Izrada web aplikacije za unos i praćenje rezultata na karate natjecanjima

Veršić, Karlo

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Prijediplomski studij računarstva

Završni rad

Izrada web aplikacije za unos i praćenje rezultata na karate natjecanjima

Rijeka, rujan 2023.

Karlo Veršić
0069085382

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Prijediplomski studij računarstva

Završni rad

Izrada web aplikacije za unos i praćenje rezultata na karate natjecanjima

Mentor: prof. dr. sc. Ivan Štajduhar

Rijeka, rujan 2023.

Karlo Veršić
0069085382

Umjesto ove stranice umetnuti zadatak za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2023.

_____
Ime Prezime

Zahvala

Zahvaljujem mentoru prof. dr. sc. Ivan Štajduharu. Zahvaljujem asistentu Arianu Skokiju na podršci tijekom pisanja ovoga rada i korisnim raspravama i savjetima. Posebno zahvaljujem obitelji, prijateljima i djevojci na podršci tijekom studiranja.

Sadržaj

Popis slika **viii**

1 Uvod **1**

2 Pregled područja **3**

2.1 Karate 3

2.1.1 Vrste natjecanja 4

2.1.2 Natjecateljske discipline 4

2.2 Excel tablice 6

2.3 Portal ekarate.eu 7

2.4 Sportdata Event Technology 7

3 Korištene tehnologije **9**

3.1 Django 9

3.2 PostgreSQL 10

3.3 Tailwind CSS 11

3.4 Docker 11

4 Projektna aplikacija **13**

4.1 Baza podataka 13

4.1.1 Autentikacija 14

4.1.2 Poslovna logika 16
4.2 Implementacija poslužitelja 20
4.2.1 Postavljanje PostgreSQL baze podataka 20
4.2.2 Autentikacija i autorizacija korisnika 22
4.2.3 Konfiguracija statičnih datoteka 26
4.2.4 Implementacija ostalih funkcionalnosti 27
4.3 Implementacija sučelja 30
4.3.1 Predlošci 30
4.3.2 Stilizacija i dizajn sučelja 31

5 Zaključak **35**

Bibliografija **37**

Sažetak **40**

Popis slika

2.1 *Natjecatelji AO i AKA na početnoj liniji* 4

2.2 *Razlika između kata i kumite discipline* 5

2.3 *Prikaz jedne kategorije županijske lige* 6

4.1 *Prikaz modela CustomUser i njegove relacije* 15

4.2 *Prikaz modela Competition i njegove relacije* 16

4.3 *Prikaz modela Competitor i njegove relacije* 17

4.4 *Prikaz modela Category i njegove relacije* 18

4.5 *Prikaz modela CompetitionCategory i njegove relacije* 18

4.6 *Prikaz modela Application i njegove relacije* 19

4.7 *Prikaz modela Match i njegove relacije* 20

4.8 *Stvaranje nove baze podataka* 21

4.9 *Prikaz opcije direktnog brisanja modela* 23

4.10 *Primjer aktivacijske poruke* 25

4.11 *Prikaz dizajna CompWiz aplikacije* 34

Poglavlje 1

Uvod

Karate je borilački sport koji kombinira fizičku snagu, tehniku i duhovnu koncentraciju. Natjecanja u karateu zahtijevaju preciznost, brzinu i kontrolu, te su često podijeljena u različite kategorije prema dobi, spolu, disciplinama i težinskim kategorijama. U svijetu karatea, vođenje natjecanja izazovno je zbog brojnih sudionika i detaljnih pravila.[1]

Upravljanje karate natjecanjima zahtijeva precizno praćenje natjecatelja, njihovih prijava, formiranje rasporeda mečeva i pravilno bodovanje. Također, potrebno je voditi evidenciju o rezultatima, sankcijama i statistici natjecanja kako bi se osigurala pravedna i transparentna konkurencija.

Trenutna rješenja za vođenje karate natjecanja suočavaju se s nekoliko izazova, uključujući nedostatak ključnih značajki za automatizaciju procesa upisivanja rezultata mečeva i premalo statistike te činjenicu da većina postojećih rješenja ne pruža adekvatno rješenje za određene vrste natjecanja. Istovremeno, postojeća rješenja često su preskupa za široku primjenu u različitim natjecanjima.

U tu svrhu osmišljena je aplikacija naziva "CompWiz" internetska aplikacija (eng. *"web application"*) koja korisnicima omogućava da brzo i jednostavno stvaraju, upravljaju i prate natjecanja, bez obzira na njihovu lokaciju ili specifične zahtjeve, tj omogućava im da brzo i učinkovito upravljaju svim dijelovima natjecanja.

Cilj ovog rada je stvoriti aplikaciju koja će omogućiti organizatorima karate natjecanja efikasno vođenje natjecanja, smanjenje potencijalnih pogrešaka u unosu po-

dataka te poboljšati iskustvo sudionika i gledatelja na natjecanjima. Rad će prikazati primjenu tehnologije i alate koji su potrebni za ostvarivanje tog cilja.

Poglavlje 2

Pregled područja

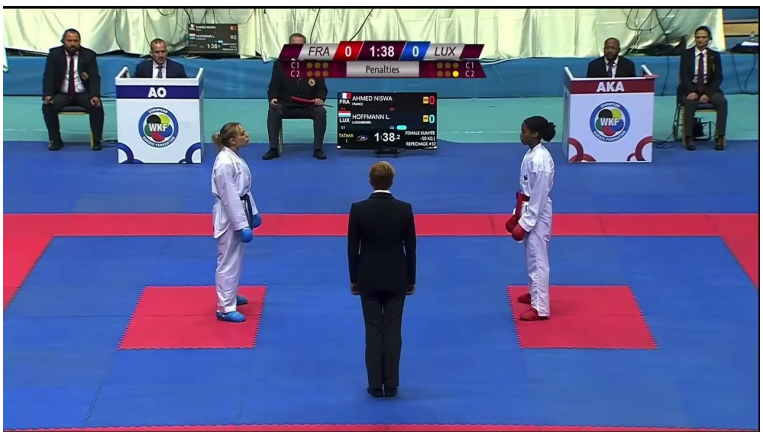
U ovom poglavlju istražit ćemo ključne koncepte karatea te postojeća rješenja i alate koji se koriste za vođenje i praćenje rezultata karate natjecanja. Karate, kao borilački sport s dugom tradicijom, ima različite načine vođenja natjecanja, ali slične izazove koji se pojavljuju unutar tog okvira. Učinkovito vođenje karate natjecanja zahtijeva preciznost, brzinu i točnost kako bi se osigurala pravedna i transparentna konkurencija.

2.1 Karate

Karate je borilački sport koji kombinira fizičku snagu, tehniku i duhovnu koncentraciju. To je sport koji se razvio u Japanu, a s vremenom je postao popularan diljem svijeta.[2] Karate natjecanja obično uključuju pojedinačne, ali i ekipne kategorije[3], a natjecatelji se boduju na temelju preciznosti tehnika, brzine i kontrole.[1]

Za bolje razumijevanje problematike vođenja karate natjecanja, važno je imati uvid u različite dijelove ovog sporta, uključujući kategorije natjecanja, bodovanje, registraciju natjecatelja i mnoge druge faktore koji čine dio vođenja takvih događanja.

Kako bi se razlikovali natjecatelji koji nose bijeli kimono, svakom je dodijeljena jedna od dvije boje; crvena i plava[1], prikazano na slici 2.1. Natjecatelj kojem je dodijeljena crvena boja naziva se *"AKA"*, a natjecatelj kojem je dodijeljena plava boja naziva se *"AO"*.



Slika 2.1 *Natjecatelji AO i AKA na početnoj liniji*

2.1.1 Vrste natjecanja

Postoje dvije vrste turnira u karateu; eliminacijski turniri i round robin turniri[1]. **Round Robin** (svi protiv svih), manje poznat tip natjecanja, gdje se svaki natjecatelj susreće s ostalim natjecateljima u istoj kategoriji. Svaki meč se boduje, a natjecatelji se rangiraju prema broju osvojenih bodova. Round Robin natjecanja često se koriste na županijskim, ali i na nekim od najvažnijih natjecanja na svjetskoj razini kao npr. Karate 1 - Premijer liga. Za razliku od round robin turnira, **Eliminacijski turniri** uključuju eliminaciju natjecatelja nakon svakog meča. Natjecatelji se uparuju u mečeve, a gubitnici ispadaju iz natjecanja, dok pobjednici napreduju dalje prema finalu.

Na višim razinama natjecanja, postoje kategorije koje spajaju određene dijelove Round Robin i eliminacijskih turnira.

2.1.2 Natjecateljske discipline

Osim različitih vrsta natjecanja, karate također uključuje različite discipline natjecanja (slika 2.2).[1] To uključuje **kumite**, disciplinu u kojoj se natjecatelji bore jedan na jedan i boduju (ili kažnjavaju) se za precizne udarce i tehniku tijekom borbe. Druga disciplina je **kata**, koja uključuje izvođenje uzoraka borbenih tehnika, tj. iz-

vođenje simulirane borbe protiv imaginarnog protivnika uz nastojanje da se očuva savršena forma. Natjecatelji su bodovani na temelju preciznosti i tehničke izvedbe.

Slika 2.2 *Razlika između kata i kumite discipline*

Postojeći pristupi vođenju natjecanja često se oslanjaju na ručni unos rezultata, tablice Excela ili specijalizirane softverske alate. Iako su ovi pristupi funkcionalni, često se suočavaju s određenim izazovima. Jedan od tih izazova je sporost takvih metoda, posebno kada se radi o većem broju natjecatelja i mečeva. Osim toga, ograničene su u značajkama za automatizaciju procesa, što može zahtijevati veliku količinu ručnog rada. Povećana mogućnost pogrešaka prilikom unosa podataka također je čest problem u ovim pristupima. Sve ovo može rezultirati manje efikasnim vođenjem natjecanja i potencijalno nezadovoljstvom sudionika i gledatelja.

Neki od postojećih softverskih alata i rješenja su preskupi za široku primjenu u različitim natjecanjima, što može ograničiti njihovu dostupnost organizatorima manjih ili lokalnih natjecanja.[4]

U nastavku ovog poglavlja, analizirat ćemo neka od postojećih rješenja i identificirati njihove prednosti i nedostatke u kontekstu vođenja karate natjecanja. Ova analiza pomoći će nam bolje razumjeti trenutno stanje u vođenju karate natjecanja i identificirati ključne značajke i funkcionalnosti koje bi trebala imati CompWiz aplikacija kako bi se prevladali nedostaci postojećih rješenja.

