Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/18BE14Antolovic

IMPLEMENTACIJA APLIKACIJE ZA SPORTSKA KLAĐENJA PRIMENOM ETHEREUM PLATFORME

IMPLEMENTATION OF SPORTS BETTING APPLICATION USING ETHEREUM PLATFORM

Igor Antolović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu predstavljeno je potencijalno rešenje za decentralizovanu aplikaciju za sportsku kladionicu u okviru Ethereum blockchain mreže. Objašnjenje su teorijske osnove i navedeni izazovi koji se susreću u ovom domenu. Opisane su terminologije vezane za ovu decentralizovanu aplikaciju kao što su blockchain tehnologija, Ethereum blockchain, koncept pametnih ugovora, Oracle entiteti i Solidity jezik. Na kraju je prikazan model i opis implementacije sistema, kao i završna zaključena zapažanja.*

Ključne reči: *blockchain, pametni ugovori, Ethereum, dapp, Solidity, Oracle entiteti, kladionica*

Abstract – *This paper presents potential solution for Ethereum decentralized application (dapp) for sports betting. Theoretical explanations are given and challenges that are met in this domain are listed. Terminologies related to this dapp such as blockchain technology, Ethereum platform, smart contracts, Oracle entities and Solidity language are described. In the end, the paper represents model and proposed implementation of software and provides final considerations.*

Keywords: *blockchain, smart contracts, Ethereum, dapp, Solidity, Oracle entities, bookmaker*

1. UVOD

Jedna od najvećih svetskih kompanija za razvoj softvera za sportsko klađenje - GammaStack, sprovela je anketu vezanu za mogućnost poboljšanja svojih usluga. Najveći deo zahteva, oko 35%, se odnosilo na implementaciju sistema integrisanih sa kriptovalutama i blockchain tehnologijom. Korisnici uvek traže brža i sigurnija rešenja za transfer novca, a blockchain tehnologija nudi dobra rešenja za ove zahteve.

Takođe, implementacijom blockchain softvera, kladionice bi mogle znatno da smanje operativne troškove vezane za tradicionalno korišćenje kreditnih kartica, uključujući i naknade za obradu elektronskih transakcija. Smanjenje troškova bi omogućilo bolje poslovanje sportskih kladionica, kao i bolje uslove za krajnje korisnike[1]. Zbog ovoga, ideja ovog rada je da pruži i analizira jednu implementaciju blockchain sistema sportske online kladionice.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Sladić, red. prof.

2. BLOCKCHAIN

Blockchain, nekad nazvan i kao „Tehnologija distribuirane glavne knjige“ (eng. Distributed Ledger Technology - DLT), skladišti transakcije digitalnih imovina (eng. digital assets) i čini njihovu istoriju nepromenjivom i transparentnom, koristeći decentralizaciju i kriptografsko heširanje (eng. cryptographic hashing). Blockchain tehnologija nudi mehanizam potvrđivanja konsenzusa kroz mrežu kompjutera koja sprovodi peer-to-peer transakcije bez potrebe za posrednikom. Svaka transakcija se validira i zajedno sa grupom drugih validiranih transakcija, se dodaje u već postojeći transakcioni lanac (eng. chain) u okviru novog bloka (eng. block). Kada se transakcija jednom doda u lanac, praktično ju je nemoguće izmeniti ili obrisati. Prvi predlog implementacije blockchain tehnologije, pojavljuje se 2008. godine, izlaskom rada pod nazivom "Bitcoin: A Peer-to-Peer Electronic Cash System" [2], čiji je autor predstavljen pod pseudonimom Satoshi Nakamoto. Nakamoto je naredne godine (2009) implementirao prvi blockchain za transakcije Bitcoin kripto valute.

2.1. Blockchain 2.0

Termin "blockchain 2.0" je nastao kako bi se napravila razlika između Bitcoin-a kao digitalne imovine i blockchain-a "kao programabilne distribuirane infrastrukture poverenja" sa dodacima novih skalabilnih karakteristika i proširivosti. Umesto da se blokchain posmatra kao mehanizam decentralizacije novca i plaćanja, blokchain 2.0 proširuje obim tehnologije kako bi omogućio decentralizaciju tržišta uopšteno. Prva ideja implementacije blockchain 2.0 tehnologije je izneta 2013. godine [3].

