

Web-aplikacija za studentske domove temeljena na naprednim dvodimenzionalnim kartama

Šegvić, Bruno

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:996208>

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

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



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 278

**WEB-APLIKACIJA ZA STUDENTSKE DOMOVE TEMELJENA
NA NAPREDNIM DVODIMENZIONALNIM KARTAMA**

Bruno Šegvić

Zagreb, veljača 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 278

**WEB-APLIKACIJA ZA STUDENTSKE DOMOVE TEMELJENA
NA NAPREDNIM DVODIMENZIONALNIM KARTAMA**

Bruno Šegvić

Zagreb, veljača 2024.

DIPLOMSKI ZADATAK br. 278

Pristupnik: **Bruno Šegvić (0023121390)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: prof. dr. sc. Ivica Botički

Zadatak: **Web-aplikacija za studentske domove temeljena na naprednim dvodimenzionalnim kartama**

Opis zadatka:

U sklopu diplomskog rada izradit će se web-aplikacija za studentske domove koja se temelji na naprednim dvodimenzionalnim kartama. U diplomskom radu potrebno je istražiti i implementirati metode automatskog stvaranja dvodimenzionalnih karata domova temeljem fotografija domova. Sustav treba prepoznajući ulazne fotografije pripremljene prema unaprijed znanoj specifikaciji automatizirano stvoriti digitalnu reprezentaciju karte doma. Aplikacija na jednom mjestu, daje sve potrebne informacije o studentskim domovima, načinu upisa u dom i svoj potrebnoj dokumentaciji. Web-aplikacija treba sadržavati lokaciju domova, broj soba, opis soba, slike, klasifikaciju po paviljonima i dodatne informacije vezane za blizinu tramvaja, menza i slično. Od standardnih funkcionalnosti mora pružiti mogućnost prijave u sustav, komentiranja, ocjenjivanja domova te traženje pojmova i filtriranje. Od naprednih funkcionalnosti potrebno je izraditi modul za pregled, stvaranje i uređivanje dvodimenzionalnih karata domova. Svi podaci spremat će se u bazu podataka, a aplikacija će biti sastavljena od klijentskog dijela i poslužiteljskog dijela.

Rok za predaju rada: 9. veljače 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 278

**WEB-APLIKACIJA ZA STUDENTSKE DOMOVE
TEMELJENA NA NAPREDNIM
DVODIMENZIONALNIM KARTAMA**

Bruno Šegvić

Zagreb, veljača 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 278

**WEB-APLIKACIJA ZA STUDENTSKE DOMOVE
TEMELJENA NA NAPREDNIM
DVODIMENZIONALNIM KARTAMA**

Bruno Šegvić

Zagreb, veljača 2024.

Zagreb, 2. listopada 2023.

DIPLOMSKI ZADATAK br. 278

Pristupnik: **Bruno Šegvić (0023121390)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: prof. dr. sc. Ivica Botički

Zadatak: **Web-aplikacija za studentske domove temeljena na naprednim dvodimenzionalnim kartama**

Opis zadatka:

U sklopu diplomskog rada izradit će se web-aplikacija za studentske domove koja se temelji na naprednim dvodimenzionalnim kartama. U diplomskom radu potrebno je istražiti i implementirati metode automatskog stvaranja dvodimenzionalnih karata domova temeljem fotografija domova. Sustav treba prepoznajući ulazne fotografije pripremljene prema unaprijed znanoj specifikaciji automatizirano stvoriti digitalnu reprezentaciju karte doma. Aplikacija na jednom mjestu, daje sve potrebne informacije o studentskim domovima, načinu upisa u dom i svoj potrebnoj dokumentaciji. Web-aplikacija treba sadržavati lokaciju domova, broj soba, opis soba, slike, klasifikaciju po paviljonima i dodatne informacije vezane za blizinu tramvaja, menza i slično. Od standardnih funkcionalnosti mora pružiti mogućnost prijave u sustav, komentiranja, ocjenjivanja domova te traženje pojmova i filtriranje. Od naprednih funkcionalnosti potrebno je izraditi modul za pregled, stvaranje i uređivanje dvodimenzionalnih karata domova. Svi podaci spremat će se u bazu podataka, a aplikacija će biti sastavljena od klijentskog dijela i poslužiteljskog dijela.