2.2 Excel tablice

Korištenje Excel tablica za vođenje karate županijske lige u formatu "round robin" često je jedini praktičan pristup, budući da trenutno nema drugih specijaliziranih alata dostupnih. Iako postoji nekoliko komercijalnih rješenja, njihova visoka cijena čini ih neprihvatljivima za županijske lige, koje obično uključuju nekoliko kola natjecanja i imaju ograničen budžet.[4]

Voditi natjecanje pomoću Excel tablica znači da organizator mora upravljati jednim Excel dokumentom koji može sadržavati 60 ili više radnih knjiga, pri čemu svaka radna knjiga predstavlja određenu kategoriju natjecatelja (slika 3.1). Korištenje tako velikog Excel dokumenta, zahtijeva značajne računalne resurse. Ovisno o specifikacijama računala, takav dokument može značajno usporiti rad računala, posebno ako se koristi stariji hardver. Pritom je potrebno imati dovoljno radne memorije (eng. *"RAM"*) kako bi se osigurala glatka i brza obrada podataka.

Također, velika količina podataka i radnih knjiga, uz sporost pri baratanju s njima, otežava organizatoru praćenje i unos rezultata mečeva, što povećava rizik od pogrešaka i nepreciznosti u vođenju natjecanja.

KARATE SAVEZ PRIMORSKO GORANSKE ŽUPANIJE

ŽUPANIJSKA KARATE LIGA 2023

KATEGORIJA: 2009.G. BORBE A

R.B R.	IME I PREZIME	KLUB	1.KOLO		2.KOLO		3.KOLO		4.KOLO		IME ŠKOLE	MJESTO ŠKOLE	UKUPNO 1. KOLO	UKUPNO 2. KOLO	UKUPNO 3. KOLO	UKUPNO 4. KOLO	UKUPNO
1.	EMILY ŠEKULJA	KACHI	1	3	0	0	0	0	0	0			6	0	0	0	6
			1	1	0	0	0	0	0	0							
2.	LENA KRIZMAN	CHAMPION	3	1	0	0	0	0	0	0			6	0	0	0	6
			1	1	0	0	0	0	0	0							
3.	PETRA KNEŽEVIĆ	CHAMPION	1	1	0	0	0	0	0	0			6	0	0	0	6
			3	1	0	0	0	0	0	0							
4.	NIKA ŠIMUNDIĆ	KASTAV	3	3	0	0	0	0	0	0			11	0	0	0	11
			2	3	0	0	0	0	0	0							
5.	NOELI KARLOVIĆ	KASTAV	3	3	0	0	0	0	0	0			11	0	0	0	11
			2	3	0	0	0	0	0	0							

1.KOLO				2.KOLO				3.KOLO				4.KOLO			
EMILY ŠEKULJA	1	3	LENA KRIZMAN	EMILY ŠEKULJA			LENA KRIZMAN	EMILY ŠEKULJA			LENA KRIZMAN	EMILY ŠEKULJA			LENA KRIZMAN
PETRA KNEŽEVIĆ	1	3	NIKA ŠIMUNDIĆ	PETRA KNEŽEVIĆ			NIKA ŠIMUNDIĆ	PETRA KNEŽEVIĆ			NIKA ŠIMUNDIĆ	PETRA KNEŽEVIĆ			NIKA ŠIMUNDIĆ
EMILY ŠEKULJA	1	3	NOELI KARLOVIĆ	EMILY ŠEKULJA			NOELI KARLOVIĆ	EMILY ŠEKULJA			NOELI KARLOVIĆ	EMILY ŠEKULJA			NOELI KARLOVIĆ
LENA KRIZMAN	1	3	PETRA KNEŽEVIĆ	LENA KRIZMAN			PETRA KNEŽEVIĆ	LENA KRIZMAN			PETRA KNEŽEVIĆ	LENA KRIZMAN			PETRA KNEŽEVIĆ
NIKA ŠIMUNDIĆ	2	2	NOELI KARLOVIĆ	NIKA ŠIMUNDIĆ			NOELI KARLOVIĆ	NIKA ŠIMUNDIĆ			NOELI KARLOVIĆ	NIKA ŠIMUNDIĆ			NOELI KARLOVIĆ
EMILY ŠEKULJA	3	1	PETRA KNEŽEVIĆ	EMILY ŠEKULJA			PETRA KNEŽEVIĆ	EMILY ŠEKULJA			PETRA KNEŽEVIĆ	EMILY ŠEKULJA			PETRA KNEŽEVIĆ
LENA KRIZMAN	1	3	NIKA ŠIMUNDIĆ	LENA KRIZMAN			NIKA ŠIMUNDIĆ	LENA KRIZMAN			NIKA ŠIMUNDIĆ	LENA KRIZMAN			NIKA ŠIMUNDIĆ
PETRA KNEŽEVIĆ	1	3	NOELI KARLOVIĆ	PETRA KNEŽEVIĆ			NOELI KARLOVIĆ	PETRA KNEŽEVIĆ			NOELI KARLOVIĆ	PETRA KNEŽEVIĆ			NOELI KARLOVIĆ
EMILY ŠEKULJA	1	3	NIKA ŠIMUNDIĆ	EMILY ŠEKULJA			NIKA ŠIMUNDIĆ	EMILY ŠEKULJA			NIKA ŠIMUNDIĆ	EMILY ŠEKULJA			NIKA ŠIMUNDIĆ
LENA KRIZMAN	1	3	NOELI KARLOVIĆ	LENA KRIZMAN			NOELI KARLOVIĆ	LENA KRIZMAN			NOELI KARLOVIĆ	LENA KRIZMAN			NOELI KARLOVIĆ

Slika 2.3 *Prikaz jedne kategorije županijske lige*

2.3 Portal ekarate.eu

Osim Excel tablica, za vođenje ostalih natjecanja lokalnog do manjeg međunarodnog opsega, dostupan je portal ekarate.eu. Prvi put korišten 2016. godine, ovaj portal, napravljen u PHP-u, često se koristi za vođenje natjecanja u Hrvatskoj.[5] Za razliku od Excel tablica, ovaj portal pruža praktičniji pristup vođenju natjecanja. Voditelji natjecanja pobjednike mečeva klikom premještaju dalje na sljedeću fazu natjecanja. Ovakav pristup smanjuje mogućnost pogrešaka i olakšava organizaciju natjecanja. Također, portal ekarate.eu omogućuje vođenje statistika koje nisu moguće na Excel tablicama.[6]

Međutim, i dalje postoje određene poteškoće u stvaranju i vođenju natjecanja. Naime, voditelji natjecanja upisuju pobjednike na temelju ispunjene liste kategorije koji su popunjavani na samom borilištu od strane drugih osoba. Ova praksa može dovesti do ljudskih pogrešaka i nepreciznosti u unosu rezultata. Također, sam portal nije naročito jednostavan za korištenje.

2.4 Sportdata Event Technology

Osim Excel tablica i portala ekarate.eu, postoji i Sportdata Event Technology, aplikacija tvrtke osnovane u Austriji i Švicarskoj 2000 godine.[7] Aplikacija SET napravljena je u Javi te podržava mnogo sportove, od kojih ćemo se bazirati samo na karate. Tokom godina, SET je postao standard za visoko rangirana natjecanja u karateu te je službeni program za vođenje svih WKF (eng. *"World Karate Federation"*) natjecanja.[8] SET je aplikacija koja rješava sve prethodno navedene probleme u vođenju karate natjecanja, točnije pruža softverska rješenja za sve procese upravljanja događanjima, uključujući online prijave, upravljanje prijavama, kotizacije, izvlačenje natjecatelja, unos rezultata, praćenje uživo te prikaz rezultata na semaforima i mnoge druge funkcionalnosti.[9]

Važno je napomenuti da visoka razina funkcionalnosti i automatske obrade podataka, koju nudi SET, može rezultirati drugim problemima, posebno visokim troškovima organizacije i vođenja natjecanja. SET često dolazi s vlastitim timom ljudi

koji trebaju biti dodatno plaćeni. Osim toga, organizacija samog natjecanja generira veće troškove, koji se dijele u dva dijela. Prvi dio najčešće fiksan, dok se drugi dio izračunava na temelju broja natjecatelja i odgovarajućeg iznosa.

Poglavlje 3

Korištene tehnologije

U ovom poglavlju bit će opisani alati korišteni za razvoj CompWiz internetske aplikacije. Ovi alati su odabrani kako bi se omogućila učinkovita izrada i upravljanje aplikacijom, te kako bi se osigurala sigurnost i skalabilnost u produkcijskom okruženju.

3.1 Django

Django je popularno internetsko razvojno okruženje [10] (eng. *"web framework"*) za Python koje se koristi za izradu CompWiz internetske aplikacije. Ovo razvojno okruženje pruža razne alate i biblioteke koji značajno olakšavaju razvoj internetskih aplikacija. Django koristi sustav za mapiranje objekata (ORM, eng. *"Object-relation mapping"*) za modeliranje baze podataka pomoću Python objekata, što omogućava programerima da lakše upravljaju podacima. Također, Django uključuje ugrađeno administratorsko sučelje koje korisnicima bez znanja SQL-a pojednostavljuje upravljanje aplikacijom.

Django promovira dobre prakse razvoja aplikacija[11], kao što je razdvajanje aplikacije na manje komponente (aplikacije), što olakšava održavanje i skalabilnost projekta. Osim toga, Django ima aktivnu zajednicu programera i obilje paketa koji se mogu koristiti kako bi se ubrzao razvoj i dodali različiti funkcionalni moduli.

Budući da aplikacija upravlja osjetljivim podacima o natjecateljima i rezultatima,

sigurnost je ključna komponenta i jedan od glavnih razloga za odabir Django-a za izradu CompWiz aplikacije. Django nudi ugrađene sigurnosne mehanizme koji pomažu zaštititi internetsku aplikaciju od različitih sigurnosnih prijetnji.

3.2 PostgreSQL

PostgreSQL je relacijski sustav otvorenog koda za upravljanje bazama podataka (RDBMS, eng. *"relation database management system"*) koji se koristi kao glavna baza podataka za CompWiz internetsku aplikaciju.

