Web aplikacija sa sustavom višekriterijskog odlučivanja i stvaranjem preporuka za pomoć pri kupnji automobila

Berišić, Emanuel

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:018150

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni prijediplomski studij Računarstvo

WEB APLIKACIJA SA SUSTAVOM VIŠEKRITERIJSKOG ODLUČIVANJA I STVARANJEM PREPORUKA ZA POMOĆ PRI KUPNJI AUTOMOBILA

Završni rad

Emanuel Berišić

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju

Ocjena završnog rada na sveučilišnom prijediplomskom studiju

Ime i prezime pristupnika:	Emanuel Berišić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4614, 27.07.2021.
JMBAG:	0165089319
Mentor:	prof. dr. sc. Goran Martinović
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Web aplikacija sa sustavom višekriterijskog odlučivanja i stvaranjem preporuka za pomoć pri kupnji automobila
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada:	U teorijskom dijelu završnog rada treba proučiti i opisati postupak izbora prikladnog automobila usklađenog s ekološkim standardima i potrebama korisnika. Uz analizu stanja u području i postojećih sličnih rješenja, potrebno je predložiti parametre, kriterije i preporuke određene ekološkim standardima u automobilima, različitim vrstama pogona, potrebama i navikama korisnika uključujući i stil vožnje, infrastrukturom punionica, te cijenom i drugim značajkama. Nadalje, treba definirati prikladne postupke višekriterijskog odlučivanja i stvaranje preporuka koji će kupcu pomoći pri izboru odgovarajućeg automobila. Na temelju

Datum prijedloga ocjene završnog rada od strane mentora:	16.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	27.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	29.09.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O IZVORNOSTI RADA

Osijek, 29.09.2024.

Ime i prezime Pristupnika:	Emanuel Berišić
Studij:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R4614, 27.07.2021.
Turnitin podudaranje [%]:	11

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija sa sustavom višekriterijskog odlučivanja i stvaranjem preporuka za pomoć pri kupnji automobila**

izrađen pod vodstvom mentora prof. dr. sc. Goran Martinović

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD 1
1.1 Zadatak završnog rada 2
2. IZAZOVI, STANJE TE POSTOJEĆA SLIČNA RJEŠENJA U PODRUČJU PREPORUKA PRI KUPOVINI AUTOMOBILA 3
2.1 Pretraživanja automobila na webu 3
2.2. Izazovi pri odabiru automobila 3
2.3 Analiza prodaje i stanje web pretraživanja automobila 4
2.4. Pregled stanja u području 5
2.5 Postojeća rješenja 5
2.5.1 Carwow 5
2.5.2 Chiriboo 6
2.5.3 Cinch 7
2.5.4 Parkers 8
3. MODEL APLIKACIJE ZA VIŠEKRITERIJSKO ODLUČIVANJE PREPORUKE PRI KUPOVINI AUTOMOBILA 10
3.1. Funkcionalni zahtjevi na web aplikaciju 10
3.1.1 Autentifikacija korisnika 10
3.1.2 Odjava korisnika 10
3.1.3 Preporuka automobila 10
3.1.4 Detalji o vozilu 10
3.2 Nefunkcionalni zahtjevi na web aplikaciju 11
3.3 Višekriterijsko odlučivanje 11
4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE 13
4.1. Korištene tehnologije i alati 13
4.1.1. HTML 13
4.1.2. SCSS 13
4.1.3. Bootstrap 13
4.1.4. JavaScript 13
4.1.5 React 14
4.1.6. Redux 14
4.1.7. Visual Studio Code 14
4.1.8. Firebase 14

4.2 Prikaz programskog rješenja 15
4.2.1 Baza podataka 15
4.2.2 Registracija korisnika 16
4.2.3 Prijava i odjava korisnika 17
4.2.4 Upitnik za korisnika 18
4.2.5 Stvaranje preporuke automobila 20
4.2.6 Prikaz automobila 21
4.2.7. Dekorativni dijelovi aplikacije 22
5. PRIKAZ NAČINA RADA I ISPITIVANJE WEB APLIKACIJE 23
5.1. Način korištenja aplikacije 23
5.2. Ispitivanje rada aplikacije 26
5.2.1 Popunjavanje upitnika 26
5.2.2 Preporučeni automobili 29
5.3. Analiza rezultata ispitivanja 30
6. ZAKLJUČAK 31
LITERATURA 32
SAŽETAK 34
ABSTRACT 35
ŽIVOTOPIS 36
PRILOZI 37

1. UVOD

Prilikom kupovine automobila, mnogi potrošači suočavaju se s izazovom odabira vozila koje najbolje odgovara njihovim potrebama i preferencijama. Razvojem dodatne opreme i novih oblika goriva i pogona, dodatno je otežan izbor automobila. Funkcionalni zahtjevi vozila, poput potrošnje goriva, sigurnosti te performansi, važni su kriteriji pri odabiru vozila, a dodatni izazov pri odabiru predstavlja i činjenica da automobili postaju simbolima identiteta i statusa. U današnje vrijeme, dostupnost informacija putem interneta potrošačima je omogućeno da istraže različite modele, usporede cijene i pročitaju recenzije, što značajno otežava odluku. Visoki troškovi održavanja, osiguranja te potencijalni problemi s pouzdanošću predstavljaju dodatne izazove s kojima se potrošači suočavaju.

Cilj ovog završnog rada je istražiti postojeća rješenja i razviti vlastitu web aplikaciju koja će korisnicima pružiti pomoć pri kupovini automobila koristeći sustav višekriterijskog odlučivanja. Aplikacija omogućava registraciju i prijavu korisnika te bit će korišten upitnik s pitanjima o automobilu kako bi se na temelju njihovih odgovora generirale personalizirane preporuke. Za razvoj aplikacije koristit će se JavaScript u kombinaciji s React.js i Redux bibliotekama, čime će se osigurati interaktivno korisničko sučelje. Firebase Cloud će biti korišten za bazu podataka, omogućujući brzu i sigurnu pohranu korisničkih podataka i odgovora. Proces višekriterijskog odlučivanja bit će implementiran pomoću posebnih algoritama koji će uzeti u obzir različite karakteristike automobila, poput cijene, potrošnje goriva i drugih preferencija korisnika. Aplikacija će biti testirana na različitim korisničkim slučajevima kako bi se osigurala točnost preporuka i optimizacija korisničkog iskustva, čime će se korisnicima znatno olakšati proces donošenja odluke o kupovini automobila.

U drugom poglavlju analizirani su mogući izazovi s kojima se mogu susresti potrošači, stanje u području i navedena moguća programska rješenja koja su dostupna potrošačima. Treće poglavlje opisuje funkcionalne i nefunkcionalne zahtjeve na web aplikaciju. U četvrtom poglavlju prikazano je programsko rješenje, a u petom poglavlju opisan je način korištenja aplikacije i analizu mogućih rezultata.

1.1 Zadatak završnog rada

U teorijskom dijelu završnog rada potrebno je prikazati i proučiti moguće probleme s kojima se susreću potrošači. Nadalje, treba definirati prikladne postupke višekriterijskog odlučivanja i stvaranje preporuka koje će kupcu pomoći pri izboru odgovarajućeg automobila. Uz analizu stanja u području i postojećih sličnih rješenja, potrebno je predložiti parametre, kriterije i preporuke određene ekološkim standardima u automobilima, različitim vrstama pogona, potrebama i navikama korisnika, uključujući stil vožnje, infrastrukturom punionica, te cijenom i drugim značajkama. Nadalje, treba definirati prikladne postupke višekriterijskog odlučivanja i stvaranje preporuka koje će kupcu pomoći pri izboru odgovarajućeg automobila. Na temelju navedenoga, treba definirati funkcionalne i nefunkcionalne zahtjeve, model, arhitekturu i dizajn web aplikacije na strani korisnika i poslužitelja. Web aplikacija treba omogućiti registriranje i prijavu korisnika, unos i pohranu značajki izbora automobila i stvaranje preporuka za kupnju s rangiranim prikazom aktualnih automobila. Web aplikacija treba biti ostvarena koristeći prikladne programske jezike, tehnologije i razvojne okvire te biti ispitana na odgovarajućim ulaznim podacima i slučajevima korištenja.

2. IZAZOVI, STANJE TE POSTOJEĆA SLIČNA RJEŠENJA U PODRUČJU PREPORUKA PRI KUPOVINI AUTOMOBILA

U ovom poglavlju bit će objašnjeni mogući izazovi koji ometaju kupce pri odabiru automobila, pružit će se uvid u tehnologije i postojeća rješenja web aplikacija koje pružaju usluge pri kupovini automobila.

2.1 Pretraživanja automobila na webu

Web pretraživanje automobila obuhvaća korištenje interneta za prikupljanje i analizu informacija o automobilima prije nego što se donese odluka o kupovini. Ovaj proces uključuje posjećivanje specijaliziranih web stranica, korištenje različitih alata za usporedbu vozila, čitanje recenzija korisnika i stručnjaka te pregledavanje multimedijalnog sadržaja poput fotografija i videozapisa vozila. Glavni je cilj omogućiti korisnicima da pronađu vozilo koje najbolje odgovara njihovim potrebama i preferencijama, uzimajući u obzir sve relevantne faktore poput cijene, performansi, sigurnosti i pouzdanosti. Prema istraživanju tvrtke Google [1], čak 92 % kupaca pretražuje vozila na društvenim mrežama prije kupnje, a čak 80% spremno je kupiti vozilo u potpunosti preko interneta, čijem je znatnom povećanju pridonijela pandemija uzrokovana COVID-19 virusom.

