**Веб апликација за преглед и управљање подацима о матичару**

Никола Младеновић

Факултет техничких наука

Универзитет у Новом Саду

Трг Доситеја Обрадовића 6, 21000 Нови Сад

mladenovic.sr7.2020@uns.ac.rs

**Сажетак - У овом раду описана је веб апликација за матичара која омогућава ефикасно управљање подацима о матичним књигама о рођенима, венчанима и умрлима. За имплементацију су коришћене технологије као што су Spring Boot, Angular и MySQL. Примена овог решења доноси предности као што су олакшано вођење књига, аутоматско ажурирање података, могућност праћења историје промена и интероперабилност са другим информационим системима у матичним службама. Кроз развој прототипа апликације детаљније је описан његова функционалност и практична примена.**

**Кључне речи - матичар; књиге рођених; књиге венчаних; књиге умрлих; подаци о рођенима; подаци о венчанима; подаци о умрлима; историја промена.**

* **УВОД**

Матичар је кључна особа у систему евиденције грађанских догађаја као што су рођења, венчања и смрти. Он има одговорност да води и ажурира књиге рођених, венчаних и умрлих, како би се осигурала тачност и легитимност података. Међутим, постоје изазови у вођењу евиденција и обради података који захтевају модернизацију и ефикасније решење.

Проблеми у традиционалном систему матичара укључују ручно вођење папирних књига и формулара. Ово може довести до грешака, губљења података и спорог приступа информацијама. Такође, грађани су често приморани да чекају на шалтерима и обављају процедуре као што је пријава рођења или промена података, што им отежава и захтева време.

Са циљем модернизације система вођења евиденција и обраде података, развијен је нови информациони систем за матичара. Овај систем омогућава ефикасно и транспарентно управљање подацима о рођенима, венчанима и умрлима. Користећи напредне технологије као што су веб апликације и базе података, матичар има могућност брзог приступа, ажурирања и претраге релевантних информација.

Нови систем нуди низ предности, укључујући:

* **Ефикасност** - Процеси вођења евиденција и обраде података постају бржи и ефикаснији. Матичар може лако приступити и ажурирати податке без потребе за ручним уносом или претраживањем папирних докумената.
* **Тачност** - Ризгрешака се значајно смањује јер се подаци уносе у електронском облику и могу се лако проверити и исправити.
* **Доступност** - Грађаним је омогућен једноставан приступ подацима и обављање процедура без потребе за физичким присуством на шалтерима. Могу приступити информационом систему преко интернета и обавити потребне захтеве.
* **Безбедност** - Подаци о грађанским догађајима се чувају у сигурној електронској окружењу, што смањује ризик од губитка или неовлашћеног приступа информацијама.
* **Аутоматизација** - Нови систем омогућава интеграцију са другим системима, као што су катастар, нотар или органи за јавну исправу. Ово умањује потребу за ручним преносом података и додатно убрзава процесе.
* **Анализа података** - Информациони систем матичара омогућава анализу података и статистичких информација, што може бити корисно за планирање и доношење одлука у различитим областима.

Укратко, нови информациони систем за матичара пружа модерно и ефикасно решење за вођење евиденција грађанских догађаја. Омогућава брз приступ, тачност и безбедност података, а такође и аутоматизацију процеса и анализу информација. Овај систем отвара пут ка напредном и дигиталном управљању подацима о грађанским догађајима, усмеравајући га ка бољој услузи грађана и општој ефикасности управљања.

* **СРОДНА РЕШЕЊА**

Овај одељак садржи преглед сродних или сличних апликација намењених прегледу и управљању подацима о грађанима.

**Mатични завод** - Матични завод је национални завод у Србији који се бави вођењем матичних књига и обрадом података о рођенима, венчанима и умрлима. На њиховом сајту можете пронаћи информације о њиховом раду, услугама и законима који регулишу матичарске послове. Такође, матични завод пружа могућност приступа онлајн услугама, као што је еШалтер за грађане, где можете обавити одређене матичарске процедуре путем интернета.

**Електронска Матична Књига** - Овај пројекат је иницијатива Министарства државне управе и локалне самоуправе у Србији и има за циљ модернизацију и дигитализацију система матичара. Електронска Матична Књига омогућава електронско вођење и обраду матичних података, што убрзава процесе и олакшава приступ информацијама за грађане.

**Градска управа Београда** - Матичне службе у Београду имају своје информационе системе и услуге које омогућавају онлајн приступ и обраду података. Грађани Београда могу приступити веб-порталу Градске управе и обавити различите матичарске процедуре, као што је пријава рођења или захтев за извод из матичне књиге венчаних.

**Локалне самоуправе** - Многе локалне самоуправе у Србији имају своје системе и услуге за матичарске послове. Оне могу пружати услуге путем својих веб-сајтова или преко шалтера за услуге грађана.

* **Коришћене технологије**

За развој клијентског дела апликације користили смо Angular, TypeScript, HTML и Bootstrap. Angular је савремена развојна платформа изграђена на TypeScript-у која нуди богат скуп компоненти за израду скалабилних веб апликација. Са Angular-ом смо имали могућност да развијемо интерактивне корисничке интерфејсе, управљамо формама, рутирамо странице и обављамо клијент-сервер комуникацију.

За стилизовање и дизајн користили смо HTML и Bootstrap. HTML је основни језик за обележавање веб страница који смо користили за структурирање садржаја апликације. Bootstrap је популарни CSS фрејмворк који нам је помогао да брзо и ефикасно додамо стилове, компоненте и респонзивни дизајн у нашу апликацију.

За развој серверског дела апликације користили смо Spring Boot. Spring Boot је моћан и лак за употребу фрејмворк за развој Java базираних веб апликација. Са Spring Boot-ом смо имали могућност да брзо креирамо и покренемо стабилне серверске апликације. Он нам је омогућио лаку конфигурацију, интеграцију са базом података и другим спољним библиотекама, као и развој безбедних и ефикасних API-ја.

За чување података користили смо MySQL базу података. MySQL је популарна релациона база података отвореног кода која се користи у бројним апликацијама широм света. Пружа нам скалабилност, поузданост и високу доступност података. Будући да је лак за коришћење и има богат скуп функционалности, MySQL је био идеалан избор за нашу апликацију.

У развоју наше апликације, ове технологије су нам пружиле могућност да ефикасно и сигурно развијемо клијентску и серверску инфраструктуру, омогућујући нам да створимо кориснички прилагодљиве интерфејсе, обрадимо и чувамо податке, и обезбедимо брзу и сигурну комуникацију између клијента и сервера.

* **Спецификација захтева**

У овом поглављу се налазе функционални и нефункционални захтеви апликације за матичара.

