

# Web aplikacija za preporučivanje i praćenje treninga i prehrane sportaša postupcima strojnog učenja

---

**Ehman, Marko**

**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://um.nsk.hr/um:nbn:hr:200:350887>

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

*Download date / Datum preuzimanja:* **2024-11-24**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni preddiplomski studij**

**WEB APLIKACIJA ZA PREPORUČIVANJE I PRAĆENJE  
TRENINGA I PREHRANE SPORTAŠA POSTUPCIMA  
STROJNOG UČENJA**

**Završni rad**

**Marko Ehman**

**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**

<b>Ime i prezime pristupnika:</b>	Marko Ehman
<b>Studij, smjer:</b>	Sveučilišni prijediplomski studij Računarstvo
<b>Mat. br. pristupnika, god.</b>	R4641, 27.07.2021.
<b>JMBAG:</b>	0165091093
<b>Mentor:</b>	prof. dr. sc. Goran Martinović
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Web aplikacija za preporučivanje i praćenje treninga i prehrane sportaša postupcima strojnog učenja
<b>Znanstvena grana završnog rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rada:</b>	U teorijskom dijelu završnog rada potrebno je proučiti i opisati postupke i izazove vezane uz pripreme sportaša koji se bave trčanjem i biciklizmom s naglaskom na treniranje i prehranu. Na temelju analize stanja u području i postojećih sličnih rješenja, potrebno je predložiti vrste, dinamiku i raspored treninga za trčanje i biciklizam, te plan prehrane uzimajući u obzir osobni profil korisnika uključujući njegovu raspoloživost, kondiciju i ograničenja. Nadalje, treba predložiti načine praćenja treninga, prehrane i stanja sportaša, te mogućnosti poboljšanja. Na temelju navedenih smjernica, treba definirati funkcionalne i
<b>Datum prijedloga ocjene završnog rada od strane mentora:</b>	03.09.2024.
<b>Prijedlog ocjene završnog rada od strane mentora:</b>	Izvrstan (5)
<b>Datum potvrde ocjene završnog rada od strane Odbora:</b>	11.09.2024.
<b>Ocjena završnog rada nakon obrane:</b>	Izvrstan (5)
<b>Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:</b>	11.09.2024.



**FERIT**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

## IZJAVA O IZVORNOSTI RADA

Osijek, 11.09.2024.

**Ime i prezime Pristupnika:**

Marko Ehman

**Studij:**

Sveučilišni prijediplomski studij Računarstvo

**Mat. br. Pristupnika, godina upisa:**

R4641, 27.07.2021.

**Turnitin podudaranje [%]:**

13

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za preporučivanje i praćenje treninga i prehrane sportaša postupcima strojnog učenja**

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

<b>1. UVOD</b> .....	1
<b>1.1. Zadatak završnog rada</b> .....	2
<b>2. STANJE U PODRUČJU PREPORUČIVANJA I PRAĆENJA TRENINGA I PREHRANE SPORTAŠA</b> .....	3
<b>2.1. Prehrana sportaša</b> .....	3
<b>2.1.1. Unos nutrijenata</b> .....	3
<b>2.1.2. Hidracija</b> .....	4
<b>2.1.3. Planiranje prehrane</b> .....	4
<b>2.2. Metode treninga sportaša</b> .....	5
<b>2.2.1. Trening izdržljivosti</b> .....	5
<b>2.2.2. Intervalni trening</b> .....	5
<b>2.2.3. Mišićna snaga</b> .....	5
<b>2.3. Pripremljenost sportaša</b> .....	6
<b>2.3.1. Testovi kondicije</b> .....	6
<b>2.3.2. Biomehanička analiza</b> .....	6
<b>2.4. Prikaz stanja u području prehrane i treninga sportaša</b> .....	6
<b>2.4.1. Izazovi u preporučivanju i praćenju treninga i prehrane sportaša</b> .....	7
<b>2.4.2. Postojeća slična rješenja</b> .....	7
<b>3. MODEL, GRAĐA I POSTUPCI STROJNOG UČENJA U WEB APLIKACIJI</b> 12	
<b>3.1. Model web aplikacije</b> .....	12
<b>3.1.1. Funkcionalni zahtjevi</b> .....	12
<b>3.1.2. Nefunkcionalni zahtjevi</b> .....	13
<b>3.1.3. Slijed korištenja funkcionalnosti</b> .....	13
<b>3.2. Građa web aplikacije</b> .....	14
<b>3.2.1. Sastavni dijelovi aplikacije</b> .....	15
<b>3.2.2. Oblikovni obrazac MVC</b> .....	15
<b>3.3. Strojno učenje i korišteni klasifikacijski algoritmi</b> .....	16
<b>3.3.1. Vrste strojnog učenja</b> .....	16
<b>3.3.2. Korišteni klasifikacijski algoritmi</b> .....	17
<b>4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE I MODEL STROJNOG UČENJA</b> .....	19
<b>4.1. Korištene programske tehnologije, alati i jezici</b> .....	19

4.1.1. HTML .....	19
4.1.2. CSS i Bootstrap .....	19
4.1.3. JavaScript .....	20
4.1.4. Python .....	20
4.1.5. Django .....	20
4.1.6. Visual Studio Code .....	20
4.2. Alati za strojno učenje i repozitorij s podacima .....	21
4.2.1. Jupyter Notebook .....	21
4.2.2. Kaggle .....	21
4.3. Razvoj klasifikacijskog modela .....	22
4.4. Programsko rješenje na strani korisnika .....	27
4.4.1. Programsko rješenje registracije i prijave korisnika .....	27
4.4.2. Programsko rješenje testiranja razine fizičke spremnosti .....	29
4.4.3. Programsko rješenje unosa prehrane i treninga .....	30
4.5. Programsko rješenje na strani poslužitelja .....	32
4.5.1. Postupak stvaranja preporuka .....	32
4.5.2. Postupak praćenja treninga i prehrane .....	34
5. PRIKAZ RADA APLIKACIJE S ISPITIVANJEM I ANALIZOM .....	38
5.1. Korištenje aplikacije .....	38
5.2. Postavke ispitivanja web aplikacije i slučajevi korištenja .....	39
5.2.1. Postavke ispitivanja .....	39
5.2.2. Prvi korisnički slučaj .....	39
5.2.2. Drugi korisnički slučaj .....	40
5.2.3. Treći korisnički slučaj .....	41
5.3. Rezultati ispitivanja s analizom .....	43
6. ZAKLJUČAK .....	45
7. LITERATURA .....	46
ŽIVOTOPIS .....	49
SAŽETAK .....	50
ABSTRACT .....	51
PRILOZI .....	52

## 1. UVOD

Usklađivanje intenziteta i količine treninga te potrebnog unosa energije jedan je od najvažnijih čimbenika uspješnosti sportaša. Poveznica između treninga i prehrane postoji već nekoliko stoljeća. Termin „trening“ definira se kao postupak obavljanja fizičke aktivnosti i konzumiranja prehrane koja pripomaže u pripremanju za sportski događaj. Svaki sportaš prilagođavat će vrstu i intenzitet treninga prema svojim predispozicijama i mogućnostima. Isto tako, na temelju svoje fizičke spremnosti, indeksa tjelesne mase i medicinskih ograničenja nastojat će prilagoditi svoju prehranu kako bi zadovoljio idealan unos energije za potrebe treninga. Danas, s napretkom tehnologije, praćenje i preporuka treninga postaju neizostavan dio svakodnevnog života sportaša. Precizno praćenje učinka i napretka kroz treninge, uz istovremeno praćenje unosa energije, ključno je za postizanje optimalnih rezultata. Biciklizam je jedan od najzastupljenijih rekreacijskih sportova koji se uglavnom prakticira u svrhu zabave, turizma ili zdravlja. S trčanjem, predstavlja jedan od najčešćih oblika ljudske rekreacije te kardio treninga. Kako bi sportaši imali što bolji uvid u njihovu izvedbu treninga, energetske potrošnje te dnevni unos kalorija, potrebno im je omogućiti jednostavan i interaktivan način praćenja plana treninga i prehrane. Kao rješenje ovakvog problema nameće se aplikacija koja će korisniku, na temelju njegove fizičke spremnosti i želje za napretkom, preporučiti plan treninga i prehrane te opciju praćenja napretka korisnika kroz cijeli proces.

Web aplikacija razvijena u ovome radu omogućit će korisniku brzo, jednostavno i interaktivno praćenje plana prehrane i treninga. Putem aplikacije zahtjeva se unos raznih parametara na temelju kojih će se postupkom strojnog učenja korisniku preporučiti proces treniranja i konzumiranja prehrane. Također, svaki korisnik će na osnovu svojih preferencija imati opciju izabrati trčanje ili biciklizam kao sport kojim će se baviti tijekom procesa treniranja. Aplikacija će nuditi opciju praćenja napretka kako bi korisnik u svakome trenutku znao u kojoj fazi procesa se nalazi.

U drugome poglavlju opisano je stanje u području biciklista i trkača te su prikazana slična postojeća programska rješenja. Treće poglavlje prikazuje model i građu aplikacije te postupke strojnog učenja koji su korišteni u svrhe preporuka. Kroz četvrto poglavlje objašnjeno je programsko rješenje web aplikacije i model strojnog učenja, a u petom poglavlju obrađuje se rad web aplikacije s ispitivanjem i analizom rezultata.

## **1.1. Zadatak završnog rada**

U teorijskom dijelu završnog rada potrebno je proučiti i opisati postupke i izazove vezane uz pripreme sportaša koji se bave trčanjem i biciklizmom s naglaskom na treniranje i prehranu. Na temelju analize stanja u području i postojećih sličnih rješenja, potrebno je predložiti vrste, dinamiku i raspored treninga za trčanje i biciklizam, te plan prehrane uzimajući u obzir osobni profil korisnika uključujući njegovu raspoloživost, kondiciju i ograničenja. Nadalje, treba predložiti načine praćenja treninga, prehrane i stanja sportaša, te mogućnosti poboljšanja. Na temelju navedenih smjernica, treba definirati funkcionalne i nefunkcionalne zahtjeve, model, arhitekturu i dizajn web aplikacije na strani korisnika i poslužitelja. Također, treba predložiti odgovarajuće postupke strojnog učenja, kao i skupove podataka za treniranje modela, postupak raspoređivanja koji će omogućiti planiranje, praćenje i korigiranje dinamike treninga, te postupak preporučivanja poželjne prehrane. Web aplikaciju s modelom strojnog učenja treba ostvariti koristeći prikladne programske jezike, tehnologije i razvojne okvire, te ga ispitati na odgovarajućim ulaznim podacima i slučajevima korištenja za navedene sportove.



## **2. STANJE U PODRUČJU PREPORUČIVANJA I PRAĆENJA TRENINGA I PREHRANE SPORTAŠA**

U ovom poglavlju daje se naglasak na važnost prehrane kod sportaša i analiziraju se različite metode treninga. Također, prikazat će se načini pripreme sportaša i analiza njihove spremnosti te ukazati na izazove prilikom praćenja i preporučivanja treninga i prehrane uz nekoliko primjera postojećih tehnoloških rješenja ovakve problematike.

### **2.1. Prehrana sportaša**

Dobar prehrambeni plan neophodan je za postizanje i održavanje optimalne atletske izvedbe. Iz tog razloga većina sportaša odlučuje se za suradnju s profesionalnim nutricionistima koji im pomažu u planiranju ishrane. Pored dnevnih planiranja obroka, nutricionisti posvećuju posebnu pozornost potrebama sportaša prije, tijekom i poslije treninga i natjecanja. Prema [2], neke od glavnih potreba koje sportaši moraju zadovoljiti prehranom su : dovoljan i precizan unos nutrijenata (bjelančevine, masti i ugljikohidrati), precizno vrijeme i frekvencija obroka, konzumiranje određenih količina prehrambenih dodataka (vitamini, minerali,...) i hidracija.

#### **2.1.1. Unos nutrijenata**

Pravilna sportska prehrana temelji se na raznovrsnoj konzumaciji hrane koja uključuje dovoljnu količinu ugljikohidrata, masti, bjelančevina, vitamina, minerala i vode. Pridržavajući se pravila o omjerima ovih kategorija hrane sportaši zadovoljavaju potrebnu količinu hranjivih sastojaka i energije kako bi uspješno obavljali treninge i natjecali se. Prema [2], u pogledu odnosa pojedinih sastavnih dijelova preporučuje se odnos : ugljikohidrati : masti : bjelančevine 4:1:1, a kod znatnijeg povišenog metabolizma omjer 5:1:1. Tjelesna aktivnost, a osobito vježbe izdržljivosti, zahtijevaju postojanu opskrbu energijom u količini koja je u skladu s potrošnjom, pa među makronutrijentima posebno mjesto imaju ugljikohidrati [1]. Ugljikohidrati se uglavnom koriste pri kraćim tjelesnim aktivnostima i glavni su izvor energije. Dnevne preporuke unosa ugljikohidrata za sportaše iznose 6-10 g/kg [1]. Neke od namirnica koje predstavljaju glavni izvor ugljikohidrata su voće, žitarice, kruh i mlijeko. Primarna uloga prehrambenih proteina je korištenje u različitim anaboličkim procesima u tijelu. Kao rezultat toga, mnogi sportaši i treneri vjeruju da trening visokog intenziteta stvara veću potrebu za proteinima. Ovo proizlazi iz ideje da ako bi više proteina ili aminokiselina bilo dostupno mišićima koji vježbaju, to bi poboljšalo sintezu proteina. Manjak bjelančevina u

prehrani izaziva teške posljedice: gubi se mišićna masa, teško zacjeljuju rane, smanjuju se organi, opada radna sposobnost, opadanje otpornosti prema infekcijama. Masti predstavljaju drugi izvor energije koji dovodi tijelu duplo više energije od ugljikohidrata. Masti su odlično gorivo (daju dvostruko više energije od ugljikohidrata i bjelančevina), imaju zaštitnu ulogu (štite organe od hladnoće), daju osjećaj sitosti (3h treba da se evakuiraju iz želudca u tanko crijevo) [2]. Preporučeni unos masti iznosi 20-35% ukupnog dnevnog energetskeg unosa.

### **2.1.2. Hidracija**

Kao što se pridržavaju plana prehrane, sportaši se moraju strogo pridržavati i plana rehidracije. Tijekom treninga koji traje dulje od 30-40 minuta, gubitak ugljikohidrata, povišena tjelesna temperatura i smanjenje volumena cirkulirajuće tekućine mogu biti važni čimbenici u izazivanju umora. Svima se njima može manipulirati unosom tekućine, ali najučinkovitiji sastav pića i optimalna količina tekućine ovisit će o individualnim okolnostima. Voda nije optimalna tekućina za konzumaciju tijekom vježbi izdržljivosti, a postoje uvjerljivi dokazi da su pića koja sadrže dodane supstrate i elektrolite učinkovitija u poboljšanju performansi. Stoga je preporuka da se tijekom vremenski dužih treninga poveća unos elektrolita i dodatnih ugljikohidrata koji su korisni pri zadržavanju tekućine u tijelu i poboljšavaju izvedbu treninga. Sportaši bi trebali pratiti koliko su znoja izgubili tijekom treninga te izračunati razliku tjelesne mase prije i poslije treninga. Tako će izračunati točnu količinu vode potrebnu za unos nakon treninga i uspješno će nadomjestiti gubitke.

### **2.1.3. Planiranje prehrane**

Planiranje prehrane vrlo je važan proces kojeg se sportaši moraju držati kako bi uspješno unosili raznovrsne i točne količine namirnica u organizam. Prilikom planiranja prehrane sportaši se moraju pridržavati nekoliko smjernica. Trebali bi jesti minimalno pet manjih obroka dnevno te paziti da ukupni dnevni energetskeg unos ne bude veći od ukupne dnevne potrošnje. Unos energije raspoređuje se po zadanim omjerima nutrijenata. Prema [6], poželjno bi bilo da svaki obrok sadrži minimalno jednu namirnicu od svake kategorije nutrijenata, a uz planiranje prehrane, vrlo je bitno i voditi plan hidracije kako bi se pravovremeno nadoknadili gubitci vode tijekom napornih treninga.

