

# Umjetna inteligencija i frontend programiranje web-a

---

**Rumac, Mateo**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Pula / Sveučilište Jurja Dobrile u Puli**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:137:593955>

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

*Download date / Datum preuzimanja:* **2024-11-30**



*Repository / Repozitorij:*

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli  
Fakultet Informatike u Puli

**MATEO RUMAC**

**UMJETNA INTELIGENCIJA I FRONTEND PROGRAMIRANJE WEB-A**

Završni rad

**Pula, rujan, 2024.**

Sveučilište Jurja Dobrile u Puli  
Fakultet Informatike u Puli

## **UMJETNA INTELIGENCIJA I FRONTEND PROGRAMIRANJE WEB-A**

Završni rad

**Mateo Rumac**

**JMBAG: 0303088205, redoviti student**

**Studijski smjer: Informatika**

**Znanstveno područje: Društvene znanosti**

**Znanstveno polje: Informacijske i komunikacijske znanosti**

**Znanstvena grana: Informacijski sustavi i informatologija**

**Predmet: Osnove IKT**

**Mentor: izv. prof. dr. sc. Snježana Babić**

**Pula, rujan, 2024.**

# SADRŽAJ:

1. UVOD.....	1
2. UMJETNA INTELIGENCIJA .....	3
2.1. Pojam umjetne inteligencije.....	3
2.2. Karakteristike umjetne inteligencije .....	5
2.3. Pojam prirodnih jezika.....	7
3. FRONTEND PROGRAMIRANJE WEBA.....	10
3.1. Općenito o webu , web tehnologiji i izradi web stranica.....	10
3.2. Pojam i obilježja frontend programiranja weba.....	13
3.3. Jezici koji se koriste u frontend programiranju.....	15
3.4. Općenito o bibliotekama za web.....	19
3.5. Općenito o aplikacijskim programskim sučeljima.....	22
4. NAČINI INTEGRACIJE UMJETNE INTELIGENCIJE U FRONTEND PROGRAMIRANJU.....	23
4.1. Pregled popularnih biblioteka i alata za umjetnu inteligenciju u frontendu .....	23
4.2. Kombinacija frontend tehnologija i AI za stvaranje kompleksnih web aplikacija .....	25
5. PRIMJER UPOTREBE/PRIMJENE UMJETNE INTELIGENCIJE U FRONTEND PROGRAMIRANJU WEBA NA PRIMJERU CHATGPT I COPILOT .....	28
5.1. Chatgpt.....	29
5.2. Copilot.....	38
5.3. Upotreba ChatGpt i Copilota u kontekstu frontend programiranja weba.....	48
6. BUDUĆI TRENDOVI RAZVOJA UMJETNE INTELIGENCIJE I FRONTEND PROGRAMIRANJA.....	50
7. ZAKLJUČAK .....	55
LITERATURA .....	57
POPIS SLIKA.....	60
SAŽETAK .....	61
SUMMARY .....	62

# 1. UVOD

Umjetna inteligencija (AI) postala je ključna komponenta u suvremenoj tehnologiji, revolucionirajući razvoj i korištenje web aplikacija. Napredak umjetne inteligencije uvelike je utjecao i na područje frontend programiranja, odgovorno za izradu korisničkog sučelja i doživljaja web stranica. Korištenjem sofisticiranih algoritama i AI alata, programeri sada mogu dizajnirati web aplikacije koje su dinamičnije, privlačnije i inteligentnije. Ovaj rad ulazi u sferu AI i mrežnog programiranja sučelja, ispitujući i teorijske temelje i stvarne primjene njihove integracije.

Cilj ovog završnog rada je istražiti potencijal uključivanja umjetne inteligencije u frontend web razvoj, ispitujući prednosti, prepreke i buduće izgleda ovog polja u nastajanju. Primarni ciljevi uključuju definiranje ključnih koncepata i značajki umjetne inteligencije, proučavanje osnova frontend programiranja, istraživanje različitih metoda integracije AI u frontend tehnologije, prikazivanje praktičnih primjera primjene AI u frontend programiranju putem alata kao što su ChatGPT i Copilot i raspravljanje o nadolazećim trendovima i izazovima u području web razvoja s integracijom umjetne inteligencije. Fokus ovoga rada je istražiti mogućnosti i metodologije za uključivanje umjetne inteligencije u frontend web programiranje, procjenu trenutnih alata, tehnologija i praksi koje olakšavaju takvu integraciju, te procjenu njihovog utjecaja na korisničko iskustvo i ukupnu učinkovitost weba. aplikacije.

Početni dio rada daje sažeti sažetak predmeta, ciljeva i strukture poglavlja. Raspravlja se o teorijskim temeljima umjetne inteligencije koji obuhvaćaju definiranje koncepata, temeljnih principa i primjenu obrade prirodnog jezika unutar područja umjetne inteligencije. Dodatno, predstavljeno je ispitivanje frontend web razvoja, pokrivajući osnove web tehnologija, elemente frontend programiranja, jezike koji se obično koriste, prevladavajuće biblioteke i značaj API-ja u ovom području. U djelu koji se fokusira na uključivanje umjetne inteligencije u razvoj sučelja, pruža se opsežan pregled knjižnica i alata koji se često koriste za AI u programiranju sučelja. Također raspravlja se o nadolazećim trendovima u ovom području, sinergiji između frontend

tehnologija i umjetne inteligencije za izradu sofisticiranih web aplikacija, kao i preprekama i budućim izgledima frontend programiranja s integriranom umjetnom inteligencijom. Dodatno, istražuju se specifične instance primjene umjetne inteligencije u frontend programiranju, kao što su ChatGPT i Copilot, nudeći detaljno ispitivanje i demonstraciju svakog alata.

Zaključno, svrha ovoga rada je predstaviti temeljitu analizu postojećeg područja umjetne inteligencije u frontend programiranju. Kroz ovo istraživanje identificirani su ključni nalazi koji rasvjetljavaju trenutne mogućnosti i ograničenja umjetne inteligencije u ovom području.

## **2. UMJETNA INTELIGENCIJA**

Umjetna inteligencija (AI) brzo je postala istaknuta sila u modernoj tehnologiji, revolucionirajući industrije svojim napretkom i sposobnostima. Evolucija umjetne inteligencije osnažila je strojeve za izvršavanje zadataka koji obično zahtijevaju ljudsku inteligenciju, uključujući prepoznavanje govora, donošenje odluka, tumačenje vizualnih sadržaja i razumijevanje prirodnog jezika. Ovo poglavlje prikazuje definiciju, razlikovna obilježja i nijanse prirodnog jezika u području umjetne inteligencije, pozivajući se na relevantne studije i resurse unutar ovog područja.

### **2.1. Pojam umjetne inteligencije**

Umjetna inteligencija, također poznata kao strojna inteligencija, odnosi se na sposobnost računalnih sustava da izvršavaju zadatke koji obično zahtijevaju ljudski intelekt, poput razumijevanja govora, donošenja odluka, tumačenja vizualnih podataka i razumijevanja prirodnog jezika. Ovaj pojam obuhvaća različite tehnologije i metodologije koje omogućuju strojevima da rade samostalno ili uz minimalno ljudsko vodstvo. Kako definiraju Russell i Norvig (2020), AI uključuje metode koje omogućuju strojevima da steknu znanje kroz iskustvo, prilagode se novim informacijama i izvršavaju zadatke na način sličan ljudima. Ove metode obuhvaćaju područja strojnog učenja i dubokog učenja.

Strojno učenje podskup je umjetne inteligencije koji računalima omogućuje učenje i poboljšanje na temelju podataka i iskustava bez potrebe za eksplicitnim programiranjem. Ovi se algoritmi koriste za prepoznavanje uzoraka unutar podataka, predviđanje rezultata i donošenje informiranih odluka na temelju tih obrazaca. Na primjer, strojno učenje može se primijeniti za filtriranje neželjene e-pošte, predlaganje proizvoda prilagođenih individualnim preferencijama ili predviđanje fluktuacija na financijskim tržištima (Russell, Norvig, 2020).

Duboko učenje, grana strojnog učenja, koristi zamršene neuronske mreže za prosijavanje golemih skupova podataka i otkrivanje zamršenih obrazaca koji možda neće biti odmah vidljivi. Ove neuronske mreže modelirane su prema arhitekturi ljudskog mozga, s međusobno povezanim slojevima čvorova ili "neurona" koji obrađuju i analiziraju podatke. Ova tehnologija omogućuje računalima da razaznaju zamršene slike, razumiju prirodni jezik i izvršavaju zadatke poput prepoznavanja lica i automatskog prijevoda jezika. Glavna ilustracija dubinskog učenja na djelu je stvaranje sofisticiranih modela kao što je GPT-3, koji koristi ogromne skupove podataka i sofisticirane algoritme za proizvodnju artikuliranog i kontekstualno relevantnog teksta na temelju unosa korisnika (Russell, Norvig, 2020).

John McCarthy, ključna osoba u uspostavljanju umjetne inteligencije, definira AI kao polje posvećeno razvoju inteligentnih strojeva kroz kombinaciju znanstvenih principa i inženjerskih tehnika. Ova definicija naglašava raznolik niz disciplina koje pridonose stvaranju AI sustava, uključujući znanost, matematiku, inženjerstvo i tehnologiju. McCarthyjev koncept umjetne inteligencije nadilazi samo tehničke aspekte, obuhvaćajući etička i društvena razmatranja koja prate napredak inteligentnih strojeva (Goodfellow et. al., 2016).

Umjetna inteligencija može se kategorizirati u dvije glavne vrste: uska inteligencija i opća inteligencija. Uska umjetna inteligencija specijalizirana je za obavljanje određenog zadatka, poput prepoznavanja govora ili igranja igara poput šaha. S druge strane, opći AI ima za cilj razviti sustave sa širokim kognitivnim sposobnostima koji mogu nadmašiti ljudsku inteligenciju u više domena. Dok su trenutne aplikacije umjetne inteligencije uglavnom usredotočene na usku inteligenciju, u tijeku su napor da se napreduje prema stvaranju opće umjetne inteligencije, koja ima potencijal revolucionirati naše životne stilove i radna okruženja (Goodfellow et. al., 2016).

Napredak UI tehnologija donosi mnoštvo etičkih problema, poput onih koji se tiču privatnosti, sigurnosti i učinaka na zapošljavanje. Dok korisničko sučelje ima potencijal povećati produktivnost i pojednostaviti svakodnevne zadatke, ono također ima potencijal dovesti do otpuštanja i proširiti jaz između socioekonomskih klasa. Ključno je oprezno pristupiti razvoju umjetne



inteligencije, uzimajući u obzir etičke posljedice i nastojeći ravnomjerno raspodijeliti prednosti (McCarthy, 2007).

Općenito, umjetna inteligencija je dostignuće vrijedno pažnje na području tehnologije, otvarajući mogućnosti za stvaranje sustava koji imaju sposobnost samostalnog stjecanja znanja, prilagodbe novim informacijama i rješavanja zamršenih izazova. Kako AI nastavlja napredovati i podvrgnuti se istraživanju, obećava revoluciju u širokom rasponu područja u društvu, uključujući ali ne ograničavajući se na zdravstvo, obrazovanje, poslovanje i rekreaciju.

## **2.2. Karakteristike umjetne inteligencije**

Umjetna inteligencija definirana je nizom značajki koje joj omogućuju da se uhvati u koštac sa složenim zadacima, čineći je svestranom i relevantnom u raznim industrijama. Kao što su primijetili Goodfellow, Bengio i Courville (2016) sustavi korisničkog sučelja odlikuju se svojom sposobnošću samousmjerenog učenja, razumijevanja prirodnog jezika, donošenja odluka i osjetilne percepcije. Među tim osobinama, autonomno učenje ističe se kao posebno ključno. Kroz autonomno učenje, sustavi korisničkog sučelja mogu apsorbirati informacije i iskustva, prilagođavajući svoje ponašanje kako bi poboljšali svoju izvedbu. Srž ove mogućnosti su algoritmi strojnog učenja, koji koriste statističke tehnike za analizu podataka i identificiranje obrazaca. Koristeći te obrasce, sustavi mogu donositi zaključke i predviđati bez oslanjanja na eksplicitne upute. Ovaj aspekt omogućuje sustavima korisničkog sučelja da se neprestano usavršavaju i prilagođavaju okolnostima koje se mijenjaju, pokazujući se neprocjenjivim u dinamičnim okruženjima kao što su financijska tržišta i medicinska dijagnostika (McCarthy, 2007).

Sposobnost sustava korisničkog sučelja da razumiju i analiziraju ljudski jezik ključna je za razne primjene, od virtualnih pomoćnika do alata za prevođenje jezika. Obrada prirodnog jezika (NLP) igra ključnu ulogu u omogućavanju ovim sustavima da tumače i obrađuju unos teksta i govora na način koji oponaša ljudsko razumijevanje. Kroz tehnike kao što su raščlanjivanje, semantička analiza i prepoznavanje entiteta, NLP algoritmi mogu učinkovito dešifrirati i proizvesti jezik. Ova funkcionalnost služi kao temelj za interaktivne

aplikacije koje zahtijevaju besprijekornu komunikaciju s korisnicima, čime se u konačnici povećava zadovoljstvo i produktivnost korisnika (Jurafsky, Martin, 2019).

Umjetna inteligencija ima sposobnost obrade golemih količina podataka i korištenja tih informacija za donošenje informiranih odluka brzinom i preciznošću koja nadmašuje ljudske. Ova se sposobnost pokazala neprocjenjivom u područjima gdje je bitno brzo i točno donošenje odluka, poput područja autonomnih vozila, financijske analize i zdravstvene skrbi. Upotrebom algoritama za donošenje odluka AI može procijeniti rizik, učinkovito rasporediti resurse i predvidjeti buduće događaje na temelju prošlih podataka. Na primjer, u području autonomnih vozila, AI sustavi kontinuirano analiziraju podatke senzora kako bi u djeliću sekunde donijeli odluke u vezi s navigacijom i izbjegavanjem prepreka (Vaswani et. al., 2017).

Korištenjem senzora i kamera, UI sustavi imaju mogućnost promatranja svoje okoline, identificiranja različitih objekata i pojedinaca te reagiranja na bilo kakve promjene u okruženju. Kroz osjetilnu percepciju, UI sustavi mogu stupiti u kontakt s fizičkim svijetom na način koji podsjeća na ljudsku interakciju. Računalni vid, ključna tehnologija u ovoj domeni, koristi algoritme za prepoznavanje uzoraka za tumačenje vizualnih podataka. Ovaj skup vještina neophodan je za zadatke poput autonomne vožnje, robotskih funkcija i sigurnosnih mjera. Na primjer, autonomna vozila oslanjaju se na kamere i LIDAR senzore za izradu karata svog okruženja, otkrivanje prepreka i manevriranje u stvarnom vremenu (Brown et al., 2020).

Ove karakteristike omogućuju korištenje sustava korisničkog sučelja u različitim područjima, uključujući ali ne ograničavajući se na zdravstvenu skrb, autonomna vozila i personalizirane preporuke unutar e-trgovine. Unutar zdravstvenog sektora, sustavi korisničkog sučelja koriste se za zadatke kao što su analiza medicinskih slika, predviđanje ishoda liječenja i prilagođavanje terapija na temelju podataka o pojedinačnim pacijentima (Vaswani et. al., 2017., 6001). U području e-trgovine, algoritmi za preporuke iskorištavaju korisničke preferencije i podatke o ponašanju kako bi predložili proizvode koji su u skladu s interesima korisnika, u konačnici povećavajući prodaju i povećavajući zadovoljstvo kupaca. Integracija ovih značajki u više industrija ne samo da poboljšava učinkovitost i preciznost, već također olakšava

inovativne načine stupanja u kontakt s klijentima i poslovanja. Sa stalnim tehnološkim napretkom, očekuje se da će se mogućnosti i svestranost umjetne inteligencije proširiti, predstavljajući nove mogućnosti i prepreke u raznim sektorima.

## **2.3. Pojam prirodnih jezika**

Prirodni jezik je vrsta jezika koja se prirodno razvija među ljudima kroz komunikaciju i društvene interakcije. Obrada prirodnog jezika (NLP) ključni je aspekt umjetne inteligencije koji računalima omogućuje razumijevanje, tumačenje i odgovaranje na ljudski jezik. Kako su definirali Jurafsky i Martin [4], NLP pokriva širok spektar funkcija, kao što su prepoznavanje govora, razumijevanje teksta, generiranje jezika i strojno prevođenje. NLP algoritmi koriste metode iz računalne lingvistike, statistike i strojnog učenja za analizu i rukovanje lingvističkim podacima. Ovaj napredak u tehnologiji omogućuje računalima da razumiju jezik na sličan način kao i ljudi, dešifrirajući ne samo pojedinačne riječi već i zamršene jezične strukture i kontekste (Devlin et. al., 2018).

Prepoznavanje govora ključna je komponenta obrade prirodnog jezika (NLP), koja sustavima omogućuje pretvaranje izgovorenih riječi u pisani tekst. Ova se tehnologija naširoko koristi u raznim aplikacijama, kao što su virtualni pomoćnici kao što su Siri i Google Assistant, koji su sposobni razumjeti govorne upute i reagirati na njih. Proces prepoznavanja govora obuhvaća različite metode, uključujući akustično modeliranje, prepoznavanje fonema i jezično modeliranje, a sve zajedno rade kako bi jamčile preciznu i učinkovitu konverziju govora u tekst (Hochreiter, Schmidhuber, 1997., 1740).

