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

SUSTAV ZA GLASANJE POMOĆU BLOCKCHAIN TEHNOLOGIJE

Seminarski rad iz kolegija Blockchain tehnologija i kriptovalute

Luka Lovretić i Marko Budimir

Osijek, 2023.

SADRŽAJ

1. UVOD

U današnjem digitalnom dobu, gdje su povjerenje, transparentnost i sigurnost ključni za održavanje integriteta kritičnih sustava, tradicionalni izborni sustavi suočavaju se s brojnim izazovima. Ti sustavi često postaju žrtvom problema vezanih uz prijevare i manipulaciju. Kao rezultat toga, sve je veća potreba za sigurnijom i pouzdanijom alternativom koja može revolucionirati način na koji provodimo izbore. Tu ulogu preuzima tehnologija blockchain-a. Blockchain, prvotno stvoren za olakšavanje kriptovaluta poput Bitcoina, izrastao je kao revolucionarna i transformirajuća tehnologija s ogromnim potencijalom u različitim područjima. Njegova decentralizirana i transparentna priroda, zajedno s karakteristikama poput nepromjenjivosti i mehanizama konsenzusa, čini ga idealnim kandidatom za rješavanje ranjivosti tradicionalnih izbornih sustava. Iskorištavanjem tehnologije blockchain-a, potencijalno možemo stvoriti sigurniji, transparentniji i učinkovitiji izborni sustav koji osigurava integritet demokratskog procesa. Iako tehnologija blockchaina pruža velika obećanja, nije bez briga i kritika. Pitanja privatnosti, skalabilnosti i potencijalne ranjivosti među ključnim su pitanjima koja se postavljaju u vezi s izbornim sustavima temeljenim na blockchain tehnologiji.

2. TRADICIONALNI SUSTAVI ZA GLASANJE

Tradicionalni sustavi za glasanje su oni koji se temelje na fizičkom odlasku birača na biračko mjesto, gdje se identificiraju i dobiju glasački listić koji popunjavaju i ubacuju u glasačku kutiju. Tradicionalni sustavi za glasanje mogu biti različiti po načinu popunjavanja i brojanja listića, kao i po vrsti izbora i izbornog sustava.

2.1. Ograničenja i ranjivosti

Tradicionalni izborni sustavi, unatoč širokoj primjeni, pogođeni su brojnim ograničenjima i ranjivostima koje narušavaju integritet izbornog procesa. Ti sustavi često se oslanjaju na papirnate glasačke listiće ili elektroničke glasačke strojeve, koji su podložni raznim oblicima prijevare i manipulacije. Jedan od glavnih izazova kod tradicionalnih izbornih sustava je mogućnost lažnog predstavljanja birača ili višestrukog glasanja. U sustavima s papirnatim listićima, pojedinci mogu predati više listića ili se predstaviti kao drugi birači s pravom glasa, čime se narušava točnost i pravednost rezultata izbora. Slično tome, elektronički glasački strojevi su podložni hakiranju ili manipulaciji, što omogućuje zlonamjernim pojedincima manipuliranje brojanjem glasova. Osim toga, tradicionalni izborni sustavi nedostaju transparentnosti, što otežava osiguravanje točnosti u glasanju i brojanju glasova. U sustavima s papirnatim listićima, proces ručnog brojanja i bilježenja glasova može dovesti do ljudske pogreške i omogućiti namjerne ili nenamjerne pogreške. S druge strane, elektroničkim glasačkim strojevima često nedostaje transparentnost u unutarnjem funkcioniranju, što otežava provjeru točnosti rezultata. Osim toga, tradicionalni izborni sustavi suočavaju se s izazovima u osiguravanju biračima privatnost. U sustavima s papirnatim listićima, samo glasanje često se obavlja javno, što može dovesti do prisile ili zastrašivanja birača. Elektronički glasački strojevi, unatoč obećanju privatnosti birača, suočavaju se s problemima vezanim uz povredu podataka ili neovlašteni pristup osobnim podacima o glasanju.

