Upotreba Blockchain tehnologije

Filip, Bruno

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVO

BRUNO FILIP

UPOTREBA *BLOCKCHAIN* TEHNOLOGIJE

ZAVRŠNI RAD

ČAKOVEC, 2022.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVO

BRUNO FILIP

UPOTREBA *BLOCKCHAIN* TEHNOLOGIJE

THE USE OF BLOCKCHAIN TECHNOLOGY

ZAVRŠNI RAD

Mentor:

dr. sc. Sanja Brekalo

ČAKOVEC, 2022.

Sažetak

Tema je završnog rada „blockchain“ tehnologija, njena primjena te osmišljavanje i izrada aplikacije koja primjenjuje „blockchain“ tehnologiju.

U radu je „blockhain“ tehnologija primijenjena kod posredovanja vlasništva zemljišta te se temelji na principu rada privatne „blockchain“ mreže (engl. permissioned blockchain). U aplikaciju je integrirana proizvoljno izrađena mapa koja sadrži 25 zemljišta kojima je moguće posredovati.

Izrađena aplikacija ima tri razine upravljanja aplikacijom: gost, korisnik i administrator. Gostom se smatra svaki posjetitelj internetske aplikacije koji ima mogućnost registracije ili prijave u sustav. Prijavljeni korisnik može slanjem zahtjeva vlasniku zemljišta zatražiti da vlasnik prenese zemljište na njega. Ključnu ulogu za dodavanje novih transakcija vlasništva zemljišta u „blockchain“ imaju korisnici sa statusom nadležnog tijela, koji glasaju nad potvrđenim zahtjevima korisnika. Kada je postignuta glasačka većina, dodaje se novi blok.

Tehnologije koje su korištene za izradu aplikacije su ASP.NET platforma , C# programski jezik za programiranje na ASP.NET platformi, HTML jezik kojim se definiraju elementi na internetskoj stranici, CSS za definiranje prikaza elemenata na internetskoj stranici, Bootstrap programski okvir (engl. framework) koji pruža već gotove predloške za prezentaciju elemenata internetske stranice, JavaScript skriptni jezik koji omogućuje interaktivnost internetske stranice te JQuery programski okvir koji omogućuje korištenje već gotovih JavaScript funkcija za obavljanje programskih zadataka.

Ključne riječi: *blockchain, C#, JavaScript, HTML, CSS*

SADRŽAJ:

1. Uvod 6
2. *Blockchain* 7
2.1. Povijest *blockchain* tehnologije 7
2.2. Vrste *blockchain* sustava 7
2.2.1. Javni *blockchain* 7
2.2.2. Privatni *blockchain* 8
2.3. *Hash* funkcija 9
2.3.1. SHA-256 *hash* algoritam 9
2.4. Kriptografija javnog ključa 10
2.5. Transakcije 11
2.5.1. UTXO 11
2.6. Struktura *blockchain* lanca 12
2.6.1. Struktura *blockchain* lanca 13
2.6.2. Zaglavlje bloka 13
2.7. Merkleovo stablo 14
2.8. Algoritmi za konsenzus 15
2.8.1. *Proof of work* 16
2.8.2. *Proof of stake* 17
3. Kriptovalute 18
3.1. *Bitcoin* (BTC) 18
3.2. *Ether* (ETH) 18
3.2.1. Pametni ugovori 19
3.2.2. *Ethereum virtual machine* 19
3.3. *Litecoin* (LTC) 19
3.4. *Ripple* (XRP) 20
4. Upotreba *blockchain* tehnologije 21
4.1. Plaćanje usluga i proizvoda kriptovalutama 21
4.2. Lanac opskrbe 21
4.3. Zdravstvo 22
4.4. Obrazovanje 22
4.5. *Internet of things* 22
5. Aplikacija 23

5.1. Korištene tehnologije 23
5.1.1. .NET platforma i ASP.NET 23
5.1.2. HTML, CSS i *Bootstrap* 23
5.1.3. *JavaScript* 23
5.2. Tri razine upravljanja aplikacijom 24
5.2.1. Gost 24
5.2.2. Korisnik 27
5.2.3. Korisnik nadležnog tijela 29
5.3. *Blockchain* aplikacije 30
5.3.1. Transakcija vlasništva zemljišta 30
5.3.2. Dodavanje transakcija u blok 31
5.3.3. Dohvaćanje zemljišta iz *blockchaina* 33
6. Zaključak 34
7. Reference 35
8. Popis slika i tabele 37
8.1. Slike 37
8.2. Tabele 38

1. Uvod

Blockchain tehnologija takav je način razmjene informacija između korisnika da su zapisi unutar sustava, kada su jednom upisani, nepromjenjivi. *Blockchain* sustav može biti javan ili privatan. Realiziranjem javnog *blockchain* sustava omogućuje se da je sustav decentraliziran pa nema potrebe za središnjim tijelom koje održava i kontrolira sustav. *Blockchain* tehnologija može se koristiti u različite svrhe kao sustav spremanja i praćenja informacija kod kojih je bitno da se informacije koje su jednom upisane više ne mogu mijenjati pa se zbog toga može pokazati kao dobar alat u poslovanju organizacija. Danas se najviše primjenjuje kod razmjene digitalnih kriptovaluta kao što su *bitcoin*, *ether* i ostale. *Blockchain* je poput glavne knjige zapisa (engl. *ledger*) u kojoj su zapisane transakcije koje se nalaze u povezanim blokovima *blockchaina* te pomoću njegovih svojstva te transakcije mogu biti provjerene i nitko ih ne može uređivati ili brisati jer će svaka promjena biti vidljiva i odbačena. Svaki *blockchain* sustav ima svoj algoritam za konsenzus, tj. način kako se novi blokovi dodaju u njega. *Blockchain* sustavi koriste kriptografiju i *hashiranje* podataka (engl. *hashing*) radi sigurnosti, kompresije, validacije i nepromjenjivosti informacija.

2. Blockchain

2.1. Povijest *blockchain* tehnologije

Stuart Haber i Scott Stornetta objavili su 1991. godine teorijski rad pod imenom „*How to Time-Stamp a Digital Document*“ u kojem opisuju rješavanje problema kako ovjeriti u kojem je točno vremenu neki dokument kreiran, odnosno kada je zadnji put izmijenjen. Taj proces, nazvan vremensko označavanje (engl. *timestamping*), koristi se *hashiranjem* i digitalnim potpisima (engl. *digital signatures*) kao mjerom sigurnosti kako se ne bi moglo promijeniti pravo vrijeme kreiranja ili izmjene dokumenta. [1]

Vremensko označavanje ima veliku ulogu u radu *blockchaina* te je pomoću te ideje nepoznata osoba pod pseudonimom Satoshi Nakamoto 2008. godine objavila rad pod imenom „*Bitcoin: A Peer-to-Peer Electronic Cash System*“, u kojemu opisuje sustav koji bi omogućio plaćanje gdje su korisnici povezani *online* „svaki sa svakim“ (engl. *peer-to-peer*) mrežu te se tako uklanja potreba za središnjim regulatornim financijskim tijelom, npr. bankom. Sustav je dizajniran tako da se stavi vremenska oznaka na svaku transakciju koje se pritom *hashiraju* i stavljaju u blokove te se rješavanjem matematičkih problema za koje je potrebno određeno vrijeme dodaju blokovi. U radu autor adresira i problem duplog trošenja (engl. *double spending*), to jest plaćanja istim novcem više od jedanput. Satoshi Nakamoto danas se smatra kreatorom *blockchain* sustava. [2]

2.2. Vrste *blockchain* sustava

Ovisno o njegovoj namjeni *blockchain* sustav može biti javan ili privatan. [3]

2.2.1. Javni *blockchain*

Javni *blockchain* sustav (engl. *permissionless blockchain*) dizajniran je po prvobitnoj ideji kreatora *blockchain sustava*, Satoshija Nakamota.

Sustav je decentraliziran, što znači da nema centralnog regulatornog tijela, već se temelji na modelu ravnopravnih partnera. Kada se korisnik spoji u *blockchain* sustav pomoću računala ili nekog drugog elektroničkog uređaja kao što je *smartphone*, tada ga

promatramo kao čvor mreže povezan na *peer-to-peer* način s ostalim čvorovima. Čvorovi su ravnopravni, ali ovisno o tome kako korisnik koristi *blockchain* sustav, čvor može biti različitog tipa sukladno njegovoj ulozi.

Razlikujemo tri vrste čvorova:

1. Puni čvor (engl. *full node*) - Puni čvor sadrži punu kopiju *blockchain* lanca. U vrijeme pisanja ovog završnog rada puna kopija *bitcoinova blockchain* sustava iznosila je oko 350 GB, stoga korisnik koji želi poprimiti ulogu punog čvora mora izdvojiti dosta prostora za pohranu. Dodavanjem novih blokova u sustav puni čvorovi međusobno komuniciraju, provjeravaju validnost novog dodanog bloka te u slučaju da je blok važeći, ažuriraju svoju kopiju *blockchain* lanca.

2. Čvor rudar (engl. *mining node*) - Čvorom rudarom smatra se svaki korisnik *blockchain* sustava koji se natječe u pronalasku rješenja zagonetke *proof of work* algoritma, potrebnim za dodavanje novog čvora u *blockchain* sustav. Može, ali i ne mora, sadržavati cijeli *blockchain* lanac.