Pouzdanost i integritet podataka su od velikog značaja za aplikaciju CompWiz, gdje se bilježe i prate podaci o natjecateljima, rezultatima mečeva i drugim važnim informacijama. PostgreSQL nudi mnoge prednosti[12], uključujući ekonomičnost, pouzdanost, skalabilnost te podršku za ACID transakcije, akronim za atomičnost, konzistentnost, izolaciju i trajnost (eng. *"Atomicity, Consistency, Isolation, Durability"*). ACID transakcije su transakcije koje sadrže četiri osnovna svojstva transakcija u bazi podataka. Atomičnost znači da svaka transakcija mora biti izvršena u cijelosti ili uopće ne. Konzistentnost podrazumijeva da transakcije moraju održavati konzistentnost podataka. Izolacija znači da se transakcije izvršavaju neovisno jedna o drugoj. Trajnost osigurava da potvrđene transakcije ostanu trajno sačuvane u bazi podataka.[13] Takve transakcije omogućavaju integritet i dosljednost podataka čak i kod hardverskih ili softverskih problema.

Također nudi napredne funkcionalnosti poput proširenih tipova podataka, pohranjenih postupaka i punotekstualnog pretraživanja. PostgreSQL je podržan od strane velike zajednice korisnika i razvojnih programera, što osigurava njegovu održivost i kontinuirano unapređenje. Kao ključni element CompWiz aplikacije, PostgreSQL omogućava sigurno pohranjivanje i upravljanje podacima o natjecanjima, natjecateljima i drugim entitetima koji su bitni za efikasno vođenje karate natjecanja.

3.3 Tailwind CSS

Tailwind CSS je moderni stilski okvir (eng. *"Cascading Style Sheets framework"*) otvorenog koda koji se koristi za oblikovanje i stilizaciju korisničkog sučelja[14] aplikacije CompWiz. Ovaj okvir se ističe svojom visokom fleksibilnošću i prilagodljivošću u dizajniranju i stiliziranju korisničkog sučelja, što je ključno za stvaranje intuitivnog i privlačnog korisničkog iskustva.

Tailwind CSS primjenjuje atomski pristup CSS-u, što znači da su sva stilska pravila sastavljena od manjih komada, ili razreda (eng *"class"*). Ovi razredi se primjenjuju direktno na HTML elemente kako bi se postigao određeni izgled ili ponašanje. Ovaj koncept omogućava programerima preciznu kontrolu nad stilizacijom, čime se izbjegava nepotrebno preklapanje stilova i olakšava održavanje koda.

Tailwind CSS također dolazi s velikim brojem gotovih stilskih klasa koje olakšavaju oblikovanje elemenata kao što su tipografija, boje, margine, odmak, i mnogi drugi stilski atributi. Osim toga, Tailwind CSS omogućava prilagodbu stilova putem konfiguracijskih datoteka, što olakšava usklađivanje dizajna sa specifičnim zahtjevima aplikacije.

Osim Tailwind CSS-a, korišten je i Flowbite[15] dodatak za Tailwind CSS koji je dodatno olakšao izgradnju korisničkog sučelja. Flowbite donosi gotove komponente i stilove koje se lako integriraju u internetsku aplikaciju, čime se ubrzava proces razvoja i osigurava dosljedan dizajn svih komponenti. Bez potrebe za ručnim definiranjem svake pojedinačne komponente, pri razvoju CompWiz aplikacije uštedilo se na vremenu i trudu potrebnom za oblikovanje korisničkog sučelja.

Kombinacija Tailwind CSS-a i Flowbite-a osigurala je da CompWiz aplikacija ima konzistentan i atraktivan dizajn, uz minimalne napore u razvoju koje će korisnici cijeniti i koristiti s lakoćom.

3.4 Docker

Docker je platforma za kontejnerizaciju koja je ključna komponenta za postizanje skalabilnosti, sigurnosti i pouzdanosti u produkcijskom okruženju[16] za CompWiz

aplikaciju. Kontejnerizacija je tehnologija koja omogućava pakiranje aplikacije i njenih potrebnih resursa u izolirane kontejnere, što olakšava njihovo implementiranje i upravljanje na različitim računalima i serverima, tj. omogućava nam jednostavno premještanje aplikacije iz razvojnog u produkcijsko okruženje.

Docker kontejneri osiguravaju izolaciju resursa, sprječavajući bilo kakve smetnje od drugih aplikacija na istom serveru. To znači veću stabilnost i pouzdanost za našu aplikaciju. Također, Docker omogućava jednostavno skaliranje aplikacije prema potrebama bez kompliciranih postupaka. Upravljanje aplikacijom postaje jednostavnije zahvaljujući alatu Docker Compose, koji omogućava jednostavno upravljanje više kontejnerima putem jedne konfiguracijske datoteke. Nadalje, Docker pojednostavljuje izradu sigurnosnih kopija podataka aplikacije, čime se osigurava zaštita važnih informacija.

Docker tehnologija čini CompWiz aplikaciju skalabilnom, pouzdanom i jednostavnom za upravljanje u produkcijskom okruženju. To jamči visoku dostupnost i sigurnost za naše korisnike.

Poglavlje 4

Projektna aplikacija

Aplikacija je organizirana u dvije glavne pod-aplikacije, a to su aplikacija za autentikaciju i aplikacija za poslovnu logiku. Aplikacija za autentikaciju pruža sustav autentikacije korisnika koji ima funkcionalnost prijave i upravljanja korisnicima (klubovima, članovima klubova, savezima, osoblju). S druge strane, aplikacija competition wizard nudi nam sve funkcionalnosti vezane za sama natjecanja; stvaranje natjecanja, prijave na natjecanja, prijave natjecatelja, stvaranje kategorija, upravljanje natjecanjima (vođenje samih mečeva), pregled rezultata i slično.

U ovom poglavlju ćemo detaljno razmotriti funkcionalnosti aplikacije završnog rada Compwiz.

4.1 Baza podataka

Početak rada na aplikaciji sastojao se u skiciranju baze podataka na ploču, nakon čega su te skice prenesene u Django modele. Kako su dodavani novi modeli, baza podataka se prilagođavala kako bi bila prikladna za nove modele. U određenom trenutku razvijanja aplikacije došlo je do potrebe korištenja značajke (eng. *"attribute"*) tipa niz (eng. *"array"*). Naime, većina kategorija se sastoji od nekoliko godišta spojenih u jednu kategoriju[3], zbog čega je izbor značajke tipa niz bilo intuitivno i jednostavno rješenje problema (kodni isječak 4.1).

Obzirom da zadana Django baza podataka (SQLite) ne podržava značajke tipa

niz. Kako bi se omogućila podrška za nizove, odlučeno je zamijeniti SQLite bazu podataka s PostgreSQL bazom podataka[17], koja pruža podršku za ovakav tip podataka. No čak i da SQLite ima podršku za značajke tipa niz, prelazak na PostgreSQL bio je neizbježan obzirom da SQLite nije idealan za produkcijsko okruženje zbog svojih ograničenja u performansama, veličini baze podataka, podršci za napredne funkcionalnosti i nedostatku podrške za više korisnika.[18]

Kodni isječak 4.1 primjer JSON datoteke za kategoriju

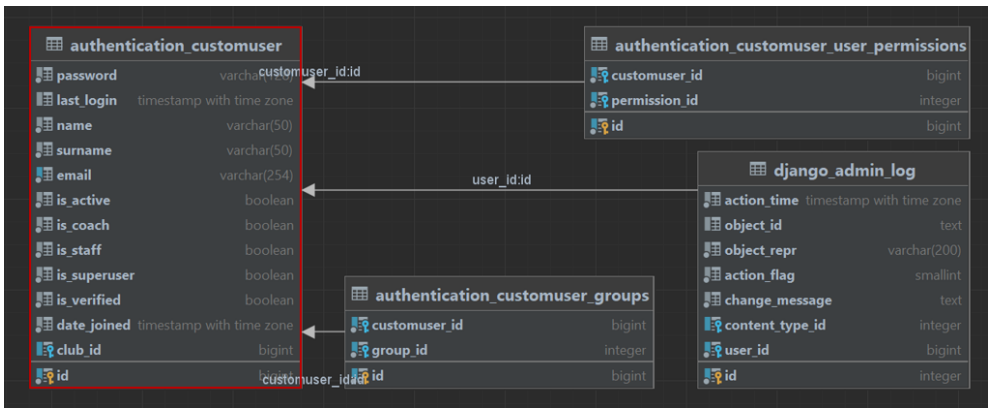
```
2    "kata": [{
3      "age": [
4        7, 8, 9
5      ],
```

4.1.1 Autentikacija

Django dolazi s ugrađenim sustavom za autentikaciju koji omogućuje upravljanje korisničkim računima i sigurnosnim mehanizmima u internetskim aplikacijama.[19] Ovaj ugrađeni sustav za autentikaciju omogućuje registraciju korisnika, prijavu, odjavu, upravljanje lozinkama i provjeru prava pristupa resursima. Koristi se za upravljanje sesijama i sigurno čuvanje lozinki korisnika. Ovaj ugrađeni sustav za autentikaciju štedi vrijeme i trud programerima, omogućujući im brzu implementaciju sigurnosti korisničkih računa u Django aplikacijama za završni rad.

Zadani Django korisnički model uključuje različite značajke. To uključuje username za jedinstveni identifikator korisnika potreban za prijavu te password za spremanje kriptirane lozinke. Nadalje, model sadrži opcionalne značajke kao što su *"email"*, *"first_name"* i *"last_name"* za dodatne informacije o korisniku. Također, ima polja *"is_active"* za označavanje aktivnosti korisnika, *"is_staff"* za označavanje administrativnih ovlasti i *"is_superuser"* za označavanje superkorisničkih privilegija. Model također sadrži značajke za praćenje datuma pridruživanja (*"date_joined"*) te za grupiranje korisnika (*"groups"*) i dodjeljivanje dozvola (*"user_permissions"*). Django također omogućuje prilagodbu korisničkog modela kako bi odgovarao specifičnim potrebama projekta, uključujući dodavanje dodatnih polja u korisničke profile ili implementaciju različitih metoda autentikacije.[19]

Aplikacija ***authentication*** je sačinjena od posebno prilagođenog korisničkog modela, nazvanog ***"CustomUser"*** (prikazan na slici 4.2) kako bi se omogućilo korištenje e-mail adrese kao jedinstvenog identifikatora korisnika umjesto korisničkog imena (eng. *username*). Osim zamjene jedinstvenog identifikatora, dodane su još neke značajke: *"is_coach"* za označavanje korisnika kao trenera, *"is_verified"* za provjeru je li korisnik verificiran od strane kluba, saveza ili administratora te opcionalna značajka "*club_id*" koji povezuje korisnika sa klubom (ako je korisnik član nekog kluba).