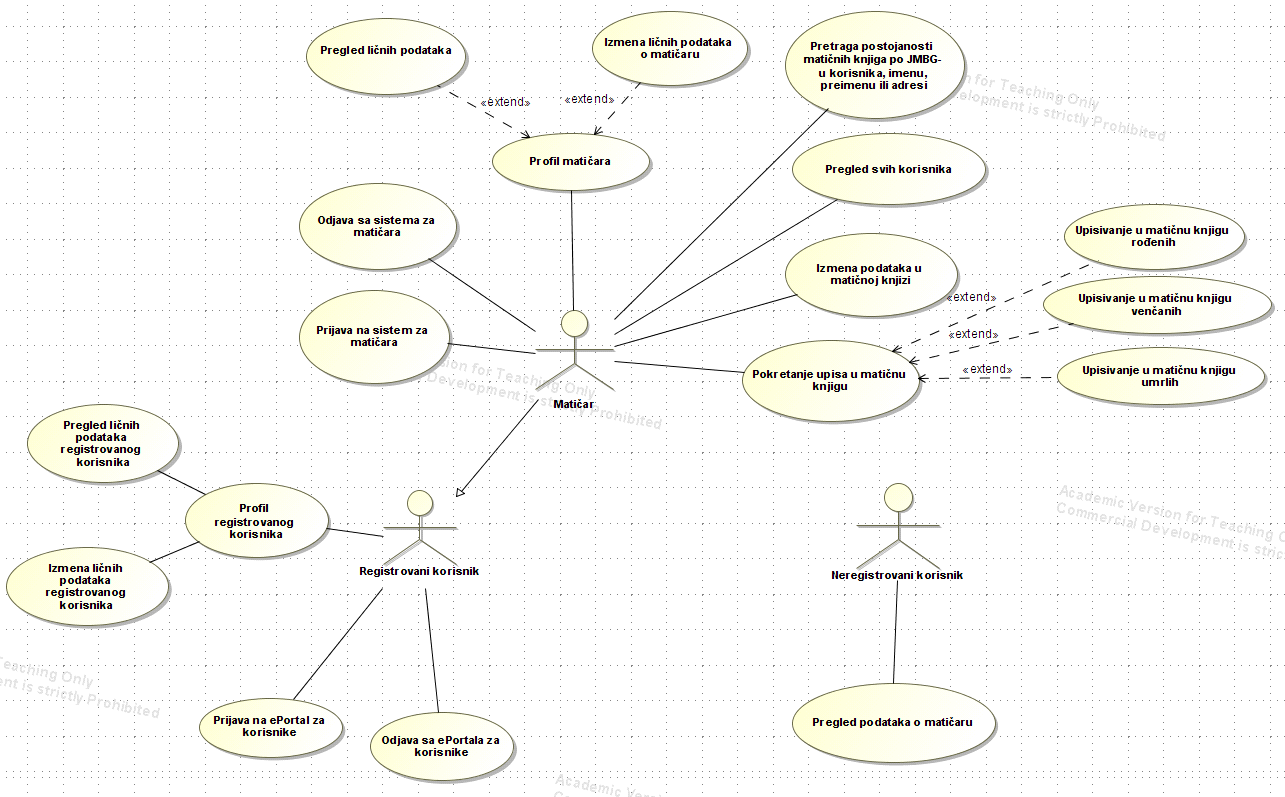
* ***Спецификација функционалних захтева***

У овом делу су описани сви функционални захтеви које је потребно да испуњава један информациони систем за матичара.

Функционални захтеви овог софтверског решења су представљени *UML* дијаграмом случајева коришћења. Кроз употребу UML дијаграма случајева коришћења, описао сам различите интеракције између корисника и система. Идентификовао сам главне актере који су укључени у систем и дефинисао њихове улоге и циљеве.

Кроз детаљно мапирање случајева коришћења, приказао сам функционалности система и различите сценарије интеракције између корисника и система. Ови дијаграми су ми омогућили да јасно дефинишем захтеве система и разумем корисничке потребе.

Кроз UML дијаграм случајева коришћења сам моделоваоглавне функционалности система, идентификовао корисничке циљеве и приказао токове догађаја који се дешавају током интеракције са системом. Ови дијаграми су ми пружили јасан и структуриран приказ како корисници користе систем и шта се очекује од система у одговору.



У дијаграму случаја коришћења, имамо следеће актере: **матичар**, **регистровани корисник** и **нерегистровани корисник**.

***Матичар*** има следеће случајеве коришћења:

* Уписивање новорођенчади: Требало би да се омогући матичару да лако упише новорођенчади у родни лист, унесе податке о новорођенчади и родитељима, и да изда извод из матичне књиге рођених.
* Уписивање венчања: Требало би да се омогући матичару да лако упише венчања у матичну књигу венчаних, унесе податке о венчању и брачним партнерима, и да изда извод о венчању.
* Уписивање смрти: Требало би да се омогући матичару да лако упише смрти у матичну књигу умрлих, унесе податке о особи која је преминула и о њеним рођацима, и да изда извод о смрти.
* Прикупљање и чување докумената у бази података: Требало би да се омогући матичару да лако прикупља и чува докуменате везане за матичне књиге у базу података.
* Претраживање матичних књига: Требало би да се омогући претраживање матичних књига на основу различитих критеријума, као што су име особе, датум рођења, датум венчања итд.
* Издавање извода: Требало би да се омогући издавање извода из матичних књига на захтев грађана или других надлежних органа.
* Издавање извода из матичних књига: Требало би да се омогући матичару да издаје изводе из матичних књига, како би се корисницима омогућило да добију информације о некретнинама.
* Провера о постојаности корисника: Матичар би требало да има могућност проверавања да ли корисник постоји у матичној књизи. Његова постојаност може да се провери претрагом његовог ЈМБГ-а, имена и презимена, адресе.
* Комуникација са другим системима: Апликација би требало да буде у могућности да комуницира са другим системима у геодетском заводу, као што су катастар и општине.

***Регистровани корисник*** има следеће случајеве коришћења:

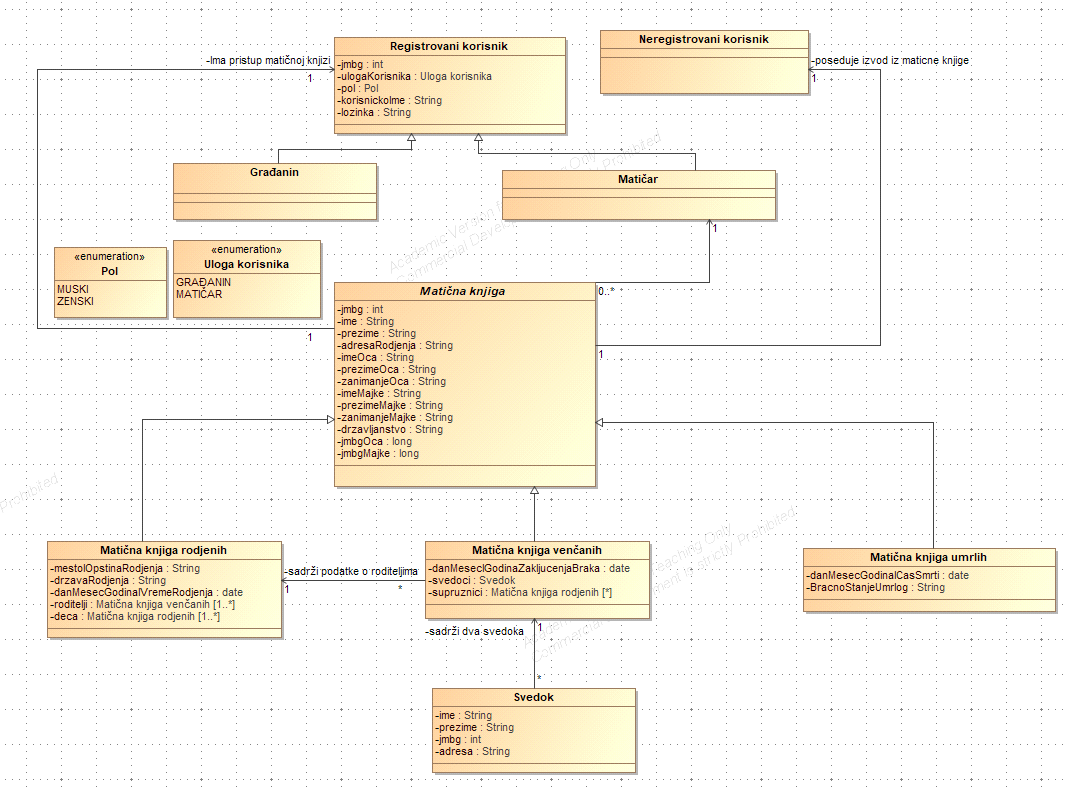
* Преглед свог профила: Регистровани корисник може да прегледа информације о свом профилу, укључујући контактне податке и личне информације.
* Преглед матичних књига: Омогућава регистрованом кориснику да приступи и прегледа матичне књиге рођених, венчаних и умрлих особа.

