

Fakultet elektrotehnike i računarstva

Seminarski rad u okviru kolegija *Seminar 2*

# Analiza i implementacija Web AR tehnologije za interakciju s 3D modelima u stvarnom okruženju

Matea Kranjčić

Zagreb, svibanj 2025.

## Sadržaj

Uvod.....	3
Pregled područja Web AR tehnologije .....	4
Pregled područja proširene stvarnosti (AR) .....	4
Web AR tehnologija i tehnološki okviri za razvoj.....	4
Razvoj Web AR aplikacije.....	5
Analiza zahtjeva .....	5
Poslovni zahtjevi .....	5
Korisnički zahtjevi .....	6
Funkcijski zahtjevi .....	6
Nefunkcijski zahtjevi .....	7
Odabir tehnologija.....	7
Arhitektura aplikacije.....	8
Backend (server-side).....	9
Frontend (client-side).....	9
Tok podataka .....	9
Testiranje .....	10
Upute za korištenje .....	10
Zaključak .....	14
Literatura .....	15
Sažetak .....	16
Ključne riječi .....	16

# Uvod

Proširena stvarnost (AR) odnosi se na situaciju u kojoj je cilj nadopuniti korisnikovu percepciju stvarnog svijeta dodavanjem virtualnih objekata [1]. Glavni problem koji se rješava u okviru ovog seminarskog rada odnosi se na ograničenost korištenja proširene stvarnosti (AR) i njene tehnologije širom spektru korisnika zbog njene složenosti i česte potrebe za posebnim aplikacijama. Uobičajeni pristup aplikacijama proširene stvarnosti uključuje instalaciju mobilnih aplikacija, ponekad i korištenje posebnih platformi (npr. iOS ARKit ili Android ARCore). Samim time otežana je i upotreba na različitim uređajima. Sve to predstavlja veliku prepreku krajnjem korisniku koji bi koristio AR tehnologiju. Razvoj Web AR tehnologije omogućio je implementaciju proširene stvarnosti u web preglednike, bez dodatnih instalacija na uređaje. Time je proširena stvarnost korisniku dostupna direktno kroz web preglednik te je sama interakcija s AR tehnologijom znatno pojednostavnjena.

Praktični zadatak ovog seminarskog rada bio je razvoj funkcionalne Web AR aplikacije koja pokazuje mogućnosti interakcije s 3D modelima u stvarnom okruženju, na web pregledniku. Zamišljen je razvoj aplikacije koja koristi kameru uređaja za detekciju markera te potom prikaz 3D modela. Aplikacija bi, uz jednostavno sučelje koje korisniku omogućuje lakše korištenje bez prethodnog iskustva, osiguravala kompatibilnost s različitim web preglednicima i uređajima. Rad sadrži analizu osnova Web AR tehnologije i tehnoloških okvira, detaljnu dokumentaciju procesa razvoja aplikacije te upute za korištenje iste.

# Pregled područja Web AR tehnologije

## Pregled područja proširene stvarnosti (AR)

Proširena stvarnost (AR) odnosi se na situaciju u kojoj je cilj nadopuniti korisnikovu percepciju stvarnog svijeta dodavanjem virtualnih objekata [1]. Ona dodaje digitalne elemente u stvarni svijet gdje korisnik ostaje u kontaktu s fizičkim svijetom. Za razliku od AR svijeta koji je bliži stvarnom okruženju, virtualna stvarnost (VR) potpuno zamjenjuje stvarni svijet virtualnim okruženjem. Povijesni pregled razvoja AR tehnologije jasno je prikazan u citiranom radu „A Survey on Web-based AR Applications“ [1]. Od ranog razvoja 1960-tih godina, kroz komercijalizaciju i širenje 2000-tih do današnjeg intenzivnog korištenja mobilnih uređaja.

Postoji više vrsta AR sustava, a najkorišteniji su „marker-based“ i „markerless“ sustavi. „Marker-based“ sustav jest sustav temeljen na prepoznavanju markera koji pomaže u otkrivanju objekta u prisutnosti kamere na zaslonu uređaja [2]. „Markerless“ sustav, odnosno sustav bez markera, često se naziva „location-based“ sustav, odnosno sustav temeljen na lokaciji. Primjene AR tehnologije u različitim industrijama (obrazovanje, medicina, turizam, industrija i sl.) opisane su u radu „A Survey on Web-based AR Applications“ [1].