Sposobnost razumijevanja teksta ključni je element obrade prirodnog jezika. Ova funkcionalnost omogućuje sustavima da dešifriraju i analiziraju temeljno značenje pisanog sadržaja, identificirajući odnose i veze između riječi i rečenica. Algoritmi za razumijevanje teksta igraju ključnu ulogu u raznim primjenama, kao što je analiza osjećaja, koja uključuje procjenu osjećaja i

emocija prenesenih u tekstu, te u pronalaženju informacija, gdje se koriste za prepoznavanje i izdvajanje relevantnih informacija iz opsežnih zbirki teksta (Mikolov et. al., 2013).

Generiranje jezika je vrhunska tehnologija koja sustavima omogućuje stvaranje teksta koji glatko teče i zvuči kao da ga je napisao čovjek. Ovaj napredni alat koristi se u raznim aplikacijama, uključujući automatsku izradu sadržaja i chatbotove, gdje sustav treba dati odgovore koji imaju smisla kao odgovor na upite korisnika. Algoritmi koji stoje iza generiranja jezika oslanjaju se na sofisticirane neuronske mreže kako bi analizirali obrasce u jeziku i proizveli tekst koji je ne samo gramatički točan, već i kontekstualno prikladan. Korištenjem ovih algoritama programeri mogu osigurati da je tekst koji generiraju njihovi sustavi koherentan i relevantan za razgovor koji se vodi (Sutskever, Vinyals, 2014., 3109).

Strojno prevođenje ključni je aspekt obrade prirodnog jezika, omogućavajući besprijekoran prijevod teksta s jednog jezika na drugi bez ljudske intervencije. Ova se tehnologija koristi u popularnim alatima kao što je Google Translate, koristeći sofisticirane algoritme za brzu analizu i tumačenje jezika u stvarnom vremenu. Korištenjem algoritama za strojno prevođenje, sustavi mogu učinkovito uhvatiti jezične strukture i dešifrirati značenje, osiguravajući precizan i tečan prevod.

Napredak NLP tehnologija otvorio je put za razvoj vrhunskih modela kao što su ChatGPT i Copilot. Ovi modeli koriste najsuvremenije algoritme dubokog učenja za proizvodnju prirodnog i kohezivnog teksta kao odgovor na upite korisnika. Korištenjem opsežnih skupova podataka i zamršenih neuronskih mreža, ovi modeli mogu shvatiti kontekst i generirati informativne i relevantne odgovore. ChatGPT, na primjer, može se uključiti u dijalog s korisnicima, adresirati upite i dostaviti informacije na način koji je vrlo sličan ljudskoj interakciji. S druge strane, Copilot služi kao vrijedan alat za programere, nudeći prijedloge koda i rješenja temeljena na unosu i kontekstu programiranja (Silver et. al., 2016., 486 – 487).

Općenito, umjetna inteligencija revolucionira način na koji se bavimo tehnologijom i obrnuto, u nizu oblika i funkcija. Napredak obrade prirodnog jezika (NLP) omogućuje računalima razumijevanje i analizu ljudskog jezika, što dovodi do besprijekornije i intuitivnije interakcije s tehnologijom. Predviđa

se da će stalne inovacije u dizajnu korisničkog sučelja (UI) imati ključnu ulogu u oblikovanju budućnosti tehnologije i društva, nudeći nove načine za poboljšanje komunikacije, učinkovitosti i ukupne produktivnosti (Silver et. al., 2016., 488). Kako NLP modeli postaju sve sofisticiraniji, mogućnosti za njihovu upotrebu će se širiti, uvodeći duboke transformacije u različitim industrijama i svakodnevnim iskustvima.

## **3. FRONTEND PROGRAMIRANJE WEBA**

Frontend web programiranje obuhvaća segment razvoja web stranica i aplikacija usmjerenih na korisničko sučelje i način interakcije korisnika s web sadržajem. Ovaj aspekt razvoja ključan je za jamstvo da su web stranice ne samo funkcionalne, već i vizualno privlačne, intuitivne i dostupne svim korisnicima. Uz rastuću proliferaciju web aplikacija i potražnju za responzivnim dizajnom, frontend programiranje postalo je jedno od najkritičnijih područja u području razvoja softvera. U ovom poglavlju prikazati će se temeljne koncepte povezane s web tehnologijama i karakteristike frontend programiranja te raspraviti o različitim programskim jezicima, bibliotekama i sučeljima za programiranje aplikacija (API) koji se obično koriste u ovom području.

### **3.1. Općenito o webu , web tehnologiji i izradi web stranica**

World Wide Web je opsežna mreža koja se sastoji od međusobno povezanih dokumenata i izvora, a svi su dostupni putem Interneta. Korisnici mogu komunicirati s tim elementima putem web preglednika, koji služe kao ulaz u ovaj golemi digitalni krajolik. Početak Weba može se pratiti unatrag do kasnih 1980-ih kada je Tim Berners-Lee predstavio njegovu prvu iteraciju. Od svog nastanka, Web je doživio značajnu transformaciju, razvijajući se u sofisticirani ekosustav koji obuhvaća široku lepezu aplikacija i usluga. Ove temeljne tehnologije — HTML, CSS i JavaScript — rade u tandemu kako bi stvorile bogata, interaktivna web iskustva koja korisnici očekuju. Kao što je primijetio World Wide Web Consortium (W3C) 2018., sinergija između ovih tehnologija ključna je za sposobnost weba da podržava različite i složene funkcionalnosti, što ga čini nezamjenjivim alatom u modernom digitalnom dobu (W3C, 2018., 12). U središtu funkcionalnosti weba nalazi se nekoliko temeljnih tehnologija. HTML, ili jezik za označavanje hiperteksta, igra ključnu ulogu osiguravajući strukturalni okvir za web sadržaj, omogućujući programerima da organiziraju i učinkovito prezentiraju informacije na web stranicama. CSS, poznat kao

Cascading Style Sheets, odgovoran je za vizualnu prezentaciju, uključujući izgled, boje i fontove, čime se poboljšava estetska privlačnost i korisničko iskustvo web stranica. JavaScript, svestrani skriptni jezik, dodaje sloj interaktivnosti i dinamičnosti web stranicama, omogućujući značajke poput animacija, provjera valjanosti obrazaca i ažuriranja u stvarnom vremenu.

HTML služi kao temeljna tehnologija interneta, ključna za uspostavljanje strukture i značenja web stranica. Svaka komponenta na web stranici, uključujući zaglavlja, tekstualne blokove, slike i hiperveze, specificirana je pomoću HTML oznaka. Tijekom vremena, kako su zahtjevi i očekivanja korisnika i web dizajnera rasli, HTML je doživio značajan napredak. Ova su ažuriranja uvela podršku za razne nove značajke, poput mogućnosti ugradnje multimedijskog sadržaja i uključivanja interaktivnih elemenata izravno u web stranice (W3C, 2018., 14).

CSS, kao moćan stilski jezik, omogućuje dizajnerima i programerima da jasno odvoje vizualnu prezentaciju web stranice od njezine osnovne strukture. Korištenjem CSS-a moguće je pažljivo upravljati vizualnim aspektima web elemenata, kao što su boje, tipografija, razmaci i cjelokupni izgled. Štoviše, CSS olakšava responzivni dizajn, koji dinamički prilagođava izgled web stranice na različitim uređajima i dimenzijama zaslona (Meyer, 2015., str. 32). Ova prilagodljivost ključna je za osiguravanje da web stranice ostanu praktične i jednostavne za korištenje na mobilnim uređajima, što je nužnost koja je postala sve važnija usporedo s porastom broja mobilnih korisnika. S druge strane, JavaScript donosi sloj dinamičnosti i interaktivnosti na web stranice, omogućujući korisnicima da se bave elementima stranice bez potrebe za punim ponovnim učitavanjem. Koristeći JavaScript, programeri mogu izraditi interaktivne obrasce, kreirati animacije, manipulirati DOM-om (Document Object Model) i uspostaviti veze s pozadinskim poslužiteljima za dohvaćanje ili prijenos podataka (Duckett, 2011., 67). Ovaj skriptni jezik je ključan u razvoju bogatih, interaktivnih korisničkih iskustava, koja su postala zaštitni znak suvremenih web aplikacija.

Razvoj web stranice ne obuhvaća samo tehničku stranu stvari, već uključuje i značajne kreativne dimenzije, poput dizajna i korisničkog iskustva (UX/UI). Proces web dizajna uključuje izradu vizualnog identiteta stranice koji uključuje strateški raspored elemenata, odabir boja, tipografiju i razne druge grafičke

komponente. Primarni cilj dizajna je stvoriti vizualno privlačno i korisniku jednostavno sučelje. S druge strane, UX dizajn fokusiran je na poboljšanje cjelokupnog korisničkog iskustva, nastojeći interakciju s web stranicom učiniti što jednostavnijom i ugodnijom (Krug, 2014., 45).

Na tehničkom planu, web razvoj obuhvaća proces prevođenja dizajnerskih koncepata u funkcionalni kod. To uključuje ne samo pisanje potrebnog koda, već i integraciju web stranice s nizom usluga, baza podataka i API-ja kako bi se poboljšala njezina funkcionalnost. Web programeri imaju ključnu odgovornost za optimizaciju web stranice za brzinu i performanse. Moraju osigurati da korisnici uživaju u besprijekornom iskustvu, bez produljenih vremena učitavanja i tehničkih grešaka. To uključuje korištenje različitih tehnika i najboljih praksi kako bi web stranica radila glatko i učinkovito. Prema Krugu (2014., 47), postizanje ove razine optimizacije ključno je za pružanje najboljeg mogućeg iskustva korisnicima.

S obzirom na brzu evoluciju web tehnologija i promjenjiva očekivanja korisnika, proces razvoja web stranica zahtijeva stalna poboljšanja i prilagodbu novim standardima. Na primjer, pojava HTML5 i CSS3 uvela je mnoštvo inovacija koje olakšavaju sofisticiranije funkcije i poboljšanu izvedbu. Nadalje, JavaScript okviri poput Reacta i Angulara revolucionirali su mogućnost izrade zamršenih i vrlo interaktivnih aplikacija. Ova poboljšanja zahtijevaju da web programeri budu u toku s najnovijim trendovima i kontinuirano usavršavaju svoje vještine kako bi zadovoljili rastuće i promjenjive zahtjeve web razvoja (Duckett, 2011., 78).

U modernom krajoliku razvoja web stranica, proces je postao znatno zamršeniji zbog uključivanja različitih ključnih elemenata kao što su optimizacija za tražilice (SEO), mjere sigurnosti podataka, dostupnost weba i osiguravanje kompatibilnosti na mnoštvu preglednika i uređaja. Ovi dodatni slojevi složenosti zahtijevaju zajednički pristup, gdje programeri i dizajneri moraju raditi u bliskoj koordinaciji kako bi proizveli optimalnu web stranicu. Ovaj zajednički napor neophodan je za uspješno prevladavanje višestrukih izazova koji se pojavljuju tijekom izrade web stranice, kao što je opisao Meyer (2015., 102). Svaki aspekt, od poboljšanja vidljivosti stranice u rezultatima tražilice do zaštite korisničkih podataka, osiguravanja da svi korisnici, uključujući one s invaliditetom, mogu pristupiti stranici, i održavanja



funkcionalnosti na različitim platformama i uređajima, doprinosi ukupnoj složenosti. Posljedično, postizanje najboljeg mogućeg ishoda u razvoju web stranica sada zahtijeva holistički i integrirani pristup, kombinirajući tehničku stručnost s kreativnim dizajnom i precizno planiranje s prilagodljivom izvedbom.

### **3.2. Pojam i obilježja frontend programiranja weba**

Frontend programiranje weba odnosi se na segment razvojnog procesa u koji korisnici izravno sudjeluju i promatraju ga. Za razliku od backend razvoja, koji upravlja poslužiteljima, bazama podataka i temeljnom poslovnom logikom, frontend razvoj se koncentrira na korisničko sučelje (UI) i korisničko iskustvo (UX). Središnji aspekti razvoja sučelja uključuju osiguravanje odziva i poticanje interaktivnosti. Responzivnost se odnosi na sposobnost web stranica i aplikacija da se neprimjetno prilagode različitim uređajima i veličinama zaslona, jamčeći da će ostati funkcionalne i vizualno ugodne na različitim platformama, od stolnih računala do mobilnih telefona. Interaktivnost, s druge strane, omogućuje korisnicima da se bave elementima na web stranici, kao što su obrasci, navigacijski izbornici i gumbi, čime se značajno poboljšava cjelokupno korisničko iskustvo (Duckett, 2011., 34).

Temeljni aspekt razvoja sučelja je integracija suvremenih metodologija kao što su mobilna pristupačnost i poboljšanje performansi. Mobilna pristupačnost, koja se često naziva dizajnom na prvom mjestu mobitela, pojavila se kao norma u razvoju sučelja zbog sve većeg broja korisnika koji pristupaju web stranicama putem pametnih telefona i tableta. Ovaj pristup uključuje početnu izradu web stranica za kompaktne zaslone mobilnih uređaja i njihovu naknadnu prilagodbu za zaslone stolnih računala. Takav proces zahtijeva pažljivu brigu kako bi se osiguralo da web stranice ostanu funkcionalne, brze i estetski ugodne na manjim ekranima, a da istovremeno očuvaju optimalno korisničko iskustvo na većim uređajima (Marcotte, 2010., 23).

Osim osiguravanja mobilne pristupačnosti, optimizacija performansi kritičan je aspekt razvoja sučelja. Brzina kojom se web stranica učitava ima značajan

utjecaj na korisničko iskustvo jer stranice koje se sporo učitavaju mogu dovesti do frustracije korisnika i viših stopa napuštanja stranice. Kako bi se uhvatili u koštac s tim problemima, programeri sučelja koriste različite strategije optimizacije. To uključuje smanjivanje CSS i JavaScript datoteka kako bi se smanjila njihova veličina, smanjivanje broja HTTP zahtjeva koje postavlja preglednik, komprimiranje slika radi uštede propusnosti i korištenje mreža za isporuku sadržaja (CDN) za učinkovitiju distribuciju sadržaja. Ove tehnike zajedno pridonose bržem učitavanju stranica, čime se poboljšava cjelokupno korisničko iskustvo (Wroblewski, 2009., 57).

Razvoj frontenda ne uključuje samo stvaranje vizualnih i interaktivnih aspekata web stranica i aplikacija, već i integraciju s raznim API-jima i vanjskim uslugama. Te su integracije ključne za poboljšanje funkcionalnosti i korisničkog iskustva digitalnih platformi. Na primjer, sustavi za autentifikaciju korisnika omogućuju sigurne procese prijave, dok pristupnici plaćanja olakšavaju besprijekorne i sigurne financijske transakcije. Dodatno, povezivanje s društvenim mrežama omogućuje značajke kao što je prijava na društvene mreže, gdje korisnici mogu pristupiti stranici koristeći svoje postojeće vjerodajnice za društvene mreže. Ova sposobnost komunikacije i interakcije s vanjskim sustavima proširuje mogućnosti web stranica i aplikacija, čineći ih svestranijima i lakšima za korištenje (Fielding, 2000., 19).

Ključni aspekt razvoja sučelja je poštivanje web standarda i osiguravanje pristupačnosti. Programeri moraju jamčiti da njihove web stranice besprijekorno rade na različitim preglednicima i da su u skladu sa smjernicama koje su postavile organizacije poput World Wide Web Consortiuma (W3C). Pristupačnost uključuje izradu web stranica koje mogu koristiti osobe s invaliditetom, uključujući one koji imaju oštećenje vida ili sluha. To uključuje korištenje semantičkog HTML-a, osiguravanje kompatibilnosti stranice s čitačima zaslona i nuđenje alternativnih opcija za multimedijski sadržaj (W3C, 2018., 56). U biti, programeri imaju zadatak stvoriti inkluzivno digitalno okruženje koje zadovoljava potrebe svih korisnika, bez obzira na njihova fizička ograničenja. Ovo ne samo da poboljšava korisničko iskustvo, već također osigurava usklađenost sa pravnim standardima i najboljom praksom u web razvoju. Razmatranja pristupačnosti mogu uključivati implementaciju navigacije tipkovnicom, pružanje tekstualnih

opisa za slike i korištenje odgovarajućih kontrasta boja za bolju čitljivost. Usredotočujući se na ove elemente, programeri pridonose pravednijem internetu prilagođenijem korisniku.

Osim bezbrojnih tehničkih razmatranja, razvoj sučelja obuhvaća značajan kreativni aspekt koji se odnosi na vizualnu privlačnost web stranica. To uključuje zajednički napor između dizajnera i programera kako bi se zajamčilo da su web stranice ne samo operativno učinkovite, već i vizualno uvjerljive i dosljedne vizualnom identitetu marke. Estetika stranice, u kombinaciji s njezinom funkcionalnošću, igra ključnu ulogu u oblikovanju korisničkog iskustva. Ova sinergija između dizajna i upotrebljivosti značajno utječe na ukupnu percepciju kvalitete web stranice ili aplikacije (Garrett, 2010., 72).

Naposljetku, frontend web razvoj predstavlja izazovan, ali vrlo važan aspekt izrade web stranica i aplikacija. Zahtijeva jedinstvenu mješavinu tehničke stručnosti, kreativne vizije i dubokog razumijevanja korisničkih zahtjeva. Ovaj višestruki pristup omogućuje razvojnim programerima da proizvedu web sadržaj koji ne samo da ispunjava stroge kriterije za funkcionalnost, izvedbu i vizualnu privlačnost, već također pruža iznimno korisničko iskustvo. Usklađujući ove elemente, frontend programeri igraju ključnu ulogu u osiguravanju da su web aplikacije učinkovite i privlačne, čime ispunjavaju visoka očekivanja modernih korisnika.