2.2. Izazovi pri odabiru automobila

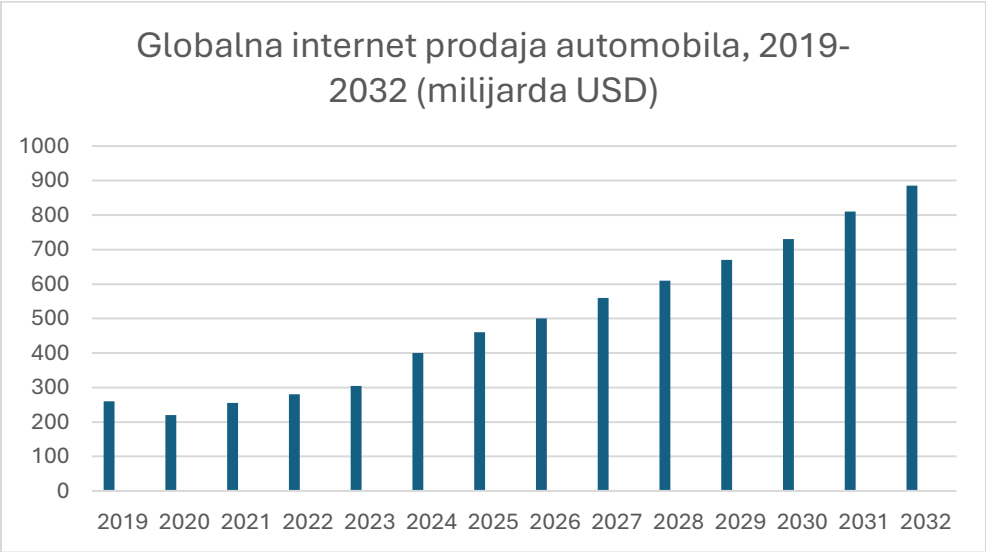
Jedan od glavnih izazova pri odabiru automobila je nedovoljna informiranost potrošača, kao i loš marketing auto salona. Potrošači često dobiju nejasnu sliku o automobilu te na temelju ugrubo prikazanih specifikacija automobila ne znaju jasno tumačiti prikazane informacije. Ponuda u obližnjim auto salonima također stvara problem. Nedostatak opreme ili modela mogu naštetiti percepciji potrošača jer je svaki potrošač unikatan, odnosno, svatko traži nešto drugo u automobilu. Prema izvoru [2], sve više salona ulaže u prostor koji omogućuje bolje prikazivanje automobila samom korisniku ne bi li tako uklonili probleme Internet pretraživanja. U taj prostor postavljaju visokokvalitetnu rasvjetu koja se predstavlja boljom od sunčevog svjetla te kamere s visokom rezolucijom i mogućim 360° videozapisom. Time kupac dobiva čistu sliku automobila te uvid u moguća oštećenja i nedostatke. Neki saloni nude besplatnu dostavu na adresu kupca što uvelike poboljšava potrošačko iskustvo. Problem kupcima predstavljaju i web stranice s manjkom informacija koje često mogu prikazivati i neistinite (nerealne) tehničke detalje o automobilu, kao

što su potrošnja i ubrzanje. Neki web portali mogu biti zastarjeli ili neprilagođeni za mobilne uređaje, što korisnicima otežava pristup informacijama u pokretu. S druge strane, kupovina automobila od ovlaštenog prodavača često donosi prednosti kao što su jamstvo tehničke ispravnosti i garanciju samog vozila, no često se sličan automobil može pronaći kod privatnog prodavača za nižu cijenu, što kupca često dovodi u opasnost od prevara.

2.3 Analiza prodaje i stanje web pretraživanja automobila

Sve više se može uočiti porast internetske prodaje automobila. Razvojem tehnologije te poboljšanjem kvalitete videozapisa i umjetne inteligencije, omogućeno je dobivanje kristalno jasne slike automobila. Sve više proizvođača nudi web podršku za kupovinu automobila. Jedna od ključnih značajki je napredno filtriranje i usporedba vozila, što korisnicima omogućuje precizno definiranje kriterija pretraživanja, poput marke, modela, cijene, godine proizvodnje, kilometraže te mnogih drugih specifikacija.

Na slici 2.1 prikazan je graf po uzoru na [3] na kojem je prikazano povećanje broja on-line prodaje automobila. Također, prikazana je analiza za 2032. godinu, gdje se predviđa udvostručenje prodaje. Najveći čimbenik porasta internetske prodaje bila je pandemija virusa COVID-19. Tijekom pandemije zabilježen je pad prodaje, no to se pripisuje smanjenju prometa zbog višemjesečne karantene. Za to vrijeme potrošači su uvidjeli prednosti koje pruža internetska prodaja.



Slika 2.1. Istraživanje o online prodaji automobila [1]

Recenzije automobila uvelike pomažu korisnicima pronaći prikladan automobil, a mogu se pronaći u sklopu foruma gdje vlasnici dijele svoje stavove o automobilu. Isto su tako vrlo važne emisije gdje auto stručnjaci rade detaljnu analizu automobila i pružaju potrošačima profesionalnu pomoć pri odabiru automobila. Također postoje razni testovi automobila kao što je takozvani test sjevernog jelena ili NCAP test. Euro NCAP kreirao je sustav sigurnosne ocjene s pet zvjezdica kako bi potrošačima, njihovim obiteljima i tvrtkama olakšao usporedbu vozila te im pomogao u prepoznavanju najsigurnijeg izbora za njihove potrebe [4].

2.4. Pregled stanja u području

Trenutno stanje na tržištu web aplikacija za višekriterijsko odlučivanje pri kupovini automobila pokazuje različite primjere. Iako postoje brojne aplikacije koje korisnicima omogućuju usporedbu cijena i osnovnih karakteristika automobila, većina njih ne uzima u obzir druge parametre koji su vrlo bitni pri izboru automobila. Web aplikacije poput Njuškala [5] uglavnom pružaju filtriranje specifikacija automobila, no tako korisnici koji ne mogu odrediti koji parametri za izbor automobila su njima potrebni ostaju uskraćeni. Zbog toga trebaju provesti vrijeme proučavajući kako bi saznali točne podatke. Nedostaju aplikacije koje bi integrirale napredne algoritme višekriterijskog odlučivanja, kao što su Analitički hijerarhijski proces (AHP) ili TOPSIS. Pomoću njih korisnik vrlo lako može odrediti prioritete i preferencije pri odabiru automobila te dobiti objektivnu usporedbu različitih modela prema svojim kriterijima.. Iz tog razloga odlučeno je napraviti web aplikaciju koja bi korisnicima trebala pomoći u postizanju odluke pri kupovini automobila.

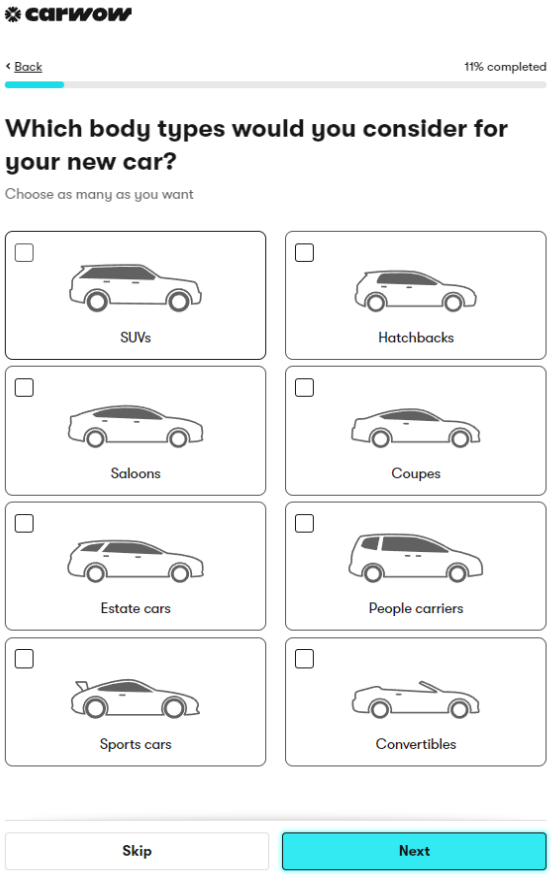
2.5 Postojeća rješenja

Postoje razne web aplikacije koje mogu pomoći pri odabiru automobila. U većini slučajeva, od korisnika se zahtijeva da sam filtrira odabire prema performansama, godini proizvodnje, tipu pogona, modelu itd. Prikazano je nekoliko web aplikacija kojima je svrha preporuka automobila prema potrebama korisnika.

2.5.1 Carwow

Carwow [6] je popularna web platforma za britansko tržište koja pruža potrošačima uvid u cijene automobila uspoređujući cijene u različitim auto salonima. Ponude se mogu pregledavati prema specifičnim kriterijima poput cijene, marke, modela i specifikacija vozila. Na platformi su

dostupne recenzije i ocjene vozila, što olakšava donošenje informiranih odluka. Također je omogućena zamjena starog za novo, kao i oglašavanje vlastitog auta. Na slici 2.2 prikazano je pitanje za korisnika u kojem treba odabrati vrstu karoserije.



Slika 2.2 Prikaz sučelja web aplikacije Carwow

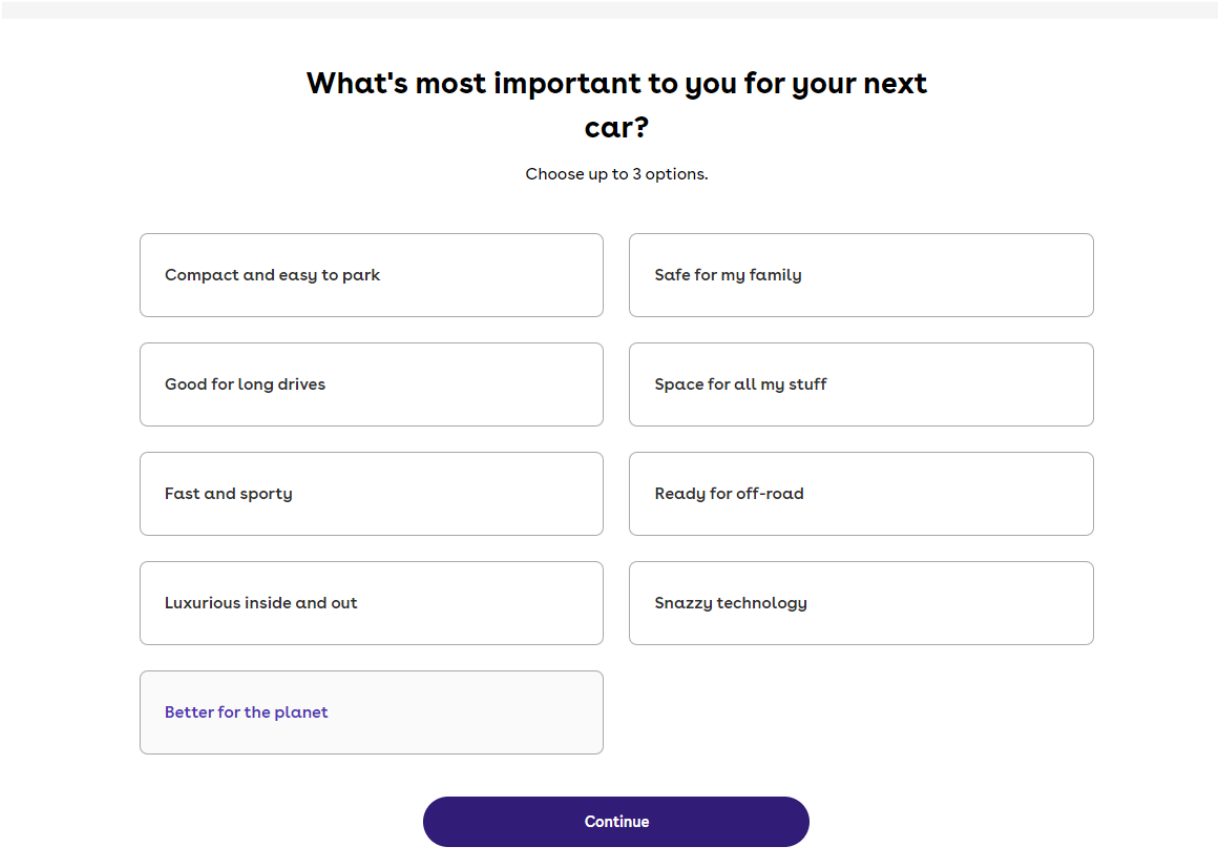
2.5.2 Chiriboo

Chiriboo [7] je hrvatska web aplikacija koja korisnicima nudi personalizirane preporuke automobila na temelju njihovih individualnih preferencija i potreba. Nudi mogućnosti filtriranja i uspoređivanja vozila. Pruža ponudu novih automobila na hrvatskom tržištu i razne olakšice pri dodavanju opreme automobila. Na slici 2.3 prikazan je izbor mogućnosti koje su dane korisniku. Moguće je pretraživati ponudu vozila i zatražiti ponudu prodavača. Dodatna funkcionalnost omogućava korisnicima da vide specifikacije više vozila, međusobno ih usporede te donesu odluku.