## Web AR tehnologija i tehnološki okviri za razvoj

Web AR tip je proširene stvarnosti prilagođen za web preglednike, bez potrebe za dodatnim instalacijama aplikacija. Prednosti WebAR-a s obzirom na uobičajene AR aplikacije su dostupnost i jednostavnost korištenja (smanjuje količinu prostora za pohranu na mobilnim uređajima [2]), omogućen rad na svim uređajima s modernim web preglednicima neovisno o operacijskom sustavu, brza ažuriranja (bez potrebe ažuriranja aplikacije kod korisnika) i sl.

AR.js predstavlja najnovije Web AR rješenje temeljeno na three.js koje može raditi na svim platformama i bilo kojem pregledniku koji podržava WebRTC i WebGL<sup>1</sup>. Three.js JavaScript biblioteka pomaže pri radu s 3D modelima u pregledniku. A-Frame jest web okvir temeljen na HTML-u, što znatno olakšava razvoj aplikacije. Razvijen je i zamišljen kao jednostavan, ali moćan alat za stvaranje VR i AR sadržaja.<sup>2</sup>

---

<sup>1</sup> <https://ar-js-org.github.io/AR.js-Docs/>

<sup>2</sup> <https://aframe.io/docs/1.7.0/introduction/>

# Razvoj Web AR aplikacije

Razvijena Web AR aplikacija predstavlja primjer pokretanja proširene stvarnosti direktno u web pregledniku bez potrebne instalacije dodatnih aplikacija. Svrha aplikacije jest upravo testiranje koncepta Web AR tehnologije i njezinih mogućnosti. Aplikacija također služi kao primjer integracije React.js-a s bibliotekama AR.js i A-Frame koristeći „marker-based“ sustav. Osnovne funkcionalnosti aplikacije su prepoznavanje „Hiro“ markera, prikazivanje 3D modela iznad prepoznatog markera te praćenje markera i prikazivanje 3D modela ovisno o njegovom pomicanju u stvarnom vremenu.

## Analiza zahtjeva

### Poslovni zahtjevi

Poslovni zahtjevi (*software project requirements*<sup>3</sup>) definiraju kako se projektom upravlja, koji resursi su dostupni te koja ograničenja postoje na razini projekta [5]. Ovi zahtjevi usmjereni su na proces razvoja, a ne na sami proizvod.

1) Vremensko ograničenje

Aplikacija mora biti završena do roka predviđenog za izradu seminarskog rada (22. svibnja 2025.).

2) Resursno ograničenje

Aplikacija mora biti implementirana uz minimalni trošak - besplatni alati i platforme (Node.js, Express.js, React, A-Frame, AR.js, Render).

3) Ograničenja testiranja

Aplikacija mora biti testirana na različitim uređajima kako bi se osigurala kompatibilnost.

4) Edukacijska svrha

Aplikacija treba demonstrirati mogućnosti Web AR tehnologije u edukativne svrhe.

5) Demonstracija tehnologije

Aplikacija treba služiti kao primjer koncepta (engl. *proof of concept*) integracije proširene stvarnosti (AR) u web okruženje.

---

<sup>3</sup> <https://www.computer.org/education/bodies-of-knowledge/software-engineering/v4>

## Korisnički zahtjevi

Korisnički zahtjevi opisuju zadatke koje korisnik mora moći obaviti služeći se aplikacijom.

1) Intuitivno korištenje

Korisnik može koristiti aplikaciju bez posebne obuke ili detaljnih uputa.

2) Brzo pokretanje

Korisnik može skenirati marker odmah nakon učitavanja stranice.

3) Odobranje pristupa kameri

Korisnik može odobriti pristup kameri nakon učitavanja stranice.

4) Prikaz 3D modela

Korisnik može vidjeti 3D model učitani nad skeniranim markerom.

5) Promjena kuta gledanja

Korisnik može vidjeti 3D model iz različitih kutova (perspektiva) pomicanjem kamere oko postavljenog markera.