3. Laki čvor (engl. *lightweight node*) – Laki čvor je čvor koji ne sadrži punu kopiju *blockchain* lanca niti je tzv. rudar. Laki čvor sadrži samo sva zaglavlja blokova *blockchain* lanca, koja su 1000 puta manja od punog *blockchain* lanca, pa u tom slučaju *smartphone* ili tablet može funkcionirati kao laki čvor i koristiti se aplikacijom novčanika (engl. *wallet*) za obavljanje transakcija. [3,4]

2.2.2. Privatni *blockchain*

Privatan *blockchain* sustav (engl. *permissioned*) je sustav koji nije javan, nego da bi korisnik mogao pristupiti sustavu, mora biti pozvan od administratora privatnog *blockchain* sustava. Ovakav zatvoreni tip *blockchain* sustava može se koristiti unutar poslovanja jedne ili dviju ili više organizacija. Nasuprot javnog *blockchain* sustava, privatan *blockchain* sustav nije javan, a ne mora biti ni transparentan korisnicima unutar sustava. [3]

2.3. *Hash* funkcija

Hashiranje podataka takav je način kriptiranja podataka da se za podatak gotovo bilo kakve duljine (ograničene odabirom algoritma za *hashiranje*) dobije uvijek *hash* poruka određene duljine. Rad *blockchain* tehnologije uvelike se zasniva na *hash* funkcijama. Na primjer, *hashiraju* se transakcije radi sigurnosti, anonimnosti i izrade korijena Merkleovog stabla transakcija pomoću kojeg se može provjeriti je li transakcija važeća ili nije. [3]

2.3.1. SHA-256 *hash* algoritam

Hash algoritam koji se najčešće koristi za kriptiranje podataka u *blockchain* sustavima je SHA-256 algoritam.

SHA-256 algoritam jedan je od sigurnih algoritma za *hashiranje* (engl. *secure hash algorithm*) propisanih standardom zvanim *Secure Hash Standard*, izdanim od američkog Nacionalnog instituta za standarde i tehnologiju (NIST), čiji rezultat daje poruku veličine 256 bita. SHA-256 algoritam nazvan je sigurnim algoritmom zbog toga što je dešifriranje ulazne poruke iz rezultata SHA-256 algoritma, tj. iz *hash* poruke, kao i pronaći dvije različite ulazne poruke koje bi davale isti *hash* rezultat, računalno prezahtjevan posao za računalnu moć današnjih računala. Kao i kod većine *hash* funkcija, promjenom samo jednog bita ulazne poruke dobiva se sasvim drugi *hash* rezultat. [5]

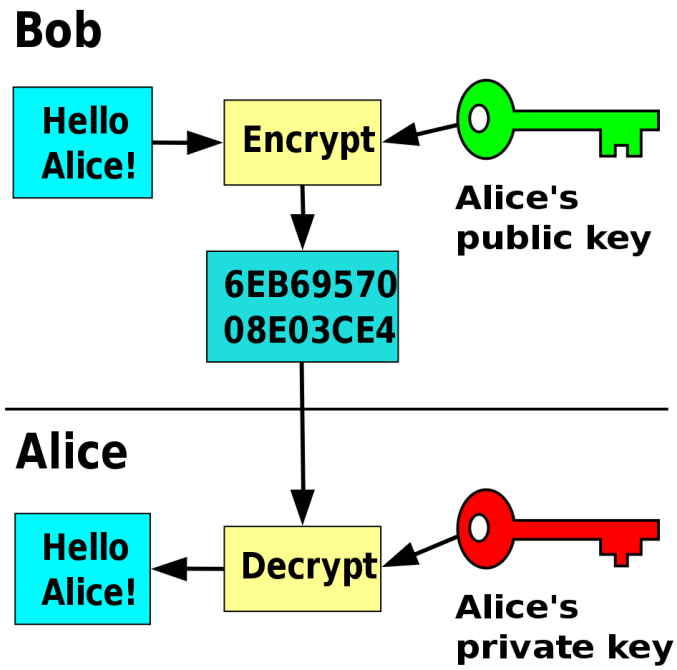
Primjer 1:

SHA-256(“Blockchain”**)**=
ef7797e13d3a75526946a3bcf00daec9fc9c9c4d51ddc7cc5df888f74dd434d1

SHA-256(“Blockchain”**)**=
625da44e4eaf58d61cf048d168aa6f5e492dea166d8bb54ec06c30de07db57e1

2.4. Kriptografija javnog ključa

Kriptografija javnog ključa način je kriptiranja podataka pomoću javnog i privatnog ključa koji su matematički povezani. Javni ključ može biti javno dostupan, no privatni ključ mora se čuvati u tajnosti. Skup podataka, tj. poruka, kriptira se javnim ključem te se može dekriptirati samo odgovarajućim privatnim ključem.



Slika 1. Kriptografija javnoga ključa

Izvor: https://en.wikipedia.org/wiki/Public-key_cryptography#/media/File:Public_key_encryption.svg

Kako bi transakcija bila važeća i dodana u blok, mora biti potpisana digitalnim potpisom te se digitalni potpis obavlja pomoću privatnoga ključa. Drugim riječima, privatni ključ koristi se za plaćanje kriptovalutom. Pomoću javnoga ključa moguće je provjeriti je li transakcija važeća. Kod *bitcoin blockchain* sustava *bitcoin* adresa naziv je za adresu na koju se može slati *bitcoin* kriptovaluta. Ta adresa kreirana je pomoću javnoga ključa, no nije mu jednaka. [3,6]

2.5. Transakcije

Transakcija u *blockchain* sustavu koji se temelji na kriptovalutama premještanje je određene količine kriptovalute s jednog mjesta na neko drugo mjesto. Ključan su dio *bitcoin blockchain* sustava i ostalih *blockchain* sustava koji koriste kriptovalute jer se takvi *blockchain* sustavi temelje na tome da se razmjena kriptovalute provjerava, obavlja i sprema bez prisutnosti nadležnog tijela kao što je banka.

Transakcija u *bitcoin blockchain* sustavu sastoji se od:

1.) ID transakcije – jedinstvena *hash* vrijednost transakcije
2.) Iznos – količina kriptovalute koja se premješta
3.) Ulaz – sav iznos ili dijelovi iznosa kriptovalute koji se šalje
4.) Izlaz – poslani iznos kriptovalute čiji je novi vlasnik korisnik koji prima kriptovalutu, te, ako postoji, iznos ostatka koji se vraća primatelju.

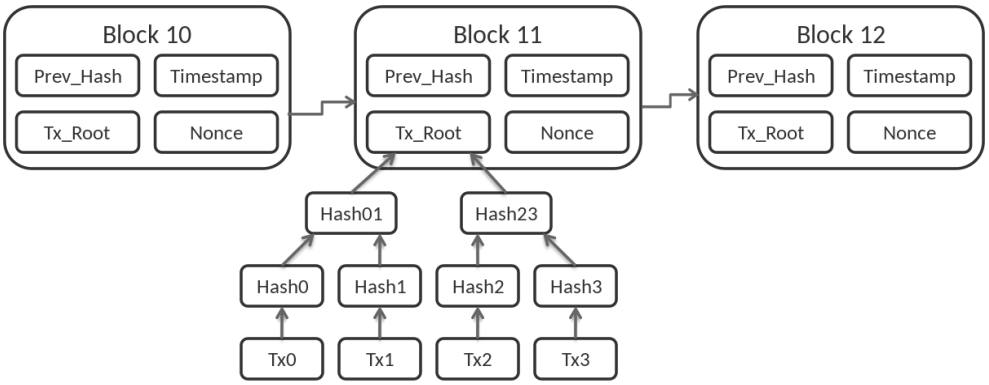
Transakcija može imati više ulaza, tj. više dijelova grupiranih jedinica kriptovalute zvanih UTXO (*unspent transaction output*), i maksimalno dva izlaza, jedan za poslani iznos primatelju, te jedan za ostatak, ako ga ima, koji se vraća pošiljatelju. [3]

2.5.1. UTXO

Kao što je napomenuto, svaka transakcija može imati više ulaza i maksimalno dva izlaza - poslani iznos novom vlasniku i ostatak koji se vraća pošiljatelju. Te izlaze transakcija nazivamo UTXO (engl. *unspent transaction output*). Kada određeni iznos kriptovalute, tj. UTXO, izlazi iz transakcije, tada se nakon provjere on šalje korisniku kojem je dodijeljen. *Bitcoin blockchain* sustav ne prati svaku jedinicu kriptovalute zasebno, nego se u *blockchain* spremaju nepotrošeni izlazi transakcija, koji se opet koriste i kombiniraju kada se koriste kod ulaza u nove transakcije. Tako se na primjer kod plaćanja 1 BTC u nekoj transakciji može povezivati više UTXO izlaza manjih od 1 BTC koje korisnik posjeduje, te ako je iznos veći, vraća mu se ostatak. [4]

2.6. Struktura *blockchain* lanca

Blockchain lanac skup je blokova povezanih u lanac, koji se neprestano širi dodavanjem novih blokova jednog za drugim. Prvi blok u *blockchain* lancu nazivamo „genesis block“. Kreiran je prvi puta u *bitcoin blockchain* lancu od Satoshija Nakamota 2009. godine. Nakon prvog „genesis blocka“ slijede blokovi dodani od korisnika. Svi blokovi osim prvoga sadrže *hash* prethodnoga bloka, koji služi kao referenca na prethodni blok te se tako omogućuje da su blokovi povezani u lanac. *Hash* bloka služi kao njegova glavna oznaka te se dobiva dvostrukim *hashiranjem* zaglavlja bloka SHA-256 algoritmom. *Hash* bloka nije spremljen unutar njega, nego se on izračunava kada je to potrebno, no *hash* prethodnog bloka mora se nalaziti unutar bloka da bi služio kao referenca za povezivanje. Osim *hasha* prethodnog bloka, za provjeru kronološkog reda blokova može služiti i vremenski žig, koji sadrži stvarno vrijeme kada je blok dodan u *blockchain*. [4]



Slika 2. *Blockchain* – lanac blokova

Izvor: *https://en.wikipedia.org/wiki/Blockchain#/media/File:Bitcoin_Block_Data.svg*

2.6.1. Struktura *blockchain* lanca

Na blok unutar *blockchain* lanca možemo gledati kao na spremnik informacija, odnosno transakcija, koji osim tih podataka sadrži i neke dodatne parametre. Struktura bloka može varirati ovisno o kojem *blockchain* sustavu se radi te o njegovoj namjeni. U tablici 1. prikazana je struktura bloka u *bitcoin blockchain* sustavu. [4]

Tablica 1. – Struktura bloka u *bitcoin blockchain* sustavu

Veličina	**Naziv polja**	**Opis**
4 bajta	Veličina bloka	Veličina bloka u bitovima
80 bajta	Zaglavlje bloka	Parametri smješteni u zaglavlju bloka
1 – 9 bajta	Broj transakcija	Ukupan broj transakcija koje se nalaze u bloku
Promjenjivo	Transakcije	Lista transakcija unutar bloka

Izvor: Andreas M. Antonopoulos, *Mastering Bitcoin,* O'Reilly Media Inc., 2015.