Slika 2.3 Prikaz sučelja web aplikacije Chiriboo

2.5.3 Cinch

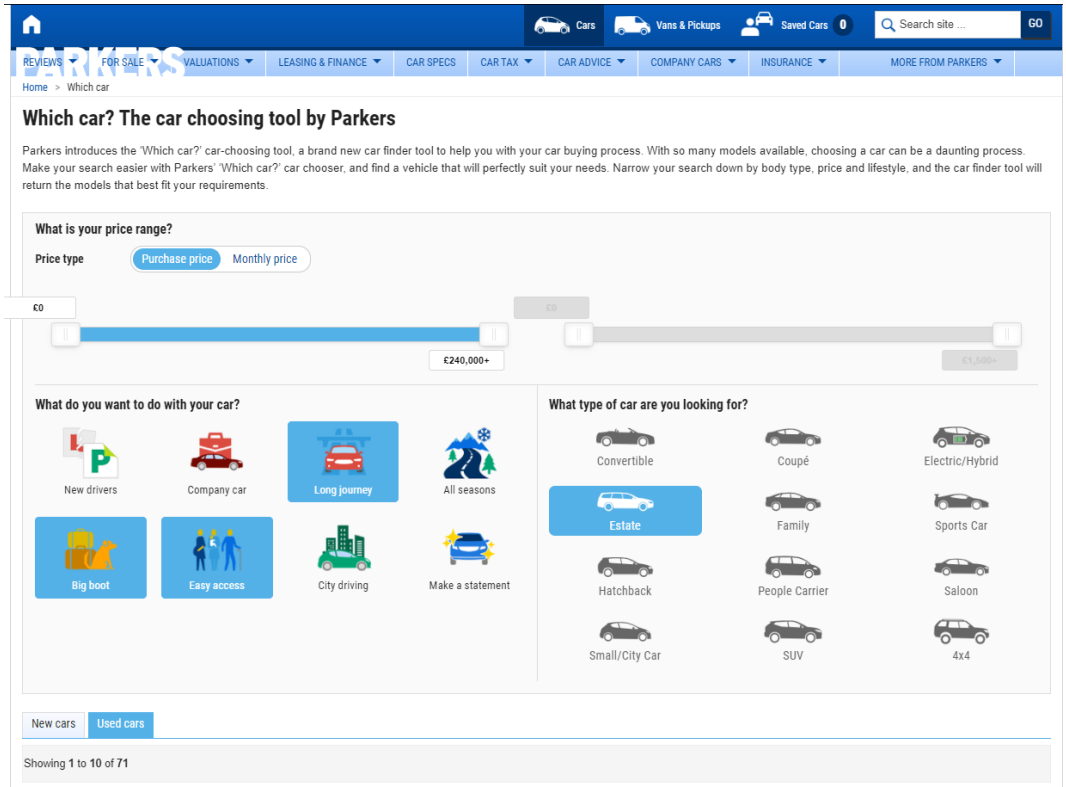
Cinch [8] je platforma koja nudi jednostavan način kupovine rabljenih automobila. Slabije informiranim korisnicima pruža pomoć u pronalaženju automobila koji im odgovara. Odgovaranjem na pitanja korisnik sužava izbor za kupovinu novog vozila. Na slici 2.4 prikazana su pitanja temeljena na onome što je od najveće važnosti za korisnika, npr. broj putnika u autu, važnost parkiranja u urbanom području, ekološki standardi. Cinch pruža detaljne informacije o svakom vozilu, uključujući povijest automobila i izvještaje o tehničkom stanju, kako bi korisnicima omogućio donošenje najboljih odluka prilikom kupovine.



Slika 2.4 Prikaz sučelja web aplikacije Cinch

2.5.4 Parkers

Parkers [9] nudi široku ponudu informacija i pomagala vezanih uz automobile. Na stranici se mogu pronaći detaljne recenzije automobila i točni podaci o svakom automobilu, ovisno o vrsti paketa koji posjeduje. Alat za procjenu automobila na osnovu registracijskog broja vozila je dostupan na stranici. Na slici 2.5 prikazan je alat koji pomaže korisniku pri donošenju odluke pri kupovini automobila. Od korisnika se zahtijeva unos cijene automobila, tipa karoserije i označavanje za što mu je potreban automobil.



Slika 2.5 Prikaz sučelja web aplikacije Parkers

3. MODEL APLIKACIJE ZA VIŠEKRITERIJSKO ODLUČIVANJE PREPORUKE PRI KUPOVINI AUTOMOBILA

U ovom poglavlju bit će objašnjeni funkcionalni i nefunkcionalni zahtjevi web aplikacije.

3.1. Funkcionalni zahtjevi na web aplikaciju

Funkcionalni zahtjevi [10] su zahtjevi koje korisnik postavlja kao sadržaje koje aplikacija nužno treba sadržavati. Predstavljeni su kao unos koji se pruža aplikaciji, zadatak koji ta aplikacija izvrši i rezultat. U nastavku su predstavljeni zahtjevi koje web aplikacija treba sadržavati.

3.1.1 Autentifikacija korisnika

Pri ulasku korisnika na web aplikaciju, korisnik će se moći registrirati svojim e-mail računom. Bit će potrebno upisati e-mail račun, ime i prezime te lozinku, te će pri svakim novim ulaskom na stranicu korisnikov odabir biti zapamćen. Nakon registracije, korisnik će se moći prijaviti koristeći samo e-mail račun i lozinku.

3.1.2 Odjava korisnika

Korisnici imaju mogućnost odjave sa svog korisničkog računa. Ova funkcionalnost bitna je za zaštitu privatnosti i sigurnosti korisničkih podataka. Klikom na gumb za odjavu, korisnik će se vratiti na početnu stranicu s formama za prijavu i registraciju.

3.1.3 Preporuka automobila

Nakon uspješne prijave, korisniku će biti omogućeno započinjanje s pronalaskom odgovarajućeg automobila tako što će odgovoriti na nekoliko pitanja vezanih uz specifikacije automobila. Korisnik će odgovoriti na jednostavna pitanja kao što su cijena, oblik automobila, broj sjedala, potrošnja i vrsta goriva. Korisniku će biti omogućeno vraćanje na prijašnja pitanja i promjenu odgovora.

3.1.4 Detalji o vozilu

Ponuđeni će automobil imati karticu s detaljnim informacijama. Korisnici će moći pregledati specifikacije i slike preporučenog automobila. U slučaju sličnog izbora, korisniku će biti ponuđen izbor između nekoliko vozila. Na kartici će biti dostupne i ostale specifikacije kao što su performanse vozila (ubrzanje do 100 km/h i krajnja brzina vozila), veličina utovarnog prostora,

godina proizvodnje i slično. U slučaju vozila s unutarnjim izgaranjem, korisnik će moći vidjeti CO2 emisiju i emisijski standard vozila. Električna vozila imat će iskazan kapacitet baterije, prosječnu potrošnju energije te snagu vozila u kWh.

3.2 Nefunkcionalni zahtjevi na web aplikaciju

Nefunkcionalni zahtjevi [11] u programskom inženjerstvu su karakteristike sustava koje nisu povezane s točnom funkcionalnošću ili ponašanjem. To su ključni aspekti kvalitete proizvoda koji se pri ubrzanom razvoju često zanemaruju. Oni opisuju kako bi sustav trebao izvoditi određene aspekte, umjesto da definiraju što sustav treba raditi. Neki od primjera zahtjeva su brzina odgovora sustava i njegova skalabilnost, zaštita od neovlaštenog pristupa, jednostavnost korištenja, lakoća održavanja i proširivost te podržavanje različitih sustava. Važnost nefunkcionalnih zahtjeva leži u zadovoljstvu korisnika i utječe na uspjeh sustava. Aplikacija je izgrađena da bude jednostavna za korištenje i ugodna oku korisnika te da sadržava interaktivne dijelove i detalje. Dostupna je na većini uređaja kao što su stolna računala, laptopi i mobiteli. Upitnik je sastavljen da bude razumljiv korisniku, što je ostvareno korištenjem jednostavnih fraza koje ne zahtijevaju posebno znanje o automobilima. Izgled preporučenog automobila je interaktivan s velikim slikama i pregršt detalja o vozilu.

3.3 Višekriterijsko odlučivanje

Višekriterijsko odlučivanje [12] koristi se pri donošenju odluka kada je zadano više kriterija koji se trebaju uzeti u obzir. U kontekstu automobila, svaki je korisnik jedinstven te zahtijeva specifične kriterije i parametre pri odlučivanju. Jedna od najpopularnijih metoda je analitička hijerarhija (AHP). Omogućuje korisnicima da izvrše parno uspoređivanje kriterija i parametara te pomaže u određivanju relativne važnosti svakog kriterija. Ostale metode, poput TOPSIS i PROMETHEE, također omogućuju analizu i rangiranje opcija pomoću višekriterijskih analiza. U ovom specifičnom slučaju, korisnicima je predstavljen jednostavan upitnik pomoću kojeg se donosi odluka o preporučenom automobilu.

U aplikaciji je korišten način višekriterijskog odlučivanja koji uzima u obzir bitne parametre koji uključuju cijenu automobila, oblik automobila, najveći mogući kapacitet osoba u automobilu, tip goriva i potrošnju. Na temelju tih parametara, koji su spremljeni u bazi, te upitnika ispunjenog od strane korisnika, algoritmom će se odlučiti najbolji automobil za korisnika. Za svaki odgovor na pitanje u upitniku, automobilima iz baze će se dodjeljivati bodovi, ovisno o tome zadovoljavaju li

uvjete postavljene u upitniku. Nakon završetka upitnika, algoritam će na osnovu najvećeg broja ostvarenih bodova odlučiti koji će automobil biti preporučen korisniku.

U konačnici, važno je da se višekriterijskim odlučivanjem donese informirana odluka koja uzima u obzir sve bitne kriterije i parametre. Time se poboljšava korisničko iskustvo i zadovoljstvo, čime se pridonosi rastu i širenju aplikacije.

