

# Web aplikacija za unos, rješavanje i ocjenjivanje Sudoku igara

---

**Matijević, Maja**

**Master's thesis / Diplomski rad**

**2021**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:479479>

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

*Download date / Datum preuzimanja:* **2024-07-13**

*Repository / Repozitorij:*

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



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

**Sveučilišni studij**

**WEB APLIKACIJA ZA UNOS, RJEŠAVANJE I  
OCJENJIVANJE SUDOKU IGARA**

**Diplomski rad**

**Maja Matijević**

**Osijek, 2021.**

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
<b>2. PREGLED POSTOJEĆIH RJEŠENJA.....</b>	<b>2</b>
<b>2.1. Kratka povijest.....</b>	<b>2</b>
<b>2.2. Pravila Sudoku igre .....</b>	<b>2</b>
<b>2.3. Postojeća rješenja.....</b>	<b>3</b>
2.3.1. Web aplikacije .....	3
2.3.2. Algoritmi za rješavanje Sudokua .....	8
<b>3. MODEL SUDOKU WEB APLIKACIJE.....</b>	<b>10</b>
<b>3.1. Zahtjevi na web aplikaciju .....</b>	<b>10</b>
3.1.1. Prijava i registracija .....	13
3.1.2. Rješavanje i provjera Sudoku igara .....	13
3.1.3. Unos novih igara i objavljivanje.....	13
3.1.4. Dodatne mogućnosti .....	14
<b>3.2. Korištene tehnologije .....</b>	<b>15</b>
3.2.1. Korisničko sučelje web aplikacije .....	15
3.2.2. Node.js.....	15
3.2.3. Baza podataka i Heroku platforma .....	16
<b>4. IMPLEMENTACIJA SUDOKU WEB APLIKACIJE .....</b>	<b>17</b>
<b>4.1. Sučelja izradene aplikacije <i>Sudoku Network</i> .....</b>	<b>18</b>
4.1.1. Početna stranica i rješavanje Sudokua .....	18
4.1.2. Unos nove igre.....	22
<b>4.2. Specifičnost aplikacije za registriranog korisnika .....</b>	<b>23</b>
4.2.1. Registracija i prijava .....	23
4.2.2. Objavljivanje i ocjenjivanje Sudokua .....	26
4.2.3. Korisnički profil .....	27
<b>4.3. Mogućnosti unaprjeđenja izradene aplikacije .....</b>	<b>27</b>
<b>5. ZAKLJUČAK.....</b>	<b>29</b>
<b>LITERATURA .....</b>	<b>30</b>
<b>SAŽETAK.....</b>	<b>32</b>
<b>ABSTRACT .....</b>	<b>32</b>

**ŽIVOTOPIS..... 33**

**PRILOZI..... 34**

## 1. UVOD

Već dugi niz godina ljudi različitih dobi iz cijelog svijeta vole rješavati Sudoku zagonetke. Bez obzira na to što se u današnje vrijeme razvilo mnogo programa koji imaju spobnost efikasno i brzo bez pogreške riješiti i najteži Sudoku, ljudi i dalje pronalaze uživanje i kognitivni izazov u ručnom rješavanju. Iz tog razloga s razvojem interneta nastao je velik broj aplikacija koje sadrže najrazličitije Sudoku igre dostupne ljubiteljima istih.

Jedan od velikih izazova koje tehnologija i automatizacija ipak bolje rješava je pronalazak Sudokua s jedinstvenim rješenjem. Stoga su poneke aplikacije donijele korisnicima mogućnost takve provjere kako bi znali može li se Sudoku koji su sami smislili riješiti na samo jedan način, to jest je li u potpunosti ispravan.

U sklopu ovog rada izradit će se web aplikacija koja će biti kombinacija stranice za rješavanje Sudoku zagonetki te provjeru jedinstvenosti rješenja osobno unesenih Sudokua. Osim toga, u web aplikaciju će se uvesti brojne druge mogućnosti kao što je registracija korisnika, ocjenjivanje, objavljivanje jedinstvenih igara. Bit će izrađeno jednostavno i pregledno sučelje sa svim funkcijama koje bi jedan Sudoku korisnik mogao zatrebati i na taj će se način objediniti postojeće web aplikacije na tu temu u jednu funkcionalnu i poboljšanu cjelinu. Koristit će se jednostavni algoritmi i moderne tehnologije za razvoj web aplikacija.

Rad je organiziran tako da su nakon uvoda, u drugom poglavlju analizirana i opisane postojeće web aplikacije za rješavanje Sudoku igara. Obavljena je usporedba tih aplikacija prema značajkama koje će biti implementirane u sklopu ovog diplomskog rada. U trećem poglavlju opisani su zahtjevi na aplikaciju, a u četvrtom je detaljnije objašnjeno konkretno programsko rješenje i implementacija aplikacije. Peto poglavlje donosi zaključak.

## 2. PREGLED POSTOJEĆIH RJEŠENJA

Od davne prošlosti do modernog doba sve su se brže razvijale brojne zagonetke, igre brojevima i mozgalice. Jedna od danas najpopularnijih i najučestalijih jest Sudoku, koji se u Japanu igra već oko 1000 godina. Sudoku je vrsta matematičke zagonetke koja zahtijeva korištenje logike za uspješno rješavanje. Iako u današnje vrijeme gotovo svako računalo može bez problema riješiti i najteže Sudoku zagonetke, ljudi ih i dalje često rješavaju, bilo iz čiste zabave ili radi poboljšanja načina razmišljanja.

### 2.1. Kratka povijest

Sudoku je u modernom obliku započeo 1783. godine kada je Leonhard Euler sastavio latinski kvadrat dimenzija 9x9. Ne zna se točno kada je Sudoku prvi put službeno objavljen, ali prva sudoku zagonetka za koju je to poznato objavljena je 1979. godine u njujorškom časopisu Dell magazine pod nazivom Number place [1]. Osamdesetih godina u Japanu je osnovana izdavačka kuća Nikoli koja je ubrzo postala vrlo popularna i osim klasičnih Sudoku zagonetki, objavljuje i različite varijante iste, ali i brojne druge mozgalice. Kao još jedna važna prekretnica u popularnosti Sudoku zagonetki navodi se računalni program koji je kreirao sudac iz Hong Konga, Wayne Gould, 1997. godine, a koji je ostvario generiranje mnoštva unikatnih Sudoku zagonetki koje su svoje mjesto pronašle u velikom broju časopisa. Na taj način Gould je znatno pridonio popularizaciji ove danas široko rasprostranjene mozgalice.

Prateći razvoj tehnologije i Interneta, Sudoku zagonetke pojavile su se u brojnim web i mobilnim aplikacijama i danas ih rješavaju milijuni ljudi diljem svijeta, a varijante aplikacija su brojne i neprestano nastaju nove i bolje.

### 2.2. Pravila Sudoku igre

Sudoku je vrsta zagonetke u kojoj je potrebno brojevima popuniti prazne ćelije unutar mreže držeći se zadanih pravila. Mreže mogu biti različitih veličina, ovisno o vrsti Sudoku zagonetke, ali standardni i pri tom najzastupljeniji oblik jest 9x9 Sudoku. U tom slučaju, u polja se unose brojevi od 1 do 9. Osnovno pravilo u klasičnoj Sudoku zagonetki jest da se pri popunjavanju polja brojevima, niti jedan broj ne smije ponavljati više od jednom u jednom retku, stupcu i odjeljku, koji je u ovom slučaju, veličine 3x3. Na početku igre zadani su početni brojevi, a njih u klasičnoj igri treba biti najmanje 17, s obzirom na to da je 17 najmanji otkriveni broj početnih brojeva s kojim Sudoku može imati jedinstveno rješenje. Ostale brojeve popunjava osoba koja rješava Sudoku. Pravi Sudoku može imati samo jedno rješenje.

Osim klasičnog 9x9, postoje i manje i veće verzije, primjerice 6x6 ili 12x12, a osim toga postoje i Sudoku s izmijenjenim, to jest dodanim pravilima. Primjerice, X-Sudoku sadrži sva osnovna pravila za redak, stupac i odjeljak, a dodano mu je još jedno pravilo: brojevi se ne smiju ponavljati ni na dijagonalama mreže. S druge strane, postoje Sudoku zagonetke bez dodanih pravila, ali s izmijenjenim granicama odjeljaka u kojima oni nisu pravokutni. To su samo neki od brojnih primjera različitih Sudokua.

## 2.3. Postojeća rješenja

Postoji velik broj aplikacija čija je glavna svrha rješavanje Sudoku zagonetki. U ovom će radu biti fokus na web aplikacijama, s obzirom na to da će kao konačno rješenje ovog rada biti izrađena web aplikacija za unos, rješavanje i ocjenjivanje Sudoku igara. Stoga je za početak bilo potrebno pronaći i proučiti već postojeće web aplikacije i algoritme kako bi se definirali konačni ciljevi ovog rada.

Pronađen je velik broj web aplikacija koje pružaju samo osnovne funkcionalnosti kao što je rješavanje Sudokua te generiranje novih. Takvi jednostavni primjeri neće se uzimati u obzir.

### 2.3.1. Web aplikacije

Istraživanjem već postojećih web aplikacija za rješavanje i pomoć pri rješavanju Sudoku zagonetki, pronađen je velik broj aplikacija. U daljnjem tekstu ovog potpoglavlja kratko će se analizirati aplikacije s funkcionalnostima koje su motivirale ovaj diplomski rad i njegovu izvedbu. Analizom postojećih web aplikacija stvara se bolji pogled na osnovne karakteristike koje bi rezultat ovog rada trebao donijeti, što omogućuje lakše isticanje nedostataka i potencijalnih poboljšanja nove aplikacije na zadanu temu. Nakon ove analize bit će detaljno postavljeni zahtjevi uzevši u obzir postojeće aplikacije.

U tablici **Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**2.1. navedene su karakteristike koje će biti dio zahtjeva na aplikaciju koja će biti izrađena u okviru ovog rada. Vidljivo je da niti jedna pronađena web aplikacija ne sadrži sve potrebne karakteristike, no svaka sadrži barem jednu i stoga su navedene u tablici. Velik broj postojećih aplikacija nije naveden u tablici zbog svoje trivijalnosti.