## **2.2. Metode treninga sportaša**

U ovom potpoglavlju prikazat će se glavne metode treninga sportaša s naglaskom na bicikliste i trkače. Baveći se ovim sportovima, svaki sportaš odlučuje želi li naglasak treninga staviti na kratke udaljenosti ili će se više posvetiti treningu izdržljivosti i dužim kilometražama. Obje metode treninga izvode se u kombinaciji s treninzima snage koji dodatno ojačavaju mišiće i zglobove.

### **2.2.1. Trening izdržljivosti**

Prema [7], trening izdržljivosti unaprjeđuje sposobnost mobilizacije i transporta masnih kiselina i glukoze u aktivne mišiće u kojima se njihovom oksidacijom generira energija potrebna za rad. Tijekom ovakve vrste treninga vrlo je važno paziti na otkucaje srca kako se ne bi prekoračio aerobni prag, jer u suprotnom ovakva vrsta treninga gubi smisao. Kod trkača ovakvi treninzi uglavnom traju do 180 min, a kod biciklista mogu potrajati i nekoliko sati.

### **2.2.2. Intervalni trening**

Izvođenje ovakve vrste treninga zahtjeva trčanje i vožnju bicikla visokim intenzitetom s periodima oporavka. Najčešće se odabire nekoliko intervala koji se odrađuju određeni broj minuta, a period između njih koristi se za oporavak. Otkucaji srca su tijekom intervalnog treninga visoki te sportaši uglavnom osjećaju velike napore. Cilj ovakvih treninga je razvoj anaerobne izdržljivosti, mentalne snage i brzine. Najčešća pogreška koju sportaši prave za vrijeme intervalnog treninga je pravljenje predugačke pauze između intervala.

### **2.2.3. Mišićna snaga**

Trening mišićne snage, odnosno trening dizanja utega svakako treba prakticirati s već navedenim tipovima treninga. Iako kod trkača i biciklista najveći dio posla obavljaju noge, nikako ne treba zapostaviti i gornji dio tijela. Kod trčanja je jako bitna ispravna forma kako bi se iskoristio sav potencijal, a pri ostvarivanju pravilnog trčanja pomaže jakost gornje dijela tijela. Isto tako, jačanje mišića trbuha i ruku poboljšat će stabilizaciju kralježnice, odnosno tehniku ruku i mogućnost zadržavanja istog tempa trčanja. Kod biciklista se trening mišićne snage uglavnom bazira na mišićima nogu, trbuha i donjeg dijela leđa. Kao glavna biciklistička vježba ističe se čučanj kojim se pokreću kukovi, koljena i skočni zglobovi. Kontinuiranim izvođenjem treninga snage može se ostvariti veliki napredak u trčanju i biciklizmu.

## **2.3. Pripremljenost sportaša**

Prema [7], konačan uspjeh sportaša najviše će ovisiti o njegovoj fizičkoj spremnosti. Zato je u sportu naglasak na fizičke treninge, uz pokoje mentalne treninge koji sportašima pomažu probijati mentalne barijere i ostvariti zacrtane ciljeve. Pripremljenost sportaša može se testirati brojnim kondicijskim testovima, a biomehanička analizom određuju se osnovni kinematički parametri strukture gibanja u sportu.

### **2.3.1. Testovi kondicije**

Jedan od najpoznatijih testova kondicije je Cooperov test. To je dvanaestominutni test trkačke kondicije koji na jednostavan način mjeri aerobnu kondiciju i procjenjuje maksimum VO<sub>2</sub>. Cooperov test zahtijeva od osobe koja se testira da trči ili hoda što je više moguće u razdoblju od 12 minuta. Cilj testa je izmjeriti maksimalnu udaljenost koju je pojedinac prešao tijekom dvanaestominutnog razdoblja i obično se provodi na stazi za trčanje postavljanjem čunjeva na različite udaljenosti kako bi se omogućilo mjerenje udaljenosti [10]. Nakon testiranja, vrlo je lako ocijeniti dobivene rezultate preko postojeće Cooperove tablice.

FTP (Functional Threshold Power) test je mjera koja se koristi za testiranje biciklističke spremnosti. Osim što se koristi za mjerenje biciklističke kondicije, ovim testom mogu se odrediti i metrike poput razine stresa tijekom vožnje ili faktor intenziteta. Ovaj je test jedan od najtežih za izvođenje za sportaše jer zahtijeva vožnju s najvećom održivom snagom 20 minuta. Rezultat FTP je 95% prosječne snage tijekom ovog intervala. Poznavajući svoj FTP, svaki biciklistički trening bit će optimiziran trenutnoj fizičkoj spremi biciklista [12].

### **2.3.2. Biomehanička analiza**

Skup postupaka za određivanje osnovnih kinematičkih i kinetičkih parametara strukture gibanja naziva se biomehanička analiza. Prema [13], ti se parametri mogu izraziti numerički te mogu biti vrlo korisni za određivanje karakteristika i kvalitete izvedbe strukture kretanja. Ova analiza nam pomaže u trenažnom procesu gdje dobivene rezultate analiziramo i koristimo za daljnje usavršavanje sportaševih sposobnosti.

## **2.4. Prikaz stanja u području prehrane i treninga sportaša**

Ovo potpoglavlje daje prikaz izazova koji se pojavljuju prilikom planiranja, preporučivanja i

praćenja prehrane i treninga kod sportaša. Također, prikazat će se i analizirati postojeća tehnološka rješenja koja su pokušala riješiti većinu spomenute problematike.

#### **2.4.1. Izazovi u preporučivanju i praćenju treninga i prehrane sportaša**

Praćenje i preporučivanje treninga i prehrane sportaša nije nimalo jednostavan proces te je popraćen brojnim izazovima. Prvi problem koji se nameće jesu individualne razlike između sportaša. Drugim riječima, svaki sportaš ima drugačije genetske predispozicije, razinu kondicije, zdravstvena ograničenja i probleme ili osobne ciljeve. Vrlo je teško napraviti univerzalan plan prehrane ili treninga jer će svaki pojedinac drugačije reagirati na istu vrstu treninga i prehrambeni režim zbog razlika u brzini rada metabolizma i drugih bioloških čimbenika. Drugi problem predstavljaju ozljede i promjene u fizičkoj formi i performansama sportaša. Ako dođe do takvih situacija, automatski se prehrana i treninzi moraju prilagođavati novonastalim uvjetima. Konačno, razna natjecanja, putovanja i natjecateljski raspored mogu prouzročiti teškoće pri održavanju optimalne razine treninga i ispravne prehrane. Napredak tehnologije uveliko je pomogao u praćenju intenziteta treniranja i napretka sportaša. Prema [13,14], neki od postojećih algoritama i metoda za preporuke su sustavi zasnovani na pravilima (engl. Rule-based Systems), sustavi zasnovani na kolaborativnom filtriranju, sustavi zasnovani na sadržaju te oni koji koriste metode strojnog i dubokog učenja. Sustavi zasnovani na pravilima koriste unaprijed definirana pravila i logiku za davanje preporuka, a sustavi zasnovani na kolaborativnom filtriranju podatke mnogih korisnika kako bi napravili preporuke za pojedinca. Prednost sustava zasnovanih na sadržaju je visoka razina personalizacije, a sustava koji koriste metode strojnog i dubokog učenja sposobnost prilagodbe i visoka preciznost. Sustav praćenja napretka treninga i unosa prehrane putem aplikacija ostvaruje se upotrebom IoT uređaja, snimanjem hrane putem kamere i računalnog vida, gamifikacijom i personalizacijom korisničkog iskustva te automatskim praćenjem unosa. Web aplikacija završnoga rada koristi metode strojnog učenja za preporuku i praćenje treninga i prehrane. Daljnji razvoj u ovom području vjerojatno će donijeti još sofisticiranije metode za poboljšanje zdravlja i performansi sportaša.

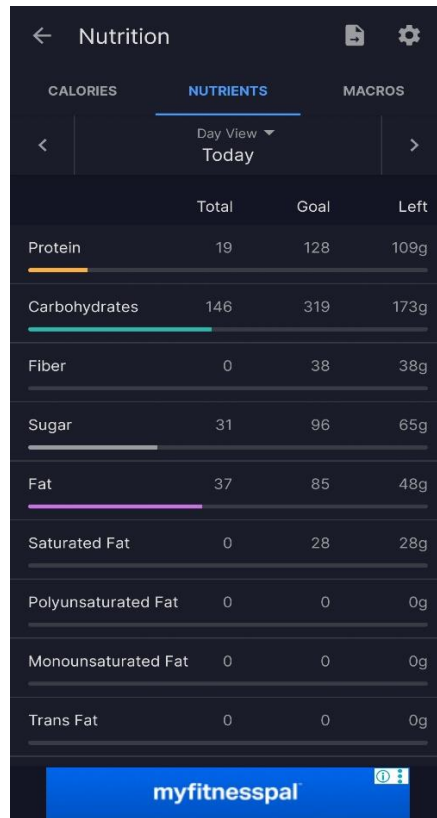
#### **2.4.2. Postojeća slična rješenja**

Kao što je već napomenuto, na tržištu postoje tehnološka rješenja koja pokušavaju riješiti problematiku preporučivanja i praćenja treninga i prehrane sportaša. U nastavku je dan opis takvih

aplikacija i analiza njihovih glavnih značajki.

#### **2.4.2.1. Mobilna aplikacija MyFitnessPal**

MyFitnessPal mobilna je aplikacija koju su 2005. godine osmislili Albert i Mike Lee. Aplikacija je dostupna na Android i iOS uređajima. Mike Lee napravio je aplikaciju jer mu je bilo potrebno pratiti svoje obroke provodeći prehrambeni režim prije svojega vjenčanja [15]. Postepenim razvojem aplikacija je dosegla oko 200 milijuna korisnika, što ju čini najpopularnijom aplikacijom za zdravlje i trening. Glavna zadaća aplikacije je omogućiti korisniku praćenje količine kalorijskog unosa kroz obroke i potrošnje preko treninga. Korisnici će se prilikom početka upotrebe aplikacije registrirati i napraviti svoj profil s pomoću kojega će pratiti svoje rezultate i napredak. Na temelju podataka o fizičkim predispozicijama, aplikacija svakom korisniku preporučuje dnevni unos kalorija kako bi ostvario zacrtane ciljeve u određenom periodu. S pomoću dnevnika prehrane i treninga, korisnici mogu bilježiti u aplikaciji svoje obroke i vježbe koje su odradili trenirajući. Aplikacija će na temelju podataka iz dnevnika ponuditi informacije o nutritivnom sastavu svakoga obroka te ukupan broj kalorija koji je utrošen za vrijeme treninga. Također, aplikacija pamti najčešće obroke i vrste treninga korisnika i tako olakšava dodavanje proizvoda u dnevnik prehrane. Još jedna od značajki je skeniranje QR koda proizvoda s pomoću kojeg aplikacija prepoznaje o kojoj se namirnici radi. Nadalje, ako korisnik zaboravi unijeti podatke o obroku dobit će obavijest od aplikacije koja ga podsjeća da unese potrebne informacije. Korisnici aplikaciju mogu koristiti besplatno, no ona pruža i dodatne značajke i usluge koje se mogu koristiti uz plaćanje. Na slici 2.1 prikazane su nutritivne vrijednosti koje je korisnik prehranom unio tijekom jednog dana.



Slika 2.1. Prikaz nutritivnih vrijednosti namirnica konzumiranih u jednome danu

#### 2.4.2.2. TrainingPeaks

TrainingPeaks pruža cjelovito web, mobilno i desktop rješenje za pametno i učinkovito praćenje treninga izdržljivosti. Trenutno je najpopularnija online i mobilna platforma za treniranje na svijetu. Predstavlja besplatni online dnevnik treninga koji korisnicima omogućuje učitavanje treninga, postavljanje ciljeva, dodavanje događaja i pregled sažetaka treninga. Kako bi se iskoristio sav potencijal aplikacije, omogućena je opcija pretplate od 10 dolara mjesečno s kojom se ostvaruje uvid u detaljniju analizu podataka o treningu i kondiciji [16]. Korisnici preko aplikacije mogu odabrati sport kojim se bave (trčanje, biciklizam, plivanje,...) i potražiti brojne planove treninga za taj tip sporta. Omogućena je i opcija traženja trenira kako bi sportaši potražili stručnu pomoć prilikom procesa pripreme za natjecanje ili optimizacije njihovih trening rutina. Aplikacija nudi uvid u brojne blogove o različitim temama preko kojih se korisnici mogu informirati i poboljšati svoje znanje o sportu i nutricionizmu. Na slici 2.2 nalazi se prikaz planova treninga za biciklizam koje korisnici mogu provoditi ako plate određenu svotu novca, a na slici 2.3 prikazani su blogovi s tematikom nutricionizma i zdrave prehrane sportaša.

## Cycling Training Plans

SHOP BY SPORT SUBTYPE:

Road MTB Gran Fondo/Century Cyclocross Time Trialing Base Indoor

---

### Find your Plan

Keyword Search

What are you training for?  
Cycling

Select Sport Subtype

Structured Plans Only [Learn more](#)

Price  
Any Price

Skill Level  
Any Skill Level

6,991 Results Sort By Best Sellers ↓

English

**10** weeks **6.9** hrs/week **4-5** workouts/week


**10 Week Weight loss, Improve VO2, FTP and W P/KG! Lose those pounds! Fat Burner W/12-week S&C Plan!**

Beginner to Competitive

★ 4.36 (74)

**BEST SELLER**

**\$35.00**



by Optimum Coaching

---

**4** weeks **4.7** hrs/week **4-5** workouts/week


**4 Week Intermediate FTP Boost Training Plan (Zwift Compatible)**

Intermediate to Competitive

★ 4.54 (190)

**BEST SELLER**

**\$12.50**




by Big Development Coaching

---

**3** weeks **5.9** hrs/week **4-9** workouts/week


**The Mini VO2 Max Squeeze & Stretch Plan**

**BEST SELLER**



Slika 2.2. Prikaz planova treninga za biciklizam

## Daily Nutrition





#DAILY-NUTRITION

### Balancing High-Carb Nutrition Needs with Dental Health

In the modern age of sports, high-carb nutrition plans are essential for performance, but they pose significant risks to dental ...

[Read Article](#)

 Taylor Warren





#DAILY-NUTRITION

### 4 Reasons You Should Eat More Fiber

No matter what your diet looks like, chances are you could use more fiber. Here's why fiber is crucial to ...

[Read Article](#)

 Phil White




#DAILY-NUTRITION

### The Dairy-Free Olympian

Dotsie Bausch credits her plant-based diet for helping her win an Olympic silver medal at 39 years old. Here's her ...