2.2. Problemi prijevare i manipulacije

Ranjivosti tradicionalnih izbornih sustava stvaraju brojne probleme povezane sa zlonamjernim aktivnostima. Primjeri kupovine glasova, umetanja listića i prisile prijavljeni su na raznim izborima diljem svijeta, što narušava javno povjerenje u demokratski proces. Manipulacija rezultatima izbora još je jedna važna briga kod tradicionalnih izbornih sustava. Centralizirana priroda tih sustava čini ih podložnima manipulaciji, bilo putem fizičke manipulacije papirnatih listića ili hakiranja elektroničkih glasačkih strojeva. Takva manipulacija narušava pravičnost i točnost izbornih rezultata. U sustavima s papirnatim listićima, ručno rukovanje i brojanje listića mogu biti podložni namjernoj manipulaciji. Slično tome, elektronički glasački strojevi koji se oslanjaju na zatvoreni izvorni kod softvera otežavaju provjeru temeljnih algoritama i osiguravanje da se glasovi točno bilježe i broje Sveukupno, ograničenja i ranjivosti tradicionalnih izbornih sustava zahtijevaju istraživanje alternativnih rješenja koja mogu riješiti ove probleme i osigurati integritet izbornog procesa. Blockchain tehnologija nudi obećavajuću mogućnost za revoluciju u izbornim sustavima pružajući poboljšanu sigurnost, transparentnost i nepromjenjivost. Iskorištavanjem decentralizirane prirode blockchain-a i njegovih mehanizama suglasnosti, potencijalno se mogu prevladati izazove koje postavljaju tradicionalni izborni sustavi i izgraditi pouzdaniji i učinkovitiji izborni sustav.

3. BLOCKCHAIN TEHNOLOGIJA

Blockchain tehnologija revolucionarni je koncept koji je privukao značajnu pažnju posljednjih godina. To je decentralizirana i distribuirana tehnologija glavne knjige(engl. *Ledger*) koja pruža transparentnost, sigurnost i nepromjenjivost u raznim industrijama. U svojoj srži, blockchain je lanac blokova, gdje svaki blok sadrži popis transakcija. Te transakcije su grupirane zajedno i dodane u blockchain kronološkim redoslijedom. Ključna načela blockchain tehnologije uključuju decentralizaciju, mehanizme konsenzusa i nepromjenjivost.

Prednosti blockchain tehnologije protežu se izvan njenih temeljnih načela. Ona pruža dodatnu sigurnost putem korištenja enkripcije i kriptografskih tehnika, smanjujući rizik od curenja podataka(engl. *data leak*) i neovlaštenog pristupa. Transparentnost koju pruža blockchain omogućuje javnu kontrolu, osiguravajući da proces glasanja ostane otvoren i odgovoran.

3.1. Decentralizacija

Decentralizacija je jedno od temeljnih načela blockchain tehnologije. Za razliku od tradicionalnih centraliziranih sustava gdje centralna vlast ima kontrolu nad sustavom, blockchain djeluje na distribuiranoj mreži računala nazvanoj čvorovi. Ovi čvorovi surađuju kako bi provjerili i zabilježili transakcije, osiguravajući da nijedno pojedinačno tijelo ili skupina nema potpunu kontrolu nad sustavom.

3.2. Mehanizmi konsenzusa

Mehanizmi konsenzusa koriste se u blockchain tehnologiji kako bi se postiglo suglasje o valjanosti transakcija i spriječila prijevara. Najpoznatiji mehanizam konsenzusa je dokaz o radu (engl. *proof-of-work*, PoW), koji zahtijeva od čvorova rješavanje kompleksnih matematičkih zadataka kako bi potvrdili transakcije. Osigurava da većina čvorova u mreži suglasno prihvati redoslijed i zakonitost transakcija, što čini zlonamjernim sudionicima manipulaciju nad sustavom poprilično teškom.

3.3. Nepromjenjivost

Nepromjenjivost je još jedna ključna značajka blockchain tehnologije. Nakon što se transakcija doda u blockchain, gotovo je nemoguće izmijeniti ili izbrisati. To se postiže kriptografskim *hash* funkcijama koje generiraju jedinstvene identifikatore za svaki blok, stvarajući lanac blokova koji su međusobno povezani i nepovratni.

4. SUSTAVI GLASANJA TEMELJENI NA BLOCKCHAIN TEHNOLOGIJI

Izborni sustavi temeljeni na blockchain tehnologiji pružaju inovativno rješenje za prevladavanje ograničenja i ranjivosti tradicionalnih izbornih sustava. Jedna od osnovnih prednosti izbornih sustava temeljenih na blockchain tehnologiji je transparentnost koju pružaju. Zahvaljujući decentraliziranoj prirodi blockchaina, sve transakcije, uključujući glasove, zabilježene su na javnoj knjizi kojoj svi sudionici imaju pristup. To omogućuje biračima da provjere točnost i integritet procesa glasanja, osiguravajući da su njihovi glasovi izraženi onako kako su namjeravali i da nisu podvrgnuti manipulaciji. Osim transparentnosti, izborni sustavi temeljeni na blockchain tehnologiji pružaju i poboljšanu sigurnost. Korištenjem kriptografskih tehnika i decentraliziranih mehanizama konsenzusa, ti sustavi smanjuju rizik od prijevare i manipulacije. Svaki glas sigurno je šifriran i povezan s jedinstvenim digitalnim identitetom birača, što gotovo onemogućuje zlonamjernim sudionicima da izmijene ili krivotvore glasove. Još jedna ključna prednost izbornih sustava temeljenih na blockchain tehnologiji je provjerljivost. Nepromjenjivost blockchain-a osigurava da se jednom kada se glas zabilježi, ne može mijenjati ili brisati. Ta značajka omogućuje jednostavnu i točnu provjeru procesa glasanja, omogućujući neovisnu verifikaciju i potvrdu rezultata. Također olakšava identifikaciju eventualnih nepravilnosti ili nesuglasica, povećavajući ukupno povjerenje i sigurnost u izborni sustav.