Tablica 2.1. Usporedba postojećih web aplikacija prema relevantnim karakteristikama

Web aplikacije Značajke	Sudoku solver [2]	Sudoku helper [3]	Sudoku solver [4]	Sudoku solver [5]	Sudoku wiki [6]	Sudoku exchange [7]
jasno i pregledno sučelje	da	da	da	da	ne	da
različite vrste Sudoku zagonetki	da	ne	da	ne	da	samo po težinama
provjera jedinstvenog rješenja	ne	djelomično	da	ne	djelomično	djelomično
automatski prikaz greške (poklapanja brojeva i sl.)	ne	da	da	da	da	da
odabir već postojeće igre	ne	da	da	da	da	da
objava igre s jedinstvenim rješenjem kako bi ju mogli riješavati drugi korisnici	ne	ne	ne	ne	ne	ne
prijava korisnika u sustav, spremanje prethodnih igara	ne	ne	ne	ne	ne	ne
filtriranje postojećih igara prema težini i vrsti	ne	ne	ne	ne	ne	djelomično

Proučivši tablicu, vidljivo je da web aplikacija [2] *Sudoku spoiler* sadrži najmanji broj funkcionalnosti iz tablice, ali je navedena kao osnovni primjer Sudoku aplikacije za automatski pronalazak rješenja. Navedena aplikacija sadrži veći broj različitih vrsta Sudoku igara, različitih pravila te različitih veličina. Glavna je funkcionalnost pronalazak rješenja igre za upisane brojeve, ali ne i korisničko rješavanje zagonetki. Sučelje je jednostavno, a klikom na tipku *Solve* prikazuje se do 10 mogućih rješenja, čak i ako nije upisan niti jedan broj (slika 2.1).



1	2	3	4	5	6	7	8	9
7	8	9	1	2	3	4	5	6
4	5	6	7	8	9	1	2	3
3	1	2	8	4	5	9	6	7
6	9	7	3	1	2	8	4	5
8	4	5	6	9	7	3	1	2
2	3	1	5	7	4	6	9	8
9	6	8	2	3	1	5	7	4
5	7	4	9	6	8	2	3	1

Solution 1 of 10

Next

Unsolved

Reset

Slika 2.1. Sudoku spoiler – prikaz nekoliko rješenja za sva prazna polja

U web aplikaciji [3], istaknuta značajka koju gotovo svaka web aplikacija za rješavanje Sudokua treba imati jest automatski prikaz pogreški koji je prikazan na slici 2.2. te nemogućnost prikaza cjelokupnog rješenja ukoliko ne postoji jedinstveno rješenje. Osim toga, web aplikacija je prilično jednostavna jer je moguće isključivo rješavanje klasičnog 9x9 Sudokua.

## Sudoku Helper

by [J.P. Grossman](#)

9	8								
8						7			
7									
6	5		2	4					
5						9			
4									
3	5								
2									
1					6				
	a	b	c	d	e	f	g	h	i

b6 and b3 cannot both be 5

---

Next Hint    Reset Hints

Solve Puzzle

Reset Board    Erase Board

Print Board    Print State

---

[Instructions](#)   [Strategies](#)

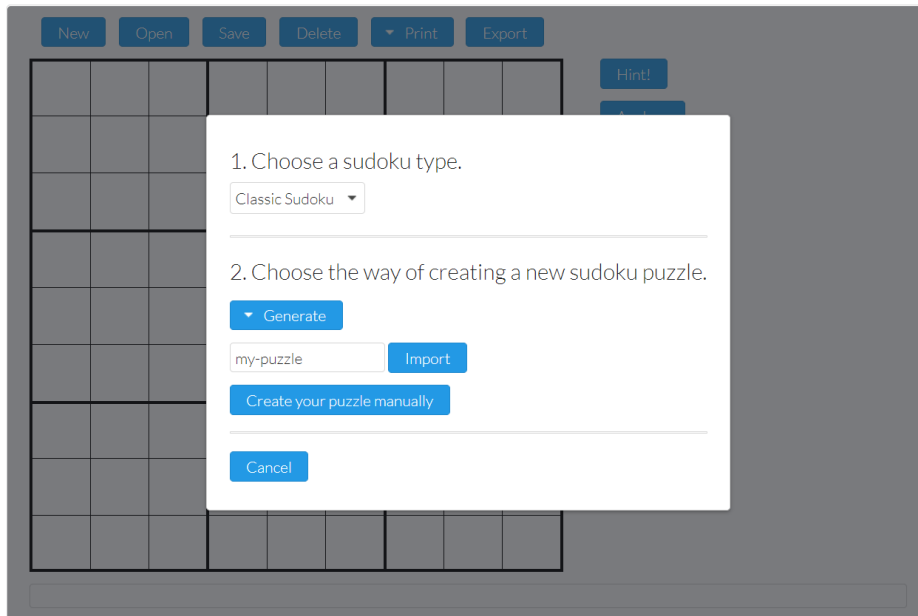
---

Select a puzzle:

---- www.scanraid.com/sudoku.htm ----

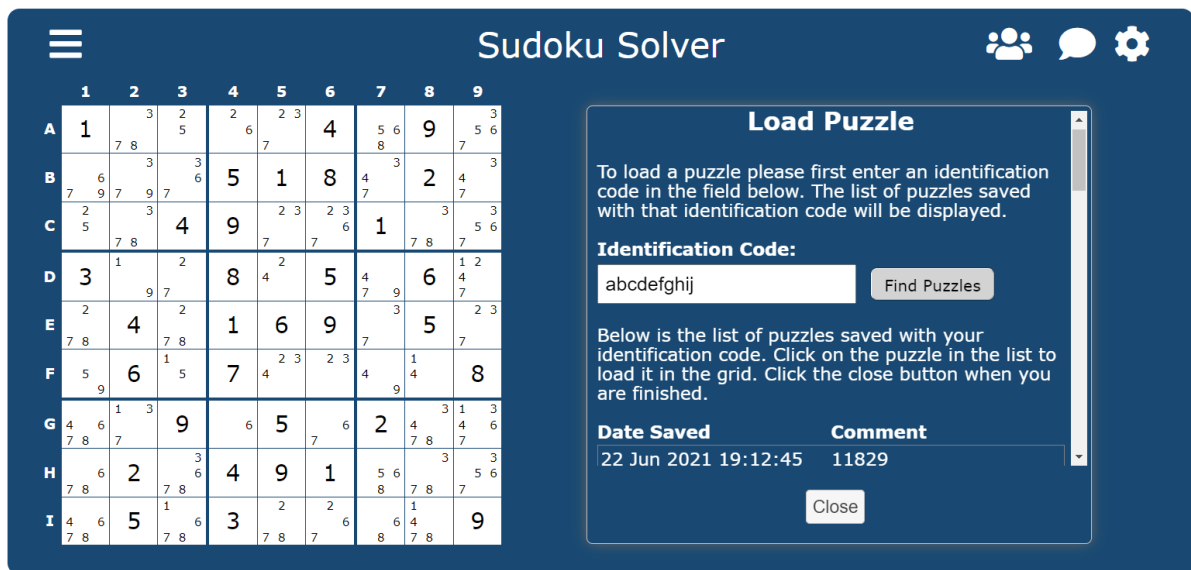
Slika 2.2. Sudoku Helper – prikaz pogreške

Sljedeća web aplikacija [4] slična je prethodnoj, no posebnijom ju čini mogućnost otvaranja prijašnje spremljenih zagonetki koristeći naziv koji je unešen pri spremanju (slika 2.3.). Također dopušta prikaz mogućih kandidata brojeva u svakoj ćeliji, što korisniku olakšava rješavanje. Provjera točnosti rješenja može se uključiti ili isključiti, a aplikacija sadrži i provjeru postoji li jedinstveno rješenje.



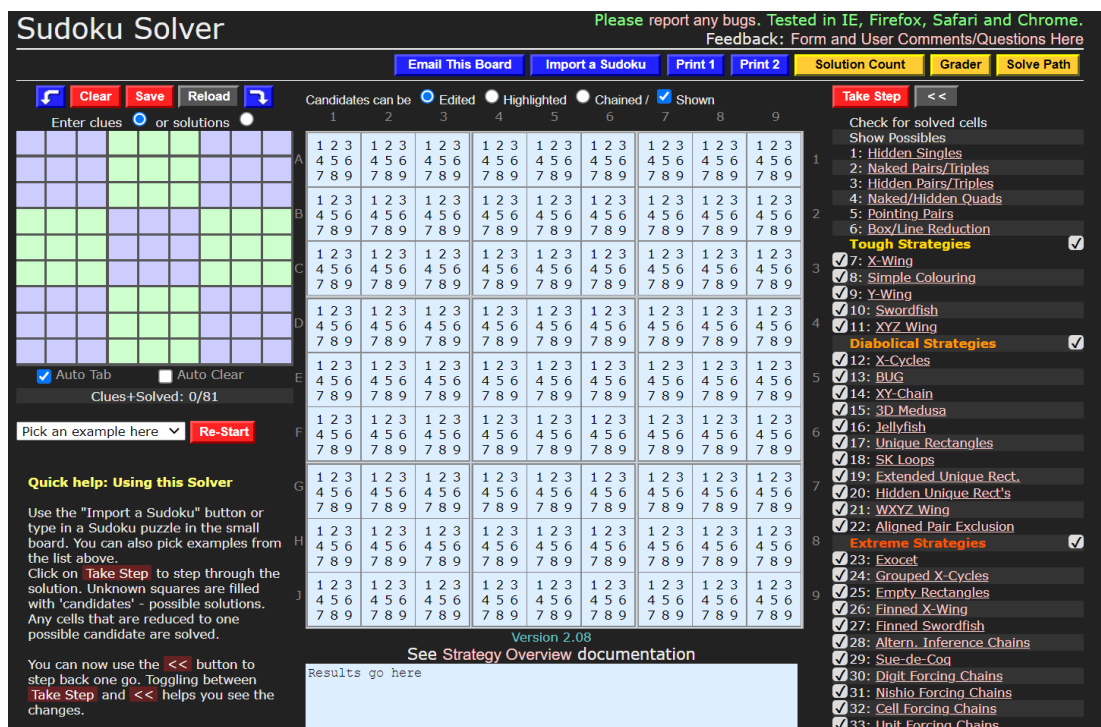
Slika 2.3. Sudoku Solver – učitavanje spremljene ili nove igre

Web aplikacija [5] sadrži većinu karakteristika već navedenih aplikacija, a osim toga omogućava učitavanje postojećih igara pomoću identifikacijskog koda, što se može vidjeti na slici 2.4.. Moguće je spremiti trenutnu igru te ju kasnije pomoću spomenutog koda učitati. Identifikacijski kod ima trajanje od 60 dana. Spremanje Sudokua korisnicima može znatno poboljšati iskustvo korištenja aplikacije jer se mogu vratiti na već pronađenu igru i ponovno ju riješiti.