[Read Article](#)

 Dotsie Bausch

[More Daily Nutrition Articles](#)

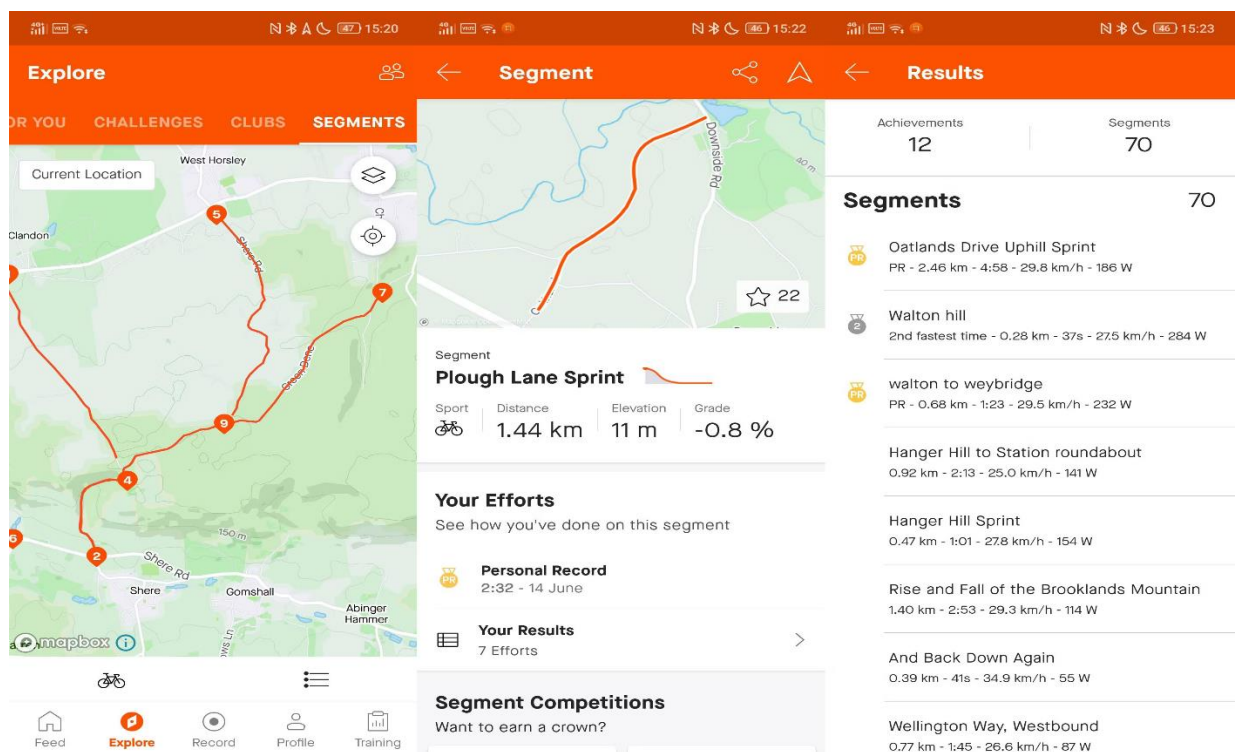
Slika 2.3. Prikaz blogova s tematikom nutricionizma i pravilne prehrane

### 2.4.2.3. Strava

Strava je on-line platforma i mobilna aplikacija koja omogućuje sportašima da prate i unose



bilješke o njihovim treninzima. Također, uključuje značajke kao što su tablice s najboljim rezultatima i osobnim rekordima. Strava omogućuje biciklistima, trkačima, plivačima i ostalim sportašima da na jednostavan način prate svoj napredak i povezuju se s drugim sportašima. Strava je osnovana 2009. godine i od tada je postala jedna od najpopularnijih platformi za praćenje biciklizma i trčanja. Trenutno ima više od 36 milijuna korisnika i koristi se u 195 zemalja. Aplikacija ima mogućnost sinkronizacija s većinom uređaja, kao što su telefon, GPS sat ili uređaj za mjerenje otkucaja srca. Predstavlja društvenu mrežu za sportaše jer pruža mogućnost dijeljenja informacija o aktivnosti s prijateljima i objavljivanja komentara na tuđe aktivnosti. Na slici 2.4 prikazani su detalji treninga, uključujući informacije o svakom segmentu pređene rute.



Slika 2.4. Prikaz detalja treninga s informacijama o svakom segmentu pređene rute

### **3. MODEL, GRAĐA I POSTUPCI STROJNOG UČENJA U WEB APLIKACIJI**

U ovom poglavlju prikazat će i opisati model web aplikacije, s funkcionalnim i nefunkcionalnim zahtjevima. Građa aplikacije predstaviti će se dijelovima aplikacije te arhitekturnim obrascem. Također, predstaviti će se osnovne strojnog učenja i korišteni klasifikacijski algoritmi.

#### **3.1. Model web aplikacije**

U ovom završnom radu web aplikacija ostvarit će se s pomoću obrasca MVC (Model-View-Controller) koji se često koristi u programskom inženjerstvu za odvajanje pojedinih dijelova aplikacije u komponente, ovisno o njihovoj namjeni. Za implementaciju komponenti na strani korisnika (engl. frontend) i na strani poslužitelja (engl. backend) koristit će se besplatni programski okvir Django, temeljen na Pythonu, uz HTML (HyperText Markup Language), Bootstrap i JavaScript. Za pohranu podataka koristit će se relacijska baza podataka SQLite koja je ugrađena unutar okvira Django. Funkcionalni i nefunkcionalni zahtjevi web aplikacije objašnjeni su u nastavku.

##### **3.1.1. Funkcionalni zahtjevi**

Aplikacija ima početnu stranicu na kojoj se nalaze osnovne informacije o sportovima biciklizmu i trčanju, ali i važnosti zdrave prehrane kod sportaša. Korisnici imaju mogućnost registriranja kako bi napravili osobni profil. Nakon registriranja, korisnik se treba prijaviti u aplikaciju koristeći svoje korisničko ime i lozinku. Kada se korisnik uspješno prijavi, ima mogućnost ispunjavanja obrasca na temelju kojeg se analizom podataka donosi zaključak o korisnikovoj fizičkoj spremnosti i svrstava ga u određenu kategoriju fizičke spremnosti. Svrstavanje korisnika u kategorije fizičke spremnosti implementirat će se uz pomoć metoda strojnog učenja i prethodno istreniranog modela. Tijekom ispunjavanja obrasca korisnik ima mogućnost odabrati želi se tijekom procesa treniranja baviti biciklizmom, trčanjem ili oboje. Ovisno o kategoriji kojoj korisnik pripada i njegovim preferencijama za napredak, bit će mu preporučen plan prehrane i treninga. Korisnici imaju stranicu osobnog profila na kojoj se nalaze osobni podaci korisnika, informacije o napretku, popis zajednica (klubova) kojima se korisnik pridružio i sl. Nadalje, korisnici imaju mogućnost objave informacija o obavljenom treningu koji će biti javno vidljiv njihovim prijateljima ili svim drugim korisnicima, a komentari ispod objave također će biti mogući. Omogućit će se slanje zahtjeva za

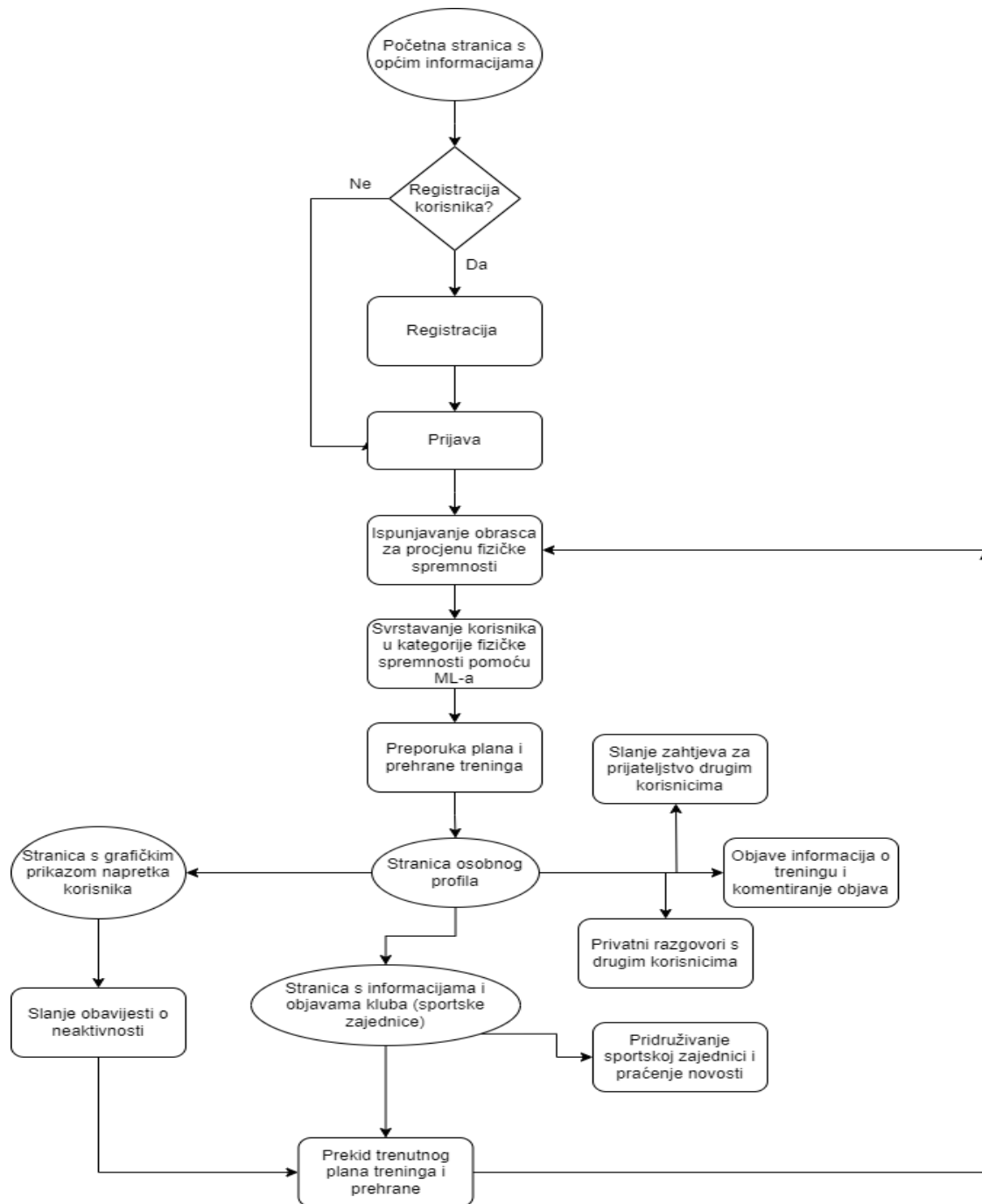
prijateljstvo kako bi korisnici mogli lakše izmjenjivati iskustva i dojmove. Također, postojat će i privatni razgovori preko kojih će korisnici privatno razgovarati jedni s drugima. Aplikacije također nudi pristup raznim sportskim zajednicama i klubovima koje će na svojem profilu moći objavljivati informacije o treninzima, natjecanjima i druženjima. Napredak korisnika moći će se pratiti preko grafičkog prikaza na kojem će se vidjeti rezultati treninga i kalorijski unos. Ako korisnik nije redovan i ne pridržava se plana, aplikacija će mu poslati obavijest kao podsjetnik. Korisnici će u svakom trenutku imati opciju prekida trenutnog plana te će ponovno moći popuniti obrazac i započeti s drugačijim treninzima i prehranom.

### **3.1.2. Nefunkcionalni zahtjevi**

Nefunkcionalni zahtjevi u programskom inženjerstvu odnose se na karakteristike programskog sustava koje nisu povezane s određenom funkcionalnošću ili ponašanjem. Oni opisuju kako bi sustav trebao raditi, a ne što bi trebao raditi [18]. Nefunkcionalni zahtjevi važni su jer mogu imati značajan utjecaj na ukupnu kvalitetu i uspjeh programskog sustava. Često su usko povezani s performansama, sigurnošću i upotrebljivošću sustava. Oni također pomažu osigurati da se sustav može održavati, prenositi i biti usklađen s relevantnim zakonima i propisima. U web aplikaciji završnog rada, performanse sustava pokušat će se dovesti na razinu da vrijeme odziva i učitavanja bude najviše nekoliko sekundi. Sigurnost će se osigurati autentifikacijom svakom korisnika koji će imati jedinstvene korisničke račune, a pojedinim podacima i dijelovima aplikacije pristupat će samo ovlaštene korisnici. Kod aplikacije bit će strukturiran tako da ga je lako održavati i proširivati. Prilagodljivost aplikacije uvidjet će se preko responzivnog dizajna koji će omogućiti funkcioniranje aplikacije na različitim uređajima i zaslonima. Jednostavno i intuitivno korisničko sučelje pružit će korisnicima lakšu navigaciju, pristupačnost i brže izvršavanje zadataka. Krajnje, modularnost aplikacije pružit će mogućnost dodavanja novih funkcionalnosti aplikacije bez ometanja postojećih.

### **3.1.3. Slijed korištenja funkcionalnosti**

Na slici 3.1 prikazan je slijed korištenja funkcionalnosti koje su opisane u prethodnim odlomcima.



Slika 3.1. Prikaz slijeda korištenja funkcionalnosti web aplikacije za preporučivanje plana treninga i prehrane

### 3.2. Građa web aplikacije

U ovom potpoglavlju opisat će se glavni dijelovi aplikacije te pobliže pojasniti oblikovni obrazac MVC i njegovu ulogu i primjenu u programskom inženjerstvu.

### 3.2.1. Sastavni dijelovi aplikacije

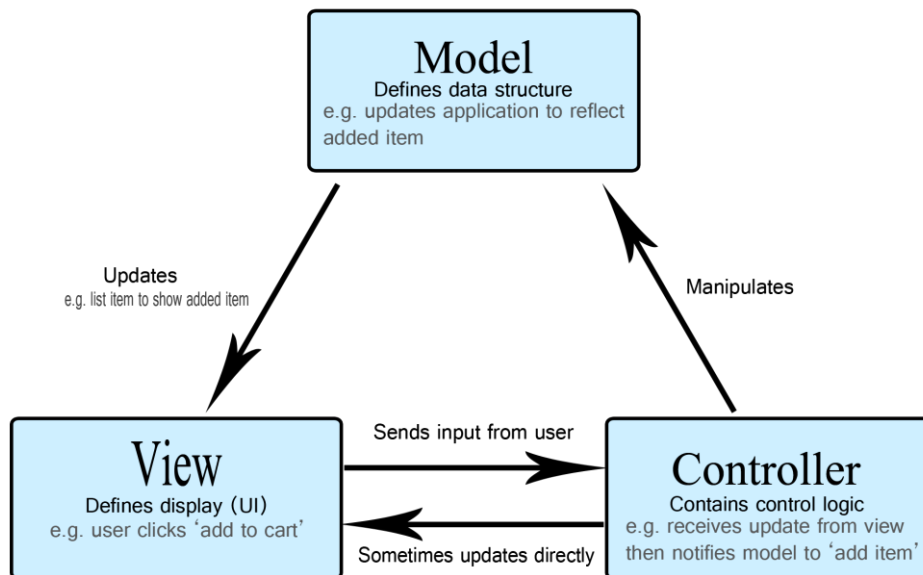
Web aplikacija za preporuku plana treninga i prehrane sportaša imat će jednostavnu i intuitivno korisničko sučelje. Ono će sadržavati komponente poput navigacijske trake s pomoću koje će korisnici lako navigirati kroz aplikaciju, razne forme za unos podataka o treninzima i prehrani, ali i grafikone koji će vizualizirati i olakšati praćenje napretka. Za razvoj dijela aplikacije na strani poslužitelja koristit će se modeli (eng. „models“) za rukovanje logikom podataka i oblikovanje baze podataka, pogledi (eng. „views“) koji će upravljati logikom aplikacije i funkcionalnostima, a predlošci (eng. „templates“) će oblikovati strukturu dijela aplikacije vidljivu korisniku. Svi podaci pohranjivat će se unutar relacijske baze podataka.