Postupak izražavanja i zabilježbe glasova u izbornom sustavu temeljenom na blockchain tehnologiji relativno je jednostavan. Birači prvo stvaraju digitalni identitet koji je sigurno pohranjen na blockchainu. Pri izražavanju svog glasa, digitalni identitet birača koristi se za šifriranje i potvrdu njihovog izbora. Šifrirani glas zatim se bilježi na blockchain-u. Nakon završetka razdoblja glasanja, rezultati se mogu jednostavno izračunati i potvrditi putem podataka.

5. PROGRAMSKO RJEŠENJE

5.1. Implementacija pametnog ugovora

5.1.1. Pametni ugovor

Pametni ugovor je programska kod koja se izvršava na blockchainu, digitalnom registru koji je distribuiran na više računala. Ovi ugovori omogućuju dvjema ili više stranama da se međusobno angažiraju u sigurnom i automatiziranom načinu, bez potrebe za posrednicima poput banaka ili pravnih institucija. Pametni ugovori sadrže uvjete i pravila dogovorena između stranaka, a izvršavaju se samo ako su ispunjeni svi uvjeti. Oni pružaju transparentnost, pouzdanost i neizmjenjivost, jer su informacije zapisane na blockchainu i ne mogu se izbrisati ili mijenjati. Primjene pametnih ugovora su mnogobrojne, uključujući financijske transakcije, upravljanje nekretninama, osiguranje, logistiku i mnoge druge industrije. Ovi ugovori mogu automatizirati procese, smanjiti troškove i smanjiti rizik od prijevara. Da bi se izvršili, pametni ugovori zahtijevaju korištenje kriptovaluta poput Ethera, koji se koristi na Ethereum blockchain-u, jednoj od najpopularnijih platformi za izvršavanje pametnih ugovora. Pametni ugovori predstavljaju inovativnu tehnologiju koja ima potencijal da promijeni način na koji se obavljaju transakcije i ugovori.

5.1.2. Solidity

Solidity je programski jezik koji se koristi za pisanje pametnih ugovora na Ethereum blockchain-u. To je jezik više razine koji je posebno dizajniran za izradu decentraliziranih aplikacija (DApps) i izvršavanje pametnih ugovora. Solidity je statički tipiziran jezik, što znači da programer mora deklarirati vrste podataka koje se koriste u programu. On podržava nasljeđivanje, apstraktne tipove podataka i složene strukture podataka poput nizova i mape. Ovaj jezik ima sličnu sintaksu kao i programski jezik JavaScript, što ga čini pristupačnim programerima koji su već upoznati s JavaScriptom ili sličnim jezicima. Solidity omogućuje programerima da definiraju funkcije, događaje i modifikatore koji mogu nadgledati i kontrolirati izvršavanje pametnih ugovora. Solidity ima ugrađene biblioteke koje pružaju funkcionalnosti poput matematičkih operacija, manipulacije vremenom i kriptografskih funkcija. Programeri mogu koristiti ove biblioteke za izgradnju složenih pametnih ugovora. Da bi se razvili pametni ugovori u Solidity-u, programeri koriste razvojne okoline poput Ethereum Wallet-a ili Solidity Compiler-a za prevođenje koda u bajtkod

koji se izvršava na Ethereum virtualnoj mašini (EVM). Solidity je ključan alat za razvoj pametnih ugovora na Ethereum blockchain-u i omogućuje programerima da iskoriste puni potencijal decentraliziranih aplikacija i blockchain tehnologije.

5.1.3. Hardhat

Hardhat je razvojni okvir (engl. *framework*) za razvoj i testiranje pametnih ugovora na Ethereum blockchain-u. Pruža programerima alate i funkcionalnosti za olakšano upravljanje, kompajliranje, implementiranje i testiranje pametnih ugovora. Hardhat podržava programski jezik Solidity i omogućuje programerima integraciju s Ethereum mrežom kroz svoje razvojno okruženje. Također pruža simulirano lokalno okruženje za testiranje pametnih ugovora bez stvarne interakcije s mrežom. Ovaj razvojni okvir vrlo je prilagodljiv i omogućuje programerima konfiguraciju raznih parametara za svoje projekte. Hardhat podržava automatizirane zadatke, poput kompajliranja pametnih ugovora, izgradnje distribuiranih aplikacija i testiranja s korisnički-definiranim scenarijima. Hardhat također dolazi s velikim skupom biblioteka i alata koji olakšavaju razvoj pametnih ugovora. To uključuje alate za upravljanje depozitima, integraciju s popularnim razvojnim okruženjima i mogućnost generiranja testnih podataka. Ukratko, Hardhat je moćan razvojni okvir koji pomaže programerima razvijanje, testiranje i implementaciju pametnih ugovora na Ethereum blockchain s lakšim upravljanjem projektima i učinkovitijim procesom razvoja.