Slika 2.4. Sudoku Solver – učitavanje igre spremljene pomoću identifikacijskog koda

Web aplikacija [6] sastoji se od brojnih korisnih značajki, no uključuje vrlo nepregledno sučelje (slika 2.5.). Unutar Sudoku sučelja moguće je odabrati različite opcije, krenuti od praznih polja ili odabrati već postojeće primjere te proučiti mnogobrojne strategije rješavanja Sudokua i uz pomoć ih primijeniti. Za uspješno korištenje i iskorištavanje svih mogućnosti, potrebno je pročitati sve upute i testirati funkcionalnost jer stranica nije sasvim intuitivna i mogućnosti nisu jasne na prvi pogled.



Slika 2.5. Sudoku Wiki

Posljednje promatrana aplikacija [7] sastoji se od mnogih već viđenih značajki, a novost je to što u postavkama sadrži brojne opcije kao što je tamni način rada, uključivanje timera, prikaz animacije nakon uspješno riješene zagonetke i slično. Sve te značajke poboljšavaju korisnički doživljaj aplikacije i doprinose ugodnijem korištenju.

### 2.3.2. Algoritmi za rješavanje Sudokua

Kako bi web aplikacija za rješavanje Sudokua bila uspješno kreirana, potrebno je koristiti algoritam koji će rješavati i provjeravati unikatnost rješenja.

Važno je spomenuti najnaprednije i najmodernije algoritme koji donose veću efikasnost i brzinu rješavanja Sudoku igara. Iako se najčešće i dalje koriste metode iscrpnog pretraživanja (engl. *exhaustive search, brute-force*) metode u kojima se isprobavaju različite kombinacije dok program ne dođe do točnog rješenja, postoje brojne optimizacije istih ili sasvim novi algoritmi.

Donald Knuth je u svom radu predložio algoritam *Dancing Links* [8] koji optimizira njegov *backtracking X* algoritam i pronalazi sva rješenja za točan problem pokrića (engl. *exact cover problem*). To je problem koji govori da, ukoliko je zadana binarna matrica koja se sastoji isključivo od nula i jedinica, potrebno je pronaći skup redaka koji sadrže točno jednu jedinicu u svakom stupcu. Knuthov algoritam pokazao se kao iznimno učinkovit i brz te se zbog toga često koristi.

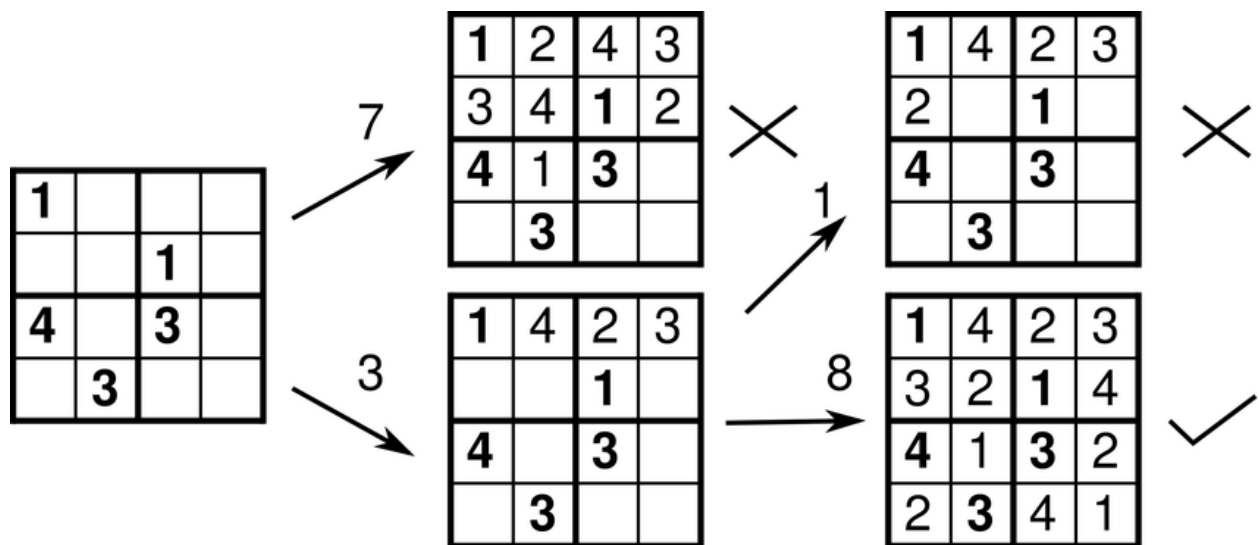
Nadalje, postoje algoritmi koji na neki način imitiraju ljudsko ponašanje pri rješavanju, koristeći pri tom poznate tehnike rješavanja krenuvši od najjednostavnijih prema težima kako bi emulirali ljudsko ponašanje, a jedan takav je *Pencil-and-paper algorithm* [9]. Osim toga, za rješavanje Sudokua istražena je i mogućnost korištenja genetskih algoritama koji imitiraju koncept prirodne selekcije te se pokazala prilično uspješnom, osobito za teže probleme [10].

Postoje i razne optimizacijske tehnike, a posebno zanimljiva je optimizacija kolonije mravi (engl. *ant-colony optimisation*) [11] koja se služi načinom na koji mravi pomoću feromona pronalaze najkraći put do neke destinacije.

Cilj ovog diplomskog rada nije pronaći novi algoritam za što efikasnije rješavanje Sudokua, stoga će se koristiti već postojeći *backtracking* algoritam uz male preinake kako bi bio efikasniji i kako bi se preskočili neki nepotrebni koraci. Taj će se algoritam koristiti jer je jednostavan i u svakom slučaju dolazi do rješenja, a njegova efikasnost ne ovisi o težini Sudokua. Također, uz dodatne funkcije može biti iznimno brz. *Backtracking* [12] metoda je jedan od najpoznatijih i potencijalno najlakših, ali vremenski najzahtjevnijih algoritama. *Backtracking* algoritam funkcionira na način da prolazi redom po poljima te pokušava pogoditi točan broj u svakom polju, a ukoliko dođe do

polja u kojem se mogući broj ne podudara s ostatkom rješenja ili se ne može upisati više niti jedan broj, vraća se te ispravlja kako bi došao do točnog rješenja. Upravo zbog toga je dobio naziv *backtracking*. Na slici 2.6. prikazan je jednostavan primjer *backtracking* algoritma. Brojevi na strjelicama označavaju broj koraka koji su poduzeti bez grananja i nailaska na greške. Vidljivo je da nakon određenog broja unešenih podataka može doći do greške, a ukoliko se to dogodi isprobavaju se druge vrijednosti kako bi u konačnici dobili točno rješenje.

U sklopu ovog diplomskog rada u obični *backtracking* algoritam bit će dodane funkcije koje pri svakom koraku prvo traži ćelije koje sadrže gole samce (eng. *naked singles*) i skrivene samce (eng. *hidden singles*). Goli samci predstavljaju ćelije u kojima postoji samo jedan mogući broj za unijeti, a koji je istovremeno i jedini kandidat za tu ćeliju. Ćelija koja sadrži skrivenog samca, također može sadržavati isključivo taj broj, ali u ćeliji postoje i drugi kandidati, stoga je samac skriven. Osim tih funkcija, u web aplikaciji ovog diplomskog rada pri svakom upisivanju nove vrijednosti neće se upisivati u prvu praznu ćeliju, već u ćeliju s najmanje mogućnosti. Pomoću tih funkcija znatno se može skratiti vrijeme izvršavanja i smanjiti grananje petlji.



Slika 2.6. Primjer *backtracking* algoritma, preuzeto s [13]

