Tehničko veleučilište u Zagrebu

Stručni studij računarstva

**IZRADA WEB APLIKACIJE TRAINFLOW U ANGULARU**

Završni rad

Kolegij: Završni rad

Ime i prezime nastavnika: Ognjen Staničić, dipl.ing, v. pred.

Ime i prezime studenta: Domagoj Čirko

Zagreb, kolovoz 2025.

# Sažetak

U ovom radu prikazuje se razvoj web aplikacije TrainFlow koristeći moderne tehnologije: Angular, Typescript, Java Spring Boot, MySQL, HTML i CSS. Aplikacija korisnicima omogućuje jednostavno praćenje sportskih aktivnosti, kreiranje i aktiviranje personaliziranih trening planova, spremanje istih planova te vođenje evidencije o odrađenim treninzima. Osim toga, omogućeno je postavljanje osobnih ciljeva i praćenje napretka putem intuitivnog korisničkog sučelja.

Backend aplikacije razvijen je u programskom jeziku Java korištenjem Spring Boot frameworka. Putem Spring-a omogućen je rad s bazom podataka, autentifikacija i autorizacija korisnika, izrada REST API-ja te poslovna logika nužna za upravljanje trening planovima i zapisima o treninzima i aktivnostima. Podaci se pohranjuju u MySQL bazu uz pomoć JPA i Hibernate ORM-a, čime se osigurava pouzdano i sigurno rukovanje podacima. Frontend dio implementiran je u Angularu koristeći TypeScript, HTML i CSS koji nam omogućuje responzivno korisničko iskustvo i jednostavnu integraciju s REST API-jem.

**Ključne riječi**: web aplikacija, trening, Spring, dizajn, Java, MySQL, Angular, Typescript, frontend, backend, REST API

# Sadržaj

[Sažetak 2](#_Toc207192430)

[Sadržaj 3](#_Toc207192431)

[Popis kratica i oznaka: 4](#_Toc207192432)

[Popis slika 5](#_Toc207192433)

[1. UVOD 1](#_Toc207192434)

[2. Opis i funkcionalnosti aplikacije TrainFlow 2](#_Toc207192435)

[2.1 Opis aplikacije 2](#_Toc207192436)

[2.2 Funkcionalnosti aplikacije 3](#_Toc207192437)

[2.3 Primjeri korištenja 4](#_Toc207192438)

[3. Tehnologije 7](#_Toc207192439)

[3.1 Backend 7](#_Toc207192440)

[**3.1.1** **Java Spring Boot** 7](#_Toc207192441)

[**3.1.2** **Spring Boot** 9](#_Toc207192442)

[**3.1.3** **MySQL Baza podataka** 11](#_Toc207192443)

[**3.1.4** **Primjer korištenja baze podataka u TrainFlowu** 13](#_Toc207192444)

[3.2 Frontend 15](#_Toc207192445)

[**3.2.1** **HTML** 16](#_Toc207192446)

[**3.2.2** **CSS** 18](#_Toc207192447)

[**3.2.3** **JavaScript/TypeScript** 20](#_Toc207192448)

[4. Izrada dizajna 22](#_Toc207192449)

[4.1 Ciljevi i ciljane skupine 22](#_Toc207192450)

[4.2 Persone 23](#_Toc207192451)

[4.3 Sitemap ili Arhitektura Informacija 25](#_Toc207192452)

[4.4 Wireframe ili žičani kostur 26](#_Toc207192453)

[4.5 Odabir boja 28](#_Toc207192454)

[4.6 Izrada prototipa u Figmi 29](#_Toc207192455)

[5. Zaključak 31](#_Toc207192456)

[6. Literatura 32](#_Toc207192457)

# Popis kratica i oznaka:

API - Application Programming Interface

CSS – Cascading Shading Sheets

HTML – Hypertext Markup Language

JPA – Java Persistence API

MySQL – My Structured Query Language

ORM – Object-Relational Mapping

REST - Representational state transfer

# Popis slika

[Slika 1. Početna stranica TrainFlowa 5](#_Toc207192458)

[Slika 2. Register Stranica 5](#_Toc207192459)

[Slika 3. Login Stranica 5](#_Toc207192460)

[Slika 4. Jedan od tri dijelova stranice za kreiranje planova 6](#_Toc207192461)

[Slika 5. Dio MyProfile stranice 6](#_Toc207192462)

[Slika 6. Java logo 8](#_Toc207192463)

[Slika 7. Prikaz Training Controllera 10](#_Toc207192464)

[Slika 8. Primjer stvaranja tablice u MySQL-u 12](#_Toc207192465)

[Slika 9. Prikaz dodavanja nove aktivnosti u bazu podataka pomoću Eloquent ORM-a 14](#_Toc207192466)

[Slika 10. Prikaz dijela početne stranice u HTML-u 17](#_Toc207192467)

[Slika 11. Prikaz jednostavnog koda u CSS-u 19](#_Toc207192468)

[Slika 12. Prikaz dijela TypeScript dokumenta Your Training Plans stranice 21](#_Toc207192469)

[Slika 13. Prikaz persone za TrainFlow 23](#_Toc207192470)

[Slika 14. Primjer druge persone za TrainFlow 24](#_Toc207192471)

[Slika 15. Prikaz sitemap-a za TrainFlow 25](#_Toc207192472)

[Slika 16. Prikaz početne stranice pomoću wireframe-a 27](#_Toc207192473)

[Slika 17. Palete boja korištene u aplikaciji 28](#_Toc207192474)

[Slika 18. Prototip početne stranice TrainFlow 30](#_Toc207192475)

# UVOD

Internet je jedan od najvažnijih izuma modernog doba koji je uvelike olakšao svakodnevni život pojedinca. Jedna od usluga interneta, web aplikacije, značajno su promijenile način na koji ljudi rade, komuniciraju i obavljaju svakodnevne aktivnosti. Web aplikacije su interaktivni računalni programi koji se izvode na internetu. Omogućuju korisnicima pristup informacijama i uslugama u stvarnom vremenu, često putem preglednika, čime se eliminira potreba za instalacijom dodatnog softvera.

U ovom radu prikazat će se razvoj web aplikacije TrainFlow, čija je svrha olakšati korisnicima praćenje i organizaciju fizičkih aktivnosti te pridonijeti zdravijem načinu života. U današnje vrijeme, ljudi sve više počinju voditi brigu o svome zdravlju i fizičkoj aktivnosti. Ova aplikacija osmišljena je kako bi pružila podršku korisnicima u postizanju njihovih ciljeva. Aplikacija omogućuje kreiranje i korištenje različitih trening planova, praćenje aktivnog plana te vođenje dnevnika treninga (workout logova). Na taj način korisnici mogu bilježiti i analizirati vlastiti napredak, bolje razumjeti svoje navike i postići željene ciljeve.

Razvoj aplikacije ostvaren je korištenjem Java Spring Boot frameworka za backend sustav, dok će se MySQL koristiti za pohranu podataka. Angular i TypeScript upotrijebljeni su za razvoj frontend dijela web aplikacije, dok su HTML i CSS korišteni kako bi se osigurala responzivnost i vizualna privlačnost korisničkog sučelja. U radu će se detaljno opisati svi dijelovi razvoja, od početnog planiranja i definiranja funkcionalnosti do implementacije i integracije svih dijelova sustava. Cilj ovog rada je prikazati cjelokupan proces razvoja web aplikacije TrainFlow te pokazati kako tehnologija može unaprijediti kvalitetu života pojedinca kroz jednostavno praćenje fizičkih aktivnosti i rezultata.