4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE

U ovome poglavlju će biti opisani programski jezici, tehnologije i alati korišteni za izradu web aplikacije, te programska rješenja na poslužiteljskoj i korisničkoj strani.

4.1. Korištene tehnologije i alati

4.1.1. HTML

HTML [13] (eng. Hypertext Markup Language) je označni jezik koji pomaže pri stvaranju web aplikacija. Napisao ga je Tim Berners-Lee 1993. godine. Daje opis o strukturi sadržaja unutar HTML datoteke te pomaže u prikazivanju teksta, slika i drugih oblika na web aplikaciji. Elementi se sastoje od oznaka i sadržaja. HTML je službena preporuka W3C (eng. World Wide Web Consortium). Najnovija je verzija HTML 5 koja je ujedno korištena u web aplikaciji.

4.1.2. SCSS

Sassy Cascading Style Sheets (SCSS) [14] je stilski jezik koji je nadskup CSS-a, razvijen 2006. godine od strane Hampton Catlina. Stilski je jezik koji prvenstveno pomaže oblikovati kod napisan u HTML-u. Proširuje mogućnosti standardnog CSS-a omogućavajući bolju produktivnost i lakše održavanje stilskih listi. Njegova sintaksa slična je CSS-u, uz dodatne funkcionalnosti, kao što su varijable i ugnježđivanje.

4.1.3. Bootstrap

Bootstrap [15] je frontend razvojni okvir koji pruža programerima zbirku sintakse kako bi im pomogao pri stvaranju web aplikacije. Pruža dizajniran kod u HTML-u i CSS-u koji je moguće umetnuti u vlastitu web aplikaciju te ga prilagoditi potrebama aplikacije. Nastao je 2011. godine za potrebe društvene mreže Twitter.

4.1.4. JavaScript

JavaScript [16] najkorišteniji je programski jezik koji služi za stvaranje web stranica. Objavljen je 1995. godine od strane NetScape-a, te je prvo nazvan LiveScript. Koristi se za razvoj interaktivnih HTML stranica. Pruža interakciju s korisnikom, stvaranje animacija i dinamičko stvaranje HTML sadržaja. Koristi se za izradu web aplikacija i poslužitelja, igara na pregledniku, mobilnih aplikacija, sigurnosnih značajki. Glavne su mu prednosti jednostavna sintaksa te globalna podržanost od svih glavnih preglednika.

4.1.5 React

ReactJS [17] je popularna JavaScript biblioteka za izgradnju korisničkih sučelja koja pruža dinamično i interaktivno sučelje web aplikacije. Razvila ju je tvrtka Facebook. Ističe se po komponentnoj arhitekturi gdje svaka komponenta ima svoje stanje i svojstva. Također koristi virtualni DOM (eng. Document Object Model) koji je brži od stvarnog DOM-a. Fleksibilan je alat i vrlo lako komunicira s drugim alatima i sučeljima.

4.1.6. Redux

Redux [18] je predvidljiva biblioteka za upravljanje stanjem aplikacija koja se najčešće koristi u kombinaciji s React sučeljem. Razvijen je kako bi olakšao upravljanje složenim stanjima u aplikacijama, omogućujući centralizirano skladištenje i jednostavno upravljanje promjenama stanja. Ključne komponente su skladišta, akcije i reduktori. U skladištu se čuvaju stanja aplikacije, akcije su objekti koji opisuju što se događa u aplikaciji te se šalju u skladišta kako bi pokrenuli promjene stanja. Reduktori su funkcije koje primaju trenutno stanje i akciju te vraćaju novo stanje. Glavne prednosti su predvidljivost, centraliziranost stanja, lako testiranje i ispravljanje pogrešaka.

4.1.7. Visual Studio Code

Visual Studio Code [19] je besplatan uređivač izvornog koda koji je razvio Microsoft. Dizajniran je da bude jednostavan za korištenje sa širokim rasponom značajki i proširenja koja ga čine prikladnim za različite programske jezike i razvojne zadatke. Dostupan je za Windows, macOS i Linux operacijske sustave. Kompatibilan je s većinom programskih jezika i sučelja, također pruža proširenja koja omogućuju dovršavanje koda, kontrolu verzija i ispravljanje pogrešaka. Također pruža integrirani terminal i podršku za Git i SVN.

4.1.8. Firebase

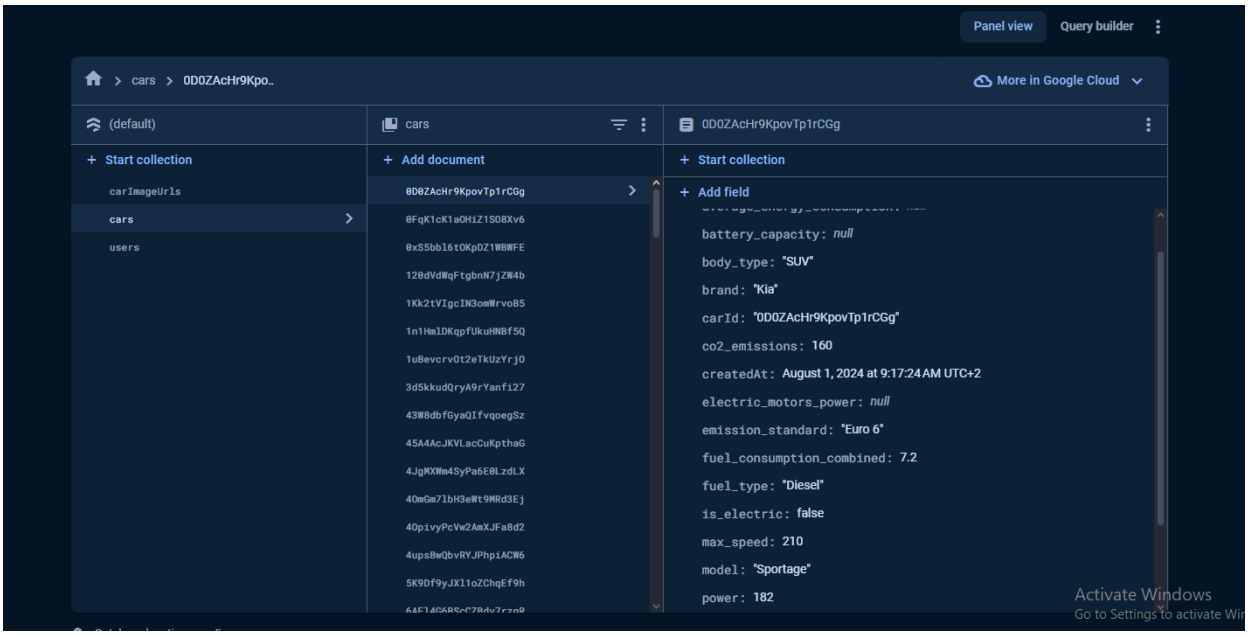
Firebase je platforma za razvoj aplikacija koju je razvila tvrtka Google, koja omogućava programerima mobilnih i web aplikacija da brzo i jednostavno izgrade svoju aplikaciju. Ova platforma pruža usluge pohrane podataka, autentifikaciju korisnika, analitiku i hosting. Jedne od ključnih karakteristika su stvaranje cloud baze podataka i sinkronizacija u stvarnom vremenu. Omogućena je podrška autentifikacije korisnika putem e-pošte i lozinke. Pruža analitiku same aplikacije i omogućava web smještaj aplikacije. Omogućuje vrlo lako testiranje i pruža izvještaje o greškama u stvarnom vremenu.

4.2 Prikaz programskog rješenja

U nastavku će biti opisani ključni dijelovi programskog koda.

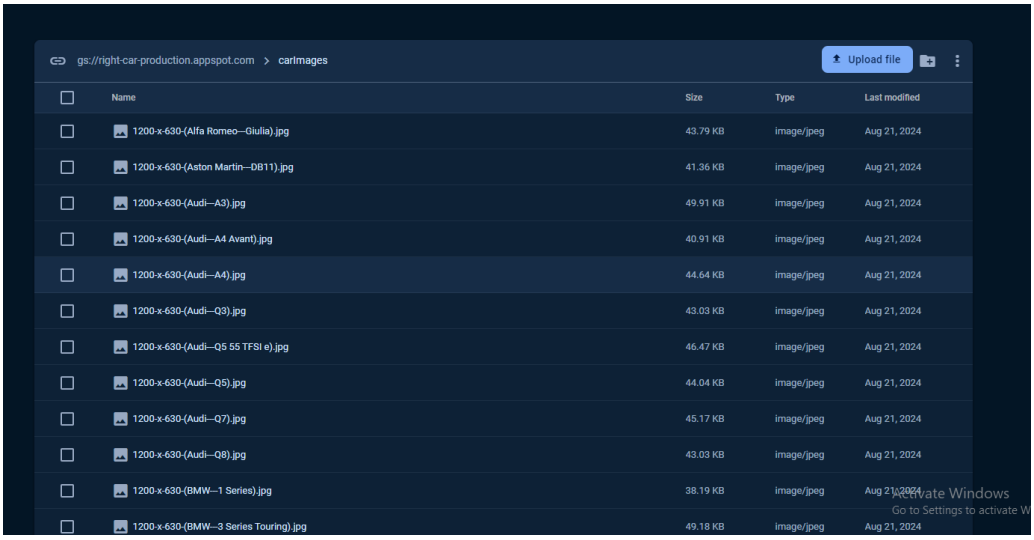
4.2.1 Baza podataka

Putem Cloud Firestore-a pohranjeni su automobili i njihove specifikacije, a također su pohranjeni podatci registriranih korisnika i slike u obliku URL, kao što je prikazano na slici 4.1.