Rok za predaju rada: 9. veljače 2024.

Sadržaj:

1. Uvod	1
1.1 Predmet i cilj rada	1
1.2 Izvori podataka i metode istraživanja	1
1.3 Struktura rada	1
2. Općenito o aplikaciji	2
2.1. Potreba za aplikacijom	2
2.2 Detekcija objekata na slikama	4
2.3 Arhitektura web aplikacije	4
3. Tehnologije korištene pri izradi aplikacije	8
3.1. TensorFlow	8
3.2 Vue 3	10
3.3. Django	11
3.4. PostgreSQL	16
3.5. Konva.js	18
4. Opis rada sustava	21
4.1 Model za detekciju objekata	21
4.2. Početna stranica	23
4.3. Stranica pojedinog studentskog doma	25
4.4. Registracija i prijavljivanje u sustav	28
4.5. Korisnički profil	30
4.6. Administracijsko sučelje	32
4.6.1. Izmjena podataka	32
4.6.2. Unos novih podataka	35
5. Zaključak	42
Literatura	43

1. Uvod

1.1 Predmet i cilj rada

S obzirom na velik broj studentskih domova u Republici Hrvatskoj, ali i brojne razlike među njima, za svakog studenta iznimno je važno donijeti informiranu odluku o odabiru pojedinog studentskog doma. Trenutno su podaci o ponudi studentskih domova, njihovim adresama, kapacitetima i kategorijama raštrkani čime je pronalazak studentskog doma otežan. Cilj ovog rada je pojednostaviti i ubrzati proces pronalaska i odabira adekvatnog studentskog doma svim studentima na području Republike Hrvatske.

1.2 Izvori podataka i metode istraživanja

Prilikom izrade rada korišteni su sekundarni izvori podataka prikupljeni s relevantnih internetskih stranica, dok je glavnina rada odrađena programiranjem, najviše u JavaScriptu i Pythonu.

1.3 Struktura rada

Rad se sastoji od pet cjelina. Prvo poglavlje odnosi se na predmet i cilj rada te su u njemu ukratko obrazložene metode istraživanja, kao i izvori podataka. U drugom poglavlju identificirana je potreba za aplikacijom te je obrazložena arhitektura same web aplikacije, dok su u trećem dijelu objašnjenje sve tehnologije koje su se koristile pri izradi aplikacije. U četvrtom dijelu rada detaljno je opisan rad same aplikacije, prijavljivanja u sustav, administracijskog sučelja te je sve popraćeno i s vizualima. U posljednjem dijelu rada ukratko je iznesen zaključak autora.

2. Općenito o aplikaciji

2.1. Potreba za aplikacijom

Početak osamdesetih godina prošlog stoljeća, u Republici se Hrvatskoj stavlja sve veći naglasak na dostupnosti povoljnog stanovanja za studente. Navedena je činjenica usko povezana s rastom studentske populacije u državi te dovodi do toga da studentski domovi u Republici Hrvatskoj predstavljaju bitnu infrastrukturu omogućavajući studentima ekonomski pristupačan smještaj za vrijeme studiranja.

Studenti u Republici Hrvatskoj imaju mogućnost prijaviti se na natječaj za smještaj u studentskom domu tijekom svojeg studiranja sukladno Pravilniku o uvjetima i načinu ostvarivanja prava redovitih studenata na subvencionirano stanovanje (NN 63/2019), a student koji se želi prijaviti na natječaj za smještaj prijavu podnosi u studentskom centru u gradu u kojem studira.