# Opis i funkcionalnosti aplikacije TrainFlow

Aplikacija **TrainFlow** razvijena je kako bi korisnicima pružila jednostavan i pregledan način praćenja vlastitih fitness aktivnosti. Poseban naglasak stavljen je na intuitivno korištenje, responzivan dizajn i mogućnost personalizacije. Korisnici kroz aplikaciju mogu kreirati i prilagoditi vlastite planove treninga ili odabrati jedan od ponuđenih, evidentirati planirane i odrađene vježbe te kroz vrijeme pratiti svoj napredak. Uz pregled prijašnjih logova i usporedbu rezultata, TrainFlow omogućuje jasniji uvid u ostvarene ciljeve. Korištenjem sigurnih korisničkih računa i stabilne arhitekture aplikacija nudi privatnost podataka te pouzdan rad na svim uređajima.

## Opis aplikacije

TrainFlow je moderna web aplikacija namijenjena svim korisnicima koji žele na jednostavan, pregledan i pouzdan način pratiti svoje treninge, planirati tjedne rasporede i pratiti osobni napredak. Ideja aplikacije nastala je iz potrebe za centraliziranim alatom koji spaja funkcionalnosti praćenja treninga, vođenja bilješki, korištenja već gotovih planova te personalizacije prema željama korisnika. Aplikacija kombinira backend u Spring Bootu i MySQL bazu podataka s modernim frontend rješenjima u Angularu. Kroz pregledno sučelje korisnici mogu stvarati vlastite planove, koristiti predložene programe, bilježiti odrađene treninge i pratiti povijest logova. Posebna pažnja posvećena je intuitivnom korištenju i vizualnom dizajnu koji prati principe jednostavnosti i jasnoće.

Jedna od glavnih značajki aplikacije je fleksibilnost u kreiranju planova. Korisnici mogu definirati ciljeve, odabrati vježbe i postaviti težinu, broj ponavljanja i serija prema vlastitim preferencijama. Osim toga, dostupni su i unaprijed pripremljeni planovi koje korisnik može spremiti i koristiti kao polaznu točku. Svaki trening dan detaljno je strukturiran, čime se olakšava organizacija i dosljednost u izvođenju vježbi.

Prilikom dizajniranja aplikacije naglasak je stavljen na preglednost i jednostavnost korištenja. Struktura stranica je organizirana tako da korisnik već na početnoj stranici dobije uvid u sve mogućnosti – od pregleda predloženih planova, preko kreiranja vlastitog plana, do vođenja evidencije o odrađenim treninzima. Za svakog korisnika se kreira personalizirani profil u kojem se čuvaju spremljeni planovi i povijest treninga. Time se stvara privatno i prilagođeno iskustvo koje pomaže održavanju motivacije i kontinuiteta.

Posebna vrijednost aplikacije je mogućnost vođenja evidencije o odrađenim treninzima. Korisnici nakon svake sesije mogu unijeti stvarne rezultate, usporediti ih s planiranim vrijednostima te dodati bilješke. Time se stvara osobni dnevnik napretka koji motivira i omogućuje analizu dugoročnih rezultata. Ovakav pristup potiče disciplinu i daje jasniji uvid u ostvarene ciljeve.

## Funkcionalnosti aplikacije

Aplikacija TrainFlow obuhvaća niz funkcionalnosti koje zajedno tvore cjelovit sustav za planiranje, izvođenje i praćenje treninga. Sve značajke dizajnirane su tako da budu intuitivne i jednostavne za korištenje, bez obzira na prethodno iskustvo korisnika s digitalnim alatima za fitness. U nastavku su opisane ključne mogućnosti koje aplikacija pruža:

* **Kreiranje vlastitih planova treninga** – korisnici mogu odrediti raspored dana u tjednu, odabrati ili dodati vježbe te definirati broj serija, ponavljanja, težinu i odmore između vježbi. Na taj način plan postaje potpuno prilagodljiv individualnim ciljevima.
* **Pregled i primjena predloženih planova** – TrainFlow sadrži kolekciju unaprijed izrađenih planova usmjerenih na različite ciljeve (npr. povećanje snage, redukcija težine, izgradnja kondicije). Korisnik ih može jednostavno spremiti i koristiti kao temelj za vlastite izmjene.
* **Spremanje planova u osobni profil** – svaki registrirani korisnik ima svoju kolekciju spremljenih planova. Time se osigurava da su planovi privatni i dostupni isključivo njihovom vlasniku, bez utjecaja na druge korisnike.
* **Bilježenje odrađenih treninga** – nakon svake sesije korisnik može unijeti ostvarene serije, ponavljanja i korištene težine te ih usporediti s planiranim vrijednostima. Uz to je moguće dodati bilješke o dojmu, umoru ili eventualnim poteškoćama.
* **Bilježenje odrađenih treninga** – nakon svake sesije korisnik može unijeti ostvarene serije, ponavljanja i korištene težine te ih usporediti s planiranim vrijednostima. Uz to je moguće dodati bilješke o dojmu, umoru ili eventualnim poteškoćama.
* **Upravljanje logovima** – korisnik ima mogućnost brisanja pogrešno unesenih podataka. Sustav prije brisanja traži potvrdu kako bi se spriječile neželjene greške.
* **Korisnička autentifikacija** – prijava i registracija osiguravaju da su podaci dostupni samo vlasniku računa, čime se garantira privatnost i sigurnost informacija.
* **Responzivan dizajn** – aplikacija je optimizirana za rad na različitim uređajima, od stolnih računala do mobilnih telefona, što omogućuje jednako iskustvo u bilo kojem okruženju.
* **Jednostavno i pregledno sučelje** – moderan dizajn, jasne boje i pregledna struktura stranica čine aplikaciju ugodnom za korištenje i motiviraju korisnike na redovito praćenje treninga.

Sve funkcionalnosti međusobno su povezane i čine logičan tijek korištenja: od planiranja, preko bilježenja i praćenja, do analize ostvarenih rezultata. TrainFlow tako pruža zaokruženo rješenje koje podržava dugoročan i sustavan napredak korisnika.

## Primjeri korištenja

Kako bi se bolje prikazala funkcionalnost aplikacije u praksi, u nastavku su prikazane glavne komponente web aplikacije:

* Početna stranica - ova stranica prikazuje uvodnu poruku i glavne prednosti aplikacije. Korisniku se odmah nudi izbor između kreiranja vlastitog plana ili pregleda predloženih. Prikazana je samo hero sekcija, dok se ostatak vidi skrolanjem prema dolje

A person doing push ups

AI-generated content may be incorrect.

Slika . Početna stranica TrainFlowa

Izvor (Autor)

* A screenshot of a login form

  AI-generated content may be incorrect.A screenshot of a login form

  AI-generated content may be incorrect.Login/Register stranice - stranice za prijavu i registraciju ključne su za autentifikaciju korisnika. Na registraciji se unose osnovni podaci, dok prijava omogućava pristup svim personaliziranim funkcionalnostima.

Slika . Register Stranica

Izvor (Autor)

Slika . Login Stranica

Izvor (Autor)

* A screenshot of a computer

  AI-generated content may be incorrect.Stranica za kreiranje plana – omogućava korisniku da definira dane treninga, odabere vježbe i postavi parametre poput serija, ponavljanja, težine i odmora… Prikazan je prvi od tri dijela stranice za kreiranje planova

Slika . Jedan od tri dijelova stranice za kreiranje planova

Izvor (Autor)

* A screenshot of a computer

  AI-generated content may be incorrect.MyProfile stranica – prikazan je dio MyProfile stranice koja se sastoji od informacija o korisniku, njegov BMI (Body Mass Index), te njegove spremljene planove

Slika . Dio MyProfile stranice

Izvor (Autor)

# Tehnologije

U ovom dijelu rada detaljno će biti opisane tehnologije i programski jezici koji su korišteni u razvoju web aplikacije. Razvoj fitness web aplikacije TrainFlow obuhvaća dva ključna segmenta: backend, koji je zadužen za obradu podataka te poslovnu logiku aplikacije i frontend, koji se odnosi na korisničko sučelje i interaktivnost. Također, korištena je i baza podataka za sigurno pohranjivanje i upravljanje podacima korisnika.

