Seminarski rad za predmet Tehnologije i sistemi E-uprave

Digitalizovan sistem sudstva

Jovan Stojaković

Fakultet tehničkih nauka , Univerzitet u Novom Sadu

Trg Dositeja Obradovića 6 21000 Novi Sad

[stojakovic.sr55.2020@uns.ac.rs](mailto:stojakovic.sr55.2020@uns.ac.rs)

Sažetak:

Jedan od ključnih koraka za napredovanje jedne države u ekonomskom i političkom smislu svakako prestavlja digitalizacija pravnog sistema te drzave. Uvođenje digitalnih tehnologija u sudstvo ima potencijal da značajno poboljša opšti sistem pravosuđa, smanji mogućnost nastanka grešaka usled neprosleđivanja dokumenata, gubljenja istih i olakša obavljanje posla sudija. Ovaj seminarski rad ima za cilj da pruži uvid u jednu od mogućnosti za postizanje digitalizacije sudstva.

Ključne reči:

Sudstvo, sudija, digitalizacija, presuda, komunikacija.

Uvod:

U današnje vreme, suočavamo se sa izazovima u pravosudnom sistemu koji su posledica nedovoljne organizacije i nedostatka efikasne saradnje među institucijama koje učestvuju u njemu. Kao rezultat toga, pojedinci koji su odgovorni za krivična dela često pronalaze načine da izbegnu kaznu.

Količina papira, slanje dokumenata između različitih institucija i mogućnost njihovog gubljenja zbog ljudske greške predstavljaju ozbiljnu prepreku za brzu i ispravnu proceduru u pravnom sistemju.

Da bi se sve ovo izbeglo, došlo se na ideju digitalizacije četiri sistema :

* Policije
* Tužilaštva
* Sudstva
* Građanskog servisa za informisanje i slanja žalbi

Digitalizacijom ova četiri sistema, povećala bi se efikasnost rada sistema, smanjila bi se mogućnost stvaranja greške i cela pravna procedura bi brže tekla. Takođe, doprinelo bi se boljoj informisanosti građanstva u vezi pravnih postupaka pokrenutih protiv njih i pružila bi se mogućnost građanima da šalju prijave na druge građane, sa mogućnosti da prijava bude anonimna.

Cilj ovog izlaganja je da prikaže kako se može pristupiti digitalizaciji ova četiri sistema i uspostaviti veza između njih.

Srodna istraživanja:

Da bi se lakše predstavilo kako ovaj sistem finkcioniše, možemo ga uporediti sa sistemom zakazivanja treninga u teretani.

Da bi trener zakazao trening u datom terminu potrebno je da u istom tom terminu ne postoji već prethodno zakazan trening. Kada se utvrdi da je termin slobodan trening se zakazuje. Članovima teretane stiže obavestenje da je trening zakazan. Kada dodje vreme održavanja treninga, trener u zavisnosti da li se neko pojavio konstatuje da li je trening održan. Ako je trening održan, trener može da ostavi komentar treninga koji članovi teretane mogu da pročitaju.

Neka u ovom slučaju, zakazivanje treninga u određenom terminu od strane trenera, bude isto što je zakazivanje suđena od strane sudije. Kao što članu teretane stiže obaveštenje o zakazanom trenigu tako građaninu stiže poziv za sud. Ako se član teretane ne pojavi na treningu, trener kontatuje da trening nije održan, isto tako će biti za suđenje. Kada se trening održi, trener daje komentar trening, u ovom slučaju bi to bilo pisanje izveštaja o parnici od strane sudije.

Korišćena tehnologija i alati za izradu

Za implementaciju poslovne logike (Backend-a) aplikacije, koristili smo Java programski jezik i Spring Boot alat, dok smo za razvoj koristili IntelliJ IDE okruženje.

Za kreiranje korisničkog interfejsa (Frontend-a) aplikacije, koristili smo TypeScript i Angular framework.

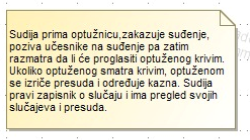
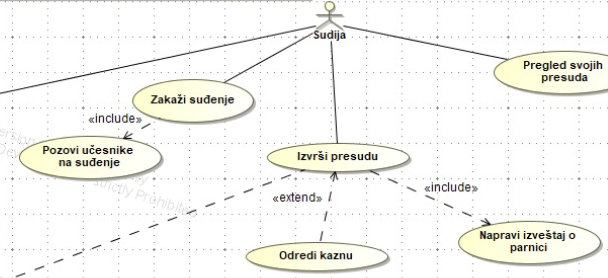
Kao sistem za upravljanje bazom podataka, koristili smo MySQL.

Takođe, važno je napomenuti da smo za sistem Jednog logovanja (Single Sign-on) koristili Google-ovu platformu za prijavu. Takođe, ova aplikacija je deo veće mikroservisne aplikacije koja obuhvata četiri sistema: policiju, tužilaštvo, sudstvo i građanski servis. Svaki od ovih sistema ima svoj korisnički interfejs i bazu podataka. Sve ove komponente su implementirane u Docker kontejnerima radi bolje organizacije i upravljanja.

Funkcionalnosti aplikacije

Funkcionalnosti ove aplikacije jesu da sudijia ima uvid u sve podignute optužnice, ima mogućnost da zakaže suđenje za svaku optužnicu u određenom terminu, kao i da izrekne presudu i odredi odgovarajću kaznu optuženom ako nakon razmatranja odredi da je optuženi kriv. Kada sudija zakaže suđenje, građaninu stiže obaveštenje u terminu održavanja suđenja. Nakon suđenja sudija pravi zapisnik o slučaju. Optuženom stiže obavestenje o izrečenoj presudi i kazni. Sudija ima uvid u svoje slučajeve i presude.

Slika1a

 Slika1b

Slika1a I Slika 1b: Dijagram slučajeva korišćenja sudije– prikazuje sve funkcionalnosti koje su omogućene jednom sudiji. Slikom čoveka prikazani su učesnici, oblačićima su prikazane funkcionalnosti koje sudija ima, dok je u desnom delu prikazan opšti tok funkcionalnosti.