## Funkcijski zahtjevi

Funkcijski zahtjevi predstavljaju očekivane funkcionalnosti sustava tj. akcije i procese koje sustav mora biti sposoban izvršiti [6]. Oni su odraz potreba budućih korisnika. Za razliku od poslovnih zahtjeva, ovi zahtjevi opisuju što programski proizvod treba obaviti.

1) Pristup kameri uređaja

Aplikacija mora moći pristupiti kameri korisničkog uređaja (računala, tableta, mobitela) kroz web preglednik, a korisnik mora dozvoliti pristup kameri.

2) Detekcija markera

Aplikacija mora precizno odrediti položaj markera u prostoru te ga pratiti ako se miče.

3) Prikaz 3D modela

Aplikacija mora prikazati 3D model na očitani marker te konstantno ažurirati položaj 3D modela ako se marker pomiče.

4) Različiti zasloni

Aplikacija mora moći prikazivati sadržaj na zaslonima različitih veličina (monitor računala, mobitel, tablet).

## Nefunkcijski zahtjevi

Nefunkcijski zahtjevi vidljivi su aspekti programskog proizvoda koji nisu direktno povezani s funkcijama koje izvršava.

1) Pogodnost za korištenje (usability)

Korisničko sučelje mora biti intuitivno; AR iskustvo mora biti razumljivo bez prethodnog predznanja.

2) Pogodnost za ponovno korištenje (reusability)

Struktura koda aplikacije mora omogućiti laku izmjenu u daljnjim ažuriranjima aplikacije kao i lako dodavanje novih 3D modela u budućnosti.

3) Pouzdanost (dependability)

Aplikacija ne smije gubiti vezu s markerom, ako se marker nalazi unutar vidnog polja kamere.

4) Pristupačnost

Aplikacija mora biti dostupna bez posebne opreme ili instalacije.

## Odabir tehnologija

Za backend aplikacije odabrala sam Node.js i Express.js. Node.js je višepatformsko JavaScript runtime okruženje koje omogućuje izradu poslužitelja, web aplikacija, alata naredbenog retka i skripti<sup>4</sup>. Express.js je lagan i fleksibilan okvir za usmjeravanje (engl. routing framework)<sup>5</sup>.

Za moju trenutnu (osnovnu) funkcionalnost aplikacije, REST API nije bio potreban jer aplikacija nema potrebu za komunikaciju s resursima (3D model je „hardkodiran“). Backend u trenutnoj verziji aplikacije služi za inicijalno učitavanje aplikacije, a AR funkcionalnost se izvodi u potpunosti na klijentskoj strani i ne zahtjeva daljnju komunikaciju s backendom.

Za frontend aplikacije odabrala sam kombinaciju React komponenti (koje upravljaju aplikacijom) i A-Frame i AR.js biblioteke (koje omogućuju AR funkcionalnost u pregledniku). Komponenta ARScene.js upravlja AR prikazom, header.ejs i indeks.ejs definiraju osnovnu strukturu HTML-a, a bundle.js sadrži sav kod potreban za pokretanje React aplikacije i AR funkcionalnosti.

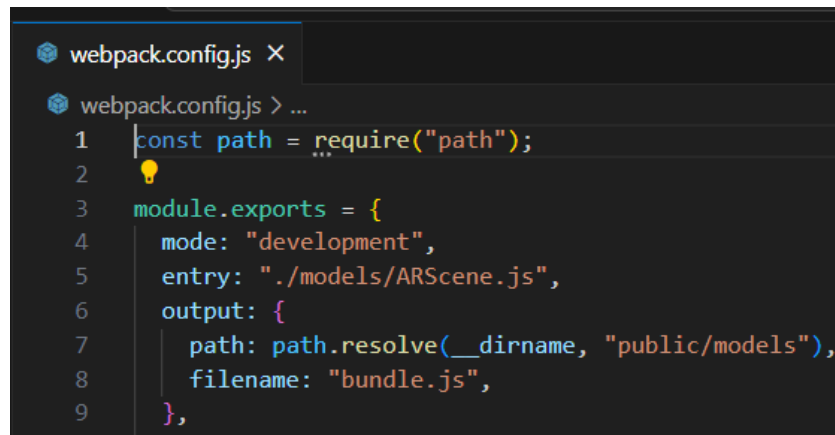
---

<sup>4</sup> <https://nodejs.org/en>

<sup>5</sup> <https://expressjs.com/>

Webpack je statički program za povezivanje/pakiranje modula (engl. module bundler) za moderne JavaScript aplikacije<sup>6</sup>. Webpack je u mojoj aplikaciji bio neophodan jer se React temelji na sustavu modula, što zahtjeva alat za pakiranje (npr. Webpack, Vite). Isto tako omogućuje organizaciju koda u odvojene komponente/datoteke (ARScene.js).

