanje HTTP zahteva, prijem podataka sa servera i njihov prikaz na stranici, a i interakcija sa korisnikom postiže se korišćenjem ovog jezika.

Neki od osnovnih principa na kojima bi trebalo da se zasnivaju AJAX aplikacije:

* Minimalan protok - AJAX aplikacije bi trebalo da šalju i primaju sa servera što je moguće manje informacija
* AJAX aplikacije predstavljaju korisniku drugačije modele interakcije od tradicionalnih web aplikacija. Nasuprot “klik – čekaj“ prirodi standardnih web aplikacija, AJAX predstavlja modele interakcije koji su slični desktop aplikacijama. Bez obzira koji se model interakcije koristi, najbitnije je obezbediti konzistentnost kako bi korisnik znao šta treba sledeće da uradi.
* Izbegavaju se nepotrebni elementi na strani kao što su animacije ili delovi koji trepere.
* Izbegava se preuzimanje cele strane - celokupna komunikacija sa serverom, nakon učitavanja inicijalne web strane treba da bude zadatak AJAX engine –a

AJAX omogućava različite funkcionalnosti:

* Validacija podatka u realnom vremenu - Podaci na formama kao što su: user IDs, serial numbers, postal codes, i ostali koji zahtevaju validaciju od strane servera, mogu biti provereni pre submit- ovanja forme.
* Autocompletion i autosuggest - Određeni delovi podataka koji se unose od strane korisnika(posetioca): e-mail adresa, ime, grad i ostali se mogu automatski kompletirati, tj. dovršiti odmah nakon što se unesu samo početni delovi.
* Učitavanje na zahtev - U skladu sa iniciranim događajem od strane korisnika, HTML strana može povući veću količinu podataka u pozadini i omogućiti učitavanje kompletne strane mnogo brže.
* Sofisticiran korisnički interfejs, kontrole i efekti – Kontrole kao što su meniji, tabele podatka, tekst editori, kalendari, status bar-ovi i sl. Omogućavaju korisnicima bolju interakciju, bez potrebe za učitavanjem kompletne stranice.
* Osvežavanje podataka - HTML stranice povlače podatke sa servera i omogućavaju najsvežije podatke kao što su rezultati, cene akcija, vreme ili neke druge podatke iz specifičnih aplikacija.
* Delimičan submit – bez potrebe za učitavanjem cele forme.
* Stranica kao aplikacija – web stranice postaju slične desktop aplikacijama

## SQL

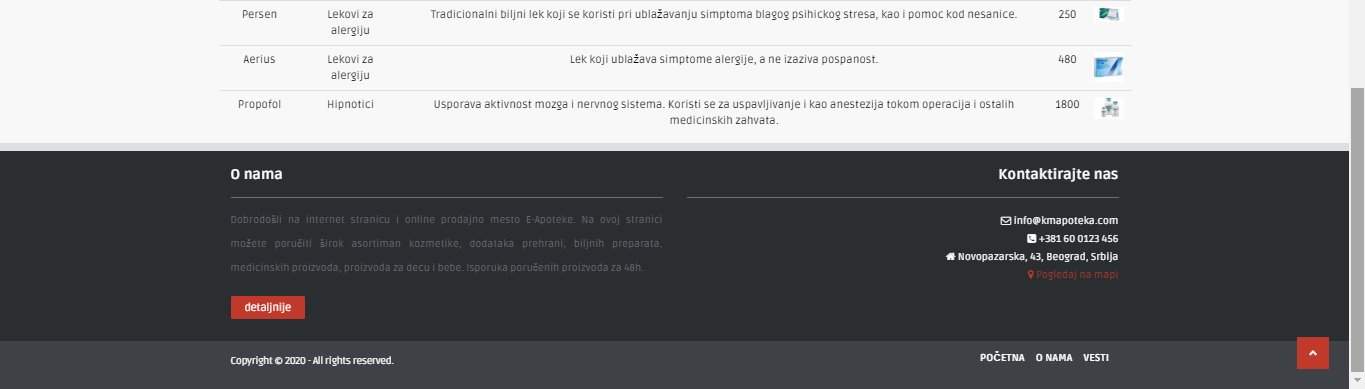
SQL je standardni programski jezik za pristup i manipulaciju podacima iz relacionih baza podataka i podržan je od strane svih servera relacionih baza podataka. U relacionim bazama podaci su skladišteni preko niza tabela. Svaka tabela sadrži kolone koje opsiuju atribute podataka, a svaki red je instanca podataka.