## Backend

Backend aplikacije implementiran je u programskom jeziku Java koristeći Spring Boot framework. Ovaj dio odgovoran je za operacije koje nisu vidljive korisnicima, poput rada sa bazom podataka (pohranjivanje, dohvaćanje, ažuriranje podataka), autentifikacije i autorizacije korisnika, implementacije poslovne logike te komunikacije s frontendom putem API-ja. Te tehnologije omogućuju pouzdan i siguran rad aplikacije u pozadini.

### **Java Spring Boot**

Java je objektno orijentirani programski jezik koji je 1995. Godine razvila tvrtka Sun Microsystems, dok ga danas održava Oracle Corporation. Od početka je Java postala jedan od najutjecajnijih i najraširenijih programskih jezika ikad, ponajprije tbog svoje filozofije „write once, run anywhere“(jednom napiši, pokreni svugdje). Ta filozofija omogućila je da programi napisani u Javi mogu raditi na gotovo svim operacijskim sustavima i platformama koje imaju instaliranu Java Virtual Machine (JVM). Time je postignuta visoka razina funkcionalnosti i neovisnosti o platformi, što je bila revolucionarna značajka u vrijeme kada je jezik nastao.

Jedna od ključnih prednosti Jave jest to što je objektno orijentirani jezik. Java je u potpunosti objektno orijentirani jezik. Ovaj jezik koristi klase i objekte za manipulaciju podacima. Jezik Java koristi gotove dijelove klasa ili korisnički definirane klase za izmjenu vrijednosti podataka. [1] To olakšava strukturiranje složenih sustava, ponovno korištenje koda i održavanje aplikacija. Osnovni principi poput enkapsulacije, nasljeđivanja i polimorfizma omogućuju izradu modularnih i proširivih aplikacija. Također, Java nudi bogat standardni paket biblioteka (Java Standard Edition API) koji uključuje funkcionalnosti za rad s kolekcijama, ulazno-izlaznim operacijama, mrežnom komunikacijom, grafičkim sučeljima i mnogim drugim područjima.

A logo of a coffee cup

AI-generated content may be incorrect.

Slika . Java logo

(Izvor: [2])

Sigurnost i stabilnost su velike prednosti Jave. JVM uključuje mehanizme za upravljanje memorijom, poput automatskog „garbage collectiona“, čime se smanjuje mogućnost curenja memorije i drugih grešaka. Osim toga, Java koristi strogu tipizaciju i robustan sustav rukovanja iznimkama, što pridonosi stabilnosti aplikacija. Zahvaljujući ovim značajkama, Java se često koristi u razvoju kritičnih sustava gdje su pouzdanost i sigurnost od najveće važnosti, poput bankarskih sustava, medicinskih aplikacija ili velikih poslovnih rješenja.

S obzirom na svoju skalabilnost, Java se koristi u razvoju širokog spektra aplikacija – od malih desktop aplikacija do velikih enterprise sustava i distribuiranih aplikacija u oblaku. Posebno je popularna u razvoju web aplikacija, gdje se u kombinaciji s različitim frameworkovima (npr. Spring, Hibernate) koristi za izradu robusnih backend sustava. Jedan od razloga dugovječnosti Jave je i njezina aktivna zajednica i ekosustav. Brojne biblioteke, alati i frameworkovi omogućuju programerima da ubrzaju razvoj i jednostavnije implementiraju složene funkcionalnosti. Alati poput Maven-a i Gradle-a olakšavaju upravljanje ovisnostima i automatizaciju procesa izgradnje aplikacija. IDE-ovi poput IntelliJ IDEA, Eclipse i NetBeans dodatno poboljšavaju produktivnost programera nudeći napredne alate za razvoj, testiranje i održavanje.

Zbog svih ovih razloga, Java je korištena za razvoj backend dijela web aplikacije TrainFlow. Njezina pouzdanost, sigurnost, bogata zajednica i kompatibilnost s alatima poput Spring Boot-a čine je idealnim izborom za složeniji sustav koji zahtijeva skalabilnost, stabilnost i dugoročnu održivost.

### **Spring Boot**

Spring Boot je moderan open-source framework razvijen u sklopu šireg Spring ekosustava s ciljem pojednostavljenja izrade i implementacije web aplikacija temeljenih na Javi. Prvi put je predstavljen 2014. godine od strane tvrtke Pivotal Software, a ubrzo je postao jedan od najpopularnijih alata za razvoj backend sustava. Danas je Spring *de facto* okvir za razvoj u Javi, a tvrtke diljem svijeta koriste ga u svojim sustavima. [3] Glavna ideja Spring Boot-a jest eliminirati složenost konfiguracije tipične za tradicionalni Spring framework te omogućiti brzu i jednostavnu izradu aplikacija koje su spremne za produkciju.

Jedna od najvećih prednosti Spring Boot-a je automatizirana konfiguracija. Dok je u klasičnom Spring frameworku programer morao ručno konfigurirati veliki broj parametara (npr. povezivanje s bazom podataka, definiranje bean-ova…), Spring Boot prepoznaje kontekst aplikacije i automatski konfigurira potrebne komponente. Time se značajno smanjuje količina koda potrebna za inicijalizaciju projekta, a razvoj postaje brži i učinkovitiji.

Spring Boot dolazi s integriranim ugrađenim aplikacijskim poslužiteljima (Tomcat, Jetty, Undertow), što znači da aplikaciju nije potrebno dodatno implementirati na vanjski server – dovoljno je pokrenuti aplikaciju kao standardnu Java aplikaciju. To omogućuje jednostavno testiranje, lokalno pokretanje i brzo postavljanje u produkciju. Za komunikaciju između frontend i backend dijela, Spring Boot omogućuje jednostavnu izradu REST API-ja. Korištenjem anotacija poput @RestController, @RequestMapping i @GetMapping, programeri mogu vrlo brzo definirati rute i metode za obradu korisničkih zahtjeva. Također, Spring Boot pruža integriranu podršku za JSON format, što je ključno za razmjenu podataka između aplikacija.

A screen shot of a computer program

AI-generated content may be incorrect.

Slika . Prikaz Training Controllera

Izvor (Autor)

Na slici je prikazan dio Spring Boot kontrolera koji služi za rukovanje zahtjevima vezanim uz trening planove. Klasa je anotirana s @RestController, što znači da izlaže REST API metode, te s @RequestMapping koji definira osnovni URL put za sve metode u klasi. Korištenjem različitih anotacija poput @GetMapping i @PostMapping definiraju se krajnje točke API-ja za dohvaćanje predloženih planova (getSuggestedPlans), dohvaćanje kompletnog plana prema identifikatoru (getFullTrainingPlan), dohvaćanje korisnikovih planova (getUserTrainingPlans) te spremanje plana za korisnika (savePlanForUser). Ove metode komuniciraju sa servisnim slojem aplikacije, čime se postiže jasna odvojenost odgovornosti između kontrolera i poslovne logike. Ovaj primjer ilustrira osnovnu strukturu i način rada REST kontrolera u Spring Boot okruženju.