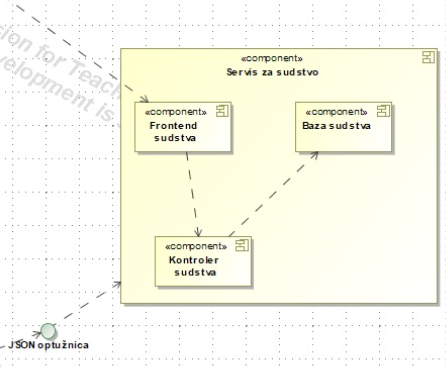
Dizajn i implementacija aplikacije

Ova aplikacija se sastoji od tri glavne komponente:

* Poslovna logika (Backend)
* Korisnički interfejs (Frontend)
* Baza podataka

Da bi se ovaj sistem vizuelno predstavio, koristili smo dijagram komponenti koji prikazuje ove tri glavne komponente i njihovu međusobnu interakciju.

Slika2



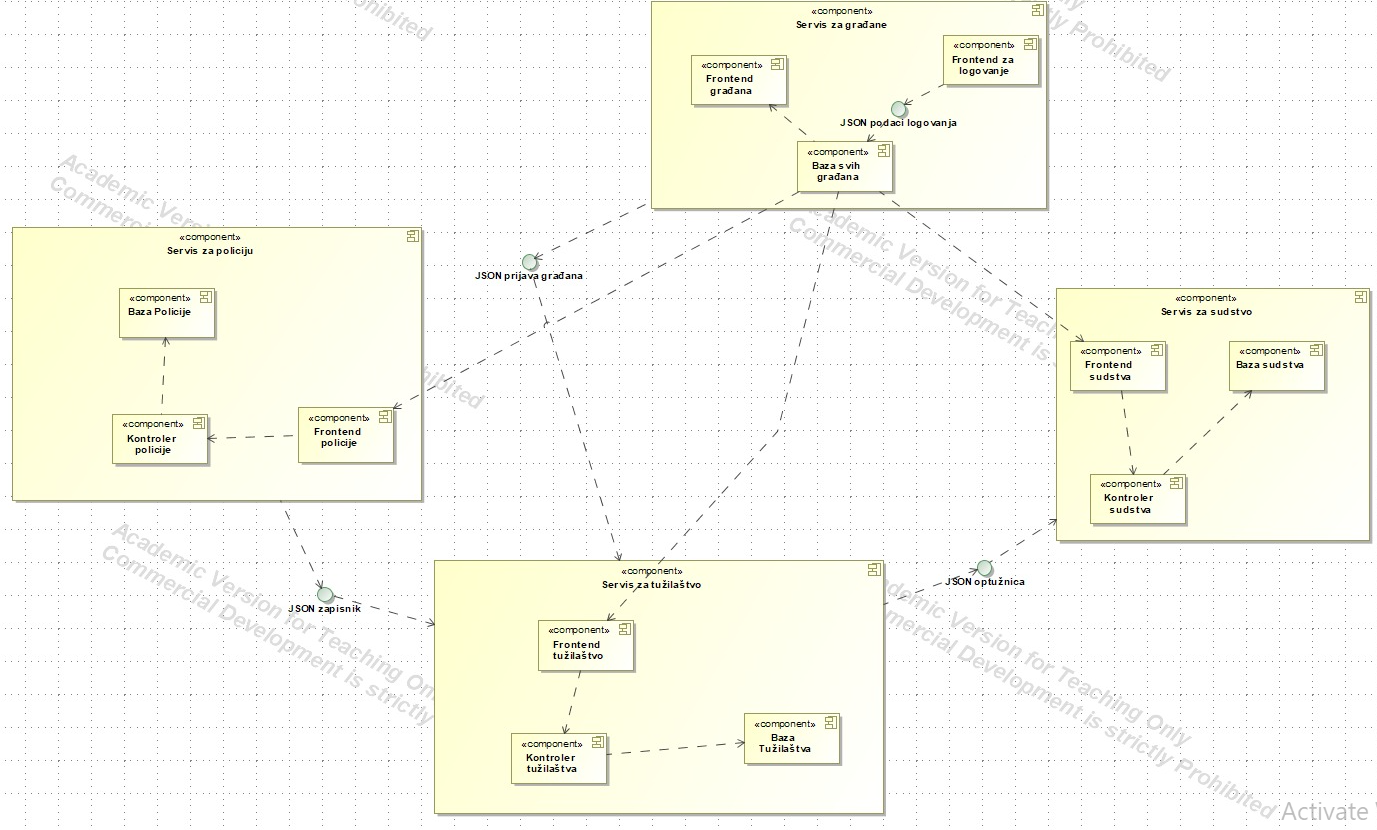
Slika2: Dijagram komponenti sudstva– prikazuje komponente naše aplikacije, kao i naziv i tip ulaznog podatka naše aplikacije (JSON optužnica).

Ova aplikacija je deo šireg pravnog sistema koji obuhvata četiri mikroservisne aplikacije:

* Policija
* Tužilaštvo
* Sud
* Gradjanske usluge

Na slici prikazanoj ispod može se videti kako je to predstavljeno dijagramom komponenti:

Slika3

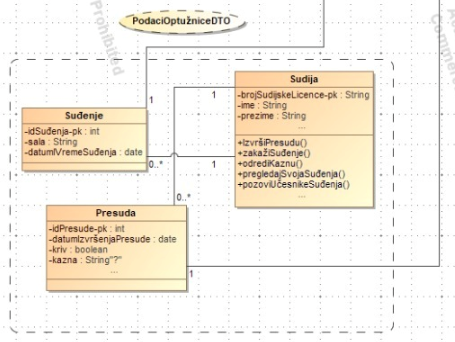


Slika3: dijagram komponenti kompletnog mikroservisnog sistema. Svaki od ovih sistema je predstavljen velikim pravougaonikom, dok su manje celine unutar svakog sistema predstavljene manjim pravougaonicima. Kružići na dijagramu prikazuju podatke koji se šalju i primaju između sistema.