5.1.4. Kreirani pametni ugovor

Kod sa slike 5.1. predstavlja pametni ugovor za glasanje na Ethereum blockchain-u. Definirana je struktura kandidata s imenom i brojem glasova te polje kandidata. Kroz konstruktor se inicijaliziraju kandidati, postavlja se vlasnik ugovora te se određuje trajanje glasanja. Funkcija dodavanja kandidata omogućuje vlasniku ugovora unos novih kandidata. Glasanje se provodi putem funkcije *vote()* koja provjerava valjanost glasova, a rezultati se pohranjuju u polje kandidata i mapiranje glasača. Također postoje funkcije za dohvat svih kandidata i provjeru statusa glasanja. Ovaj pametni ugovor omogućuje sigurno i autonomno glasanje na blockchainu bez potrebe za posrednicima.

contracts > Voting.sol

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.0;

contract Voting {
    struct Candidate {
        string name;
        uint256 voteCount;
    }

    Candidate[] public candidates;
    address owner;
    mapping(string => bool) public voters;

    uint256 public votingStart;
    uint256 public votingEnd;

    constructor(string[] memory _candidateNames, uint256 _durationInMinutes) {
        for (uint256 i = 0; i < _candidateNames.length; i++) {
            candidates.push(
                Candidate({name: _candidateNames[i], voteCount: 0})
            );
        }
        owner = msg.sender;
        votingStart = block.timestamp;
        votingEnd = block.timestamp + (_durationInMinutes * 1 minutes);
    }

    modifier onlyOwner() {
        require(msg.sender == owner);
        _;
    }

    function addCandidate(string memory _name) public onlyOwner {
        candidates.push(Candidate({name: _name, voteCount: 0}));
    }

    function vote(uint256 _candidateIndex, string memory voterHash) public {
        require(getVotingStatus(), "Voting has ended.");
        require(!voters[voterHash], "You have already voted.");
        require(
            _candidateIndex < candidates.length && _candidateIndex >= 0,
            "Invalid candidate index."
        );

        candidates[_candidateIndex].voteCount++;
        voters[voterHash] = true;
    }

    function getAllVotesOfCandidates() public view returns (Candidate[] memory) {
        return candidates;
    }

    function getVotingStatus() public view returns (bool) {
        return (block.timestamp >= votingStart && block.timestamp < votingEnd);
    }
}
```

Slika 5.1. Kreirani pametni ugovor

5.2. Back-end rješenje projekta

Back-end rješenje služi kao baza podataka za spremanje brojeva osobnih iskaznica osoba s pravom glasa. Ovo rješenje omogućuje provjeru valjanosti broja iskaznice prilikom glasanja putem interneta. Kroz intuitivno sučelje, front-end šalje zahtjeve serveru koji pretražuje bazu podataka i vraća rezultat koji predstavlja prisustvo, odnosno odsustvo broja osobne iskaznice u bazi podataka. Na temelju tih rezultata pretrage, korisnicima se preko front-end sučelja dopušta ili onemogućuje glasanje osobe. Baza podataka kreirana je preko PostgreSQL programa. Slika 5.2. prikazuje cijeli back-end kod.

```
const express = require("express");
const knex = require("knex");
const app = express();
const cors = require("cors");
const port = 8080;

const db = knex({
  client: 'pg',
  connection: {
    host: "127.0.0.1",
    user: "postgres",
    password: "postgres",
    database: "voters"
 }
});

app.use(cors())

app.get("/voter/:id_number", async(req,res) => {
  db.select("*").from("voters").where("id_number", "=", req.params.id_number).then(voters =>{
    res.json(voters.length != 0);
  })
});