Posebna prednost Spring Boot-a leži u njegovom sigurnosnom modulu – Spring Security. On omogućuje implementaciju autentifikacije i autorizacije korisnika, zaštitu API-ja te primjenu sigurnosnih standarda kao što su JWT (JSON Web Token) i OAuth2. Time se osigurava da samo ovlašteni korisnici imaju pristup određenim funkcionalnostima aplikacije, što je posebno važno za aplikacije poput TrainFlowa, gdje se obrađuju osobni podaci korisnika.

Zbog svega navedenog, Spring Boot je odabran kao temelj backend dijela aplikacije TrainFlow. Njegova jednostavnost, sigurnost, podrška za rad s bazama podataka i integracija s modernim tehnologijama čine ga idealnim rješenjem za aplikaciju koja mora biti sigurna, skalabilna i jednostavna za održavanje.

### **MySQL Baza podataka**

MySQL jedan je od najpopularnijih sustava za upravljanje relacijskim bazama podataka (RDBMS) otvorenog koda. Razvila ga je švedska tvrtka MySQL AB 1995. godine, a kasnije ga je preuzeo Oracle Corporation. MySQL se koristi za pohranjivanje, organiziranje i upravljanje podacima te je temelj mnogih modernih web aplikacija. Ako je vjerovati mnogim strastvenim obožavateljima MySQL-a, MySQL je brži, pouzdaniji i jeftiniji – ili, jednostavno rečeno, bolji od bilo kojeg drugog sustava baza podataka (uključujući komercijalne sustave poput Oraclea i DB2-a) [4].

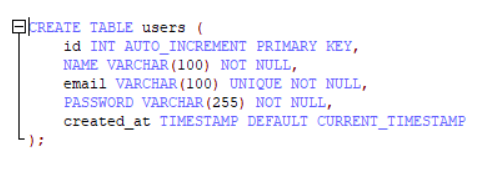
Njegova jednostavnost, pouzdanost i brzina čine ga idealnim izborom za širok spektar projekata, od malih web stranica do kompleksnih sustava koji zahtijevaju velike količine podataka. Jedna od ključnih karakteristika MySQL-a njegova je relacijska struktura, koja organizira podatke u tablice vezane ključevima. Ovo omogućuje učinkovito modeliranje podataka te izradu složenih odnosa između različitih entiteta, što je korisno za aplikacije poput fitness sustava koje moraju pohranjivati podatke o korisnicima, fizičkim aktivnostima, unosu kalorija, ciljevima i slično.

**Prednosti MySQL-a**:

* **Brzina i Performanse**: MySQL dizajniran je da pruži visoke performanse čak i pri velikim operećenjima. Njegov optimizirani sustav za izvršavanje upita omogućuje brzo dohvaćanje i obradu podataka.
* **Podrška za velike količine podataka**: Može upravljati velikim bazama podataka te istovremeno rukovati s više korisnika bez utjecaja na performanse.
* **Jednostavno korištenje**: MySQL ima intuitivnu sintaksu koja olakšava rad programerima. Većina funkcionalnosti može se implementirati kroz jednostavne SQL naredbe.
* **Sigurnost**: Pruža napredne sigurnosne mehanizme, uključujući autentifikaciju korisnika, kontrole pristupa te enkripciju podataka.
* **Podrška za više platformi**: MySQL je kompatibilan s različitim operativnim sustavima, uključujući Windows, Linux i macOS, što ga čini fleksibilnim za razvoj u različitim okruženjima.

MySQL omogućuje izvršavanje raznih vrsta upita za manipulaciju podacima, uključujući:

* **SELECT**: Dohvaća podatke iz tablica
* **INSERT**: Dodaje nove zapise u tablice
* **UPDATE**: Ažurira postojeće zapise
* **DELETE**: Briše zapise
* **JOIN**: Povezuje podatke iz više tablica



Slika . Primjer stvaranja tablice u MySQL-u

Izvor (Autor)

Na slici je prikazan SQL upit kojim se kreira tablica users u bazi podataka. Tablica sadrži nekoliko ključnih polja: id služi kao primarni ključ i automatski se povećava za svakog novog korisnika, name pohranjuje ime korisnika i ne smije ostati prazno, dok je email definiran kao jedinstven i obavezan atribut kako bi se spriječila registracija istih adresa više puta. Lozinka se sprema u polju password s većom dužinom (255 znakova) kako bi bilo dovoljno prostora za hashirane vrijednosti. Konačno, atribut created\_at automatski bilježi datum i vrijeme kreiranja zapisa pomoću CURRENT\_TIMESTAMP funkcionalnosti. Ovakva struktura predstavlja temelj za sigurno i pouzdano upravljanje korisničkim računima.

### **Primjer korištenja baze podataka u TrainFlowu**

U radu je korištena baza podataka za pohranjivanje i upravljanje ključnim podacima, poput informacija o korisnicima, njihovim treninzima, kreiranim planovima te postavljenim ciljevima. Na taj način omogućeno je učinkovito praćenje i analiza podataka, što je nužno za personalizirani pristup i praćenje napretka korisnika aplikacije.

U aplikaciji TrainFlow implementiran je rad s MySQL bazom podataka koristeći Spring Data JPA i Hibernate ORM. Ovaj pristup omogućuje jednostavnu manipulaciju podacima kroz Java objekte bez potrebe za ručnim pisanjem SQL upita. JPA koristi repository sloj, koji je povezan s tablicama baze podataka, dok se instance entiteta automatski povezuju s pojedinim redovima u tim tablicama. Omogućuje besprijekornu integraciju s relacijskim bazama podataka pomoću objektno-relacijskog mapiranja (ORM), čime se uklanja potreba za ponavljajućim SQL upitima. [5] Za rad s podacima koriste se metode poput spremanja, ažuriranja, brisanja i dohvaćanja, koje Spring Data generira automatski na temelju definiranih sučelja.

Osnovne metode za rad s bazom podataka:

* save() – spremanje novih podataka ili ažuriranje postojećih
* deleteById() – brisanje zapisa prema ID-u
* findById() – dohvaćanje zapisa na temelju ID-a
* findAll() – dohvaćanje svih zapisa

Primjerice, za unos novih korisničkih aktivnosti u tablicu *workout\_logs* koristi se metoda save(), gdje se objekt koji sadrži podatke o aktivnosti (trajanje, potrošene kalorije, vrsta vježbe) automatski mapira na polja tablice. Hibernate pri tome vodi brigu o povezivanju objekta s odgovarajućim SQL upitima.

Dodatno, JPA omogućuje definiranje vlastitih metoda koristeći Query metode ili JPQL. Na primjer:

* findByUserId() – dohvaćanje svih aktivnosti određenog korisnika
* findByDateBetween() – dohvaćanje aktivnosti u zadanom vremenskom rasponu

Dinamičnost baze podataka

Korištenje Spring Boot migracija uz pomoć Hibernate-a i opcionalno alata poput Flyway ili Liquibase omogućuje jednostavno proširivanje i prilagođavanje strukture baze podataka. Dodavanje novih polja ili tablica može se implementirati bez narušavanja postojećih funkcionalnosti, što olakšava daljnji razvoj i održavanje aplikacije.

A screen shot of a computer program

AI-generated content may be incorrect.