2.6.2. Zaglavlje bloka

U bloku je 80 bajta rezervirano za zaglavlje bloka, u kojem su smještene dodatne informacije vezane za blok, kao što je trenutna verzija softvera *blockchaina* kada je blok dodan u lanac, *hash* prijašnjega bloka koji služi kao pokazivač na prethodni blok, *hash* korijena Merkleova stabla u kojem su sažete sve transakcije, vrijeme nastanka bloka, težinska oznaka te *nonce*, broj koji je rješenje zagonetke koju je potrebno riješiti da bi se blok dodao u *blockchain*. [4]

Struktura zaglavlja bloka u *bitcoin blockchainu* opisana je u Tablici 2.

Tablica 2. – Struktura zaglavlja bloka u *bitcoin Blockchain* sustavu

Veličina	**Naziv polja**	**Opis**
4 bajta	Verzija	Broj verzije softvera
32 bajta	*Hash* prijašnjega bloka	Referenca prijašnjemu čvoru u b*lockchainu*
32 bajta	Korijen Merkleova stabla	*Hash* korijena Merkleova stabla svih transakcija bloka
4 bajta	Vremenski žig	Vrijeme kada je blok dodan u *blockchain*
4 bajta	Težinska oznaka	Težina rješavanja problema za dodavanje bloka lancu
4 bajta	*Nonce*	Broj koji je rješenje problema ovoga bloka

Izvor: Andreas M. Antonopoulos, *Mastering Bitcoin,* O'Reilly Media Inc., 2015.

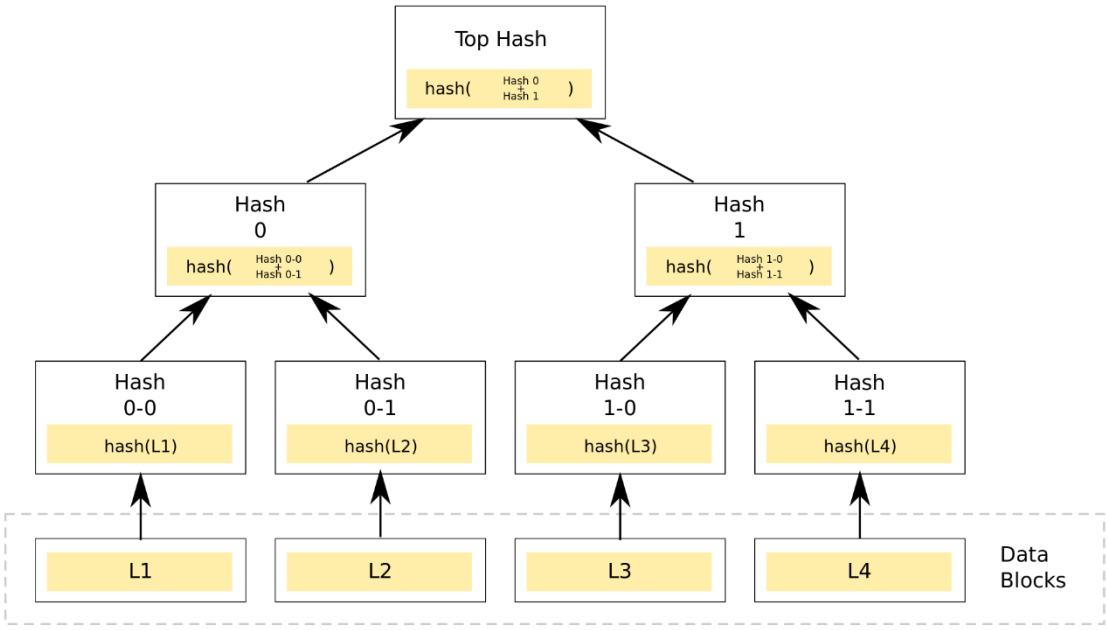
2.7.Merkleovo stablo

U zaglavlju bloka nalazi se *hash* korijena Merkleova stabla.

Merkleovo stablo binarno je *hash* stablo koje ulazne podatke *hashira* te kombinirajući *hasheve* ulaznih podataka njihovim ponovnim *hashiranjem* izrađuje strukturu binarnoga stabla čiji se korijen naziva korijen Merkleova stabla.

U *blockchain* sustavu Merkleovo se stablo koristi da bi se sažele sve transakcije unutar bloka u jedan *hash* čija je veličina 32 bajta, neovisno o tome koliko transakcija sadrži blok, pa se time omogućuje da čvorovi ne moraju imati pohranjen cijeli *blockchain* da bi provjerili validnost neke transakcije, već samo zaglavlja blokova *blockchaina*, s obzirom

na to da se u njima nalazi *hash* korijen Merkleova stabla pomoću kojeg se može provjeriti sadrži li taj blok transakciju. Upravo to mogu raditi laki čvorovi (engl. *lightweight nodes*). Preuzimanjem samo zaglavlja svih blokova *blockchaina* laki čvorovi idu od bloka do bloka te provjeravaju sadrži li *hash* korijena Merkleova stabla *hash* te transakcije. Laki čvor ne može vidjeti tu transakciju, nego samo provjeriti nalazi li se ona u određenom bloku ili ne. Ako neki blok sadrži tu transakciju, znači da je ta transakcija važeća. Taj proces naziva se **Simplified Payment Verification (SPV)**. [3,4]



Slika 3. Merkleovo stablo

Izvor: *https://en.wikipedia.org/wiki/Merkle_tree#/media/File:Hash_Tree.svg*

2.8. Algoritmi za konsenzus

Algoritmi za konsenzus metode su koje opisuju način dodavanja novih blokova u *blockchain* lanac. [3]

2.8.1. *Proof of work*

U *bitcoin blockchain* sustavu dogovoreni konsenzus za dodavanje novog bloka u *blockchain* naziva se *proof of work*. Ova metoda za dodavanje novoga bloka u *blockchain* lanac zahtijeva potrošnju energije i veliku računalnu moć. *Proof of work* temelji se na tome da se čvorovi koje zovemo „rudari“ (engl. *miners*) natječu u tome da prvi pronađu rješenje zagonetke, u čemu ih motivira nagrada koju rudar dobiva kada prvi dođe do rješenja i objavi blok u mrežu. Rješenje zagonetke koje rudari traže je broj koji nazivamo *nonce*. *Nonce* je broj koji, kada se *hashira* SHA-256 algoritmom, daje *hash* rezultat koji na početku sadrži određeni broj nula.

Primjer:

SHA-256(“blockchain” + nonce**)** = *hash* vrijednost koji mora početi s “000000”

SHA-256(“blockchain1”**)** =
db0b9c1cb5e9c680dfff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10

SHA-256(“blockchain100”**)** =

b32f87d6380dbf3b1d13c7ff12e40b270e6788c3481ac42bb3b693edd197bdff

SHA-256(“blockchain10730895”**)** =

000000ca1415e0bec568f6f605fcc83d18cac7a4e6c219a957c10c6879d67587

Broj nula na početku rješenja *hashirane* vrijednosti *nonce* broja utječe na težinu pronalaska rješenja zagonetke. Dodavanjem nula težina zagonetke eksponencijalno raste.

Kod *bitcoin blockchain* sustava težina rješavanja zagonetke regulira se svaka 2 tjedna da bi se održavao standard dodavanja novog bloka u sustav, što iznosi oko 10 minuta. Kad jednom rudar pronađe rješenje zagonetke, blok s rješenjem šalje na mrežu te ga ostali čvorovi provjeravaju. Ostali čvorovi moraju se složiti tako da je rješenje prihvatljivo. Ako jest, dodaju ga u svoju kopiju *blockchain* lanca te šalju blok dalje na provjeru drugim blokovima. Čvor ne mora nužno prihvatiti validan blok, no to mu je u interesu jer najdulji provjereni *blockchain* lanac prihvaća se kao validan *blockchain* te ako želi imati ispravnu kopiju *blokchaina*, čvor prihvaća novo dodani ispravni blok.

Prvobitna nagrada za pronalaženje rješenja i dodavanje novog bloka u *bitcoin blockchain* sustavu je bila 50 BTC, no zamišljeno je da se nagrada prepolovi svakih 210 000 blokova,

pa sada u 2021. godini iznosi 6.25 BTC. Kada će u opticaju biti 21 milijun BTC, taj broj će pasti na 0 i više se neće dodavati novi *bitcoini* u sustav, već će nagrada za objavljivanje bloka u *blockchain* proizlaziti iz naknada za transakcije koje se nalaze unutar bloka. [3,4]

2.8.2. *Proof of stake*

Proof of stake algoritam za konsenzus nastao je idejom da se pokuša izbaciti korištenje velike količine energije i računalne moći da bi se novi blok dodao u lanac. Temelji se na tome da korisnik ulaže određeni broj kriptovalute u sustav, i čim veći udio u sustavu neki korisnik ima, tim je veća vjerojatnost da će on objaviti novi blok. U *proof of stake* algoritmu za konsenzus značajnu ulogu ima pojam ***coin age*.**

Coin age je mjera davanja važnosti kriptovaluti s obzirom na to koliko je dugo neki korisnik posjeduje, a iznosi produkt količine dobivene svote i broja dana koliko je korisnik posjeduje, a mjerna jedinica je *coin-day*.