Slika 4.1 *Prikaz modela CustomUser i njegove relacije*

Za ostvarivanje ove promjene, definirali smo novu komponentu nazvanu ***"CustomUserManager"***[20] (prikazano na kodnom isječku 4.2). UserManager je posebna klasa koja uključuje pomoćne metode poput ***"create_user"*** i ***"create_superuser"***.

Kodni isječak 4.2 Primjer pomoćne metode

```
class CustomUserManager(BaseUserManager):
    def create_user(self, email, password, **extra_fields):
        if not email:
            raise ValueError(_('The Email must be set'))
        email = self.normalize_email(email)
        user = self.model(email=email, **extra_fields)
        user.set_password(password)
        user.save()
```

```
13            return user
```

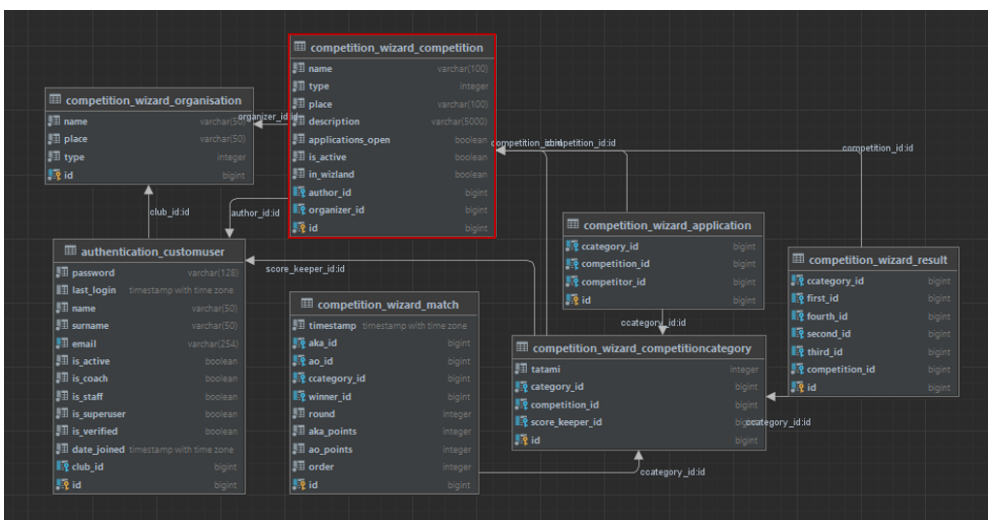
Za kraj, potrebno je u glavnoj konfiguracijskoj datoteci ***"settings.py"*** dodati liniju prikazanu na kodnom isječku 4.3, kojom se definira model koji će se koristiti za autorizaciju korisnikal.

Kodni isječak 4.3 Primjer pomoćne metode

```
5 AUTH_USER_MODEL = 'authentication.CustomUser'
```

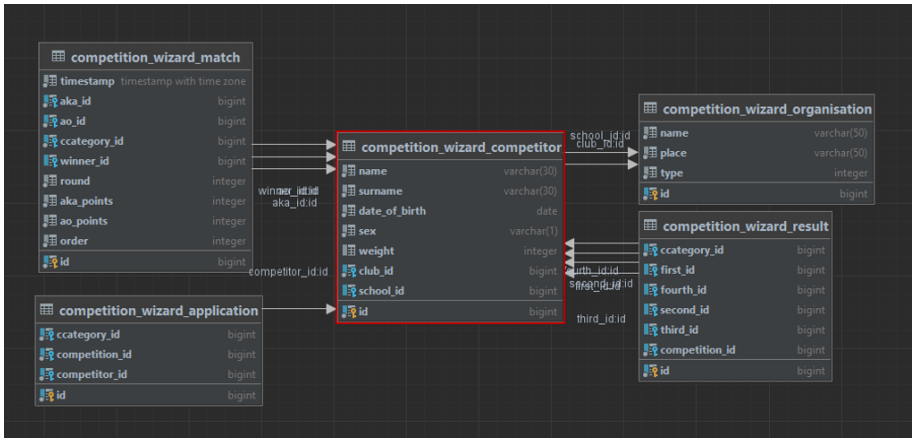
4.1.2 Poslovna logika

Aplikacija poslovne logike dalje je podijeljena u nekoliko modela koji predstavljaju različite aspekte karate natjecanja. Glavni model, model ***"Competition"*** (slika 4.2) određuje karate natjecanje i sadrži informacije kao što su ime natjecanja, vrsta natjecanja, datumi početka i kraja te lokacija. Osim toga sadrži tri logičke varijable koje označavaju je li natjecanje aktivno te jesu li prijave na natjecanje otvorene. Također sadrži informacije o autoru tj. korisniku koji je napravio natjecanje, te klub koji je organizator natjecanja.



Slika 4.2 *Prikaz modela Competition i njegove relacije*

Model ***"Competitor"*** (slika 4.3) određuje pojedinačne karate natjecatelje i sadrži informacije kao što su ime, prezime, datum rođenja, spol i klub kojem pripadaju. Opcionalne značajke su škola natjecatelja koja se koristi za statistiku županijskih liga kako bi se napravio poredak škola, te težina natjecatelja kako bi se u budućnosti mogao implementirati autoamtski odabir kategorije na osnovu spola i težine.

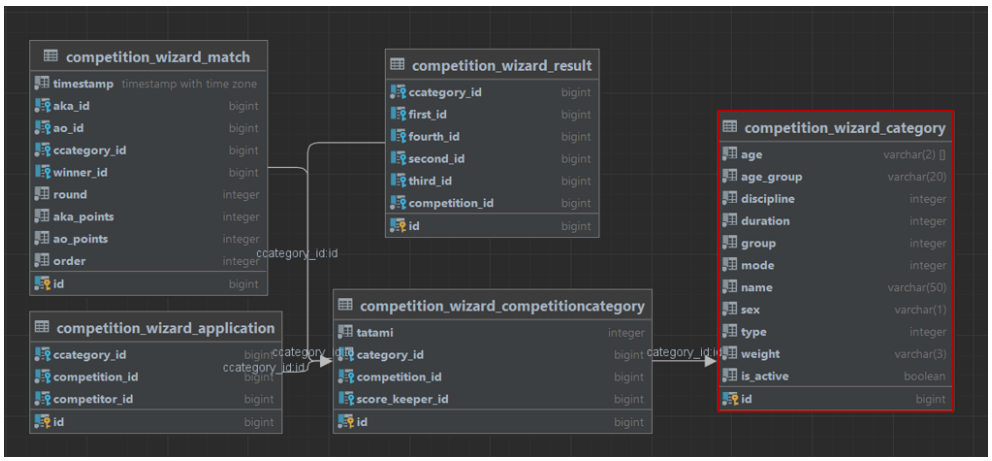


Slika 4.3 *Prikaz modela Competitor i njegove relacije*

Model ***"Category"*** (slika 4.4) služi za pohranu informacija o svim mogućim kategorijama koje su korištene u aplikaciji CompWiz.

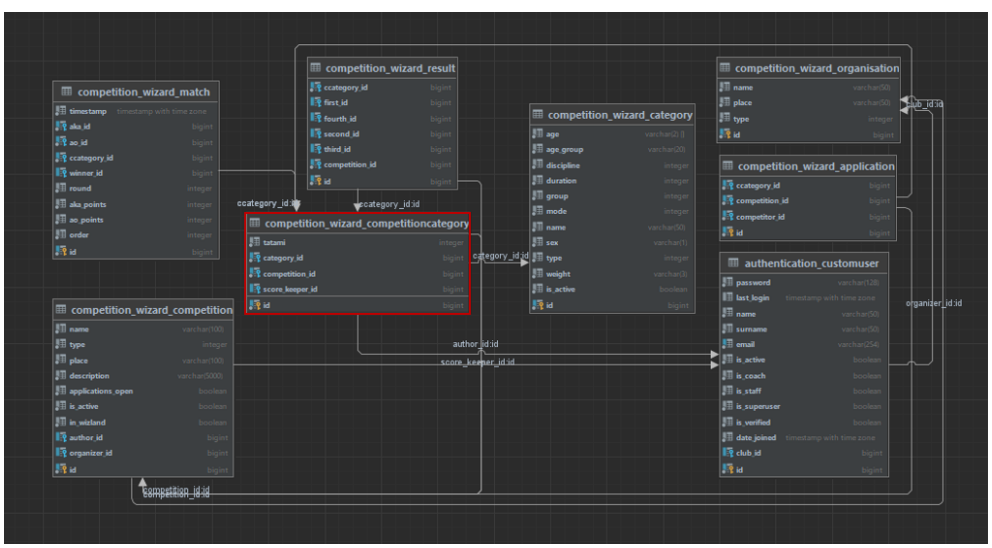
Ovi podaci uključuju informacije o starosnoj dobi kategorije, disciplini, trajanju kategorije, grupi (kategorije u karateu obično se dijele na grupe A, B, WKF, veterane i parakarate[3]; grupa A i grupa B predstavljaju dvije kvalitativne grupe natjecatelja mlađeg uzrasta, dok su WKF kategorije obično starije i koriste se za službene WKF natjecanja, "veterani" najčešće označavaju natjecatelje starije od 35 godine, a grupa "parakarate" obuhvaća sve natjecatelje s bilo kakvim invaliditetom), imenu, spolu, težini, modu (eliminacijski ili round robin) i tipu kategorije (individualni ili timski). Također sadrži jednu logičku varijablu kojom se označuje je li kategorija trenutno aktivna.

Model ***"CompetitionCategory"*** (slika 4.5) igra ključnu ulogu kao veza između modela "Competition" i "Category." Ovaj model sadrži strane ključeve koji omogućuju povezivanje s tri različita modela: natjecanje, kategorija i zapisničar (eng.