### **3. MODEL SUDOKU WEB APLIKACIJE**

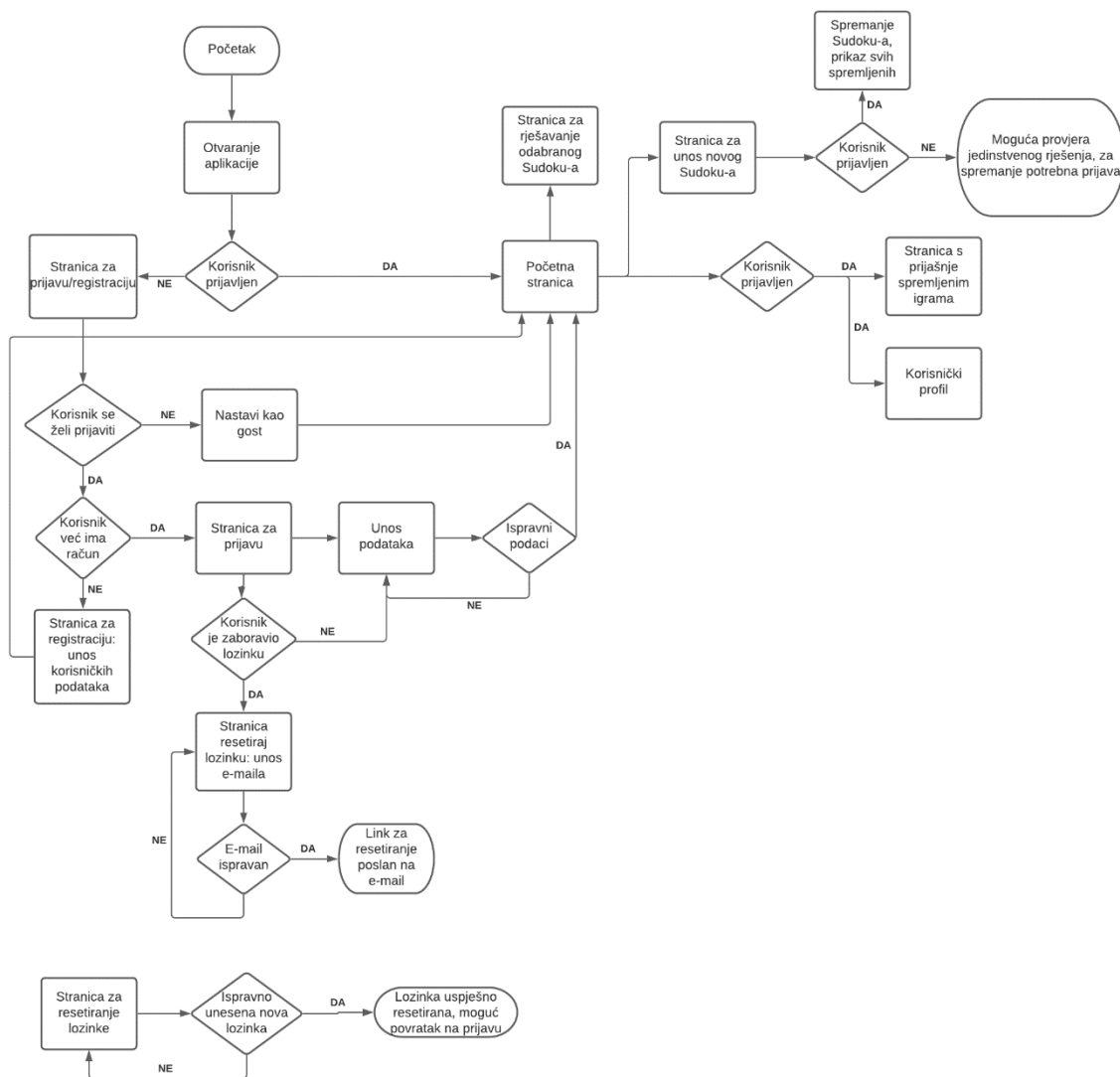
Zadatak ovog diplomskog rada bio je napraviti web aplikaciju za rješavanje, provjeru i objavljivanje Sudoku zagonetki. Aplikacija treba sadržavati što više korisnih značajki već postojećih web aplikacija i pri tome biti kombinacija njihovih najboljih svojstava, uz preinake i dodatke koji nisu pronađeni niti u jednoj postojećoj aplikaciji.

Većina postojećih web aplikacija nije se oslanjala na korisničku prijavu, a spremanje igara vršilo se, ako uopće, putem kodova koji su imali određeno trajanje. Ovaj bi diplomski rad trebao donijeti mogućnost prijave korisnika, ali i rješavanje Sudokua bez prijave. Prijavljeni bi korisnik imao više mogućnosti unutar aplikacije od korisnika „gosta“.

U poglavlju koje slijedi bit će detaljnije opisani zahtjevi na spomenutu web aplikaciju te ukratko objašnjene tehnologije koje će se koristiti pri izradi web aplikacije. Na temelju zahtjeva na aplikaciju bit će poznat daljnji smjer razvoja iste.

#### **3.1. Zahtjevi na web aplikaciju**

Zahtjevi na web aplikaciju definiraju se prema željenim značajkama iz prethodne analize postojećih rješenja. Potrebno je kombinirati određene funkcionalnosti i napraviti plan i model kojeg se treba pridržavati tijekom izrade web aplikacije kako bi došlo do željenog rezultata. U svrhu toga, izrađen je dijagram toka koji se nalazi na slici 3.1. Dijagram toka okvirno predstavlja tijek kojim se kreće korisnik kroz aplikaciju te mogućnosti koje mu se otvaraju prilikom prijave u sustav.



Slika 3.1. Dijagram toka aplikacije

Izrada aplikacije odvijat će se u više iteracija, a svaka od njih unijet će nove važne značajke i funkcionalnosti u aplikaciju.

Kao što je prikazano na dijagramu, pri ulasku na stranicu svakom korisniku aplikacija treba omogućiti odabir između prijave, registracije ili nastavka na početnu stranicu bez prijave. Na taj način se korisnik, ukoliko želi, može prijaviti u aplikaciju i prije nego što vidi početnu stranicu. Prva iteracija izrade aplikacije treba se sastojati isključivo od kreiranja prethodno spomenutog početnog prikaza te sučelja za prijavu i registraciju korisnika. Nakon što je taj dio gotov i testiran, slijedi izrada početne stranice. Dolaskom na početnu stranicu, svaki korisnik, bilo prijavljen ili ne, trebao bi vidjeti listu Sudokua s jedinstvenim rješenjem koji su spremni za rješavanje. Taj prikaz najvažniji je dio aplikacije i potrebno je prikazati sve objavljene Sudokue koji se nalaze unutar baze podataka web aplikacije zajedno s njihovim relevantnim informacijama. Izrada početne

stranice trebala bi rezultirati tipkom za preusmjerenje na stranicu za dodavanje novog Sudokua te listom postojećih Sudokua koji su dostupni za rješavanje. Nadalje, korisniku treba pružiti mogućnost filtriranja prikazanih Sudokua prema kriterijima kao što su vrsta, težina i vrijeme objavljivanja, a svaki Sudoku mora sadržavati pregledan prikaz osnovnih podataka o sebi. Takav prikaz služi za pojednostavljenje pretraživanja i pronalaska željenih Sudoku zagonetki pojedinih korisnika. Kada početna stranica bude spremna, potrebno je omogućiti klik na tipku *Riješi* koja se mora nalaziti na svakoj kartici pojedinog Sudokua čime korisnika aplikacija treba odvesti na stranicu odabranog Sudokua. Na toj stranici, potrebno je prikazati Sudoku mrežu s prethodno unesenim početnim brojevima. Preostala polja služit će za korisničko ispunjavanje istih, a nakon ispunjavanja, ako je prijavljen u aplikaciju, korisniku treba omogućiti da spremi svoje vrijeme rješavanja i procijeni težinu rješavanog Sudokua. Rješavanje Sudoku zagonetki jedna je od glavnih značajki web aplikacije koja se izrađuje u sklopu ovog diplomskog rada s obzirom na to da niti jedna aplikacija ovakvog tipa nije potpuna bez mogućnosti rješavanja zagonetki koje se nalaze unutar same aplikacije. Najbitniji dijelovi ovog dijela implementacije web aplikacije su algoritam za rješavanje Sudokua te prikaz Sudoku mreže i kretanje kroz istu. Osim navedenih odlika web aplikacije, potrebno je dodati funkciju usmjerenja korisnika na stranicu za dodavanje novog Sudokua, a na kojoj može ispunjavati praznu Sudoku mrežu te provjeravati je li pronašao jedinstveni Sudoku. Za provjeru se treba koristiti isti algoritam kao i za rješavanje, to jest provjeru korisničkog rješenja. Također, važno je omogućiti dodavanje različitih vrsta Sudoku zagonetki kako bi iskustvo rješavanja i dodavanja Sudokua bilo dinamičnije. Nakon pronalaska jedinstvenog Sudokua, korisniku treba ponuditi spremanje Sudokua, pri čemu ponovno treba biti prijavljen kako bi mogao doći do navedene mogućnosti. Spremanjem bi u isto vrijeme korisnik mogao i objaviti Sudoku, što znači da bi se on našao na početnoj stranici i bio vidljiv ostalim korisnicima (prijavljenim i neprijavljenim) te bi ga svi mogli rješavati. Nakon implementacije svih navedenih mogućnosti, preostaje dodavanje korisničkog profila, stranice s prikazom njegovih spremljenih Sudokua te tipke za odjavljivanje. Sve to treba se nalaziti unutar zaglavlja aplikacije i to isključivo nakon što je korisnik prijavljen u aplikaciju.

Kao dodatna funkcija za lakše korištenje na različitim uređajima i općenito, pri rješavanju i dodavanju novog Sudokua, treba biti omogućen unos brojeva putem posebnih tipki s brojevima te pri rješavanju unos mogućih kandidata u ćelije (eng. *pencil marks*). Također, za vrijeme upisivanja brojeva bitno je označiti pogreške u mreži.



### **3.1.1. Prijava i registracija**

Obavezan zahtjev na aplikaciju prema kojem se ona znatno razlikuje od većine postojećih web aplikacija jest mogućnost registracije i prijave korisnika. U aplikaciji treba biti omogućena registracija novih korisnika te nakon toga prijava s istim podacima. Koristit će se autentikacija korisnika kako bi se povećala sigurnost i očuvali korisnički podaci.

Prijava i registracija igraju važnu ulogu pri čuvanju korisničkih Sudokua kako bi svi mogli biti na jednom mjestu i kako bi ih korisnik mogao rješavati i dijeliti s drugima. Upravo zbog toga su važan dio web aplikacije za rješavanje Sudokua. Uvođenjem korisničke prijave izbjegava se spremanje Sudokua pomoću različitih identifikacijskih kodova koje bi korisnik morao pamtit i zapisivati, a pri tome ne bi imao popis svih svojih igara na jednom mjestu.

### **3.1.2. Rješavanje i provjera Sudoku igara**

Kao što je vidljivo iz poglavlja 2.2. ovog rada, postojeće web aplikacije koje su istražene uglavnom su bila rješenja koja sadrže mogućnost automatskog rješavanja Sudokua na temelju nekoliko unesenih vrijednosti te provjera postoji li jedinstveno rješenje. Neke od tih aplikacija uvele su učitavanje već postojećih primjeraka igara, ali je to uglavnom bilo pomoću identifikacijskog koda ili je postojao ograničen broj igara te nije bilo omogućeno korisničko dodavanje u tu listu. U okviru ovog rada izrađuje se aplikacija koja će objediniti provjeru Sudokua za jedinstvenim rješenjem, to jest kreiranje novog Sudokua i samo rješavanje tog Sudokua ili nekih drugih već postojećih. Korisnik će moći rješavati Sudoku igre bez obzira je li registriran ili nije, a objaviti i dodati ih u svoj profil će ih moći, dakako, samo prijavljen korisnik. Neregistriran korisnik dobit će mogućnost unosa nove igre i provjere ima li ona jedinstveno rješenje, a ukoliko ju želi objaviti ili spremiti za privatno rješavanje, bit će potrebna registracija. Na taj način bit će moguće dijeljenje igara među korisnicima. Motivacija za dodavanje obje mogućnosti, točnije i rješavanja i dodavanja novih igara, nastala je iz nedostatka postojećih web aplikacija s istim mogućnostima.