Slika 4.1 Prikaz baze podataka automobila

Slika 4.2 prikazuje da su slike spremljene na Storage. Time se štede resursi Cloud Firestore-a jer se slike ne spremaju direktno u njega, već se pozivaju putem URL-a.



gs://right-car-production.appspot.com > carImages

Upload file

Name	Size	Type	Last modified
1200-x-630-(Alfa Romeo—Giulia).jpg	43.79 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Aston Martin—DB11).jpg	41.36 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—A3).jpg	49.91 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—A4 Avant).jpg	40.91 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—A4).jpg	44.64 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—Q3).jpg	43.03 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—Q5 55 TFSI e).jpg	46.47 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—Q5).jpg	44.04 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—Q7).jpg	45.17 KB	image/jpeg	Aug 21, 2024
1200-x-630-(Audi—Q8).jpg	43.03 KB	image/jpeg	Aug 21, 2024
1200-x-630-(BMW—1 Series).jpg	38.19 KB	image/jpeg	Aug 21, 2024
1200-x-630-(BMW—3 Series Touring).jpg	49.18 KB	image/jpeg	Aug 21, 2024

Slika 4.2 Prikaz baze podataka slika automobila

4.2.2 Registracija korisnika

Na slici 4.3 prikazana je funkcija *signUp* koja služi za registraciju novog korisnika putem Firebase autentifikacije i baze podataka. U tijeku izvršenja, kod koristi funkciju *createUserWithEmailAndPassword* za stvaranje novog korisnika koristeći e-mail i lozinku. Ako je registracija uspješna, kreira se objekt koji uključuje jedinstveni identifikator korisnika. Ovi podaci se potom pohranjuju u bazu podataka pod zbirkom *users*.

```
JS authAction.js ×
src > store > actions > JS authAction.js
1   import { createUserWithEmailAndPassword, signInWithEmailAndPassword, signOut } from 'firebase/auth';
2   import { auth, db } from '../../config/firebase';
3   import { doc, getDoc, setDoc } from 'firebase/firestore';
4   import { toast } from 'react-toastify';
5
6   export const signUp = ({ payload, onComplete = () => { } }) => async (dispatch) => {
7     dispatch({ type: 'SIGN_UP_PENDING' });
8     try {
9       const { email, password } = payload;
10      const userCredential = await createUserWithEmailAndPassword(auth, email, password)
11      const user = userCredential.user;
12      const userData = {
13        uid: user.uid,
14        createdAt: new Date(),
15        ...payload,
16      }
17      await setDoc(doc(db, 'users', user.uid), userData);
18      localStorage.setItem('user', JSON.stringify(userData));
19      onComplete();
20      toast.success('Signed up successfully')
21      dispatch({ type: 'SIGN_UP_SUCCESS', payload: userData });
22    } catch (error) {
23      console.log('error', error.message)
24      toast.error(error.message || 'An unknown error occured')
25      dispatch({ type: 'SIGN_UP_FAILURE', payload: error.message });
26    }
27  };
```

Slika 4.3 Prikaz funkcije za registraciju novog korisnika

4.2.3 Prijava i odjava korisnika

Na slici 4.4 je prikazana funkcija *signIn* koja služi za prijavu korisnika putem e-mail i lozinke koristeći Firebase autentifikaciju. Prvo šalje akciju kako bi označila početak prijave. Zatim koristi *signInWithEmailAndPassword* kako bi prijavila korisnika i dohvatila njegove podatke iz baze. Ti se podaci pohranjuju u lokalnu bazu, a ako dođe do greške, funkcija hvata i prijavljuje grešku.

```
export const signIn = ({ payload, onComplete = () => { } }) => async (dispatch) => {
  dispatch({ type: 'SIGN_IN_PENDING' });
  try {
    const { email, password } = payload;
    const userCredential = await signInWithEmailAndPassword(auth, email, password);
    const user = userCredential.user;

    const userDoc = await getDoc(doc(db, 'users', user.uid));
    const userData = userDoc.exists() ? userDoc.data() : null;

    localStorage.setItem('user', JSON.stringify(userData));
    toast.success('Signed in successfully')
    onComplete();
    dispatch({ type: 'SIGN_IN_SUCCESS', payload: userData });
  } catch (error) {
    console.log('error', error.message)
    toast.error(error.message || 'An unknown error occured')
    dispatch({ type: 'SIGN_IN_FAILURE', payload: error.message });
  }
};
```

Slika 4.4 Prikaz funkcije za prijavu korisnika

Funkcija *logout* prikazana na slici 4.5 služi za odjavu korisnika iz aplikacije. Prvo šalje akciju LOGOUT_PENDING kako bi označila početak odjave. Zatim koristi *signOut* iz Firebase autentifikacije kako bi odjavila korisnika i uklanja podatke o korisniku iz lokalne baze. Nakon uspješne odjave šalje se akcija LOGOUT_SUCCESS. Ako dođe do greške, hvata grešku, prikazuje poruku o grešci i šalje akciju LOGOUT_FAILURE.

```
export const logout = () => async (dispatch) => {
  dispatch({ type: 'LOGOUT_PENDING' });
  try {
    await signOut(auth);
    localStorage.removeItem('user');
    toast.success('Logged out successfully')
    dispatch({ type: 'LOGOUT_SUCCESS' });
  } catch (error) {
    console.log('error', error.message);
    toast.error(error.message || 'An unknown error occured')
    dispatch({ type: 'LOGOUT_FAILURE', payload: error.message });
  }
};
```

Slika 4.5 Prikaz funkcije za odjavu korisnika

4.2.4 Upitnik za korisnika

Na slici 4.6 prikazana su dva od šest pitanja koja se postavljaju korisniku u upitniku. Jedno od pitanja odnosi se na potrošnju goriva. Korisniku su ponuđena tri moguća odgovora:

1. potrošnja goriva mu je presudna pri kupovini,

2. potrošnja goriva nije mu presudna, ali bi svakako htio uštedjeti na gorivu,

3. potrošnja goriva nije bitna te pridaje prednost performansama automobila.

U prvom slučaju, ako korisnik odabere da mu je potrošnja goriva presudna pri kupovini, povezuje se uvjet da potrošnja goriva treba biti ispod 5.5 litara na 100 kilometara. U drugom slučaju, ako korisniku nije toliko bitno, ali bi svakako htio uštedjeti na gorivu, uvjet se povećava na 10 litara na 100 kilometara. U trećem slučaju, ako korisniku potrošnja goriva nije bitna i želi auto s boljim performansama, povezuje se uvjet da snaga automobila treba biti veća od 300 konjskih snaga i da najveća moguća brzina mora biti veća od 240 kilometara na sat. Isto tako može se vidjeti pitanje o obliku karoserije, gdje je moguće izabrati između terenca, limuzine, manjeg auta ili dvosjeda. Potrebno je navesti da su uvjeti postavljeni tako da bi odgovarali većini korisnika.

```
{
  id: 'fuelEfficiency',
  question: 'Koliko vam je bitna potrošnja goriva?',
  options: [
    { id: 'veryImportant', label: 'Vrlo bitno – Želim minimizirati troškove goriva.',
      query: [{ field: 'fuel_consumption_combined', operator: '<', value: 5.5 }] },
    { id: 'somewhatImportant', label: 'Djelomično bitno – Volio bih uštedjeti na gorivu, ali nije presudno.',
      query: [{ field: 'fuel_consumption_combined', operator: '<', value: 10 }] },
    {
      id: 'notImportant', label: 'Nije bitno – Dajem prednost performansama.', query: [
        { field: 'power', operator: '>=', value: 300 },
        { field: 'max_speed', operator: '>=', value: 240 }
      ]
    }
  ]
},
{
  id: 'bodyType',
  question: 'Koji tip vozila preferirate?',
  options: [
    { id: 'sedan', label: 'Limuzina', query: [{ field: 'body_type', operator: '==', value: 'Sedan' }] },
    { id: 'hatchback', label: 'Hatchback', query: [{ field: 'body_type', operator: '==', value: 'Hatchback' }] },
    { id: 'suvCrossover', label: 'SUV/Crossover', query: [{ field: 'body_type', operator: 'in', value: ['SUV', 'Crossover'] }] },
    { id: 'coupeConvertible', label: 'Coupe/Kabriolet', query: [{ field: 'body_type', operator: 'in', value: ['Coupe', 'Convertible'] }] }
  ]
},
```

Slika 4.6 Dio programskog koda pitanja za korisnika

Na slici 4.7 prikazane su funkcije za kretanje kroz upitnik. Funkcija *nextQuestion* koristi se za pomicanje aktivnog pitanja na sljedeće pitanje u upitniku, postavljanje pogleda na vrh stranice i ažuriranje aktivnog stanja. Također, korisniku je omogućeno kretanje unazad pomoću funkcije *previousQuestion*. Funkcija *handleSelect* ažurira stanje odgovora korisnika i čuva svaki odgovor u objektu gdje je ključ ID pitanja, vrijednost odgovor. Nakon što se sačuva odgovor prelazi se na sljedeće pitanje. Kada korisnik odgovori na sva pitanja prestaje se s upitnikom i preusmjerava se na stranicu koja prikazuje ponuđeni automobil.

```
const nextQuestion = () => {
  if (activeQuestionIndex + 1 <= questions.length - 1) {
    setTimeout(() => {
      window.scroll(0, 0);
      setActiveQuestionIndex(activeQuestionIndex + 1)
    }, 500);
  }
};

const previousQuestion = (e) => {
  e.preventDefault();
  if (activeQuestionIndex !== 0) {
    setActiveQuestionIndex(activeQuestionIndex - 1)
  }
};
```

```
const handleSelect = (questionId, answer) => {
  setAnswers(prev => ({
    ...prev,
    [questionId]: answer
  }));
  nextQuestion();
};

const submitHandler = (e) => {
  e.preventDefault();
  dispatch(fetchCarsData({
    answers,
    onComplete: () => {
      history.push('/car-results');
    }
  }));
};
```

Slika 4.7 Programski kod za kretanje kroz upitnik

4.2.5 Stvaranje preporuke automobila

Jedna od ključnih funkcija aplikacije je *fetchCarsData*, prikazana na slici 4.8, koja upravlja interakcijom s bazom podataka te služi za dohvaćanje podataka o automobilima na temelju odgovora korisnika. U ovoj funkciji implementirano je višekriterijsko odlučivanje obrazloženo u poglavlju 3.3. Funkcija iterira kroz odgovore korisnika, te za svaki odgovor koji sadrži određene upite generira uvjete pretrage koji se pohranjuju u polje *conditions*. Na temelju tih uvjeta kreira se upit prema bazi podataka koristeći Firestore metodu *where*, čime se omogućuje filtriranje automobila prema specifičnim parametrima kao što su potrošnja goriva, snaga motora ili oblik karoserije. Funkcija *matchingScore* prikazana na slici 4.9 dio je višekriterijskog odlučivanja izračunava broj bodova za svaki automobil na temelju podudarnosti njegovih karakteristika s korisnikovim odgovorima. Ova funkcija iterira kroz sve automobile u bazi i provjerava svaki postavljeni uvjet uspoređujući stvarne vrijednosti automobila s traženim vrijednostima korisnika. Svaki put kada automobil zadovolji uvjet, dodaje mu se bod. Na kraju, automobili se sortiraju prema broju ostvarenih bodova, tako da se prikazuju oni automobili koji najbolje odgovaraju korisnikovim zahtjevima. Stanje aplikacije se potom ažurira ovisno o uspješnosti dohvaćanja rezultata iz baze podataka, te se korisniku prikazuju automobili koji su najbolje rangirani prema zadanim kriterijima.

```
import { collection, getDocs, query, where } from 'firebase/firestore';
import { db, storage } from '../../config/firebase';
import { getDownloadURL, listAll, ref } from "firebase/storage";

export const fetchCarsData = ({ answers, onComplete = () => { } }) => {
  return async (dispatch) => {
    dispatch({ type: 'FETCH_CARS_PENDING' });
    try {
      const carRef = collection(db, 'cars');
      let carQuery = carRef;

      let conditions = [];
      Object.values(answers).forEach(answer => {
        if (answer.query) {
          answer.query.forEach(condition => {
            conditions.push(condition);
          });
        }
      });

      const createQuery = (conditions) => {
        let queryRef = carRef;
        conditions.forEach(condition => {
          queryRef = query(queryRef, where(condition.field, condition.operator, condition.value));
        });
        return queryRef;
      };
```