Relational Data Base Management Systems (RDBMS), odnosno sistemi za upravljanje relacionim bazama podataka su se u praksi pokazali kao najbolji način upravljanja podacima.

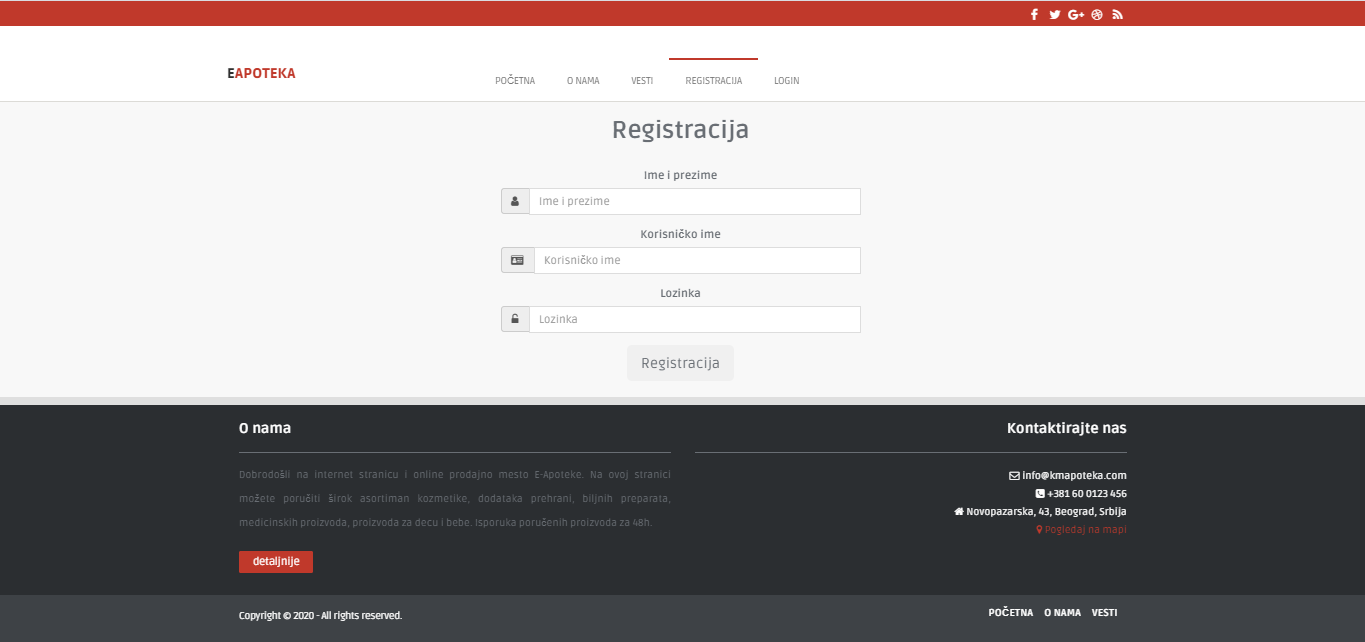
# Korisničko uputstvo – E-Apoteka

Izgled početne strane je dinamičan i zajednički za obe role korisnika.