### **3.1.3. Unos novih igara i objavljivanje**

Pri unosu nove igre, to jest provjere ima li ona jedinstveno rješenje, neregistrirani korisnik moći će samo obaviti spomenutu provjeru. Ukoliko je pronašao Sudoku s jedinstvenim rješenjem i ima želju tu igru objaviti za druge korisnike ili spremiti sebi kako bi ju mogao kasnije rješavati, bit će potrebna registracija ili prijava u slučaju da korisnik već ima korisnički račun. Nakon što se korisnik prijavi, moći će učiniti već spomenuto i na taj način kasnije ponovno pristupiti spremljenoj igri. Ukoliko korisnik odluči objaviti igru, bit će mu ponuđena ta mogućnost, a to znači da će ona u budućnost biti vidljiva na početnoj stranici te će ju i drugi korisnici vidjeti i imati

priliku riješiti. Ako se korisnik ipak odluči zadržati Sudoku za sebe, samo će on moći vidjeti taj Sudoku u svom korisničkom profilu, ali će i dalje postojati tipka pomoću koje je može objaviti za druge korisnike.

#### **3.1.4. Dodatne mogućnosti**

Osim osnovnih značajki web aplikacije za rješavanje, provjeru i objavljivanje Sudoku igara, moguće je dodati još brojne značajke koje poboljšavaju korisničko iskustvo na stranici. Tu dolazi u obzir implementacija bržih algoritama za rješavanje i provjeru Sudoku igara, dodavanje različitih tema izgleda aplikacije, prilagodba aplikacije za ljude s poteškoćama, internacionalizacija unosom većeg broja stranih jezika i slično. Za sada takve i slične značajke ostaju kao mogućnost i smjer u kojem se aplikacija može u budućnosti razvijati.

## 3.2. Korištene tehnologije

U ovom će dijelu biti kratko navedene tehnologije koje su korištene za izradu korisničkog sučelja aplikacije te rukovanje s podacima iz baze podataka i postavljanje baze podataka. Neke od tehnologija koje su korištene uključuje React, Node.js, Redis i MySQL, a kod je pisan unutar Visual Studio Codea. Tehnologije su odabrane pretežito zbog svojih širokih mogućnosti i jednostavnosti te preglednosti koda.

### 3.2.1. Korisničko sučelje web aplikacije

Za izradu korisničkog sučelja i komunikaciju s bazom podataka kao uređivač teksta otvorenog koda korišten je Microsoftov Visual Studio Code koji sadrži podršku za brojne programske jezike te ispravak pogrešaka i koda. Ima ugrađeni Git i prilagodljivo sučelje s brojnim ekstenzijama koje je moguće instalirati za lakši rad.

Unutar Visual Studio Codea za izradu korisničkog sučelja korištena je React biblioteka koju održava Facebook. React je biblioteka otvorenog koda koja se može koristiti kao baza pri razvoju jednostraničnih ili mobilnih aplikacija (koristeći React Native). Glavni zadatak React biblioteke je održavanje stanja i prenošenje podataka modelu objekta dokumenta (engl. *Document Object Model, DOM*) i zbog toga takve aplikacije uglavnom zahtijevaju korištenje dodatnih biblioteka za usmjeravanje (engl. *routing*) ili održavanje stanja, pri čemu se onda koriste React Router i Redux biblioteke.

Spomenute biblioteke koriste se i u izradi aplikacije u sklopu ovog diplomskog rada. Osim njih, koriste se i brojne druge biblioteke koje olakšavaju rad i smanjuju opseg koda, a među njima se nalazi i Material UI [14] biblioteka kojom aplikacija dobiva pristup već stiliziranim komponentama i animacijama koje je moguće prilagoditi potrebama aplikacije.

### 3.2.2. Node.js

Za izradu *backenda* koristi se Node.js okruženje koje se piše u Javascript programskom jeziku i izvršava Javascript kod. Razlog zašto je odabran baš Node.js za *backend* jest to što je jednostavan, pouzdan, skalabilan i osigurava dobre performanse. Također, piše se u Javascript kodu, što je jednostavnije od učenja novog jezika i na taj su način i *backend* i *frontend* pisani u istom programskom jeziku.

Nadalje, upravljač Node paketima (engl. *Node Package Manager, NPM*) [15] koji dolazi s instalacijom Node.js-a sadrži ugrađenu podršku za upravljanje paketima. NPM modul Express.js koji je korišten u aplikaciji za izradu diplomskog rada jedan je od najčešće korištenih i standard

za većinu današnjih Node.js aplikacija. Express.js olakšava povezivanje s bazom podataka te upravljanje rutama, zahtjevima i međuslojevima za provjeru podataka.

Za spremanje korisničkih informacija i pristupnih tokena korišten je Redis. Redis je baza podataka otvorenog koda koja podržava spremanje podataka u radnu memoriju ili na disk te podržava mnogo tipova podataka kao što su stringovi, setovi, sortirani setovi, liste i mape.

### **3.2.3. Baza podataka i Heroku platforma**

Podaci o korisnicima i sačuvanim Sudoku igrama moraju biti negdje spremljeni kako bi im se moglo pristupiti s *backenda* kada je to potrebno unutar aplikacije. Za to je korištena MySQL baza podataka, a unutar *backend* projekta nalazi se direktorij koji sadrži datoteke s funkcijama koje sadrže SQL naredbe za dohvaćanje potrebnih podataka iz baze podataka.

Baza podataka kreirana je pomoću Heroku platforme [16], a za uspostavljanje konekcije s bazom podataka unutar Node.js-a koristila se *createPool* funkcija, unutar koje se navode informacije za povezivanje s bazom podataka, kao što su port, poslužitelj, korisničko ime, lozinka i slično.

Nakon što se aplikacija izradi lokalno, bit će stavljena u pogon na platformu Heroku, koja dopušta direktno povezivanje s Gitom.

## 4. IMPLEMENTACIJA SUDOKU WEB APLIKACIJE

U ovom poglavlju bit će detaljnije opisano izrađeno programsko rješenje za rješavanje, provjeru i objavljivanje Sudoku igara. Bit će opisani svi dijelovi aplikacije redom kojim su izrađeni te korisničko sučelje, ali i ono što se događa u pozadini. Aplikacija se razvijala koristeći iterativni model razvoja programske podrške koji se može vidjeti na slici 4.1. Iterativni model odabran je zbog rane mogućnosti opažanja pogrešaka, opsežnijeg testiranja po manjim modulima, smanjenog vremena izvršavanja te jednostavnijeg praćenja tijeka aplikacije.



Slika 4.1. Iterativni model razvoja programske podrške, preuzeto sa [17]

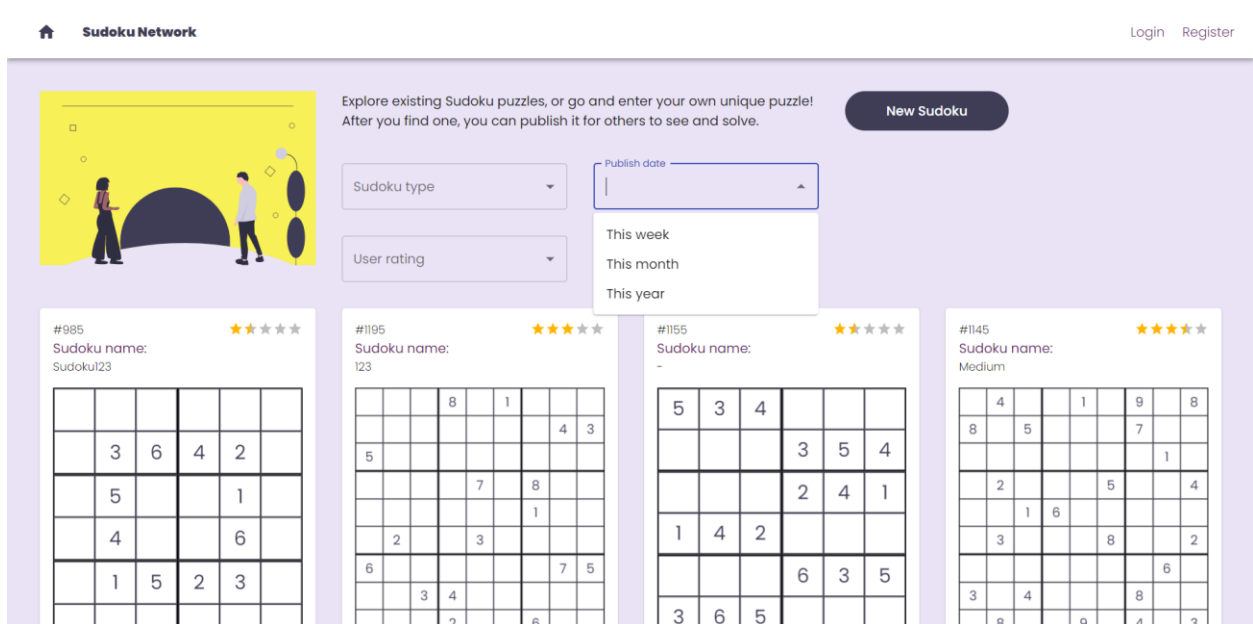
Iterativni model sastoji se prvo od planiranja i definiranja zahtjeva i osnovnog dizajna. Zatim započinje iterativni proces u kojem se implementira određeni dio proizvoda i svakom iteracijom poboljšava i dodaje nove značajke prema zahtjevima. Najvažnije kod ovog modela je definirati one zahtjeve koji su najvažniji i kojih se treba pridržavati do kraja. Osim njih, moguće je da u pojedinoj iteraciji dođe do promjena, stoga pri definiranju zahtjeva nije potrebno napraviti potpunu specifikaciju istih. Dizajn programske podrške trebao bi se pridržavati početno definiranog dizajna. S iteracijama se nastavlja sve dok potpuni proizvod nije implementiran. Ovakav postupak omogućava testiranje i implementaciju manjih dijelova proizvoda i pri tome verifikaciju korisničkih zahtjeva u svakom ciklusu.