Webpack generira izlaznu datoteku bundle.js „bundling“ procesom. Prvi korak stvaranja bundle.js datoteke jest konfiguracijska datoteka webpack.config.js koja definira kako će se bundle kreirati:



```
webpack.config.js X
webpack.config.js > ...
1  const path = require("path");
2
3  module.exports = {
4    mode: "development",
5    entry: "./models/ARScene.js",
6    output: {
7      path: path.resolve(__dirname, "public/models"),
8      filename: "bundle.js",
9    },
10  }
```

Zatim u git bash-u pokrenuti naredbu „npx webpack --mode development“ ili samo naredbu „npm run build“ ukoliko je gornja naredba definirana u datoteci package.json. Nakon što se on kreira iz datoteke ARScene.js, datoteka indeks.ejs uključit će ga kao skriptu „<script src=\"/models/bundle.js"></script>“.

Za deployment i hosting aplikacije sam koristila Render<sup>7</sup> platformu. Platforma za vrijeme pisanja seminarskog rada nudi besplatno korištenje za osobne projekte te je povezana s GitHub-om. Na taj način svakom naredbom „push“, pokreće se novi deploy na Render-u.

## Arhitektura aplikacije

Aplikacija koristi klasičnu MVC (Model-View-Controller) arhitekturu prilagođenu za Node.js, Express.js i React s AR funkcionalnostima. Dijeli se na backend (server-side) i frontend (client side).

---

<sup>6</sup> <https://webpack.js.org/>

<sup>7</sup> <https://render.com/>



- ▼ SEMINAR2\_WEBAR
  - ▼ models
    - JS ARScene.js
    - ≡ lowpoly\_origami\_penguin.glb
  - > node\_modules
  - ▼ public
    - ▼ models
      - JS bundle.js
      - ≡ bundle.js.LICENSE.txt
    - ▼ styles
      - # main.css
    - <> ar-direct-test.html
    - ≡ lowpoly\_origami\_penguin.glb
    - <> model-test.html
    - ≡ scene.gltf
  - ▼ routes
    - JS index.js
  - ▼ views
    - ▼ partials
      - <> header.ejs
      - <> index.ejs
  - ⚙ .env
  - { } package-lock.json
  - { } package.json
  - JS server.js
  - ⚙ webpack.config.js

## Backend (server-side)

Datoteka server.js predstavlja ulaznu točku aplikacije koja pokreće Express server.

Mapa routes sadrži datoteku indeks.js koja upravlja HTTP zahtjevima i usmjerava ih na tražene resurse.

Mapa views sadrži EJS datoteke/predloške koji se renderiraju na serveru – header.ejs i indeks.ejs.

Datoteka .env sadrži varijable okoliša (engl. environment variables) koje se koriste u aplikaciji.

## Frontend (client-side)

Mapa models sadrži ARScene.js koja je ključna React komponenta za upravljanje AR funkcionalnostima. Ona integrira A-Frame i AR.js.

Gore spominjana datoteka webpack.config.js definira kako se React komponente „bundle-aju“ u datoteku bundle.js.

Mapa public sadrži podmapu styles sa CSS datotekom te podmapu models s gore spomenutom datotekom bundle.js. Također sadrži i HTML datoteke za testiranje ispravnosti 3D modela bez React framework-a (ar-direct-test.html, model-test.html) i 3D modele (lowpoly\_origami\_penguin.glb, scene.gltf).

## Tok podataka

Pri inicijalnom učitavanju korisnik šalje zahtjev serveru. Express ruter (u ovom slučaju routes/indeks.js) procesira taj zahtjev. Server renderira EJS datoteku/predložak koja u sebi sadrži referencu na bundle.js.