2.2. Pametni ugovori

Ključna inovacija u okviru Ethereum platforme je uvođenje kompjuterskih programa u blokove blockchain sistema. Ovi programi su nazvani "pametni ugovori" (eng. smart contracts). Pametni ugovori predstavljaju samo-izvršavajuće ugovore gde su uslovi i odredbe umesto rečima, napisani u nekom programskom jeziku. Kako se ovi programi pokreću na svim čvorovima blockchain mreže nezavisno, ne postoji potreba za trećom stranom koja nadgleda da li obe strane poštuju svoje obaveze navedene u ugovoru. Kompjuterski kod na blockchain-u je nepromenljiv, kao i svi podaci, što znači da ne postoji opasnost od izmene ugovora [4].

Pametne ugovore na osnovu toga da li zavise od spoljnih informacija (informacije koje dolaze spolja, van blockchain sistema) možemo podeliti na: *determinističke* (za pokretanje ovakvih ugovora nisu potrebne informacije iz

spoljašnosti sistema) i *nedeterminističke* (ovi ugovori zavise od informacija koje se ne nalaze na blockchain sistemu, već treba da dođu iz spoljašnosti sistema). Fundamentalno ograničenje pametnih ugovora predstavlja to što oni sami po sebi ne mogu da stupaju u interakciju sa podacima i sistemima koji postoje izvan njihovog originalnog blockchain okruženja što predstavlja problem pri realizaciji nedeterminističih ugovora.

2.3. Oracle entiteti i Chainlink

Kako bi nedeterministički ugovori funkcionisali, uvode se entiteti koji se nazivaju oracles, čija je uloga da omoguće povezivanje blockchain sistema sa off-chain podacima (podacima van blockchain sistema). Zbog toga što su izolovani od spoljnih sistema, blockchain sistemi poseduju svoje najvrednije osobine kao što je snažan konsenzus o validnosti korisničkih transakcija, sprečavanje napada dvostruke potrošnje i ublažavanje zastoja u okviru mreže. Oracle entiteti zapravo predstavljaju nove delove infrastrukture blockchain okruženja, koji omogućavaju da se premosti prepreka u bezbednoj komunikaciji blockchain-a sa spoljašnjim svetom.

Pošto podaci koje oracle entiteti dostavljaju blockchain sistemima direktno određuju ishode pametnih ugovora, od krucijalne je važnosti da oracle mehanizam bude ispravan kako bi se ugovor izvršavao tačno onako kako se očekuje. Potpuno rešenje oracle problema zahteva decentralizaciju oracle entiteta kako bi se sprečila manipulacija podacima, netačnost i zastoji u sistemu.

Decentralizovana oracle mreža (eng. Decentralized Oracle Network - DON) kombinuje više nezavisnih operatera oracle čvorova i više pouzdanih izvora podataka kako bi se uspostavila end-to-end decentralizacija [5].

Chainlink predstavlja trenutni standard industrije za DON (Decentralized Oracle Network). Chainlink je tehnologija otvorenog koda koju je zajednički razvila velika zajednica programera. Chainlink je podržan na svim EVM kompatibilnim blockchain mrežama [6].

3. ETHEREUM

Ethereum je blockchain platforma sa sopstvenom kriptovalutom, koja se zove Ether (ETH) ili Ethereum, koja podržava rad sa pametnim ugovorima koji se definišu sopstvenim programskim jezikom platforme, koji se zove Solidity. Korisnici Ethereum blockchain-a mogu da kreiraju, objavljuju, monetizuju i koriste aplikacije na platformi, kao i da koriste svoju ETH kriptovalutu za plaćanje [7]. Decentralizovane aplikacije koje nastaju primenom pametnih ugovora na ovoj platformi se popularno nazivaju "DApps".
Uvodni rad za Ethereum je prvobitno objavljen 2013. godine od strane kanadsko-ruskog programera pod imenom Vitalik Buterin. Ethereum mreža je otpočela sa radom 30. jula 2015. godine, kada je kreiran početni (eng. genesis) blok ove blockchain mreže.

4. SOLIDITY

Solidity programski jezik se koristi za pisanje ugovora u okviru Ethereum platforme, i on se može predstaviti kao objektno orijentisan, statički pisani (tip promenljive je poznat u vreme kompajliranja) jezik visokog nivoa za implementaciju pametnih ugovora koji podseća na C++.