### 3.2.2. Oblikovni obrazac MVC

MVC (Model-View-Controller) je obrazac u dizajnu programskog rješenja koji se obično koristi za implementaciju korisničkih sučelja, podataka i logike upravljanja [19]. Naglašava razdvajanje poslovne logike programa i prikaza podataka i informacija. Ovo "odvajanje briga" omogućuje bolju podjelu rada i bolje održavanje. Neki drugi obrasci dizajna temelje se na MVC-u, poput MVVM (Model-View-ViewModel), MVP (Model-View-Presenter) i MVW (Model-View-Whatever). Tri glavne komponente obrasca MVC mogu se opisati na sljedeći način:

1. Model: upravlja podacima i poslovnom logikom.
2. Pogled (eng. „View“): upravlja izgledom i prikazom.
3. Kontroler (eng. „Controller“): usmjerava naredbe do dijelova modela i prikaza

U primjeru web aplikacije koja se prezentira u ovom završnom radu obrazac MVC koristit će se u obliku MVT (Model-View-Template). U ovakvoj arhitekturi pogled zamjenjuje kontrolera, a predlošci predstavljaju poglede. Po uzoru na [19], na slici 3.2 prikazana je arhitektura MVC obrasca.



Slika 3.2. Prikaz arhitekture i funkcionalnosti oblikovnog obrasca MVC

### 3.3. Strojno učenje i korišteni klasifikacijski algoritmi

U ovom potpoglavlju opisat će se vrste strojnog učenja i istaknuti njihove karakteristike i razlike. Prikazat će se i način rada klasifikacijskih algoritama korištenih prilikom treniranja modela za raspoređivanje korisnika u kategorije fizičke spremnosti.

#### 3.3.1. Vrste strojnog učenja

Prema [20], zadatak algoritma strojnog učenja je pronaći prirodne uzorke i poveznice u podacima te na temelju toga steći uvid i zatim odlučiti i predviđati. Osnovna podjela strojnog učenja može se svesti na nadzirano i nenadzirano učenje. Kod nadziranog učenja algoritmi rade s poznatim skupom ulaznih podataka („značajke“) i s poznatim skupom izlaznih podataka („oznaka“). Cilj je pronaći funkciju, odnosno model, koji opisuje odnos između ulaznih i izlaznih vrijednosti. Postoje klasifikacijski i regresijski algoritmi nadziranog učenja, ovisno o tome predviđa li se diskretna vrijednost ili kontinuirana. Primjeri klasifikacijskih algoritama su: logistička regresija, stablo odlučivanja, K-najbližih susjeda i stroj potpornih vektora. Najpoznatiji primjer regresijskog problema je linearna regresija. Nenadzirano učenje ne sadrži oznake, tj. izlazne varijable. Cilj nenadziranog učenja je otkriti strukturu podataka, odnosno grupirati slične elemente u nakupine

(engl. Clusters) ili smanjiti dimenzionalnost sustava. Algoritam k-means najpoznatiji je i najjednostavniji algoritam grupiranja podataka.

### 3.3.2. Korišteni klasifikacijski algoritmi

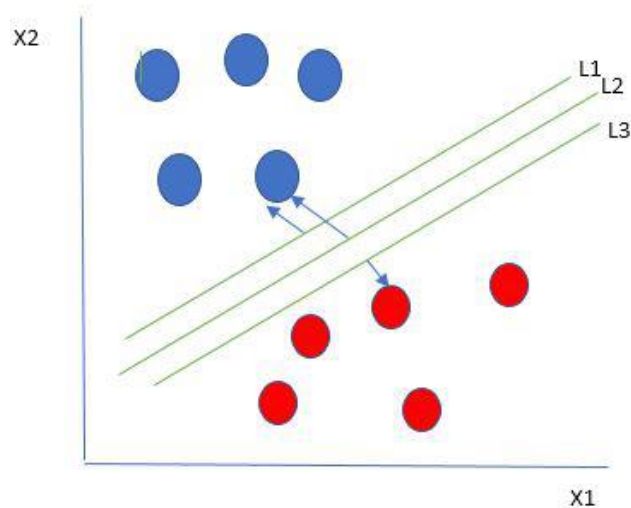
Prema [20], algoritmi strojnog učenja koji se najčešće koriste za rješavanje klasifikacijskih problema su logistička regresija, algoritam k-najbližih susjeda i stroj potpornih vektora.

Logička regresija [21] analizira odnos između jedne ili više neovisnih varijabli i klasificira podatke u diskretne klase. Model daje binarni ili dihotomni ishod ograničen na dva moguća ishoda: da/ne, 0/1 ili točno/netočno. Opsežno se koristi u prediktivnom modeliranju, gdje model procjenjuje matematičku vjerojatnost pripada li instanca određenoj kategoriji ili ne. Logistička regresija koristi logističku funkciju koja se naziva sigmoidna funkcija za mapiranje predviđanja i njihovih vjerojatnosti. Sigmoidna funkcija odnosi se na krivulju u obliku slova S koja pretvara bilo koju stvarnu vrijednost u raspon između 0 i 1. Na primjer, ako je izlaz sigmoidne funkcije iznad 0,5, izlaz se smatra 1. S druge strane, ako je izlaz je manji od 0,5, izlaz je klasificiran kao 0. Glavne prednosti logističke regresije su: jednostavnost implementacije, prikladna za linearno odvojive skupove podataka i mogućnost vrednovanja relevantnosti svakog prediktora.

Algoritam k-najbližih susjeda (eng. „k-Nearest Neighbor“) [22] jedan je od najjednostavnijih algoritama strojnog učenja koji se temelji na tehnici nadziranog učenja. k-NN je neparametarski algoritam, što znači da ne donosi nikakve pretpostavke o temeljnim podacima. Također se naziva algoritmom „lijenog učenja“ jer ne uči odmah iz skupa za trening, već pohranjuje skup podataka i u vrijeme klasifikacije izvodi radnju na skupu podataka. k-NN algoritam radi pronalaženjem k najbližih susjeda danoj podatkovnoj točki na temelju metrike udaljenosti, kao što je Euklidova udaljenost. Klasa ili vrijednost podatkovne točke tada se određuje većinom glasova ili prosjekom k susjeda. Ovaj pristup omogućuje algoritmu da se prilagodi različitim obrascima i napravi predviđanja na temelju lokalne strukture podataka.

SVM (eng. „Support Vector Machine“) [23] moćan je algoritam strojnog učenja koji se koristi za zadatke linearne ili nelinearne klasifikacije, regresije, pa čak i otkrivanja iznimnih vrijednosti (eng. „outliers“). Primjenjuje se u različitim područjima, poput klasifikacije teksta, klasifikacije slika, otkrivanja neželjene pošte, identifikacije rukopisa, analize ekspresije gena, prepoznavanja lica i otkrivanja anomalija. Glavni cilj SVM algoritma je pronaći optimalnu hiper-ravninu u N-

dimenzionalnom prostoru koja može razdvojiti podatkovne točke u različitim klasama unutar prostora značajki. Hiperravnina nastoji maksimizirati marginu (eng. „margin“) između najbližih točaka različitih klasa. Dimenzija hiper-ravnine ovisi o broju značajki. Na slici 3.3 prikazan je proces maksimiziranja udaljenosti margine od najbližih točaka različitih klasa.



Slika 3.3. Prikaz procesa maksimiziranja margine u algoritmu SVM

Prema [20], najčešće korištene metrike za vrednovanje klasifikacijskih algoritama su točnost, preciznost, odziv i F1 rezultat. Točnost je najčešće korištena mjera u svakodnevnoj analizi modela. Definiira se kao omjer broja točno predviđenih slučajeva u odnosu na ukupan broj predviđenih slučajeva. Preciznost je mjera koja prikazuje omjer istinitih pozitivnih rezultata u odnosu na ukupan broj pozitivnih rezultata koje model predviđa. Odziv, poznat i kao stopa istinitih pozitivnih rezultata (engl. True Positive Rate), mjeri koliko je model uspješan u pronalaženju svih pozitivnih primjera. Ova mjera je važna kada je cilj minimizirati broj lažno negativnih rezultata. F1 rezultat je harmonijska sredina preciznosti i odziva, te je koristan u situacijama gdje je važno balansirati između ta dva aspekta modela.



## **4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE I MODEL STROJNOG UČENJA**

U ovome poglavlju bit će opisane tehnologije, alati i programski jezici korišteni pri izradi web aplikacije. Također, prikazat će se razvoj klasifikacijskog modela te programska rješenja koja su implementirana na korisničkoj i poslužiteljskoj strani aplikacije.

### **4.1. Korištene programske tehnologije, alati i jezici**

U ovome potpoglavlju opisat će se tehnologije, alati i jezici koji su korišteni tijekom izrade web aplikacije završnoga rada.

#### **4.1.1. HTML**

HTML [24] je kratica za HyperText Markup Language. To je standardni označni jezik za izradu web stranica. Omogućuje stvaranje i strukturu odjeljaka, odlomaka i poveznica korištenjem HTML elemenata (građevinskih blokova web stranice) kao što su oznake i atributi. Programeri koriste HTML kod za dizajniranje načina na koji preglednik prikazuje elemente web stranice, kao što su tekst, hiperveze i medijske datoteke. Korisnici se mogu lako kretati i umetati veze između povezanih stranica i web-mjesta budući da se HTML često koristi za umetanje hiperveza. Isto tako, HTML omogućuje organiziranje i oblikovanje dokumenata, slično Microsoft Wordu.

#### **4.1.2. CSS i Bootstrap**

CSS [25] (Cascading Style Sheets) stilski je jezik za određivanje načina na koji se dokumenti prikazuju korisnicima — kako su stilizirani, postavljeni itd. Promjena boje i veličine odlomaka i hiperveza, posebni efekti i animacije, različiti fontovi i ostale vizualne karakteristike definiraju se s pomoću CSS-a.

Bootstrap [26] je besplatna zbirka alata otvorenog koda koja se koristi za izradu responzivnih web stranica i web aplikacija. To je najpopularniji HTML, CSS i JavaScript okvir za razvoj responzivnih, mobilnih web stranica. Danas su web stranice izrađene s pomoću Bootstrapa kompatibilne sa svim preglednicima (IE, Firefox i Chrome) i sa svim veličinama zaslona (stolna računala, tableti i telefoni).

### **4.1.3. JavaScript**

JavaScript [27] lagani je programski jezik koji obično koriste web programeri za dodavanje dinamičkih interakcija web stranicama, aplikacijama, poslužiteljima, pa čak i igrama. Radi besprijekorno s HTML-om i CSS-om, nadopunjujući CSS u oblikovanju HTML elemenata, a istovremeno pruža korisničku interakciju, mogućnost koja nedostaje samom CSS-u. Široko rasprostranjene primjene JavaScripta u razvoju weba, mobilnih aplikacija i igara čine ga vrijednim jezikom za učenje.

### **4.1.4. Python**

Python [28] je računalni programski jezik koji se često koristi za izradu web stranica i programa, automatizaciju zadataka i provođenje analize podataka. Python je postao glavni čimbenik znanosti o podacima, omogućujući analitičarima podataka i drugim stručnjacima da koriste jezik za provođenje složenih statističkih izračuna, stvaranje vizualizacija podataka, izradu algoritama strojnog učenja, manipuliranje i analizu podataka i dovršavanje drugih zadataka povezanih s podacima. Python se često koristi za razvoj backend dijela web stranice ili aplikacije.. Uloga Pythona u web razvoju može uključivati slanje podataka na poslužitelje i s njih, obradu podataka i komunikaciju s bazama podataka, URL usmjeravanje i osiguravanje sigurnosti. Python nudi nekoliko okvira za web razvoj, a najčešće korišteni su Django i Flask.

### **4.1.5. Django**

Django [29] je Python web okvir visoke razine koji omogućuje brz razvoj sigurnih web stranica koje se mogu lako i jednostavno održavati. Besplatan je i otvorenog koda, ima uspješnu i aktivnu zajednicu, sjajnu dokumentaciju i mnogo opcija za besplatnu i plaćenu podršku. Slijedi popularni oblikovni obrazac MVC koji programerima olakšava razvoj aplikacija.

### **4.1.6. Visual Studio Code**

Visual Studio Code [30] (poznatiji kao VS Code) besplatni je Microsoftov uređivač teksta otvorenog koda. Dostupan je za Windows, Linux i macOS. Iako je uređivač relativno lagan, uključuje neke snažne značajke koje su VS Code učinile jednim od najpopularnijih alata za razvojno okruženje u posljednje vrijeme. VS Code podržava širok raspon programskih jezika od Jave, C++ i Pythona do CSS-a, Go-a i Dockerfile-a. Korisničko sučelje VS Code-a omogućuje

laku i intuitivnu interakciju u usporedbi s drugim uređivačima teksta.

## **4.2. Alati za strojno učenje i repozitorij s podacima**

U ovome potpoglavlju navest će se i opisat alati korišteni prilikom strojnog učenja i dati detaljniji uvid u repozitorij na kojem se nalaze skupovi podataka korišteni s metodama strojnog učenja u ovome završnome radu.

### **4.2.1. Jupyter Notebook**

Jupyter Notebook [31] (ranije poznat kao IPython Notebook) je interaktivna web aplikacija za stvaranje i dijeljenje računalnih dokumenata. Projekt je najprije nazvan IPython, a kasnije je 2014. preimenovan u Jupyter. Riječ je o potpuno otvorenom proizvodu, a korisnici mogu besplatno koristiti sve dostupne funkcije. Podržava više od 40 jezika uključujući Python, R i Scalu. Bilježnica je promjenjiva datoteka spremljena u ipynb formatu. Kerneli su procesi koji pokreću interaktivni kod u određenom programskom jeziku i vraćaju rezultat korisniku. Oni također odgovaraju na zahtjeve za dovršetak kartice i introspekciju. Jupyter bilježnice mogu se pretvoriti u otvoreni standardni format kao što su HTML LaTeX, PDF, Markdown i Python. Jupyter bilježnice koriste se u razne svrhe. Intenzivno se koriste u svrhe istraživanja podataka budući da to uključuje puno ponavljanja. Također se koriste u drugim tijekovima rada u znanosti o podacima kao što su eksperimenti sa strojnim učenjem i modeliranje.

### **4.2.2. Kaggle**

Kaggle [32] je platforma za podatkovna znanstvena natjecanja, gdje se podatkovni znanstvenici i inženjeri strojnog učenja mogu međusobno natjecati u stvaranju najboljih modela za rješavanje specifičnih problema ili analizu određenih skupova podataka. Platforma također nudi zajednicu u kojoj korisnici mogu surađivati na projektima, dijeliti kod i skupove podataka te učiti iz rada drugih. Osnovan 2010., Google je kupio Kaggle 2017., a platforma je sada dio Google Clouda. Kaggle je domaćin raznih natjecanja koje sponzoriraju organizacije, u rasponu od predviđanja medicinskih ishoda do klasificiranja slika ili identificiranja lažnih transakcija. Uz natjecanja, Kaggle nudi i javne skupove podataka, bilježnice za strojno učenje i razne vrste poduka kako bi korisnicima pomogao u učenju i poboljšanju svojih vještina u znanosti o podacima i strojnom učenju. Iako danas privlači brojne početnike, Kaggle koriste i mnogiiskusni podatkovni znanstvenici kako bi usavršili svoje vještine, izgradili portfelje i povezali se s drugim industrijama.

Za potrebe izrade ove web aplikacije korištena su dva skupa podataka preuzeta s platforme Kaggle. Nad tim podacima primijenjeni su algoritmi strojnog učenja kako bi se razvio klasifikacijski model. Više informacija o razvoju klasifikacijskog modela i detaljan opis procesa nalazi se u nastavku.

### 4.3. Razvoj klasifikacijskog modela

U ovom potpoglavlju opisat će se razvoj klasifikacijskih modela koji su se koristili za postupak svrstavanja korisnika u razine fizičke spremnosti, a preuzeti su s platforme Kaggle. Korišteni skupovi podataka bili su pogodni jer njihove značajke predstavljaju najvažnije čimbenike treninga s pomoću kojih se određuje pripremljenost sportaša. Isto tako, skupovi podataka imaju slične značajke pa je proces obrade podataka i stvaranja modela bio jednostavniji. Korištena su dva skupa podataka, jedan za trčanje i jedan za biciklizam. Oba skupa podataka sastoje se od vrijednosti rezultata treninga koje je izmjerio jedan korisnik. U tablicama 4.1 i 4.2 opisane su značajke koje su korištene za treniranje modela i s pomoću kojih se procjenjuje pripremljenost sportaša koji koriste aplikaciju.