Slika 4.8 Programski kod funkcije *fetchCarsData*

```
allCars = allCars.map(car => {
  const matchingScore = conditions.reduce((score, condition) => {
    const carValue = car[condition.field];
    switch (condition.operator) {
      case '==':
        if (carValue == condition.value) score++;
        break;
      case '>':
        if (carValue > condition.value) score++;
        break;
      case '>=':
        if (carValue >= condition.value) score++;
        break;
      case '<':
        if (carValue < condition.value) score++;
        break;
      case '<=':
        if (carValue <= condition.value) score++;
        break;
      case '!=':
        if (carValue != condition.value) score++;
        break;
      case 'in':
        if (Array.isArray(condition.value) && condition.value.includes(carValue)) score++;
        break;
      default:
        console.warn(`Unsupported operator: ${condition.operator}`);
        break;
    }
```

Slika 4.9 Programski kod funkcije *matchingScore*

Funkcija prikazana na slici 4.10 dohvaća URL slike iz baze podataka koja se prikazuje na stranici s odabranim automobilom.

```
export const getImageUrlByName = (imageName) => async (dispatch) => {
  try {
    const imageRef = ref(storage, `carImages/${imageName}`);
    const url = await getDownloadURL(imageRef);
    return url;
  } catch (error) {
    console.error('Error getting image URL:', error.message);
    throw error;
  }
};
```

Slika 4.10 Funkcija za dohvaćanje slike automobila

4.2.6 Prikaz automobila

Kod stranice s preporučenim automobilom prikazana je na slici 4.11. Korišteni su parametri iz baze pridruženi predloženom automobilu. Parametri su prikazani pomoću ikonica.

```
const carFeatures = cars.length > 0 ?
  [
    { label: "Ubrzanje", value: cars[currentCarIndex].acceleration ?? "Null", icon: "fa-tachometer-alt", color: "#007bff" },
    { label: "Prosječna potrošnja energije", value: cars[currentCarIndex].average_energy_consumption ?? "Null", icon: "fa-battery-full", color:
    { label: "Kapacitet baterije", value: cars[currentCarIndex].battery_capacity ?? "Null", icon: "fa-battery-full", color: "#007bff" },
    { label: "Tip karoserije", value: cars[currentCarIndex].body_type ?? "Null", icon: "fa-car", color: "#6c757d" },
    { label: "CO2 Emisije", value: cars[currentCarIndex].co2_emissions !== null ? `${cars[currentCarIndex].co2_emissions} g/km` : "Null", icon:
    { label: "Snaga električnih motora", value: cars[currentCarIndex].electric_motors_power ?? "Null", icon: "fa-bolt", color: "#ffc107" },
    { label: "Standard emisije", value: cars[currentCarIndex].emission_standard ?? "Null", icon: "fa-leaf", color: "#20c997" },
    { label: "Kombinirana potrošnja goriva", value: cars[currentCarIndex].fuel_consumption_combined !== null ? `${cars[currentCarIndex].fuel_co
    { label: "Vrsta goriva", value: cars[currentCarIndex].fuel_type ?? "Null", icon: "fa-oil-can", color: "#17a2b8" },
    { label: "Električni", value: cars[currentCarIndex].is_electric ? "Yes" : "No", icon: "fa-bolt", color: "#ffc107" },
    { label: "Maksimalna brzina", value: cars[currentCarIndex].max_speed !== null ? `${cars[currentCarIndex].max_speed} km/h` : "Null", icon: '
    { label: "Model", value: cars[currentCarIndex].model ?? "Null", icon: "fa-car-side", color: "#6c757d" },
    { label: "Snaga", value: cars[currentCarIndex].power !== null ? `${cars[currentCarIndex].power} HP` : "Null", icon: "fa-bolt", color: "#ffc
    { label: "Cijena", value: cars[currentCarIndex].price !== null ? `$${cars[currentCarIndex].price}` : "Null", icon: "fa-dollar-sign", color:
    { label: "Broj sjedala", value: cars[currentCarIndex].seats !== null ? cars[currentCarIndex].seats : "Null", icon: "fa-chair", color: "#343
    { label: "Kapacitet prtljažnika", value: cars[currentCarIndex].trunk_capacity !== null ? `${cars[currentCarIndex].trunk_capacity} L` : "Nul
    { label: "Godina Proivodnje", value: cars[currentCarIndex].years_of_production ?? "Null", icon: "fa-calendar-alt", color: "#fd7e14" }
  ] : [];
```

Slika 4.11 Programski Kod prikaza automobila

4.2.7. Dekorativni dijelovi aplikacije

Kako bi aplikacija izgledala ugodno oku korisnika, na početnoj stranici napravljen je interaktivni klizač koji prikazuje fotografije. U tu svrhu korištena je biblioteka *react-slick* koja ima ugrađene komponente i funkcije za dekoriranje klizača. Na slici 4.12 prikazan prikazane su funkcije za kretanje kroz klizač, te postavljanje interaktivnih dijelova kao što su strjelice za kretanje kroz klizač, vrijeme svake animacije i brzinu prelaska.

```
const HeroSlider = () => {
  const dispatch = useDispatch();
  const sliderRef = useRef(null);
  const { landingImages, landingImagesFetched } = useSelector((state) => state.car);

  const handleNext = () => {
    sliderRef.current.slickNext();
  };

  const handlePrev = () => {
    sliderRef.current.slickPrev();
  };

  const settings = {
    arrows: true,
    dots: true,
    fade: true,
    infinite: true,
    autoplay: true,
    speed: 500,
    slidesToShow: 1,
    slidesToScroll: 1,
    waitForAnimate: false,
  };

  useEffect(() => {
    if (!landingImagesFetched) {
      dispatch(fetchLandingImages());
    }
  }, [landingImagesFetched]);
```

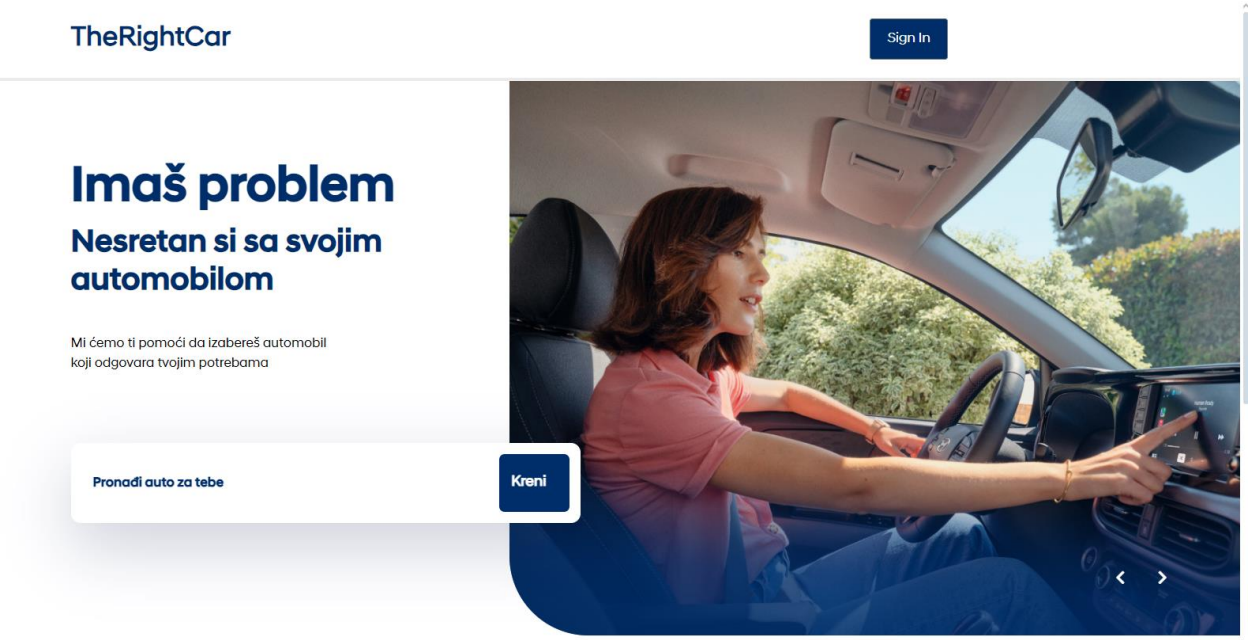
Slika 4.12 Kod interaktivnog klizača

5. PRIKAZ NAČINA RADA I ISPITIVANJE WEB APLIKACIJE

U ovom poglavlju opisan je način korištenja aplikacije od strane korisnika, dizajn stranice i ispitivanje rada i točnosti.

5.1. Način korištenja aplikacije

Pri otvaranju web aplikacije korisniku je prikazan početni zaslon kao što je prikazano na slici 5.1. U zaglavlju su prikazani ime web stranice i gumb za prijavi ili registraciju. Nalazi se interaktivni klizač koji prikazuje fotografije, kao i gumb koji korisnika preusmjerava na upitnik.

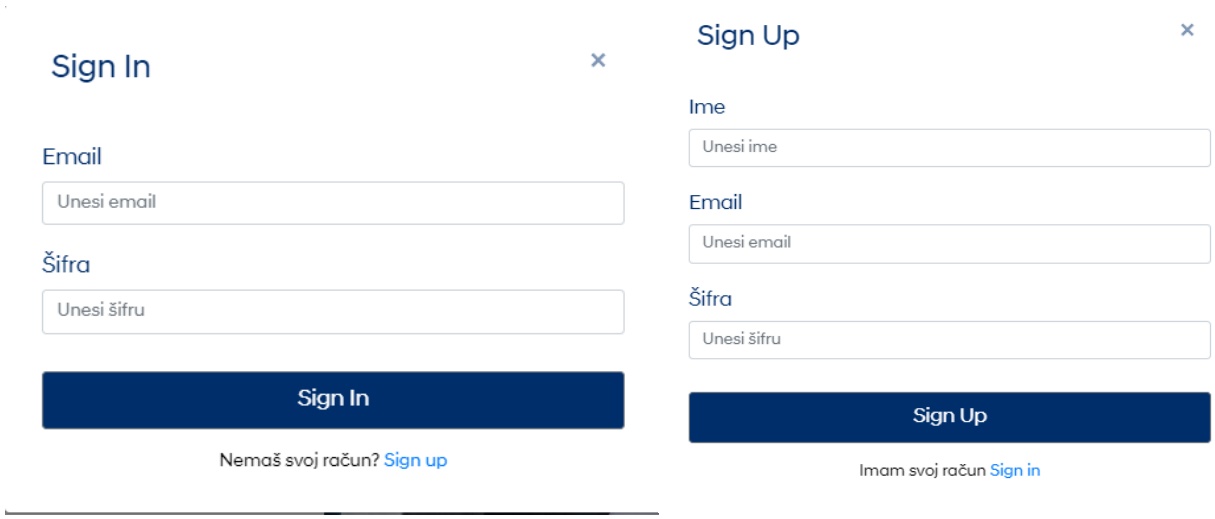


Slika 5.1 Izgled početne stranice

Isto tako, na početnoj stranici nalazi se klizač prikazan na slici 5.2 koji prikazuje četiri nasumične slike automobila sadržanih u bazi.