U godini 2022./2023. na natječaj za studentski dom moglo se prijaviti u jedanaest gradova u Republici Hrvatskoj i to:

1. Zagreb
2. Osijek
3. Rijeka
4. Zadar
5. Split
6. Pula
7. Šibenik
8. Karlovac
9. Varaždin
10. Slavonski brod
11. Virovitica

Specifičnosti i različitosti studentskih domova u Republici Hrvatskoj čine da pronalazak sveobuhvatnih informacija o studentskim domovima u različitim gradovima nije jednostavan. Određeni domovi su kategorizirani, dok drugi nisu pa se studenti često suočavaju s nedostatkom jasno definiranih smjernica o tome kojem domu pripada određena kategorija. Uz to, različiti gradovi, a i sami domovi, imaju različite specifičnosti u ponudi paviljona, kao i raznolike socijalne i kulturne aktivnosti. Moguće je i da unutar istog grada postoji više domova, svaki sa svojim vlastitim karakteristikama i kapacitetima.

Ranije navedeni razlozi dovode do toga da studenti moraju pretraživati različite izvore podataka kako bi prikupili sve relevantne informacije za odabir i upis u studentski dom. Lakše pretraživanje svih potrebnih informacija o studentskim domovima pojednostavnilo bi donošenje informirane odluke studenata o odabiru doma koji najbolje odgovara individualnim potrebama i preferencijama studenta ali i ubrzalo cijeli proces.

S obzirom na veliku popularnost i potrebu za smještajem u studentskim domovima, napravljena je web aplikacija koja pruža sveobuhvatne informacije o domovima diljem Republike Hrvatske. Cilj aplikacije je olakšati studentima proces pronalaska doma koji najbolje odgovara njihovim potrebama, mogućnostima i preferencijama. Pružanjem pregledne platforme studentima omogućuje se usporedba domova, pojednostavljuje se pristup informacijama, ubrzava se proces odabira i centraliziraju se svi potrebni podaci o studentskim domovima.

Aplikacija sadrži vizualni prikaz svakog studentskog doma unutar Republike Hrvatske s opisom, adresom, kapacitetom i detaljnim informacijama o svakom paviljonu, uključujući kategoriju, broj i opis. Sama aplikacija koristi prepoznavanje objekta (engl. Object recognition) pri unosu slika.

2.2 Detekcija objekata na slikama

Prepoznavanje objekata (engl. Object recognition) predstavlja tehnologiju unutar područja računalnog vida koja omogućava računalima prepoznavanje i identifikaciju objekata na fotografijama, videozapisima i sličnom. Tehnologija se temelji na strojnom učenju i dubokom učenju gdje algoritmi analiziraju velike skupove podataka kako bi naučili prepoznavati obrasce i karakteristike različitih objekata. Glavni cilj sustava za detekciju objekata jest automatsko prepoznavanje objekata na temelju njihovih karakteristika, što omogućuje računalima da razumiju i tumače okolinu slično ljudskom vidu.

U detekciji objekata postoji nekoliko razina složenosti:

1. Detekcija objekta: identifikacija prisutnosti objekta na slici ili u videu;
2. Klasifikacija objekata: pripisivanje identificiranog objekta određenoj kategoriji ili klasi (npr. automobil, osoba, paviljon, itd.);
3. Segmentacija objekta: identifikacija i označavanje pojedinačnih piksela koji pripadaju određenom objektu što omogućuje preciznije razumijevanje oblika objekta [1].