Distance	Udaljenost koju je korisnik prešao tijekom treninga trčanja (u kilometrima)
Calories	Ukupna kalorijska vrijednost potrošena tijekom treninga trčanja
Time	Vrijeme trajanja treninga trčanja (u minutama)
Avg HR	Prosječan broj otkucaja srca tijekom treninga trčanja
Avg Run Cadence	Prosječna kadenca tijekom treninga trčanja
Avg Pace	Prosječna brzina (broj minuta po kilometru)
Weighted Pace	Utegnuti tempo (decimalna vrijednost u rasponu [0,1])
Intensity	Intenzitet treninga (decimalna vrijednost)
Fitness	Izlazna veličina koja predstavlja razinu pripremljenosti (0 -low, 1-medium, 2-high)

Tablica 4.1. Prikaz opisa značajki korištenih u skupu podataka za trčanje

Distance	Udaljenost koju je korisnik prešao tijekom treninga biciklizma (u kilometrima)
Calories	Ukupna kalorijska vrijednost potrošena tijekom treninga biciklizma
Avg HR	Prosječan broj otkucaja srca tijekom treninga biciklizma
Avg Speed	Prosječna brzina vožnje tijekom treninga biciklizma (km/h)
Max Speed	Maksimalna brzina tijekom treninga biciklizma (km/h)
Avg Bike Cadence	Prosječna kadenca tijekom treninga biciklizma
Power	Snaga tijekom treninga biciklizma (W)
Weighted Pace	Utegnuti tempo (decimalna vrijednost u rasponu [0,1])
Intensity	Intenzitet treninga (decimalna vrijednost)
Fitness	Izlazna veličina koja predstavlja razinu pripremljenosti (0 -low, 1-medium, 2-high)

Tablica 4.2. Prikaz opisa značajki korištenih u skupu podataka za biciklizam

Zato što su skupovi podataka vrlo slični, odnosno imaju skoro identične značajke, postupak treniranja modela i pripreme skupa podataka bio je jako sličan. Iz tog razloga, cijeli postupak prikazat će se preko skupa podataka za trčanje. Priprema podataka započela je izbacivanjem stršećih i izostalih vrijednosti te uklanjanjem duplikata. Jedina razlika između ova dva skupa podataka je što je značajka „Time“ ostavljena u skupu podataka za trčanje, a u drugom skupu morala se izbaciti zbog previše izostalih ili neuobičajenih vrijednosti. Na slici 4.1 prikazan je postupak pretvaranja značajki „Time“, „Avg Pace“ i „Best Pace“ u minute (prikaz kao decimalni broj).

```

v Pretvaranje stupaca koji predstavljaju vremenske intervale u minute

[ ] data[['Time', 'Avg Pace', 'Best Pace']].tail()

Time Avg Pace Best Pace
684 0:55:33 6:59 4:34
685 0:42:24 7:02 5:39
686 0:45:17 7:34 6:05
687 0:30:28 7:20 5:54
688 0:29:32 7:21 6:16

[ ] def time_to_minutes(time):
    result = 0
    if not isinstance(time, list):
        return time
    if len(time) == 2:
        result+=float(time[0])
        result+=float(time[1]) / 60
    if len(time) == 3:
        result+=float(time[0]) * 60
        result+=float(time[1])
        result+=float(time[2]) / 60
    return float(result)

def convert(time_column):
    time = time_column
    time = time.apply(lambda x: x.split(":") if isinstance(x, str) else x)
    time = time.apply(time_to_minutes)
    return time.round(2)

```

Slika 4.1. Prikaz pretvaranja intervalnih značajki u minute

Sve vrijednosti udaljenosti prebačene su iz milja u kilometre zbog praktičnosti aplikacije. Također, brzina je prikazana kao broj minuta potrebnih za prelazak udaljenosti od 1 kilometra pa se i ta značajka pretvorila iz milja u kilometre. Na slici 4.2 prikazan je postupak pretvorbe.

```

v Pretvaranje udaljenosti, prosječne brzine i najveće brzine iz milja u kilometre

[ ] kilometers_distance = data['Distance'] * 1.609344
    data['Distance'] = kilometers_distance.round(2)

[ ] minutes_per_kilometer = data['Avg Pace'] * 0.621371192
    best_pace_min_per_kilometer = data['Best Pace'] * 0.621371192
    data['Avg Pace'] = minutes_per_kilometer.round(2)
    data['Best Pace'] = best_pace_min_per_kilometer.round(2)

```

Slika 4.2. Prikaz pretvorbe značajki iz milja u kilometre

Zatim su se uklonili svi nepotrebni stupci kako model ne bi imao preveliku dimenzionalnost. Skup podataka u početku nije imao značajku „Fitness“ koja bi predstavljala razinu spremnosti sportaša na temelju njegova treninga. Zato je smišljen algoritam s pomoću kojeg je dodana nova značajka „Fitness“ koja kao oznaka predstavlja razinu spremnosti sportaša. Ideja algoritma bila je da svaka

značajka koja raste porastom razine spremnosti ostane pozitivnog predznaka, a svaka koja se smanjuje negativnog. Zbrajanjem značajki dobila se izlazna veličina. Na slici 4.3 prikazan je postupak dodavanja značajke „Fitness“.

```
fitness_columns = ['Avg Run Cadence', 'Avg HR', 'Weighted Pace', 'Intensity', 'Distance', 'Time', 'Calories', 'Avg Pace']
fitness = data[fitness_columns].copy()
for col in fitness_columns:
    fitness[col] = pd.to_numeric(fitness[col], errors='coerce')

fitness["Scaled_Cadence"] = minmax_scale(fitness['Avg Run Cadence'])
fitness["Scaled_HR"] = minmax_scale(fitness["Avg HR"])
fitness["Scaled_Intensity"] = minmax_scale(fitness["Intensity"])
fitness["Scaled_Distance"] = minmax_scale(fitness["Distance"])
fitness["Scaled_Calories"] = minmax_scale(fitness["Calories"])
fitness["Scaled_Time"] = minmax_scale(fitness["Time"])
fitness["Scaled_Pace"] = minmax_scale(fitness["Avg Pace"])

fitness["Fitness_Level"] = (
    fitness["Scaled_Intensity"] +
    fitness["Weighted Pace"] +
    fitness["Scaled_Cadence"] -
    fitness["Scaled_HR"] +
    fitness["Scaled_Distance"] +
    fitness["Scaled_Calories"] -
    fitness["Scaled_Time"] -
    fitness["Scaled_Pace"]
)

data['Fitness'] = fitness['Fitness_Level']
data = data.dropna(subset=['Fitness'])
print(data.count())
data.head()
```

Slika 4.3. Prikaz dodavanja oznake (izlazne veličine) „Fitness“

Nakon dodavanja značajke „Fitness“, ona je mapirana u vrijednosti 0 (niska razina), 1 (srednja razina) i 2 (visoka razina). Svaka vrijednost manja od 0.3 mapirana je u 0, vrijednosti između 0.3 i 0.7 mapiranje su u 1, a vrijednosti veće od 0.7 mapirane su u 2. Na slici 4.4 prikazano je mapiranje vrijednosti.

```
▼ Podjela fitness-a u tri kategorije: 0(low), 1(medium) i 2(high)
▶ data['Fitness'] = pd.cut(data['Fitness'], bins=[-float('inf'), 0.3, 0.7, float('inf')], labels=[0, 1, 2], include_lowest=True, right=False)
data.head(25)
```

Slika 4.4. Prikaz mapiranja vrijednosti oznake „Fitness“

Nakon pripreme skupa podataka, odrađeno je skaliranje značajki i podjela skupa podataka na skupove za učenje, validaciju i testiranje. Funkcija *scale\_dataset* kao prvi parametar prima skup podataka, koji dijeli na skup ulaznih značajki i skup izlaznih značajki. Drugi parametar određuje hoće li se izvršiti preuzorkovanje kako bi vrijednosti svih kategorija bile ravnomjerno zastupljene. Na slici 4.5 prikazan je postupak skaliranja, preuzorkovanja i podjele skupa podataka.

```
data = data.replace(',', '.', regex=True)
train, valid, test = np.split(data.sample(frac=1), [int(0.6*len(data)), int(0.8*len(data))])
```

Metoda koja skalira značajke i dodavanje RandomOverSampler-a kako bi svake razine fitness-a bilo jednako u treningu modela

```
def scale_dataset(dataframe, oversample=False):
    x = dataframe[dataframe.columns[:-1]].values
    y = dataframe[dataframe.columns[-1]].values

    scaler = StandardScaler()
    x = scaler.fit_transform(x)

    if oversample:
        ros = RandomOverSampler()
        x, y = ros.fit_resample(x, y)

    data = np.hstack((x, np.reshape(y, (-1,1))))

    return data, x, y

train, x_train, y_train = scale_dataset(train, oversample=True)
valid, x_valid, y_valid = scale_dataset(valid, oversample=False)
test, x_test, y_test = scale_dataset(test, oversample=False)
```

Slika 4.5. Prikaz postupka skaliranja, preuzorkovanja i podjele skupa podataka

Tijekom procesa treniranja modela korištena su tri klasifikacijska algoritma. Uspoređivane su performanse rada algoritma logističke regresije, K-najbližih susjeda i stroja potpornih vektora. Na slici 4.6 prikazani su rezultati mjera klasifikacije za sva tri algoritma, izračunati tijekom 10 pokretanja.

algoritam	preciznost	odziv	f1-score	točnost
knn	0.852	0.854	0.849	0.861
log-reg	0.929	0.926	0.922	0.928
svm	0.939	0.929	0.927	0.935

Slika 4.6. Prikaz prosječnih rezultata mjera vrednovanja klasifikacije za tri različita algoritma tijekom 10 mjerenja

Iz dobivenih rezultata može se uočiti da je algoritam stroja potpornih vektora imao najbolje rezultate za ovakav tip problema. Stoga je taj algoritam integriran u aplikaciju i korišten za proces svrstavanja korisnika u kategorije razine spremnosti. Naposljetku, optimalni hiper-parametri određeni su s pomoću rešetkaste pretrage kako bi model postigao još bolje rezultate. Postupak određivanja optimalnih hiper-parametara s pomoću rešetkaste pretrage prikazan je na slici 4.7.



## Primjer traženja optimalnih hiperparametara SVM modela pri čemu se koristi rešetkasta pretraga

```
SVM_model2 = svm.SVC()

param_grid = {
    "C" : [1, 10, 100, 100],
    "gamma" : [10, 1, 0.1, 0.01]
}

svm_gscv = GridSearchCV(SVM_model2, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
svm_gscv.fit(X_train, y_train)

print(svm_gscv.best_params_)
print(svm_gscv.best_score_)
print(svm_gscv.cv_results_)

{'C': 100, 'gamma': 0.1}
0.9801980198019802
```

### 4.7. Prikaz postupka određivanja optimalnih hiperparametara s pomoću rešetkaste pretrage

Nakon rešetkaste pretrage izabran je konačan model koji ima hiper-parametar 'C' vrijednosti 100, 'gamma' vrijednosti 0.1 i korištenjem radijalne bazne funkcije. Implementacija modela u aplikaciju ostvaruje se uz pomoć 'pickle' biblioteke i metode 'dump' za serijalizaciju modela.

## 4.4. Programsko rješenje na strani korisnika

### 4.4.1. Programsko rješenje registracije i prijave korisnika

Svaki korisnik koji prvi puta koristi web aplikaciju treba izraditi korisnički račun. Prilikom registracije, korisnik unosi svoje korisničko ime, e-poštu te lozinku koja mora biti potvrđena. Ako je registracija obavljena uspješno, korisnik će automatski biti prijavljen u aplikaciji i taj prvi put neće morati obavljati još i prijavu. Postupak registracije ostvaren je s pomoću metode *signup* koja je prikazana na slici 4.8 te svaki korisnik prilikom registracije mora stvoriti drugačije korisničko ime. Spremanje podataka korisnika obavlja se preko *SignUpForm*, a istovremeno se stvara novi objekt tipa *Account* s pomoću kojeg će se pratiti aktivnost korisnikovih radnji kroz aplikaciju.

```

def signup(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            user = form.save()
            username = form.cleaned_data.get('username')
            raw_password = form.cleaned_data.get('password1')
            email = form.cleaned_data.get('email')

            Account.objects.create(
                user=user,
                name=username,
                mail=email,
                preferred_sport='running'
            )

            user = authenticate(username=username, password=raw_password)
            login(request, user)
            return redirect('/')
        else:
            form = SignUpForm()
    return render(request, 'registration/signup.html', {'form': form})

```

Slika 4.8. Programski kod registracije korisnika

Prijava korisnika implementirana je s pomoću Django pogleda *LoginView*. Korišten je i „django.contrib.auth.urls“ koji uključuje sve URL-ove vezane za autentifikaciju korisnika. Slika 4.9 prikazuje HTML kod obrasca prijave korisnika. Nakon uspješne prijave korisnika, korisnik se preusmjerava na početnu stranicu web aplikacije.

```

<div class="card-body p-4 p-sm-5">
  <h2 class="card-title text-center mb-4">Login</h2>
  <form method="post" action="{% url 'login' %}">
    {% csrf_token %}
    {% if form.non_field_errors %}
      <div class="alert alert-danger" role="alert">
        {{ form.non_field_errors }}
      </div>
    {% endif %}
    <div class="mb-3">
      <label for="{{ form.username.id_for_label }}" class="form-label">{{ form.username.label }}</label>
      <input type="text" name="{{ form.username.name }}" class="form-control" id="{{ form.username.id_for_label }}" required>
      {% if form.username.errors %}
        <div class="alert alert-danger mt-2" role="alert">
          {{ form.username.errors }}
        </div>
      {% endif %}
    </div>
    <div class="mb-3">
      <label for="{{ form.password.id_for_label }}" class="form-label">{{ form.password.label }}</label>
      <input type="password" name="{{ form.password.name }}" class="form-control" id="{{ form.password.id_for_label }}" required>
      {% if form.password.errors %}
        <div class="alert alert-danger mt-2" role="alert">
          {{ form.password.errors }}
        </div>
      {% endif %}
    </div>
    <div class="d-grid gap-2 mt-4">
      <button type="submit" class="btn btn-primary btn-lg">Login</button>
    </div>
  </form>
  <p class="text-center mt-4 mb-0">
    <a href="{% url 'password_reset' %}">Forgot password?</a>
  </p>
</div>

```

Slika 4.9. Programski HTML kod prijave korisnika

#### 4.4.2. Programsko rješenje testiranja razine fizičke spremnosti

Testiranje korisnikove fizičke spremnosti osmišljeno je tako da korisnik ispunjava obrazac podacima o rezultatu treninga. Polja obrasca odgovaraju značajkama modela čiji je postupak pripreme i treninga opisan u prethodnom potpoglavlju. Na slici 4.10 prikazan je HTML kod obrasca za unos podataka o treningu na temelju kojih se donosi odluka o razini fizičke spremnosti korisnika.

```

<div class="container">
  <h1>Test your running skills!</h1>
  <form method="post">
    {% csrf_token %}
    <div class="form-group">
      <label for="distance">Distance (km):</label>
      <input type="number" class="form-control" id="distance" name="distance" step="0.01" required>
    </div>
    <div class="form-group">
      <label for="calories">Calories:</label>
      <input type="number" class="form-control" id="calories" name="calories" required>
    </div>
    <div class="form-group">
      <label for="time">Time (minutes):</label>
      <input type="number" class="form-control" id="time" name="time" step="0.01" required>
    </div>
    <div class="form-group">
      <label for="avg_hr">Average Heart Rate (bpm):</label>
      <input type="number" class="form-control" id="avg_hr" name="avg_hr" required>
    </div>
    <div class="form-group">
      <label for="avg_run_cadence">Average Run Cadence (steps per minute):</label>
      <input type="number" class="form-control" id="avg_run_cadence" name="avg_run_cadence" required>
    </div>
    <div class="form-group">
      <label for="avg_pace">Average Pace (minutes per km):</label>
      <input type="number" class="form-control" id="avg_pace" name="avg_pace" step="0.01" required>
    </div>
    <div class="form-group">
      <label for="weighted_pace">Weighted Pace (minutes per km):</label>
      <input type="number" class="form-control" id="weighted_pace" name="weighted_pace" step="0.01" required>
    </div>
    <div class="form-group">
      <label for="intensity">Intensity:</label>
      <input type="number" class="form-control" id="intensity" name="intensity" step="0.01" required>
    </div>
    <div class="form-group" style="margin-top: 20px;">
      <button type="submit" class="btn btn-primary">Predict Fitness Level</button>
    </div>
  </form>