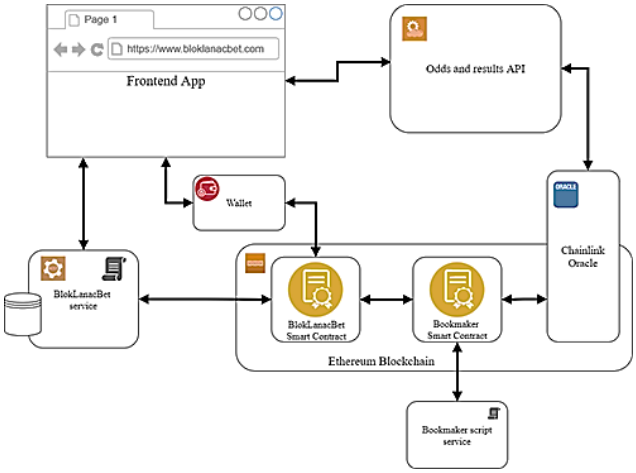
Solidity jezik podržava: nasleđivanje, biblioteke i kompleksne korisnički definisane tipove. Pametni ugovori se u ovom jeziku pišu poput klasa u standardnim objektno orijentisanim jezicima. Kao i klase, svaki ugovor ima svoja polja, koja mogu biti private i public, svoje metode (u Solidity jeziku funkcije), kao i konstruktor [8].

5. MODEL I IMPLEMENTACIJA SISTEMA

Implementirani deo sistema, koji predstavlja decentralizovanu aplikaciju - DApp, se sastoji od: F*ront-end* klijentske aplikacije, *BlokLanacBet servisa* sa svojom bazom podataka, *BlokLanacBet pametnog ugovora*, *Bookmaker pametnog ugovora* i *Bookmaker script serivisa*. Pored ovoga, tu su i eksterne komponente: *Wallet*, *Chainlink Oracle* i *Odds and results API* koje nisu direktno implementirane u okviru sistema, ali su korišćene da obezbede potrebne servise kako bi celokupan sistem funkcionisao.

5.1. Arhitektura sistema

Na slici 5.1 prikazana je arhitektura celokupnog sistema.



Slika 5.1 *Arhitektura sistema*

Ključni deo sistema predstavljaju dva pametna ugovora, *BlokLanacBet ugovor* i *Bookmaker ugovor*. Uloga *Bookmaker ugovora* je da preko *Chainlink Oracle entiteta*, od *Odds and results API-a* dobavlja podatke vezane za kvote i rezultate sportskih događaja i skladišti ih na blockchain-u. Dakle, *Bookmaker ugovor* obezbeđuje sve relevantne podatke za sportsko klađenje direktno na blockchain-u. Uloga *BlokLanacBet ugovora* je da kreira opklade, koristeći kvote koje dobija od *Bookmaker ugovora*, kao i da procesira ishode opklada, nakon što su svi relevantni događaji za konkretnu opkladu završeni, koristeći rezultate koje dobija od *Bookmaker ugovora*. Razlog zašto je odlučeno da se funkcionalnosti ovako podele, je taj što potencijalni *Bookmaker ugovor* mogu koristiti i više ovakvih kladionica, odnosno poslovnih entiteta.

Kladionice koje nemaju nameru da se bave kompleksnim procesom kalkulacija kvota, ili da brinu o tome kako da dostave ove kvote kao i rezultate na blockchain, mogle bi da plaćaju servise *Bookmaker pametnog ugovora*. Kladionice na ovaj način zarađuju na opkladama, a Bookmaker zarađuje naplaćujući svoje servise. U okviru ovog projekta, isti entitet je odgovoran za ova dva pametna ugovora, tako da konkretna *BlokLanacBet*

kladionica ne mora da plaća usluge *Bookmaker pametnom ugovoru*.

Kako se pametni ugovori ne mogu pokretati "sami od sebe", *Bookmaker script servis* služi da pokrene funkcije *Bookmaker pametnog ugovora* onda kada je to potrebno. Sve troškove transakcija potrebnih da se ove funkcije izvrše snosi Bookmaker biznis entitet, koji je u ovom slučaju isti entitet koji održava celokupni sistem.