***Нерегистровани корисник*** има следећи случај коришћења:

* Преглед матичних књига: Нерегистровани корисник има могућност да приступи и прегледа матичне књиге рођених, венчаних и умрлих особа. Међутим, ограничен је само на преглед информација и нема могућност да уноси или ажурира податке.

***Нефункционални захтеви***

* Безбедност и приватност: Апликација треба да има механизме за заштиту података и омогући приступ само овлашћеним корисницима. Треба да се примене мере заштите података, криптографија и заштита од неовлашћеног приступа.
* Перформансе: Апликација треба да обради захтеве корисника у прихватљивом временском оквиру. Ово укључује брзо претраживање и обраду података, минимално време чекања и ефикасну употребу ресурса.
* Скалабилност: Апликација треба да буде скалабилна и да се лако пристосава растућем броју корисника и обиму података. Треба да подржава равномерну расподелу терета и да се адаптира на променљиве услове коришћења.
* Доступност: Апликација треба да буде доступна у сваком тренутку и да има минимално време неактивности или прекида рада. Треба да се предузму мере за резервно копирање података и осигурање континуитета пословних операција.
* Корисничко искуство: Апликација треба да буде интуитивна и лака за коришћење, са јасним и разумљивим интерфејсом. Треба да пружи угодно и задовољавајуће корисничко искуство и да омогући приступ апликацији на различитим уређајима.
* Системске зависности: Апликација треба да буде компатибилна са различитим оперативним системима и конфигурацијама хардвера. Треба да задовољи системске захтеве и да буде способна да се интегрише.



Класни дијаграм који смо израдили у UML-у садржи следеће класе и везе:

* Регистровани корисник (апстрактна класа):

Класа која представља регистрованог корисника апликације.

Служи као основна класа за конкретне класе: Грађанин и Матичар.

* Грађанин:

Класа која представља грађанина.

Везана је за апстрактну класу Матична књига преко асоцијације.

* Матичар:

Класа која представља матичара.

Везана је за апстрактну класу Матична књига преко асоцијације.

Матичар управља Матичним књигама.

* Пол (енумерациона класа):

Класа која представља пол особе.

Користи се у класама Грађанин и Матичар.

* Улога (енумерациона класа):

Класа која представља улогу регистрованог корисника.

Користи се у класи Грађанин.

* Матична књига (апстрактна класа):

Класа која представља матичну књигу.

Основна класа за конкретне класе: Матична књига рођених, Матична књига венчаних и Матична књига умрлих.

Повезана је са класом Матичар преко асоцијације.

* Матична књига рођених:

Класа која представља матичну књигу рођених.

Везана је за апстрактну класу Матична књига преко наслеђивања.

* Матична књига венчаних:

Класа која представља матичну књигу венчаних.

Везана је за апстрактну класу Матична књига преко наслеђивања.

* Матична књига умрлих:

Класа која представља матичну књигу умрлих.

Везана је за апстрактну класу Матична књига преко наслеђивања.

Овај дијаграм описује хиерархију класа и везе између њих. Регистровани корисник је апстрактна класа која служи као основа за грађанина и матичара. Постоје две енумерационе класе, пол и улога, које се користе у класи грађанин. Матична књига је апстрактна класа која представља основу за конкретне класе матичне књиге рођених, венчаних и умрлих. Матичар управља матичним књигама. Овај дијаграм пружа преглед класа и њихових веза, што помаже у разумевању организације и веза између класа у апликацији матичара.

* **ИМПЛЕМЕНТАЦИЈА**

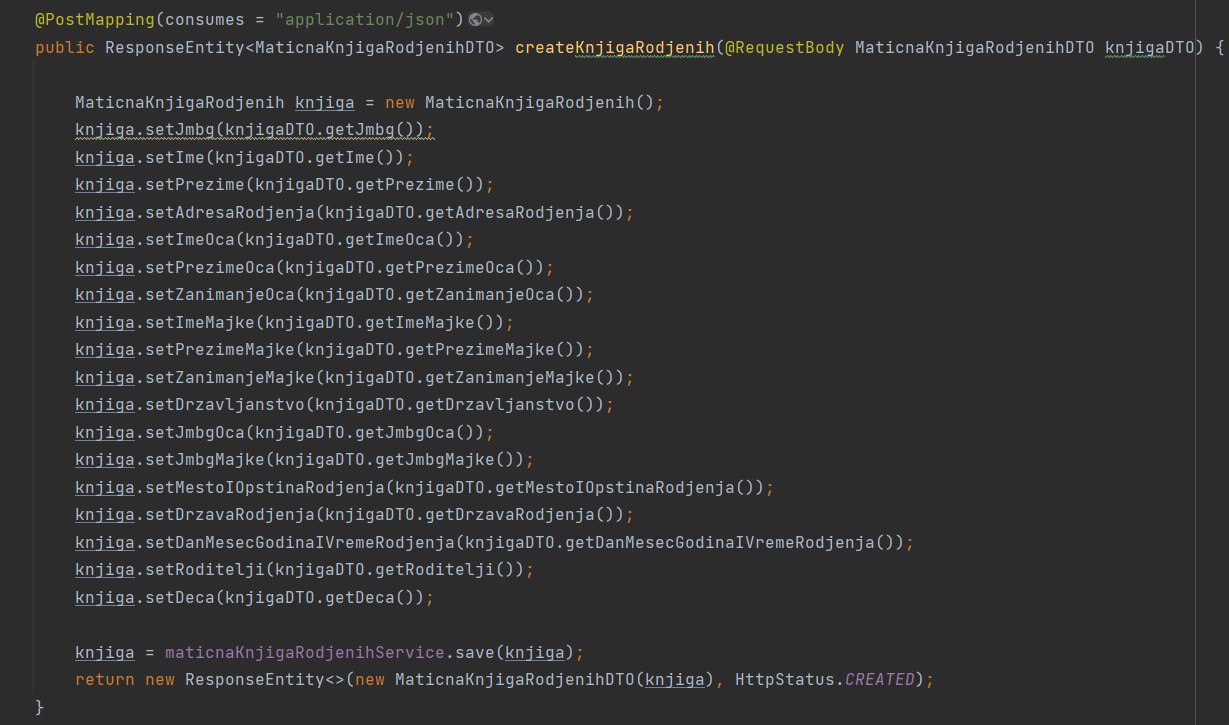
У овом одељку семинарског рада ће бити приказана имплементација функционалности за матичара. Биће објашњен начин на који су имплементирани одређени процеси, укључујући пријаву на систем, унос и ажурирање података у матичним књигама, претрагу и преглед матичних записа и издавање извода. Детаљно ће бити описана имплементација и на серверској и на клијентској страни апликације.