Zatim preglednik (klijent) učitava bundle.js koji sadrži React aplikaciju koja inicijalizira AR komponente. Tada nastupaju A-Frame i AR.js pomoću kojih se 3D model učitava na detektirani marker.

AR.js kontinuirano osvježava kameru i traži marker. Na taj detektirani marker, A-Frame pozicionira 3D model koji se konstantno ažurira ovisno o pomicanju markera ili kamere.

## Testiranje

Testiranje aplikacije provedeno je na jednom uređaju (Microsoft Surface Pro 8) s fokusom na testiranje aplikacije na više različitih web preglednika. Primarni cilj bio je potvrditi tehnološku izvodljivost stoga je zanemareno ograničenje testiranja na jednom uređaju. Aplikacija je testirana na Google Chrome pregledniku, Mozilla Firefox pregledniku te na Microsoft Edge pregledniku. Aplikacija je na sva tri preglednika bila potpuno funkcionalna nakon odobrenja dozvole za korištenje kamere. Prepoznavanje markera bilo je brzo i precizno, a prikaz 3D modela stabilan.

Kod je vidljiv na sljedećem linku: [https://gitlab.com/ma\\_tea/webarsem2.git](https://gitlab.com/ma_tea/webarsem2.git).

## Upute za korištenje

Prije pokretanja aplikacije, korisnik mora imati marker ili na drugom uređaju ili isprintan na papiru. U slučaju ove aplikacije, koristi se javno dostupni „Hiro marker“ (Slika 1 Hiro marker).

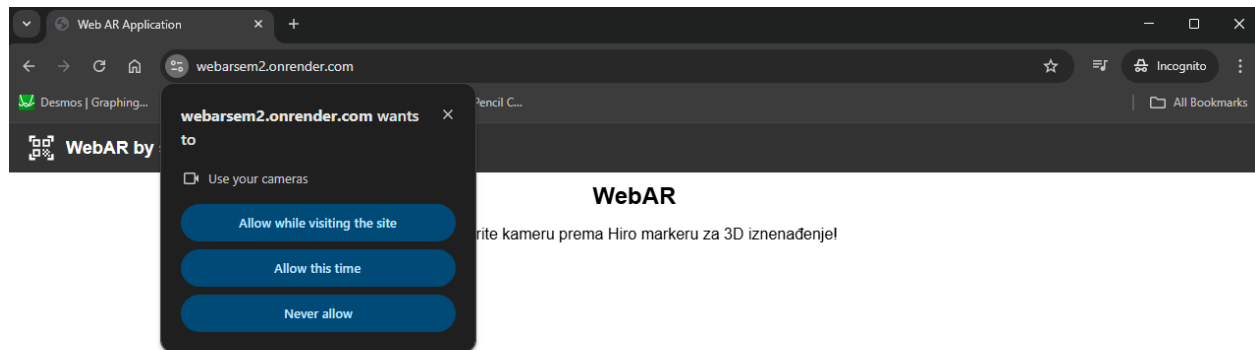


*Slika 1 Hiro marker*

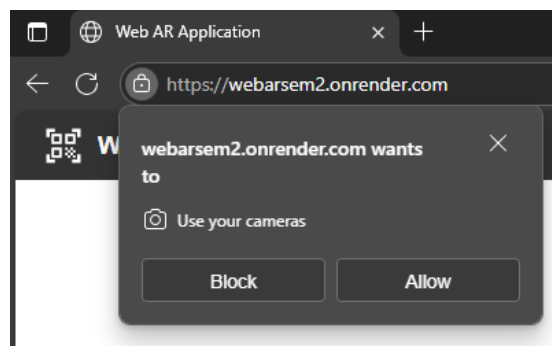
Marker je dostupan i na službenoj stranici <https://aframe.io/blog/arjs/>.

Aplikacija se pokreće otvaranjem poveznice <https://webarsem2.onrender.com/>. Korisniku će se na uređaju odmah prikazati početna stranica koja je ujedno i jedina stranica.