app.listen(port, () => {
  console.log("Server listening on port ${port}");
});

```

Slika 5.2. Izgled cijelog back-end koda

U ovom kodu se koristi JavaScript jezik zajedno s Express.js i Knex.js bibliotekama kako bi se izgradio serverski dio aplikacije. Prvo se učitavaju potrebne biblioteke, poput "express" za stvaranje servera, "knex" za rad s bazom podataka te "cors" za omogućavanje komunikacije između različitih domena. Zatim se konfigurira veza s bazom podataka. Koristi se PostgreSQL baza podataka koja je lokalno postavljena na adresi "127.0.0.1" ,tj. „localhost“. Za pristup bazi koriste se korisničko ime "postgres" i lozinka "postgres", a odabrana baza podataka je "voters". Nakon toga definira se ruta "/voter/:id_number" koja će obrađivati HTTP GET zahtjev. Kroz ovu

rutu korisnik može poslati broj osobne iskaznice kao dio URL-a. Tada se iz baze podataka selektiraju svi podaci iz tablice "voters" gdje je broj iskaznice jednak poslanom broju. Rezultati selekcije se šalju kao odgovor klijentu. Konačno, kao odgovor na zahtjev, šalje se vrijednost *true* ako postoji pronađeni rezultat ili *false* ako nema pronađenih rezultata. Zadnji korak je pokretanje servera na određenom *portu* (u ovom slučaju *port* 8080). U konzoli se ispisuje poruka koja potvrđuje da je server uspješno pokrenut.