Za izradu aplikacije sudstva, unutar poslovne logike potrebno je definisati sledeće pojmove(objekte):

* Sudija– predstavlja samog sudiju. Od atributa ovaj element sadrži: broj sudijske licence koji je ujedno i njegov jedinstveni identifikator, ime i prezime sudije.
* Suđenje– sudija ima uvid u podignute optužnice i za svaku optužnicu može da zakaže suđenje, na suđenju se utvrđuje da li je osuđeni kriv i u skladu sa tom odlukom izriče se presuda i određuje eventualna kazna . Od atributa sadrži: jedinstveni identifikator suđenja,datum i vreme suđenja i salu u kojoj će se održati suđenje.
* Presuda – Sudija donosi presudu u skladu sa tim šta je utvrđeno na suđenju (da li je osuđenik kriv ili ne). Od atributa ovaj element sadrži : jedinstveni identifikator presude, datum izvršenja presude, da li je kriv ili nije (boolean), i kaznu (ako je kriv).

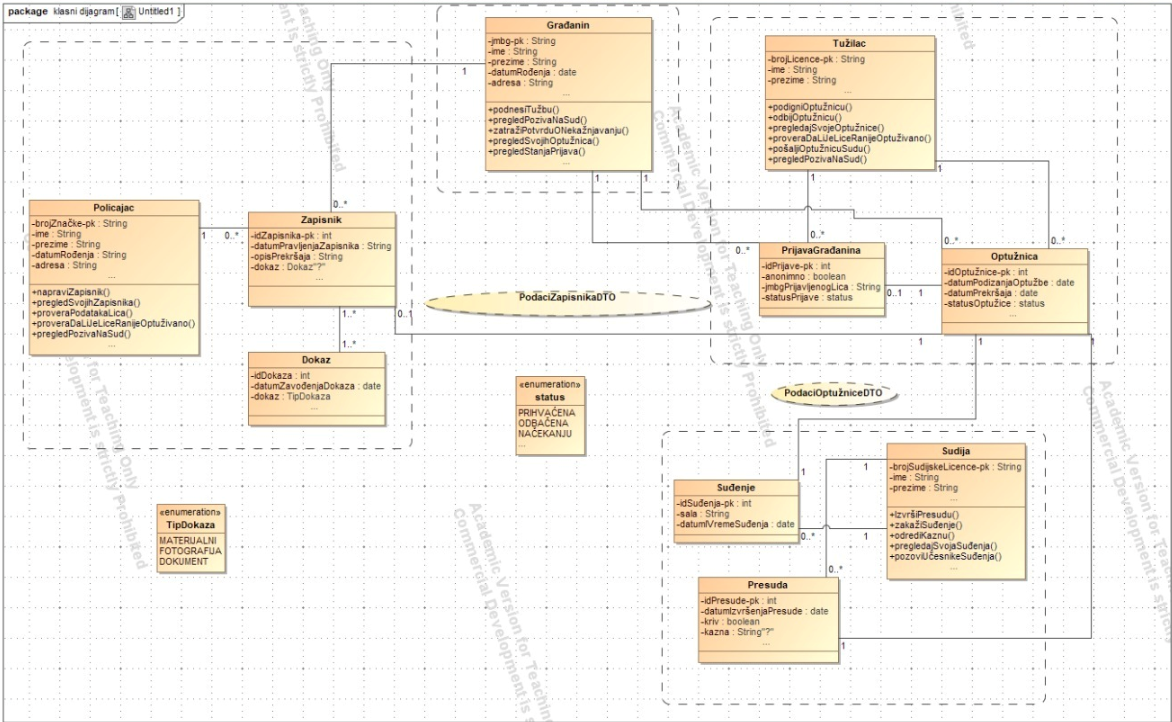
Slika4



Slika4: klasni dijagram aplikacije sudstva– prikazuje sve objekte(pravougaonici) koji su potrebni da bi se napravila aplikacija, kao i ulazni i izlazni podatak naše aplikacije(elipsa)

Na slici ispod, pomoću klasnog dijagrama, prikazana je cela konstrukcija pravnog sistema kao jedne celine. U opisu slike je bliže opisano značenje svake stvari na slici.

Slika5



Slika5: klasni dijagram cele mikroservisne aplikacije – prikazuje sve potrebne objekte za izradu celokupnog pravnog sistema kao i podatke koji se prosleđuju između servisa. Svaki sistem prikazan je isprekidanom linijom, pravougaonicima su prikazani objekti dok su elipsama prikazani podaci koji se prosleđuju između servisa. Unutar objekata, u gornjem odeljku navedeni su atributi koji opisuju svaki objekat, dok su ispod njih navedene funkcionalnosti koje su dostupne tom objektu.

Implementacija

Unutar klase Sudija navedeni su svi atributi (broj sudijske licence, ime i prezime ) ime same klase - Sudija, kao i ime tabele unutar koje se čuvaju podaci – sudije :

Slika6



Slika6: Prikaz klase Sudija unutar našeg koda.

Unutar klase Presuda navedeni su svi atributi id presude, ime sudije, id optužnice, kriv (da li jeste ili nije)) ime same klase - Presuda, kao i ime tabele unutar koje se čuvaju podaci – presude:

Slika7

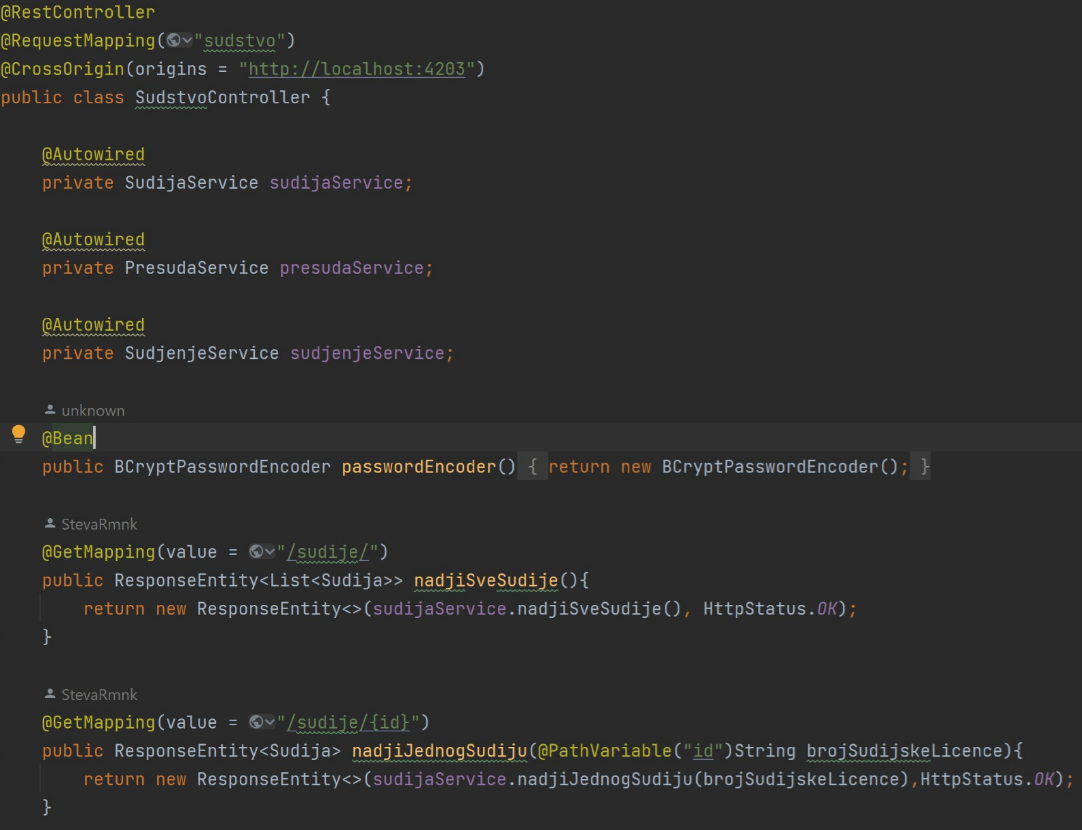


Slika7: Prikaz klase Presuda unutar našeg koda.

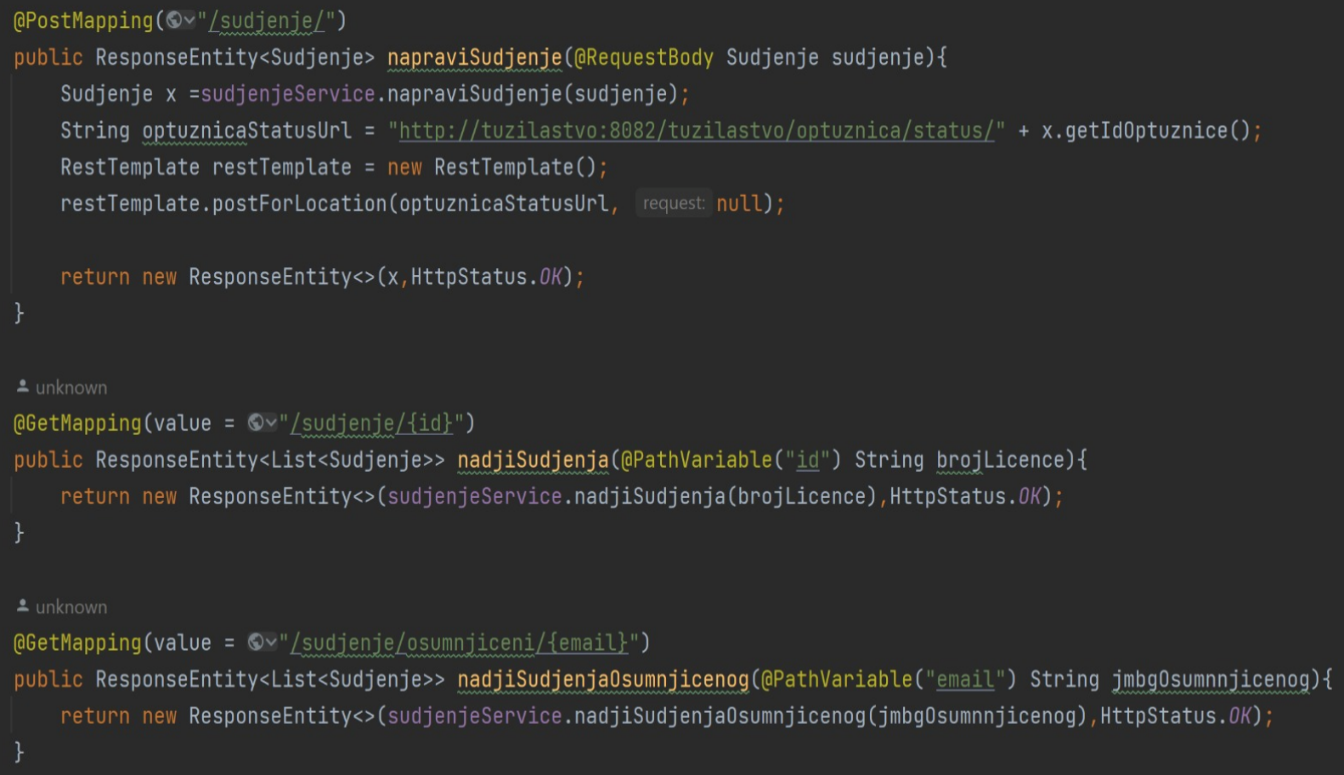
Da bi se omogućilo sudiji da izvrši bilo koju od gore pomenutih funkcionalnosti potrebno je kreirati kontroler unutar koga je potrebno definisati odgovarajuće endpointe – krajnje tačke URL-a, na čiji se odlazak pomoću pretraživača pokreće odgovarajuća poslovna logika.