***Имплементација серверског дела апликације***

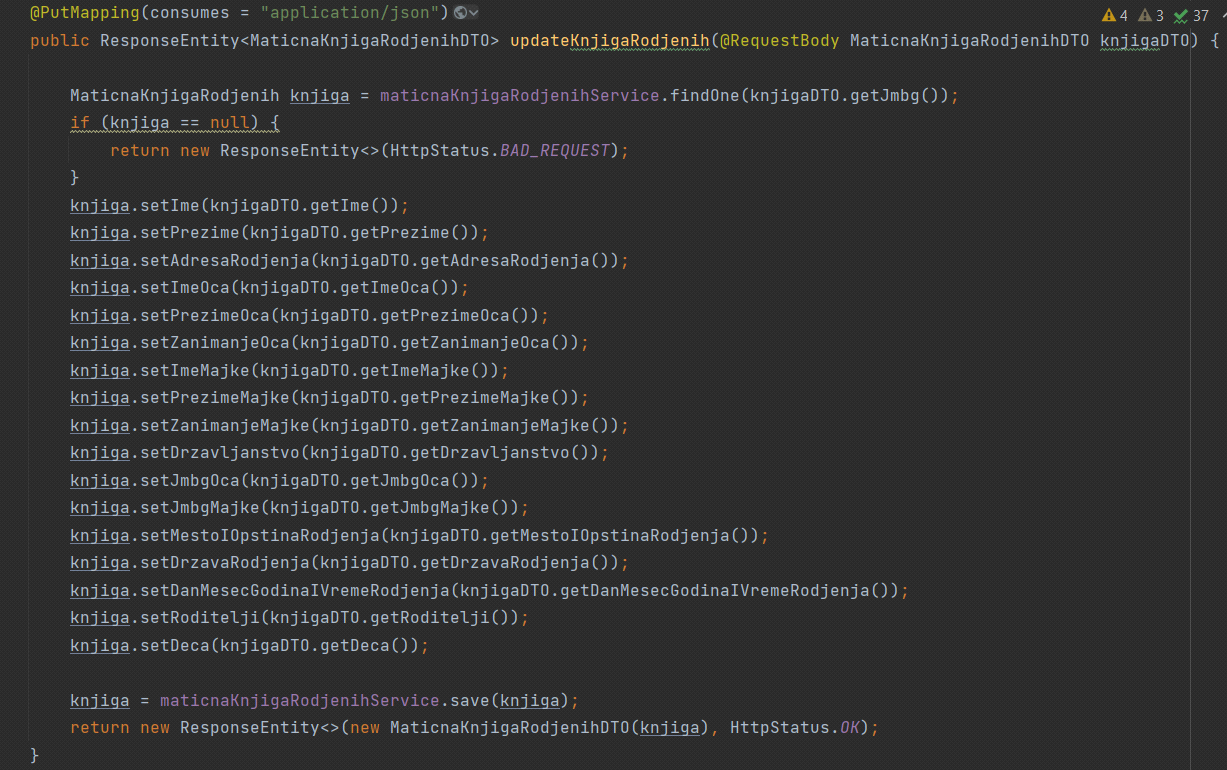
Ово су неке од функционалности на серверском делу апликације:



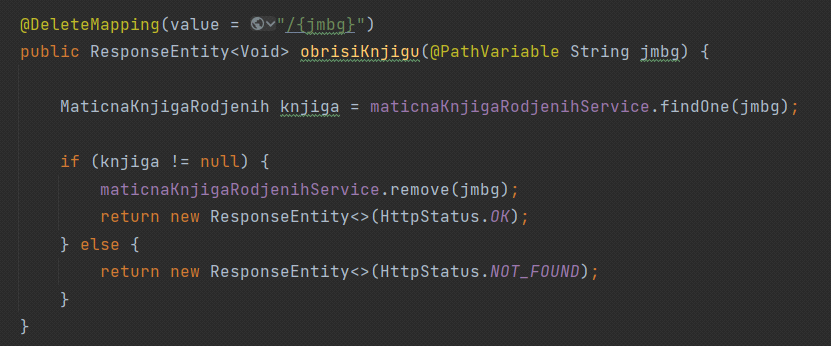
У серверском делу имплементирана је метода за добијање свих књига рођених. Ова метода користи HTTP GET захтев и има путању "/all". У телу методе, прво се добијају све књиге рођених позивом функције findAll() сервиса maticnaKnjigaRodjenihService. Затим се прољеће све добијене књиге и за сваку књигу се креира објекат MaticnaKnjigaRodjenihDTO, који садржи релевантне информације о књизи рођених. На крају, листа књига DTO објеката се враћа као одговор клијенту у виду ResponseEntity са HTTP статусом OK.



Такође имплементирана је метода за додавање књиге рођених. Ова метода користи HTTP POST захтев и прихвата JSON објекат типа MaticnaKnjigaRodjenihDTO као улазни параметар. У телу методе, креира се нови објекат MaticnaKnjigaRodjenih и постављају се вредности атрибута на основу података из добијеног DTO објекта. Након тога, књига рођених се чува позивом функције save() сервиса maticnaKnjigaRodjenihService. На крају, као одговор клијенту се враћа ResponseEntity са HTTP статусом CREATED и креирана књига рођених у облику MaticnaKnjigaRodjenihDTO.



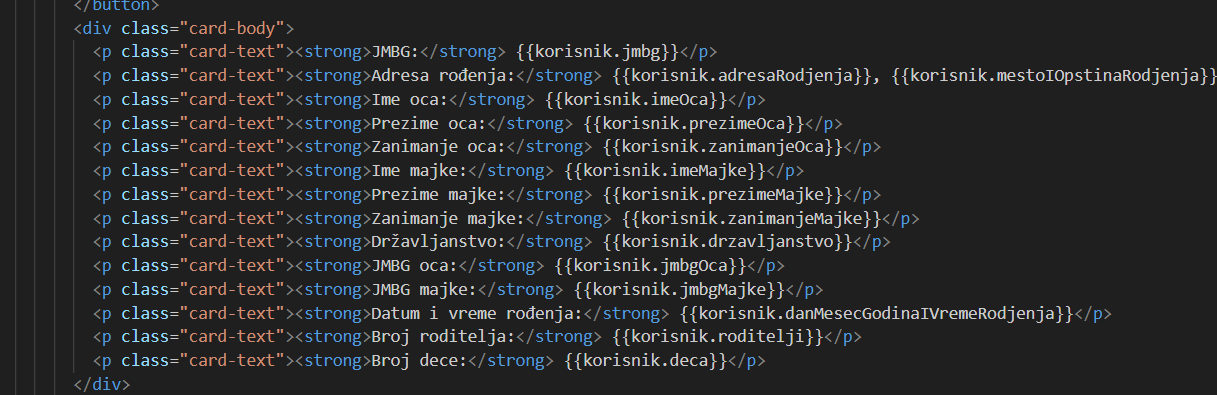
У серверском делу имплементирана је метода за ажурирање књиге рођених. Ова метода користи HTTP PUT захтев и прихвата JSON објекат типа MaticnaKnjigaRodjenihDTO као улазни параметар. У телу методе, прво се позива функција findOne() сервиса maticnaKnjigaRodjenihService да би се пронашла постојећа књига рођених на основу JMBG-а. Уколико књига не постоји, враћа се ResponseEntity са HTTP статусом BAD\_REQUEST. У супротном, атрибути књиге рођених се ажурирају на основу података из добијеног DTO објекта. Након ажурирања, књига рођених се чува позивом функције save() сервиса maticnaKnjigaRodjenihService. На крају, као одговор клијенту се враћа ResponseEntity са HTTP статусом OK и ажурирана књига рођених у облику MaticnaKnjigaRodjenihDTO.