Primjer:

Korisnik dobiva na 10 jedinica određene kriptovalute.

Prvi dan *coin age* tih 10 jedinica ima *coin age* = 10 * 1 = 10 *coin-days.*

Trideseti dan *coin age* tih istih 10 jedinica ima *coin age* = 10 * 30 = 300 *coin-days*.

Vremensko označavanje sprječava moguće prijevare, tj. izmjenu *coin agea*. Jednom potrošeni novac gubi svoj dosadašnji *coin age* te se resetira na nulu. Kod *proof of stake* algoritma za konsenzus dodaje se novi tip transakcije pod imenom ***coinstake***, kod koje korisnik sam sebi uplaćuje određeni broj jedinica kriptovalute kao polog kojemu je poželjan čim veći *coin age*. Korisnici i kod *proof of stake* algoritma za konsenzus trebaju pronaći rješenje određene *hash* vrijednosti koja se generira za svakog korisnika posebno, no to ne iziskuje toliko vremena i računalne snage kao kod *proof of work* algoritma za konsenzus da bi se dodao novi blok u lanac jer težina pronalaska rješenja ovisi o pologu i starosti novca koji neki korisnik sadrži u *coinstakeu*, pa će tako korisnici s velikim pologom i starosti novca vrlo brzo pronaći rješenje i objaviti novi blok. Kada neki korisnik objavi novi blok, za to je nagrađen ovisno o tome koliki je *coin age* njegova novca iznosio, i *coin age* uloga mu se resetira na nulu, pa se na taj način smanjuju njegove šanse da u nekom kraćem vremenu opet objavi novi blok. [7]

3. Kriptovalute

Kriptovaluta je oblik digitalne valute koja se nalazi u sigurnom kriptografski zaštićenom sustavu. Ta digitalna valuta ima određenu tržišnu vrijednost pa se može mijenjati za stvarni novac te ima kupovnu moć. Pomoću kriptografije kriptovaluta se može koristiti za sigurnu razmjenu, a pogodna je i za izradu decentraliziranog sustava bez nadležnog tijela. [8]

3.1. *Bitcoin* (BTC)

Kao što je već spomenuto, Satoshi Nakamoto je 2008. objavio rad u kojemu opisuje način realiziranja kriptovalute pod imenom *bitcoin*, kojom se uklanja potreba za središnjim regulatornim tijelom kao što je banka, te 2009. godine objavljuje prvi „genesis block“ blok *blockchain* lanca. Prvi rudar koji je dodao sljedeći blok nakon „genesis blocka“ dobio je nagradu koja je tada iznosila 50 BTC te je prvih 50 BTC izašlo u opticaj. Svakih 210 000 blokova nagrada za dodavanje bloka lancu prepolovit će se, a kad dođe na nulu, rudari će kao nagradu za dodavanje bloka lancu dobivati naknade iz transakcija koje se nalaze u bloku. Transakcije na koje se plaća veća naknada imat će veći prioritet da se čim prije dodaju u *blockchain*. Za vrijeme pisanja ovog rada jedan bitcoin iznosi oko 45 000 USD. [4, 9]

3.2. *Ether* (ETH)

Ether je kriptovaluta koja je nastala na *blockchainu* pod imenom Ethereum. Ethereum *blockchain* je *blockchain* sustav koji svojim korisnicima omogućuje da pišu aplikacije poznate kao „pametni ugovori“ (engl. *smart contracts*), koje će se izvršavati na Ethereum *blockchainu*. Kao i kod *bitcoin blockchain* sustava, Ethereum *blockchain* sustav također koristi *proof of work* algoritam za konsenzus, no zbog toga što su se rudari kod *bitcoin blockchain* lanca počeli koristiti posebno izrađenim hardverskim sustavima koji imaju veliku računalnu moć, kod Ethereum *blockchian* sustava problemi koji se moraju riješiti za dodavanje novih blokova dizajnirani su tako da su, osim računalne moći, i memorijski zahtjevni. Stoga, zahtjevom većeg memorijskog prostora smanjuju dominaciju rudara koji se koriste hardverskim sustavima posebno izrađenim za *rudarenje*. [10]

3.2.1. *Ethereum virtual machine*

Ethereum virtual machine (EVM) je okruženje na kojem se izvršavaju pametni ugovori Ethereum *blockchain* sustava. Kada korisnici koriste pametne ugovore pisane u nekom od viših programskih jezika kao što je *Solidity*, EVM je zadužen za kompilaciju tih pametnih ugovora u EVM *bytecode* jezik te ih tada može izvršavati. Svaki čvor Ethereum *Blockchain* sustava ima pokrenut EVM i izvršava sve instrukcije, što je neefikasna metoda što se tiče brzine rada sustava, no to omogućuje da je sustav decentraliziran, da nema prekida u radu sustava i da se podatci u sustavu ne mogu mijenjati. [10, 11]

3.2.2. Pametni ugovori

Ključan dio Ethereum *blockchain* sustava su pametni ugovori. Pametan ugovor je ustvari kôd napisan od korisnika *blockchain* sustava kojim se mogu definirati uvjeti prije nego se neka transakcija izvrši. Jedan primjer pametnog ugovora bio bi da se premjesti određena količina kriptovalute s računa korisnika samo ako on posjeduje više od određene količine kriptovalute propisane pametnim ugovorom, ili da se šalje određeni broj kriptovalute na račun nekog drugog korisnika određeni dan u mjesecu. Navedeni uvjeti zadaju se programskim kodom, no korisnici mogu pronaći i već gotove pametne ugovore dostupne na mreži te se njima koristiti, a da ih sami ne programiraju. Najpoznatiji programski jezik za pisanje pametnih ugovora u Ethereum *blockchain* sustavu je Solidity. [11, 12]

3.3. *Litecoin* (LTC)

Litecoin kriptovaluta nastala je kao odgovor na vremenski dugo trajanje obavljanja transakcije na *bitcoin blockchain* sustavu. *Litecoin blockchain* sustav objavljen je u listopadu 2011. godine. Kreator Charlie Lee izradio je *Litecoin blockchain* sustav koristeći kôd *bitcoinova blockchain* sustava i njegovim nadograđivanjem. Umjesto SHA-256 algoritma za *hashiranje*, *Litecoin blockchain* sustav koristi Scrypt algoritam, koji kod pronalaska rješenja *proof of work* algoritma za konsenzus iziskuje više memorijskog prostora od SHA-256 algoritma. U *Litecoin blockchain* sustavu objava novog bloka traje

znatno kraće nego kod *bitcoin blockchaina*, oko 2,5 minute. Trenutna nagrada za objavljivanje novog bloka iznosi 12.5 LTC, a maksimalan broj kriptovalute koji će biti u opticaju iznosit će 84 milijuna. [13]

3.4. *Ripple* (XRP)

Ripple blockchain sustav osmislio je 2011. godine tim od tri člana: Davida Schwartza, Arthura Britta i Jeda McCaleba, koji su 2012. godine objavili *blockchain* u javnost te otvorili tvrtku koja se danas naziva *Ripple*. Osnivači *Ripple blockchain* sustava su prije izdavanja sustava javnosti stavili u opticaj 100 bilijuna XRP, od kojih su 20 bilijuna podijelili među sobom, a 80 su bilijuna XRP rezervirali za razvoj i korištenje sustava. Kod *Ripple blockchain* sustava korisnici ne obavljaju *proof of work* algoritam za konsenzus da bi zaradili XRP kriptovalutu, već *Ripple* stavlja određenu količinu kriptovalute na tržište kada to odluči učiniti te ga korisnici *Ripple blockchain* sustava mogu kupiti. *Ripple blockchain* sustav osmišljen je da bi se koristio kao način plaćanja između banaka iz različitih država diljem svijeta. Transakcije na *Ripple blockchain* sustavu obavljene su za manje od 5 sekundi. [14,15]

4. Upotreba *blockchain* tehnologije

Blockchain tehnologija nastala je kao sustav plaćanja kriptovalutom te se danas sve više i više koristi kao mogućnost plaćanja kod kupnje i zahtijevanja usluga. Osim kod plaćanja, *blockchain* tehnologija može se koristiti i za realizaciju poslovnih sustava, kao i u druge svrhe. [3,16]

4.1. Plaćanje usluga i proizvoda kriptovalutama

Danas već mnoge tvrtke koriste kriptovalute kao mogući način plaćanja za njihove proizvode i usluge. Microsoft je još 2014. godine omogućio kupnju svog digitalnog sadržaja pomoću *bitcoina* na svojoj *Windows phone* i *Xbox* platformi. Ubrzo nakon Microsofta PayPal je na svojoj platformi omogućio plaćanje *bitcoin* kriptovalutom, s ciljem da makne naknade na transakcije plaćanja kreditnim karticama, koje su znatno manje plaćanjem *bitcoin* kriptovalutom. Starbucks u svojim prodavaonicama omogućuje plaćanje putem svoje Starbucks aplikacije na *smartphone* uređajima, na koju je 2020. godine dodao mogućnost da se proizvodi mogu plaćati *bitcoin* kriptovalutom. Plaćanje ne ide izravno, već aplikacijom novčanika Bakkt koja *bitcoin* pretvara u američke dolare. [17]