Kontroler sudstva izgleda kao na slikama 8,9,10 i 11:

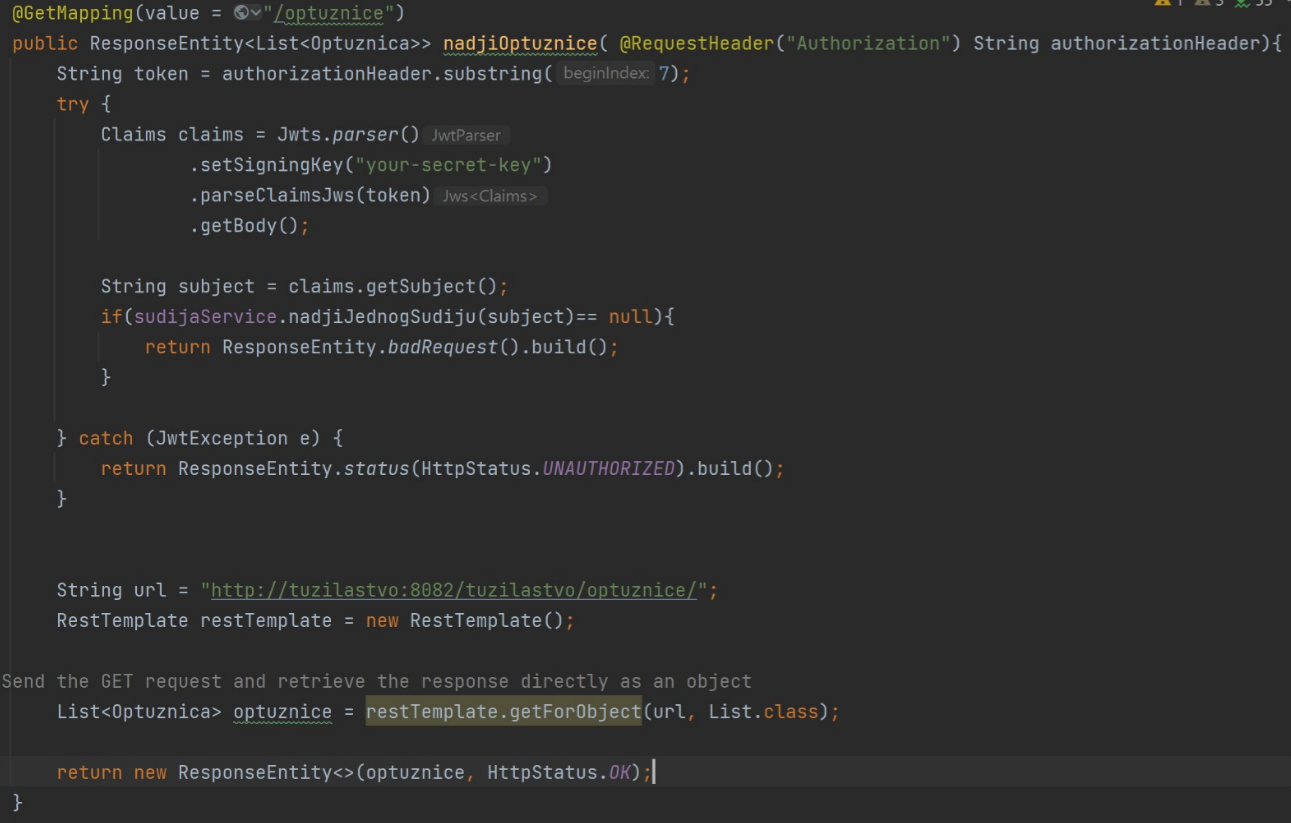
Slika8



Slika9



Slika10



Slika11



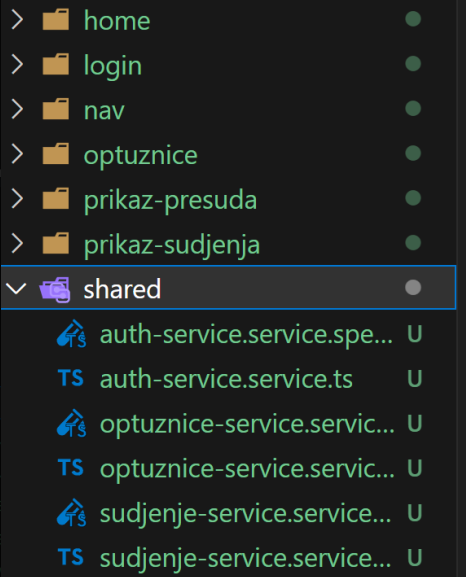
Slike 8, 9, 10 I 11 prikazuju niz metoda koje se aktiviraju kada sudija pristupi određenoj putanji (vrednosti URL -a kojima sudija treba da pristupi, nalaze se unutar zagrada pored anotacija). Svaka od ovih metoda ima svoje ime koje jasno opisuje funkcionalnost koju ta metoda obavlja.

Sve ove metode i funkcionalnosti koje su prikazane gube svoj značaj ukoliko sudija nema na raspolaganju korisnički interfejs koji mu omogućava interaktivno pristupanje tim metodama. Da bismo to postigli, neophodno je razviti Angular aplikaciju koja će sudiji omogućiti da na jednostavan način koristi ove funkcionalnosti.

Koristeći Angular, možemo kreirati korisnički interfejs koji se sastoji od različitih komponenti. Komponente predstavljaju vizuelne elemente koji se mogu koristiti na više mesta u aplikaciji, kao što su stranice, navigacioni meniji i drugi elementi. Ovaj pristup omogućava ponovno iskorišćavanje vizuelnih elemenata širom aplikacije.Da bismo ostvarili komunikaciju između korisničkog interfejsa (Frontenda) i poslovne logike (Backenda), koristimo servise. Servisi su komponente koje sadrže metode za komunikaciju. Kroz ove servise, korisnički interfejs može slati zahteve ka Backendu, dobijati odgovore i efikasno komunicirati sa poslovnom logikom.

U ovom slučaju, potrebno je kreirati komponente koje su prikazane na slici ispod

Slika12



Slika 12: komponente koje je potrebno kreirati

Svaka od ovih komponenti, unutar sebe ima 4 fajla: HTML fajl, CSS fajl i dva TypeScript fajla.