BlokLanacBet servis predstavlja RESTful servis vezan za opklade, sa svojom bazom podataka, koji služi i za pokretanje *BlokLanacBet pametnog ugovora*. *BlokLanacBet servis* ima ulogu da: obezbedi skladištenje opklada, obezbedi dostupnost istorije opklada za svakog korisnika i pokreće funkciju za procesiranje opklada u okviru *BlokLanacBet pametnog ugovora*. Sve troškove transakcija potrebnih da se funkcija procesiranja opklade izvrši snosi BlokLanacBet biznis entitet, koji je u ovom slučaju isti entitet koji održava celokupni sistem.

Front-end klijentska aplikacija služi korisnicima da ostvare interakciju sa sistemom. Preko ove aplikacije korisnik kreira opkladu. Da bi se opklada kreirala, korisnik prvo mora da se konektuje na sistem pomoću svog kripto novčanika - *Wallet-a*. Opklada se kreira tako što korisnik bira ishode željenih sportskih događaja, a zatim unosi željeni ulog. Nakon ovoga, *Wallet* prvo proverava da li korisnik ima dovoljno sredstava da stavi uneti ulog na opkladu, a zatim predlaže cenu gasa i traži potvrdu od korisnika za konkretnu blockchain transakciju. Korisnik može da podesi cenu gasa i tek nakon potvrde transakcije u okviru *Wallet-a*, transakcija se šalje na blockchain ka *BlokLanacBet pametnom ugovoru*. Sve troškove transakcije za stavljanje opklade snosi korisnik. Takođe, putem ove aplikacije korisnik ima pristup istoriji svojih opklada.

5.2. Bookmaker pametni ugovor

Bookmaker ugovor nasleđuje ChainlinkClient ugovor, iz Chainlink biblioteke pametnih ugovora. Ovaj ugovor obezbeđuje slanje zahteva za podatke o utakmicama ka Chainlink oracle čvorovima. Ova funkcionalnost je podržana *sendChainlinkRequestTo(oracle,request,fee)* funkcijom. Da bi se ova funkcija pozvala, prvo je potrebno kreirati *Chainlink.Request* objekat, kojem se prosleđuje: *jobId*, *adresa* ugovora i *selektor funkcije* ugovora. Nakon kreiranja ovog objekta, definisani su parametri relevantni za zahtev, a to su: *HTTP metoda* koja se koristi, *Query parametri* koji se dodaju na GET zahtev, *path* parametar, koji predstavlja json path.

Preostali parametri *sendChainlinkRequestTo* funkcije, oracle i fee, kao i jobId se mogu pronaći na market.link sajtu i zavise od željenog oracle čvora (adrese oracle ugovora), kao i job-a za izvršavanje. *Fee* predstavlja cenu u LINK tokenima koju je potrebno platiti oracle čvoru da ispuni svoj job.

5.3. BlockLanacBet pametni ugovor

Ovaj pametni ugovor podržava sve funkcije u vezi sa opkladama, koristeći Bookmaker ugovor za potrebne podatke. BlokLanacBet ugovor koristi Bookmaker ugovor preko definicije IBookmaker interfejsa. Da bi ovo funkcionisalo, IBookmaker mora da dobije konkretnu adresu Bookmaker ugovora i ova adresa se pri kreiranju BlokLanacBet ugovora prosleđuje njegovom konstuktoru, u okviru kog se čuva u address polju *bookmakerContract*. Na ovaj način, svaki put kada je potrebno poslati poruku ka Bookmaker ugovoru, pomoću ove adrese je moguće kreirati IBookmaker i pozivati interfejsom definisane funkcije koje dobavljaju potrebne podatke. Pored adrese Bookmaker ugovora, u okviru konstruktora BlokLanacBet ugovora se dodeljuje vrednost i polju *owner*, koje služi da zapamti vlasnika ugovora. Ovaj podatak je bitan jer je vlasnik ugovora dužan da uplaćuje na račun ugovora kako bi ugovor mogao da isplati dobitne opklade i jedini on ima pravo da preuzima sredstva sa računa ugovora. Vlasnik može direktno da uplaćuje sredstva na račun ugovora, slanjem obične transakcije bez podataka ka adresi ugovora.