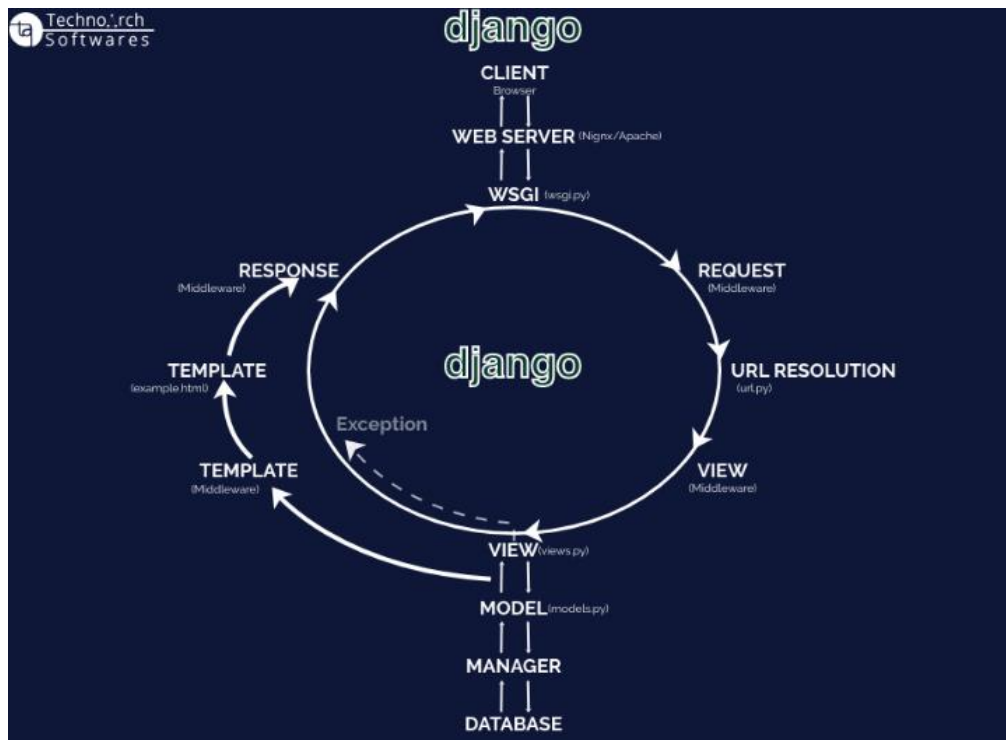
U aplikaciji korištene su prve dvije razine, to jest detekcija objekata (kao što su paviljoni, sportski tereni i slično) i klasifikacija tih objekata. S dovoljno velikim skupom za treniranje, ova aplikacija ima potencijal primjene širi od područja Republike Hrvatske.

2.3 Arhitektura web aplikacije

Aplikacija se sastoji od baze podataka, poslužitelja i korisničke aplikacije. Korisnici ostvaruju interakciju sa sustavom tako što koriste aplikaciju koja šalje zahtjeve poslužitelju, a poslužitelj potom obrađuje te zahtjeve koristeći podatke iz baze podataka. HTTP (engl. Hypertext Transfer Protocol) podržava različite tipove zahtjeva (engl.

request) koji se koriste za komunikaciju između klijenta, odnosno web preglednika i poslužitelja. Osnovni tipovi HTTP zahtjeva su:

1. GET
2. POST
3. PUT
4. DELETE
5. PATCH
6. HEAD
7. OPTIONS
8. TRACE



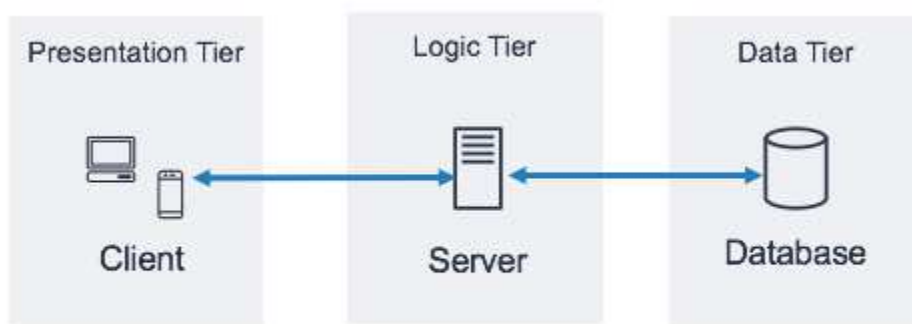
Slika 1: Django ciklus zahtjeva i odgovora [2]

Od navedenih zahtjeva, u okviru obrađene aplikacije korišteni su: GET, POST i PATCH.