```
async function vote() {
  let data = true;
   await axios.get("http://localhost:8080/voter/"+idNumber.toString()).then((response) => {
    data = response.data;
  });
```

Slika 5.3. Dohvaćanje broja iskaznice unutar frontend-a

U ovom dijelu koda u front-endu može se vidjeti poziv HTTP zahtjeva prema back-endu kako bi se provjerio broj iskaznice. Definirana je asinkrona funkcija *vote()* koja će se izvršiti prilikom glasanja. Unutar ove funkcije, postavljena je inicijalna vrijednost varijable *data* na *true*. Zatim se koristi Axios biblioteka za izvršavanje GET zahtjeva. *idNumber* je varijabla koja sadrži broj iskaznice pretvoren u string. Ovaj zahtjev dobiva podatke iz back-enda. Korištena je metoda *then()* kako bi se obradio odgovor (*response* varijabla) dobiven od servera. U ovom slučaju, vrijednost *data* se postavlja na vrijednost *response.data*, što znači da će *data* biti *true* ako je broj iskaznice prisutan u bazi podataka ili *false* ako nije prisutan. Kasnije se *data* varijabla koristi za daljnji tijek programa.

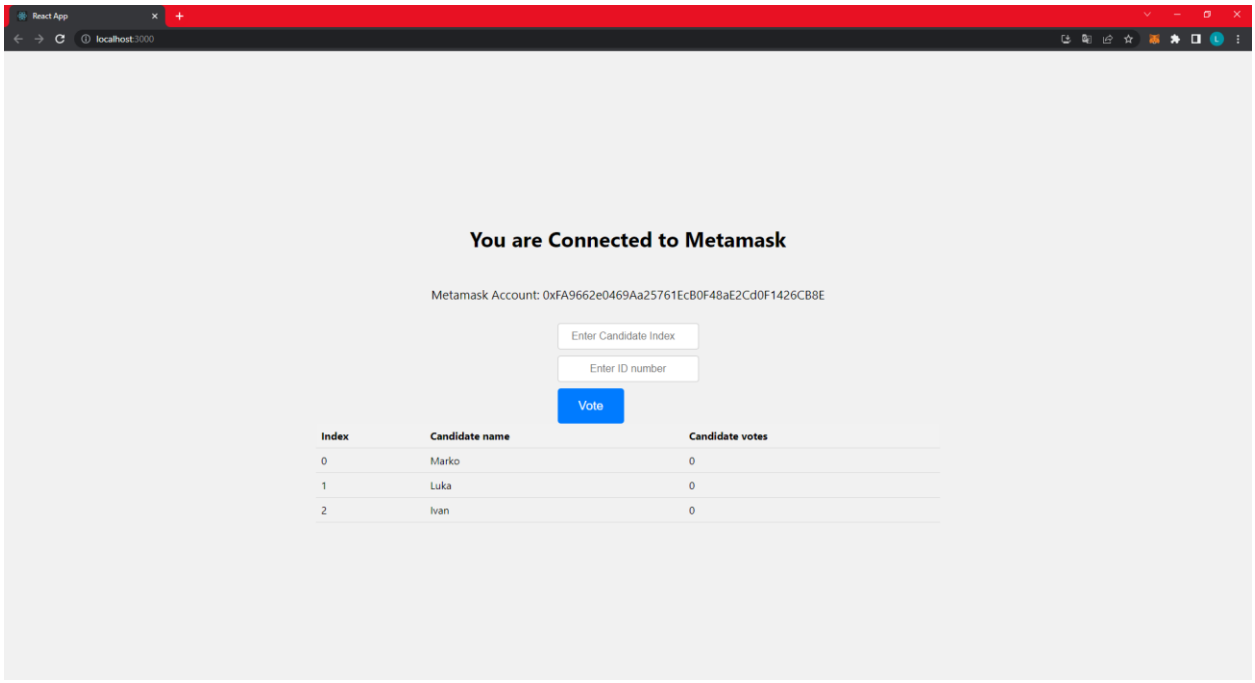
5.3. Front-end rješenje projekta

Front-end omogućuje korisnicima glasanje putem pametnog ugovora na Ethereum mreži. Aplikacija koristi biblioteku ethers za komunikaciju s Ethereum mrežom i interakciju s pametnim ugovorom. Kada korisnik pokrene aplikaciju, može se povezati s Metamaskom (Slika 5.4.), popularnim Ethereum novčanikom. Ako je povezan, korisniku se prikazuje sučelje za glasanje(Slika 5.5.). U suprotnom, prikazuje se sučelje za prijavu(Slika 5.6.). Sučelje za glasanje prikazuje informacije o Metamask računu korisnika, listu kandidata i formu za unos broja koji predstavlja određenog kandidata i identifikacijski broj. Nakon što korisnik unese broj kandidata i

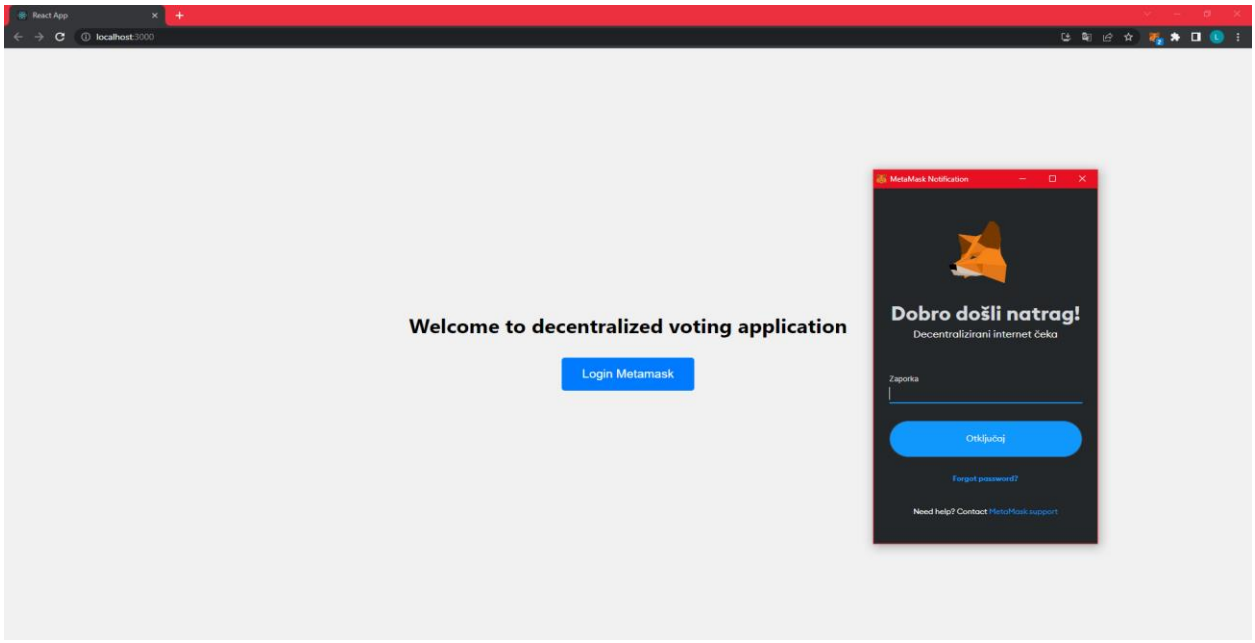
identifikacijski broj, može pritisnuti gumb za glasanje. Kada se pritisne gumb za glasanje, putem API-ja(engl. *Application Programming Interface*)(Slika 5.7.), aplikacija provjerava ima li osoba s unesenim identifikacijskim brojem pravo glasa. Ako osoba ima pravo glasa, aplikacija koristi ethers biblioteku za slanje transakcije na pametni ugovor s odabranim brojem kandidata i identifikacijskim brojem. Ako je glasanje uspješno, aplikacija nakon nekog vremena ažurira trenutne rezultate izbora(Slika 5.8.). Također, aplikacija omogućuje dohvaćanje liste kandidata(Slika 5.9.) putem ethers biblioteke. Kada se dohvate podaci o kandidatima, oni se formatiraju i prikazuju korisniku. Aplikacija koristi različite React kukice(engl. *hook*) poput *useState* i *useEffect* kako bi pratila stanja, reagirala na promjene i izvršavala određene radnje.