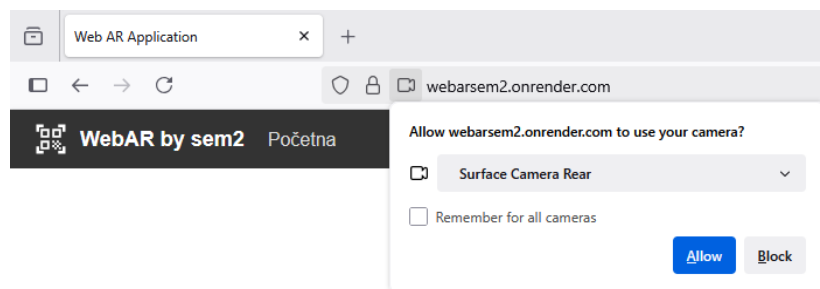
Preglednik će od korisnika tražiti dopuštenje za korištenje kamere, što se vidi na slikama Slika 2 Primjer upita na pregledniku Google Chrome, Slika 3 Primjer upita na pregledniku Microsoft Edge i Slika 4 Primjer upita na pregledniku Firefox.



*Slika 2 Primjer upita na pregledniku Google Chrome*

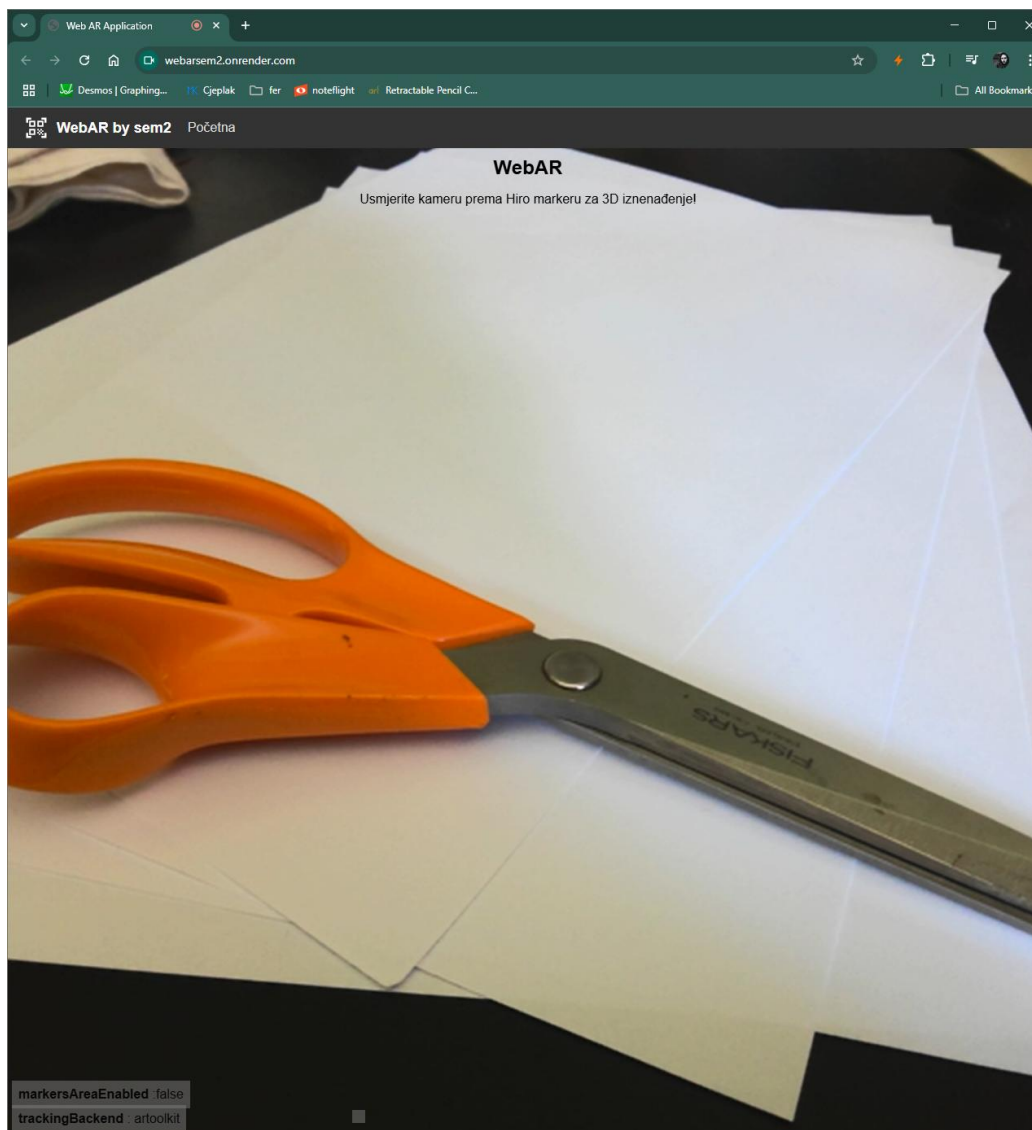


*Slika 3 Primjer upita na pregledniku Microsoft Edge*



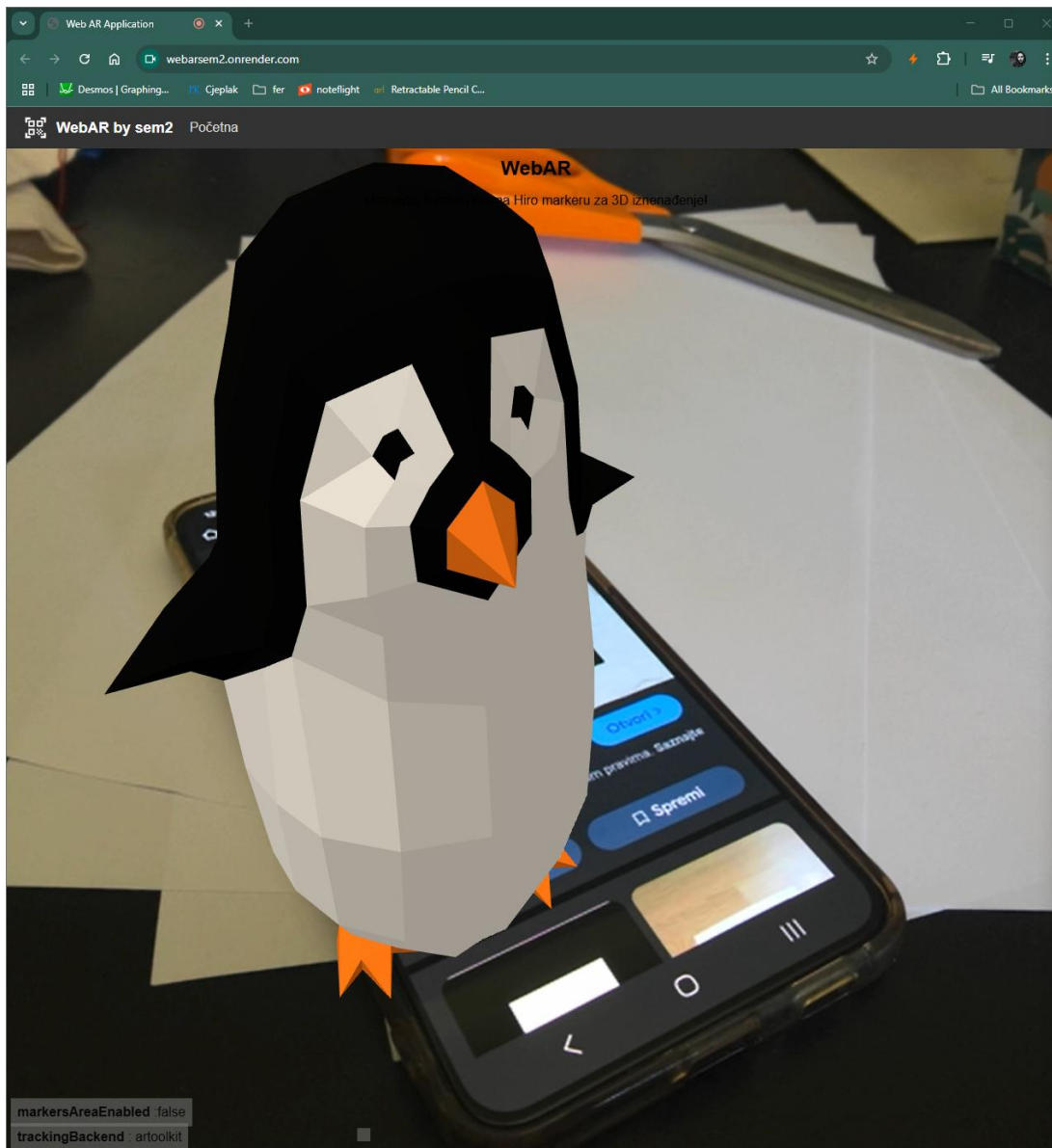
*Slika 4 Primjer upita na pregledniku Firefox*

Kada korisnik dopusti korištenje kamere uređaja, na početnoj stranici preko cijele stranice pojavit će se sadržaj koji vidi kamera (Slika 5 Primjer rada aplikacije).