Slika 4.4 *Prikaz modela Category i njegove relacije*

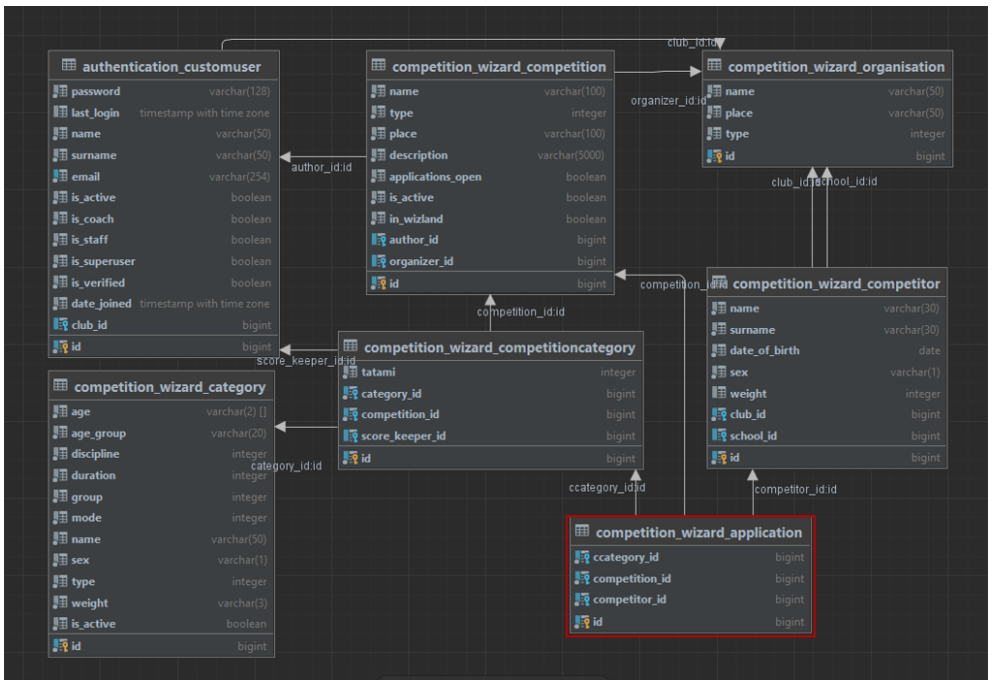
"score-keeper") (Customuser model). Osim toga, ima još jedana značajka koja označuje tatami (borilište) na kojem će se održavati mečevi.



Slika 4.5 *Prikaz modela CompetitionCategory i njegove relacije*

Model ***"Application"*** (slika 4.6) služi kao poveznica između modela "CompetitionCategory" i "Competitor" te sadrži informacije o natjecateljima koji su se prijavili za sudjelovanje u određenoj kategoriji na nekom natjecanju. Sastoji se od tri

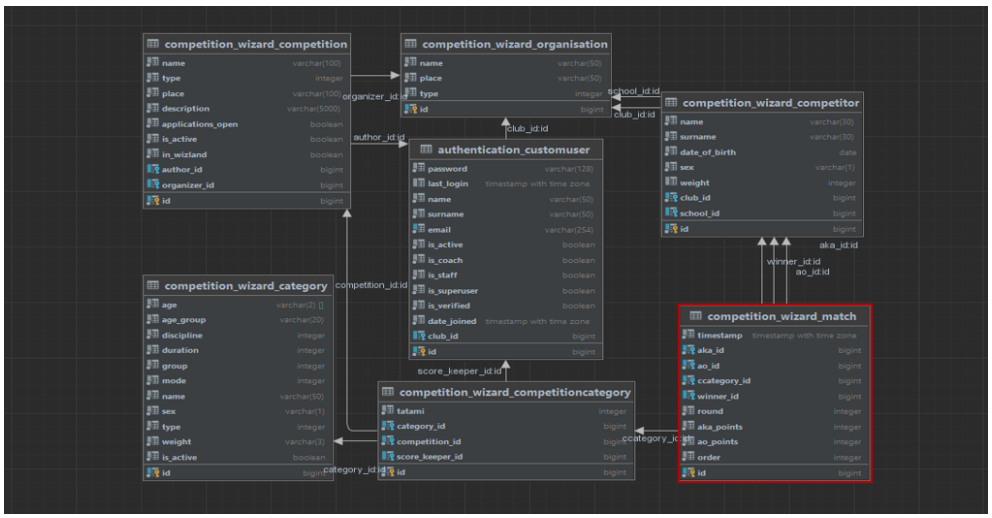
strana ključa; natjecateljska kategorija, natjecanje i natjecatelj. Iako preko modela *"CompetitionCategory"* možemo doći do podatka o kojem je natjecanju riječ, prilikom razvijanja aplikacija ustanovilo se da je bolje, brže i jednostavnije pristupiti određenim podacima ako postoji direktna veza sa modelom natjecanja.



Slika 4.6 *Prikaz modela Application i njegove relacije*

Model ***"Match"*** (slika 4.7) predstavlja pojedinačni meč između dva natjecatelja i sadrži informacije o natjecateljima koji su sudjelovali, njihovim bodovima i pobjedniku meča, vremenu kada je meč odigran te kategoriji u kojoj je odigran. Opcionalne informacije su runda koja je bitna za round robin natjecanja sa najčešće 3 runde i redni broj meča koji se također koristi kod round robin natjecanja kako bi poredak borbi uvijek bio jednak.

Osim tih modela, ostaju nam modeli ***"Organisation"*** koji predstavlja određene organizacije (klub, savez, škola,...) i ***"Result"*** koji nam služi za spremanje rezultata određene kategorije tako da ne trošimo vrijeme na računanje rezultata svaki put kada ih želimo vidjeti, a sastoji se od stranih ključeva za natjecanje, kategoriju i svako od mjesta (prvo, drugo i dva treća).



Slika 4.7 *Prikaz modela Match i njegove relacije*

Posljednji model ne koristi se kao klasični model, već kao model koji koristimo za prenošenje datoteke na Compwiz (kodni isječak 4.4).[21]

Kodni isječak 4.4 model za prijenos datoteke

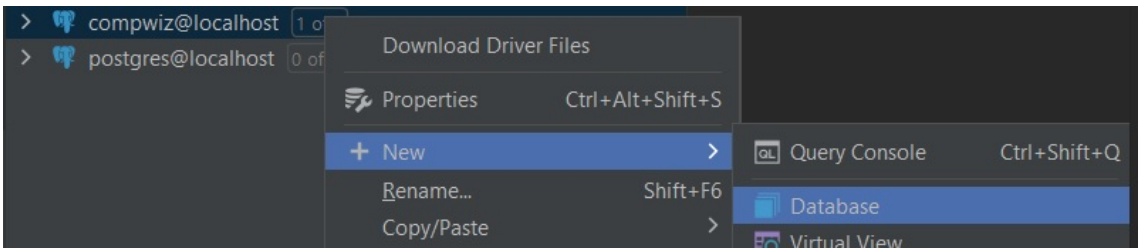
```
from django.db import models
class Sample(models.Model):
    attachment = models.FileField()
```

4.2 Implementacija poslužitelja

4.2.1 Postavljanje PostgreSQL baze podataka

Prvi važan korak u uspostavljanju PostgreSQL baze podataka uključuje stvaranje same baze. Iako postoji više načina za to, zbog jednostavnosti koristio se program DataGrip. Korištenjem ovog alata, nova baza podataka može se stvoriti u samo nekoliko klikova kao što je prikazano na slici 4.8.

Nakon što smo uspješno stvorili bazu podataka, sljedeći korak je konfiguracija PostgreSQL baze. Ovaj proces je relativno jednostavan i uključuje zamjenu zadane



Slika 4.8 *Stvaranje nove baze podataka*

konfiguracije s konfiguracijom prikazanom na kodnom isječku 4.5. Osim prethodno definiranih *"ENGINE"* i *"NAME"* značajki, moramo dodati nekoliko značajki koje nisu bile potrebne za SQLite[22]; *"USER"*, *"PASSWORD"*, *"HOST"* i *"PORT"*. Opcionalna značajka *"CONN_MAX_AGE"* dodala se je kako bi se izbjegavalo zadržavanje starih veza i osiguralo održavanje pouzdane veze s bazom podataka

Kodni isječak 4.5 konfiguracija PostgreSQL baze podataka

```
88 DATABASES = {
89     'default': {
90         'ENGINE': 'django.db.backends.postgresql',
91         'NAME': 'compwiz',
92         'USER': 'postgres',
93         'PASSWORD': 'admin',
94         'HOST': 'localhost',
95         'PORT': '5432',
96         'CONN_MAX_AGE': 600,
97     }
```

Kako bismo potvrdili ispravnu konfiguraciju baze podataka, izrađujemo migracije i izvršavamo migraciju baze, prikazano na kodnom isječku 4.6. Ako ne primimo poruku o grešci, to znači da smo uspješno zamijenili SQLite bazu podataka sa PostgreSQL bazom podataka.

Kodni isječak 4.6 stvaranje migracija i migriranje baze podataka

```
1 python manage.py makemigrations
2 python manage.py migrate
```

U produkcijskom okruženju, osjetljive podatke treba čuvati sigurno i zaštićeno. Jedan od načina je korištenje konfiguracijske datoteke ili varijabli okoline (eng. *"environment variables"*) kako bi ih odvojili od izvornog koda aplikacije.[22] Na taj način osjetljivi podaci kao što su lozinke ili ključevi za pristup bazi podataka neće biti izloženi u samom kodu aplikacije i bit će sigurno pohranjeni na serveru. Konfiguracijsku datoteku se još može enkriptirati kako bi dodatno osigurali te osjetljive podatke. Važno je strogo paziti na sigurnost ovih podataka u produkcijskom okruženju kako bi se spriječio neovlašten pristup ili curenje informacija.

4.2.2 Autentikacija i autorizacija korisnika

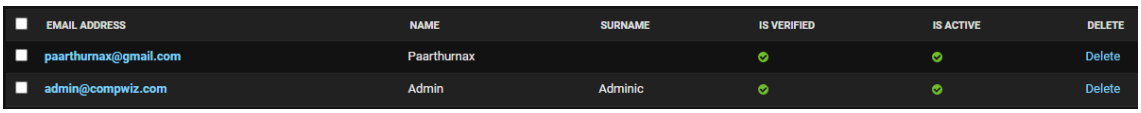
Za autentikaciju i autorizaciju korisnika koristi se sustav za autentikaciju korisnika koji je ugrađen u Django.[19]

Na samom početku potrebno je stvoriti novu Django aplikaciju naredbom *"python manage.py startapp authentication"*. Izvršavanjem te naredbe u početnom direktoriju stvara se novi direktorij *"authentication"*. Zatim je potrebno registrirati novo stvorenu aplikaciju na način da ju nadopišemo na popis instaliranih aplikacija u *"settings.py"* datoteci kao što je prikazano na kodnom isječku 4.7.