### **3.3. Jezici koji se koriste u frontend programiranju**

Frontend programiranje uvelike ovisi o nekoliko bitnih jezika koji olakšavaju razvoj i upravljanje web sadržajem. U središtu ovih jezika je HTML (Hypertext Markup Language), koji djeluje kao temeljni blok za izgradnju web stranica (Slika 1). HTML je odgovoran za pružanje strukturalnog okvira kroz niz oznaka koje ocrtavaju različite elemente na stranici, uključujući ali ne ograničavajući se na naslove, odlomke, slike i hiperveze. Kao primarni jezik za izradu web stranica, HTML ima ključnu ulogu u osiguravanju semantičke organizacije sadržaja, što je od vitalnog značaja i za pristupačnost i za optimizaciju tražilice (SEO). Pojava HTML5 uvela je mnoštvo novih značajki

koje poboljšavaju integraciju multimedijских elemenata i naprednih funkcionalnosti, eliminirajući potrebu za dodatnim dodacima (W3C, 2018., str. 20). Ova evolucija označava značajan korak naprijed u mogućnostima stvaranja i upravljanja web sadržajem.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

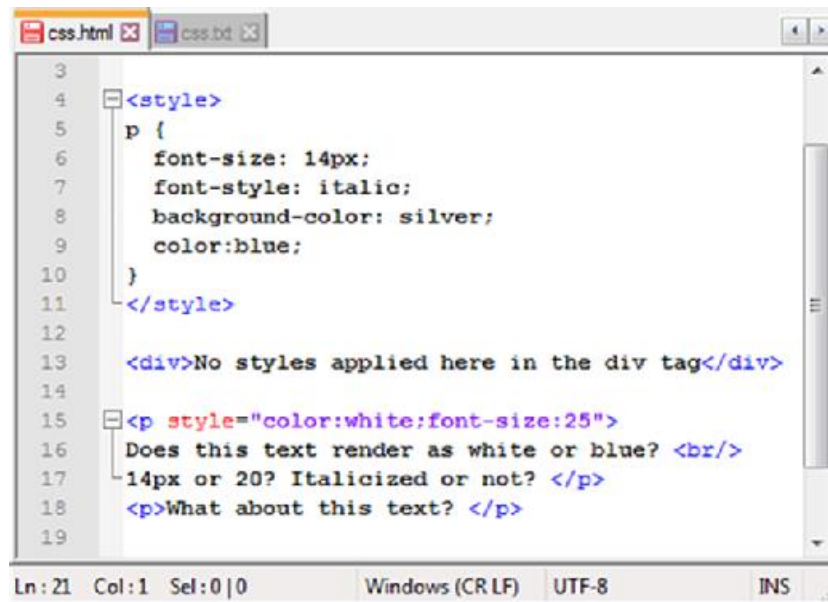
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

**Slika 1. Primjer programškoga jezika HTML**

*(<https://webprogtutorials.wordpress.com/html-examples/>)*

Cascading Style Sheets (CSS) ključni su za stiliziranje i dizajn web stranica, pružajući programerima alate za kontrolu vizualne prezentacije elemenata na web stranici. Definiranjem boja, fontova, izgleda, margina i drugih aspekata dizajna, CSS pomaže u postizanju jedinstvenog i privlačnog izgleda na svim web stranicama (Slika 2). Jedna od značajnih prednosti CSS-a je odvajanje sadržaja od prezentacije, što omogućuje primjenu istih stilova na više stranica odjednom. Ovo odvajanje pojednostavljuje proces održavanja i ažuriranja dizajna web stranice. Najnovija iteracija, CSS3, uvodi mnoštvo naprednih značajki kao što su animacije, fleksibilni izgledi kroz Flexbox i sustav rasporeda rešetki poznat kao CSS Grid. Ova poboljšanja značajno povećavaju vizualnu privlačnost i prilagodljivost web stranica, čineći moderni web dizajn dinamičnijim i svestranijim (Meyer, 2015., 45).



```
3
4 <style>
5   p {
6     font-size: 14px;
7     font-style: italic;
8     background-color: silver;
9     color: blue;
10  }
11 </style>
12
13 <div>No styles applied here in the div tag</div>
14
15 <p style="color:white;font-size:25">
16 Does this text render as white or blue? <br/>
17 14px or 20? Italicized or not? </p>
18 <p>What about this text? </p>
19
```

Slika 2. Prikaz CSS programskog jezika

(<https://www.theserverside.com/definition/cascading-style-sheet-CSS>)

JavaScript igra ključnu ulogu u poboljšanju interaktivnosti i dinamičke prirode web stranica. Olakšava razvoj različitih elemenata, uključujući padajuće izbornike, animacije, provjere obrazaca i ažuriranja sadržaja u stvarnom vremenu, a sve to bez potrebe za ponovnim učitavanjem stranice. Radeći na strani klijenta, JavaScript osigurava brzu i brzu interakciju korisnika. Nadalje, pojava JavaScript okvira kao što su React, Angular i Vue.js revolucionirala je web razvoj, omogućujući stvaranje zamršenih Single Page Applications (SPA) koje pružaju korisničko iskustvo usporedivo s iskustvom aplikacija za stolna računala (Duckett, 2011., 78).

Kao dodatak temeljnim programskim jezicima, razvoj sučelja uključuje sofisticirane alate kao što su predprocesori i nadskupovi jezika. Jedan istaknuti primjer je Sass (Syntactically Awesome Stylesheets), koji djeluje kao predprocesor za CSS. Sass poboljšava CSS dodavanjem značajki kao što su varijable, ugniježđena pravila i mixins, čime se olakšava stvaranje modularnijeg i učinkovitijeg koda. Korištenjem Sass-a programeri mogu postići organiziraniju strukturu za stiliziranje, minimizirati ponavljajući kod i pojednostaviti upravljanje opsežnim projektima (Meyer, 2015., 89).

TypeScript predstavlja značajno poboljšanje JavaScripta uključivanjem statičkog tipkanja, što uvelike olakšava razvoj i održavanje velikih aplikacija. Uvođenje statičkih tipova omogućuje programerima provjeru pogrešaka

tijekom kompajliranja, a ne tijekom izvođenja, čime se smanjuje vjerojatnost pogrešaka u zamršenim projektima. Ova je značajka osobito korisna u složenim bazama kodova gdje rano otkrivanje potencijalnih problema može uštedjeti znatno vrijeme i trud. Iako je TypeScript stekao značajnu privlačnost u razvoju aplikacija temeljenih na Angularu, njegovo usvajanje se širi na razna druga razvojna okruženja. Ovo široko prihvaćanje može se pripisati njegovoj besprijekornoj kompatibilnosti s JavaScriptom, omogućujući programerima da integriraju TypeScript u postojeće projekte s relativnom lakoćom (Bierman, Abadi i Torgersen, 2014., 12).

Programerima su dostupni i razni drugi alati, poput predprocesora kao što je LESS za CSS. LESS olakšava pisanje i upravljanje listovima stilova dodavanjem značajki koje nisu dostupne u uobičajenom CSS-u, kao što su varijable, ugniježđivanje i miješanje. To može uvelike pojednostaviti proces stvaranja i održavanja stilova. Osim toga, postoje transpileri kao što je Babel, koji programerima omogućuju korištenje najnovijih značajki JavaScripta čak i ako okruženje u kojem se kôd izvodi izvorno ne podržava te značajke. Babel preuzima moderni JavaScript kôd i transformira ga u verziju koja je kompatibilna sa starijim okruženjima. Ovi su alati neprocjenjivi za programere sučelja jer pomažu u pisanju koda koji je ne samo moderniji i čitljiviji, već ga je i lakše održavati tijekom vremena. Korištenjem ovih alata, programeri mogu značajno smanjiti količinu vremena koju troše na razvojne zadatke, što im omogućuje da se više koncentriraju na poboljšanje funkcionalnosti i poboljšanje korisničkog iskustva. To u konačnici dovodi do učinkovitijeg i djelotvornijeg procesa razvoja. Prema izvješću tvrtke Babel iz 2019. (str. 25), ovi su alati ključni za pojednostavljenje radnih procesa i osiguravanje da programeri mogu držati korak s krajolikom web tehnologija koji se brzo razvija.

U biti, jezici koji se koriste u frontend programiranju ključni su za izgradnju suvremenih, funkcionalnih i vizualno atraktivnih web stranica. Integracija temeljnih jezika kao što su HTML, CSS i JavaScript s naprednim alatima kao što su Sass, TypeScript i Babel omogućuje programerima sučelja da konstruiraju zamršene aplikacije koje se pridržavaju rigoroznih standarda korisničkog iskustva i tehničke izvedbe. Kako web tehnologije budu napredovale, ti će jezici i alati biti ključni u definiranju budućeg krajolika weba.

### 3.4. Općenito o bibliotekama za web

Biblioteke za web programiranje nude programerima zbirku unaprijed napisanog koda i funkcionalnosti koje značajno ubrzavaju proces izgradnje web aplikacija. Korištenjem ovih biblioteka, programeri mogu pojednostaviti svoj radni tijek, jer omogućuju opsežno ponovno korištenje koda i nude niz unaprijed definiranih komponenti i funkcija dizajniranih za rješavanje uobičajenih izazova u web razvoju. Ovaj pristup omogućuje programerima da zaobiđu potrebu za stvaranjem rješenja za osnovne funkcionalnosti ispočetka, čime im se pruža sloboda da se koncentriraju na jedinstvene i specifične zahtjeve svojih projekata. Posljedično, korištenje ovih biblioteka ne samo da povećava učinkovitost, već također promiče dosljednost i pouzdanost u procesu razvoja.

React, široko rasprostranjena JavaScript biblioteka koju je stvorio Facebook, stekla je ogromnu popularnost među programerima. Ova biblioteka olakšava razvoj sofisticiranih korisničkih sučelja korištenjem metodologije zasnovane na komponentama (Slika 3). U ovom pristupu, aplikacija se razlaže na manje, ponovno upotrebljive komponente. Ova modularna struktura ne samo da poboljšava organizaciju koda, već također pojednostavljuje održavanje i ponovno korištenje komponenti u različitim dijelovima aplikacije. Jedna od istaknutih značajki Reacta je njegova uporaba virtualnog DOM-a (Document Object Model), što značajno poboljšava performanse aplikacije. Umjesto ponovnog renderiranja cijele stranice, React ažurira samo one dijelove korisničkog sučelja koji su prošli kroz promjene. Ovaj selektivni mehanizam ažuriranja osigurava učinkovitije i responzivnije korisničko iskustvo (React, 2024).

```
Windows PowerShell
PS D:\> mkdir ReactProjects

Directory: D:\

Mode                LastWriteTime         Length Name
----                -
d-----           06.06.2022.    10:56         ReactProjects

PS D:\> cd .\ReactProjects\
PS D:\ReactProjects> npx create-react-app my-app
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
Creating a new React app in D:\ReactProjects\my-app.
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
npm WARN config init.module Use '--init-module' instead.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
npm WARN config init.module Use '--init-module' instead.
added 1381 packages in 1m
199 packages are looking for funding
  run 'npm fund' for details
Initialized a git repository.
Installing template dependencies using npm...
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
npm WARN config init.module Use '--init-module' instead.
npm WARN deprecated source-map-resolve@0.6.0: See https://github.com/lydell/source-map-resolve#deprecated
added 39 packages in 8s
199 packages are looking for funding
  run 'npm fund' for details
Removing template package using npm...
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
npm WARN config init.module Use '--init-module' instead.
removed 1 package, and audited 1420 packages in 4s
199 packages are looking for funding
  run 'npm fund' for details
6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.
Created git commit.
Success! Created my-app at D:\ReactProjects\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

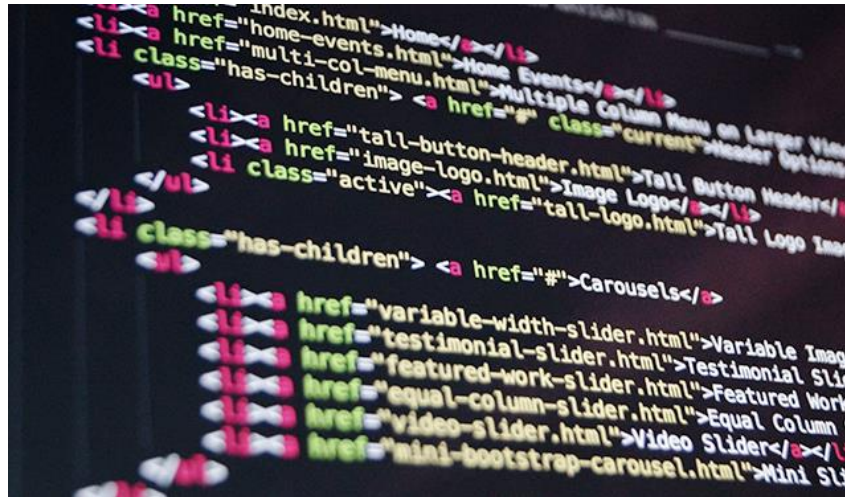
Happy hacking!
```

Slika 3. Prikaz instalacije React biblioteke

[\(https://blog.openreplay.com/how-to-build-your-own-react-components-library/\)](https://blog.openreplay.com/how-to-build-your-own-react-components-library/)

Još jedna dobro prihvaćena biblioteka je Angular, koju je stvorila Google zajednica. Angular nadilazi običnu biblioteku; on služi kao potpuni okvir namijenjen razvoju dinamičnih web aplikacija. Nudi sveobuhvatno rješenje za različite aspekte razvoja aplikacija, uključujući upravljanje stanjem, validaciju obrazaca, navigacijsko usmjeravanje, interakcije s API-jem, testiranje i poboljšanja performansi. Angular koristi TypeScript, nadskup JavaScript-a koji uključuje statičko tipiziranje, čime se pojednostavljuje proces izgradnje složenih aplikacija. Osim toga, Angular podržava dvosmjerno vezivanje podataka, značajku koja osigurava da se sve izmjene napravljene u korisničkom sučelju odmah odražavaju u osnovnom podatkovnom modelu, i obrnuto (Angular, 2024).





Slika 4. Prikaz html u Angularu

(<https://medium.com/@anooprvarrier/angular-library-the-complete-developer-guide-7238742a7842>)

Pored Reacta i Angulara, postoje brojne druge dobro prihvaćene biblioteke, kao što je Vue.js, koja je poznata po svom jednostavnom pristupu i prilagodljivosti. Vue.js nudi postupan proces učenja, omogućujući programerima da započnu jednostavnim poboljšavanjem statičkih web stranica interaktivnim značajkama, a zatim postupno napreduju do stvaranja sofisticiranih, jednodimenzionalnih aplikacija. Biblioteka koristi reaktivna svojstva, osiguravajući da se korisničko sučelje automatski ažurira kao odgovor na promjene podataka, čime se pojednostavljaju procesi razvoja i održavanja aplikacija (You, 2016., 33).

Osim ovih primarnih biblioteka, postoji niz specijaliziranih biblioteka koje se bave specifičnim aspektima web razvoja. Na primjer, jQuery je dobro uspostavljena biblioteka koja olakšava zadatke poput manipulacije DOM-om, rukovanja događajima i animacija. Međutim, njezino korištenje je opalo u korist modernijih alata kao što su React i Angular. Još jedan primjer je D3.js, koja je posebno usmjerena na stvaranje interaktivnih vizualizacija podataka. Dodatno, Lodash nudi niz funkcija koje pojednostavljuju operacije s nizovima, objektima i kolekcijama.

### 3.5. Općenito o aplikacijskim programskim sučeljima

Sučelja za programiranje aplikacija, koja se obično nazivaju API-ji, služe kao vitalni kanali koji olakšavaju komunikaciju između prednjih aplikacija i pozadinskih poslužitelja ili usluga. Ova sučelja nude dobro definirane metode za dohvaćanje podataka, podnošenje podataka i interakciju sustava korištenjem standardiziranih protokola kao što je Representational State Transfer (REST) ili GraphQL. Važnost API-ja u suvremenom web razvoju ne može se precijeniti, budući da oni omogućuju razdvajanje frontend i backend funkcionalnosti. Ovo odvajanje povećava fleksibilnost i skalabilnost aplikacija, olakšavajući neovisno upravljanje, ažuriranje i proširenje različitih komponenti. Prema Fieldingu (2000., 15), ovaj arhitektonski pristup je ključan u poticanju robusnijih i prilagodljivijih softverskih ekosustava.