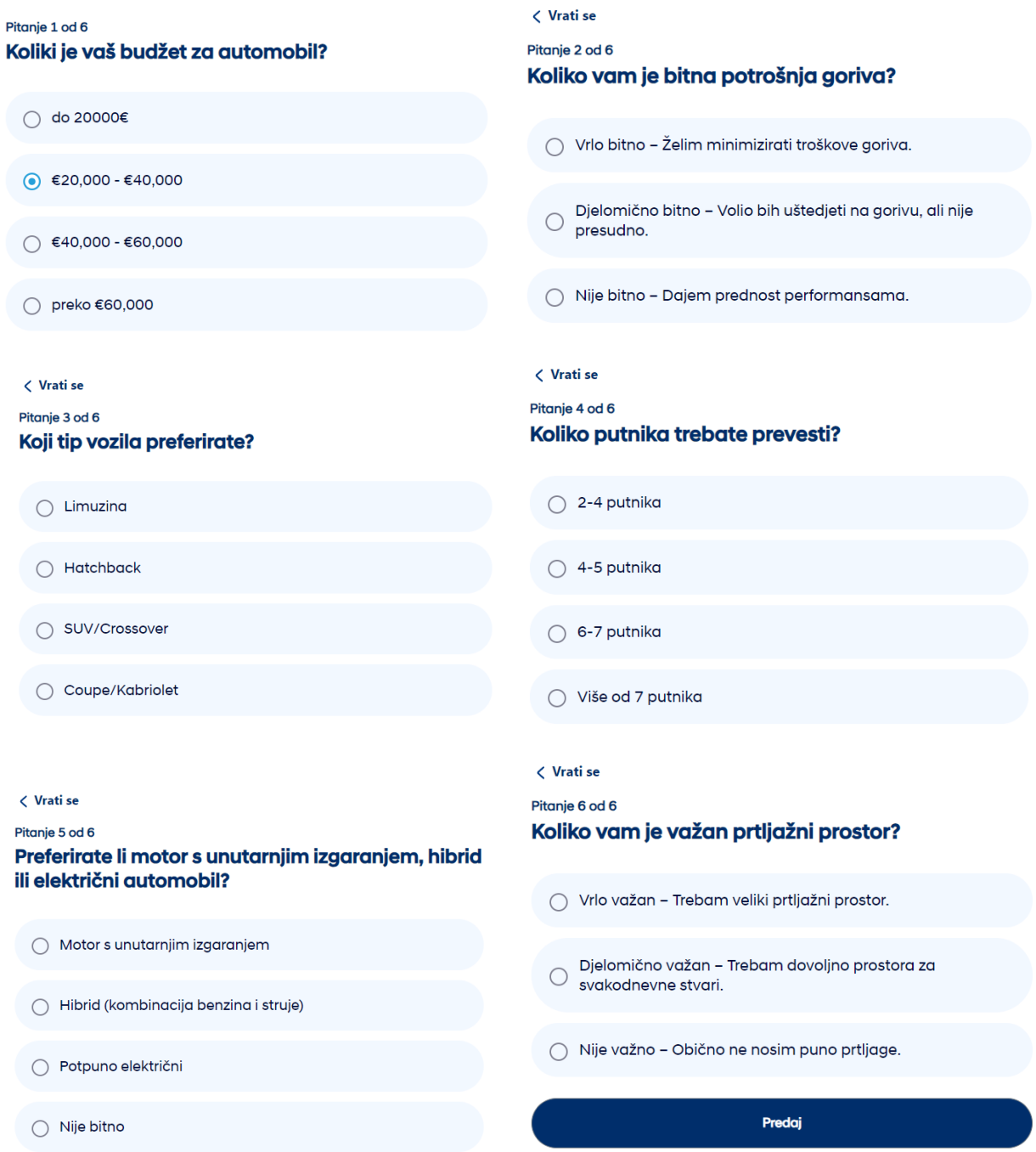
Slika 5.2 Izgled interaktivnog klizača

Prilikom prvog ulaska na web aplikaciju korisnik je zatražen registrirati se tako što će unijeti ime, e-mail adresu i lozinku kao što je prikazano na slici 5.3. Prilikom svakog slijedećeg ulasku u aplikaciju korisnik će se moći prijaviti svojim računom, kao što je prikazano na slici 5.4.



Slike 5.3 i 5.4 Izgled prozora za registraciju i prijavu korisnika

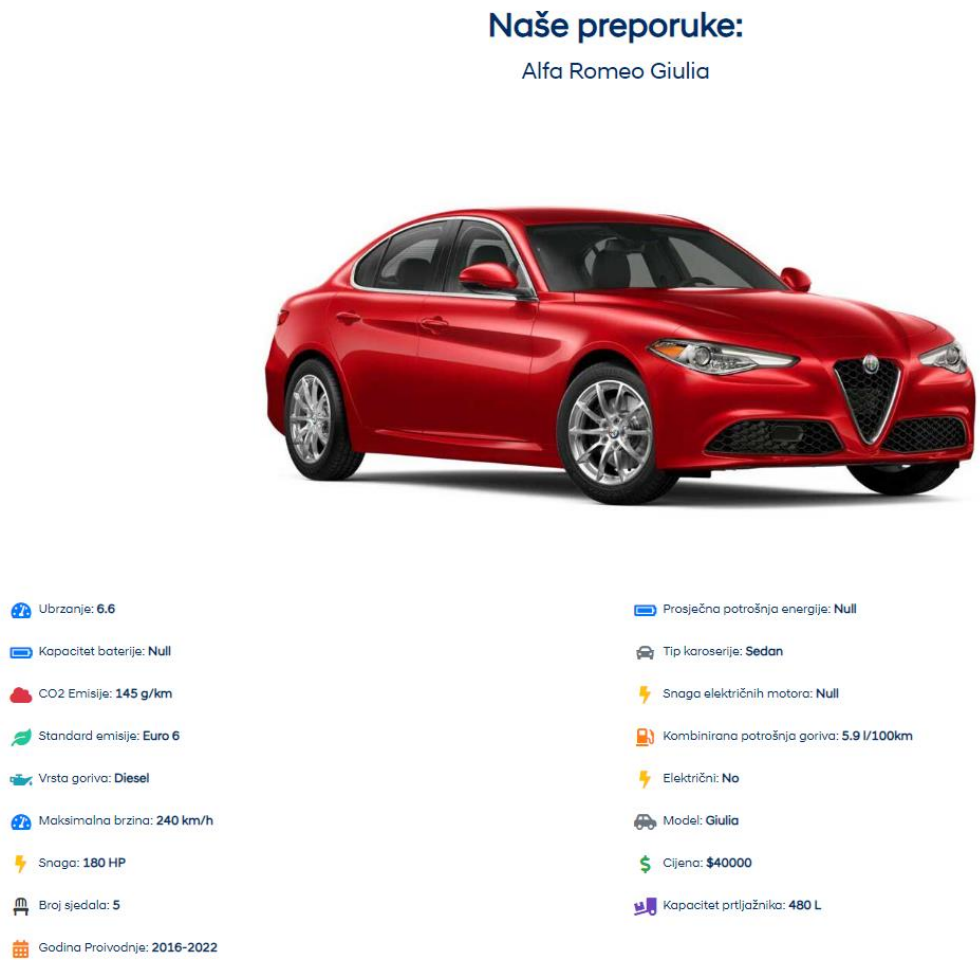
Pritiskom na gumb "Kreni", korisnik započinje s upitnikom. Upitnik prikazan na slikama od 5.5 do 5.10 sastavljen je od šest pitanja gdje korisnik treba odabrati odgovor na svako pitanje. Nakon odgovora na pitanje, omogućeno je vraćanje na prethodno pitanje kako bi korisnik mogao promijeniti svoj odgovor.



Slike 5.5 do 5.10 Izgled upitnika

Nakon što je korisnik odgovorio na pitanja u upitniku i predao svoje odgovore, preusmjerava se na stranicu gdje je prikazana preporuka za automobil. Na slici 5.11 prikazano je naziv i slika automobila, dok su u nastavku navedene specifikacije vozila. Korisniku su malim ikonama

prikazane informacije poput ubrzanja, snage vozila, godine proizvodnje, prosječne potrošnje i sličnih specifikacija



Slika 5.11 Izgled stranice s preporukom automobila

5.2. Ispitivanje rada aplikacije

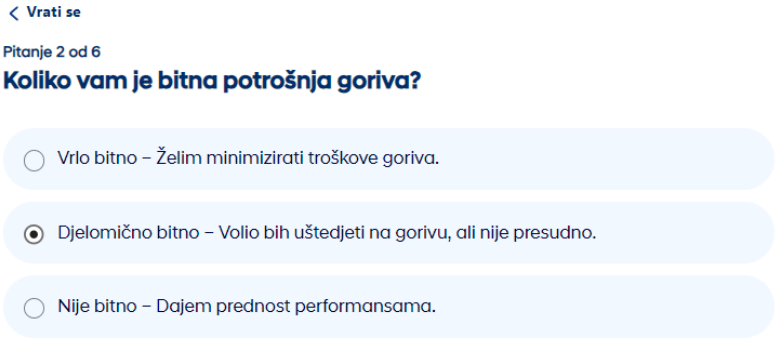
Ispitivanje rada aplikacije će pokazati ispunjavanje glavnih funkcionalnosti. Ispitivanje će biti provedeno tako što će se proći kroz upitnik s odgovorima i prikazivanjem automobila koji je preporučen.

5.2.1 Popunjavanje upitnika

Za početak potrebno je pokrenuti upitnik na početnoj stranici. Ispitat će se preporuka automobila koristeći specifičan slučaj. Korisnikov je budžet između 20 i 40 tisuća eura, te će se tako biti odgovoreno na prvo pitanje.

Slika 5.12 Odgovor na prvo pitanje

Nakon toga korisnik odlučuje da mu potrošnja goriva nije presudna, ali bi volio uštedjeti na gorivu ako je to moguće.