4.2. Lanac opskrbe

Jedna od grana poslovanja u kojoj je primjenjiva *blockchain* tehnologija sustav je praćenja poslovanja tj. logistika. U nekim ju dijelovima poslovanja već upotrebljavaju Walmart i BMW. Pomoću *blockchain* tehnologije mogu se pratiti proizvodi ili materijali od proizvođača pa sve do potrošača, ili se njome koristiti u skladištenju dobara. Pomoću *blockchain* tehnologije lako je pratiti na kojem se mjestu točno nalazi proizvod ili materijal, kakvo je njegovo stanje, i kada se jednom podatci o njemu promijene, više ih nitko drugi ne može mijenjati. [18]

4.3. Zdravstvo

Blockchain tehnologija mogla bi se implementirati u zdravstveni sustav tako da bi se pacijentima dala mogućnost da drugim institucijama i organizacijama daju uvid u svoje medicinske zapise na način da bi pacijenti izdavali prava za pristup zapisima. [19]

U većini zdravstvenih sustava medicinski zapisi pacijenata su centralizirani i pristup zapisima ograničen je na nadležna tijela zdravstvenog sustava. Pomoću implementiranja *blockchain* tehnologije mogao bi se dizajnirati sustav koji bi rasporedio kontrolu pristupa medicinskoj povijesti pacijenta od strane bolnica, izdavača osiguranja, samog pacijenta ili nekih drugih institucija koje bi zahtijevale medicinsku povijest pacijenta na siguran način koji bi bio otporan na izmjene postojećih zapisa te bi se smanjilo vrijeme potrebno za dohvat medicinske povijesti pacijenta. [16]

4.4. Obrazovanje

Upotrebom *blockchain* tehnologije mogli bi se spremati zapisi o obrazovanju pojedinaca, npr. ocjene, svjedodžbe, diplome, certifikati i mnogi drugi dokumenti koji dokazuju stupanj obrazovanja pojedine osobe. Time bi se obrazovnim ustanovama, vladajućim tijelima i poslodavcima lako mogao pružiti uvid u obrazovanje pojedinca. [20]

4.5. *Internet of things*

U današnje vrijeme elektronički su uređaji svuda oko nas, od računala pa do kućanskih aparata koji sve više imaju mogućnost spajanja na internetsku mrežu kako bismo njima mogli bežično manipulirati. Spajanje velike količine elektroničkih uređaja, tj. stvari koje obavljaju različite poslove i imaju mogućnost zajedničke komunikacije u mrežu nazivamo „internet of things“. *Blockchain* tehnologija pomoću pametnih ugovora pružila bi mogućnost da uređaji u mreži mogu samostalno obavljati mikrotransakcije kriptovalutom ili vrijednim informacijama. Također, pružila bi način spremanja transakcija i informacija između njih na decentraliziran način. [19]

5. Aplikacija

5.1. Korištene tehnologije

5.1.1. .NET platforma i ASP.NET

.NET platforma skupina je biblioteka, alata i programskih jezika za izradu aplikacija. Programski jezici *Visual Basic*, F# te C#, programski jezik korišten u izradi ove aplikacije, pripadaju .NET platformi.

ASP.NET Microsoftov je programski okvir (engl. *framework*) koji se temelji na .NET platformi te joj proširuje funkcionalnost alatima i bibliotekama namijenjenim za razvoj mrežnih aplikacija i servisa. [21]

5.1.2. HTML, CSS i Bootstrap

HTML (engl. *HyperText Markup Language*) je jezik koji se koristi za definiranje strukture mrežne stranice i elemenata koji se nalaze na njoj. [22]

CSS (engl. *Cascading Style Sheets*) je jezik koji služi za opisivanje izgleda HTML elemenata, tj. kako će se elementi definirani HTML-om prikazivati na mrežnoj stranici. [23]

Bootstrap je javni (engl. *open source*) HTML, CSS i JavaScript programski okvir (engl. *framework*) koji pruža već gotove predloške i funkcionalnosti koje je moguće implementirati u dizajn mrežne stranice. [24]

5.1.3. JavaScript

JavaScript je skriptni jezik koji omogućuje implementiranje raznih mogućnosti na mrežnoj stranici. Pomoću njega se dodaje interaktivnost mrežnoj stranici dohvaćanjem stanja HTML elemenata i manipuliranjem njima. [25]

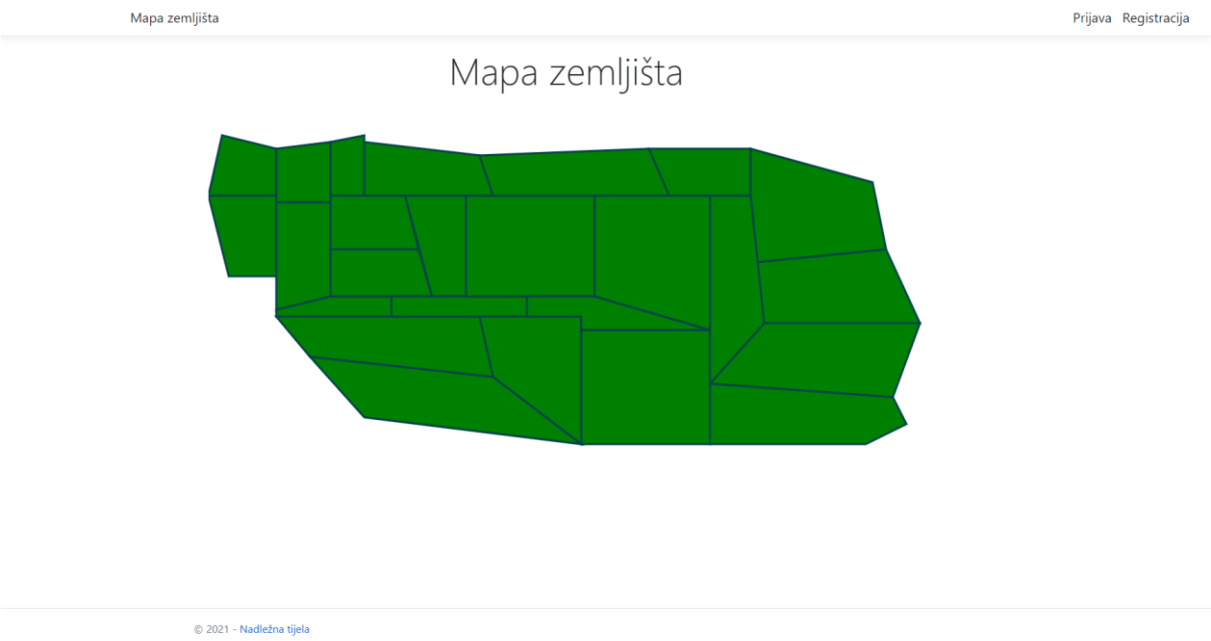
JQuery je biblioteka JavaScript jezika koja pruža već gotove funkcije za postizanje nekih od mogućnosti kao što su jednostavnije upravljanje JavaScript događajima (engl. *event handling*) i AJAX pozivi prema serveru. [26]

5.2. Razine upravljanja aplikacijom

Upravljanje aplikacijom izvršava se na trima korisničkim razinama: kao gost, korisnik i korisnik nadležnog tijela. Ovisno o korisničkoj razini upravljanja aplikacijom, posjetitelj mrežne aplikacije ima na raspolaganju određena prava i mogućnosti.

5.2.1. Gost

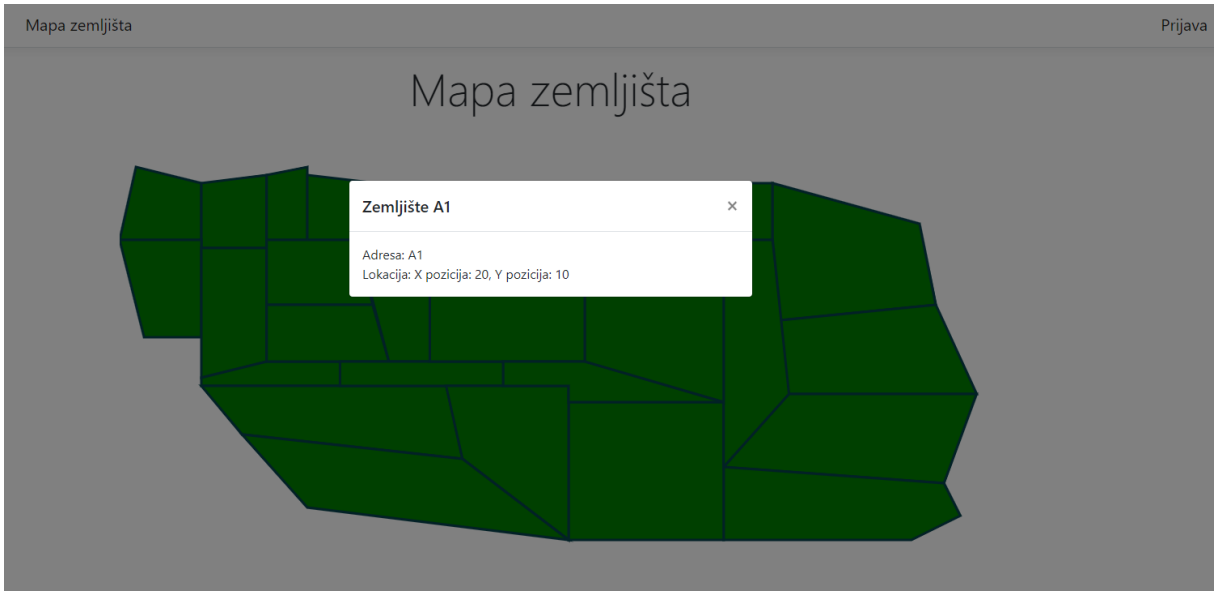
Gostom se smatra svaki posjetitelj mrežne aplikacije koji tijekom svog posjeta nije prijavljen. Gost ima mogućnost posjeta početne mrežne stranice i mogućnost registriranja ili prijave u sustav mrežne aplikacije.