Kodni isječak 4.7 registracija aplikacije Authentication

```
34 INSTALLED_APPS = [
35     ...,
36     'authentication',
37 ]
```

Po završetku registracije nove aplikacije na redu je definiranje putanja (eng. *"urls"*) aplikacije koje omogućuju autentikaciju i autorizaciju korisnika (kodni isječak 4.8). Glavne putanje koje bi trebala sadržavati aplikacija za autentikaciju uključuju registraciju, prijavu i odjavu. Isto tako, potrebna je i staza za brisanje korisnika. Ta putanja omogućuje nam dodavanje opcije brisanja direktno iz modela *"CustomUser"* na Django admin stranici, što olakšava proces brisanja korisnika bez potrebe za ulaskom u svaki model pojedinačno (slika 4.9). Ta značajka je olakšala testiranje same *authentication* aplikacije.



Slika 4.9 *Prikaz opcije direktnog brisanja modela*

Kako bismo se zaštitili od neželjene pošte (eng. *"spam"*), implementirana je i potvrda putem e-pošte. Bez potvrde e-pošte, korisnicima nije dopušten pristup aplikaciji CompWiz, stoga smo dodali dodatnu putanju za aktivacija korisničkog računa.

Kodni isječak 4.8 definirane putanje za autentikaciju

```
urlpatterns = [
    path('activate/<uidb64>/<token>/', views.activate, name='activate'),
    path('customuser/<int:pk>/delete', views.delete_user, name='user-delete'),
    path('login', views.login_user, name='login'),
    path('logout', views.logout_user, name='logout'),
    path('register', views.register_user, name='register')]
```

Nakon definiranja putanja, slijedi definiranje pogleda (eng. *"views"*). Za svaku rutu definiramo svoj pogled, počevši s pogledom za registraciju korisnika. Kako bi spremanje korisnika u bazu podataka prilikom registracije bilo što jednostavnije, koriste se Django obrasci (primjer obrasca prikazan na kodnom isječku 4.9).[23]

Kodni isječak 4.9 primjer Django obrasca

```
class RegisterUserForm(UserCreationForm):
    email = forms.EmailField()
    name = forms.CharField(max_length=50)
    surname = forms.CharField(max_length=50)

    class Meta:
        model = CustomUser
        fields = ('name', 'surname', 'email', 'password1', '
```

```
    password2')
```

Obrasci su Python objekti koji omogućavaju prikupljanje podataka od korisnika na internetskim stranicama. Pomoću Django obrazaca, definiraju se polja za unos podataka i može se prilagoditi kako će se podaci prikupljati, validirati i spremati u bazu podataka. Ovo olakšava i ubrzava proces registracije korisnika te osigurava da se ispravni podaci pohranjuju u sustav. Kada korisnik popuni registracijski obrazac, obrazac se sprema, što znači da se novi korisnik pohranjuje u bazu podataka. Nakon što se korisnik pohrani u bazu, odmah se autenticira kako korisnik ne bi morao ručno izvršiti prijavu nakon registracije. Nakon autentikacije, stvara se aktivacijska poruka za novog korisnika, što je prikazano na kodnom isječku 4.10. Svaka poruka je jedinstvena jer koristi novo generirani token koji se generira za svakog korisnika. Dodaje se email pošiljatelja te se stvara email poruka pomoću ***"django.core.mail.EmailMessage()"***.[24]

Kodni isječak 4.10 definiranje aktivacijske poruke

```
92      subject = _("Activate your account.")
93      message = render_to_string('registration/email.html', {
94          'user': user,
95          'domain': get_current_site(request).domain,
96          'uid': urlsafe_base64_encode(force_bytes(user.pk)),
97          'token': default_token_generator.make_token(user),
98      })
99      from_email = 'noreply@compwiz.com'
100
101     email = EmailMessage(subject, message, from_email, [
    to_email])
102     email.content_subtype = 'html'
103     email.send()
```

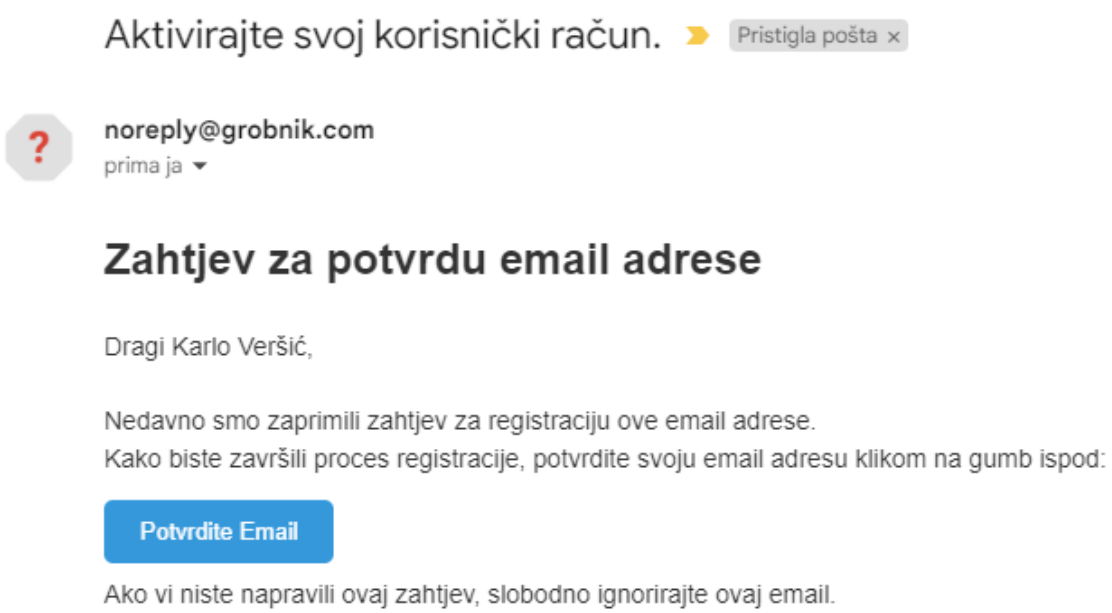
Prije slanja e-pošte potrebno je definirati kako, kuda i tko će poslati e-poštu, točnije potrebno je definirati varijable prikazane u kodnom isječku 4.11.

Kodni isječak 4.11 Varijable potebne za slanje e-pošte

```
168 EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
```

```
169 MAILER_EMAIL_BACKEND = EMAIL_BACKEND
170 EMAIL_HOST = 'rijeka.riteh.hr'
171 EMAIL_HOST_PASSWORD = '*******'
172 EMAIL_HOST_USER = 'kversic@riteh.hr'
173 EMAIL_PORT = 465
174 EMAIL_USE_SSL = True
175 DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
```

Nakon definiranja potrebnih varijabli i slanja e-pošte, korisnika se preusmjerava na početnu stranicu gdje ga se moli da potvrdi svoju email adresu kako bi završio proces registracije. Radi jednostavnosti potvrde email adrese, dodan je gumb koji predstavlja direktnu vezu na stranica pružatelja usluga e-pošte korisnika. Klikom na poveznicu u e-pošti (primjer e-pošte na slici 4.10), korisnik pristupa ***"activate"*** pogledu.



Slika 4.10 *Primjer aktivacijske poruke*

Taj pogled najprije koristi UID kako bi dohvatio određenog korisnika i potom provjerava je li dobiveni token valjan pomoću

"default_token_generator.check_token(user, token)". Ako je token valjan, korisnik postaje verificiran i preusmjerava se natrag na početnu stranicu. Iako je korisnički račun sada verificiran, još uvijek nije odobren od strane administratora. Dodatni korak odobrenja uveden je kako bi se smanjila mogućnost stvaranja neželjenih korisničkih računa. Iako se može činiti kao dodatni ručni rad, budući da novi treneri nisu česta pojava, ovo je najefikasniji način filtriranja nepoželjnih korisnika.

Preostali pogledi, poput ***"login"*** i ***"logout"*** implementirani su vrlo jednostavno, koristeći ugrađene Django funkcije poput *"authenticate()"* i *"logout()"*.[19] Slično tome, pogled za brisanje korisnika je implementiran u samo nekoliko linija koda, na način da se najprije dohvaća korisnički objekt pomoću *"get_object_or_404()"* funkcije i zatim briše taj objekt jednostavnom naredbom *"delete()"*.

4.2.3 Konfiguracija statičnih datoteka

Django omogućava jednostavno upravljanje statičnim datotekama u internetskim aplikacijama kao što su CSS, JavaScript, slike i ostali resursi koji se koriste za izgled i funkcionalnost internetskih stranice. Tu funkcionalnost postiže ugrađenim sustavom za rukovanje i posluživanje statičkih datoteka, a za njegovu konfiguraciju i upravljanje potrebno je prvo definirati određene postavke unutar glavne konfiguracijske datoteke ***"settings.py"***.[25]

Za početak definiramo direktorij koji će se koristiti za pohranjivanje svih statičnih datoteka kao što je prikazano na kodnom isječku 4.12.

Kodni isječak 4.12 definiranje direktorija statičnih datoteka

```
145 STATIC_ROOT = 'static/'
146 STATIC_URL = 'static/'
```

Iako su ***"STATIC_ROOT"*** i ***"STATIC_URL"*** osnovne postavke za pravilno upravljanje statičkim datotekama u Django aplikaciji, u Compwiz aplikaciji moramo dodatno definirati nekoliko konfiguracija. Razlog tome je što se koristi *"NodeJS"* kako bi se omogućilo korištenje Tailwind CSS-a.

Najprije je potrebno definirati apsolutni put do NodeJS-a[26] (prikazano na kodnom isječku 4.13), nakon čega je potrebno dodati jos jedan pretraživač statičnih dato-

teka (prikazano na kodnom isječku 4.14). Za kraj potrebno je jos dodati ***"node_ modules"*** u direktorije statičnih datoteka (prikazano na kodnom isječku 4.15).