Slika . Prikaz dodavanja nove aktivnosti u bazu podataka pomoću Eloquent ORM-a

Izvor (Autor)

## Frontend

Frontend dio aplikacije odgovoran je za prikaz korisničkog sučelja i interakciju s korisnicima. Njegova glavna svrha je osigurati jednostavno i jasno korisničko iskustvo (engl. User Experience – UX) kroz intuitivan dizajn i preglednu navigaciju. Frontend razvoj ključan je za estetiku i funkcionalnost aplikacije te obuhvaća tri temeljne tehnologije:

* **HTML (HyperText Markup Language):** koristi se za strukturiranje sadržaja stranice, uključujući naslove, obrasce, kartice s planovima treninga i ostale elemente.
* **CSS (Cascading Style Sheets):** zadužen je za stilizaciju elemenata HTML-a, poput boja, fontova i rasporeda elemenata. U ovom projektu CSS je korišten za prilagodbu dizajna i postizanje modernog izgleda fitness aplikacije.
* **TypeScript/Angular:** omogućuje interaktivnost aplikacije, dinamičko ažuriranje sadržaja i povezivanje s backendom putem API poziva. U projektu je korišten Angular framework, koji olakšava razvoj komponenti i organizaciju koda.

Angular pruža modularnu strukturu te omogućuje lako rukovanje podacima s backend dijela aplikacije. Na ovaj način frontend nije samo prezentacijski sloj, već aktivno komunicira s backendom, omogućujući korisniku rad s trening planovima, bilježenje odrađenih vježbi i praćenje napretka.

### **HTML**

HTML (HyperText Markup Language) osnovni je jezik za izradu web stranica i aplikacija. Prva verzija izašla je 1993. godine, a napisao ju je Tim Berners-Lee. Predstavlja temeljnu tehnologiju na kojoj se grade svi ostali slojevi web stranica, uključujući dizajn i funkcionalnost. HTML omogućuje strukturiranje sadržaja i definiranje elemenata poput naslova, odlomaka, slika, tablica, obrazaca i poveznica.

HTML nije programski jezik nego označni jezik. Označni jezik sastoji se od označnih kodova [6]. Svaka oznaka u HTML-u obično se sastoji od otvarajuće i zatvarajuće oznake, unutar kojih se nalazi sadržaj ili atributi. Atributi pružaju dodatne informacije o elementu, poput stila, identifikatora ili klase te omogućuju njegovu prilagodbu.

Danas je u upotrebi najnovija verzija HTML-a, HTML5. HTML5 donosi brojne napredne značajke, poput podrške za multimedijske sadržaje, bolje semantičke oznake i mogućnosti lokalnog spremanja podataka. Ove značajke omogućuju izradu dinamičnih i interaktivnih web stranica bez potrebe za dodatnim tehnologijama.

Glavne značajke HTML-a:

1. **Strukturiranje sadržaja** – HTML omogućuje organizaciju sadržaja pomoću elemenata poput naslova (<h1> do <h6>), odlomaka (<p>), popisa (<ul> i <ol>), tablica (<table>), slika (<img>) i mnogih drugih elemenata.
2. **Multimedijalna podrška** - HTML5 uvodi elemente poput <audio> i <video>, čime je postalo moguće direktno ugraditi audio i video sadržaj u web stranice, bez potrebe za vanjskim dodacima.
3. **Forme za unos podataka** - HTML omogućuje kreiranje obrazaca pomoću elemenata poput <form>, <input>, <textarea> i <button>, što je ključno za interaktivne aplikacije koje omogućuju unos korisničkih podataka.
4. **Semantičke oznake** - Uvođenjem elemenata poput <header>, <footer>, <article> i <section>, HTML5 omogućuje bolje strukturiranje i razumijevanje sadržaja web stranica, čime poboljšava pristupačnost i optimizaciju za tražilice (SEO).

A screenshot of a computer program

AI-generated content may be incorrect.

Slika . Prikaz dijela početne stranice u HTML-u

Izvor (Autor)

Na slici je prikazan isječak HTML koda koji prikazuje dio strukture početne stranice aplikacije. Možemo vidjeti hero sekciju s glavnim naslovom i gumbom za prijavu, kao i sekciju s prikazom glavnih funkcionalnosti aplikacije (workout plan, workout tracker…). Ovaj kod predstavlja osnovu korisničkog sučelja prije nego što se na njega primijene stilovi (CSS) i funkcionalnosti (Angular/TypeScript).

**Primjena u TrainFlow-u:**

U ovom radu HTML služi za definiranje osnovne strukture korisničkog sučelja aplikacije usmjerene na treninge i praćenje napretka. Korišten je za oblikovanje stranica poput početne stranice, profila korisnika, kreiranja i prikaza planova treninga te stranice za praćenje odrađenih vježbi. Osim toga, HTML omogućuje jasnu navigaciju unutar aplikacije kroz izbornike i poveznice, čime se osigurava jednostavno korištenje.  
HTML predstavlja temeljnu tehnologiju koja omogućuje interakciju između korisnika i aplikacije te služi kao podloga za CSS (stilizaciju) i Angular/TypeScript logiku koja dodaje interaktivnost i dinamičko upravljanje podacima.

### **CSS**

CSS (Cascading Style Sheets) tehnologija je koja se koristi za stilizaciju i oblikovanje web stranica. Osnovna namjena CSS-a je da web dizajnerima omogući definiranje stilova, tj. detalja o oblikovanju kao što su raspored, veličine i boje elemenata, boje pozadine, a zatim te stilove primjenjuju na izabrane dijelove HTML stranica pomoću selektora, referenci na element, ili grupu elemenata za koje je stil predviđen [7]. Uveden je 1996. godine kako bi odvojio sadržaj od dizajna, čime je omogućena veća fleksibilnost i kontrola nad izgledom web stranica.

Korištenjem CSS-a možemo definirati izgled elemenata poput boja, fontova, razmaka, veličina i rasporeda elemenata na stranici. CSS je ključan za stvaranje vizualno privlačnih i responzivnih web stranica koje se prilagođavaju različitim uređajima, poput računala, tableta i mobilnih telefona.

Prednosti korištenja CSS-a:

* **Odvajanje sadržaja i dizajna**: Omogućuje lakše održavanje koda jer se dizajn definira u zasebnim datotekama.
* **Ponovna upotreba koda**: Jedan CSS stil može se primijeniti na više HTML stranica, što olakšava konzistentnost dizajna.
* **Fleksibilnost i responzivnost**: CSS omogućuje prilagođavanje web stranica različitim veličinama zaslona i rezolucijama.
* **Animacije i efekti**: Pomoću CSS-a moguće je dodati animacije, tranzicije i druge vizualne efekte bez potrebe za korištenjem JavaScripta.

Vrste CSS-a:

1. **Inline CSS**: Stilovi se primjenjuju unutar samog HTML elementa (npr., style atribut).
2. **Internal CSS**: Stilovi se definiraju unutar <style> oznake u zaglavlju HTML dokumenta.
3. **External CSS**: Stilovi se definiraju u vanjskoj datoteci (npr., style.css) i povezuju s HTML dokumentom putem <link> oznake.

CSS je u radu korišten za stilizaciju svih elemenata aplikacije, čime se osiguralo da je sučelje estetski privlačno i jednostavno za korištenje.

A computer screen shot of a black screen

AI-generated content may be incorrect.

Slika . Prikaz jednostavnog koda u CSS-u

Izvor (Autor)

Na slici je prikazan CSS kod za početni (hero) dio aplikacije. U ovom dijelu definira se pozadinska slika, osnovni raspored elemenata pomoću flexboxa te osnovni stil naslova, teksta i gumba. Ovakav dizajn omogućuje moderan izgled početne stranice i odmah daje korisniku jasnu poruku o svrsi aplikacije.