RESTful API-ji iskorištavaju HTTP protokol za upravljanje resursima pomoću metoda kao što su GET, POST, PUT i DELETE. Ova metoda olakšava izravnu i učinkovitu komunikaciju između frontend aplikacija i poslužitelja. Nasuprot tome, GraphQL, koji je stvorio Facebook, omogućuje programerima da ocrtaju točne podatke koji su im potrebni. Ova specifičnost minimizira opterećenje mreže i poboljšava performanse aplikacija (Facebook, 2015., 18). Iskorištavanje API-ja omogućuje programerima sučelja da stvaraju sofisticirane i dinamične aplikacije sposobne za rad u stvarnom vremenu, interakciju s bazama podataka, rukovanje autentifikacijom korisnika i isporuku prilagođenog sadržaja. Ovaj modularni pristup ne samo da ubrzava proces razvoja, već i pojednostavljuje održavanje aplikacija (Fielding, 2000., 25). Integriranjem API-ja programeri mogu izraditi aplikacije koje bolje reagiraju i prilagođavaju se potrebama korisnika, osiguravajući besprijekorno i privlačno korisničko iskustvo. Ova fleksibilnost također omogućuje integraciju usluga trećih strana, omogućujući frontend aplikacijama da prošire svoju funkcionalnost izvan osnovnih značajki i uključe širok raspon vanjskih izvora podataka i usluga. Posljedično, programeri se mogu usredotočiti na poboljšanje korisničkog sučelja i iskustva dok prebacuju složene pozadinske operacije na API-je. Ovo odvajanje briga olakšava čišću arhitekturu koda i promiče ponovnu upotrebu, čineći cjelokupni proces razvoja učinkovitijim i lakšim za upravljanje

## **4. NAČINI INTEGRACIJE UMJETNE INTELIGENCIJE U FRONTEND PROGRAMIRANJU**

Umjetna inteligencija (AI) postupno postaje temelj u području web razvoja, mijenjajući način na koji se web aplikacije konceptualiziraju i konstruiraju. Uključivanje AI u frontend programiranje olakšava razvoj dinamičnijih, interaktivnijih i inteligentnijih korisničkih sučelja koja se mogu neprimjetno prilagoditi potrebama korisnika. Tehnologije kao što su strojno učenje, obrada prirodnog jezika i računalni vid sve se više koriste u sučelnim aplikacijama kako bi se poboljšalo korisničko iskustvo i ukupna izvedba web stranica. U nastavku rada prikazati će se široko korištene biblioteke i alate za AI u razvoju sučelja, ispitati nove trendove, istražiti sinergije između tehnologija sučelja i umjetne inteligencije za izradu sofisticiranih web aplikacija i pozabaviti se izazovima i budućim izgledima programiranja sučelja s integriranom umjetnom inteligencijom.

### **4.1. Pregled popularnih biblioteka i alata za umjetnu inteligenciju u frontendu**

Istaknuti i naširoko prihvaćen alat za uključivanje umjetne inteligencije u razvoj sučelja je TensorFlow.js (Slika 5). Ova robusna biblioteka olakšava izvođenje i obuku modela strojnog učenja izravno unutar web preglednika. Korištenjem TensorFlow.js, programeri mogu izvoditi obuku i izvođenje modela u stvarnom vremenu bez potrebe za pozadinskim poslužiteljima, čime se poboljšava iskustvo korisničkog sučelja. Knjižnica je svestrana i podržava širok spektar aplikacija strojnog učenja, kao što su klasifikacija slika, prepoznavanje teksta i generiranje sadržaja (TensorFlow, 2019., str. 42).

```
const model = tf.sequential({
  layers: [tf.layers.dense({ inputShape: [1], units: 1 })]
});

model.compile({ optimizer: "sgd", loss: "meanSquaredError" });

const xs = tf.tensor([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0]);
const ys = tf.tensor([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0]);

await model.fit(xs, ys, { epochs: 1000 });

model.predict(tf.tensor([10.0])).print();
```

Slika 5. Prikaz koda izgradnje modela sa tensorflow.js

(<https://www.tensorflow.org/js>)

Brain.js ističe se kao široko korištena biblioteka skrojena za razvoj sučelja, s jakim naglaskom na jednostavnost korisnika. Pruža okvir za izgradnju i uključivanje neuronskih mreža u web aplikacije, čime se omogućuje besprijekorna implementacija naprednih značajki korisničkog sučelja kao što su prediktivni unos teksta i personalizirane preporuke. Knjižnica je opremljena intuitivnim API-jem koji pojednostavljuje proces implementacije, što je čini privlačnom opcijom za programere koji žele unaprijediti svoje web aplikacije s mogućnostima strojnog učenja (Reardon, 2020., 18).

Dodatno, ml5.js široko je korištena biblioteka koja je izgrađena na temelju TensorFlow.js. Posebno je dizajniran da pojednostavi integraciju strojnog učenja u korisnička sučelja za kreativne i interaktivne napore. Primarni cilj ml5.js je omogućiti programerima da iskoriste sofisticirane modele strojnog učenja s minimalnom količinom koda. Ova biblioteka nudi niz funkcionalnosti, uključujući prepoznavanje slika, analizu zvuka i generiranje teksta, kao što je primijetio Runway (2020., 55). Korištenje ovih biblioteka ne samo da usmjerava proces uključivanja korisničkih sučelja u razvoj sučelja, već također omogućuje razvojnim programerima da kreiraju zamršene aplikacije koje pružaju bogato i privlačno korisničko iskustvo. Smanjivanjem prepreka za ulazak u strojno učenje, ml5.js potiče više inovacija i eksperimentiranja,

posebno u područjima gdje je kreativno izražavanje najvažnije. Time se promiče demokratizacija tehnologije, omogućujući širem rasponu pojedinaca, od umjetnika do edukatora, da u svoj rad ugrade napredne računalne tehnike bez potrebe za opsežnom tehničkom stručnošću.

## **4.2. Kombinacija frontend tehnologija i AI za stvaranje kompleksnih web aplikacija**

Spoj frontend tehnologija s umjetnom inteligencijom utire put za razvoj vrlo naprednih i zamršenih web aplikacija koje pružaju profinjeno korisničko iskustvo dok trenutačno izvršavaju složene zadatke. Ova harmonična mješavina omogućuje programerima da osmisle aplikacije koje nisu samo interaktivne i prilagodljive, već i vješte u rukovanju golemim količinama podataka izravno unutar preglednika. Ovo eliminira potrebu za ovisnošću o zamršenim pozadinskim sustavima, čime se pojednostavljuje proces razvoja i povećava učinkovitost. Rezultirajuće aplikacije sposobne su za izvođenje sofisticiranih funkcija s izuzetnom brzinom i preciznošću, nudeći korisnicima neusporedivu razinu angažmana i funkcionalnosti. Ova integracija označava značajan korak naprijed u području web razvoja, omogućujući stvaranje dinamičnih aplikacija vođenih podacima koje mogu odgovoriti na korisničke unose i promjene okruženja u stvarnom vremenu. Potencijal za inovacije u ovom prostoru je gotovo neograničen, dok programeri nastavljaju istraživati nove načine za iskorištavanje snage umjetne inteligencije u frontend razvoju, pomičući granice onoga što web aplikacije mogu postići.

Jedan značajan primjer ove vrste integracije je spajanje okvira React s TensorFlow.js, JavaScript bibliotekom dizajniranom za pokretanje modela strojnog učenja izravno unutar web preglednika. Koristeći ove tehnologije, razvojni programeri mogu kreirati aplikacije sposobne analizirati ponašanje korisnika u stvarnom vremenu za isporuku personaliziranih preporuka na temelju prikupljenih podataka. U području e-trgovine, takve aplikacije mogu pomno ispitati korisničke preferencije i preporučiti proizvode koji su najrelevantniji za pojedinca, čime se povećava angažman korisnika i povećava vjerojatnost konverzije (Facebook, 2017., 27). Slično, u

zdravstvenoj industriji ova se kombinacija može koristiti za analizu medicinskih podataka u stvarnom vremenu, olakšavajući precizniju dijagnostiku i razvoj personaliziranih planova liječenja.

Angular, naširoko korišteni frontend framework, postaje još moćniji kada se upari s AI bibliotekama poput ml5.js. ML5.js, koji je izgrađen na temelju TensorFlow.js, ima za cilj pojednostaviti ugradnju naprednih AI značajki u web aplikacije. Korištenjem Angulara uz ml5.js, programeri imaju mogućnost izrade aplikacija sposobnih za obavljanje zadataka kao što su prepoznavanje slika, analiza podataka i donošenje odluka u stvarnom vremenu izravno unutar preglednika. Ova metodologija smanjuje oslanjanje na pozadinsku obradu podataka, čime se pojednostavljuje razvojni proces i ubrzava isporuka aplikacije (Google, 2018., 35).

Ugradnja umjetne inteligencije u frontend aplikacije utire put stvaranju chatbota i virtualnih pomoćnika koji koriste obradu prirodnog jezika (NLP) za interakciju s korisnicima. Korištenjem JavaScript biblioteka dizajniranih za NLP, kao što su Natural i Compromise, programeri mogu izgraditi sofisticirane chatbotove sposobne razumjeti i odgovoriti na korisničke upite na razgovorni način. Ove napredne tehnologije omogućuju aplikacijama komunikaciju s korisnicima putem tekstualnih i glasovnih sučelja, nudeći odgovore, pomoć ili preporuke u stvarnom vremenu. Time se znatno smanjuje potreba za ručnim unosom podataka i poboljšava cjelokupno korisničko iskustvo (Klein, 2019., 52). Ovi inteligentni chatbotovi pronalaze aplikacije u raznim sektorima, od korisničke službe u e-trgovini do obrazovnih platformi koje pružaju prilagođene informacije i smjernice.

Štoviše, integracija frontend tehnologija s korisničkim sučeljima olakšava razvoj aplikacija proširene stvarnosti (AR) i virtualne stvarnosti (VR) koje iskorištavaju umjetnu inteligenciju za prepoznavanje objekata i isporuku interaktivnog sadržaja. Na primjer, programeri mogu upotrijebiti mješavinu Three.js za renderiranje 3D grafike i TensorFlow.js za implementaciju algoritama strojnog učenja za izradu AR aplikacija sposobnih za prepoznavanje fizičkih objekata u okruženju. Te aplikacije tada mogu trenutno ponuditi relevantne informacije ili interaktivna iskustva. Korištenjem ovih tehnologija, programeri mogu stvoriti impresivna i dinamična AR iskustva koja odgovaraju stvarnom svijetu u stvarnom vremenu, pružajući korisnicima

besprijekoran spoj digitalnih i fizičkih interakcija. Ova kombinacija ne samo da poboljšava korisničko iskustvo, već također otvara nove mogućnosti za inovacije u raznim područjima kao što su obrazovanje, zabava i maloprodaja. Integracija umjetne inteligencije s frontend tehnologijama ima potencijal za značajno poboljšanje performansi web aplikacija kroz pametne tehnike optimizacije. Korištenjem algoritama strojnog učenja moguće je dubinski analizirati podatke o ponašanju korisnika i u skladu s tim prilagoditi sadržaj stranice i resurse. To može dovesti do smanjenog vremena učitavanja i općenito optimiziranog korisničkog iskustva. Na primjer, sustavi dinamičkog učitavanja sadržaja vođeni umjetnom inteligencijom mogu predvidjeti stranice ili odjeljke koje će korisnici najvjerojatnije sljedeće posjetiti i proaktivno unaprijed učitati te resurse.

Na kraju, integracija frontend tehnologija s umjetnom inteligencijom osnažuje razvoj web aplikacija koje nadilaze puko reagiranje na radnje korisnika; te aplikacije također mogu dinamički prilagoditi svoj sadržaj i značajke na temelju specifičnih potreba korisnika. Ova metodologija nudi značajne prednosti u širokom rasponu sektora, kao što su e-trgovina, zdravstvo, obrazovanje i zabava. Prilagođavajući korisničko iskustvo da bude personaliziranije, učinkovitije i ispunjavajuće, podiže ukupnu interakciju, čineći je privlačnijom i zadovoljavajućom za krajnjeg korisnika.

## **5. PRIMJER UPOTREBE/PRIMJENE UMJETNE INTELIGENCIJE U FRONTEND PROGRAMIRANJU WEBA NA PRIMJERU CHATGPT I COPILOT**

Umjetna inteligencija (AI) evoluirala je u ključnu komponentu suvremenih tehnologija, duboko transformirajući razvoj web aplikacija. Uz stalno rastuća očekivanja korisnika za interaktivnim, personaliziranim i učinkovitim iskustvima, uključivanje umjetne inteligencije u frontend programiranje postalo je dominantan trend u razvoju web aplikacija. Među alatima koji ilustriraju mogućnosti umjetne inteligencije u programiranju sučelja su ChatGPT i Copilot. Ovi alati iskorištavaju napredne AI modele kako bi poboljšali korisnička iskustva i razvojne tijekove rada, olakšavajući stvaranje visoko sofisticiranih i inteligentnih web aplikacija.

ChatGPT, kreacija OpenAI-ja, služi kao moćan alat za generiranje prirodnog jezika, olakšavajući razvoj interaktivnih chatbota i virtualnih pomoćnika. Ovi digitalni entiteti sposobni su razumjeti korisničke upite i odgovoriti na njih na način koji vrlo nalikuje ljudskom razgovoru. Uključivanjem mogućnosti napredne obrade prirodnog jezika (NLP) u web aplikacije, ChatGPT igra ključnu ulogu u poboljšanju korisničkog iskustva. To postiže omogućavanjem personaliziranih i dinamičnih interakcija, čime cijeli angažman čini besprijeornijim i intuitivnijim.

Zajednički razvijen od strane GitHuba i OpenAI-ja, Copilot predstavlja revolucionarni resurs za programere. Ovaj inovativni alat koristi napredni model umjetne inteligencije za automatsku proizvodnju isječaka koda iz opisa funkcionalnosti. Posljedično, značajno ubrzava razvojni proces uz minimaliziranje vjerojatnosti pogrešaka. Kada se integrira u radne tijekove programiranja sučelja, Copilot olakšava brzu implementaciju zamršenih značajki, nudeći programerima neprocjenjivu pomoć u izradi čistog, učinkovitog koda. Ova poboljšana sposobnost ne samo da pojednostavljuje zadatke kodiranja, već također osigurava više standarde kvalitete koda i



dosljednosti, što ga čini nezamjenjivim sredstvom u modernom razvoju softvera.

Ovo poglavlje prikazuje primjere korištenja ChatGPT-a i Copilota u domeni frontend web razvoja, pokazujući kako ovi vrhunski alati mogu poboljšati radni tijek razvoja i poboljšati korisničko iskustvo iskorištavanjem sofisticiranih tehnologija umjetne inteligencije. Kroz ispitivanje konkretnih primjera iz, ovo poglavlje nastoji pokazati potencijal ovih alata vođenih umjetnom inteligencijom u različitim područjima web programiranja. Dodatno, ima za cilj pružiti programerima praktične smjernice za besprijekornu integraciju ovih tehnologija u njihove rutinske razvojne zadatke, čime se optimizira i učinkovitost i djelotvornost u stvaranju dinamičnih web aplikacija prilagođenih korisniku.

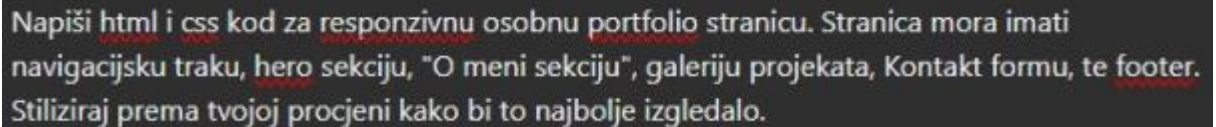
U nastavku prikazuju se primjeri umjetne inteligencije u frontend razvoju usredotočujući se na stvaranje stranice osobnog portfelja. Takva stranica služi kao ključna platforma za isticanje vještina, projekata i iskustava pojedinca. Obično ga koriste stručnjaci u tehnološkoj industriji kako bi predstavili svoj rad budućim poslodavcima ili klijentima. U prikazu osobnog portfolia koristit će se dva najsuvremenija AI alata ChatGPT i Copilot kako bi se procijenila i usporedila njihova izvedba i mogućnosti u kontekstu razvoja aspekata sučelja ove stranice portfelja. Ova usporedba pružit će uvid u to kako ovi alati umjetne inteligencije mogu poboljšati i pojednostaviti proces izgradnje sofisticirane i vizualno privlačne stranice osobnog portfelja. Izvođenje koda odvija se u Replit alatu.

## 5.1. Chatgpt

Početni zadatak bio je usredotočen na stvaranje HTML i CSS koda za dizajn responzivne web stranice osobnog portfelja. Ovaj dio treba sadržavati niz bitnih komponenti, kao što su navigacijska traka, hero sekcija, odjeljak "O meni", galerija projekata, obrazac za kontakt i podnožje (Slika 6).

Ovaj zadatak uključuje davanje uputa alatu za stvaranje temeljnog izgleda i elemenata dizajna web stranice koristeći HTML i CSS. Primarni naglasak je na osiguravanju responzivnosti web stranice, što znači da bi se trebala moći

neprimjetno prilagoditi različitim dimenzijama zaslona, uključujući i mobilne uređaje i stolna računala. Cilj je osigurati da stranica ne samo da izgleda privlačno, već i funkcionira optimalno na nizu uređaja, pružajući dosljedno i korisniku prilagođeno iskustvo bez obzira na veličinu zaslona. To podrazumijeva izradu dizajna koji je fleksibilan i prilagodljiv, osiguravajući da sadržaj ostane dostupan i vizualno kohezivan bilo da se gleda na pametnom telefonu, tabletu ili većem stolnom monitoru.



Napiši html i css kod za respozivnu osobnu portfolio stranicu. Stranica mora imati navigacijsku traku, hero sekciju, "O meni sekciju", galeriju projekata, Kontakt formu, te footer. Stiliziraj prema tvojoj procjeni kako bi to najbolje izgledalo.