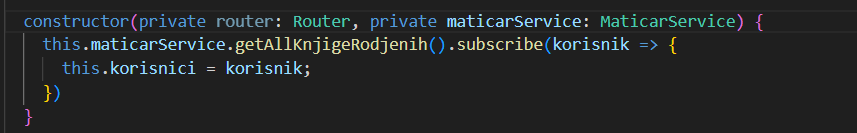


У серверском делу имплементирана је метода за брисање књиге рођених. Ова метода користи HTTP DELETE захтев и прослеђује JMBG као путању (параметар) у URL-у. У телу методе, прво се позива функција findOne() сервиса maticnaKnjigaRodjenihService да би се пронашла постојећа књига рођених на основу JMBG-а. Уколико књига постоји, позива се функција remove() сервиса maticnaKnjigaRodjenihService да би се обрисала књига рођених. Као одговор клијенту, ако је брисање успешно, враћа се ResponseEntity са HTTP статусом OK. Ако књига није пронађена, враћа се ResponseEntity са HTTP статусом NOT\_FOUND.

***Имплементација клијентског дела апликације***

У следећем одељку, прелазимо на демонстрацију клијентског дела апликације. У овом делу ће бити приказана имплементација функционалности и интерфејса које корисник може да користи за приступ и манипулацију подацима.

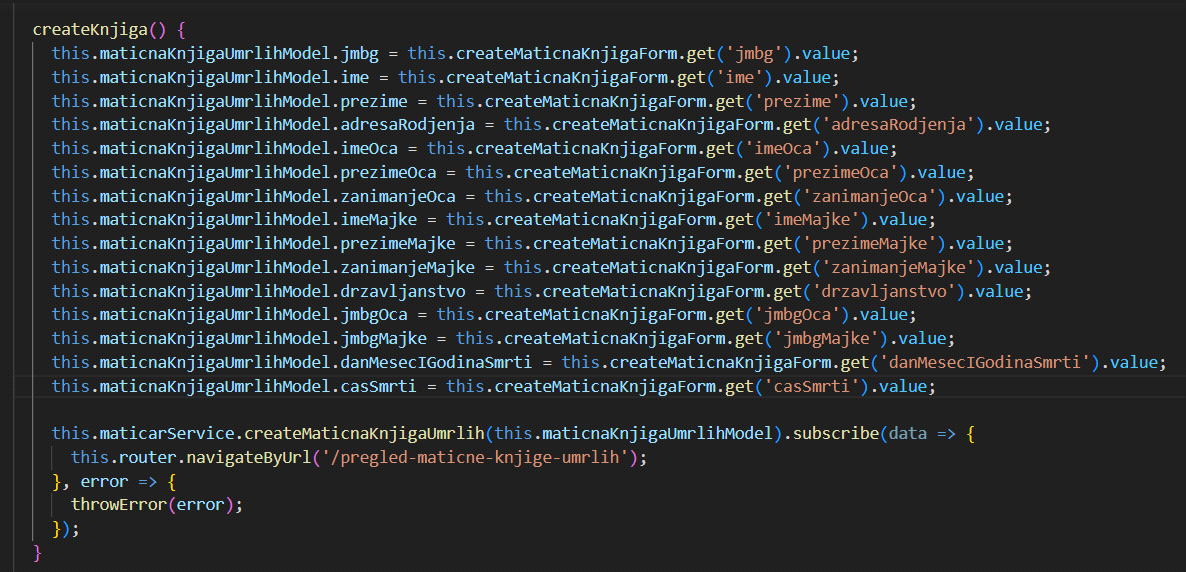




У HTML делу, ово представља приказ података из матичне књиге рођених. Користећи различите директиве, вредности прослеђене из TypeScript дела се убацују у одговарајуће место у текстуалном делу картице. На пример, приказују се ЈМБГ, адреса рођења, имена родитеља итд.

У TypeScript делу, у конструктору, користи се сервис "maticarService" за добављање свих књига рођених. Позивањем функције "subscribe", добијени подаци се додељују променљивој "korisnik". Након тога, тај низ објеката се прослеђује променљивој "korisnici", која се користи у HTML делу за приказ података.

Овај код омогућава приказ свих књига рођених на клијентској страници, користећи податке добијене са сервера.

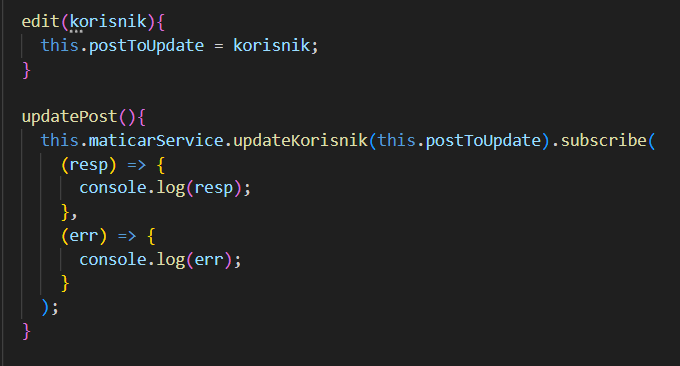


Ова функција "createKnjiga()" се позива приликом креирања нове књиге рођених. У овој функцији, вредности из форме за креирање књиге рођених се премештају у одговарајуће поље модела "maticnaKnjigaRodjenihModel".

На пример, вредност из поља "jmbg" се примењује на "jmbg" поље модела, вредност из поља "ime" се примењује на "ime" поље модела итд. Ова радња се понавља за сва поља форме.

Након што су све вредности премештене у модел, позива се функција "createMaticnaKnjigaRodjenih()" сервиса "maticarService" како би се креирала нова књига рођених. У случају успешног креирања, корисник се преусмерава на страницу за преглед матичне књиге рођених. У случају грешке, се грешка прослеђује и обрађује.

Ова функција омогућава креирање нове књиге рођених на клијентској страници и комуникацију са серверским делом апликације.

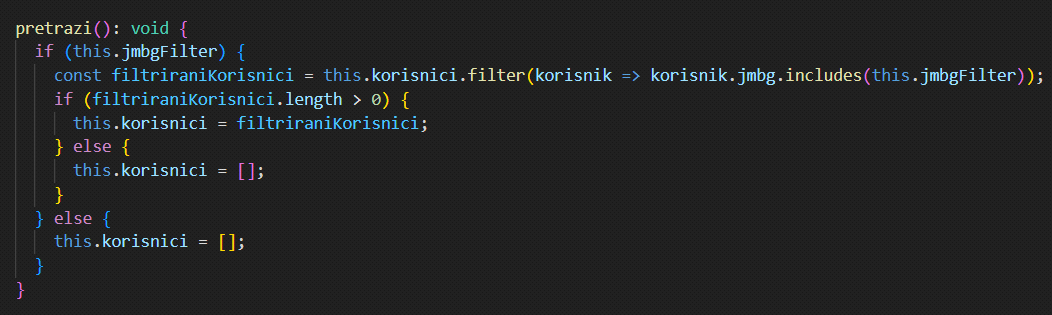


Метода "edit(korisnik)" се користи за промену постојеће књиге рођених. Прослеђени објекат "korisnik" представља податке о књизи који ће бити ажурирани. У овој методи, постојећи објекат књиге рођених се премешта у приватну променљиву "postToUpdate".

Метода "updatePost()" се користи за ажурирање постојеће књиге рођених на серверској страни. Користи се "maticarService" сервис за комуникацију са серверским делом апликације. Позива се метода "updateKorisnik()" сервиса и прослеђује се приватна променљива "postToUpdate" која садржи ажуриране податке о књизи.