## 4.1. Sučelja izrađene aplikacije *Sudoku Network*

Najvažniji dio aplikacije su sučelja koja korisniku prikazuju Sudoku mrežu (engl. *grid*) u kojoj može rješavati ili pokušati dodati novi jedinstveni Sudoku. U sljedećim potpoglavljima opisana je web aplikacija koja sadrži takva sučelja i na taj način omogućuje korisnicima okruženje u kojem mogu dodavati, rješavati i spremati Sudoku igre.

### 4.1.1. Početna stranica i rješavanje Sudokua

Ono što gotovo svaka Sudoku web aplikacija na internetu ima jest mogućnost rješavanja već zadane Sudoku igre. U ovom radu to je učinjeno na sljedeći način: korisnik (bilo registriran ili gost) ulazi na početnu stranicu na kojoj se nalazi lista Sudoku igara (slika 4.2.).

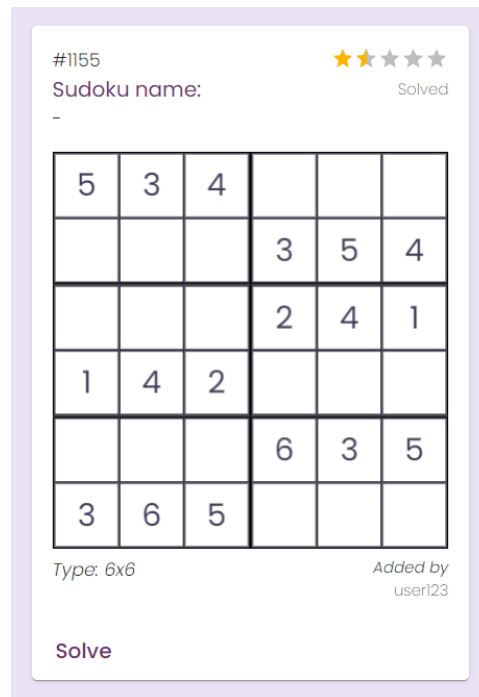


Slika 4.2. Početna stranica s listom Sudokua i mogućnostima filtriranja

Početna stranica je organizirana na način da prijavljeni korisnik vidi zaglavlje s opcijama koje mu dopuštaju odlazak na stranicu s listom Sudokua koje je korisnik osobno dodao, uređivanje korisničkog profila te odjavljivanje iz aplikacije. U centru početne stranice kao najvažniji element nalazi se lista Sudoku igara, a iznad nje opcije pomoću kojih je moguće filtrirati prikazanu listu prema kriterijima kao što su vrsta igre, vrijeme objave i korisnička procjena težine. Trenutno su implementirane tri vrste Sudoku igara, a to su klasični 9x9 Sudoku, manji 6x6 Sudoku te 9x9 X-Sudoku.

Lista Sudoku igara je popis svih igara koje su objavljene od strane svih korisnika. Svaka stavka unutar liste predstavlja karticu s osnovnim informacijama o igri, što uključuje identifikacijski broj,

prosječnu ocjenu težine prema procjenama korisnika, vrstu Sudokua i, ukoliko postoji, korisničko ime korisnika koji je objavio Sudoku te oznaka 'riješeno' ukoliko je prijavljeni korisnik već riješio spomenuti Sudoku. Artikli su poslagani u nekoliko stupaca unutar stranice, a moguće je listati prema dolje sve dok postoje artikli. Upravo zbog te značajke korisno je upotrijebiti filtriranje koje se nalazi na vrhu stranice kako bi se smanjio broj rezultata i lakše pronašao željeni Sudoku. Prve na stranici prikazane su vremenski najnovije igre. Na slici 4.3. prikazana je jedna komponenta Sudoku kartice.



Slika 4.3. Sudoku kartica s informacijama

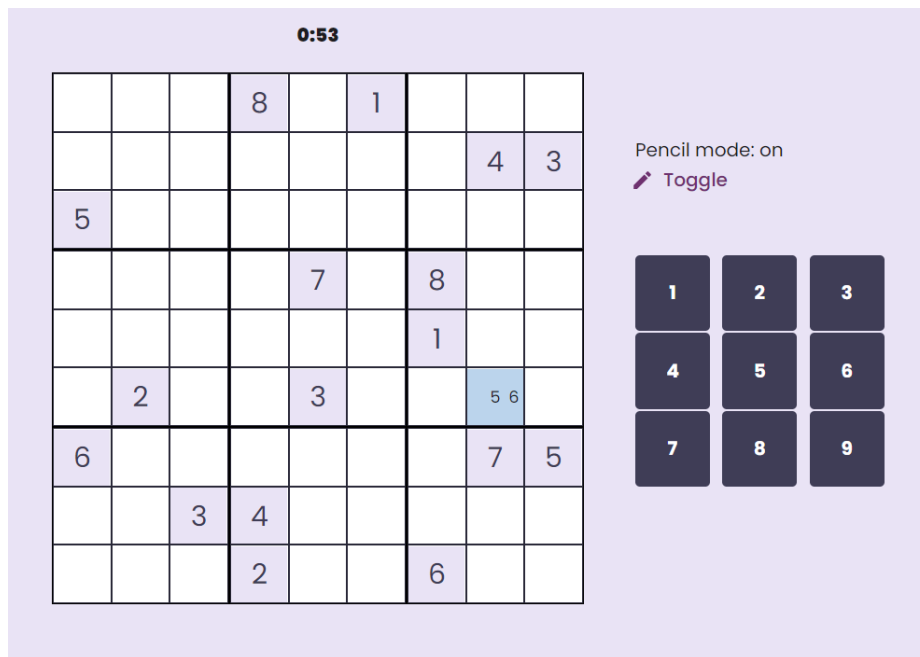
Klikom na jedan takav artikl u listi igara, korisnik se preusmjerava na stranicu koja sadrži Sudoku mrežu za rješavanje. Unutar toga nalazi se jedna Sudoku igra. Sudoku igra predstavlja mrežu polja koja mogu biti već ispunjena određenim brojem ili prazna. Jedna Sudoku igra može biti različitih vrsta, u ovom slučaju mala 6x6, klasična 9x9 ili X-Sudoku, također 9x9. Pri rješavanju Sudokua u mreži su već zadani neki brojevi koje korisnik ne može mijenjati te prazna polja koje korisnik treba ispuniti kako bi ispravno riješio Sudoku. Prilikom upisivanja brojeva u mrežu, ukoliko korisnik upiše broj koji prema pravilima ne bi smio biti na tom mjestu, prikazuje se greška i korisnik odmah zna da treba promijeniti to polje jer se ne podudara s ostalima (slika 4.4.).

5	3	4			
			3	5	4
			2	4	1
1	4	2			
	3		6	3	5
3	6	5			

Slika 4.4. Automatski prikaz pogreški unutar Sudoku mreže

Ako korisnik prvi put rješava Sudoku, tijekom rješavanja će mu biti mjereno vrijeme koje nakon što uspješno riješi ne mogu vidjeti drugi korisnici, ali će tom korisniku biti prikazano svaki sljedeći put kada ponovno uđe u istu igru. Moguće je koristiti i upis kandidata u ćelije (engl. *pencil marks*). Prethodno spomenuto prikazano je na slici 4.5. Upisivanje kandidata u ćelije pomaže pri rješavanju i pronalasku sljedećeg mogućeg broja za upis. Pored mreže nalaze se i tipke s brojevima pomoću kojih se mogu unositi u mrežu, no ta značajka bitnija je za mobilne uređaje. Na webu je lakše kretanje kroz mrežu klikanjem na ćelije, a najjednostavnije kretanje kroz ćelije jest pomoću strjelica na tipkovnici.





Slika 4.5. Upis kandidata u ćeliju

Nakon što korisnik točno ispuni Sudoku mrežu, dobiva obavijest i može se vratiti na početnu stranicu i pronaći novi Sudoku za rješavanje. Provjera točnosti obavlja se pozivom funkcije za dobivanje rješenja na *backendu* (slika 4.6.), a zatim uspoređivanjem dobivenog rješenja s rješenjem korisnika.

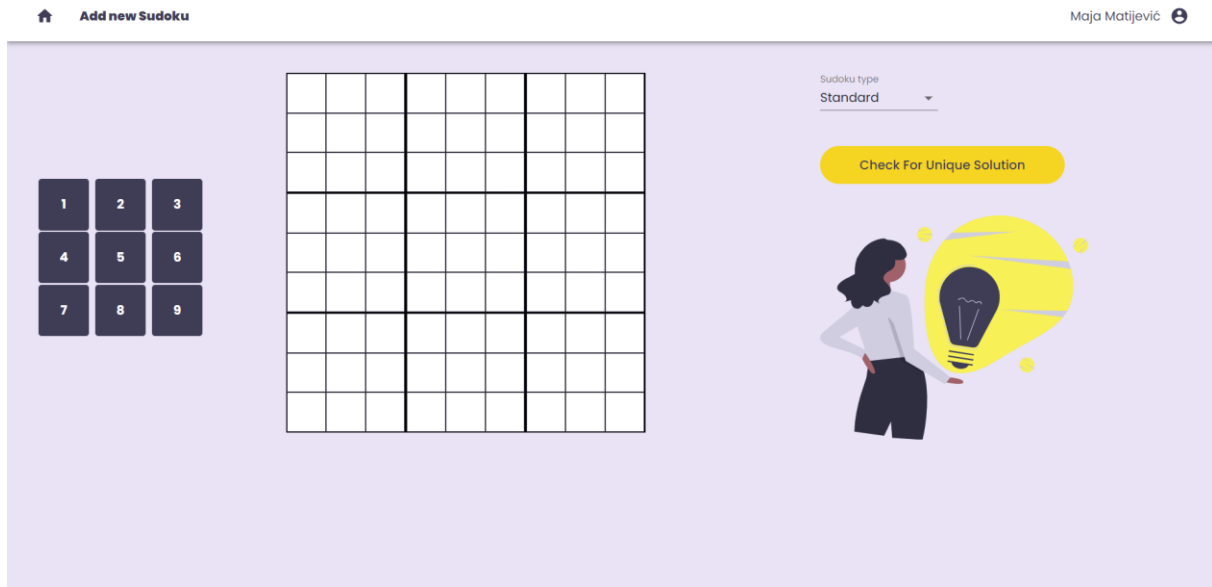
```
router.post("/getSolution", (req, res) => {
  const { board } = req.body;
  try {
    const solution = solveWrapper(board);
    const results = solution.length === 1 ? solution : undefined;
    res.json(results[0]);
  } catch (e) {
    res.sendStatus(404);
  }
});
```

Slika 4.6. Node.js funkcija za dohvaćanje rješenja

Funkcija *solveWrapper* sa slike 4.6. je dugačka i služi za rješavanje Sudokua i sadrži veći broj funkcija od kojih neke provjeravaju postojanje golih samaca (engl. *naked singles*) i skrivenih samaca (engl. *hidden singles*) te ćelija s najmanje mogućih kandidata i na taj način traže sljedeći broj za upis u mrežu.