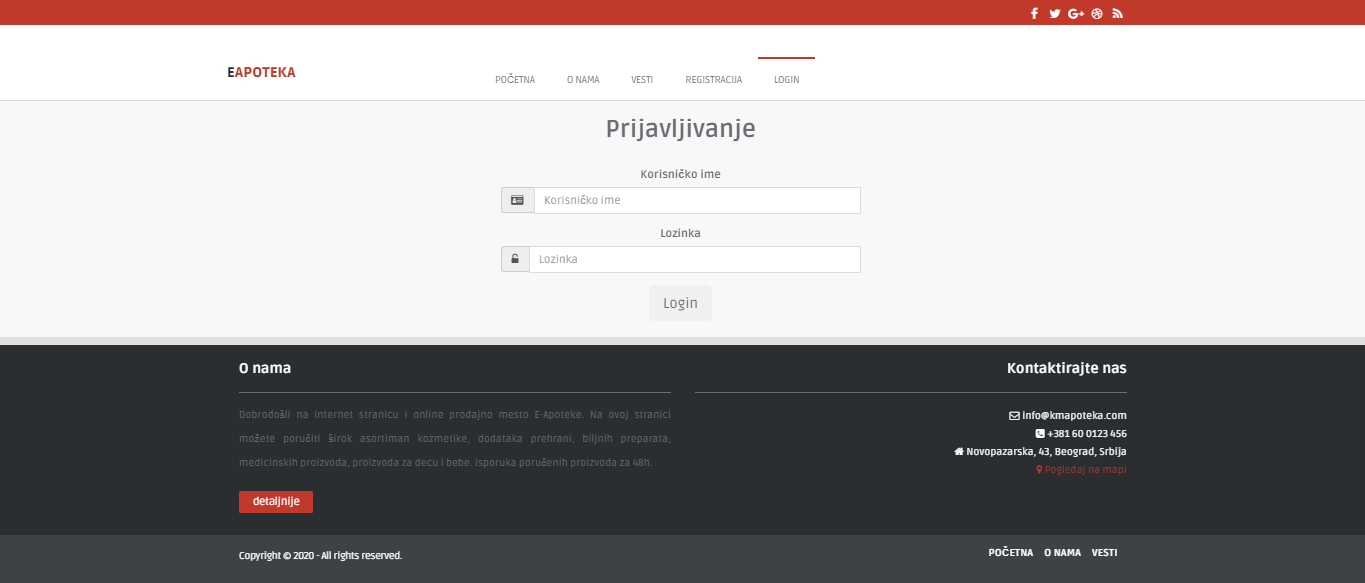




Kada se iz menija izabere opcija *Registracija*, otvara se stranica za registraciju. Korisnik u tekstualna polja unosi ime i prezime, korisničko ime i lozinku. Klikom na dugme registracija šalje se upit prema bazi podataka koji unosi novog korisnika u bazu. Nakon toga se korisnik može logovati pod tim korisničkim imenom i lozinkom. Ukoliko je korisnik uspešno registrovan, prikazuje mu se poruka da je uspešno registrovan.



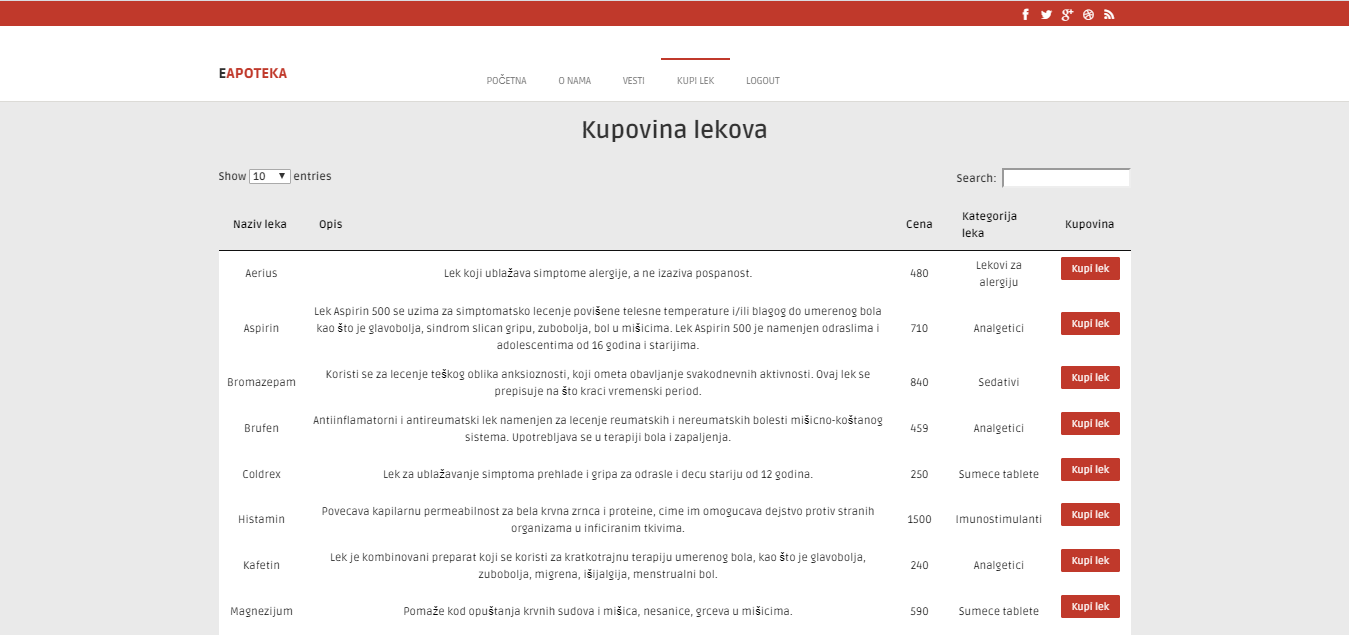
Korisnik u tekstualnim poljima unosi svoje korisničko ime i lozinku. Ukoliko takav korisnik postoji u bazi, korisnik postaje prijavljen i u zavisnosti od toga da li je u bazi označen kao administrator ili kao korisnik, razlikuje se meni bar. Ako je ulogovan adminstrator u meni baru postoji meni Administrator, a ako je ulogovan korisnik umesto toga se prikazuje meni Kupi lek. U oba slučaja vidljiv je i meni Logout. U slučaju da u bazi ne postoji korisnik sa tim korisničkim imenom i lozinkom prikazuje se poruka o neuspešnosti logovanja, tj prijave na sistem.