```

Slika 4.10. HTML kod obrasca za unos podataka o treningu

#### 4.4.3. Programsko rješenje unosa prehrane i treninga

Kako bi se pratio napredak korisnika omogućen mu je unos odrađenih treninga i konzumiranja prehrambenih proizvoda. Korisnik najprije popunjava obrasce o treningu i konzumiranom prehrambenom proizvodu nakon kojih se podaci pohranjuju i koriste za praćenje rezultata. Na slici 4.11 prikazana su polja obrazaca koja korisnik popunjava, dok je na slici 4.12 prikazan HTML kod unosa konzumiranog prehrambenog proizvoda s opcijom pretraživanja postojećih proizvoda, pri čemu se za dohvaćanje namirnica koristi besplatan API. Detaljniji opis postupka praćenja treninga i prehrane objašnjen je u nastavku.

```

class TrainingResultForm(forms.ModelForm):
    class Meta:
        model = TrainingResult
        fields = ['date', 'sport_type', 'duration', 'calories', 'distance']
        widgets = {
            'date': forms.DateInput(attrs={'type': 'date'}),
            'duration': forms.TextInput(attrs={'placeholder': 'HH:MM:SS'}),
        }

class FoodForm(forms.ModelForm):
    class Meta:
        model = Food
        fields = ['date', 'name', 'meal_type']
        widgets = {
            'date': forms.DateInput(attrs={'type': 'date'})
        }

```

Slika 4.11. Polja obrazaca za unos treninga i prehrane

```

<div class="container mt-5">
  <h1 class="text-center mb-4">Enter Nutrition Data</h1>

  <div class="row">
    <div class="col-md-6">
      <div class="card shadow mb-4">
        <div class="card-body">
          <form method="post">
            {% csrf_token %}
            {% for field in form %}
              <div class="form-group mb-3">
                <label for="{{ field.id_for_label }}">{{ field.label }}</label>
                {% render_field field class="form-control" %}
                {% if field.errors %}
                  <div class="invalid-feedback d-block">
                    {{ field.errors|join:", " }}
                  </div>
                {% endif %}
              </div>
            {% endfor %}
            <button type="submit" class="btn btn-primary">Submit</button>
          </form>
        </div>
      </div>
    </div>

    <div class="col-md-6">
      <div class="card shadow">
        <div class="card-body">
          <h3>Search for food</h3>
          <div class="input-group mb-3">
            <input type="text" id="food-search" class="form-control" placeholder="Enter food name">
            <div class="input-group-append">
              <button onclick="searchFood()" class="btn btn-secondary">Search</button>
            </div>
          </div>
          <div id="search-results" class="mt-3 row" style="max-height: 400px; overflow-y: auto;"></div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Slika 4.12. HTML kod unosa i pretrage prehrambenih proizvoda

## 4.5. Programsko rješenje na strani poslužitelja

U ovom potpoglavlju prikazat će se i opisati način ostvarivanja rada najvažnijih funkcionalnosti web aplikacije. To su postupci stvaranja preporuka za trening i prehranu, kao i praćenja treninga i prehrane korisnika.

### 4.5.1. Postupak stvaranja preporuka

Prije stvaranja preporuka korisniku, korisnik mora ispitati svoju fizičku spremnost. S pomoću istreniranih modela korisnik popunjava obrasce vezane za trening trčanja ili biciklizma nakon kojih im se izbacuje predikcija njihove fizičke spremnosti. Funkcija *running\_test* prima podatke putem POST zahtjeva, skalira ulazne značajke, te koristi unaprijed trenirani model za predikciju razine kondicije korisnika (nisku, srednju ili visoku) na temelju unesenih podataka o treningu. Rezultat predikcije se zatim vraća i prikazuje na HTML stranici, dok se eventualne greške također prosljeđuju na prikaz. Pogled *running\_test* koji obavlja pozadinsku logiku kategorizacije korisnika po trkačkoj spremnosti prikazan je na slici 4.13.

```
@login_required
@csrf_protect
def running_test(request):
    result = None
    error = None

    if request.method == 'POST':
        try:
            features = np.array([[
                float(request.POST['distance']),
                float(request.POST['calories']),
                float(request.POST['time']),
                float(request.POST['avg_hr']),
                float(request.POST['avg_run_cadence']),
                float(request.POST['avg_pace']),
                float(request.POST['weighted_pace']),
                float(request.POST['intensity'])
            ]])

            scaled_features = scaler.transform(features)
            prediction = model.predict(scaled_features)

            fitness_levels = ['Low', 'Medium', 'High']
            result = fitness_levels[int(prediction[0])]
        except Exception as e:
            error = str(e)

    return render(request, 'accounts/running_test.html', {'result': result, 'error': error})
```

Slika 4.13. Prikaz kategorizacije korisnika po trkačkoj spremnosti

Kada korisnik sazna svoju fizičku spremnost odlazi na stranicu za generiranje i preporuku plana treninga i prehrane. Korisnik treba unijeti razinu svoje fizičke spremnosti, sport kojim se želi baviti te cilj koji želi ostvariti (napredovati ili održavati razinu). Pogled *training\_nutrition\_plan* uzima podatke iz obrasca i koristi ih za generiranje plana korisniku. Na slici 4.14 prikazan je pogled *training\_nutrition\_plan*, a HTML kod prikaza planova trkačkih treninga korisniku prezentiran je na slici 4.15. Na isti način implementirani su i planovi za sportove biciklizam i duatlon te prehranu.

```
@login_required
def training_nutrition_plan(request):
    if request.method == 'POST':
        form = TrainingPlanForm(request.POST)
        if form.is_valid():
            plan = form.save(commit=False)
            sport = plan.sport
            goal = plan.goal
            fitness_level = plan.fitness_level
            return render(request, 'accounts/training_nutrition_plan.html', {
                'form': form,
                'sport': sport,
                'goal': goal,
                'fitness_level': fitness_level,
            })
        else:
            form = TrainingPlanForm()
    return render(request, 'accounts/training_nutrition_plan.html', {'form': form})
```

Slika 4.14. Prikaz spremanja podataka za plan treninga i prehrane

```
<h2 class="mb-0">Training Plan</h2>
</div>
<div class="card-body">
    {% if sport == 'running' %}
        {% if fitness_level == 'low' %}
            <p>Start with a mix of walking and jogging. Aim for 3 sessions per week.</p>
            <ul class="list-group">
                <li class="list-group-item">Week 1-2: 1 minute jog, 2 minutes walk. Repeat for 20 minutes.</li>
                <li class="list-group-item">Week 3-4: 2 minutes jog, 1 minute walk. Repeat for 25 minutes.</li>
                <li class="list-group-item">Week 5-6: 5 minutes jog, 1 minute walk. Repeat for 30 minutes.</li>
            </ul>
        {% elif fitness_level == 'medium' %}
            <p>Focus on increasing distance and introducing speed work. Aim for 4 sessions per week.</p>
            <ul class="list-group">
                <li class="list-group-item">2 easy runs (30-40 minutes)</li>
                <li class="list-group-item">1 interval session (e.g., 6-8 x 400m sprints with 90 seconds rest)</li>
                <li class="list-group-item">1 long run (45-60 minutes)</li>
            </ul>
        {% elif fitness_level == 'high' %}
            <p>Incorporate varied workouts to improve speed and endurance. Aim for 5-6 sessions per week.</p>
            <ul class="list-group">
                <li class="list-group-item">2 easy runs (40-50 minutes)</li>
                <li class="list-group-item">1 interval session (e.g., 10 x 400m at 5K pace with 90 seconds rest)</li>
                <li class="list-group-item">1 tempo run (20 minutes at half-marathon pace)</li>
                <li class="list-group-item">1 long run (75-90 minutes)</li>
                <li class="list-group-item">Optional: 1 recovery run (30 minutes very easy pace)</li>
            </ul>
        {% endif %}
    </div>
```

4.15. Generiranje planova trkačkih treninga ovisno o kategoriji fizičke spremnosti korisnika

## 4.5.2. Postupak praćenja treninga i prehrane

Ova funkcionalnost omogućuje korisniku praćenje odrađenih treninga i dnevnog kalorijskog unosa. Praćenje treninga implementirano je tako da korisnik popunjava obrazac preko kojeg unosi podatke o obavljenom treningu. Svaki trening sprema se u bazu podataka, a posljednjih 10 treninga (sortirano po datumu) prikazano je na linijskom dijagramu koji korisniku omogućava pregled potrošenih kalorija, sportske aktivnosti i datuma obavljenog treninga. Na slici 4.16 prikazana je logika spremanja podataka o treningu, gdje se podaci unose putem obrasca, validiraju i pohranjuju u bazu podataka. Prikaz linijskog dijagrama omogućen je kroz pogled *progress*, a programski kod za ovu funkcionalnost prikazan je na slici 4.17. Ovaj kod filtrira posljednjih deset unosa treninga, sortira ih po datumu i generira grafički prikaz kalorijske potrošnje u posljednjih deset treninga.

```
@login_required
def enter_training(request):
    if request.method == 'POST':
        form = TrainingResultForm(request.POST)
        if form.is_valid():
            training_result = form.save(commit=False)
            training_result.author = request.user
            training_result.save()
            request.user.account.update_last_activity()
            return redirect(reverse('accounts:progress'))
        else:
            form = TrainingResultForm()
    return render(request, 'accounts/enter_training.html', {'form': form})
```

Slika 4.16. Prikaz spremanja podataka o treningu u bazu podataka

```
@login_required
def progress(request):
    results = TrainingResult.objects.filter(author=request.user).order_by('-date')[:10]

    if not results:
        context = {'message': 'No training data available.'}
        return render(request, 'accounts/progress.html', context)

    dates = [result.date.strftime('%Y-%m-%d') for result in reversed(results)]
    calories = [result.calories for result in reversed(results)]
    sport_types = [result.sport_type for result in reversed(results)]

    plt.figure(figsize=(12, 6))
    plt.plot(range(len(dates)), calories, marker='o')
    plt.title('Calorie Expenditure for Last 10 Workouts')
    plt.xlabel('Workout Date and Sport Type')
    plt.ylabel('Calories Burned')

    plt.xticks(range(len(dates)), [f"{date}\n{sport}" for date, sport in zip(dates, sport_types)], rotation=45, ha='right')

    plt.tight_layout()

    buffer = io.BytesIO()
    plt.savefig(buffer, format='png')
    buffer.seek(0)
    plt.close()

    image_base64 = base64.b64encode(buffer.getvalue()).decode('utf-8')
```

Slika 4.17. Linijski dijagram koji prikazuje zadnjih 10 odrađenih treninga korisnika



Praćenje unosa prehrambenih proizvoda implementirano je na nešto drugačiji način. Slično kao i kod obrasca za trening, korisnik popunjava obrazac za prehranu gdje odabire datum, naziv prehrambenog proizvoda te vrstu obroka (doručak, ručak ili večera). Nakon što popuni obrazac, aplikacija koristi Nutritionix API [33] za dohvaćanje nutritivnih informacija o unesenom proizvodu. Te informacije, poput kalorija, proteina, ugljikohidrata i masti, automatski se popunjavaju i spremaju s unosom korisnika. Ova funkcionalnost omogućava korisniku da precizno prati svoju prehranu i nutritivne vrijednosti unesenih namirnica, što može biti korisno za praćenje i poboljšanje prehrambenih navika. Funkcija `enter_nutrition` prikazana na slici 4.18 implementira ovaj proces. Kada korisnik pošalje ispunjen obrazac, funkcija prvo provjerava valjanost unesenih podataka. Zatim koristi Nutritionix API za dohvaćanje nutritivnih informacija na temelju naziva unesene hrane. Ako je API zahtjev uspješan, funkcija izdvaja relevantne nutritivne podatke i sprema ih s unosom korisnika. Ovo omogućava automatsko popunjavanje detaljnih nutritivnih informacija bez potrebe da korisnik ručno unosi svaku vrijednost. Programski kod unosa prehrambenih proizvoda te korištenja API-ja za dobivanje rezultata o nutritivnim vrijednostima prikazan je na slici 4.18.

```
@login_required
def enter_nutrition(request):
    if request.method == 'POST':
        form = FoodForm(request.POST)
        if form.is_valid():
            food = form.save(commit=False)
            food.author = request.user

            api_url = "https://trackapi.nutritionix.com/v2/natural/nutrients"
            headers = {
                "x-app-id": settings.NUTRITIONIX_APP_ID,
                "x-app-key": settings.NUTRITIONIX_API_KEY,
                "Content-Type": "application/json"
            }
            data = {
                "query": food.name
            }

            response = requests.post(api_url, json=data, headers=headers)

            if response.status_code == 200:
                nutrient_data = response.json()['foods'][0]
                food.calories = nutrient_data['nf_calories']
                food.protein = nutrient_data['nf_protein']
                food.carbs = nutrient_data['nf_total_carbohydrate']
                food.fat = nutrient_data['nf_total_fat']

            food.save()
            request.user.account.update_last_activity()
            return redirect(reverse('accounts:progress'))
        else:
            form = FoodForm()
            context = {
                'form': form,
                'NUTRITIONIX_APP_ID': settings.NUTRITIONIX_APP_ID,
                'NUTRITIONIX_API_KEY': settings.NUTRITIONIX_API_KEY,
            }
            return render(request, 'accounts/enter_nutrition.html', context)
```

Slika 4.18. Unos prehrambenih proizvoda i dohvaćanje nutritivnih informacija s pomoću API-ja

Dodatna funkcionalnost pretraživanja hrane ostvarena je kako bi korisnici mogli pretraživati slične prehrambene proizvode koristeći naziv koji sadrži određeni tekst. Ovo omogućava korisnicima da brzo pronađu hranu koja odgovara njihovim unosima i preferencijama. Kod za pretragu hrane koristi JavaScript i Nutritionix API za dinamičko dohvaćanje i prikazivanje rezultata pretrage. Implementacija je ostvarena s pomoću dvije funkcije *searchFood* i *selectFood*. Funkcija *searchFood* odgovorna je za slanje upita Nutritionix API-ju s tekстом koji je korisnik unio u polje za pretragu. Kada API vrati rezultate, funkcija ih obrađuje i dinamički prikazuje prvih 10 sličnih proizvoda u obliku kartica, koje uključuju naziv hrane, količinu porcije, sliku i dugme za odabir. Ako dođe do greške tijekom pretrage, funkcija prikazuje poruku o grešci. Funkcija *selectFood* aktivira se kada korisnik klikne na dugme „Select“ na jednoj od kartica s rezultatima pretrage. Ova funkcija preuzima naziv odabrane hrane i postavlja ga u polje za unos imena hrane u obrascu, omogućavajući korisniku da brzo i jednostavno unese odabrani proizvod u obrazac za praćenje prehrane. Na slici 4.19 prikazane su obje funkcije.