Slika 4. Početna stranica mrežne aplikacije

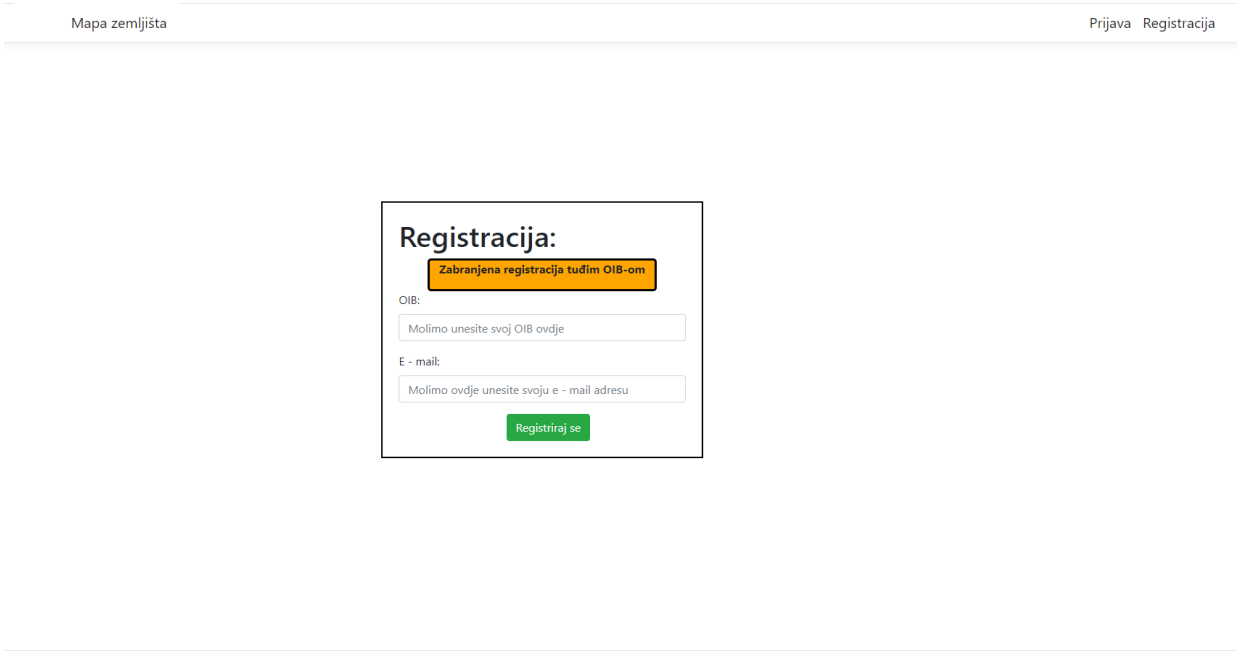
Izvor: Autor

Mapa zemljišta na početnoj stranici aplikacije sastoji se od 25 elemenata tipa s*vg polygon*, na koje se klikom miša otvara prozor o informaciji odabranoga zemljišta.



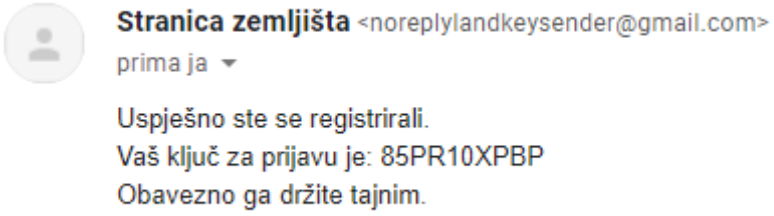
Slika 5. Prozor s podacima o zemljištu
Izvor: Autor

Gost može postati korisnik ukoliko se prijavi ili registrira u sustav. Klikom na gumb „*Registracija*“, gosta mrežne aplikacije preusmjerava se na mrežnu stranicu za registraciju.



Slika 6. Registracija korisnika
Izvor: Autor

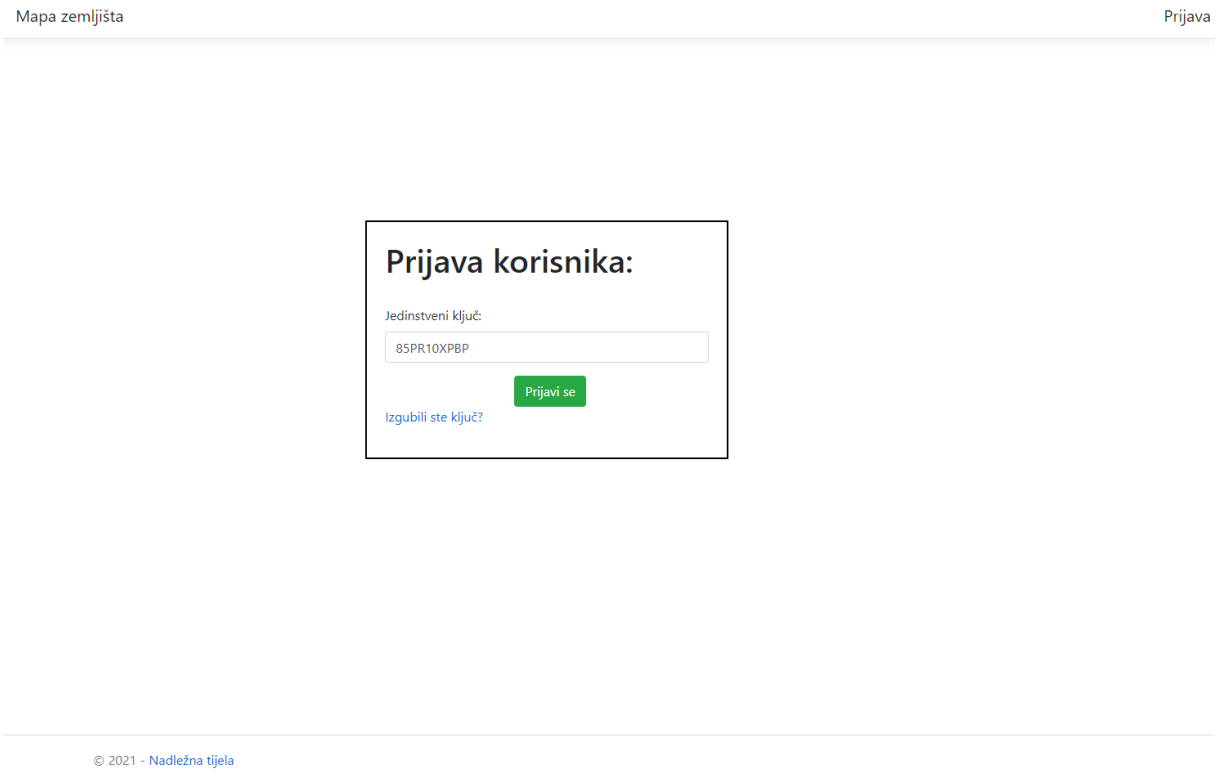
Ispravnim popunjavanjem OIB-a i e-adrese registracija je završena te se na unesenu e-adresu šalje jedinstveni ključ kojim se korisnik prijavljuje u sustav.



Slika 7. E-adresa s ključem za prijavu korisnika

Izvor: Autor

Odabirom gumba „*Prijava*“ na alatnoj traci korisnik se prijavljuje svojim ključem za prijavu dobivenim na e-adresu. Uspješnom prijavom u sustav gost stječe status korisnika.