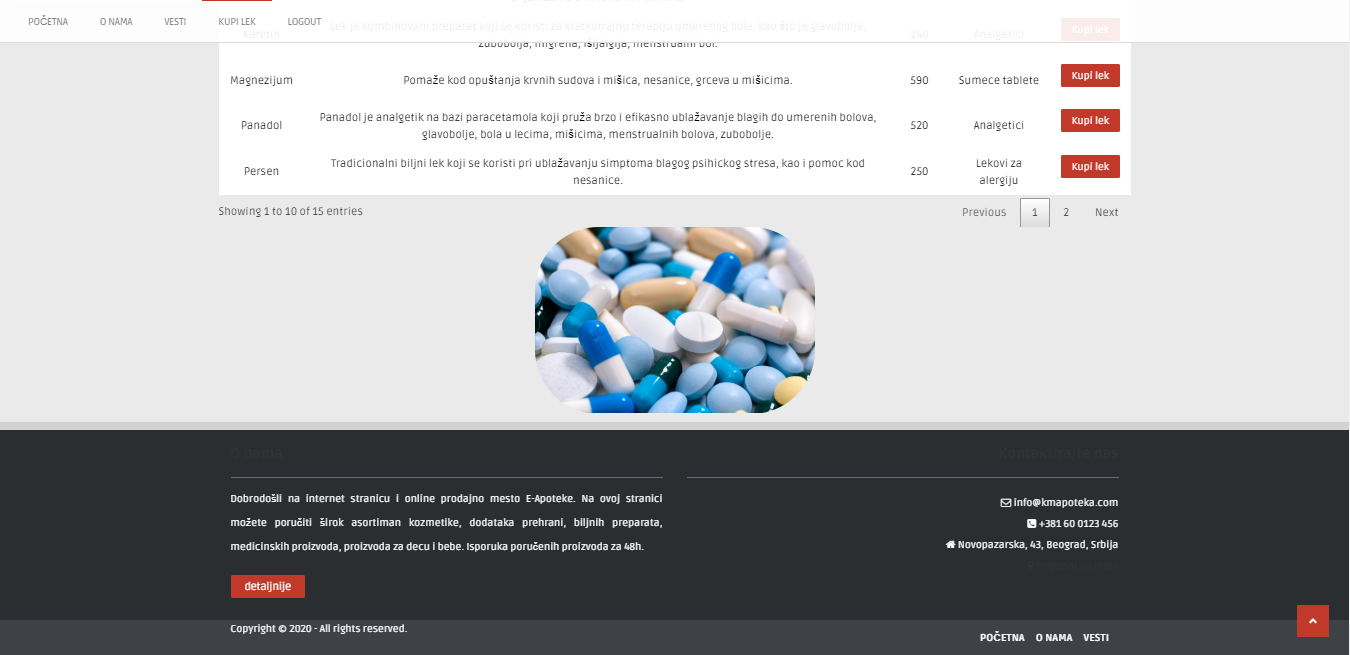


Korisnicima veb aplikacije se na uvid daju različite vesti iz oblasti zdravlja, dostupne u vidu istaknutih naslova, čiji se opširniji tekst nalazi na nekim od drugih sajtova gde linkovi vode.