GET zahtjev koristi se za dohvat resursa s poslužitelja. Kada se otvori web stranica, preglednik šalje GET zahtjev za HTML stranicom. U aplikaciji se on koristi za dohvaćanje podataka o domovima, paviljonima, korisnicima i slično.

POST zahtjev koristi se za slanje podataka prema poslužitelju kako bi se izvršila određena akcija. Kada se ispuni web obrazac klikom na “Predaj” (engl. Submit), preglednik šalje POST zahtjev s unesenim podacima. POST zahtjev u navedenoj aplikaciji koristi se za prijavu korisnika u sustav, registraciju korisnika, detekciju objekata na slikama, unos podataka o domovima i slično.

PATCH zahtjev koristi se za ažuriranje resursa na poslužitelju. Prilikom promjene imena i prezimena korisnika te u admin sučelju za promjenu podataka o studentskim domovima.



Slika 2: Arhitektura sustava [3]

Prilikom izrade web aplikacije korišten je web okvir (engl. *backend framework*) Django koji koristi arhitekturni obrazac MVT (engl. *Model-View-Template*), a koji je konceptualno vrlo sličan MVC-u (engl. *Model-View-Controller*). Model predstavlja centralnu komponentu obrasca koji se bavi definiranjem strukture podataka, interakcijom s bazom podataka i implementacijom poslovne logike. Također sadrži Django modele koji opisuju strukturu podataka u bazi podataka, obrađuje i manipulira podacima putem Django ORM (engl. *Object-Relational Mapping*), te se brine za implementaciju poslovne logike i pravila za manipulaciju podacima. [4]

Prikaz (engl. *View*) je odgovoran za vizualizaciju podataka koji se prikazuju korisnicima kroz HTTP odgovore te za obradu korisničkih zahtjeva.

Predložak (engl. *Template*) predstavlja sloj koji se bavi generiranjem statičkog sadržaja kao što je HTML, CSS i JavaScript i prikazivanjem podataka na korisničkom sučelju.

3. Tehnologije korištene pri izradi aplikacije

3.1. TensorFlow

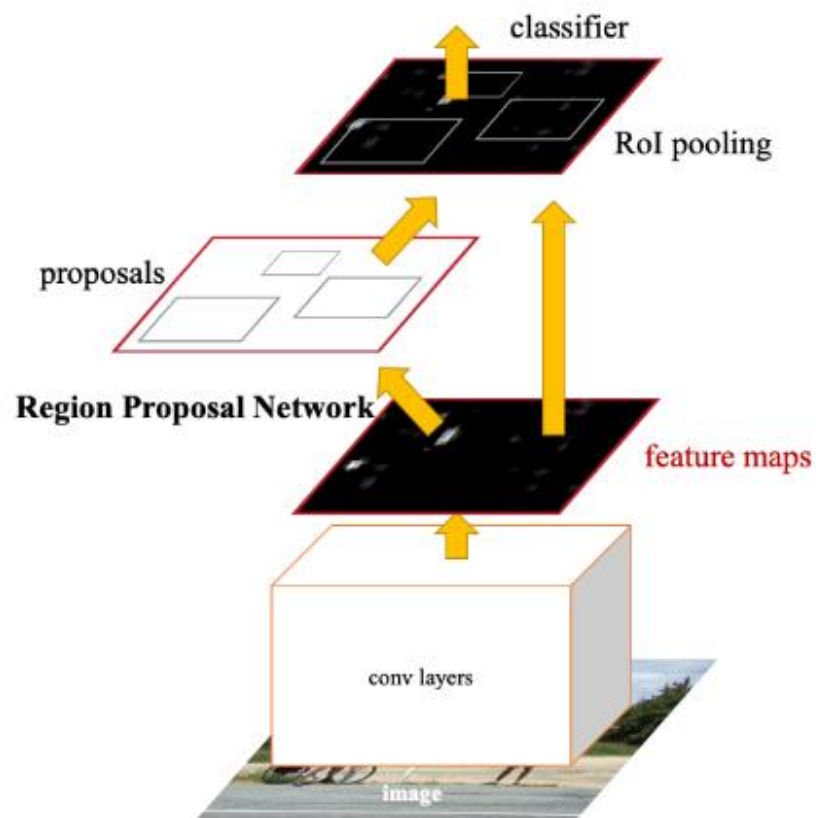
Tensorflow je otvorena platforma (engl. Open source) za strojno učenje koju je razvio Google LLC. Služi za razvoj i treniranje različitih vrsta modela za obradu podataka, a jedna vrsta modela je model za detekciju objekata (engl. Object detection). TensorFlow koristi grafički model računanja u kojem se operacije izračunavanja predstavljaju kao čvorovi u usmjerenom grafu, što omogućuje paralelno izvršavanje i optimizaciju performansi.