У случају успешног ажурирања, се добијени одговор исписује у конзоли. У случају грешке, се исписује грешка у конзоли.

Ове методе омогућавају ажурирање постојеће књиге рођених на клијентској страници и комуникацију са серверским делом апликације.



Метода "pretrazi()" се користи за pretragu knjiga rođenih na osnovu JMBG-a.

У оквиру ове методе, прво се проверава да ли постоји вредност у променљивој "jmbgFilter" која представља JMBG претрагу. Ако постоји, примењује се филтрација над низом "korisnici" који садржи све књиге рођених. Филтра се извршава на основу услова где се проверава да ли JMBG књиге рођених садржи вредност из "jmbgFilter".

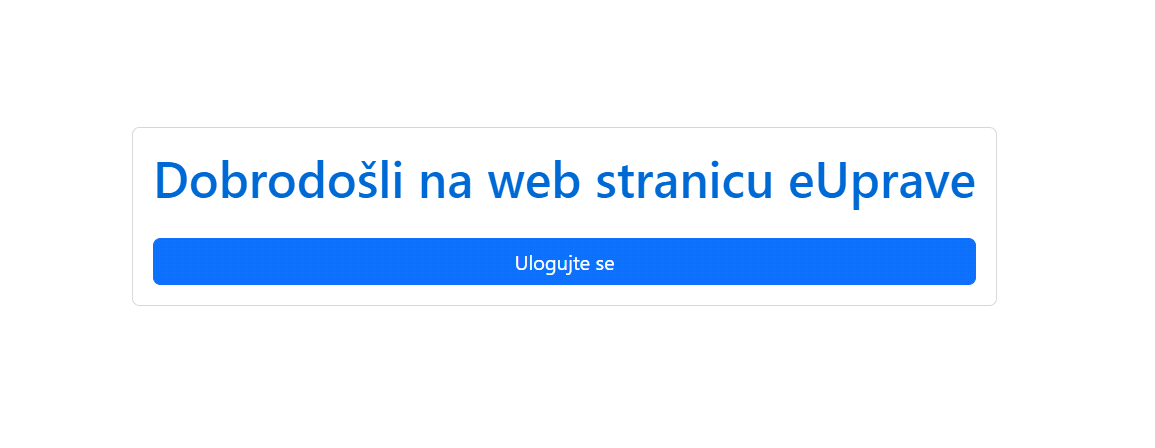
Уколико постоје књиге рођених које задовољавају услов претраге, филтрирани записи се преписују у променљиву "korisnici", чиме се приказују само филтрирани резултати. Ако нема пронађених књига рођених, променљива "korisnici" се поставља на празан низ, чиме се исписује празан резултат.

Уколико променљива "jmbgFilter" нема вредност (је празна), променљива "korisnici" се такође поставља на празан низ.

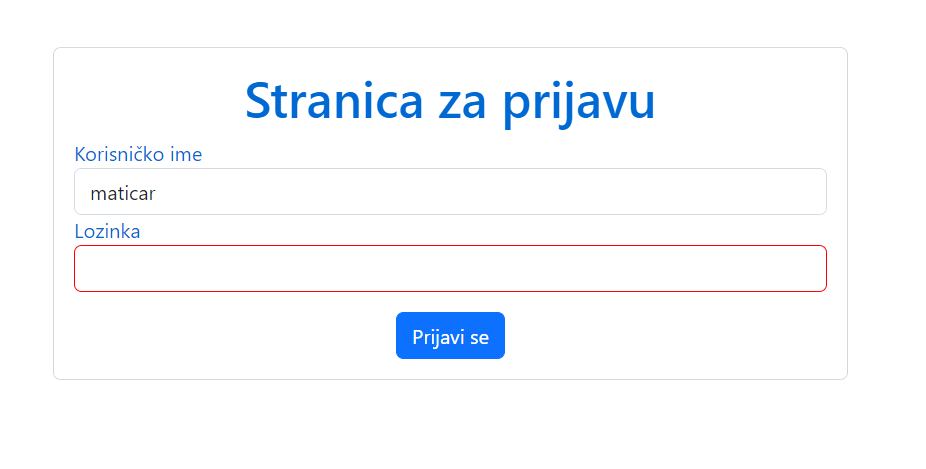
Ова метода омогућава претрагу књига рођених на клијентској страници на основу JMBG-а и приказује резултате претраге у складу са унесеним критеријумом.

* **Демонстрација**

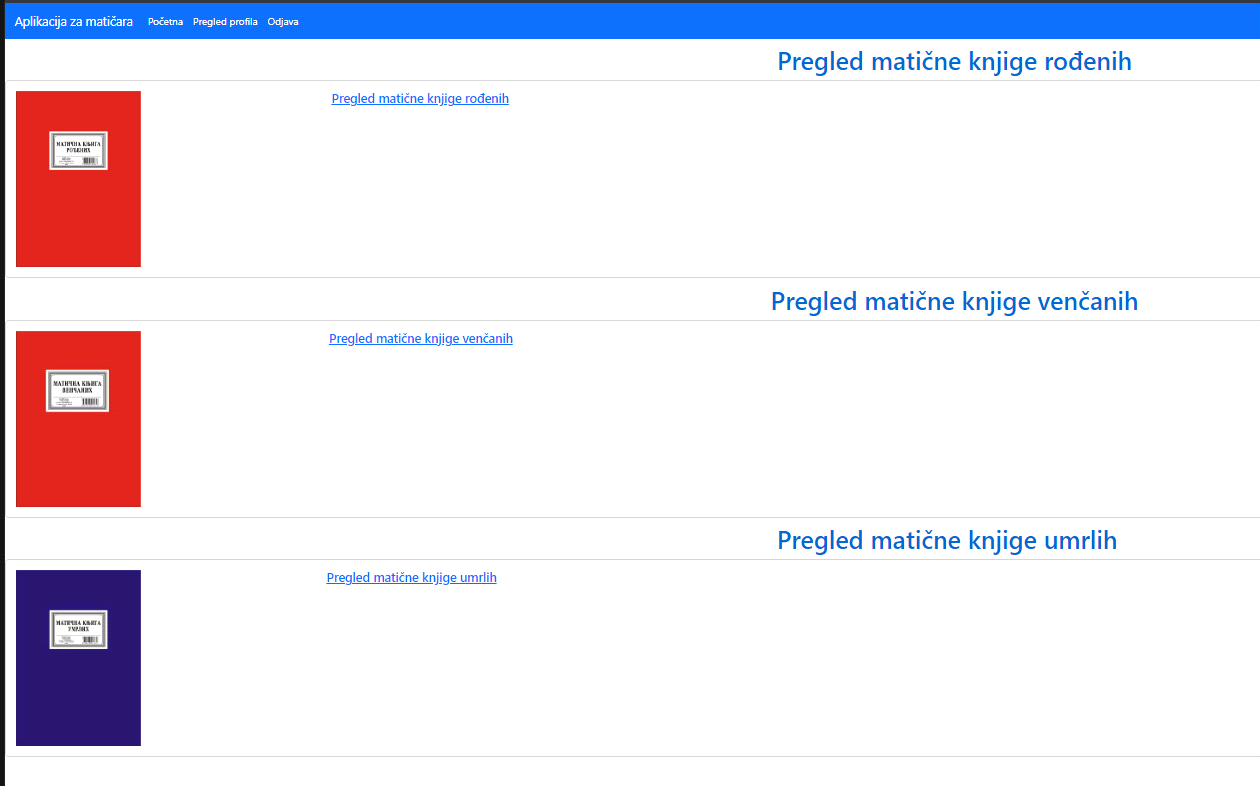
У овом одељку следи демонстрација покретања апликације и приказ слика апликације у раду.