### **JavaScript/TypeScript**

JavaScript je programski jezik koji omogućuje dodavanje interaktivnosti i dinamičkog ponašanja web stranicama i aplikacijama. Njegova glavna uloga je povezivanje statičnog HTML-a i stilizacije u CSS-u s funkcionalnostima koje korisnicima omogućuju ugodnije i efikasnije korištenje aplikacije. Danas se koristi i na klijentskoj i na serverskoj strani, a u modernom razvoju često se nadopunjuje TypeScriptom, nadogradnjom JavaScripta koja donosi tipizaciju i bolju organizaciju koda. Pisanje koda prilično je jednostavno. TypeScript omogućava upotrebu velikog broja alata poput auto-kompletiranja i jednostavnog refaktoringa već napisanog koda. [8]

U ovom projektu koristi se Angular framework, koji je temeljen na TypeScriptu. Angular omogućuje razvoj modularnih komponenti, upravljanje obrascima, validaciju podataka i jednostavnu komunikaciju s backendom putem HTTP zahtjeva.

Prednosti primjene:

* **Interaktivnost:** unos i validacija podataka u realnom vremenu (npr. kod kreiranja i praćenja trening planova).
* **Dinamičnost:** automatsko prikazivanje aktivnog plana i vježbi za trenutni dan.
* **Modularnost:** komponente su odvojene i lako se održavaju te proširuju.
* **Integracija s backendom:** dohvaćanje i spremanje podataka putem REST API-ja.

Primjena u aplikaciji:

JavaScript/TypeScript koriste se za logiku kreiranja i uređivanja planova treninga, prikaz dnevnih vježbi te bilježenje odrađenih treninga u Workout Trackeru. Na taj način frontend nije samo prezentacijski sloj, već aktivno sudjeluje u radu aplikacije, povezujući korisnika s podacima i funkcionalnostima backend sustava.

A screen shot of a computer

AI-generated content may be incorrect.

Slika . Prikaz dijela TypeScript dokumenta Your Training Plans stranice

Izvor (Autor)

Na slici je prikazan TypeScript kod Angular komponente YourPlansComponent. Ova komponenta zadužena je za upravljanje korisničkim planovima treninga. Implementira OnInit kako bi se prilikom inicijalizacije dohvatili svi korisnički kreirani planovi i spremljeni planovi iz baze putem servisa AppService.

Bitni dijelovi:

* getFullTrainingPlan(planId: number) dohvaća puni plan treninga prema ID-u i sprema ga u selectedPlan.
* getSavedPlans() dohvaća sve planove spremljene za korisnika i pohranjuje ih u savedPlans.
* Servis koristi Observable pattern (subscribe) za asinkroni rad s podacima i rukovanje pogreškama.

Ukratko: komponenta služi kao poveznica između korisničkog sučelja i backend podataka za prikaz i odabir trening planova.

# Izrada dizajna

## Ciljevi i ciljane skupine

Prije početka izrade aplikacije važno je jasno definirati glavne ciljeve i odrediti ciljane skupine korisnika. Dobro postavljeni ciljevi usmjeravaju razvoj aplikacije tako da zadovoljava stvarne potrebe korisnika, dok razumijevanje publike omogućuje kreiranje rješenja prilagođenog njihovim očekivanjima i navikama.

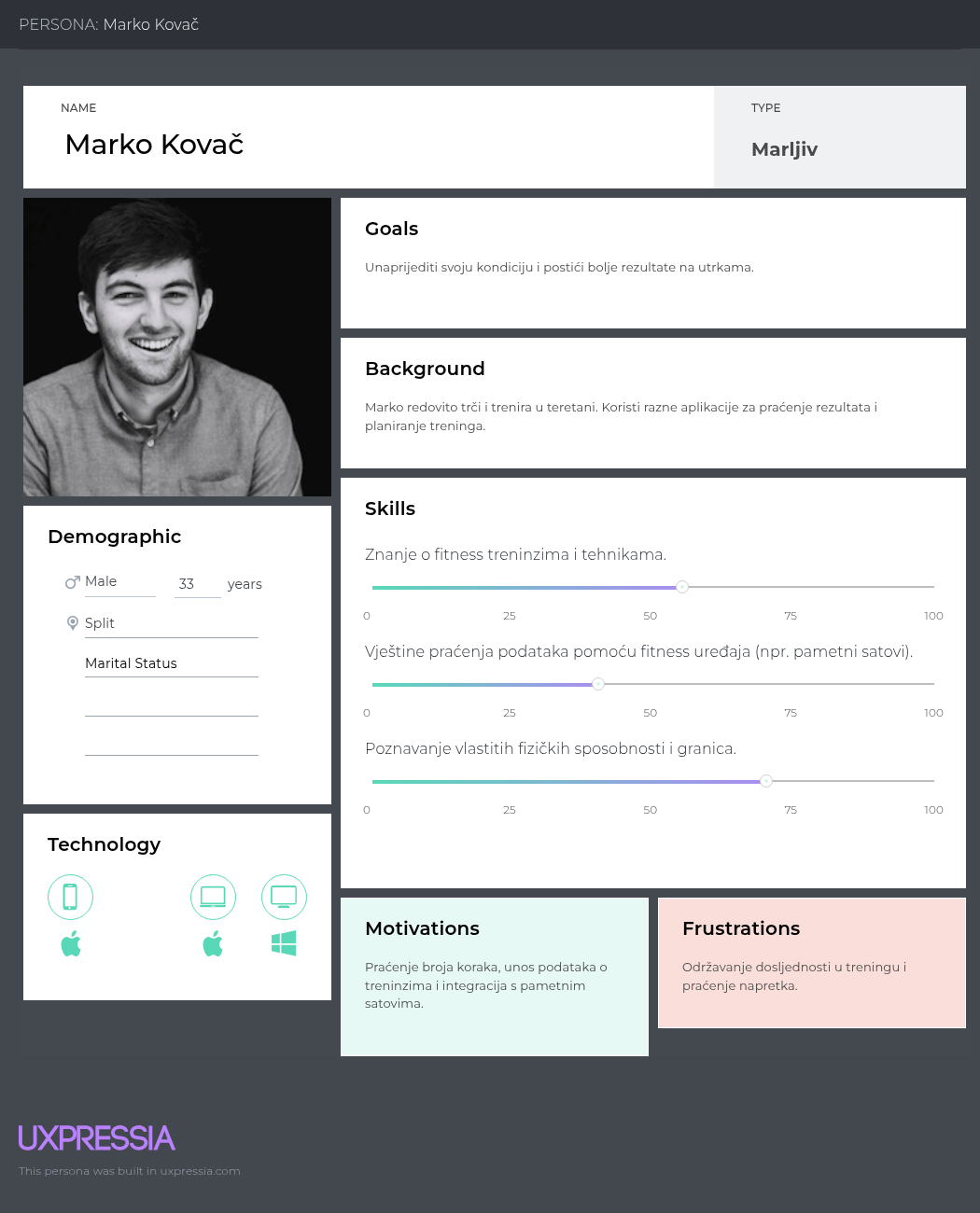
Glavni cilj aplikacije je omogućiti korisnicima praćenje i organizaciju vlastitih treningplanova na jednostavan i učinkovit način. Ključni ciljevi uključuju:

* **Praćenje treninga**: korisnici mogu unositi i spremati podatke o odrađenim vježbama, serijama, ponavljanjima i opterećenju.
* **Izrada i upravljanje planovima**: omogućeno je kreiranje personaliziranih planova treninga prema ciljevima i razini iskustva.
* **Postavljanje i praćenje napretka**: aplikacija pomaže u praćenju ostvarenih rezultata u odnosu na zadane ciljeve.
* **Jednostavno korisničko iskustvo**: pregledno i intuitivno sučelje omogućuje lako snalaženje i brzi pristup funkcijama.