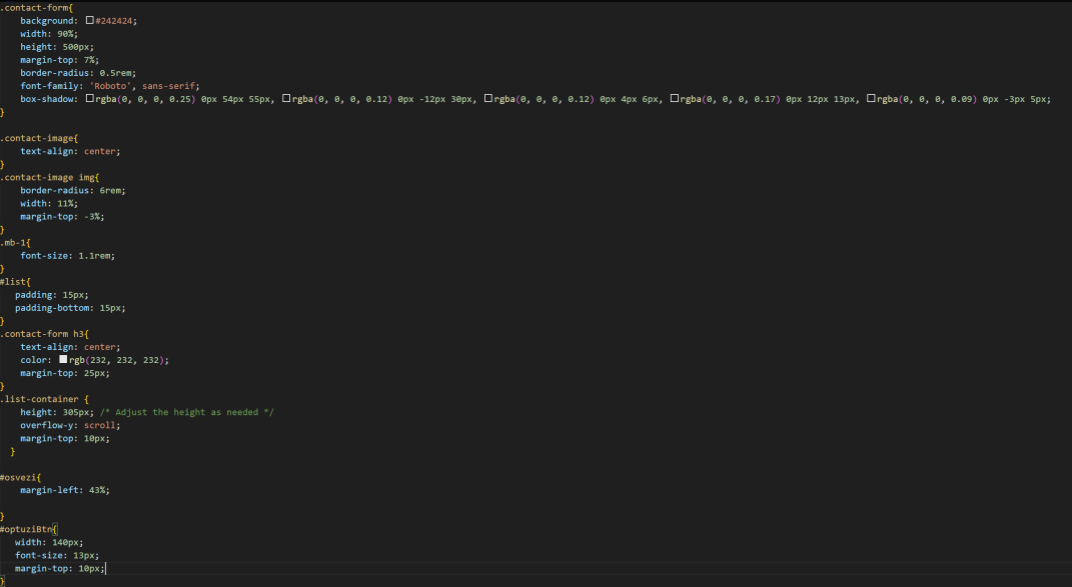
Unutar HTML fajla, piše se HTML kod, koji daje osnovnu strukturu komponente.Na slici ispod dat je prikaz HTML koda jedne od komponenti.

Slika13



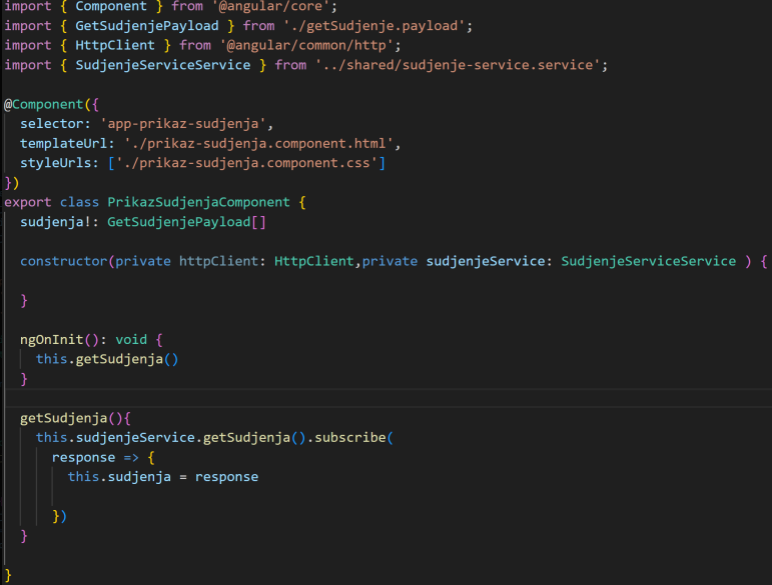
Slika13: prikaz HTML koda komponente – prikaz suđenja

Unutar CSS fajla piše se CSS kod koji će dodatno stilizovati komponente.Na slici ispod, prikazan je primer izgleda CSS fajla komponente prikaz suđenja.

Slika14Slika14: Izgled CSS fajla komponente – prikaz suđenja

Unutar TypeScript fajla, piše se dodatna potrebna logika i pozivaju se metode iz servisa koji su kreirani. Na slici ispod, dat je prikaz sadržaja TypeScript fajl.

Slika15



Slika15: Izlged TypeScript fajla komponente “prikaz suđenja” unutar koje se poziva metoda iz servisa koja prikazuje sva suđenja.

Unutar fajlova servisa, vrši se pristupanje endpointovima koji su definisani u kontrolerima java koda, omogućavajući time komunikaciju između različitih delova aplikacije sudstva.

Slika16



Slika 16: Izgled TypeScript fajla za servis “prikaz I zakazivanje suđenja”.

Demonstracija rada aplikacije

Kada korisnik pokrene aplikaciju i otvori web pretraživač, prikazue mu se prva starnice sudstva koja izgleda ovako:

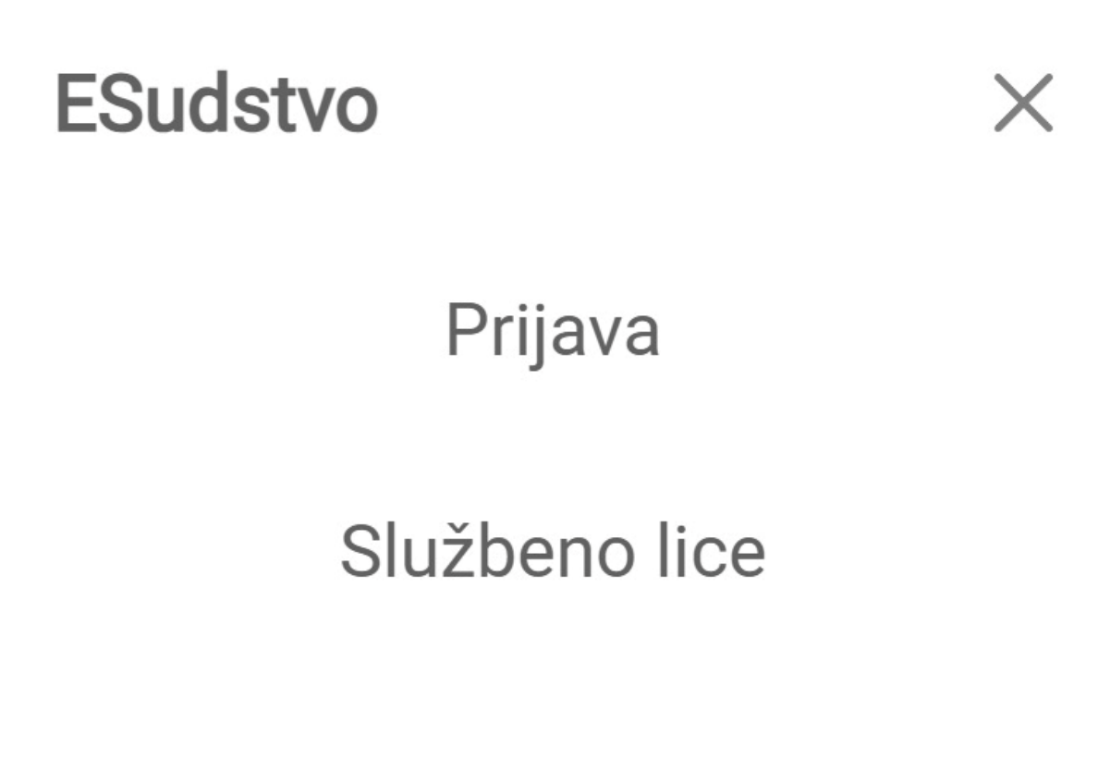
Slika17



Slika17: Izgled početne stranice sudstva.