Slika 5 Primjer rada aplikacije

Kada korisnik stavi Hiro marker pred kameru, pojavit će se 3D model pingvina (Slika 6 Primjer vidljivog 3D modela).



Slika 6 Primjer vidljivog 3D modela

Primjer rada aplikacije vidljiv je na sljedećoj poveznici:

<https://www.youtube.com/watch?v=-EJZwQf5ir0> .

## Zaključak

U okviru ovog seminarskog rada uspješno je razvijena Web AR aplikacija koja pokazuje mogućnosti implementacije proširene stvarnosti direktno u web preglednike bez potrebe za dodatnim instalacijama. Glavni cilj rada, rješavanje problema ograničene dostupnosti AR tehnologije zbog potrebe za posebnim aplikacijama, ostvaren je kroz razvoj jednostavne Web AR aplikacije. Aplikacija koristi „marker-based“ sustav za prepoznavanje „Hiro“ markera te prikazuje 3D model pingvina. Kombinacijom React.js-a, A-Frame-a i AR.js-a postignuta je kompatibilnost s različitim web preglednicima (Google Chrome, Mozilla Firefox, Microsoft Edge). Aplikacija pokazuje kako Web AR omogućuje korištenje većem broju korisnika, neovisno o operacijskom sustavu ili vrsti uređaja na kojem ju pokreću. Ovaj rad potvrđuje da Web AR tehnologija predstavlja budućnost proširene stvarnosti, čineći ju dostupnijom većem broju korisnika te olakšavajući njezinu integraciju u obrazovne svrhe.

# Literatura

- [1] Behrang Parhizkar, Ashraf Abbas M. Al-Modwahi, Arash Habibi Lashkari, Mohammad Mehdi Bartaripou, Hossein Reza Babae, *A Survey on Web-based AR Applications*
- [2] Nitika, Tanuja Kumari Sharma, Saumya Rajvanshi, Keshav Kishore, *A study of Augmented Reality performance in web browsers (WebAR)*
- [3] Xiuquan Qiao, Pei Ren, Schahram Dustdar, Fellow IEEE, Ling Liu, Fellow IEEE, Huadong Ma, Junliang Chen, *Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights*
- [4] Apoorva K M S, *Introduction to AR.js: Bringing Augmented Reality to the Web*, poveznica: <https://codestax.medium.com/introduction-to-ar-js-bringing-augmented-reality-to-the-web-b5c88c2d4448>; pristupljeno 30. travnja 2025.
- [5] AR.js – Augmented Reality on the Web, poveznica: <https://ar-js-org.github.io/AR.js-Docs/>; pristupljeno 22. svibnja 2025.
- [6] Hironori Washizaki (urednik), *Guide to the Software Engineering Body of Knowledge*, Version 4.0, IEEE Computer Society, 2024; <https://www.computer.org/education/bodies-of-knowledge/software-engineering> ; pristupljeno 25. travnja 2025.

## Sažetak

Web AR aplikacija predstavlja inovativno rješenje koje kombinira tehnologiju proširene stvarnosti (AR) s web tehnologijama. Korištenjem Aframe.io i AR.js, aplikacija omogućuje korisnicima interakciju s 3D modelima u stvarnom okruženju bez potrebe za instalacijom same aplikacije. Razvijena aplikacija omogućuje funkcionalnosti poput pristupa modelima putem markera te manipulacije samih modela. Projekt pokazuje potencijal AR tehnologije u poboljšanju korisničkog iskustva i interakcije sa sadržajem kroz pristupačno i višeplatformsko rješenje.

## Ključne riječi

Proširena stvarnost, AR, Web AR, Aframe, AR.js, 3D modeli, markeri