```
99  async function connectToMetamask() {
100   if (window.ethereum) {
101     try {
102       const provider = new ethers.providers.Web3Provider(window.ethereum);
103       setProvider(provider);
104       await provider.send("eth_requestAccounts", []);
105       const signer = provider.getSigner();
106       const address = await signer.getAddress();
107       setAccount(address);
108       console.log("Metamask Connected : " + address);
109       setIsConnected(true);
110       canVote();
111     } catch (err) {
112       console.error(err);
113     }
114   } else {
115     console.error("Metamask is not detected in the browser")
116   }
117 }
```

Slika 5.4. *connectToMetaMask* funkcija



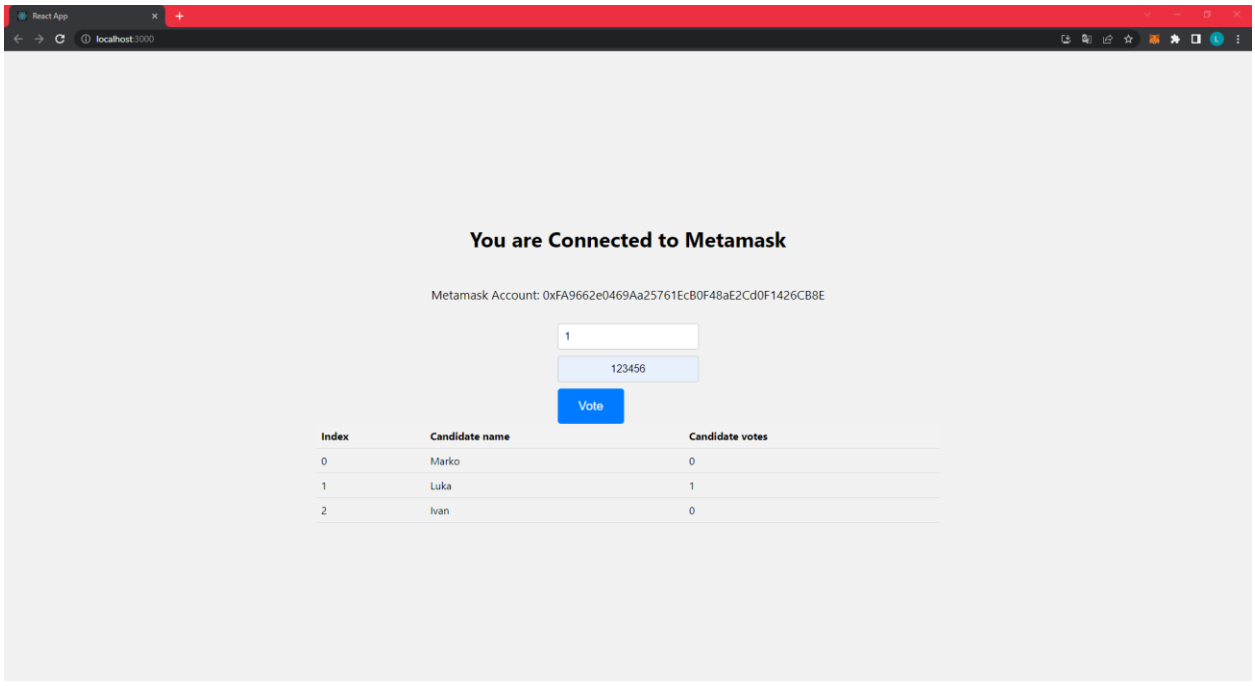
Slika 5.5. Sučelje za glasanje



Slika 5.6. Sučelje za prijavu

```
async function vote() {
  let data = true;
  await axios.get("http://localhost:8080/voter/" + idNumber.toString()).then((response) => {
    data = response.data;
  });
  if (data) {
    setErrorMessage(null);
    const provider = new ethers.providers.Web3Provider(window.ethereum);
    await provider.send("eth_requestAccounts", []);
    const signer = provider.getSigner();
    const contractInstance = new ethers.Contract(
      contractAddress, contractAbi, signer
    );
    try {
      const hashedIdNumber = SHA256(idNumber).toString();
      const tx = await contractInstance.vote(number, hashedIdNumber);
      await tx.wait();
    } catch (error) {
      setErrorMessage(error["reason"].split(":")[1]);
    }
    canVote();
  } else {
    setErrorMessage("Person with this ID number has no right to vote.")
  }
}
```

Slika 5.7. *vote()* funkcija



Slika 5.8. Ažurirano sučelje za glasanje