Ovu funkcionalnost obezbeđuje funkcija *receive()*. Pored *receive()* funkcije tu je i *withdraw(uint)* funkcija koja omogućava vlasniku ugovora da povuče sredstava sa računa pametnog ugovora na svoj račun. Stavljanje opklada se obavlja putem *placeBet(address gambler, Bet calldata bet)* funkcije.

Funkcija *processBetResults(address,uint)* predstavlja funkciju koja podržava procesiranje opklada. Ova funkcija se poziva onog trenutka kada su sve utakmice u okviru opklade završene. Funkcija dobavlja rezultate od Bookmaker ugovora i na osnovu logike vezane za poređenje rezultata utakmica, određuje da li je opklada dobitna ili ne. Svaka dobitna opklada se isplaćuje u okviru ove funkcije.

5.4. Truffle, deployment i testiranje ugovora

Kao blockchain DApp razvojno okruženje je korišćen *Truffle Suite* [9]. Truffle Suite je skup alata koji omogućavaju kompletan razvoj DApp aplikacija. Truffle alati koji su korišćeni u okviru ove implementacije su: *Truffle* (razvojno okruženje i framework za testiranje) i *Ganache* (lokalni blockchain za razvoj i testiranje).

Kao realna testna blockchain mreža u okviru ove implementacije je iskorišćena Kovan Testnet mreža [10]. Deployment pametnih ugovora na ovu blockchain mrežu (Kovan Testnet) se izvršava naredbom *truffle migrate --network kovan.* Nakon što je deployment izvršen, potrebno je prebaciti LINK tokene na adresu Bookmaker pametnog ugovora, kako bi mogao da plati oracle čvoru za servise koje mu obezbeđuje.

U okviru ovog framework-a, svaki test se definiše putem funckije *it* i testovi se grupišu u okviru *contract* funckije. Ova funckija obezbeđuje da se nad ugovorima ponovo radi deployment na podrazumevani Ethereum klijent, u ovom slučaju Ganache blockchain, tako da se testovi unutar ove funkcije pokreću sa inicijalnim stanjem ugovora. Testovi se pokreću *truffle test* naredbom.

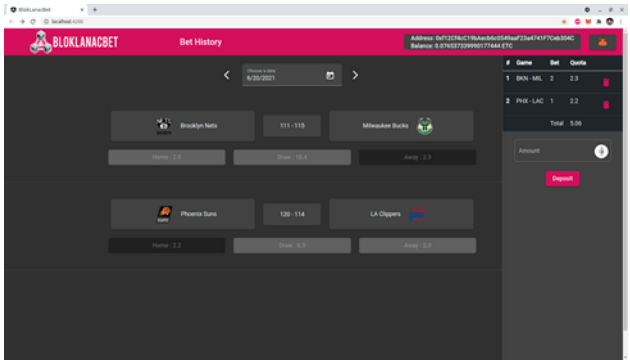
5.5. Off-Chain komponente

Zbog jednostavnosti potrebne logike, BlokLanacBet servis je realizovan pomoću jednostavnog REST API-a sa svojom NoSQL bazom podataka, koji služi isključivo za realizaciju CRUD operacija nad entitetom opklada, uz Truffle skpritu koja služi da pozove BlokLanacBet ugovor a zatim rezultate prosledi ovom REST API-u . Za implementaciju REST API-a je iskorišćen “fake REST API” koji je obezbeđen od strane json-server npm paketa

[11]. Bookmaker servis je takođe realizovan pomoću Truffle skripte koja poziva Bookmaker ugovor. Odds and Results API je implementiran pomoću istog json-server paketa koji je ranije korišćen, s tim da je odrađen i deployment ovog API-a na javni server zbog potreba Oracle čvora. Da bi Oracle čvor mogao da dostavlja podatke, API mora biti javno dostupan. Za dobijanje realnih podataka je korišćen Rapid API, koji predstavlja najveće dostupno tržište API-ja. Konkretan API koji je korišćen je API-NBA [12], koji dostavlja informacije o svim dostupnim podacima o NBA utakmicama. Razlog zašto se direktno nije mogao koristiti API-NBA, nego su podaci morali biti modifikovani, je taj što je bilo potrebno dostaviti Bookmaker pametnom ugovoru rezultate i kvote u jedinstvenom string formatu.