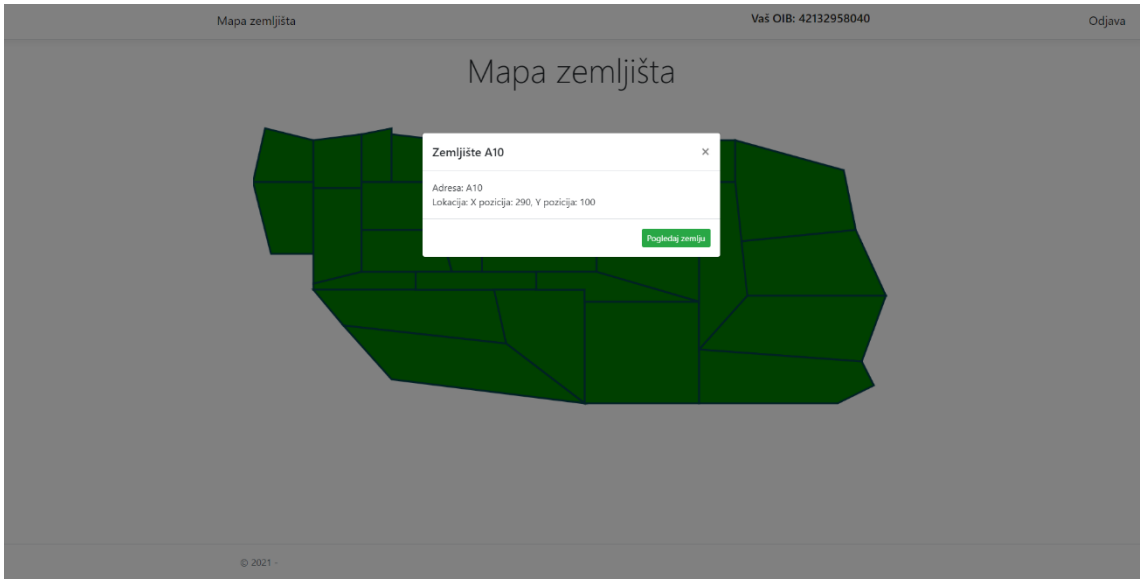


Slika 8. Prijava korisnika

Izvor: Autor

5.2.2. Korisnik

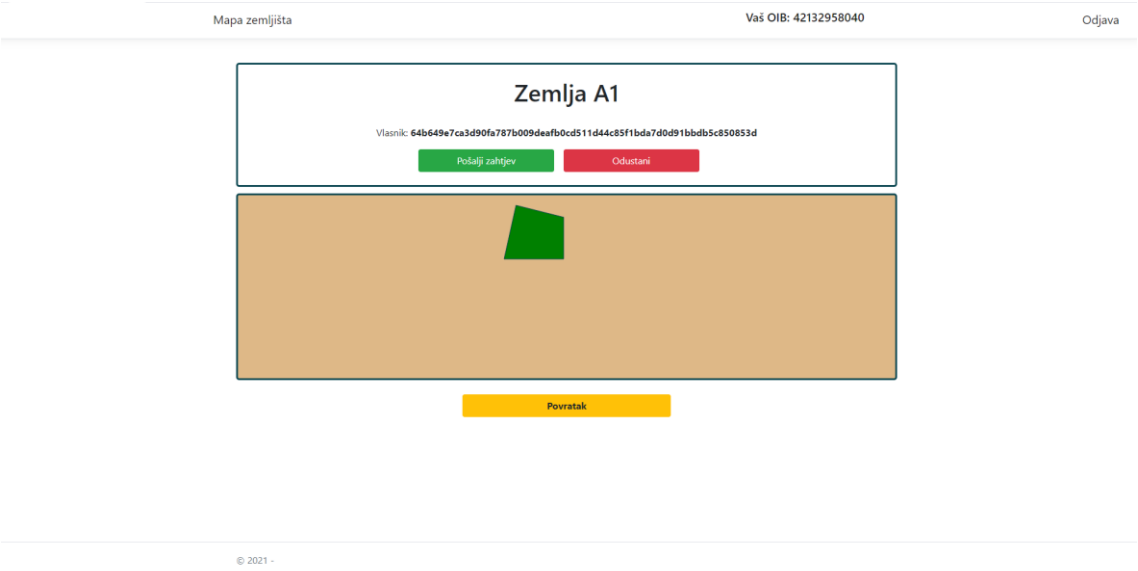
Upravljačka korisnička razina *korisnik* omogućuje korisniku mrežne aplikacije da drugim korisnicima šalje zahtjev za ustupanje vlasništva zemljišta te da prihvaća ili odbija zahtjeve za prijenos svog zemljišta drugim korisnicima, ako ih ima.



Slika 9. Prozor s podat cima o zemljištu nakon prijave

Izvor: Autor

Prijavljenom korisniku dodaje se mogućnost da pogleda odabranu zemlju, čime ga se preusmjerava na stranicu za pregled zemlje.



Slika 10. Mrežna stranica za pregled zemlje

Izvor: Autor

Korisnik odabirom gumba „*Pošalji zahtjev*“ vlasniku zemljišta šalje zahtjev za prijenos vlasništva zemljišta. Kada se vlasnik zemlje prijavi u sustav, očekuje ga crveni gumb „*Zahtjevi*“, čijim se odabirom korisnika preusmjerava na stranicu sa zahtjevima prijenosa njegova zemljišta.

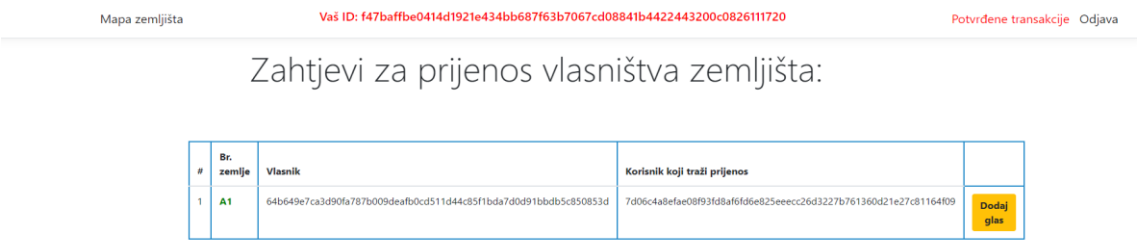
Slika 11. Lista zahtjeva za prijenos vlasništva zemljišta - korisnik

Izvor: Autor

Korisnik vidi koji mu od drugih korisnika šalje zahtjev za prijenos vlasništva zemljišta te ga može prihvatiti ili odbiti. Prihvaćanjem zahtjeva obavlja se digitalni potpis pomoću kriptografije javnog ključa te se transakcija vlasništva zemljišta šalje nadležnim tijelima da ju dodaju u sustav.

5.2.3. Korisnik nadležnog tijela

Korisnik nadležnog tijela je korisnik koji nema mogućnost posredovanja zemljištima mrežne aplikacije, nego je njegova svrha da djeluje kao glasačko tijelo koje glasa nad važećim transakcijama vlasništva zemljišta, koje se prilikom stjecanja većine glasova korisnika nadležnog tijela dodaje u *blockchain* lanac.



Slika 12. Lista zahtjeva za prijenos vlasništva zemljišta – korisnik nadležnog tijela

Izvor: Autor

Klikom na gumb „*Potvrđene transakcije*“ korisnika nadležnog tijela preusmjerava se na mrežnu stranicu koja sadrži listu svih provedenih i važećih transakcija. Odabirom gumba „*Dodaj glas*“ pridodaje se glas tog korisnika nadležnog tijela transakciji uz koju se gumb nalazi. Nakon što je dodao svoj glas, čeka se da i drugi korisnici nadležnog tijela glasaju. Ukoliko je skupljena većina glasova, transakcija se dodaje u *blockchain*.

5.3. *Blockchain* aplikacija

Aplikacija posredovanja vlasništva zemljišta koristi privatan tip *blockchain* tehnologije (engl. *permissioned blockchain*) za praćenje, spremanje i potvrđivanje transakcija vlasništva zemljišta. Kopije *blockchain* lanca održavaju korisnici nadležnih tijela. Oni su zaduženi za dodavanje novih blokova u lanac pa na njih možemo gledati kao puni čvor (engl. *full node*) *blockchain* mreže. Kada se postigne konsenzus za dodavanje novog bloka u *blockchain* lanac, tj. kada važeća transakcija sakupi više od polovine broja mogućih glasova, blok je prihvaćen i ažuriraju se kopije *blockchain* lanca svih korisnika nadležnog tijela.

5.3.1. Transakcija vlasništva zemljišta

Ključan dio *blockchain* lanca aplikacije posredovanja vlasništva zemljišta transakcije su vlasništva zemljišta. U njima su zapisane pojedinosti o transakciji, poput *hasha* transakcije, vremena stvaranja transakcije, dosadašnjeg vlasnika zemljišta i korisnika koji je zatražio prijenos vlasništva zemljišta.

```
public class LandTransaction
{
    public string transactionHash { get { return Crypto.ComputeSHA256(landOwnerHash + landRequesterHash + landId.ToString()); } }
    public string landOwnerHash { get; set; }
    public string landRequesterHash { get; set; }
    public string landId { get; set; }
    public Status transactionStatus { get; set; }
    public string location { get; set; }
    public DateTime timestamp { get; set; }
    public byte[] digitalSignature { get; set; }
    public byte[] publicKey { get; set; }
    public List<string> authorityUsersThatVoted { get; set; }
    public int votes { get; set; }

    public LandTransaction(string landOwnerHash, string landRequesterHash, string location, Land land)
    {
        this.landOwnerHash = landOwnerHash;
        this.landRequesterHash = landRequesterHash;
        this.landId = land.ToString();
        this.transactionStatus = Status.Request;
        this.location = location;

        this.digitalSignature = null;
        this.publicKey = null;
        this.authorityUsersThatVoted = new List<string>();
        this.votes = 0;

        timestamp = DateTime.Now;
    }


}
```

Slika 13. Klasa LandTransaction

Izvor: Autor

Bitan dio transakcije zemljišta je digitalan potpis. Kada vlasnik zemljišta prihvati zahtjev na transakciju, generiraju se javni i privatni ključ te se njima kreira digitalni potpis koji se zajedno s javnim ključem sprema u transakciju radi moguće provjere. Podatak koji se potpisuje je *hash* transakcije. Ispravan digitalan potpis nužan je da bi se transakcija smatrala važećom.

```
[HttpPost]
public IActionResult AcceptLandTransaction(string transactionHash)
{
    var keyPair = Crypto.GenerateRandomKeyPair();
    string dataToSign = transactionHash;

    var signature = Crypto.SignData(dataToSign, (RsaKeyParameters)keyPair.Private);

    List<LandTransaction> pendingTransactions = Helper.GetObjectListFromJson<LandTransaction>(ObjectType.PendingLandTransactions);
    LandTransaction landTransaction = pendingTransactions.Find(element => element.transactionHash == transactionHash);

    pendingTransactions.RemoveAll(element => element.transactionHash == transactionHash);
    pendingTransactions.RemoveAll(element => element.landId == landTransaction.landId);

    landTransaction.digitalSignature = signature;

    SubjectPublicKeyInfo publicKeyInfo = SubjectPublicKeyInfoFactory.CreateSubjectPublicKeyInfo(keyPair.Public);
    byte[] serializedPublicBytes = publicKeyInfo.ToAsn1Object().GetDerEncoded();

    landTransaction.publicKey = serializedPublicBytes;

    RsaKeyParameters publicKey = (RsaKeyParameters) PublicKeyFactory.CreateKey(landTransaction.publicKey);

    landTransaction.transactionStatus = Status.Signed;

    pendingTransactions.Add(landTransaction);
    Helper.WriteObjectListToJson<LandTransaction>(pendingTransactions, ObjectType.PendingLandTransactions);

    return Json("OK");
}
```

Slika 14. Metoda za potvrdu zahtjeva prijenosa vlasništva zemljišta od strane vlasnika