Slika 5.13 Odgovor na drugo pitanje

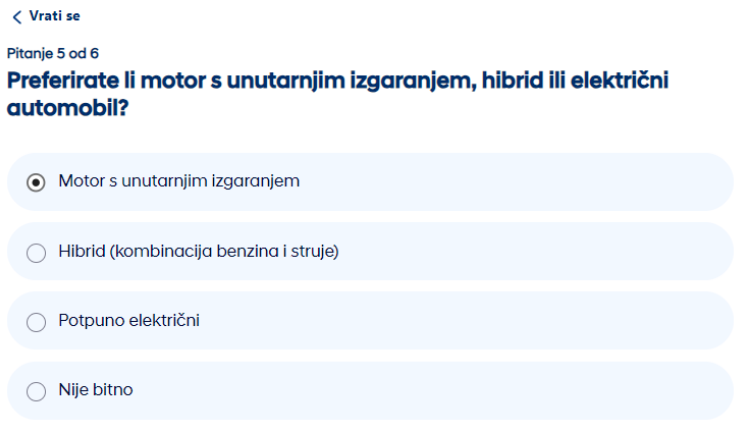
Na treće pitanje korisnik odgovara kako se najugodnije osjeća u limuzini te kako mu taj oblik najviše odgovara njegovim potrebama.

Slika 5.13 Odgovor na treće pitanje

Korisniku je potrebno mjesta do pet putnika te tako odgovara na četvrto pitanje.

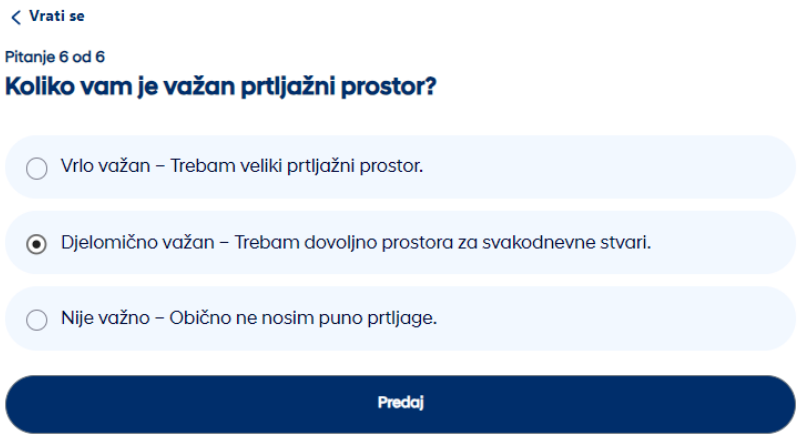
Slika 5.14 Odgovor na četvrto pitanje

Na petom pitanju korisnik odabire motor s unutarnjim izgaranjem.



Slika 5.15 Odgovor na peto pitanje

Na zadnje pitanje korisnik odabire odgovor da mu je potreban prtljažni prostor dovoljno velik za svakodnevne stvari.



Slika 5.16 Odgovor na šesto pitanje

5.2.2 Preporučeni automobili

Nakon popunjavanja upitnika, korisnik je dobio na izbor par automobila koji odgovaraju njegovim potrebama. Na slici 5.17 prikazana je Škoda Octavia čija je cijena otprilike 29 tisuća eura, prosječne potrošnje goriva 5.7 litara na 100 km, čiji je oblik karoserije limuzina, može prevoziti do 5 putnika, tip goriva je dizel te ima 590 litara prtljažnog prostora. Od dodatnih informacija prikazana je emisija ugljikovog dioksida, oznaka europskog standarda, maksimalna brzina u kilometrima po satu, te ubrzanje od 0 do 100 km/h izraženo u sekundama.

Slika 5.17 Preporučeni automobil

5.3. Analiza rezultata ispitivanja

Pri ispitivanju rada aplikacije, proveden je proces popunjavanja upitnika pri čemu je odgovaranjem na pitanja dobivena preporuka idealnog automobila. Prema odgovorima na upitnik, potreban je automobil cjenovnog ranga između 20 i 40 tisuća eura. Potrošnja goriva nije presudna, no traženi automobil treba biti limuzina koja može prevoziti do pet putnika. Zatraženo je da automobil bude s motorom s unutarnjim sagorijevanjem i dovoljno velik prostor za svakodnevne aktivnosti. Vodeći se odgovorima, dana preporuka je Škoda Octavia koja zadovoljava svaki uvjet i tim rezultatom ispitivanja potvrđeno je da aplikacija funkcionira.

6. ZAKLJUČAK

Web aplikacija za pomoć pri kupovini automobila pomoću višekriterijskog odlučivanja može biti izuzetno koristan alat za potrošače koji žele pronaći vozilo koje najbolje odgovara njihovim potrebama. Ona omogućuje pronalaženje automobila prema različitim kriterijima poput cijene, tipa vozila, performansi, potrošnje i drugih značajki. Izgrađena je u okolini Visual Studio Code, a za izradu korisničkog sučelja korišten je React.js i Redux. S druge strane, za izradu web aplikacije na poslužiteljskoj strani korišten je Firebase kako bi se omogućila jednostavna integracija baze podataka, autentifikacija i pohrana podataka. Web aplikacija je ispitana i za granične slučajeve korištenja koji pokazuju njenu točnost u radu.

Ova aplikacija ima potencijal značajno olakšati proces odlučivanja kod kupovine automobila, posebno za potrošače koji nemaju dovoljno znanja ili vremena da istražuju sve mogućnosti na tržištu. Višekriterijsko odlučivanje, koje aplikacija omogućuje, pomaže u stvaranju preporuka prema specifičnim potrebama korisnika, što dodatno pridonosi njenoj korisnosti. Aplikaciju je moguće unaprijediti omogućavanjem spremanja korisnikovog odabira što bi poboljšalo korisnikovo iskustvo, jer bi tako mogli pratiti različite vrste odabira i vlastite prethodne pretrage. Isto tako, pomoglo bi omogućavanje praćenja stvarnih ponuda automobila i dovođenja korisnika u kontakt s prodavačem. Time bi se privukao širi krug korisnika i povećala vrijednost i kvaliteta aplikacije.

LITERATURA

[1] Understanding the Car Buyer's Online Journey, ACVMAX, https://www.acvmax.com/blog/understanding-the-car-buyers-online-journey [3. 9. 2024.]

[2] J. Jablonska, Digitization in automotive retail in 2021 and beyond, McKinsey & Company, 2021,https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/digitization-in-automotive-retail-in-2021-and-beyond [3.9.2024.]

[3] Global Online Car Buying Market- Industry Dynamics, Market Size, And Opportunity Forecast to 2032, Astute Analytica, 2024.,https://www.astuteanalytica.com/industry-report/online-car-buying-market [10. 7. 2024.]

[4] How To Read The Stars, Euro NCAP, https://www.euroncap.com/en/about-euro-ncap/ [10. 7. 2024.]

[5] Web aplikacija Njuškalo, https://www.njuskalo.hr [10.9.2024.]

[6] Web aplikacija carwow, https://www.carwow.co.uk/buy-a-car [10. 7. 2024.]

[7] Web aplikacija Chiriboo, https://chiriboo.hr [10. 7. 2024.]

[8] Web aplikacija Cinch, https://www.cinch.co.uk [10. 7. 2024.]

[9] Web aplikacija Parkers, https://www.parkers.co.uk [10. 7. 2024.]

[10] Non-functional Requirements in Software Engineering [online], Geeks for Geeks, dostupno na: https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/ [20. 8. 2024.]

[11] M. Abdulhafeez et al. , Prioritizing Non-Functional Requirements in Agile Process Using Multi Criteria Decision Making Analysis, Pujnab India, March 15, 2023, pp. 24631 – 24634. [20. 8. 2024.]

[12] S. Ghosh, et al. ,An integrated multi-criteria decision-making and multivariate analysis towards sustainable procurement with application in automotive industry, Supply Chain Analytics, 2023, pp. 1-4. [20.8.2024.]

[13] J. Niederst Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, O'Reilly Media, 2012. pp. 58. [20. 8. 2024.]

[14] What Is SCSS? A Beginner's Guide for Developers, dostupno na: https://www.upwork.com/resources/what-is-scss [15. 8. 2024.]

[15] About Bootstrap, dostupno na: https://getbootstrap.com/docs/4.1/about/overview/ [15.8.2024.]

[16] A Trip Back in Time: the History of JavaScript, dostupno na: https://softteco.com/blog/history-of-javascript?WPACRandom=1724156634471 [15.8. 2024.]

[17] React Introduction , dostupno na: https://www.geeksforgeeks.org/reactjs-introduction/ [15.8. 2024.]

[18] Redux Essentials, Part 1: Redux Overview and Concepts, dostupno na: https://redux.js.org/tutorials/essentials/part-1-overview-concepts [15. 8. 2024.]

[19] Visual Studio Code, dostupno na: https://blog.cloudanalogy.com/10-features-of-vs-code-every-developer-should-know/ [15. 8. 2024.]

SAŽETAK

Cilj ovog završnog rada je izrada web aplikacije za pomoć pri kupovini automobila korištenjem višekriterijskog odlučivanja. Razvijena web aplikacija korisniku omogućuje registraciju i prijavu, odgovaranje na upitnik s pitanjima prilagođena korisniku neovisno o znanju o automobilima, te stvaranje preporuke automobila na osnovu odgovora, odnosno višekriterijskog odlučivanja. Korišten je programski jezik JavaScript, te React.js i Redux biblioteke.Tom kombinacijom tehnologija omogućeno je stvaranje interaktivnog korisničkog sučelja. Za bazu podataka korišten je Firebase Cloud. Aplikacija je ispitana za različite korisničke slučajeve, a rezultati ispitivanja pokazuju njenu učinkovitost u stvaranju preporuka korisniku i time olakšava proces kupovine automobila.

Ključne riječi: prodaja automobila, stvaranje preporuka, višekriterijsko odlučivanje, web aplikacija.

ABSTRACT

Title: Web application with a multi-criteria decision-making system and creation of recommendations for assistance in buying a car

The aim of the final work is a web application for assistance in buying a car using multi-criteria decision making. The developed web application allows the user to register and log in, answer a questionnaire with questions adapted to the user regardless of automotive knowledge, and create a car recommendation based on the answers. The JavaScript programming language was used, along with the React.js and Redux libraries. This combination of technologies enables the creation of an interactive user interface. Firebase Cloud was used for the database. The application has been tested for various cases and the test results show its effectiveness in creating recommendations for the user and thereby facilitating the process of buying a car.

Keywords: car, multi-criteria decision making, questionnaire, web application.

ŽIVOTOPIS

Emanuel Berišić rođen je 30. listopada 2002. godine u Osijeku. Pohađao je osnovnu školu Retfala u Osijeku, a nakon toga upisuje Isusovačku klasičnu gimnaziju s pravom javnosti u Osijeku. Nakon srednjoškolskog obrazovanja upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, preddiplomski sveučilišni studij Računarstva, smjer Računalno inženjerstvo.

PRILOZI

Prilog 1. Završni rad u formatu docx.

Prilog 2. Završni rad u formatu PDF.

Prilog 3. Programski kod web aplikacije