U gornjem levom uglu, pored ESudstvo logotipa, postoji ikona navigacionog menija koja se prikazuje kao tri horizontalne linije. Kada korisnik pritisne tu ikonu, otvara se navigacioni meni koji sadrži opcije "Prijava" i "Službeno lice".

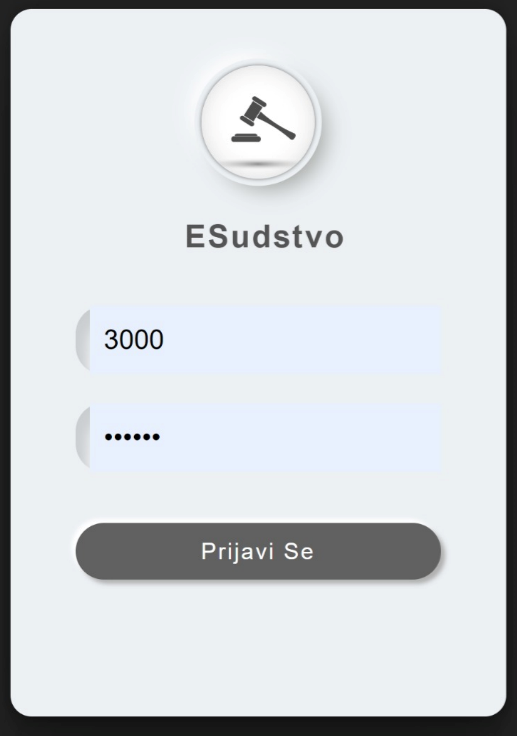
Slika18



Slika18: Izgled navigacionog menija.

Odabirom dugmeta „Službeno lice“, otvara se nova stranica za logovanje sudije na sistem. Ta stranica izgleda ovako:

Slika19

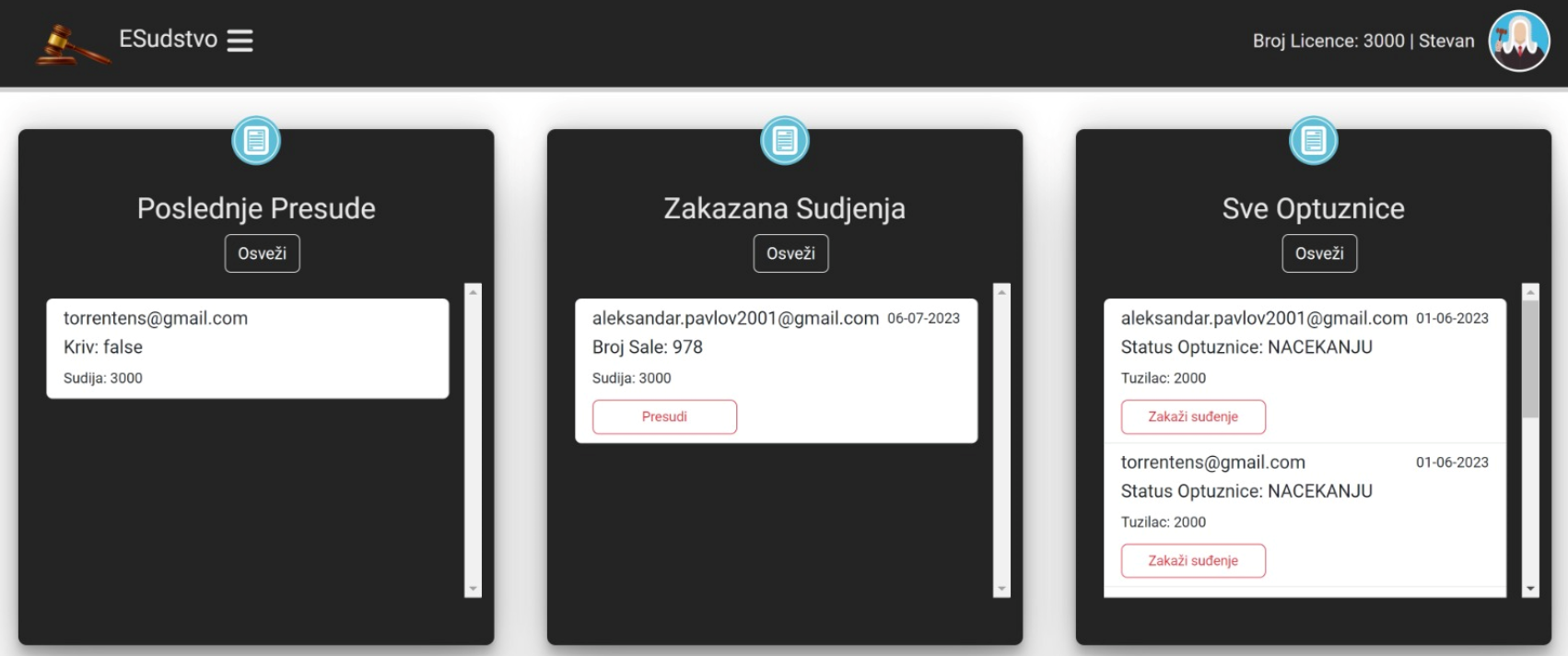


Slika19: Izgled stranice za logovanje sudije.