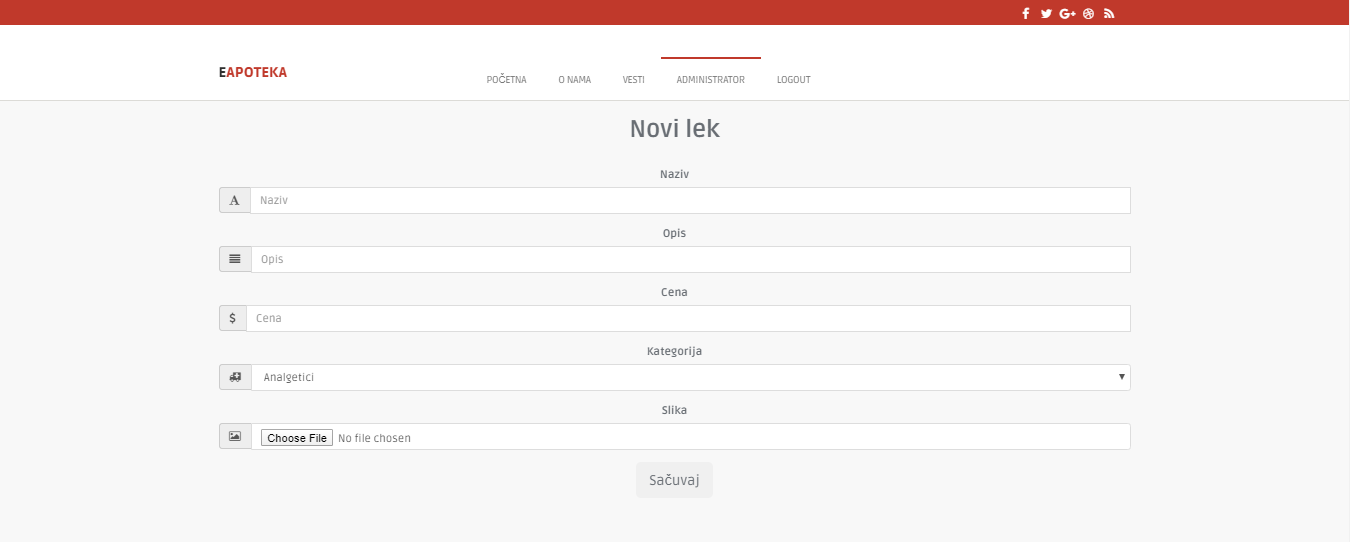


U slučaju da je ulogovan korisnik u meni baru je vidjiv meni Kupi lek. Kada korisnik to izabere, otvara se stranica prikazana na slici. Korisnik ima mogućnost da izabere koji lek želi da kupi. Iz tabele bira određeno pakovanje i pritiska dugme Kupi lek. Šalje se zahtev prema bazi koji dodaje novi objekat Kupovina koji ima polje daLiJeObavljena i čija je default vrednost false.





Ukoliko je ulogovan administrator, u meni baru je vidljiv meni Administrator. U tom meniju moguće je dodavanje novog leka. Administrator u tekstualna polja unosi naziv, opis, cenu, bira kategoriju iz odgovarajućeg polja grafičko-korisničkog interfejsa i bira sliku željenog leka, koja je dostupna u njegovim fajlovima. Pritiskom na dugme Sačuvaj šalje se zahtev za dodavanje novog leka. U slučaju da je novi lek uspešno zapamćen u bazi, administratoru se prikazuje poruka o uspešnosti. Isto se događa i u slučaju da lek ne može da se zapamti u bazi.



Administrator ima mogućnost da potvrdi korisnikov zahtev za kupovinom. U tabeli su izlistani svi zahtevi sa nazivom leka koji je poručen, opisom, imenom kupca koji je taj zahtev podneo i datumom podnošenja zahteva za kupovinom. To su svi objekti iz tabele kupovina. U slučaju da je vrednost atributa daLiJeObavljena true, u tabeli je prikazano Potvrđen/a. U suprotnom je prikazano Nije potvrđen/a. Administrator klikom na dugme Potvrdi, potvrđuje zahtev za kupovinom i time menja status atributa daLiJeObavljena na true.



Kao i u slučaju potvrđivanja zahteva, isto se odigrava i u slučaju brisanja zahteva. Administrator klikom na dugme Obriši šalje zahtev bazi da se obriše izabran objekat iz tabele kupovina. Kada se osveži stranica prikazuje se izmenjana tablela.

# Literatura

1. Materijali sa sajta http://www.elab.rs/
2. Materijali sa sajta http://www.w3schools.com/