Kodni isječak 4.13 definiranje puta do NodeJS-a

```
123 NPM_BIN_PATH = r"C:\Program Files\nodejs\npm.cmd"
```

Kodni isječak 4.14 dodavanje pretraživača statičnih datoteka

```
149 STATICFILES_FINDERS = [
150     ...,
151     'npm.finders.NpmFinder']
```

Kodni isječak 4.15 dodavanje node_modules u statične datoteke

```
156 STATICFILES_DIRS = (
157     ...,
158     ('node_modules', os.path.join(BASE_DIR, 'node_modules/')),)
```

4.2.4 Implementacija ostalih funkcionalnosti

Od preostalih funkcionalnosti, bitno je napomenuti samo neke jer su mnoge od njih implementirane na trivijalan način kao što je prikazano u kodnom isječku 4.16.

Kodni isječak 4.16 implementacija prikaza i stvaranja klubova

```
11 def all_view(request):
12     form = NewClubForm(request.POST or None)
13     if form.is_valid():
14         organisation, created = Organisation.objects.
    get_or_create(name=request.POST.get('name'),
15 
        place=request.POST.get('place'),
16 
        type=1)
17 
18         if created:
19             messages.success(request, 'Club created
    successfully.')
```

```
20          else:
21              messages.warning(request, 'Club not created.')
22
23      club_count = Organisation.objects.all().filter(type=1)
24      return render(request, 'competition_wizard/clubs/list.html'
    , {'type': 'club',
25
          'clubs': club_count,
```

Jedna od manje trivijalnih funkcionalnosti je prikaz svih borbi u određenoj kategoriji round robin natjecanja koja je prikazana u kodnom isječku 4.17.

Kodni isječak 4.17 implementacija prikaza svih mečeva

```
11 competitors_points = get_points(ccategory, competitors)
12 first_round_points, second_round_points, third_round_points =
    competitors_to_round_points(competitors_points)
```

Kako bismo ispravno ispunili tablicu bodova, potrebno je najprije dohvatiti bodove svih natjecatelja koji sudjeluju u određenoj kategoriji. Dohvaćanje bodova je jednostavno, ali i sporo obzirom na sve petlje kroz koje se mora proći (kodni isječak 4.18).

Kodni isječak 4.18 dohvaćanje bodova

```
437 def get_points(ccategory, competitors):
438     points = []
439     for competitor in competitors:
440         competitor_matches = Match.objects.all().filter(
    ccategory=ccategory, aka=competitor).order_by(
441             'order') | Match.objects.all().filter(ccategory=
    ccategory, ao=competitor).order_by('order')
442         competitor_points = []
443         for round_num in range(1, 4):
444             competitor_round_points = 0
445             for match in competitor_matches:
446                 if match.round == round_num:
447                     if match.aka == competitor:
```

```
448                         competitor_round_points += match.
    aka_points
449                     elif match.ao == competitor:
450                         competitor_round_points += match.
    ao_points
451             competitor_points.append(competitor_round_points)
452         points.append(competitor_points)
453     return points
```

Dobivene bodove moramo transponirati kako bi mogli odvojiti bodove na prvo, drugo i treće kolo. Funkcija koja transponira matricu prikazana je u kodnom isječku 4.19.

Kodni isječak 4.19 implementacija transponiranja matrice

```
420 def competitors_to_round_points(matrix):
421     num_rows = len(matrix)
422     num_cols = len(matrix[0])
423 
424     # Stvori novu transponiranu matricu s brojem stupaca i
    redaka zamijenjenim
425     transposed_matrix = [[0 for j in range(num_rows)] for i in
    range(num_cols)]
426 
427     # Petlja kroz svaki element u originalnoj matrici
428     for i in range(num_rows):
429         for j in range(num_cols):
430             # Postavi transponirani element na trenutni element
     u originalnoj matrici
431             transposed_matrix[j][i] = matrix[i][j]
432     return transposed_matrix
```

Još jedna funkcionalnost koju vrijedi spomenuti je dohvaćanje svih natjecatelja koji su pobijedili u vise od 80% mečeva (kodni isječak 4.20). Da bi se to postiglo, upit najprije koristi funkciju *"aggregate()"* i *"Sum"* za zbrajanje svih bodova objekata "Competitor". Zatim koristi ***"Q"*** funkciju[27] za prebrojavanje broja mečeva koje

je svaki natjecatelj dobio. Zatim se grupiraju po rundama i broji koliko je rundi natjecatelj osvojio. Rezultat se pohranjuje u varijablu *"matches_won"*. Nakon toga računa se ukupni broj odigranih mečeva (mečevi u kojima je natjecatelj dobio barem 1 bod). Naposljetku, izračunamo postotak pobjeda kao *"dobiveni mečevi"* / *"odigrani mečevi"*.

Kodni isječak 4.20 dio upita za pronalaženje najboljih natjecatelja

```
187         # Izracunaj ukupan broj odigranih meceva od strane
    natjecatelja s bodovima razlicitim od nule
188         total_matches_played = Match.objects.filter(
189             (models.Q(aka_id=competitor.id) | models.Q(ao_id=
    competitor.id)) &
190             (models.Q(aka_points__gt=0) | models.Q(
    ao_points__gt=0))
191         ).count()
```

4.3 Implementacija sučelja

4.3.1 Predlošci

U ovom dijelu razmotrit ćemo upotrebu Django predložaka (eng. *"templates"*) u CompWiz aplikaciji za poboljšanje dizajna i strukture internetskog sučelja. Predlošci su ključni dio Django okvira za izradu internetskih stranica jer omogućavaju organizaciju i ponovnu upotrebu HTML koda na različitim stranicama.[28]

Jedan od temeljnih koncepta u upotrebi Django predložaka u CompWiz aplikaciji je ***"base.html"***.[29] Ovaj bazni predložak sadrži zajedničke elemente i komponente koje se pojavljuju na svim stranicama internetske aplikacije. To uključuje navigacijsku traku, zaglavlje, podnožje i ostale dijelove koji su konzistentni na svim stranicama (primjer prikazan na kodnom isječku 4.21). Korištenjem baznog predloška, olakšava se održavanje dosljednog izgleda i ponašanja internetskog sučelja, što u konačnici rezultira uštedom vremena potrebnog za pisanje nove stranice. Primjer prazne datoteke koja proširuje *"base.html"* može se vidjeti u kodnom isječku 4.22.

Kodni isječak 4.21 Prikaz uključivanja navigacijske i bočne trake

```
{% include 'competition_wizard/navbar.html' %}
<div class="flex pt-16 z-0 overflow-hidden bg-gray-400 h-full">
    {% if user.is_authenticated %}
        {% include 'competition_wizard/sidebar.html' %}
    {% endif %}
```

Kodni isječak 4.22 Prikaz prazne datoteke

```
{% extends 'competition_wizard/base.html' %}
{% load static %}

{% block title %}
{% endblock %}

{% block heading %}
{% endblock %}

{% block content %}
    {% if user.is_authenticated %}
    {% endif %}
{% endblock %}

{% block scripts %}
```

4.3.2 Stilizacija i dizajn sučelja

Kako bismo olakšali proces stilizacije i dizajna korisničkog sučelja te izbjegli nepotrebno ponavljanje koda, koristili smo CSS okvir Flowbite. Ovaj okvir pruža gotove stilizirane komponente i jednostavne načine prilagodbe izgleda, što nam omogućava brži i konzistentniji dizajn sučelja.[15]

Kako bi se Flowbite mogao koristiti, najprije je potrebno instalirati Tailwind CSS u Django projekt pomoću naredbe ***"npm install -D tailwindcss"***.[30] Nakon instalacije, pokreće se naredba ***"npx tailwindcss init"*** kako bi se stvorila kon-

figuracijska datoteka ***"tailwind.config.js"***. Ova datoteka omogućava prilagodbu i konfiguraciju Tailwind CSS-a prema potrebama projekta. Nakon toga, u glavnu CSS datoteku kopira se kod prikazan u kodnom isječku 4.23. te se pokreće naredba za kompilaciju CSS-a: ***"npx tailwindcss -i ./static/src/input.css -o ./static/src/output.css –watch"***, čime Tailwind CSS postaje spreman za korištenje.

Kodni isječak 4.23 Konfiguracija "input.css" datoteke

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Sljedeće na redu je instalirati Flowbite jednostavnom naredbom ***"npm install flowbite"***, nakon čega je potrebno konfiguritati ***"tailwind.config.js"*** datoteku kao što je prikazano u kodnom isječku 4.24. Nakon uspješne konfiguracije u HTML datoteku se kopira naredba prikazana u kodnom isječku 4.25.

Kodni isječak 4.24 Konfiguracija "tailwind.config.js" datoteke

```
module.exports = {
    plugins: [
        require('flowbite/plugin')
    ]
    content: [
        './templates/**/*.html',
        './node_modules/flowbite/**/*.js'
      ],
      theme: {
        extend: {},
      },
        plugins: [],
}
```

Kodni isječak 4.25 Uključivanje Flowbite JavaScript datoteke

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/flowbite
    /1.8.1/flowbite.min.js"></script>
```

Posljednji korak prije korištenja Flowbite komponenti je učitavanje Tailwind CSS oznaka, prikazano naredbom u kodnom isječku 4.25.

Kodni isječak 4.26 Uvoz Tailwind oznaka

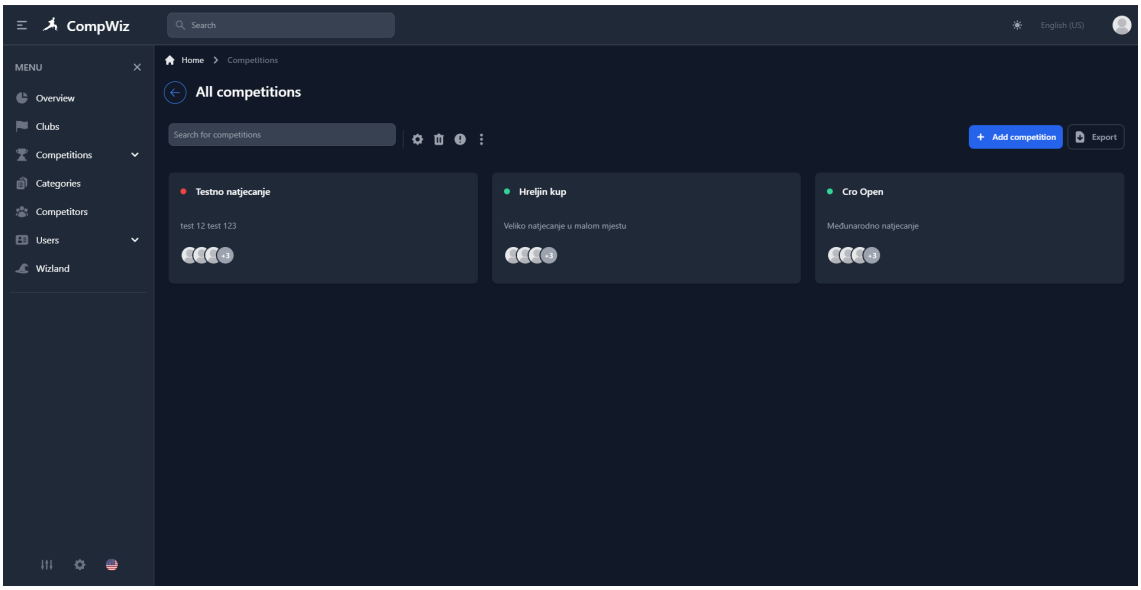
```
{% load tailwind_tags %}
```

Uvoz Tailwind CSS oznaka omogućava pristup različitim razredima za oblikovanje elemenata na internetskoj stranici. Idealno mjesto za uvoz Tailwind oznaka je datoteka ***"base.html"***. Nakon uvoza Tailwind CSS-a, možemo početi primjenjivati Flowbite komponente i prilagoditi izgled našeg sučelja. Primjer jedne Flowbite komponente može se vidjeti u kodnom isječku 4.27.

Kodni isječak 4.27 Prikaz koda Flowbite komponente