Izvor: Autor

5.3.2. Dodavanje transakcija u blok

Kao što je već napomenuto, potpisane transakcije vlasništva zemljišta dodaju se u *blockchain* prilikom skupljanja većine glasova korisnika nadležnog tijela. Kada korisnik nadležnog tijela doda svoj glas određenoj transakciji, izvršava se provjera digitalnog potpisa te transakcije. Prilikom uspješne potvrde digitalnog potpisa dodaje se jedan glas transakciji te se provjerava je li postignuta glasačka većina. Ako jest, transakcija se dodaje u novi blok *blockchain* lanca.

```
[HttpPost]
public IActionResult AddVoteForPendingTransaction(string transactionHash)
{
    List<LandTransaction> pendingTransactions = Helper.GetObjectListFromJson<LandTransaction>(ObjectType.PendingLandTransactions);

    LandTransaction landTransaction = pendingTransactions.Find(tr => tr.transactionHash == transactionHash);

    pendingTransactions.RemoveAll(tr => tr.transactionHash == transactionHash);

    bool success = Crypto.VerifySignature(landTransaction.transactionHash, landTransaction.digitalSignature,
                                          (RsaKeyParameters)PublicKeyFactory.CreateKey(landTransaction.publicKey));

    if (success == true)
    {
        landTransaction.votes += 1;
        landTransaction.authorityUsersThatVoted.Add(Crypto.ComputeSHA256(HttpContext.Session.GetLoggedUserHash()));

        if (((float)landTransaction.votes/(float)NUMBER_OF_AUTHORITY_USERS) < 0.5f)
        {
            landTransaction.transactionStatus = Status.Voting;
            pendingTransactions.Add(landTransaction);
        }
        else
        {
            landTransaction.transactionStatus = Status.Verified;
            Blockchain.Blockchain.AddBlock(landTransaction);
        }
    }

    Helper.WriteObjectListToJson<LandTransaction>(pendingTransactions, ObjectType.PendingLandTransactions);

    return Json("OK");
}
```

Slika 15. Metoda za dodavanje novog bloka s transakcijama u *blockchain*

Izvor: Autor

Nakon dodavanja novog bloka *blockchain* se nanovo ažurira za svakog korisnika nadležnog tijela.

```
public static void RewriteBlockchains(Block newBlock)
{
    List<AuthorityUser> authorityUsers = Helper.GetObjectListFromJson<AuthorityUser>(ObjectType.AuthorityUser);

    List<Block> currentBlockchain = GetObjectListFromJson<Block>(ObjectType.Blockchain);
    currentBlockchain.Add(newBlock);

    foreach (AuthorityUser authorityUser in authorityUsers)
    {
        string blockchainPath = GetBlockchainPath(authorityUser.authorityUserLoginKeyHash);

        string json = JsonConvert.SerializeObject(currentBlockchain, Formatting.Indented);
        System.IO.File.WriteAllText(blockchainPath, json);
    }
}
```

Slika 16. Metoda za ažuriranje *blockchain* lanca u slučaju promjene (dodavanja novog bloka)

Izvor: Autor

5.3.3. Dohvaćanje zemljišta iz *blockchaina*

Da bi aplikacija dohvatila najnovije informacije o vlasništvima svih zemljišta, provjerava *blockchain* lanac od zadnjeg bloka prema prvom i za svako zemljište dohvaća njezinu zadnju transakciju.

```
public static List<LandTransaction> GetLatestLandDataList()
{
    List<LandTransaction> result = new List<LandTransaction>();

    for(int i = blockchain.IndexOf(blockchain.Last()); i >= 0; i--)
    {
        if (result.Count() > 0)
        {
            foreach (LandTransaction land in blockchain.ElementAt(i).data)
            {
                if (result.Any(element => element.landId == land.landId))
                {
                    continue;
                }
                else
                {
                    if (i != 0)
                    {
                        SetLandOwner(land);
                    }
                    result.Add(land);
                }
            }
        }
        else
        {
            foreach (LandTransaction land in blockchain.ElementAt(i).data)
            {
                if (i != 0)
                {
                    SetLandOwner(land);
                }
                result.Add(land);
            }
        }
    }

    return result;
}
```

Slika 17. Metoda za dohvaćanje najnovijih informacija o zemljištima

Izvor: Autor

6. Zaključak

Blockchain tehnologija pokazala se revolucionarnom u smislu da se pomoću nje može realizirati način plaćanja bez prisustva središnjeg tijela. Ujedno omogućuje uklanjanje naknada koje naplaćuju banke nad transakcijama. Nakon nekog vremena pronašle su se i druge primjene *blockchain* tehnologije, u slučajevima kad je bitno da su informacije koje ulaze u sustav nepromjenjive, pa su zbog toga neke organizacije odlučile implementirati *blockchain* tehnologiju u sustav svog poslovanja. Zbog svojstva *blockchain* tehnologije postiže se transparentnost pa se time uklanja mogućnost prijevare unutar sustava. Kako se sve više i više aspekata u stvarnome svijetu digitalizira, *blockchain* tehnologija djeluje kao primamljiv sustav kojim se može realizirati sigurno spremanje informacija bez njihova pohranjivanja od nekog upravljačkog tijela.

7. Reference

1. Stuart Haber, W. Scott Stornetta, *How to Time-Stamp a Digital Document*, 1991., https://link.springer.com/content/pdf/10.1007%2F3-540-38424-3_32.pdf
2. Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008., https://bitcoin.org/bitcoin.pdf
3. Dylan Y., Peter M., Nik R., Karen S., *Blockchain Technology Overview*, 2018., https://csrc.nist.gov/CSRC/media/Publications/nistir/8202/draft/documents/nistir8202-draft.pdf
4. Andreas M. Antonopoulos, *Mastering Bitcoin,* O'Reilly Media Inc., 2015.
5. Federal Information Processing Standards, *Secure Hash Standard* (SHS) 2015., https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
6. IBM, *Public Key Cryptography*, 2021., https://www.ibm.com/docs/en/integration-bus/10.0?topic=overview-public-key-cryptography
7. Sunny King, Scott Nadal, *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*, 2012., https://www.chainwhy.net/upload/default/20180619/126a057fef926dc286accb372da46955.pdf
8. Eli Dourado, Jerry Brito, *Cryptocurrency*, 2014., https://www.researchgate.net/publication/298792075_Cryptocurrency
9. Coin Market Cap, *Bitcoin*, 8.8.2021., https://coinmarketcap.com/currencies/bitcoin/
10. Ethereum community, *What is Ethereum*, 2016., https://ethdocs.org/en/latest/introduction/what-is-ethereum.html
11. Ethereum community, *Contracts,* 2016., https://ethdocs.org/en/latest/contracts-and-transactions/contracts.html
12. IBM, *What are smart contracts on blockchain*, https://www.ibm.com/topics/smart-contracts
13. Christian Ariely, *What is Litecoin*, https://coin.info/litecoin/
14. Ripple, 2021., https://ripple.com/xrp/
15. CoinDesk, *About Ripple*, 2021., https://www.coindesk.com/price/xrp
16. Manav Gupta, *Blockchain for dummies*, IBM Limited Edition, 2017.

17. Andrew Lisa, *10 Major Companies That Accept Bitcoin*, 2021., https://finance.yahoo.com/news/10-major-companies-accept-bitcoin-110009655.html
18. Forbes, *Blockchain 50*, 2012., https://www.forbes.com/sites/michaeldelcastillo/2020/02/19/blockchain-50/?sh=2ead8e707553
19. Michael C., Jonah C, Gary G., Simon J., Neha N., *The Impact of Blockchain Technology on Finance: A Catalyst for Change*, 2018.
20. Techwire Asia, *Here's how blockchain could transform higher education*, 2021., https://techwireasia.com/2021/04/heres-how-blockchain-could-transform-higher-education/
21. Microsoft, *What is ASP .NET*, 2021., https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet
22. MDN Web Dock, *HTML*, 2021., https://developer.mozilla.org/en-US/docs/Web/HTML
23. MDN Web Dock, *CSS*, 2021., https://developer.mozilla.org/en-US/docs/Web/CSS
24. MDN Web Dock, *Bootstrap*, 2021., https://developer.mozilla.org/en-US/docs/Glossary/Bootstrap
25. MDN Web Dock, *JavaScript*, 2021., https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
26. JQuery, *What is jQuery?*, 2021., https://jquery.com/

8. Popis slika i tabele

8.1. Slike

Slika 1. *Kriptografija javnog ključa*

Slika 2. *„Blockchain“ – lanac blokova*

Slika 3. *Merkleovo stablo*

Slika 4. *Početna stranica mrežne aplikacije*

Slika 5. *Prozor s podatcima o zemljištu*

Slika 6. *Registracija korisnika*

Slika 7. *E-adresa s ključem za prijavu korisnika*

Slika 8. *Prijava korisnika*

Slika 9. *Prozor s podatcima o zemljištu nakon prijave*

Slika 10. *Mrežna stranica za pregled zemlje*

Slika 11. *Lista zahtjeva za prijenos vlasništva zemljišta - korisnik*

Slika 12. *Lista zahtjeva za prijenos vlasništva zemljišta – korisnik nadležnog tijela*

Slika 13. *Klasa LandTransaction*

Slika 14. *Metoda za potvrdu zahtjeva prijenosa vlasništva zemljišta od strane vlasnika*

Slika 15. *Metoda za dodavanje novog bloka sa transakcijama u „blockchain“*

Slika 16. *Metoda za ažuriranje „blockchain“ lanca u slučaju promjene (dodavanja novog bloka)*

Slika 17. *Metoda za dohvaćanje najnovijih informacija o zemljištima*

8.2. Tabele

Tablica 1. – *Struktura bloka u „bitcoin blockchain“ sustavu*

Tablica 2. – *Struktura zaglavlja bloka u „bitcoin blockchain“ sustavu*