### 4.1.2. Unos nove igre

Početna stranica sadrži još jednu opciju, a to je tipka koja vodi na sučelje slično onom za rješavanje Sudokua. No, u ovom slučaju, radi se o dodavanju nove igre. U takvoj situaciji pred korisnikom se nalazi prazna Sudoku mreža u koju upisuje brojeve (slika 4.7.).



Slika 4.7. Sučelje za dodavanje novog Sudokua

Pri unosu nove igre, korisnik može odabrati tip Sudokua od prethodno spomenuta tri. Tada se mijenja mreža u ovisnosti o tipu.

Kao i kod rješavanja, prikazuje se pogreška ukoliko se ponavljaju brojevi u pojedinim stupcima ili redovima, ali ovdje je najvažnija tipka koja omogućava korisniku provjeru je li pronašao jedinstveno rješenje. Nakon što je pronašao rješenje, nudi mu se mogućnost objave kako bi ju i drugi korisnici mogli vidjeti na početnoj stranici i riješiti. Objava je moguća samo za registrirane korisnike i taj slučaj je više opisan u sljedećem potpoglavlju.

Provjera jedinstvenosti Sudokua, a i samog rješenja obavlja se na *backendu* koristeći istu funkciju. Funkcija vraća trodimenzionalno polje sa svim mogućim rješenjima Sudokua koja su u obliku dvodimenzionalnog polja (slika 4.8.), s tim da se zaustavlja na zadanom broju kako ne bi prolazila kroz sve iteracije.

```
const grid = [
  [0, 0, 4, 0, 5, 0, 0, 0, 0],
  [9, 0, 0, 7, 3, 4, 6, 0, 0],
  [0, 0, 3, 0, 2, 1, 0, 4, 9],
  [0, 3, 5, 0, 9, 0, 4, 8, 0],
  [0, 9, 0, 0, 0, 0, 0, 3, 0],
  [0, 7, 6, 0, 1, 0, 9, 2, 0],
  [3, 1, 0, 9, 7, 0, 2, 0, 0],
  [0, 0, 9, 1, 8, 2, 0, 0, 3],
  [0, 0, 0, 0, 6, 0, 1, 0, 0],
];
```

*Slika 4.8. Primjer oblika polja koje se šalje funkciji za rješavanje Sudokua*

Ukoliko se traži jedinstvenost zadanog Sudokua, dovoljno je prebrojati dobivena dvodimenzionalna polja jer jedinstveni Sudoku treba imati jedno rješenje.

## **4.2. Specifičnost aplikacije za registriranog korisnika**

Određeni dijelovi aplikacije dostupni su korisniku samo ako je registriran i prijavljen u aplikaciju. Prijava je u tom slučaju nužna kako bi se moglo obaviti spremanje podataka kojima korisnik želi kasnije ponovno pristupiti.

### **4.2.1. Registracija i prijava**

Ako korisnik nije prethodno prijavljen u aplikaciju, prva stranica koja će mu se prikazati jest jednostavna stranica koja od njega traži da odabere želi li se registrirati, prijaviti ili nastaviti kao gost (slika 4.9).

# Welcome

Sudoku network is a place where you can create, check and solve various Sudoku puzzles. Once you manage to create a new unique puzzle, feel free to share, so others can solve it too.

Have fun!

Log In

Don't have an account? Sign up here



Continue as guest

*Slika 4.9. Inicijalno sučelje za neprijavljenog korisnika*

Ukoliko nastavi kao gost, može rješavati Sudoku igre, ali uz prijavu ili registraciju, otvara mu se još nekoliko dodatnih značajki, kao što je objava Sudokua i korisnički profil.

Klikom na tipku koja vodi na registraciju, prikazana je stranica s formom u koju korisnik treba upisati osobne podatke: ime i prezime, e-mail, korisničko ime i lozinku. U formi se provjerava jesu li sva polja ispunjena i jesu li podatci dovoljno dugački te je li unesen ispravan e-mail. Nakon toga korisnik se može uspješno registrirati i dodan je u bazu podataka te se odmah obavlja i prijava kako ne bi morao ponovno upisivati podatke. Nakon registracije, korisnik se nalazi na početnoj stranici na kojoj bi se nalazio i u slučaju da je odabrao nastaviti kao gost, ali s nekim dodatnim mogućnostima.

U slučaju da korisnik već posjeduje korisnički račun na web aplikaciji, može otići na prijavu. U tom slučaju otvara mu se jednostavna forma u koju samo treba upisat svoj e-mail i lozinku (slika 4.10.). Ako je upisao ispravne podatke koji postoje u bazi podataka korisnika, uspješno se prijavljuje i, kao i u prošlom slučaju, nalazi na početnoj stranici.

# Log in to Sudoku Network



Slika 4.10. Korisnička prijava u aplikaciju

Nakon prijave, unutar *frontend* aplikacije koristi se React Redux biblioteka, to jest Redux.js Toolkit [18] pomoću kojeg se spremanju informacije o trenutno prijavljenom korisniku. Redux omogućuje dohvaćanje stanja iz svih komponenti aplikacije te na taj način olakšava upravljanje stanjima koja se često mijenjaju kroz više komponenti aplikacije. Spremanjem korisničkih podataka u Redux jednostavno se mogu dohvatiti iz svake komponente, a isto tako i ukloniti pri odjavi korisnika. Redux programski kod za spremanje korisnika prikazan je na slici 4.11.

```
import { createSlice, PayloadAction } from "@reduxjs/toolkit";
import { User } from "../../types/User";

interface AuthState {
  user: User | undefined;
}

const initialState: AuthState = {
  user: undefined,
};

const authSlice = createSlice({
  name: "auth",
  initialState,
  reducers: {
    setUser: (state, action: PayloadAction<User | undefined>) => {
      state.user = action.payload;
    },
    removeUser: (state) => {
      state.user = undefined;
    },
  },
});

export const { setUser, removeUser } = authSlice.actions;
export const authReducer = authSlice.reducer;
```

Slika 4.11. Redux.js Toolkit – spremanje i brisanje korisnika

Za sigurnu prijavu korisnika koristi se JWT token te je ekriptiran podacima o trenutnoj sesiji. Koriste se dva tokena, pristupni i token za osvježavanje (engl. *refresh token*). Pristupni token ima kraće trajanje, stoga se token za osvježavanje koristi za obnovu prvoga kako korisnik ne bi neprestano morao ponovno unositi svoje podatke (slika 4.12.). Pri tom se provjerava je li token spremljen pod korisničkim identifikacijskim brojem u Redis klijentu. Ukoliko nije, token se ne može obnoviti i potrebna je ponovna prijava korisnika kako bi token bio spremljen u Redis bazu.

```
router.post("/refreshToken", verifyRefreshToken, (req, res) => {
  try {
    const refreshToken = req.body.refreshToken;
    const userInfo = jwtDecode(refreshToken);
    if (userInfo && refreshToken) {
      redisClient.get(userInfo.userID.toString(), function (err, val) {
        if (err) res.status(404);
        else {
          if (val) {
            const token = jwt.sign(
              { sub: userInfo.sub, userID: userInfo.userID },
              process.env.JWT_ACCESS_SECRET,
              {
                expiresIn: process.env.JWT_ACCESS_TIME,
              }
            );
            res.status(200).json(token);
          }
        }
      });
    } else {
      res.status(401);
    }
  } catch (e) {
    res.status(401);
  }
});
```

Slika 4.12. Backend metoda za obnavljanje tokena

Tokeni se spremaju u *Local Storage* i svaki zahtjev na *backend* u zaglavlju mora sadržavati pristupni token kako bi se zahtjev uspješno izvršio, a na *backendu* se također obavlja provjera valjanosti pristupnog tokena. Ukoliko je taj token istekao, provjerava se valjanost tokena za osvježavanje i ako je valjan, koristi se za osvježavanje pristupnog tokena, a ako nije, korisnik se mora ponovno prijaviti.

#### 4.2.2. Objavljivanje i ocjenjivanje Sudokua

Nakon što je korisnik registriran i prijavljen i odluči dodati novi Sudoku, kada pronade jedinstveno rješenje, može objaviti ili spremiti Sudoku. Ako odluči spremiti Sudoku, otvara mu se dijalog sa slike 4.13. Može unijeti naziv svog Sudokua, a može ga i objaviti za druge korisnike. Nakon objave, Sudoku će biti vidljiv na početnoj stranici na prvom mjestu i drugi će ga korisnici moći rješavati i ocjenjivati njegovu težinu.

## Save your unique Sudoku puzzle

Name your Sudoku

If you wish to publish your Sudoku for others to see, check the box

Publish Sudoku

CANCEL SAVE SUDOKU

Slika 4.13. Dijalog za objavljivanje Sudokua

Za ocjenjivanje Sudokua, korisnik također treba biti prijavljen. Nije moguće ocijeniti bilo koju igru, već samo onu koju prijavljeni korisnik uspješno riješi, i to prvi put. Kad uspješno riješi Sudoku, otvara mu se jednostavni dijalog u kojem treba ocijeniti, to jest procijeniti težinu Sudokua kojeg je prethodno riješio. Uzevši u obzir sve ocjene jedne igre, na *backendu* se računa prosječna ocjena te se ista prikazuje unutar kartice tog Sudokua i prema tom kriteriju se mogu sortirati igre na početnoj stranici.