На првој слици приказана је почетна страница апликације. Корисник има могућност да кликне на дугме за пријаву (login) које се налази на страници. Кликом на ово дугме, корисник ће бити преусмерен на страницу за пријаву где ће моћи да унесе своје податке за пријаву и настави са коришћењем апликације.

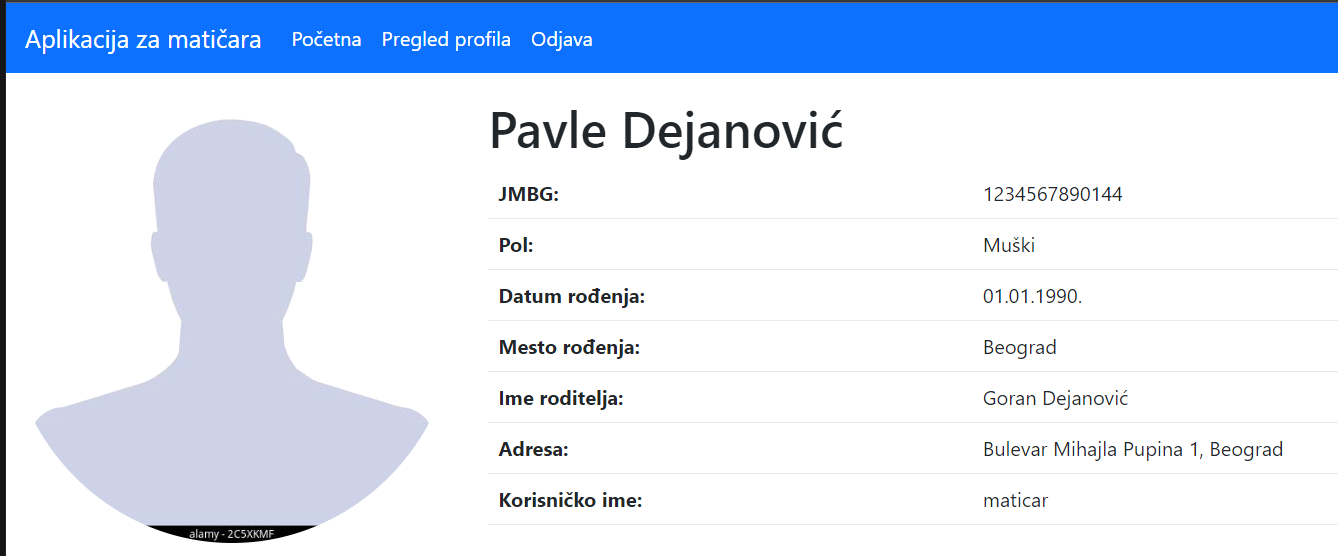


На страници за пријаву кориснику се приказује форма за унос података. Форма садржи два поља за унос: корисничко име и лозинка. Корисник треба да унесе своје корисничко име и лозинку у одговарајућа поља. Овај процес се обично користи за проверу и верификацију идентитета корисника пре него што им се омогући приступ апликацији. После успешне пријаве, кориснику се пружају додатне могућности и функционалности апликације.

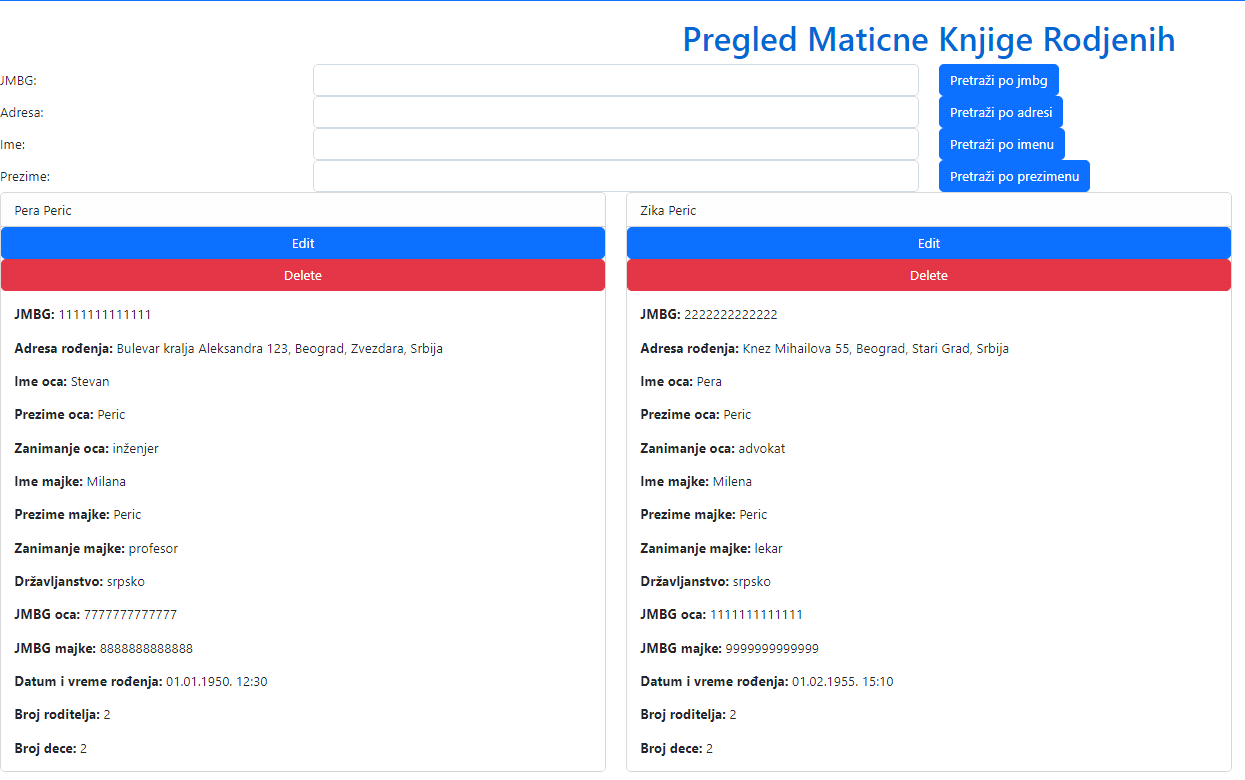


На слици се приказује почетна страна матичара. На страници су видљиве три опције: преглед матичне књиге рођених, венчаних и умрлих. Избором ових опција, матичар може приступити и прегледати релевантне информације из сваке од матичних књига. Такође, на страници се налази навигациони мени (navbar) где матичар може приступити свом профилу и извршити одјаву са апликације. Ова почетна страна пружа удобност и лак приступ различитим функционалностима и информацијама које су битне за рад матичара

.