**Slika 6. Prikaz zadavanja naredbi u programu ChatGPT**

*Izvor: samostalna izrada autora*

Nakon što je ChatGPT izgenerirao čitav kod, isti je kopiran u replit i pokrenut te je rezultat kao na slici 7:



**Slika 7. Dobiveni rezultat nakon unošenja koda**

*Izvor: samostalna izrada autora*

Nadalje, na slici 8. prikazuje se jasnu navigacijsku traku koja se sastoji od imena, te sa desne strane nalaze se navigacijski linkovi koji vode na početnu stranicu, o meni sekciju, sekciju s projektima, te na poslijetku i kontakt formu.

Također, bijeli prostor je zapravo hero sekcija, koja je velikih dimenzija i ne sadrži ništa.



**Slika 8. Prikaz navigacijske trake**

*Izvor: samostalna izrada autora*

Ispod hero sekcije nalaze se ostale sekcije osobne stranice koje su zatražene ChatGPT je izgenerirao veoma osnovni kod, bez puno stiliziranja. Dobivena web stranica ima naslov „Osobna Portfolio Stranica“ Svi navigacijski linkovi su ispravni i stranica je respozivna, osim što kada je browser smanjen na rezoluciju mobilnog ekrana, navigacijski linkovi nestaju iz navigacijske trake.



**Slika 9. Ostale sekcije osobne stranice**

*Izvor: samostalna izrada autora*

Za slijedeći prompt zadana je ChatGPT – u naredba za boju i font, kako bi se unijela boja ovom bijelom ekranu. Također, prosljeđuje mu se nasumično generirano ime osobe, preuzete sa alata „Behind the Name“, nasumično generiranih interesa sa „Perchance.org“ kako bi se oblikovala stranica prema osobi.

Moje ime je Ivan Matić, volim igrat igrice, odlazak u muzej i piti kavu. Želim da stiliziraš stranicu sa bojama i fontom koji se slažu sa mojom osobnošću.

**Slika 10. Prikaz nove naredbe ChatGPT-u**

*Izvor: samostalna izrada autora*

Nakon postavljenog upita, ChatGPT je promijenio boju navigacijske trake, footer-a, i tipke pošalji u tamno plavo, pozadina je svjetlo siva i vidimo bijeli tekst u hero sekciji koji objašnjuje posjetitelju na kojoj stranici se nalazi.

Također odabrao je fontove „Lora“ i „Roboto“ radi njihove čitljivosti (Slika 11).



**Slika 11. Prikaz promijenjene boje navigacijske trake te ostalih dodataka**

*Izvor: samostalna izrada autora*

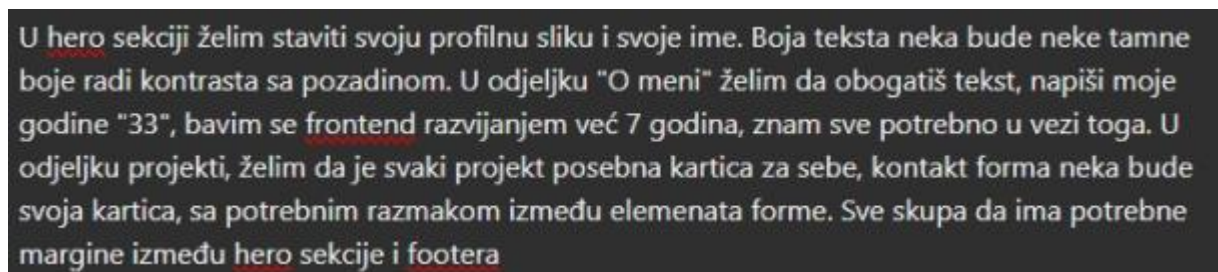
U odjeljku „O meni“ nadodan je dio upita, ali je ostavljen početni tekst (Slika 12).



**Slika 12. Prikaz dodanog djela upita i ostavljenog početnog teksta**

*Izvor: samostalna izrada autora*

U nastavku stiliziranja osobne stranice izmijeniti će se margine i padding elemenata u svim sekcijama, promijeniti će se projekte u individualne kartice, Kontakt formu će se smjestiti u vlastitu karticu, te nadopuniti „O meni“ tekst sa tekstom koji će sam ChatGPT izgenerirati na temelju nekoliko činjenica koje ću mu napomenuti, te nadodati sliku u hero odjeljku.



**Slika 13. Prikaz novih naredbi za ChatGPT**

*Izvor: samostalna izrada autora*

Nakon ovog upita, odjeljak O meni je i dalje bio kratak i sadrži samo navedene informacije.



**Slika 14. Prikaz ažuriranog odjeljka "O meni"**

*Izvor: samostalna izrada autora*

Budući da navedeni odjeljak „O meni“, odnosno prikazani tekst nije zadovoljavao informacije koje se trebaju nalaziti, otvoren je novi prozor chatGPT-a i dan mu je slijedeći prompt (Slika 15):



**Slika 15. Prikaz nove izmijenjene naredbe ChatGPT-u**

*Izvor: samostalna izrada autora*

Sada kada je postavljen tekst koji nam odgovara, isti se unosi u ostatak koda koji je generiran sa prijašnjim upitom. Također, za profilnu sliku preuzeta je adobe stock fotografija čovjeka u odijelu sa naočalama.

Prema izgeneriranom kodu ChatGPT-a vidi se kako su profilna fotografija i tekst koji se nalazi ispod nje premali za veliki hero odjeljak, te je slika loše rezolucije (Slika 16)



**Slika 16. Prikaz dobivenog rezultata**

*Izvor: samostalna izrada autora*

Nadalje, u nastavku se vidi kako se sada projekti nalaze svaki u svojoj kartici, te kontakt forma je unutar svoje kartice. Također vidljive su margine između sekcija i između kontakt forme i footera (Slika 17).



**Slika 17. Prikaz projekata (svaki u svojoj kartici)**

*Izvor: samostalna izrada autora*

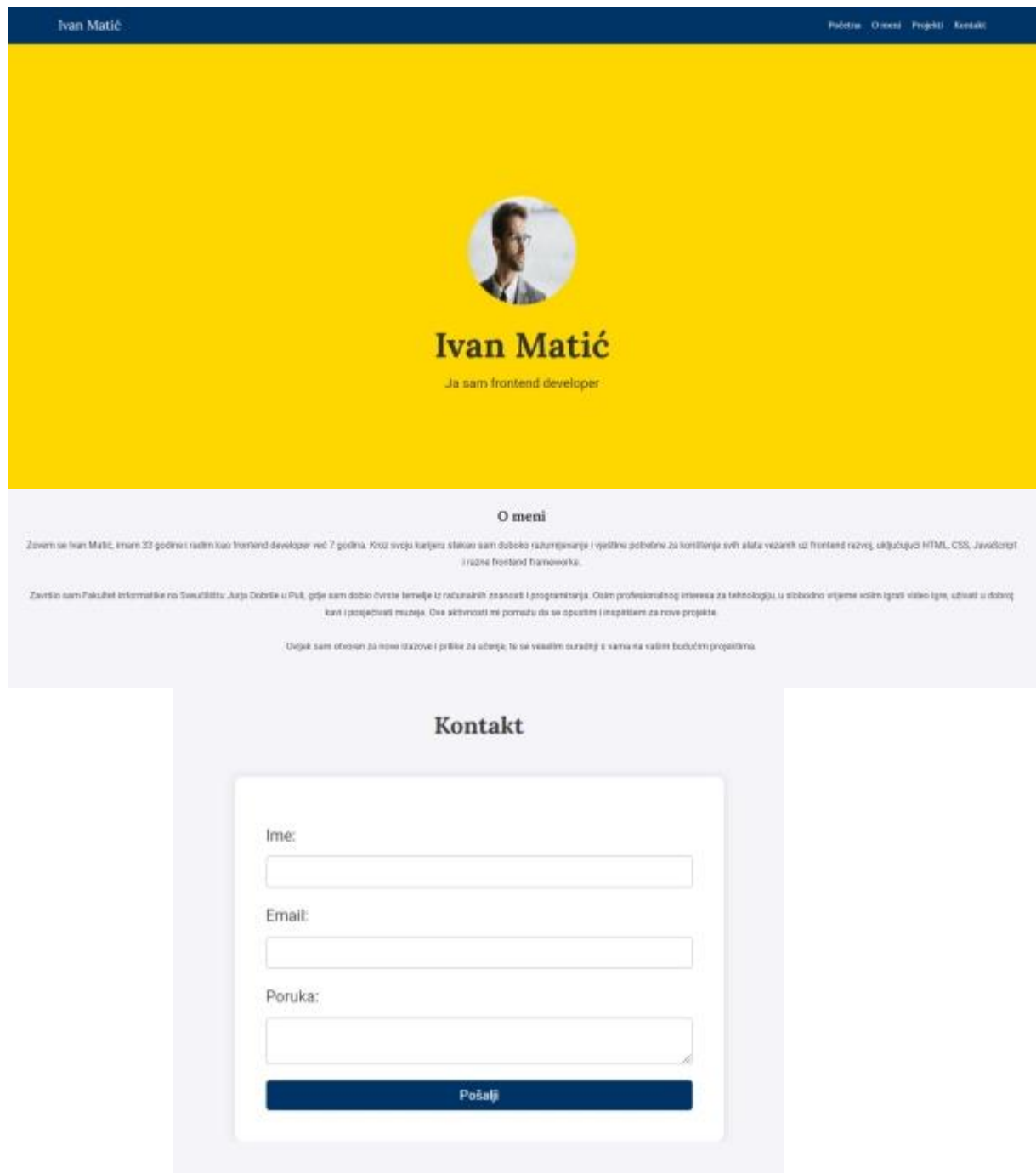
Zatim, u sljedećem promptu nastojat će se povećati sliku profila zajedno sa tekстом, dodati hero sekciji pozadinsku boju i pomaknuti naslove sekcija u sredinu. Urediti će se opisni tekst i smanjiti dimenzije kontakt forme (Slika 18).

Želim da hero sekcija ima komplementarnu pozadinsku boju, želim da su profilna slika i tekst unutar hero sekcije veći. Želim da su svi naslovi sekcija u sredini. Želim da kontakt forma bude manje široka. Želim tekst u odjeljku "O meni" da se nalazi u sredini i da je podijeljen u nove redove prikladno kako bi se izdužio odjeljak.

Slika 18. Prikaz nove naredbe

Izvor: samostalna izrada autora

Nakon ovog upita dobiven je rezultat prikazan na slici 19:



Slika 19. Prikaz rezultata zadane naredbe

Izvor: samostalna izrada autora



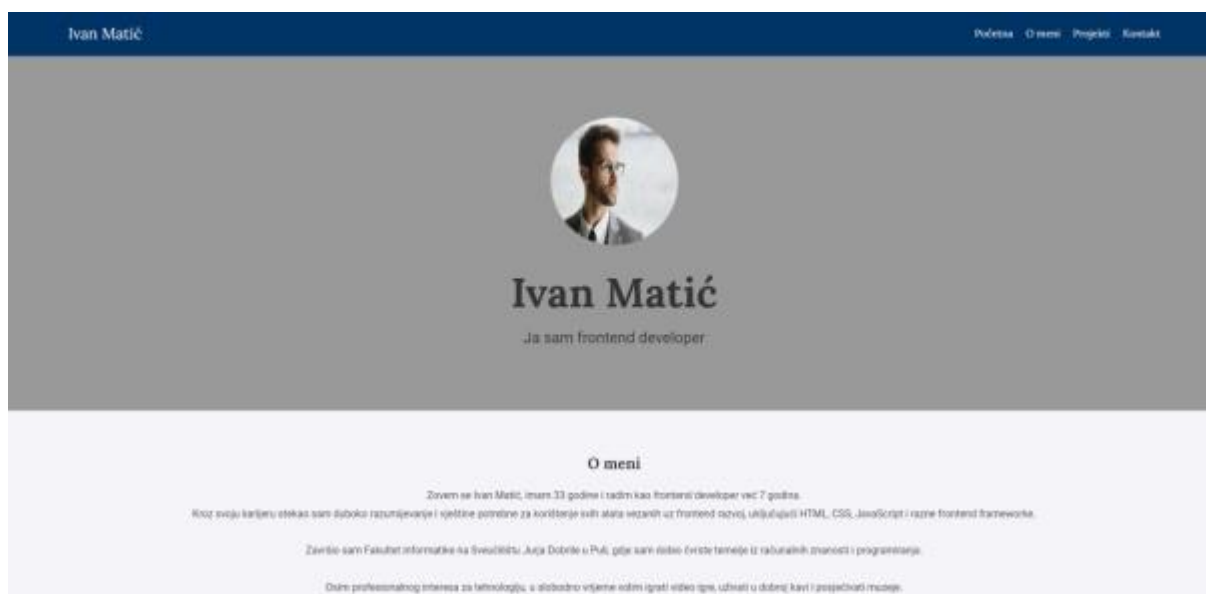
Chat GPT se odlučio staviti žutu boju koja je komplementarna tamno plavoj, profilna fotografija i tekst su malo veći, ali i dalje nedovoljno za toliki odjeljak. U sekciji „O meni“ tekst počinje od sredine ali se rasteže previše u širinu što poništava željeni efekt. Kontakt forma je sužena i izgleda puno modernije. U slijedećem upitu smanjiti će se hero sekciju, poravnati tekst da bude uži, u kartice projekata će se definirati odjeljke za fotografiju projekta i njegov opis. Također dodati će im se suptilnu animaciju kada posjetitelj drži ikonu miša iznad kartice. U kontakt formu će se staviti naslove input polja u sredinu i urediti tipku pošalji (Slika 20).

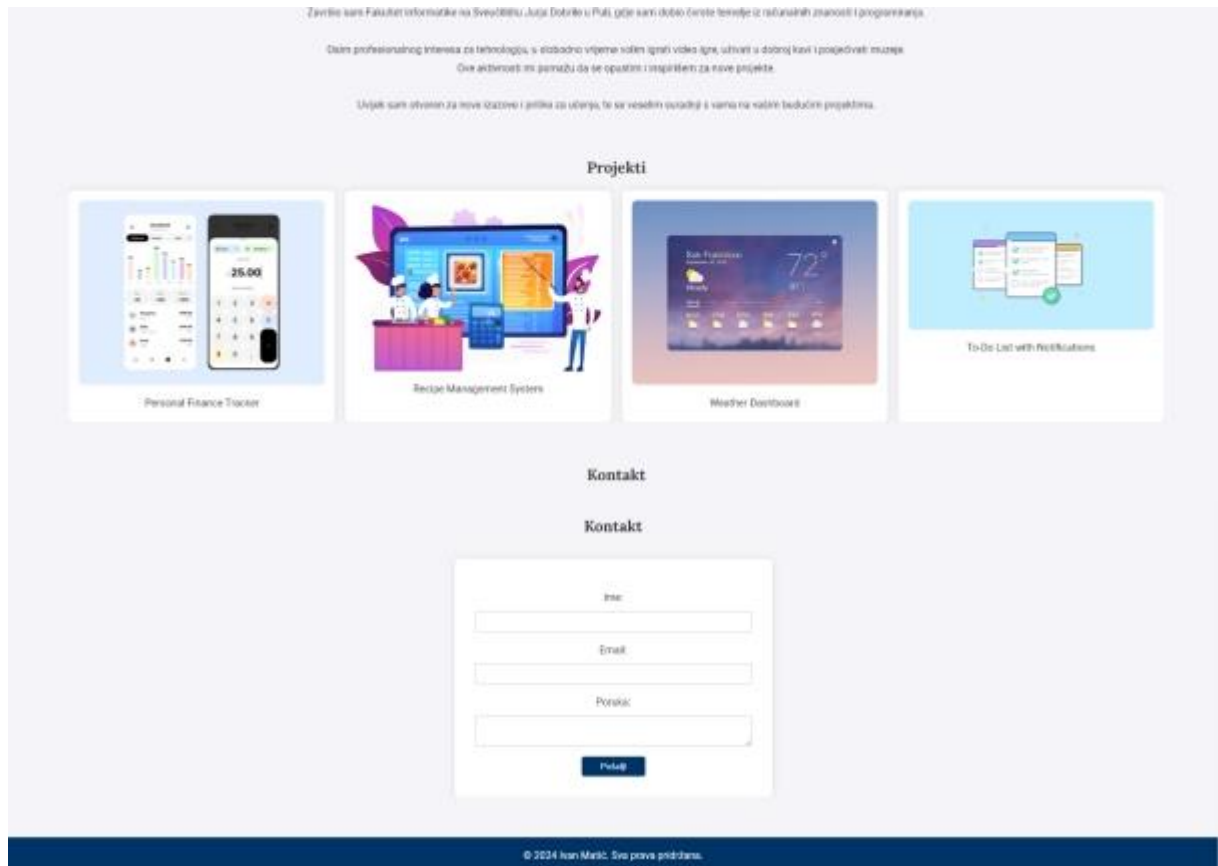
Želim da smanjiš veličinu hero sekcije, želim da tekst u odjeljku "O meni" bude više prelomljen i obostrano poravnat. U odjeljku s projektima želim da se u svakoj kartici nalazi place holder za sliku projekta, te ispod slike kratki opis samog projekta. Također želim da mouse hover na nekoj od kartica projekta ima animaciju. U kontakt formi postavi naslove input polja u sredinu, smanji širinu tipke pošalji i malo je povećaj po x osi

#### Slika 20. Prikaz nove naredbe

Izvor: samostalna izrada autora

Nakon ovog upita, dodane su slike i imena projektima i promijenjena je pozadinska boja u sivu, te stranica izgleda ka na slici 21:





**Slika 21. Prikaz potpunog frontenda za web stranicu**

*Izvor: samostalna izrada autora*

Sada je u potpunosti izrađen frontend za web stranicu. Nakon samo 5 upita upućenih ChatGPT-u uspješno je izgenerirana kvalitetna moderna osobna odnosno portfolio stranica. Uz dodatan upit chatgpt-u za tekst u odjeljku „O meni“, ukupan broj upita za ovu Portfolio web stranicu je 6, što je veoma učinkovito. Korištena je verzija Chat Gpt 4-0.