```
function searchFood() {
  var query = document.getElementById('food-search').value;
  var resultsDiv = document.getElementById('search-results');

  fetch('https://trackapi.nutritionix.com/v2/search/instant?query=' + query, {
    headers: {
      'x-app-id': '{{ NUTRITIONIX_APP_ID }}',
      'x-app-key': '{{ NUTRITIONIX_API_KEY }}'
    }
  })
  .then(response => response.json())
  .then(data => {
    resultsDiv.innerHTML = '';
    data.common.slice(0, 10).forEach(item => {
      var card = `
      <div class="col-12 mb-3">
        <div class="card">
          <div class="row no-gutters">
            <div class="col-4">
              
            </div>
            <div class="col-8">
              <div class="card-body">
                <h5 class="card-title">${item.food_name}</h5>
                <p class="card-text">Serving: ${item.serving_qty || 'N/A'} ${item.serving_unit || ''}</p>
                <button onclick="selectFood('${item.food_name}')" class="btn btn-primary btn-sm">Select</button>
              </div>
            </div>
          </div>
        </div>
      `;
      resultsDiv.innerHTML += card;
    });
  })
  .catch(error => {
    console.error('Error: ', error);
    resultsDiv.innerHTML = '<p class="text-danger">Error fetching food data. Please try again.</p>';
  });
}

function selectFood(foodName) {
  var nameInput = document.querySelector('input[name="name"]');
  if (nameInput) {
    nameInput.value = foodName;
  }
}
}
```

Slika 4.19. Prikaz funkcija za pretragu i odabir prehrambenih proizvoda

Prikaz konzumiranih prehrambenih proizvoda također je implementiran u pogledu *progress*. Korisnik može za svaki datum pogledati što je konzumirao tijekom sva tri glavna obroka. Informacije koje stoje uz prehrambene proizvode su kalorijska vrijednost te gramaža ugljikohidrata, masti i bjelančevina. Programski kod prikazan je na slici 4.20.

```
selected_date = request.GET.get('date')

if selected_date:
    selected_date = make_aware(datetime.strptime(selected_date, '%Y-%m-%d'))
    food_data = Food.objects.filter(author=request.user, date=selected_date).order_by('meal_type')
else:
    selected_date = datetime.now().date()
    food_data = Food.objects.filter(author=request.user, date=selected_date).order_by('meal_type')

food_by_meal = defaultdict(list)
for food in food_data:
    food_by_meal[food.get_meal_type_display()].append(food)

food_data_for_template = {
    'date': selected_date,
    'meals': {
        'Breakfast': food_by_meal['Breakfast'],
        'Lunch': food_by_meal['Lunch'],
        'Dinner': food_by_meal['Dinner']
    }
}

context = {
    'progress_chart': image_base64,
    'food_data': [food_data_for_template],
    'selected_date': selected_date
}