### 4.2.3. Korisnički profil

Prijavljeni korisnik ima mogućnost prikaza i mijenjanja svojih osobnih podataka. Navigacijom u profil koja se nalazi u zaglavlju, otvara se stranica s korisničkim podacima, gdje korisnik može promijeniti podatke koje želi. Također, korisnik može obrisati svoj profil i odjaviti se iz aplikacije.

### 4.3. Mogućnosti unaprjeđenja izrađene aplikacije

Web aplikacija izrađena u sklopu ovog diplomskog rada omogućuje korisnicima jednostavno i pregledno sučelje za dodavanje i rješavanje Sudoku zagonetki. Osim olakšane mogućnosti pretraživanja Sudokua pomoću filtera, aplikacija podržava rješavanje i dodavanje različitih vrsti Sudokua što doprinosi raznolikosti unutar aplikacije. Spremanje Sudokua za kasnije te objavljivanje kako bi ih i drugi korisnici mogli rješavati su neke od značajki koje čine izrađenu web aplikaciju znatno različitom od postojećih. Također, prijava korisnika dopušta povratak korisnika njegovim prijašnje spremljenim igrama te njihovo ponovno rješavanje.

Dakako, izrađena web aplikacija može se nadograditi na brojne načine. Primjerice, trenutno su unutar aplikacije podržane samo tri vrste Sudokua, što se može znatno proširiti kako bi bio podržan

što veći broj različitih Sudoku zagonetki te na taj način povećala dinamičnost. Osim toga, korišteni algoritam može se nadograditi ili zamijeniti novim algoritmima već spomenutim u ovom diplomskom radu. Sustav ocjenjivanja Sudokua, koji je trenutno korisnička procjena težine istih, mogao bi se poboljšati korištenjem automatske procjene težine Sudokua u ovisnosti o određenim kriterijima, primjerice prosječnom trajanju korisničkog rješavanja te početnom broju popunjenih ćelija unutar Sudoku mreže. Nadalje, aplikacija bi mogla biti nadograđena u obliku podrške za osobe s poteškoćama kao što je slabovidnost te takvim osobama olakšati i omogućiti korištenje ovakve vrste aplikacije, koja im za sada nije pristupačna. Na kraju, u aplikaciju se može dodati pomoć pri rješavanju Sudokua, a s tim i sučelje za podučavanje korisnika različitim tehnikama pri rješavanju.

Sve navedene mogućnosti za nadogradnju izrađene web aplikacije pokazuju velik potencijal iste za znatno poboljšanje i neprestano održavanje. Potrebno je i dodatno testiranje za pronalazak skrivenijih problema i osiguranje da su spriječene previđene pogreške.



## 5. ZAKLJUČAK

Iako se na prvi pogled čini kao da je Sudoku poprilično monoton i da ne može biti raznolik i dinamičan, to nije tako. Sudoku se kroz godine razvijao i rastao, nastao je velik broj varijanti te popularne zagonetke, a ljudima nikada nije postao dosadan. Kao umni izazov nikada neće zastarjeti, a tehnologija mu je samo donijela još veću mogućnost širenja kroz cijeli svijet.

Unatoč brojnim stranicama na internetu koje se bave temom Sudoku igara, nedostatak dovoljno dobrih web aplikacija koje obuhvaćaju sve značajke za rješavanje i evaluaciju Sudoku igara inspirirao je izradu ovog diplomskog rada.

U radu se izrađivala aplikacija koja je omogućila korisnicima preglednu i funkcionalnu stranicu na kojoj mogu riješiti, objaviti i kreirati nove Sudoku zagonetke. Korisnik se može ili ne mora registrirati ili prijaviti, a ako se odluči na to, dobiva dodatne mogućnosti – objavljivanje i spremanje novih Sudokua koje je sam kreirao. Dodavanje novog Sudokua, točnije provjera njegove jedinstvenosti, ali i provjera rješenja odvija se pomoću poboljšanog *backtracking* algoritma čije performanse i više nego zadovoljavaju zahtjeve ovog diplomskog rada. Pregledna početna stranica korisnicima dopušta pretraživanje svih objavljenih Sudokua, a nakon uspješnog rješavanja, svojom procjenom težine Sudokua mogu doprinijeti ukupnoj ocjeni istog. Dijeljenjem pronađene jedinstvene Sudoku zagonetke korisnik doprinosi rastu već postojeće baze Sudokua.

Problemi ovog rada bave se izradom web aplikacije koje će u isto vrijeme biti jednostavna, funkcionalna, ispravno implementirati algoritme za rješavanje i provjeru Sudoku igara i imati dobre performanse.

Nakon završetka ovog rada aplikacija se može nadograđivati na razne načine, primjerice poboljšanje algoritama, redizajniranje ili dodavanje novih vrsta Sudokua te brojnih drugih značajki i mogućnosti.

## LITERATURA

- [1] Sudoku [online], wikipedia.org, dostupno na: <https://en.wikipedia.org/wiki/Sudoku>, [29.8.2021.]
- [2] Sudoku Solver [online], dostupno na: <https://sudokuspoiler.azurewebsites.net/> [29.8.2021.]
- [3] J. P. Grossman, Sudoku Helper [online], dostupno na: <https://www.mathstat.dal.ca/~jpg/sudoku/> [29.8.2021.]
- [4] Sudoku solver [online], dostupno na: <https://sudokusolver.app/> [29.8.2021.]
- [5] Sudoku solver [online], Aire Technologies, dostupno na: <https://www.sudoku-solutions.com/> [29.8.2021.]
- [6] Andrew Stewart, Sudoku Wiki [online], Syndicated Puzzles Inc, 2005., dostupno na: <https://www.sudokuwiki.org/sudoku.htm> [29.8.2021.]
- [7] Grant McLean, Sudoku Exchange [online], dostupno na: <https://sudokuexchange.com/> [29.8.2021.]
- [8] M. Harrysson, H. Laestander, „Solving Sudoku efficiently with Dancing Links“ [online], Stockholm, Švedska, 2014., dostupno na: [https://www.kth.se/social/files/58861771f276547fe1dbf8d1/HLaestanderMHarrysson\\_dka nd14.pdf](https://www.kth.se/social/files/58861771f276547fe1dbf8d1/HLaestanderMHarrysson_dka nd14.pdf) [29.8.2021.]
- [9] J. F. Crook, „A Pencil-and-Paper Algorithm for Solving Sudoku Puzzles“, Notices of the AMS, br. 4, sv. 56, travanj 2009.
- [10] R. Hughes, R. V. Yampolskiy, „Solving Sudoku Puzzles with Wisdom of Artificial Crowds“, University of Louisville, Louisville, SAD, siječanj 2013.
- [11] H. Lloyd, M. Amos, „Solving Sudoku With Ant Colony Optimization“, IEEE Transactions on Games, br.3, sv.12, rujan 2020.
- [12] J. Ekström, K. Pitkäljärvi, „The backtracking algorithm and different representations for solving Sudoku Puzzles“, KTH Royal Institute of Technology, Stockholm, Švedska, 2014.
- [13] R. Pelánek, „Difficulty Rating of Sudoku Puzzles: An Overview and Evaluation,“, Masaryk University, Brno, Češka Republika, ožujak 2014.
- [14] Material UI [online], dostupno na: <https://material-ui.com/> [29.8.2021.]
- [15] npm [online], npm, Inc., 2014., dostupno na: <https://www.npmjs.com/> [29.8.2021.]
- [16] Heroku [online], Salesforce.com, 2007., <https://www.heroku.com> [29.8.2021.]

- [17] Iterativni model (slika) [online], dostupno na: [https://www.researchgate.net/figure/SDLC-Iterative-Model-2\\_fig4\\_338710620](https://www.researchgate.net/figure/SDLC-Iterative-Model-2_fig4_338710620) [29.8.2021.]
- [18] Redux.js Toolkit [online], dostupno na: <https://redux-toolkit.js.org/> [29.8.2021.]

## SAŽETAK

Sudoku je igra koja ne prestaje biti popularna u cijelom svijetu već dugi niz godina. Ubrzani razvoj tehnologije donio je brojne aplikacije za rješavanje Sudoku igara, kao i brojne algoritme koje uspješno rješava računalo.

U sklopu ovog rada analizirane su postojeće web aplikacije za rješavanje i pomoć pri rješavanju Sudoku igara. Opisani su osnovni zahtjevi na aplikaciju i faze kroz koje će korisnik prolaziti koristeći aplikaciju. Zatim je implementirano programsko rješenje koristeći React okruženje.

Aplikacija omogućuje korisniku unos nove Sudoku igre različitih varijanti, provjeru jedinstvenosti rješenja i objavljivanje te igre. Također, omogućeno je rješavanje već postojećih Sudoku igara i njihovo ocjenjivanje.

**Ključne riječi:** algoritam, React, Sudoku, web aplikacija

## ABSTRACT

### WEB APPLICATION FOR ADDING, SOLVING AND RATING SUDOKU GAMES

Sudoku is a game that never ceases to stop being popular all around the world for years on end. The evergrowing development of technology brought with it many applications for solving Sudoku, as well as various algorithms successfully solved by computers.

Existing web applications for solving and helping to solve Sudoku puzzles have been analyzed as a part of this thesis. Main application requirements and user stories have been described. The solution was implemented using React environment.

The application enables the user to enter a new Sudoku which can be of different variants, to check its uniqueness and post it for other users to see. Furthermore, it is also possible to solve already existing puzzles and rate them.

**Key words:** algorithm, React, Sudoku, web application

## **ŽIVOTOPIS**

Maja Matijević rođena je 29. lipnja 1997. godine u Osijeku. Osnovnu školu Svete Ane u Osijeku završava 2012. godine, nakon čega upisuje III. gimnaziju Osijek (prirodoslovno-matematička). Nakon završenog srednjoškolskog obrazovanja, 2016. godine upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku, današnjem Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Preddiplomski studij računarstva završava 2019. godine te iste godine na istom fakultetu upisuje diplomski studij računarstva, izborni blok Programsko inženjerstvo.

---

Potpis autora

## PRILOZI

[P1] DVD s programskim kodom

[P2] Programski kod [online], dostupno na: <https://github.com/Maja97/sudoku-network>