## 5.2. Copilot

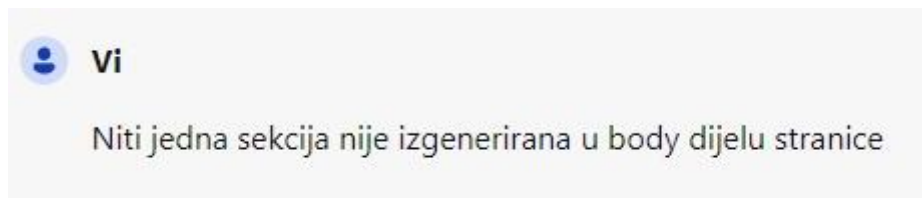
Kod primjera Copilota isto se koristila izrada a portfolio stranice, aprilikom korištenja započeto je sa istim upitom (Slika 22).

Napiši html i css kod za **responzivnu osobnu portfolio stranicu**. Stranica mora imati **navigacijsku traku**, **hero sekciju**, **„O meni sekciju“**, **galeriju projekata**, **Kontakt formu**, te **footer**. **Stiliziraj** prema **tojjoj procjeni** kako bi to **najbolje izgledalo**.

**Slika 22. Prikaz naredbe u Copilotu**

*Izvor: samostalna izrada autora*

Rezultat ovog upita bio je html kod koji nema nikakav sadržaj u body elementu stranice, što je rezultiralo samo bijelim blank ekranom. Za CSS dio koda Copilot je izgenerirao stilove za navedene sekcije, no nije ih napravio u kosturu stranice. Postavljen je dodatni upit (Slika 23):



**Slika 23. Dodatni upit programu Copilot**

*Izvor: samostalna izrada autora*

Nakon dodatnog upita portfolio stranica sada izgleda ovako (Slika 24):



**Slika 24. Izgled portfolio stranice nakon dodatnog upita**

*Izvor: samostalna izrada autora*

Nakon dodatnog upita i napravljene portfolio stranice, vidljivo je da navigacijski linkovi nisu funkcionalni, sekcije nisu nikako odvojene međusobno, te cijela stranica je kraća nego rezolucija standardnog monitora, što znači da postoji bijeli blok između kraja footera i kraja prozora pretraživača. Prateći upite koji su bili zadani ChatGPT-u slijedeći upit odnositi će se na personaliziranje same web stranice prema nasumično generiranoj osobi od ranije (Slika 25).

Moje ime je Ivan Matić, volim igrat igrice, odlazak u muzej i piti kavu. Želim da stiliziraš stranicu sa bojama i fontom koji se slažu sa mojom osobnošću.

**Slika 25. Naredba za personaliziranje web stranice**

*Izvor: samostalna izrada autora*

Budući da rezultat upita nije bio dobar jer je generirani kod je samo izmijenio boju navigacijske trake i footera u indigo, slična boji koju je i chat gpt odabrao, odlučeno je za dodavanje sljedećeg upita odmah nakon (Slika 26):

U hero sekciji želim staviti svoju profilnu sliku i svoje ime. Boja teksta neka bude tamne boje. U odjeljku "O meni" želim da piše "Zovem se Ivan Matić, imam 33 godine i radim kao frontend developer već 7 godina. Kroz svoju karijeru stekao sam duboko razumijevanje i vještine potrebne za korištenje svih alata vezanih uz frontend razvoj, uključujući HTML, CSS, JavaScript i razne frontend frameworke.

Završio sam Fakultet informatike na Sveučilištu Jurja Dobrile u Puli, gdje sam dobio čvrste temelje iz računalnih znanosti i programiranja. Osim profesionalnog interesa za tehnologiju, u slobodno vrijeme volim igrati video igre, uživati u dobroj kavi i posjećivati muzeje. Ove aktivnosti mi pomažu da se opustim i inspirišem za nove projekte.

Uvijek sam otvoren za nove izazove i prilike za učenje, te se veselim suradnji s vama na vašim budućim projektima.". U odjeljku projekti želim da je svaki projekt posebna kartica za sebe, kontakt forma neka bude svoja kartica, sa potrebnim razmakom između elemenata forme.

**Slika 26. Dodavanje upita u postojeći kod**

*Izvor: samostalna izrada autora*

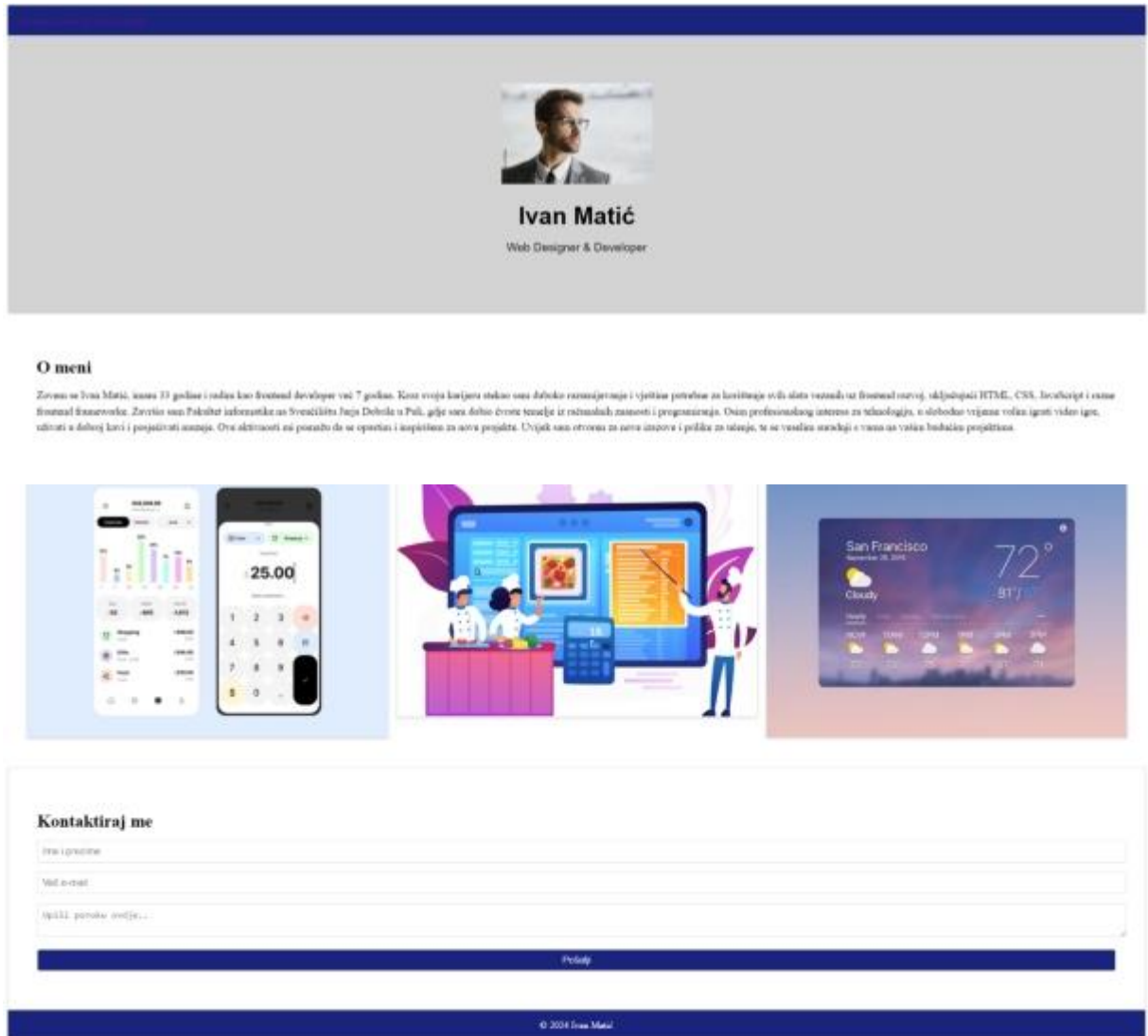
Rezultat ova dva upita je i dalje veoma loš, stranica je počela poprimati svoju formu, no i dalje copilot jednostavno ne generira potreban kod za html (Slika 27):



**Slika 27. Prikaz rezultata nakon dodavanja sljedećeg upita**

*Izvor: samostalna izrada autora*

Zbog Copilotovog limita na broj upita dnevno, duži i kvalitetniji rad na ovoj web stranici biti će puno složeniji od rada sa ChatGPT-om. Nakon ručnoga prevođenja stranice na hrvatski jezik, dodavanja profilne i fotografija projekata, te upisivanje teksta u odjeljak „O meni“ stranica sada izgleda ovako (Slika 28):



Slika 28. Izgled stranice nakon ručnog dodavanja potrebnih stavki

Izvor: samostalna izrada autora

Nakon još dodatnog samostalnog kodiranja, mijenjanja profilne fotografije iz kvadrata u oblik kruga, smanjivanje širine kontakt forme, te dodatnih upita prema Copilotu:

```
zelim skliku projekta i naslov projekata spakirati u jednu karticu, te glavni naslov h2 da bude iznad sekcije
u sredini " <h2>Projekti</h2>
<div class="portfolio">
  <div class="project">
    
    <h3>Naslov Projekta 1</h3>
  </div>
  <div class="project">
    
    <h3>Naslov Projekta 3</h3>
  </div>
  <div class="project">
    
    <h3>Naslov Projekta 4</h3>
  </div>
</div>
"
```

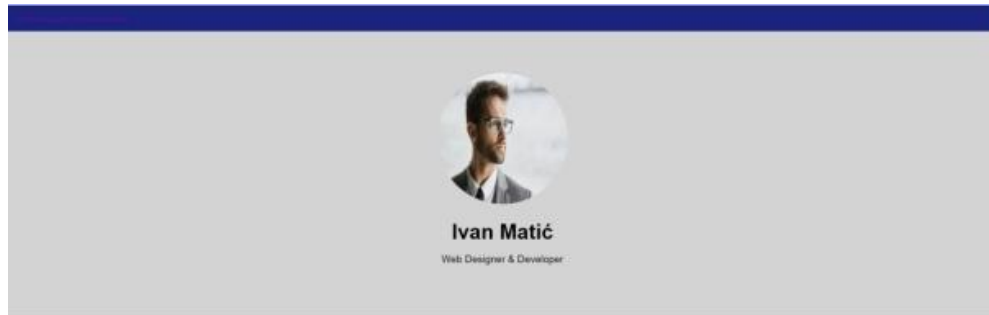
 Vi

želio bih hover effect animaciju na karticama projekata, i da je i naslov projekta unutar kartice

### Slika 29. Prikaz ručno dodanih naredbi

*Izvor: samostalna izrada autora*

Stranica sada izgleda ovako (Slika 30):



#### O meni

Zovem se Ivan Matić, imam 25 godina i radim kao front-end developer već 7 godina. Kao svja korisnik odise sam duboko razumijevanje i vještine potrebne za korištenje svih alat vernaik uz front-end razvoj, uključujući HTML, CSS, JavaScript i druge front-end tehnologije. Završio sam Fakultet Informatike na Sveučilištu Jure Dobrila u Puli, gdje sam dobio izvrsne rezultate iz računalnih znanosti i programiranja. Osim profesionalnog interesa za tehnologiju, u slobodno vrijeme volim igrati video igre, odmoriti u dobroj kafi i posjetiti muzeje. Ove aktivnosti su pomoću da se opustim i inspiriram za nove projekte. Uvijek sam otvoren za nove izazove i prilike za učenje, te se rado bavim svojim projektima.

#### Projekti



**Kontaktiraj me**

Ime i prezime

Mail e-mail

Slika 30. Trenutni izgled stranice

Izvor: samostalna izrada autora

Profilna slika sada je u obliku kruga, vidi se kako se pri držanju ikone miša na kartici projekta prikazuje naslov projekta, te pop-up animacija. Nakon još nekoliko upita (Slika 31):

```

Vi
zelim da uklonim link decoration, tekst da bude svijetle boje, i linkovi da su pomaknuti u desnu stranu, a sa lijeve strane navbara da bude moje ime "nav {
  background-color: #1a237e;
  /* Tamnija nijansa plave */
  color: #fff;
  padding: 1rem;
} " <nav>
  <a href="#home"> Početna </a>
  <a href="#about"> O meni </a>
  <a href="#portfolio"> Projekti </a>
  <a href="#contact"> Kontakt </a>
</nav>

```



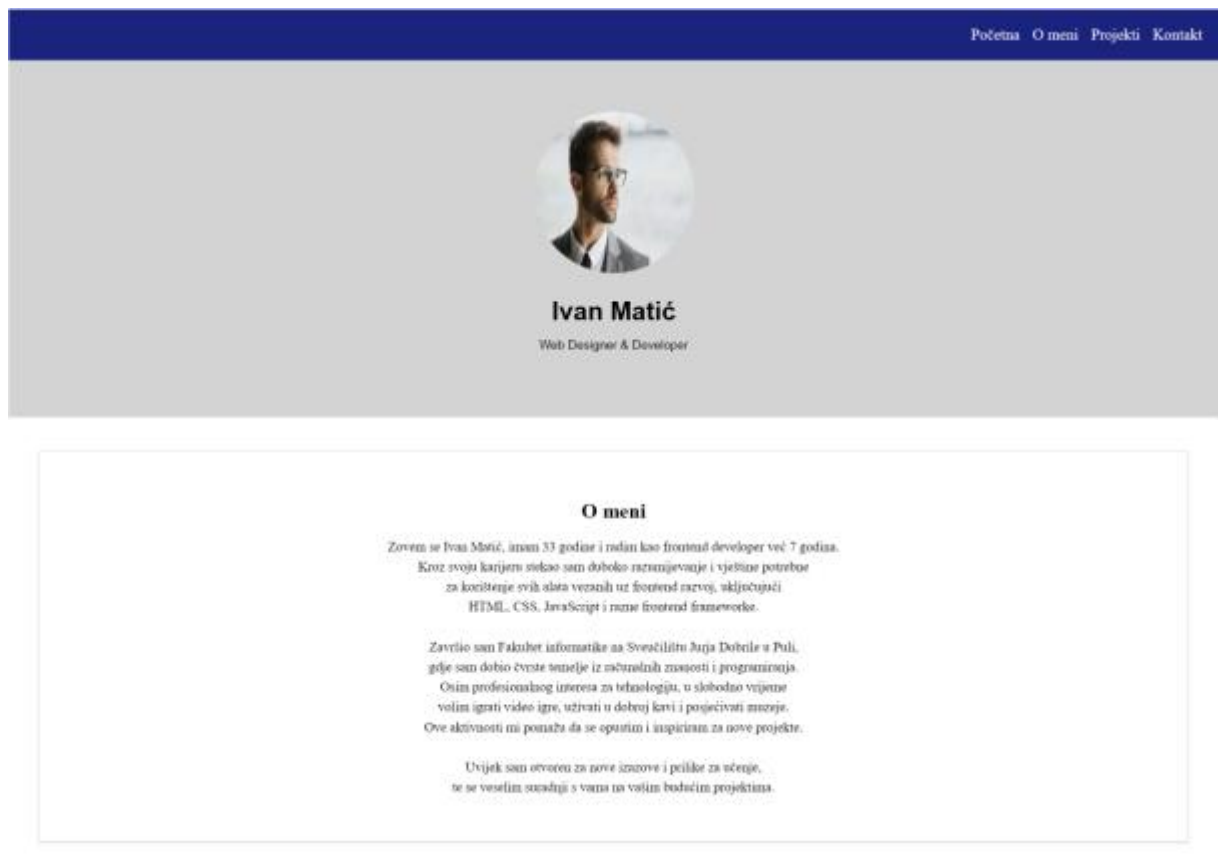
 Vi

```
zelim da span bude s lijeve strane navbara a linkovi sa desne " <nav>
<span>Ivan Matić</span>
<a href="#home">Početna</a>
<a href="#about">O meni</a>
<a href="#portfolio">Projekti</a>
<a href="#contact">Kontakt</a>
</nav>"
```