```
async function getCandidates() {
  const provider = new ethers.providers.Web3Provider(window.ethereum);
  await provider.send("eth_requestAccounts", []);
  const signer = provider.getSigner();
  const contractInstance = new ethers.Contract(
    contractAddress, contractAbi, signer
  );
  const candidatesList = await contractInstance.getAllVotesOfCandidates();
  const formattedCandidates = candidatesList.map((candidate, index) => {
    return {
      index: index,
      name: candidate.name,
      voteCount: candidate.voteCount.toNumber()
    }
  });
  setCandidates(formattedCandidates);
}
```

Slika 5.9. *getCandidates()* funkcija

6. ZAKLJUČAK

Ovaj rad bavi se problemom glasanja u tradicionalnim sustavima i prikazuje jedno rješenje temeljeno na blockchain tehnologiji. U teorijskom dijelu opisuju se ograničenja i ranjivosti tradicionalnih sustava glasanja, kao i problemi prijevare, manipulacije i nedostatka transparentnosti koji se javljaju u takvim sustavima. Također se objašnjava blockchain tehnologija, njezine glavne karakteristike i prednosti, kao i mehanizmi konsenzusa koji omogućuju decentralizaciju i nepromjenjivost podataka na blockchain-u. Analiziraju se postojeći sustavi glasanja temeljeni na blockchain tehnologiji i njihove prednosti i nedostatci u odnosu na tradicionalne sustave. Programski dio rada odnosi se na programsko rješenje koje se sastoji od pametnog ugovora koji implementira logiku glasanja na Ethereum blockchain-u. Pametni ugovor napisan je u programskom jeziku Solidity pomoću alata Hardhat. Pametni ugovor omogućuje registraciju birača, kreiranje izbora, glasanje, prebrojavanje glasova i objavljivanje rezultata na transparentan i siguran način. Blockchain tehnologija pogodna je za implementaciju sustava glasanja koji su otporni na prijevare, manipulacije i napade, te da pružaju veću transparentnost i povjerenje u izborni proces. Ukupno gledajući, blockchain tehnologija nudi veliki potencijal za unapređenje sustava glasanja. Međutim, i dalje postoje izazovi koje treba riješiti, poput skalabilnosti, sudjelovanja korisnika i osiguravanja privatnosti. Daljnja istraživanja i razvoj u području blockchain-a mogli bi dovesti do naprednijih sustava glasanja koji će zadovoljiti visoke standarde demokratskih procesa. U konačnici, implementacija blockchain tehnologije u sustave glasanja može donijeti pozitivne promjene, povećati povjerenje građana u izborni proces i osigurati integritet rezultata. S obzirom na brzi napredak tehnologije i rastuću svijest o potrebi za sigurnijim i transparentnijim sustavima glasanja, očekuje se da će blockchain nastaviti igrati važnu ulogu u demokratskim procesima širom svijeta.

7. LITERATURA

Seeburg, D. (2019). Blockchain and the supply chain: Concepts, strategies and practical applications. Kogan Page.

Chen, H. (2022). Blockchain principles and applications in IoT. Chapman & Hall.

Chen, H. (2022). Blockchain technology: Exploring opportunities, challenges, and applications. CRC Press.

Williams, S. (2022). Blockchain and Web3: Building the cryptocurrency, privacy, and security foundations of the metaverse. Wiley.

Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking digital cryptocurrencies. O'Reilly Media.

Antonopoulos, A. M. (2016). The internet of money. Merkle Bloom LLC.

Ammous, S. (2018). The bitcoin standard: The decentralized alternative to central banking. Wiley.

Tapscott, D., & Tapscott, A. (2016). Blockchain revolution: How the technology behind bitcoin is changing money, business, and the world. Penguin Random House.

Al-Fuqaha, M. A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2021). Blockchain for electronic voting system—Review and open research challenges. IEEE Access, 9, 107936–107956. https://doi.org/10.1109/ACCESS.2021.3107640

Khajashareef, S. K., Raja Rajeswari, T. S., Sandhya, N., & Chinnasamy, P. (2021). The application of the blockchain technology in voting systems: A comprehensive review. ACM Journal of Data and Information Quality, 13(2), 1–28. https://doi.org/10.1145/3439725

Raja Rajeswari, T. S., Khajashareef, S. K., Sandhya, N., & Chinnasamy, P. (2022). E-voting system using blockchain. In Algorithms for Intelligent Systems (pp. 3–12). Springer Singapore.

Kaur, A., & Singh, S. (2021). Blockchain smart contracts: Applications, challenges, and future directions. Journal of Ambient Intelligence and Humanized Computing, 12(11), 12367–12385. https://doi.org/10.1007/s12083-021-01127-0

Iansiti, M., & Lakhani, K. R. (2021). How blockchain can simplify partnerships. Harvard Business Review Digital Articles, 2–6.

IBM Corporation. (n.d.). What are smart contracts on blockchain? IBM.com. Retrieved July 6, 2023 from https://www.ibm.com/topics/smart-contracts