Unosom validnog broja licence tužioca,lozinke i klikom na dugme „Prijavi Se“ korisnik se prebacuje na novu stranicu na kojoj su prikazana tri prozora:

* Poslednje presude
* Zakazana suđenja
* Sve optužnice

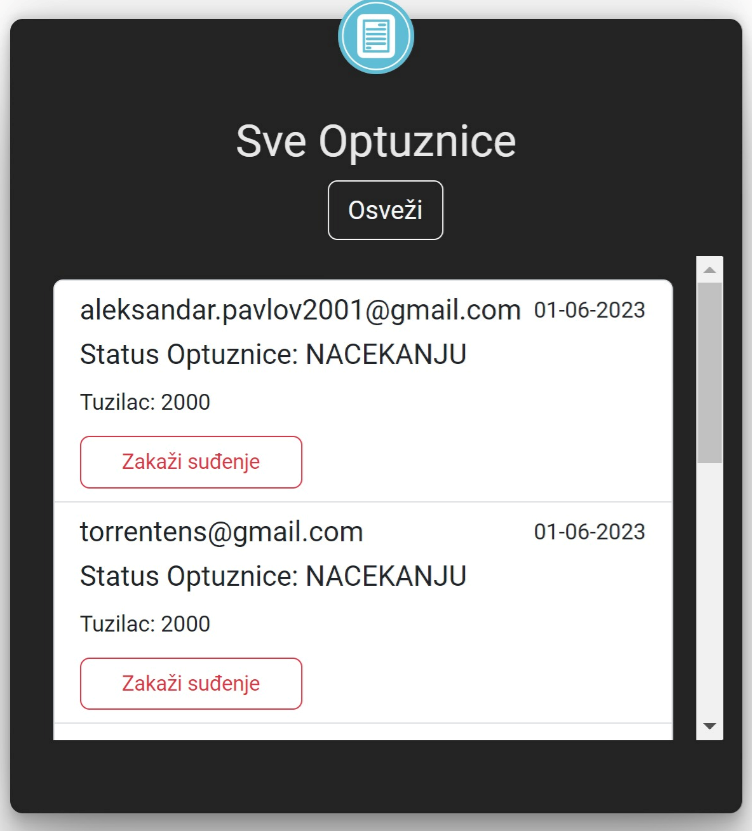
Slika20



Slika 20: izgled stranice nakon uspešnog logovanja

Kada sudija želi da zakaže neko suđenje, on to može uraditi klikom na dugme “Zakaži suđenje”, tada se podignuta optužnica prebacuje u polje u kom se nalaze zakazana suđenja.

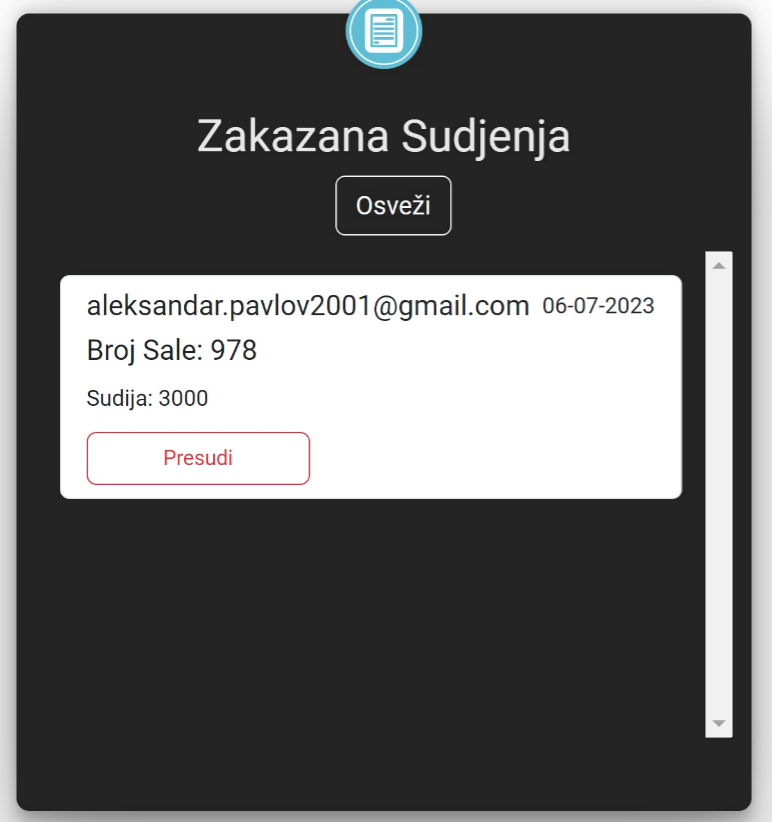
Slika21



Slika211: Prikaz podignutih optužnica i dugmeta “Zakaži suđenje”

Svako zakazano suđenje u sebi sadrži dugme “Presudi”, klikom na njega sudija donosi presudu.

Slika22



Slika22: Prikaz zakazanog suđenja i dugmeta “Presudi”.

Polje ”Poslednje presude” u sebi sadrzi sve presude ulogovanog sudije, svaka presuda sadrži email, broj sale,Broj sudijske licence i da li je optuženi kriv.

Slika23



Slika23: Prikaz poslednih presuda ulogovanog sudije.

Zaključak:

Digitalizacija pravnog sistema, uključujući sisteme policije, tužilaštva, sudstva i građanskog servisa, predstavlja ključni korak ka efikasnijem i pravednijem pravosudnom sistemu. Eliminisanje nepotrebnog papirnog rada, unapređenje komunikacije između institucija i brže obavljanje pravnih procedura donose brojne prednosti. Digitalizacija omogućava bolju informisanost građanstva, smanjuje mogućnost grešaka i doprinosi efikasnosti pravosudnog sistema. Digitalizacija sudstva ima ključnu ulogu u modernizaciji pravnog sistema i ona bi doprinela bržem I pravednijem suđenju. Ovaj rad objašnjava proces dizajniranja, implementacije i funkcionalnosti ovakve aplikacije, pružajući uvid u način na koji bi ona bila konstruisana i način na koji bi ista funkcionisala.

Reference:

https://www.w3schools.com/angular/

https://www.baeldung.com/

https://cloud.google.com/architecture/identity/single-sign-on