```
<button type="button" class="text-white bg-blue-700 hover:bg-
    blue-800 focus:ring-4 focus:ring-blue-300 font-medium
    rounded-lg text-sm px-5 py-2.5 mr-2 mb-2 dark:bg-blue-600
    dark:hover:bg-blue-700 focus:outline-none dark:focus:ring-
    blue-800">Default</button>
```

Zahvaljujući Flowbite okviru, aplikacija CompWiz ima dosljedan i privlačan dizajn koji je prikazan na slici 4.11.



Slika 4.11 *Prikaz dizajna CompWiz aplikacije*

Poglavlje 5

Zaključak

U ovom radu razvijena je aplikacija naziva "CompWiz" za učinkovito vođenje karate natjecanja. Aplikacija obuhvaća autentikaciju i autorizaciju korisnika, te pruža jednostavno sučelje za organizaciju natjecanja u karateu. CompWiz olakšava organizatorima vođenje natjecanja, upravljanje rezultatima i kategorijama te omogućava transparentno i učinkovito upravljanje karate natjecanjima.

Implementacija CompWiz aplikacije demonstrira snagu Django razvojnog okvira u izradi kompleksnih internetskih aplikacija, s posebnim naglaskom na sigurnost i skalabilnost. Ugrađeni sigurnosni mehanizmi i podrška za ACID transakcije osiguravaju zaštitu osjetljivih podataka o natjecateljima i rezultatima. Također, korištenje Tailwind CSS i Flowbite okvira omogućava brzo i dosljedno stiliziranje sučelja, što povećava atraktivnost aplikacije i korisničko iskustvo.

CompWiz aplikacija predstavlja korak naprijed u organizaciji karate natjecanja, s potencijalom da olakša i poboljša proces vođenja natjecanja za klubove, saveze i organizatore. U budućnosti, moguće je proširenje aplikacije dodavanjem dodatnih funkcionalnosti, kao npr. prijenos kategorija uživo, kako bi se dodatno unaprijedio proces organizacije i praćenja natjecanja.

Kroz ovaj rad, istražene su ključne komponente razvoja web aplikacije, uključujući implementaciju poslužitelja, autentikaciju i autorizaciju korisnika, upravljanje bazom podataka te stilizaciju i dizajn sučelja. CompWiz aplikacija predstavlja konkretan primjer primjene tih komponenata u praksi.

Kroz kontinuirani razvoj i unaprjeđenje, CompWiz ima potencijal postati vodeći alat za organizaciju karate natjecanja, pružajući korisnicima jednostavno i učinkovito rješenje za vođenje i praćenje natjecateljskih događanja u karateu.

Bibliografija

[1] World Karate Federation: "Karate natjecateljska pravila", s Interneta, https://www.karate.hr/web/sadrzaj/dokumenti/KARATE_NATJECATELJSKA_PRAVILA_2020.pdf, 3.rujna 2023.

[2] Modrić Ž.: "karate", Sportska štampa, Zagreb, 1968.

[3] Natjecateljska komisija Hrvatskog karate saveza: "KATEGORIJE za TURNIRE i OSTALA NATJECANJA za 2022. GODINU", s Interneta, https://www.karate.hr/web/sadrzaj/dokumenti/KATEGORIJE_TURNIRI_2022.pdf, 5.rujna 2023.

[4] Sportdata: "Prices and Licenses - Sportdata Event Technology", s Interneta, https://set.sportdata.org/wp/2012/07/04/prices-and-licenses/, 3.rujna 2023.

[5] BuiltWith: "ekarate.eu Technology Profile", s Interneta, https://builtwith.com/ekarate.eu, 3.rujna 2023.

[6] eKarate.eu: "Ranking - eKarate", s Interneta, https://www.ekarate.eu/hr/ranking/ranking/27.html, 3.rujna 2023.

[7] Sportdata: "PowerPoint-Präsentation", s Interneta, https://download.sportdata.org/sportdata-neu.pdf, 3.rujna 2023.

[8] Sportdata: "WKF Online Registration", s Interneta, https://www.sportdata.org/wkf/set-online/index.php, 3.rujna 2023.

[9] Sportdata: "Sportdata Event Technology – Software for professional event management", s Interneta, https://set.sportdata.org/, 3.rujna 2023.

[10] SimilarTech: "Django Market Share and Web Usage Statistics", s Interneta, https://www.similartech.com/technologies/django, 3.rujna 2023.

[11] Gowthamy Vaseekaran: "Django Framework — Best Practices", 26.siječnja 2020., s Interneta, https://gowthamy.medium.com/django-framework-best-practices-2c34137b1671, 3.rujna 2023.

[12] PostgreSQL: "PostgreSQL: About", s Interneta, https://www.postgresql.org/about/, 3.rujna 2023.

[13] IBM: "ACID properties of transactions - IBM Documentation", s Interneta, https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions, 4.rujna 2023.

[14] Tailwind CSS: "Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.", s Interneta, https://tailwindcss.com/, 4.rujna 2023.

[15] Flowbite: "Flowbite - Build websites even faster with components on top of Tailwind CSS", s Interneta, https://flowbite.com/, 4.rujna 2023.

[16] Docker: "Docker: Accelerated Container Application Development", s Interneta, https://www.docker.com/, 4.rujna 2023.

[17] PostrgeSQL Documentation: "PostgreSQL: Documentation: 15: 8.15. Arrays", s Interneta, https://www.postgresql.org/docs/current/arrays.html, 4.rujna 2023.

[18] narendrapalacharla: "Why SQLite is not for production?", 26.siječnja 2020., s Interneta, https://narendrapalach1.medium.com/why-sqlite-is-not-for-production-ba72777e3367, 4.rujna 2023.

[19] Django documentation - "User authentication in Django | Django documentation | Django", s Interneta, https://docs.djangoproject.com/en/4.2/topics/auth/, 5.rujna 2023.

[20] Django documentation - "Customizing authentication in Django | Django documentation | Django", s Interneta, https://docs.djangoproject.com/en/4.2/topics/auth/customizing/, 5.rujna 2023.

[21] Django documentation - "File Uploads | Django documentation | Django", s Interneta, https://docs.djangoproject.com/en/4.2/topics/http/file-uploads/, 5.rujna 2023.

[22] Django documentation - "Databases | Django documentation | Django", s Interneta, https://docs.djangoproject.com/en/4.2/ref/databases/, 5.rujna 2023.

[23] Django documentation - "Working with forms | Django documentation | Django", s Interneta, `https://docs.djangoproject.com/en/4.2/topics/forms/`, 5.rujna 2023.

[24] Django documentation - "Sending email | Django documentation | Django", s Interneta, `https://docs.djangoproject.com/en/4.2/topics/email/`, 5.rujna 2023.

[25] Django documentation - "How to manage static files (e.g. images, JavaScript, CSS) | Django documentation | Django", s Interneta, `https://docs.djangoproject.com/en/4.2/howto/static-files/`, 5.rujna 2023.

[26] Django-Tailwind 2.0.0 documentation- "Installation — Django-Tailwind 2.0.0 documentation", s Interneta, `https://django-tailwind.readthedocs.io/en/latest/installation.html`, 5.rujna 2023.

[27] Django documentation- "Making queries | Django documentation | Django", s Interneta, `https://docs.djangoproject.com/en/4.2/topics/db/queries/#complex-lookups-with-q-objects`, 5.rujna 2023.

[28] Django documentation - "Templates | Django documentation | Django", s Interneta, `https://docs.djangoproject.com/en/4.2/topics/templates/`, 5.rujna 2023.

[29] Django documentation - "The Django template language | Django documentation | Django", s Interneta, `https://docs.djangoproject.com/en/4.2/ref/templates/language/`, 5.rujna 2023.

[30] Flowbite - "Tailwind CSS Django - Flowbite", s Interneta, `https://flowbite.com/docs/getting-started/django/`, 5.rujna 2023.

26 done

Sažetak

U okviru ovog završnog rada razvijena je aplikacija naziva "CompWiz" koja služi za učinkovito vođenje karate natjecanja. Aplikacija koristi Django za backend i Tailwind CSS za frontend, integrira autentikaciju i autorizaciju korisnika, te pruža intuitivno sučelje za organizaciju natjecanja u karateu. CompWiz olakšava organizatorima vođenje natjecanja, upravljanje rezultatima i kategorijama te omogućava transparentno i učinkovito upravljanje karate natjecanjima.

Ključne riječi **— karate, vođenje natjecanja, Django, Tailwind CSS**

Abstract

Within the scope of this thesis, an application named "CompWiz" has been developed for the efficient management of karate competitions. The application utilizes Django for the backend and Tailwind CSS for the frontend, integrates user authentication and authorization, and provides an intuitive interface for organizing karate competitions. CompWiz simplifies competition management for organizers, enables result and category management, and allows transparent and efficient handling of karate competitions.

Keywords **— karate, competition management, Django, Tailwind CSS**