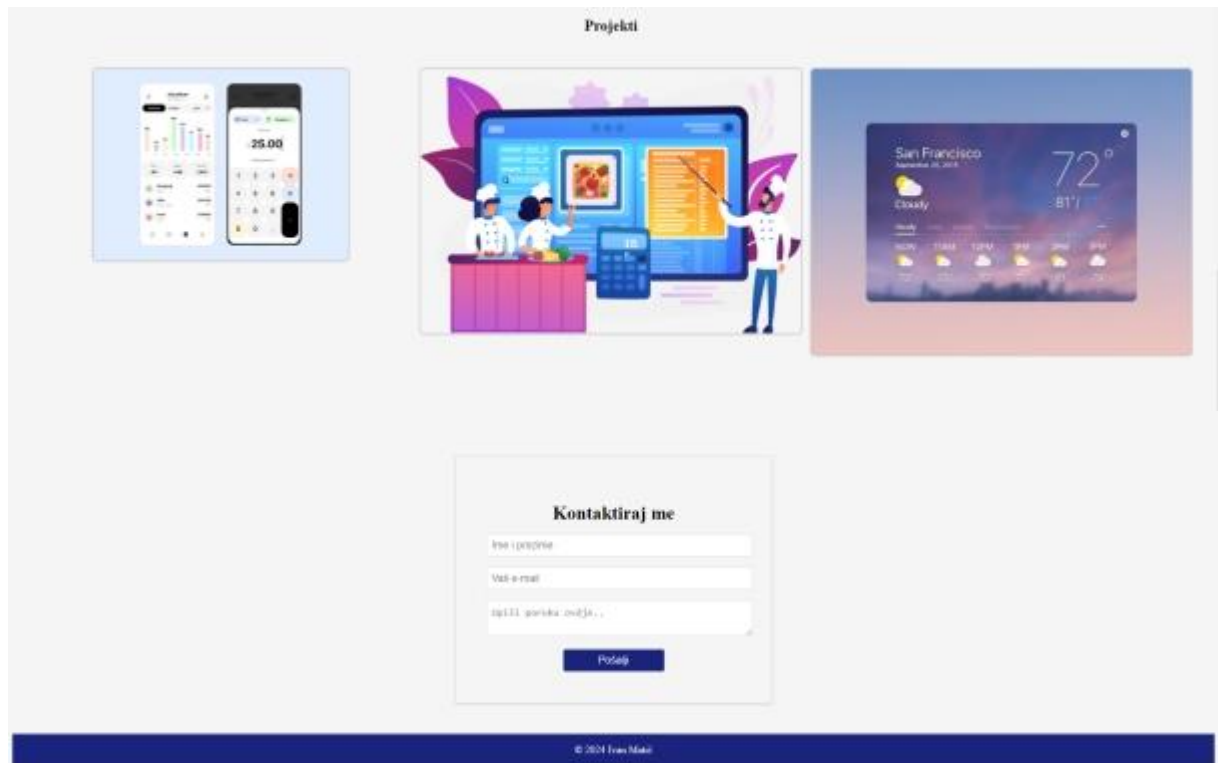
Slika 31. Dodatne naredbe programu

Izvor: samostalna izrada autora

Uz dodatno uređivanje teksta u sekciji O meni, dodavanje ruba sekcijama i stvaranja razmaka između kontakt forme i footera samostalno. Stranica sada izgleda ovako (Slika 32):





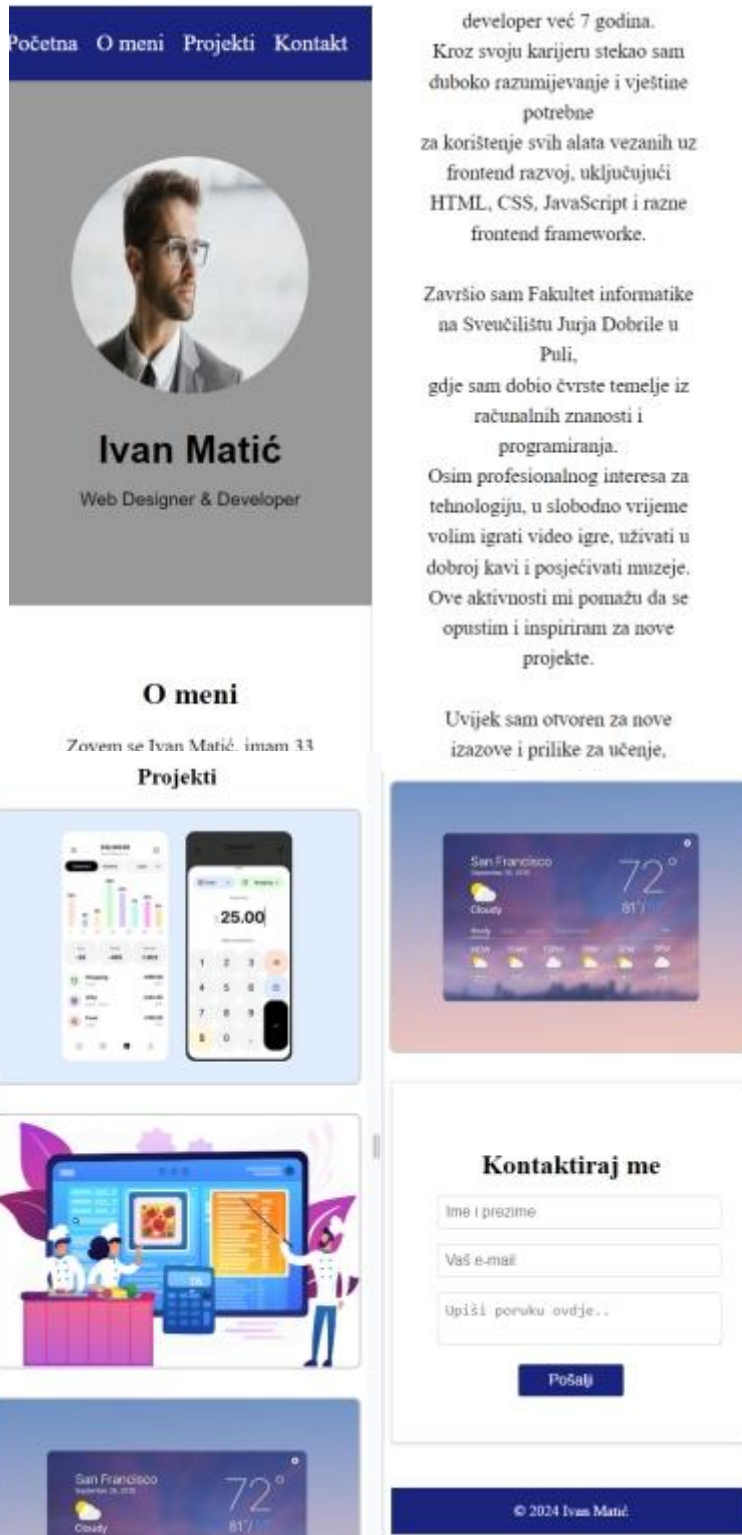


**Slika 32. Izgled stranice uz sve dosadašnje promijene**

*Izvor: samostalna izrada autora*

Svi navigacijski linkovi su funkcionalni, na klik vode na tu sekciju stranice, svi elementi su poravnati u sredini, stranica ima sve potrebne sekcije. Što se frontenda tiče može se reći da je stranica gotova.

Mobilna verzija (Slika 33):



**Slika 33. Prikaz mobilne verzije stranice**

*Izvor: samostalna izrada autora*

Nakon temeljite procjene oba alata umjetne inteligencije, ChatGPT i Microsoft Copilot, za pomoć u dizajnu i kodiranju prednjeg dijela web stranice s osobnim portfeljem, postaje očito da postoje značajne razlike u njihovim funkcionalnostima i metodologijama. Svaki alat nudi različite prednosti i ograničenja, čineći njihove jedinstvene pristupe razvojnom procesu prilično očitima. Unatoč tome što su obojica sposobni pomoći u izradi web stranice, razlike u njihovim sposobnostima i načinima rada su izrazito uočljive.

ChatGPT se pokazao kao iznimno moćan alat, sposoban izgraditi gotovo cijelu web stranicu uz minimalno vodstvo i ograničeno moje sudjelovanje. Njegova vještina u generiranju punog HTML i CSS koda, zajedno s naprednim stilom koji nadilazi osnovne potrebe, čini ga vrlo učinkovitim za temeljit razvoj sučelja. Sa samo nekoliko upita, ChatGPT je uspio proizvesti funkcionalnu i estetski ugodnu stranicu, drastično smanjujući vrijeme i trud koji su obično potrebni za finaliziranje projekta.

Suprotno tome, Copilot je pokazao ograničenije mogućnosti u različitim aspektima. Iako služi kao koristan alat za pomoć pri stvaranju određenih segmenata koda, njegova ograničenja postaju očita kada se dobije zadatak samostalnog razvoja cijele aplikacije. Tipično, Copilot generira samo rudimentarni okvir koda, zahtijevajući značajan dodatni napor od razvojnog programera kako bi se projekt razradio. Nadalje, ograničenja broja upita i ograničenja broja znakova po upitu čine interakciju s Copilotom nezgrapnom. Programeri se često nalaze u situaciji da ponavljaju upite i vrše brojne ručne prilagodbe, što u konačnici otežava učinkovitost tijeka razvoja.

Jedna od primarnih razlika između ova dva alata leži u njihovoj ukupnoj učinkovitosti. ChatGPT je pokazao izvanrednu sposobnost izvršavanja zadataka s minimalnim dodatnim uputama i prilagodbama, olakšavajući brzo dovršavanje potpuno razvijene stranice. U samo šest upita, ChatGPT je uspio učinkovito konstruirati cijelu stranicu. Nasuprot tome, Copilot je iziskivao veći broj ponavljanja i ispravaka kako bi se postigao usporediv rezultat, koji još uvijek nije dostigao kvalitetu koju pruža ChatGPT.

Osim toga, Copilot se susreće sa značajnim poteškoćama u očuvanju konteksta tijekom proširenih interakcija. Tijekom mog korištenja često sam se zatekao u potrebi da opetovano ponavljam upite ili razjašnjavam upute. Ovaj problem proizlazi iz ograničenja količine informacija s kojima Copilot može istovremeno rukovati. Kao rezultat toga, moj tijek rada često je postajao nepovezan i fragmentiran. Nasuprot tome, kada sam koristio ChatGPT, primijetio sam izrazito poboljšanje u zadržavanju konteksta, što je pridonijelo kohezivnijem i besprijekornijem iskustvu tijekom rada.

Ukratko, iako Copilot služi kao vrijedan pomoćnik za rješavanje specifičnih zadataka unutar kodiranja, ChatGPT se pokazao kao znatno napredniji i svestraniji alat za sveobuhvatan razvoj sučelja. Sposobnost ChatGPT-a da autonomno generira cijelu web stranicu s vrlo malo korisničkih unosa pozicionira ga kao puno učinkovitije i snažnije rješenje. Nasuprot tome, Copilot, sa svojim ograničenim funkcionalnostima, zahtijeva znatno više intervencija i prilagodbi od programera, što ga čini manje praktičnim za opsežne potrebe programiranja sučelja.

### **5.3. Upotreba ChatGpt i Copilota u kontekstu frontend programiranja weba**

Upotreba alata poput ChatGPT-a i Microsoft Copilota u frontend programiranju weba postala je sve češća, s ciljem povećanja produktivnosti i optimizacije koda. Dok su specifični statistički podaci o korištenju ovih alata unutar frontend programiranja ograničeni, pa gotovo da ih i nema, postoje određeni podaci koji mogu pružiti širu sliku njihove popularnosti i primjene.

Prema podacima iz 2023., ChatGPT je imao preko 100 milijuna aktivnih korisnika mjesečno, od kojih su mnogi programeri, uključujući one koji rade u frontend razvoju. ChatGPT se koristi za širok spektar zadataka u frontend programiranju, kao što su automatsko generiranje koda, otklanjanje grešaka, optimizacija JavaScript, HTML i CSS koda te pružanje objašnjenja i savjeta o

najboljim praksama u razvoju korisničkih sučelja (UI). Iako specifični podaci za frontend razvoj nisu dostupni, popularnost ovog alata među programerima pokazuje njegovu vrijednost u tom kontekstu.

Microsoft Copilot, integriran s alatima kao što su Visual Studio Code i GitHub, također je snažno prisutan u zajednici frontend programera. Prema izvještajima Microsofta, više od milijun korisnika koristi Copilot, uključujući 37.000 organizacija, za razvoj softvera, što uključuje i frontend razvoj. Copilot koristi umjetnu inteligenciju za automatsko generiranje i dovršavanje koda te pomaže programerima u svakodnevnim zadacima kao što su kreiranje korisničkih sučelja, optimizacija performansi web stranica i uklanjanje grešaka. Copilot je poznat po tome što značajno ubrzava repetitivne zadatke, poput pisanja HTML struktura i optimizacije stilova za različite preglednike, omogućujući programerima da se usredotoče na kreativne aspekte razvoja.

Iako ne postoje javno dostupni statistički podaci koji precizno kvantificiraju upotrebu ovih alata u kontekstu isključivo frontend razvoja, opći podaci o njihovoj popularnosti i korištenju među programerima jasno pokazuju da su oba alata postala ključni resursi za programere. Microsoft Copilot, na primjer, pokazao je da može poboljšati produktivnost programera za 27% prema Microsoftovim internim podacima. Ova povećanja produktivnosti često su ključna u kontekstu frontend razvoja, gdje je brzina iteracija i optimizacija korisničkih iskustava od izuzetne važnosti (Kaftan et. al., 2024).

## **6. BUDUĆI TRENDOWI RAZVOJA UMJETNE INTELIGENCIJE I FRONTEND PROGRAMIRANJA**

Kako tehnologija umjetne inteligencije nastavlja napredovati velikom brzinom, očekuje se da će ugradnja korisničkih sučelja (UI) u frontend programiranje doseći nove razine sofisticiranosti. Ova će evolucija biti potaknuta nekoliko ključnih trendova koji su spremni značajno utjecati na budući krajolik ove domene.

Jedan od najznačajnijih napredaka u novije vrijeme je uspon Edge AI tehnologija, koje uključuju pokretanje algoritama strojnog učenja izravno na vlastitim uređajima korisnika umjesto oslanjanja na udaljene poslužitelje. Ova strategija smanjuje potrebu za trajnom komunikacijom s udaljenim poslužiteljima, čime se poboljšava privatnost korisnika i smanjuje kašnjenje. Koristeći Edge AI, aplikacije mogu reagirati brže, što je ključni čimbenik u situacijama koje zahtijevaju hitne odluke, poput onih koje uključuju autonomna vozila, pametne naprave i sustave proširene stvarnosti (Liu, 2021., 89). Na primjer, umjesto da prenosi korisničke podatke na udaljene poslužitelje za obradu, Edge AI olakšava lokalnu analizu i donošenje odluka na samom uređaju, čime se povećava sigurnost i štiti privatnost.

Generativni modeli, kao što je GPT-3, prave valove u području frontend programiranja, označavajući značajan napredak u tom području. Ovi modeli koriste sofisticirane tehnike strojnog učenja za izradu dinamičnog i personaliziranog sadržaja prilagođenog interakcijama korisnika. Korištenjem generativnih modela programeri mogu stvarati aplikacije sposobne automatski proizvesti tekst, slike, pa čak i kod. Na primjer, GPT-3 se može integrirati u web aplikacije za generiranje prirodnog jezika, omogućujući korisnicima da rade s aplikacijom kao da komuniciraju s ljudskim agentom (Brown, 2020., 121). Potencijalne primjene ove tehnologije su ogromne i obuhvaćaju

industrije poput e-trgovine, marketinga i obrazovanja, gdje pružanje personaliziranog sadržaja može značajno obogatiti korisničko iskustvo.

Još jedan značajan trend je demokratizacija umjetne inteligencije, koja alate i resurse za rad s korisničkim sučeljima (UI) čini dostupnima širem rasponu programera, čak i onima s ograničenom stručnošću u strojnom učenju. Na primjer, platforme poput AutoML-a olakšavaju automatizaciju izgradnje, obuke i implementacije modela strojnog učenja. Ova automatizacija znatno smanjuje zahtjeve za naprednim tehničkim vještinama (Google, 2020., 45). Posljedično, sve veći broj programera sučelja sada može ugraditi elemente korisničkog sučelja u svoje aplikacije bez potrebe za dubljim razumijevanjem složenih algoritama. Koristeći ove alate, programeri mogu koristiti unaprijed obučene modele ili modificirati postojeće kako bi zadovoljili specifične zahtjeve svojih projekata. Ovo ne samo da ubrzava proces razvoja, već i smanjuje troškove povezane s implementacijom UI rješenja.

Osim ovih značajnih trendova, očekuje se da će frontend programiranje doživjeti značajan napredak kroz kontinuiranu evoluciju pametnih pomoćnika i chatbota koje pokreće obrada prirodnog jezika (NLP). Ovi inovativni alati već su pronašli opsežne primjene u različitim sektorima, ali kako tehnološki napredak napreduje, oni će postati sve rafiniraniji i vještiji u pružanju prilagođenih odgovora i usluga. Uključivanje NLP-a u razvoj sučelja omogućuje stvaranje aplikacija koje čine više od pukog razumijevanja upita korisnika; također mogu predvidjeti zahtjeve korisnika i ponuditi relevantne informacije ili preporuke. Ova integracija NLP-a osnažit će aplikacije za pružanje interaktivnijeg i intuitivnijeg korisničkog iskustva predviđanjem potreba i isporukom kontekstualno relevantnog sadržaja. Kako ove tehnologije budu sazrijevale, možemo očekivati da će frontend sučelja postati dinamičnija i usmjerena na korisnika, što će u konačnici poboljšati način na koji korisnici komuniciraju s digitalnim platformama.

Uzimajući u obzir ove stalne trendove, očito je da je umjetna inteligencija spremna ostati ključni element u evoluciji frontend aplikacija. Kako tehnološki napredak traje, korisnička sučelja postat će bitna komponenta web aplikacija, opremajući programere sredstvima za dizajniranje inteligentnijih, osjetljivijih i zadivljujućih korisničkih iskustava. Integracija sofisticiranih modela strojnog učenja, demokratizacija razvojnih alata i sve veća upotreba tehnologija Edge

AI pokazuju da budućnost frontend razvoja ima ogroman potencijal za inovacije i značajna poboljšanja korisničkog iskustva.