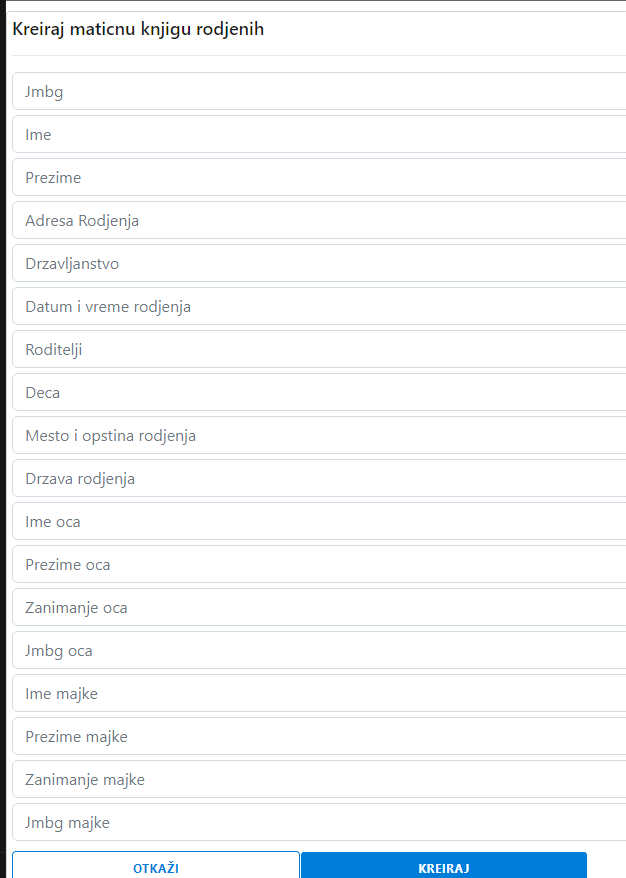
На слици се приказује профил матичара. Профил приказује све податке о матичару, укључујући његово име, презиме, корисничко име и контакт информације. Додатно, на профилу се могу приказати и други релевантни подаци о матичару, као што су јмбг, адреса рођења као и подаци о његовим родитељима. Профил матичара пружа детаљан преглед у информације о особи која обавља матичарске задатке.



На слици се приказује страница за преглед матичне књиге рођених. Матичар има могућност прегледа свих до сада унесених матичних књига рођених. На овој страници, он може извршити следеће акције:

* Преглед матичних књига: Матичар може прегледати све постојеће матичне књиге рођених, укључујући информације о именима, презименима, датумима рођења и осталим детаљима.
* Брисање матичних књига: Матичар има могућност брисања постојећих матичних књига рођених ако је то потребно.
* Ажурирање матичних књига: Матичар може ажурирати информације у вези са постојећим матичним књигама рођених, као што су име, презиме, датум рођења и друге податке.
* Претрага матичних књига: Матичар може претраживати матичне књиге рођених на основу различитих критеријума као што су ЈМБГ, адреса, име и презиме корисника. Ово му помаже да лако пронађе конкретне записе у књизи.
* Додавање нове матичне књиге: Матичар има могућност додавања нове матичне књиге рођених преко навбара, где је доступан одговарајући линк.

Ова страница омогућава матичару ефикасан приступ и манипулацију свим подацима у вези са матичним књигама рођених.



На овој слици приказана је форма за додавање матичне књиге рођених. Матичар треба да попуни следећа поља са својим подацима:

* ЈМБГ (Јединствени матични број грађана)
* Име
* Презиме
* Адреса рођења
* Име оца
* Презиме оца
* Занимање оца
* Име мајке
* Презиме мајке
* Занимање мајке
* Држављанство
* ЈМБГ оца
* ЈМБГ мајке
* Место и општина рођења
* Држава рођења
* Датум и време рођења
* Број родитеља
* Број деце

Постоје два дугмета:

"Откажи" дугме: Омогућава матичару да поништи процес додавања и врати се на претходну страницу без чувања података.

"Креирај" дугме: Када матичар попуни сва потребна поља, ово дугме служи за креирање нове матичне књиге рођених на основу унетих података.

Овај образац омогућава матичару једноставан и ефикасан начин да дода нову матичну књигу рођених у систем.

* **ЗАКЛЉУЧАК**

У овом раду је представљена апликација за информациони систем матичара и демонстриран њен рад. Приказано решење пружа матичарским службама могућност ефикасног и прецизног вођења евиденције о рођењима, венчањима и смртима, смањујући време и напор у уносу и претрази података.

Систем нуди неколико значајних функционалности, укључујући брз приступ и претрагу матичних књига, могућност ажурирања и брисања података, као и употребу филтера за прецизнију претрагу. Истовремено, омогућава сигурност и заштиту података, као и лаку управу информацијама о корисницима.

Ово решење доприноси повећању ефикасности и транспарентности рада матичарских служби, пружајући брз приступ и унос података, смањујући потребу за папирном документацијом и олакшавајући комуникацију са грађанима. Иако ова апликација испуњава већину потреба матичарских служби, увек се може усавршити како би била још успешнија и кориснија.

За даље усавршавање и побољшање апликације за матичаре, могу се применити следеће приступи:

* Расширење функционалности: Могу се додати додатни модули и функционалности који ће додатно унапредити рад матичара и корисничко искуство. Неки од могућих примера укључују:
* Увођење могућности електронске обраде и склапања уговора за венчања и друге службене акте. Ово би смањило потребу за физичким присуством грађана и уштедело време.
* Имплементација система обавештавања и алатки за комуникацију са грађанима, као што су обавештења путем е-поште или смс порука, што би учинило комуникацију бржом и ефикаснијом.
* Додавање функције генерисања извештаја и статистика, што би помогло у анализи података и праћењу радне ефикасности матичара и квалитета услуга.
* Интерфејс и корисничко искуство: Важно је посветити пажњу унапређењу интерфејса и корисничког искуства апликације. Ергономски дизајн и интуитивно коришћење су кључни за успешну примену и прихватање система од стране корисника. Ово укључује:
* Прилагођавање интерфејса истим темама и дизајну као што су матичне књиге и службени документи, што би додатно олакшало коришћење и асоцијацију са постојећим процесима.
* Усавршавање навигације и организације информација у оквиру апликације, како би корисници лако и брзо дошли до жељеног циља.

Укратко, развијена апликација за матичаре омогућава ефикасно и прецизно вођење евиденције и представља значајан корак у напредаку матичарских служби. Ипак, додатне функционалности и побољшања могу бити имплементирана како би се још боље задовољиле потребе корисника и повећала употребљивост система.

**Литература**

[1] "Увод у веб апликације: дефиниција и основни концепти." TechTarget.com

<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>

[2] "Шта је MySQL?" Oracle.com. Приступљено са

<https://www.oracle.com/mysql/what-is-mysql/>

[3] "Angular архитектура и структура пројекта." Angular.io. Приступљено са

<https://angular.io/guide/architecture>

[4] "Преглед упутства за развој." Angular.io. Приступљено са

<https://angular.io/guide/developer-guide-overview>

[5] "Шта је Java Spring Boot?" Microsoft Azure. Приступљено са

<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-java-spring-boot/#:~:text=What%20is%20Spring%20Boot%3F,create%20microservices%20and%20web%20apps>

[6] "Закон о матичним књигама." Paragraf.rs. Приступљено са

<https://www.paragraf.rs/propisi/zakon_o_maticnim_knjigama.html>

[7] "Министарство државне управе и локалне самоуправе - Матичне књиге." Министарство државне управе и локалне самоуправе. Приступљено са

<https://mduls.gov.rs/registri/maticne-knjige/?script=lat>

[8] "Шта је Bootstrap?" TechTarget.com. Приступљено са

<https://www.techtarget.com/whatis/definition/bootstrap>

[9] "Увод у Bootstrap." Hostinger.com. Приступљено са

<https://www.hostinger.com/tutorials/what-is-bootstrap/>

[10] "UML Дијаграми." SmartDraw.com. Приступљено са

<https://www.smartdraw.com/uml-diagram/>

[11] "Шта је ElasticSearch?" Sematext.com. Приступљено са

<https://sematext.com/guides/elasticsearch/>