5.6. Front-end klijentska aplikacija

Za implementaciju front-end aplikacije je korišćen Angular framework [13]. Za komunikaciju sa blockchain-om, odnosno pametnim ugovorima, kao i konekciju ka kripto novčaniku, korišćena je web3.js biblioteka [14]. Kao kripto novčanik, koji je ujedno i gateway prilikom slanja transakcija ka blockchain-u, korišćen je MetaMask [15]. Predikcije se unose klikom na jedan od tri dugmeta ispod utakmice, nakon čega se predikcija dodaje u opkladu u okviru side panel-a, koji se pojavljuje dodavanjem prve predikcije. Ekran aplikacije je prikazan na slici 5.2.



Slika 5.2 *Glavni ekran aplikacije*

Nakon što je korisnik odabrao sve željene predikcije utakmica, on unosi željeni ulog u ETH kriptovaluti i klikom na dugme "Deposit" pokreće proces stavljanja opklade, odnosno poziva se metoda angular servisa, čime se otvara prozor MetaMask ekstenzije u kojem je potrebno potvrditi Ehtereum transakciju. Nakon ovoga korisnik bi trebao da čeka da se utakmice završe, nakon čega će sistem pokrenuti funkciju pametnog ugovora koja će procesirati njegovu opkladu. Ukoliko je opklada dobitna korisniku se prikazuje obaveštenje o dobitku i isplaćuje mu se dobitni iznos.

6. ZAKLJUČAK

U radu je predstavljen predlog rešenja i implementacije za online sportsku DApp kladionicu korišćenjem Ethereum blockchain-a. U okviru rada je detaljno opisana tehnologija u kojoj je ovo rešenje implementirano i analizirane su njene prednosti kao i mane. Na osnovu ovoga, dat je predlog arhitekture jednog ovakvog sistema, kao i njegova implementacija.

Ovo rešenje uspeva da ponudi odgovore na ključne probleme koje prouzrokuju centralizovani sistemi online sportskih kladionica i njihovi sistemi za plaćanje. Spram trenutnih mogućnosti i trenutnog stanja relevantnih blockchain tehnologija, pokušano je da se ostvari što veći stepen decentralizacije sistema. Ipak, u ovom pogledu postoji veliki potencijal za napretkom. Već postoje naznake da će u skorijoj budućnosti biti implementirana rešenja koja će omogućiti da se u logiku pametnih ugovora ugrade mehanizmi koji obezbeđuju da se funkcije pametnih ugovora pokreću „same od sebe". Ovo bi uklonilo potrebu za servisima koji su zaduženi isključivo da pokreću funkcije pametnog ugovora, čime bi se postigao veći stepen decentralizovanosti i pouzdanosti.

7. LITERATURA

[1] Zack Jones, The Next Frontier For US Sports Betting Is Crypto And Blockchain Technology https://www.forbes.com/sites/zackjones/2021/09/12/blockchain-adoption-the-next-frontier-for-us-sports-betting/?sh=2fcfc9274edf (pristupljeno u oktobru 2021.)

[2] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[3] Vitalik Buterin, Ethereum Whitepaper, 2013 https://ethereum.org/en/whitepaper/

[4] Maksym Khudiakov, Blockchain 2.0: Smart Contract use cases https://www.axon.dev/blog/blockchain-2-0-smart-contract-use-cases (pristupljeno u oktobru 2021.)

[5] What Is a Blockchain Oracle? https://chain.link/education/blockchain-oracles (prustupljeno u oktobru 2021.)

[6] Chainlink https://chain.link/

[7] Jake Frankenfield, Ethereum https://www.investopedia.com/terms/e/ethereum.asp (pristupljeno u oktobru 2021.)

[8] Solidity https://docs.soliditylang.org/en/v0.8.9/

[9] Truffle Suite https://www.trufflesuite.com/

[10] Kovan Testnet https://kovan-testnet.github.io/website/

[11] JSON server https://github.com/typicode/json-server

[12] API-NBA https://rapidapi.com/api-sports/api/api-nba/

[13] Angular https://angular.io/

[14] web3js https://web3js.readthedocs.io/en/v1.5.2/

[15] MetaMask https://metamask.io/

Kratka biografija:

Igor Antolović rođen je 22.02.1996. godine u Novom Sadu. Osnovne akademske studije završio je 2019. godine. Master rad iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2022. godine.

kontakt: antolovicigor96@gmail.com