return render(request, 'accounts/progress.html', context)
```

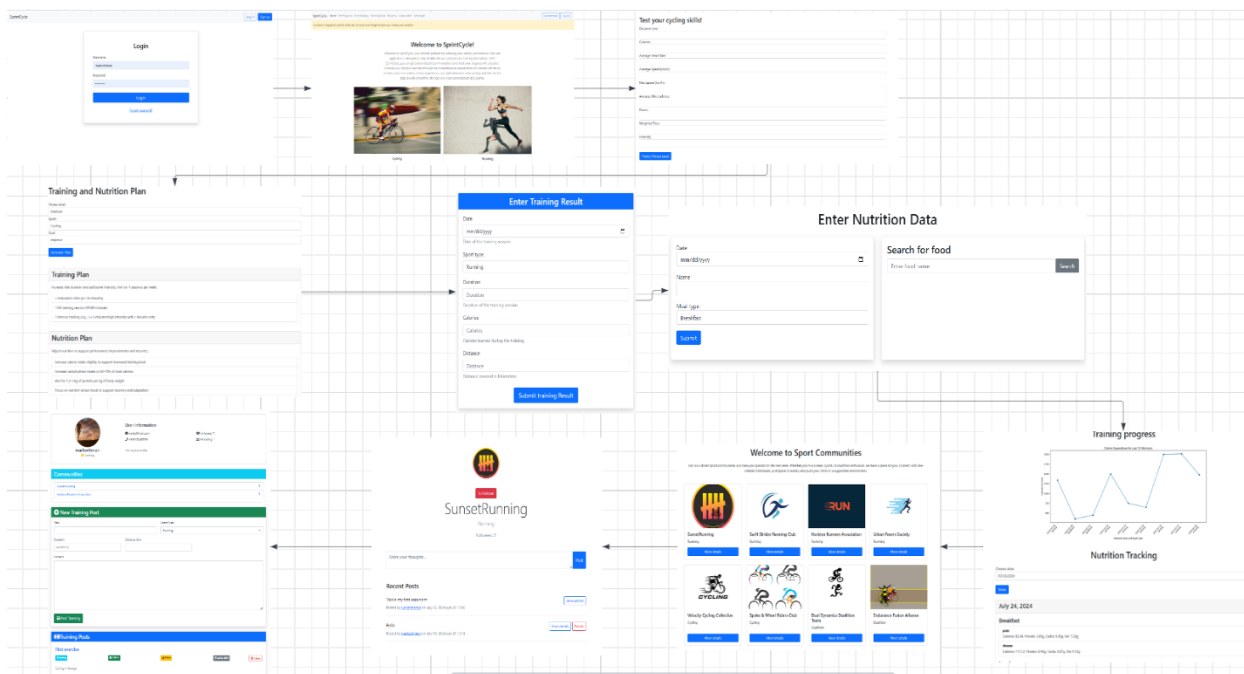
Slika 4.20. Programski kod praćenja i pregleda konzumiranih prehrambenih proizvoda po datumu i vrsti obroka

## 5. PRIKAZ RADA APLIKACIJE S ISPITIVANJEM I ANALIZOM

U ovom poglavlju prikazat će se tijekom korištenja aplikacije, provesti ispitivanje rada na temelju različitih vrsta korisnika te napraviti analiza na temelju rezultata ispitivanja.

### 5.1. Korištenje aplikacije

Svaki registrirani korisnik započinje korištenje aplikacije prijavom u svoj korisnički račun. Nakon uspješne prijave, na početnoj stranici ima mogućnost odabira provjere fizičke spremnosti u sportu biciklizma ili trčanja. Poslije dobivenih rezultata, korisnik će navigirati aplikacijom preko navigacijske trake te otići na stranicu za preporuku plana treninga i prehrane. Kada korisnik započne svoj plan treninga i prehrane, ima opciju unosa obavljenih treninga te konzumiranih prehrambenih proizvoda. Na stranici za napredak moći će vidjeti informacije o obavljenim treninzima i nutritivne vrijednosti o prehrambenim proizvodima koje je konzumirao. Dodatno, svaki korisnik može zapratiti sportske zajednice koje ga zanimaju i tako lakše pratiti novosti o druženjima, treninzima ili natjecanjima. Korisnici mogu međusobno komunicirati putem privatnih razgovora i zapratiti profile kako bi mogli pratiti rezultate treninga svojih prijatelja i kolega. Cijeli tijek korištenja web aplikacije prikazan je na slici 5.1.



Slika 5.1. Tijek korištenja web aplikacije

## 5.2. Postavke ispitivanja web aplikacije i slučajevi korištenja

U ovom potpoglavlju detaljnije će se prikazati ispitivanje aplikacije preko tri korisnička slučaja.

### 5.2.1. Postavke ispitivanja

Ispitivanje web aplikacije provest će se na najvažnijim funkcionalnostima na primjeru tri korisnika koji se bave različitim sportovima, nisu jednake fizičke spremnosti te će iz tog razloga imati i različite predložene planove intenziteta treninga i načina prehrane.

### 5.2.2. Prvi korisnički slučaj

Prvi korisnik trebao bi, na temelju podatak o svojem treningu biciklizma, biti u lošem fizičkom stanju. Također, njegov cilj je napredovati i postepeno ostvariti bolje rezultate. Unos podataka i rezultati prvog korisnika prikazani su na slici 5.2 , a slika 5.3 prikazuje plan generiran za postepeni napredak prema boljoj razini fizičke spremnosti. Planovi treninga i prehrane ostvareni su koristeći literaturu [34,35,36].

**Test your cycling skills!**

Distance (km):	20
Calories:	1200
Average Heart Rate:	176
Average Speed (km/h):	19
Max Speed (km/h):	25
Average Bike Cadence:	75
Power:	90
Weighted Pace:	0.125
Intensity:	2.5

[Predict Fitness Level](#)

▲ Predicted Fitness Level: Low

Slika 5.2. Unos podataka i rezultati fizičke spremnosti prvog korisnika

## Training and Nutrition Plan

Fitness level:  
Low

Sport:  
Cycling

Goal:  
Improve

Generate Plan

### Training Plan

Begin with shorter rides focusing on proper form. Aim for 3 sessions per week.

2 easy rides (30 minutes each)

1 slightly longer ride (45 minutes)

### Nutrition Plan

Adjust nutrition to support performance improvements and recovery.

Increase calorie intake slightly to support increased training load

Increase carbohydrate intake to 60-70% of total calories

Aim for 1.2-1.6g of protein per kg of body weight

Focus on nutrient-dense foods to support recovery and adaptation

Consider adding pre-workout and post-workout nutrition strategies

Slika 5.3. Generiran plan prehrane i treninga za prvog korisnika

### 5.2.2. Drugi korisnički slučaj

Što se tiče drugog korisnika, bavi se trčanjem i njegovi rezultati treninga su impresivni pa se kao rezultat očekuje visoka fizička spremnost. Želja drugog korisnika je zadržati ovu razinu spremnosti pa će mu se na temelju tih preferencija i predložiti daljnji plan treninga i prehrane. Unos podataka drugog korisnika i njegovi rezultati te generiranje plana prikazani su na slikama 5.4 i 5.5.

### Test your running skills!

Distance (km):  
20

Calories:  
1200

Time (minutes):  
51.20

Average Heart Rate (bpm):  
142

Average Run Cadence (steps per minute):  
172

Average Pace (minutes per km):  
3.5

Weighted Pace (minutes per km):  
0.5

Intensity:  
5.5

Predict Fitness Level

Predicted Fitness Level: High

Slika 5.4. Unos podataka i rezultati fizičke spremnosti drugog korisnika

## Training and Nutrition Plan

Fitness level:  
High

Sport:  
Running

Goal:  
Maintain

Generate Plan

### Training Plan

Incorporate varied workouts to improve speed and endurance. Aim for 5-6 sessions per week.

- 2 easy runs (40-50 minutes)
- 1 interval session (e.g., 10 x 400m at 5K pace with 90 seconds rest)
- 1 tempo run (20 minutes at half-marathon pace)
- 1 long run (75-90 minutes)
- Optional: 1 recovery run (30 minutes very easy pace)

### Nutrition Plan

Focus on balanced nutrition to support your current fitness level.

- Maintain current calorie intake
- Aim for a balanced diet with 50-60% carbohydrates, 20-30% protein, and 20-30% healthy fats
- Stay hydrated with water and electrolyte drinks during longer sessions
- Consume a mix of complex carbohydrates and lean proteins after workouts

Slika 5.5. Generiran plan prehrane i treninga za drugog korisnika

### 5.2.3. Treći korisnički slučaj

Treći korisnik bavi se i trčanjem i biciklizmom te će se na njegovom primjeru prikazati praćenje treninga i prehrane. Korisnik preko obrasca unosi rezultate treninga koje je obavio, a prikaz posljednjih 10 može pratiti preko linijskog dijagrama. Isto tako, bilježi svaki obrok koji pojede te ima pregled nutritivnih i kalorijskih vrijednosti. Proces bilježenja treninga i prehrane, odnosno unosa podatka prikazan je na slikama 5.6 i 5.7. Detaljan prikaz linijskog dijagrama i podataka o prehrani prikazani su na slikama 5.8 i 5.9.

## Enter Nutrition Data

Date 08/06/2024	Search for food eggs
Name fried eggs	duck eggs Serving: 1 egg Select
Meal type Breakfast	fried eggs Serving: 1 large Select
Submit	

Slika 5.6. Unos podataka o obroku

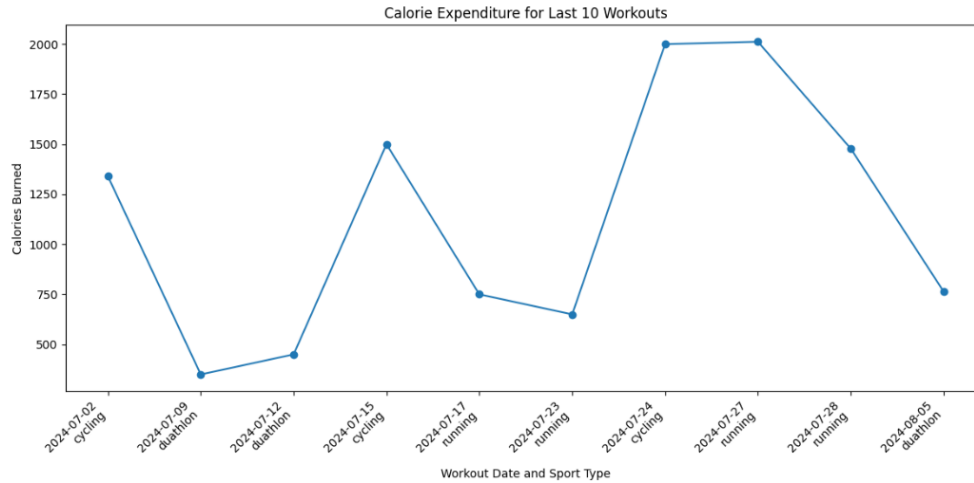
### Enter Training Result

Date 08/06/2024
Date of the training session
Sport type Running
Duration 1:10:05
Duration of the training session
Calories 650
Calories burned during the training
Distance 13
Distance covered in kilometers
Submit training Result

Slika 5.7. Unos podataka o obavljenom treningu



## Training progress



Slika 5.8. Linijski graf 10 posljednjih obavljenih treninga

## Nutrition Tracking

Choose date:  
08/06/2024

[show](#)

**August 06, 2024**

**Breakfast**

- fried eggs**  
Calories: 90.16, Protein: 6.26g, Carbs: 0.38g, Fat: 6.83g
- bacon**  
Calories: 161.46, Protein: 11.70g, Carbs: 0.59g, Fat: 12.11g
- cherry tomato**  
Calories: 3.06, Protein: 0.15g, Carbs: 0.66g, Fat: 0.03g

**Lunch**

- rice**  
Calories: 205.40, Protein: 4.25g, Carbs: 44.51g, Fat: 0.44g
- chicken breast**  
Calories: 198.00, Protein: 37.22g, Carbs: 0.00g, Fat: 4.28g

**Dinner**

- yogurt muesli**  
Calories: 239.44, Protein: 15.09g, Carbs: 44.02g, Fat: 1.88g

Slika 5.9. Pregled nutritivnih vrijednosti obroka po datumu

### 5.3. Rezultati ispitivanja s analizom

U današnje vrijeme veliki broj ljudi bavi se sportom bez pomoći trenera ili stručnih osoba. Takvi sportaši, odnosno rekreativci sami si moraju planirati težinu, intenzitet i količinu treninga te način ishrane kako bi imali energije za obavljanje istih. Kako bi se olakšao cijeli postupak, korisnici preko web aplikacije mogu provjeriti razinu svoje fizičke spremnosti i prilagoditi plan treninga i

prehrane ovisno o svojim ciljevima za napretkom ili održavanjem forme. Ispitivanje funkcionalnosti provedeno je na dva različita tipa korisnika, odnosno s različitim fizičkim predispozicijama i sportovima kojima se bave. Trkač pokazuje visoku razinu kondicije. Udaljenost od 20 km pretrčana za 51,20 minuta ukazuje na vrlo dobru izdržljivost i brzinu. Prosječni tempo od 3,5 minute po kilometru je impresivan za ovu udaljenost. Visok prosječni puls od 142 otkucaja u minuti, s visokom kadencom koraka od 172 u minuti, sugerira intenzivan napor održavan tijekom cijele aktivnosti. Utegnuti tempo od 0,5 i intenzitet 5,5 dodatno potvrđuju visoku razinu performanse. Stoga je procjena visoke razine kondicije opravdana. S druge strane, biciklist pokazuje nižu razinu kondicije. Iako je prevalio istu udaljenost (20 km), njegovi pokazatelji su manje impresivni. Prosječna brzina od 19 km/h je umjerena za rekreativnog biciklista. Viši prosječni puls (176 otkucaja/min) ukazuje na veći napor unatoč nižoj intenzivnosti aktivnosti (2,5). Niža kadenca pedaliranja (75 okretaja/min) i umjerena snaga (90 W) sugeriraju manju efikasnost. Utegnuti tempo od 0,125 je znatno niži nego kod trkača. Ovi faktori zajedno opravdavaju procjenu niske razine kondicije za biciklista.

Pouzdanost rezultata kategorizacije korisnika prema fizičkoj spremnosti rezultat je rada modela čije su performanse prikazane na slici 4.6. Konkretno, korišten je SVM (engl. Support Vector Machine) algoritam, koji pokazuje vrlo dobre performanse. Prema prikazanim metrikama, SVM model postiže preciznost od 0.939, odziv od 0.929, F1-score od 0.927 i točnost od 0.935. Ove visoke vrijednosti ukazuju na to da model pouzdano klasificira korisnike u odgovarajuće kategorije fizičke spremnosti na temelju unesenih podataka o njihovim aktivnostima.

Također, ispitivanje praćenja treninga i prehrane provedeno je na primjeru trećeg korisnika, koji se bavi i trčanjem i biciklizmom. Korisnik je unosio rezultate svojih treninga i bilježio obroke putem web aplikacije. Analiza podataka omogućila je praćenje njegovog napretka preko linijskog dijagrama, kao i pregled nutritivnih i kalorijskih vrijednosti unesenih obroka.

## 6. ZAKLJUČAK

U današnje vrijeme bavljenje sportom može predstavljati velike troškove koje si brojni ljudi ne mogu priuštiti. Kupovina sportskih rekvizita i opreme te plaćanje treninga i trenera samo su neki od troškova zbog kojih brojni sportaši moraju sami planirati cjelokupan proces bavljenja sportom. Glavni cilj izrađene web aplikacije je predočiti korisniku razinu njegove fizičke spremnosti i pomoći mu prilikom planiranja i praćenja optimalnog plana treninga i prehrane. Aplikacija se ne zadržava samo na tome, nego pruža opciju komunikacije s drugim korisnicima, praćenje njihovih rezultata te uvid u najnovije objave sportskih zajednica. Funkcionalnost kategorizacije korisnika po fizičkoj spremnosti ostvarena je tehnikama strojnog učenja ostvarenih u razvojnom okruženju Google Colab. Web aplikacija izrađena je korištenjem web okvira Django, a poslužiteljska strana ostvarena je upotrebom SQLite-a i Nutritionix API-ja.

Glavne funkcionalnosti web aplikacije detaljno su prikazane i ispitane na različitim primjerima. Dobiveni rezultati bili su očekivani i potvrđuju ispravnost rada aplikacije. Responzivan dizajn, uz ostale nefunkcionalne zahtjeve, dodatno poboljšava korisničko iskustvo te omogućuje korištenje aplikacije i na manjim uređajima. Aplikacija se trenutno ograničava na samo dva sporta, ali je moguće unaprijediti ju dodavanjem novih sportova čime bi se privukli novi korisnici. Također, proces praćenja treninga i prehrane mogao bi se optimirati dodavanjem detaljnijih informacija o odrađenom treningu, provođenjem raznih analiza za lakše praćenje napretka te omogućavanjem skeniranja barkodova prehrambenih proizvoda. Dodatno, kategorizacija korisnika po fizičkoj spremnosti mogla bi se unaprijediti korištenjem složenijih tehnika strojnog učenja kako bi se postigla još veća preciznost. Dodavanjem blogova omogućilo bi se trenerima, sportašima i nutricionistima da prenesu svoja znanja i iskustva svim korisnicima web aplikacije. U konačnici, u kontekstu praćenja i preporuke treninga i prehrane, aplikacija se pokazala kao vrijedno sredstvo koje može značajno smanjiti troškove povezane s personaliziranim treniranjem i planiranjem prehrane. Omogućujući korisnicima jasnu sliku o njihovoj fizičkoj spremnosti i personalizirane preporuke, aplikacija im pomaže u učinkovitom upravljanju treninzima i prehranom, podržavajući ih u postizanju njihovih sportskih ciljeva.

## 7. LITERATURA

- [1] B. Matijević, A. Čutić, Značaj pravilne prehrane za očuvanje zdravlja sportaša i rekreativaca, Zadar: Međunarodni stručno-znanstveni skup, zaštita na radu i zaštita zdravlja, 2016.
- [2] D. Lauš, Prehrana i sport, Bjelovar: Visoka tehnička škola u Bjelovaru, 2009.
- [3] L.L. Spriet, »Performance nutrition for athletes,« *Sports Medicine*, 2019, pp. 1-2.
- [4] J.R. Hoffman, M.J. Falvo, »Protein-which is best?,« *Journal of sports science & medicine*, br. 3, p. 118.
- [5] S. M. Shirreffs, »Hydration in sport and exercise: water, sports drinks and other drinks,« u *Nutrition bulletin*, pp. 374.-379.
- [6] M. Crnjaković, »Trening izdržljivosti (1. dio),« *RUN.HR*, 22. 12. 2014.. [Mrežno]. Available: <https://run.hr/trening-izdrzljivosti-1/>. [Pokušaj pristupa 10. 6. 2024.].
- [7] D. Tuzlić, »Osnove biciklističkog treninga - III. dio,« *3sporta*, 16. 2. 2024.. [Mrežno]. Available: <https://3sporta.com/osnove-biciklistickog-treninga-iii-dio/>. [Pokušaj pristupa 10. 6. 2024.].
- [8] »Eating for peak athletic performance,« *UNWEALTH*, 4. 3. 2019.. [Mrežno]. Available: <https://www.uwhealth.org/news/eating-for-peak-athletic-performance>. [Pokušaj pristupa 10. 6. 2024.].
- [9] Z. Jakić, »Trening snage gornjeg dijela tijela za trkače,« *fitness.com.hr*, 10. 3. 2021.. [Mrežno]. Available: <https://www.fitness.com.hr/vjezbe/savjeti-za-vjezbanje/Trening-snage-gornjeg-dijela-tijela-za-trkace.aspx>. [Pokušaj pristupa 10. 6. 2024.].
- [10] E. Quinn, »Using theCooper Test 12-Minute Run To Check Aerobic Fitness,« *verywellfit*, 25. 4. 2024.. [Mrežno]. Available: <https://www.verywellfit.com/fitness-test-for-endurance-12-minute-run-3120264>. [Pokušaj pristupa 11. 6. 2024.].
- [11] M. Constantin, »Theoretical concepts for physical preparation of athletes,« *Gymnasium*, svez. 17, br. 2, p. 155, 2016.
- [12] S. Hurley, »Functional Threshold Power: What FTP Means to Cyclists,« *TRAINERROAD*, 2024.. [Mrežno]. Available: <https://www.trainerroad.com/blog/what-ftp-really-means-to-cyclists/>. [Pokušaj pristupa 12. 6. 2024.].
- [13] D. Milanović, Teorija treninga, Zagreb: Kineziološki fakultet Sveučilišta u Zagrebu, 2013.
- [14] T.O. Bompa, C. Buzzichelli, ., *Periodization: Theory and Methodology of Training*, 2019.

- [15] A. Orin, »I'm Mike Lee, and This Is the Story Behind MyFitnessPal,« LIFEHACKER, 19. 3. 2024.. [Mrežno]. Available: <https://lifehacker.com/im-mike-lee-and-this-is-the-story-behind-myfitnesspal-1547205813> . [Pokušaj pristupa 12. 6. 2024.].
- [16] S. Yeager, »A Cyclist's Complete Guide to Understanding,« TrainingPeaks, 25. 1. 2022.. [Mrežno]. Available: <https://www.bicycling.com/training/a38795537/how-to-use-trainingpeaks/>. [Pokušaj pristupa 12. 6. 2024.].
- [17] S. Denali, »Strava Features Complete Overview,« STRAVA, 17. 7. 2022.. [Mrežno]. Available: <https://communityhub.strava.com/t5/athlete-knowledge-base/strava-features-complete-overview/tap/275>. [Pokušaj pristupa 12. 6. 2024.].
- [18] »Non-functional Requirements in Software Engineering,« [Mrežno]. Available: Non-functional Requirements in Software Engineering - GeeksforGeeks . [Pokušaj pristupa 13. 6. 2024.].
- [19] »MVC,« mdn web docs, [Mrežno]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. [Pokušaj pristupa 13. 6. 2024.].
- [20] N. Bolf, »Osvježimo znanje: Strojno učenje,« *Časopis kemičara i kemijskih inženjera Hrvatske* , svez. 70, br. 9, pp. 571.-593, 2021..
- [21] V. Kanade, »What is Logistic Regression?,« spiceworks, 8. 3. 2022.. [Mrežno]. Available: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/>. [Pokušaj pristupa 14. 6. 2024.].
- [22] »K-Nearest Neighbor (KNN) Algorithm,« GeeksforGeeks, 25. 1. 2024.. [Mrežno]. Available: <https://www.geeksforgeeks.org/k-nearest-neighbours/>. [Pokušaj pristupa 14. 6. 2024.].
- [23] »Support Vector Machine (SVM) Algorithm,« GeeksforGeeks, 10. 6. 2023.. [Mrežno]. Available: <https://www.geeksforgeeks.org/support-vector-machine-algorithm/>. [Pokušaj pristupa 14. 6. 2024.].
- [24] S. Astari, »What is HTML? Hypertext Markup Language Basics Explained,« Hostinger Tutorials, 25. 8. 2023.. [Mrežno]. Available: <https://www.hostinger.com/tutorials/what-is-html>. [Pokušaj pristupa 22. 6. 2024.].
- [25] »What is CSS?,« MDN WEB DOCS, 1. 1. 2024.. [Mrežno]. Available: [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/What\\_is\\_CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS). [Pokušaj pristupa 22. 6. 2024.].
- [26] »Bootstrap Tutorial,« GeeksforGeeks, 17. 1. 2024.. [Mrežno]. Available: <https://www.geeksforgeeks.org/bootstrap/>. [Pokušaj pristupa 22. 6. 2024.].
- [27] A. Jordana, »What is JavaScript? A Beginner's Guide to the Basics of JS,« Hostinger Tutorials, 13. 5. 2024.. [Mrežno]. Available: <https://www.hostinger.com/tutorials/what-is-javascript>. [Pokušaj pristupa

22. 6. 2024.].

- [28] »What is Python Used For? A Begginer's Guide,« Coursera, 3. 4. 2024.. [Mrežno]. Available: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>. [Pokušaj pristupa 22. 6. 2024.].
- [29] »Djang introduction,« MDN WEB DOCS, 28. 2. 2024.. [Mrežno]. Available: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> . [Pokušaj pristupa 22. 6. 2024.].
- [30] »What is Visual Studio Code?,« Educative, 2024. [Mrežno]. Available: <https://www.educative.io/answers/what-is-visual-studio-code>. [Pokušaj pristupa 22. 6. 2024.].
- [31] »Jupyter Notebook,« Domino, [Mrežno]. Available: <https://domino.ai/data-science-dictionary/jupyter-notebook>. [Pokušaj pristupa 22. 6. 2024.].
- [32] »What is Kaggle and What Is It Used For?,« Coursera, 29. 11. 2023.. [Mrežno]. Available: <https://www.coursera.org/articles/kaggle>. [Pokušaj pristupa 23. 6. 2024.].
- [33] »Nutrition & Exercise API,« Nutritionix, 2024.. [Mrežno]. Available: <https://www.nutritionix.com/business/api>. [Pokušaj pristupa 4. 8. 2024.].
- [34] T.O. Bompa, C. Buzzichelli, »Human kinetics,« u *Periodization training for sports*, 2015., pp. 78.-92.
- [35] R. Barton, *The Cycling Bible: The complete guide for all cyclists from novice to expert* , Bloomsbury Publishing, 2015, pp. 67.-88.
- [36] J. Daniels, »Daniel's running formula,« u *Human Kinetics*, 2013., pp. 46.-68.

## **ŽIVOTOPIS**

Marko Ehman rođen je 21. kolovoza 2002. godine u Osijeku. Pohađao je Osnovnu školu Frana Krste Frankopana u Osijeku. Nakon završetka osnovne škole upisuje III. gimnaziju Osijek. Poslije srednje škole upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, preddiplomski sveučilišni studij Računarstva, smjer Programsko inženjerstvo. 2024. godine osvaja prvo mjesto na natjecanju Digitalni inovacijski inkubator u području Razvoj i inoviranje AI rješenja za održivost - ICT industrija powered by Hrvatski Telekom.

## SAŽETAK

Cilj završnog rada je razvoj web aplikacije za preporuku i praćenje treninga i prehrane sportaša koji se bave trčanjem i biciklizmom. Aplikacija nudi korisnicima mogućnost procjene fizičke spremnosti, na temelju koje se preporučuje individualizirani plan treninga i prehrane. Korisnici mogu pratiti rezultate svojih treninga putem linijskog dijagrama, dok su konzumirani prehrambeni proizvodi prikazani po obrocima i datumu. Osim osnovnih funkcionalnosti praćenja treninga i prehrane, web aplikacija nudi i elemente društvene mreže. Korisnicima je omogućeno pregledavanje profila drugih korisnika, praćenje rezultata treninga, primanje obavijesti o treninzima i događanjima organiziranim od strane sportskih zajednica te privatna komunikacija među korisnicima. Aplikacija je implementirana korištenjem web okvira Django, dok je korisničko sučelje izrađeno uz pomoć HTML-a, CSS-a, Bootstrapa i JavaScript-a. Na strani poslužitelja korišteni su SQLite baza podataka i Nutritionix API za upravljanje prehrambenim informacijama. Razvoj modela uključuje primjenu strojnog učenja za preporuku treninga i prehrane, dok su glavni algoritmi testirani i vrednovani kroz različite korisničke slučajeve. Dobiveni rezultati potvrdili su ispravan rad aplikacije i njezinu upotrebljivost za krajnje korisnike.

**Ključne riječi:** biciklizam, prehrana, strojno učenje, trčanje, trening, web aplikacija.



## **ABSTRACT**

**Title:** Web application for recommending and monitoring training and nutrition for athletes based on machine learning

The goal of the final paper is the development of a web application for recommending and monitoring the training and nutrition of athletes who are involved in running and cycling. The application offers users the possibility of assessing physical fitness, based on which an individualized training and nutrition plan is recommended. Users can monitor the results of their training via a line diagram, while the food products they consume are displayed by portion and date. In addition to the basic functionality of tracking training and nutrition, the web application also offers elements of a social network. Users can view other users' profiles, track training results, receive notifications about training and events organized by sports communities, and communicate privately with users. The application was implemented using the Django web framework, while the user interface was created using HTML, CSS, Bootstrap, and JavaScript. On the server side, the SQLite database and Nutritionix API were used to manage nutritional information. The development of the model includes the application of machine learning to recommend training and nutrition, while the main algorithms were tested and evaluated through different user cases. The obtained results confirmed the correct operation of the application and its usability for end users.

**Keywords:** cycling, machine learning, nutrition, running, training, web application.

## **PRILOZI** (on-line)

Prilog 1. Završni rad u datoteci docx.

Prilog 2. Završni rad u datoteci pdf.

Prilog 3. Gitlab repozitorij programskog koda aplikacije.

Prilog 4. Gitlab repozitorij razvijenih modela strojnog učenja.