Ciljana skupina aplikacije su početnici u fitnessu, iskusni rekreativci, sportaši koji žele strukturirane treninge, te svi korisnici kojima je cilj poboljšati kondiciju, snagu ili izgled kroz vođeni plan vježbanja.

## Persone

Persone predstavljaju fiktivne, ali realistične profile korisnika koji će koristiti aplikaciju. Njihova svrha je bolje razumijevanje potreba, ciljeva i izazova ciljane publike kako bi se osigurala što bolja korisnička iskustva. Izradom personi mogu se lakše donijeti odluke o funkcionalnostima, dizajnu i prioritetima u razvoju aplikacije.



Slika . Prikaz persone za TrainFlow

Izvor (Autor)

Persone pomažu timu za razvoj da se fokusira na stvarne potrebe korisnika, što dovodi do bolje prilagođene aplikacije i zadovoljnijih korisnika. Svaka persona ima jedinstvene karakteristike koje utječu na način kako koriste aplikaciju i koje funkcionalnosti su im najvažnije. U kontekstu fitness aplikacije, razumijevanje personi pomaže u prilagodbi korisničkog sučelja, funkcionalnosti i sadržaja kako bi se osiguralo pozitivno korisničko iskustvo.

A screenshot of a computer

Description automatically generated

Slika . Primjer druge persone za TrainFlow

Izvor (Autor)

## Sitemap ili Arhitektura Informacija

*Site map* (ili *Sitemap*) je datoteka koja sadrži listu svih stranica koje se nalaze na određenoj adresi i koje su dostupne korisniku [9]. Prikazuje strukturu aplikacije ili web stranice, odnosno način na koji su sadržaji i funkcionalnosti organizirani. Cilj izrade sitemap-a je olakšati navigaciju korisnicima te osigurati da svi ključni dijelovi aplikacije budu lako dostupni. Dobro osmišljena arhitektura informacija ključna je za pozitivno korisničko iskustvo jer im pomaže u snalaženju unutar aplikacije.

Sitemap zavisi o korisnicima, sadržaju i kontekstu. To znaci da strukturiranje informacija ovisi o korisničkim potrebama, poslovnim ciljevima te različitim tipovima sadržaja. Prikazuje se pomoću grafova ili Excel tablica. Sitemap nam pomaže u vizualiziranju cjelokupne strukture aplikacije te planiranju razvoja svakog odjeljka na logičan način. Također, služi i kao vodič za timove zadužene za dizajn i implementaciju, omogućujući konzistentnost u izradi i navigaciji unutar aplikacije.

A group of rectangular objects with text

AI-generated content may be incorrect.

Slika . Prikaz sitemap-a za TrainFlow

Izvor (Autor)

Vrste sitemap-a:

* Vizualni sitemap: Grafički prikazuje hijerarhiju stranica pomoću dijagrama, što olakšava razumijevanje odnosa među stranicama.
* HTML sitemap: Korisnički pristupačna verzija sitemapa, često integrirana u web stranice kako bi posjetiteljima omogućila jednostavan pregled i brzi pristup sadržajima.
* XML sitemap: Tehnička verzija koja se koristi za indeksiranje stranica od strane tražilica poput Googlea, čime se poboljšava SEO optimizacija.

Osim što pruža pregled strukture, sitemap omogućuje identificiranje potencijalnih problema u organizaciji sadržaja prije nego što započnu faze dizajna i implementacije. U procesu izrade aplikacije, sitemap služi kao komunikacijski most između dizajnera, programera i ostalih članova tima. Jasno definirana hijerarhija stranica i povezanost među njima osigurava konzistentnost tijekom razvoja, smanjuje nesporazume i ubrzava proces donošenja odluka.

## Wireframe ili žičani kostur

Wireframe (žičani kostur) predstavlja osnovni vizualni prikaz strukture web aplikacije, pružajući pregled rasporeda elemenata na stranici bez detaljnog dizajna i boja. Glavna svrha wireframe-a je definirati pozicije elemenata poput navigacije, sadržaja, formi i grafičkih komponenti, kako bi se omogućilo planiranje korisničkog sučelja (User Interface - UI) i korisničkog iskustva (User Experience – UX). Koristeći **wireframe** stvarate jednostavnu strukturu i prikaz funkcionalnosti koje buduća stranica nudi, a kao takvog lako ga je izmijeniti po potrebama i zahtjevima klijenata [10].

Wireframe ne uključuje detalje poput boja, fontova ili dizajnerskih elemenata. Umjesto toga, fokusira se na funkcionalnost i logiku rasporeda. Izrada wireframe-a pomaže u ranoj fazi razvoja kako bi svi koji rade na projektu imali jasan pregled kako će aplikacija izgledati i funkcionirati. Korištenjem alata poput Figme i Adobe XD-a, wireframe olakšava razumijevanje korisničkih potreba i optimizira process razvoja web aplikacije.

*Važnost wireframe-a:*

1. **Jasna vizija projekta**: Uz pomoć wireframe-a, dobiva se uvid u osnovnu strukturu aplikacije.
2. **Ušteda vremena i resursa**: Planiranje izgleda i funkcionalnosti u ranoj fazi sprječava potencijalne pogreške u kasnijim fazama razvoja.
3. **Testiranje korisničkog iskustva**: Wireframe omogućuje analizu protoka korisnika kroz aplikaciju, identificiranje potencijalnih problema i poboljšanje navigacije.
4. **Fokus na funkcionalnost**: Umjesto estetike, naglasak je na logici i funkcionalnosti aplikacije

A screenshot of a website

AI-generated content may be incorrect.

Slika . Prikaz početne stranice pomoću wireframe-a

Izvor (Autor)

Wireframe je ključna faza u dizajnu aplikacije koja omogućuje strukturirano planiranje i vizualizaciju korisničkog sučelja. U slučaju TrainFlow-a, wireframe omogućuje optimalnu organizaciju elemenata koji su usmjereni na poboljšanje korisničkog iskustva, jasnoću navigacije i efikasnost interakcije. Izrada kvalitetnog wireframea osigurava stabilan temelj za daljnji razvoj aplikacije i smanjuje mogućnost pogrešaka u implementaciji.

## Odabir boja

Boje igraju ključnu ulogu u vizualnom identitetu aplikacije i korisničkom iskustvu. Pravilno odabrana paleta boja može privući korisnika, povećati čitljivost te olakšati navigaciju unutar aplikacije. Boje izazivaju određene emocionalne reakcije kod korisnika, što ih čini važnim alatom u dizajnu. Kod dizajniranja TrainFlowa, važno je uzeti u obzir funkcionalnost i psihološki utjecaj boja.

Odabir boja za TrainFlow igra ključnu ulogu u stvaranju privlačnog vizualnog identiteta i pozitivnog korisničkog iskustva. Prilikom dizajniranja trening web aplikacije, odabrane su četiri osnovne boje: **zelena**, **plava**, **bijela** **i crne** koje zajedno stvaraju skladnu i profesionalnu paletu boja. Zelena boja simbolizira zdravlje, prirodu i rast, što je idealno za aplikaciju koja promovira fitness i zdrav način života. Plava boja donosi osjećaj povjerenja, stabilnosti i smirenosti. Bijela boja predstavlja čistoću, jednostavnost i preglednost. Crna boja unosi osjećaj elegancije i snage.