Integracija umjetne inteligencije (AI) u frontend razvoj nudi brojne prednosti, ali također uvodi niz izazova kojima je potrebno posvetiti posebnu pozornost. Jedna od najznačajnijih prepreka je optimizacija performansi. Modeli umjetne inteligencije, osobito oni koji koriste strojno učenje, često zahtijevaju znatnu računalnu snagu, što može utjecati na brzinu i odziv web aplikacija. Ovaj problem je posebno problematičan na uređajima s ograničenim mogućnostima, kao što su pametni telefoni ili starija računala. Rješavanje ovog izazova zahtijeva korištenje optimiziranih modela strojnog učenja posebno dizajniranih za frontend operacije. Na primjer, korištenje smanjenih ili "laganih" modela i korištenje tehnika kompresije modela može znatno smanjiti zahtjeve za resursima, što rezultira bržim izvršenjem i poboljšanom reakcijom aplikacije (Liu, 2021., 102).

Osiguravanje privatnosti i sigurnosti korisnika predstavlja značajan izazov pri integraciji umjetne inteligencije (AI) u frontend aplikacije. Modeli korisničkog sučelja (UI) često obrađuju osjetljive korisničke podatke, što dovodi do niza problema vezanih uz sigurnost i zaštitu osobnih podataka. Kako se propisi o zaštiti podataka nastavljaju pooštrevati, kao što je Opća uredba o zaštiti podataka (GDPR) koja se primjenjuje u Europskoj uniji, usklađenost s ovim pravnim standardima postaje imperativ za web aplikacije koje koriste UI modele. To podrazumijeva implementaciju praksi kao što je minimiziranje prikupljanja podataka samo na ono što je strogo neophodno, šifriranje osjetljivih informacija radi zaštite od neovlaštenog pristupa i održavanje transparentnosti u vezi s praksama korištenja podataka. Ovi su koraci ključni za očuvanje povjerenja korisnika i pridržavanje pravnih zahtjeva (Europska unija, 2018., 78).

Značajan izazov koji se mora riješiti je osiguranje nepristranosti i poštenja unutar sustava korisničkog sučelja (UI). Modeli korisničkog sučelja obično uče iz opsežnih skupova podataka, što može rezultirati održavanjem već postojećih pristranosti ugrađenih u podatke. Ovo je pitanje posebno kritično u aplikacijama u kojima odluke imaju značajne posljedice za korisnike, kao što su sustavi preporuka ili platforme za donošenje odluka o zapošljavanju. U tim



je kontekstima ključno zajamčiti da sustavi umjetne inteligencije daju objektivne i pravedne rezultate. Postizanje ovog cilja zahtijeva kontinuirano praćenje i evaluaciju izvedbe modela, uz usvajanje strategija usmjerenih na ublažavanje pristranosti tijekom faze obuke modela (Barocas, Hardt i Narayanan, 2019., 66).

Unatoč ovim preprekama, izgledi za frontend razvoj koji uključuje umjetnu inteligenciju iznimno su svijetli. Kako tehnologija napreduje, korisnička su sučelja spremna postati bitna komponenta suvremenih web aplikacija. Korisnicima će ponuditi prilagođenije, interaktivnije i inteligentnije iskustvo. Ova će sučelja osnažiti web aplikacije da predviđaju potrebe korisnika, predviđaju njihove preferencije i automatiziraju svakodnevne zadatke, čime se značajno poboljšava cjelokupno korisničko iskustvo.

Stalni napredak alata i metodologija za ugradnju korisničkih sučelja (UI) u sučelje bit će od ključne važnosti za napredovanje. Kako se pojavljuju novi alati koji olakšavaju besprijekornu integraciju umjetne inteligencije (AI) kao što je AutoML i razne druge platforme za automatizaciju strojnog učenja programerima će postati sve izvedivije ugraditi AI u svoje aplikacije bez potrebe za opsežnom stručnošću u strojnom učenju. Ova demokratizacija tehnologije je spremna učiniti implementaciju korisničkog sučelja pristupačnijom i učinkovitijom u raznim područjima. Posljedično, ovo će potaknuti inovacije i ubrzati napredak unutar web razvoja, otključavajući mnoštvo novih prilika (Google, 2020., 45).

S brzim napretkom tehnologija kao što je Edge AI, koja omogućuje modelima umjetne inteligencije da rade izravno na korisničkim uređajima, aplikacije korisničkog sučelja spremne su postati učinkovitije i sigurnije. Ova inovacija omogućuje bržu obradu podataka i smanjuje ovisnost o uslugama u oblaku. Kada se ovi tehnološki pomoci udruže sa stalnim naporima za rješavanje problema vezanih uz izvedbu, privatnost i pristranost, otvorit će put za sljedeći val web aplikacija, čineći ih pametnijima, bržima i sigurnijima nego ikad prije. Posljedično, ugradnja umjetne inteligencije u frontend razvoj spremna je revolucionirati način na koji korisnici rade s web aplikacijama. Ova će integracija korisnicima ponuditi iskustva koja nisu samo personalizirana, već i visoko prilagodljiva. U isto vrijeme, opremit će programere robusnim alatima koji im omogućuju da osmisle inovativna rješenja sposobna odgovoriti na

zamršene izazove. Ova dvostruka korist naglašava transformativni potencijal umjetne inteligencije u preoblikovanju korisničkih iskustava i krajolika web razvoja. Kako AI nastavlja napredovati, možemo očekivati još sofisticiranije interakcije i revolucionarne aplikacije koje pomiču granice onoga što je trenutno moguće. Tekuća evolucija u ovom području ukazuje na budućnost u kojoj sinergija između umjetne inteligencije i frontend programiranja postaje kamen temeljac digitalnih iskustava, postavljajući nove standarde za angažman korisnika i genijalnost programera.

## 7. ZAKLJUČAK

Umjetna inteligencija (AI) brzo se pojavljuje kao ključni element u evoluciji web tehnologija, posebno u području frontend razvoja. Ovaj napredak olakšava stvaranje dinamičnih, interaktivnih i inteligentnih web aplikacija. U ovom radu provedo se istraživanje kako umjetna inteligencija može poboljšati frontend razvoj upotrebom specifičnih alata kao što su ChatGPT i Copilot. Ovi alati pokazuju kako ugradnja umjetne inteligencije može ne samo ubrzati i pojednostavniti razvojni proces, već i poboljšati cjelokupno korisničko iskustvo. Kako se AI nastavlja razvijati, njegov utjecaj na frontend programiranje postaje sve dublji, nudeći razvojnim programerima inovativne načine za izradu osjetljivijih i korisniku prilagođenijih aplikacija.

ChatGPT se pojavio kao iznimno moćan alat koji omogućuje korisnicima da samostalno izrade potpuno funkcionalne i estetski ugodne web stranice uz samo minimalno vodstvo. Njegova impresivna sposobnost generiranja kompletnog koda, zadržavanja konteksta i ponude vrijednih preporuka pokazuje kako korisnička sučelja mogu ublažiti znatan dio razvojnog opterećenja. Ovo značajno umanjuje potrebu za opsežnim ručnim kodiranjem i čestim intervencijama koje su obično potrebne u procesu web razvoja.

Nasuprot tome, Copilot nudi značajnu pomoć razvojnim programerima automatskim generiranjem isječaka koda, iako mu nedostaje sposobnost samostalnog razvoja cijele aplikacije. Unatoč tim nedostacima, Copilot ostaje vrijedan resurs za ubrzavanje procesa razvoja, osobito u scenarijima u kojima je programerima potrebna pomoć pri pisanju određenih segmenata koda ili poboljšanju performansi postojećeg koda.

Uključivanje korisničkih sučelja u razvoj sučelja nudi mnoštvo prednosti, ali također predstavlja i niz izazova. Ovi izazovi uključuju optimizaciju performansi sustava, zaštitu privatnosti korisnika i rješavanje pristranosti svojstvenih modelima strojnog učenja. Unatoč tome, izgledi ostaju optimistični. Kontinuirani napredak u alatima i metodologijama za integraciju korisničkog sučelja otvara nove puteve za inovacije i razvoj. Ovaj napredak će osnažiti programere da izgrade web aplikacije koje nisu samo sofisticiranije nego i sve inteligentnije. Evolucija ovih tehnologija obećava budućnost u kojoj

je potencijal za stvaranje naprednih digitalnih iskustava prilagođenih korisniku veći nego ikad.

Ukratko, umjetna inteligencija već revolucionira područje razvoja sučelja, a njezin će utjecaj još više rasti kako tehnologija napreduje. Učinkovitim korištenjem alata kao što su ChatGPT i Copilot, programeri mogu značajno povećati svoju produktivnost, skratiti razvojne cikluse i poboljšati cjelokupno korisničko iskustvo. Uključivanje umjetne inteligencije u frontend programiranje daleko je od prolaznog trenda; predstavlja vitalnu komponentu budućeg krajolika web razvoja. Kako napredujemo, sinergija između AI i razvoja sučelja obećava otključavanje novih razina učinkovitosti, kreativnosti i inovacija, postavljajući pozornicu za dinamičnije i osjetljivije digitalno okruženje. Ova poboljšanja ne odnose se samo na ubrzavanje procesa razvoja, već i na podizanje kvalitete web aplikacija, čineći ih intuitivnijim, pristupačnijim i usmjerenijim na korisnika. Stoga prihvaćanje alata i metodologija vođenih umjetnom inteligencijom nije samo opcija, već i nužnost za one koji žele ostati na čelu web razvoja. Kako se AI tehnologija nastavlja razvijati, možemo očekivati još sofisticiranije alate i značajke koje će dodatno pojednostaviti tijekove rada i otvoriti nove mogućnosti za stvaranje iznimnih web iskustava.

## LITERATURA:

1. Angular (2024) Overview of Angular libraries, dostupno na: <https://angular.dev/tools/libraries> (13.08.2024).
2. Babel. (2019). Babel Documentation. Dostupno na: <https://babeljs.io/docs/en/> (13.08.2024).
3. Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine Learning*. MIT Press.
4. Bierman, G., Abadi, M., & Torgersen, M. (2014). *Understanding TypeScript*. ACM SIGPLAN Notices, 49(3), 1-13.
5. Bostock, M. (2016). *D3.js Documentation*. Dostupno na: <https://d3js.org/> (13.08.2024).
6. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). *Language Models are Few-Shot Learners*. In *Advances in Neural Information Processing Systems* (pp. 1877-1901).
7. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805.
8. Duckett, J. (2011). *HTML & CSS: Design and Build Websites*. John Wiley & Sons.
9. European Union. (2018). *General Data Protection Regulation (GDPR)*. Official Journal of the European Union.
10. Facebook. (2017). *React Documentation*. Dostupno na: <https://reactjs.org/docs/getting-started.html> (06.08.2024).
11. Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation, University of California, Irvine).
12. Garrett, J. J. (2010). *The Elements of User Experience: User-Centered Design for the Web and Beyond*. New Riders.
13. Google. (2018). *Angular Documentation*. Dostupno na: <https://angular.io/docs> (06.08.2024).
14. Google. (2020). *AutoML Documentation*. Dostupno na: <https://cloud.google.com/automl/docs> (13.08.2024).

15. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
16. Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. *Neural Computation*, 9(8), 1735-1780.
17. Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing*. Pearson.
18. Klein, J. (2019). *Compromise.js Documentation*. Dostupno na: <https://compromise.cool/> (13.08.2024).
19. Kaftan, A.N., Hussain, M.K. & Naser, F.H. Response accuracy of ChatGPT 3.5 Copilot and Gemini in interpreting biochemical laboratory data a pilot study. *Sci Rep* 14, 8233 (2024). <https://doi.org/10.1038/s41598-024-58964-1>.
20. Krug, S. (2014). *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders.
21. Liu, Y. (2021). *Edge AI: Vision, Challenges, and Future Directions*. In *Proceedings of the IEEE*.
22. Marcotte, E. (2010). *Responsive Web Design*. A Book Apart.
23. McCarthy, J. (2007). *What is artificial intelligence?* Dostupno na: <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html> (06.08.2024).
24. Meyer, E. A. (2015). *CSS: The Definitive Guide*. O'Reilly Media.
25. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv preprint arXiv:1301.3781.
26. React (2024). Dostupno na: <https://react.dev/> (10.08.2024).
27. Reardon, A. (2020). *Brain.js Documentation*. Dostupno na: <https://brain.js.org/> (13.08.2024).
28. Runway. (2020). *ml5.js Documentation*. Dostupno na: <https://ml5js.org/> (13.08.2024).
29. Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.
30. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... & Hassabis, D. (2016). *Mastering the Game of Go*

- with Deep Neural Networks and Tree Search. Nature, 529(7587), 484-489.*
31. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks*. In *Advances in Neural Information Processing Systems* (pp. 3104-3112).
  32. TensorFlow. (2019). *TensorFlow.js Documentation*. Dostupno na: <https://www.tensorflow.org/js> (13.08.2024).
  33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is All You Need*. In *Advances in Neural Information Processing Systems* (pp. 5998-6008).
  34. W3C. (2018). *HTML & CSS Standards*. World Wide Web Consortium.
  35. Wroblewski, L. (2009). *Web Form Design: Filling in the Blanks*. Rosenfeld Media.
  36. You, E. (2016). *Vue.js Documentation*. Dostupno na: <https://vuejs.org/v2/guide/> (13.08.2024).

## POPIS SLIKA

Slika 1. Primjer programskoga jezika HTML.....	16
Slika 2. Prikaz CSS programskog jezika .....	17
Slika 3. Prikaz instalacije React biblioteke .....	20
Slika 4. Prikaz html u Angularu .....	21
Slika 5. Prikaz koda izgradnje modela sa tensorflow.js .....	24
Slika 6. Prikaz zadavanja naredbi u programu ChatGPT .....	30
Slika 7. Dobiveni rezultat nakon unošenja koda .....	30
Slika 8. Prikaz navigacijske trake .....	31
Slika 9. Ostale sekcije osobne stranice .....	31
Slika 10. Prikaz nove naredbe ChatGPT-u .....	32
Slika 11. Prikaz promijenjene boje navigacijske trake te ostalih dodataka.....	32
Slika 12. Prikaz dodanog djela upita i ostavljenog početnog teksta .....	33
Slika 13. Prikaz novih naredbi za ChatGPT .....	33
Slika 14. Prikaz ažuriranog odjeljka "O meni" .....	33
Slika 15. Prikaz nove izmijenjene naredbe ChatGPT-u.....	34
Slika 16. Prikaz dobivenog rezultata .....	35
Slika 17. Prikaz projekata (svaki u svojoj kartici) .....	35
Slika 18. Prikaz nove naredbe .....	36
Slika 19. Prikaz rezultata zadane naredbe .....	36
Slika 20. Prikaz nove naredbe .....	37
Slika 21. Prikaz potpunog frontenda za web stranicu .....	38
Slika 22. Prikaz naredbe u Copilotu .....	38
Slika 23. Dodatni upit programu Copilot.....	39
Slika 24. Izgled portfolio stranice nakon dodatnog upita .....	39
Slika 25. Naredba za personaliziranje web stranice.....	40
Slika 26. Dodavanje upita u postojeći kod .....	40
Slika 27. Prikaz rezultata nakon dodavanja sljedećeg upita .....	40
Slika 28. Izgled stranice nakon ručnog dodavanja potrebnih stavki .....	41
Slika 29. Prikaz ručno dodanih naredbi .....	42
Slika 30. Trenutni izgled stranice.....	43
Slika 31. Dodatne naredbe programu .....	44
Slika 32. Izgled stranice uz sve dosadašnje promijene .....	45
Slika 33. Prikaz mobilne verzije stranice .....	46



## SAŽETAK

Ovaj rad istražuje ulogu umjetne inteligencije (UI) u frontend programiranju weba, s naglaskom na integraciju naprednih alata poput ChatGPT-a i Copilota u proces razvoja web aplikacija. Analizom prednosti i izazova ovih alata, rad pokazuje kako UI može značajno unaprijediti korisničko iskustvo i ubrzati razvoj frontend aplikacija. ChatGPT se ističe kao alat sposoban za gotovo samostalno kreiranje cjelovitih web stranica, dok je Copilot korisniji kao pomoć pri pisanju dijelova koda. U radu su također analizirani budući trendovi razvoja UI tehnologija, uključujući Edge AI i generativne modele, te su istaknuti izazovi poput optimizacije performansi i osiguravanja privatnosti korisnika. Kroz praktične primjere korištenja alata, rad zaključuje da će UI igrati sve značajniju ulogu u budućnosti frontend programiranja, omogućujući programerima da kreiraju inteligentnije i učinkovitije web aplikacije.

**KLJUČNE RIJEČI:** umjetna inteligencija, frontend programiranje, ChatGPT, Copilot, web aplikacije

## SUMMARY

This paper explores the role of artificial intelligence (AI) in frontend web programming, with an emphasis on integrating advanced tools like ChatGPT and Copilot into the web application development process. By analyzing the advantages and challenges of these tools, the paper shows how AI can significantly improve the user experience and accelerate the development of frontend applications. ChatGPT stands out as a tool capable of almost independently creating complete websites, while Copilot is more useful as an aid when writing parts of the code. The paper also analyzed future trends in the development of AI technologies, including Edge AI and generative models, and highlighted challenges such as optimizing performance and ensuring user privacy. Through practical examples of using the tool, the paper concludes that AI will play an increasingly important role in the future of frontend programming, enabling developers to create more intelligent and efficient web applications.

**KEYWORDS:** artificial intelligence, frontend programming, ChatGPT, Copilot, web applications