A row of different colors

Description automatically generated

Slika . Palete boja korištene u aplikaciji

Izvor (Autor)

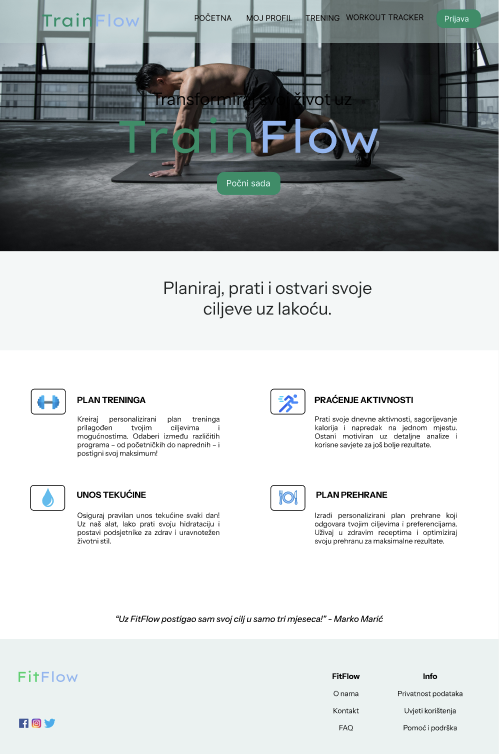
Kombinacija ovih boja omogućuje balans između dinamičnosti i smirenosti. Zajedno stvaraju profesionalan i moderan izgled koji odražava svrhu aplikacije – unapređenje zdravlja i fitnessa. Koristeći zelenu za akcijske elemente, plavu za povjerenje, bijelu za preglednost te crnu za eleganciju, postiže se skladan i funkcionalan dizajn koji korisnicima pruža ugodno iskustvo. Uz njihovu pravilnu primjenu, aplikacija ostavlja pozitivan dojam i pomaže korisnicima u postizanju njihovih fitness ciljeva.

## Izrada prototipa u Figmi

Figma je moćan alat za dizajn korisničkog sučelja i prototipiranje, popularan među dizajnerima zbog svoje intuitivnosti, kolaborativnih mogućnosti i rada u oblaku. U razvoju TrainFlow web aplikacije, Figma će se koristiti za izradu vizualnog prototipa stranica kako bi se olakšalo planiranje dizajna i osiguralo bolje korisničko iskustvo.

Prednosti Figme:

* **Rad u oblaku**: Nije potrebna instalacija na računalu, već su svi projekti dostupni s bilo kojeg uređaja putem internetskog preglednika. To omogućuje jednostavan pristup i sigurnu pohranu dizajna [11].
* **Kolaboracija u stvarnom vremena**: Prvo, Figma omogućuje dizajnerima i drugim članovima tima da rade simultano u stvarnom vremenu, što podiže suradnički tijek rada na potpuno novu razinu.
* **Prototipiranje**: Figma podržava interaktivno prototipiranje, što znači da se mogu simulirati funkcionalnosti aplikacije, poput navigacije između stranica ili reakcija na klikove.
* **Intuitivno korisničko sučelje**: Figma je prilagođena i početnicima i iskusnim dizajnerima, s bogatim setom alata za kreiranje elemenata poput gumba, obrazaca, slika i tekstova.



Slika . Prototip početne stranice TrainFlow

Izvor (Autor)

Za web aplikaciju TrainFlow Figma će se koristiti za izradu prototipa stranica poput početne stranice, profila korisnika te ostale stranice. Također će se koristiti za testiranje dizajna kako bi se osiguralo da je aplikacija vizualno privlačna i responzivna. Između ostalog, Figma će nam biti korisna za definiranje elemenata dizajna poput paleta boja, tipografija i slično.

# Zaključak

Završni rad na temu razvoja web aplikacije **TrainFlow** pokazao je važnost kvalitetnog planiranja, odabira tehnologija i usmjerenosti na korisničko iskustvo u procesu izrade moderne aplikacije. U radu su obrađeni ključni segmenti razvoja, uključujući izradu frontend i backend dijela, dizajn sučelja te implementaciju funkcionalnosti koje odgovaraju potrebama ciljanih korisnika.

Glavna svrha TrainFlow-a je omogućiti korisnicima jednostavno kreiranje, upravljanje i praćenje vlastitih planova treninga. Poseban naglasak stavljen je na pregledno i responzivno sučelje, jasnoću korištenja te funkcionalnosti koje olakšavaju svakodnevno praćenje napretka. TrainFlow nudi alate za izradu personaliziranih trening planova, praćenje odrađenih vježbi i vođenje dnevnika treninga, čime korisnicima pruža podršku u postizanju njihovih fitness ciljeva.

Za razvoj aplikacije korištene su suvremene tehnologije: **Angular** za frontend, **Spring Boot** za backend te **MySQL** baza podataka za sigurno i strukturirano pohranjivanje podataka. Ova kombinacija omogućila je stabilan rad aplikacije, dobru organizaciju podataka te intuitivno korisničko iskustvo. Proces razvoja obuhvatio je i planiranje strukture podataka, validaciju korisničkih unosa te dizajn elemenata koji pridonose profesionalnom i modernom izgledu aplikacije.

Rad na TrainFlow aplikaciji pružio je vrijedno praktično iskustvo u implementaciji teorijskog znanja, kao i u rješavanju stvarnih problema korisnika putem tehnologije. U konačnici, TrainFlow predstavlja funkcionalnu platformu koja pomaže korisnicima da prate svoj napredak i ostanu motivirani na putu prema zdravijem i aktivnijem životu. U budućnosti, aplikacija ima potencijal proširenja kroz dodatne mogućnosti poput napredne analitike, integracije s pametnim uređajima ili personaliziranih preporuka temeljenih na korisničkim podacima.

# Literatura

|  |  |
| --- | --- |
| [1] | S. S. Khandare, Programming in Java, S. Chand Limited, 2010. |
| [2] | “https://iconscout.com/,” [Online]. Available: https://iconscout.com/free-icon/java-60\_1174953. [Accessed 19 08 2025]. |
| [3] | C. E. de Oliveira, G. Turnquist and A. Antonov, Developing Java Applications with Spring and Spring Boot, Packt Publishing, 2018. |
| [4] | M. Kofler, The Definitive Guide To MySQL, Berkeley: Apress, 2003, pp. 6-7. |
| [5] | “geeksforgeeks.org,” 2025. [Online]. Available: https://www.geeksforgeeks.org/springboot/what-is-spring-data-jpa/. [Accessed 19 08 2025]. |
| [6] | »mojwebdizajn,« [Mrežno]. Available: mojwebdizajn.net. [Pokušaj pristupa 13 january 2025]. |
| [7] | S. V. Štajdohar, M. Tomiša and D. Valdec, “Oblikovanje web stranica primjenom CSS-a 3,” *Tehnički glasnik,* vol. 7, no. 4, pp. 1-2, 2013. |
| [8] | WeAreDevelopers, “debug.hr,” 29 04 2021. [Online]. Available: https://www.debug.hr/sve-sto-trebate-znati-o-programskom-jeziku-typescript/. [Accessed 19 08 2025]. |
| [9] | S. Brekalo and M. Vrtarić, “Optimizacija web stranica za internet pretraživače,” *Zbornik radova Međimurskog veleučilišta u Čakovcu,* vol. 2, no. 2, pp. 21-25, 2011. |
| [10] | »marketingiraj.me,« [Mrežno]. Available: https://marketingiraj.me/sto-wireframe-zasto-vazan/. [Pokušaj pristupa 17 January 2025]. |
| [11] | F. Staiano, “Designing and Prototyping Interfaces with Figma,” Birmingham, Packt Publishing, 2022, p. 382. |