TensorFlow koristi tenzore (engl. *tensors*), to jest višedimenzionalne nizove podataka, kao osnovnu jedinicu podataka. Tensor može biti skalar ili tenzor ranga 0, vektor, odnosno tenzor ranga 1 ili matrica, što je tenzor ranga 2. Modeli se definiraju s pomoću slojeva (engl. *Layers*) koji se sastoje od operacija nad tenzorima. U TensorFlow-u se nalaze prethodno trenirani modeli (engl. *Pre-trained models*) za detekciju objekata kao što su SSD (engl. *Single Shot Multibox Detector*), Brži R-CNN (engl. *Region-based Convolutional Neural Network*), i drugi. Ovi modeli mogu se koristiti izravno ili dodatno istrenirati na novim podacima i prilagoditi prema specifičnim zahtjevima projekta. [5]

SSD jedan je od popularnijih modela koji može detektirati različite veličine okvira i velik broj klasa objekata, a pri radu koristi konvolucijske slojeve različitih veličina kako bi detektirao objekte različitih skala.

Brži R-CNN (engl. *faster Region-based Convolutional Neural Network*) je model za detekciju objekata koji se temelji na konvolucijskim neuronskim mrežama. Ovaj model koristi dvije mreže pri čemu jednu koristi za ekstrakciju značajki, a drugu za predikciju granica objekata. Osim toga, koristi RPN (engl. Region Proposal Network) za generiranje prijedloga regija koje bi mogle sadržavati objekte. RPN predviđa granice regija i vjerojatnosti da svaka regija sadrži objekt koji se detektira. [6]

Nakon što RPN generira prijedloge regija, RoI pooling (engl. *Region of Interest Pooling*) izvlači mapu značajki iz svake regije, bez obzira na njihovu početnu veličinu. Ovim se rješava problem zahtjeva fiksne veličine slike za mrežu za otkrivanje objekata. Nakon RoI pooling-a dobivene značajke koriste se za klasifikaciju objekata i regresiju granica. Klasifikacija određuje kojoj klasi objekta regija pripada, a regresija prilagodi granice regije.



Slika 3: Vizualizacija bržeg R-CNN modela [6]

3.2 Vue 3

Vue 3 predstavlja JavaScript okvir mrežne aplikacije s otvorenim kodom (engl. *Framework*) za izgradnju korisničkih sučelja i aplikacija (engl. *Single-Page Application*). Kreirao ga je Evan You nakon što je radio za Google i koristio AngularJS, a sa željom da napravi okvir koji će imati karakteristike koje su mu se sviđale kod AngularJS-a:

“I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight.” [7]

Vue 3 se nadovezuje na standardni HTML, CSS i JavaScript te pruža deklarativni model programiranja temeljen na komponentama i reaktivnosti. Dizajniran je da bude fleksibilan i prilagodljiv. Osnovna biblioteka fokusirana je samo na sloj pogleda (engl. *View layer*), a za potrebe složenijih aplikacija postoje napredne značajke kao što su usmjeravanje, upravljanje stanjem i slično.

Vue 3 omogućuje proširenje HTML-a s HTML atributima koji se nazivaju direktivama, a koje nude funkcionalnost HTML aplikacijama. Direktive mogu biti ugrađene ili korisnički definirane.

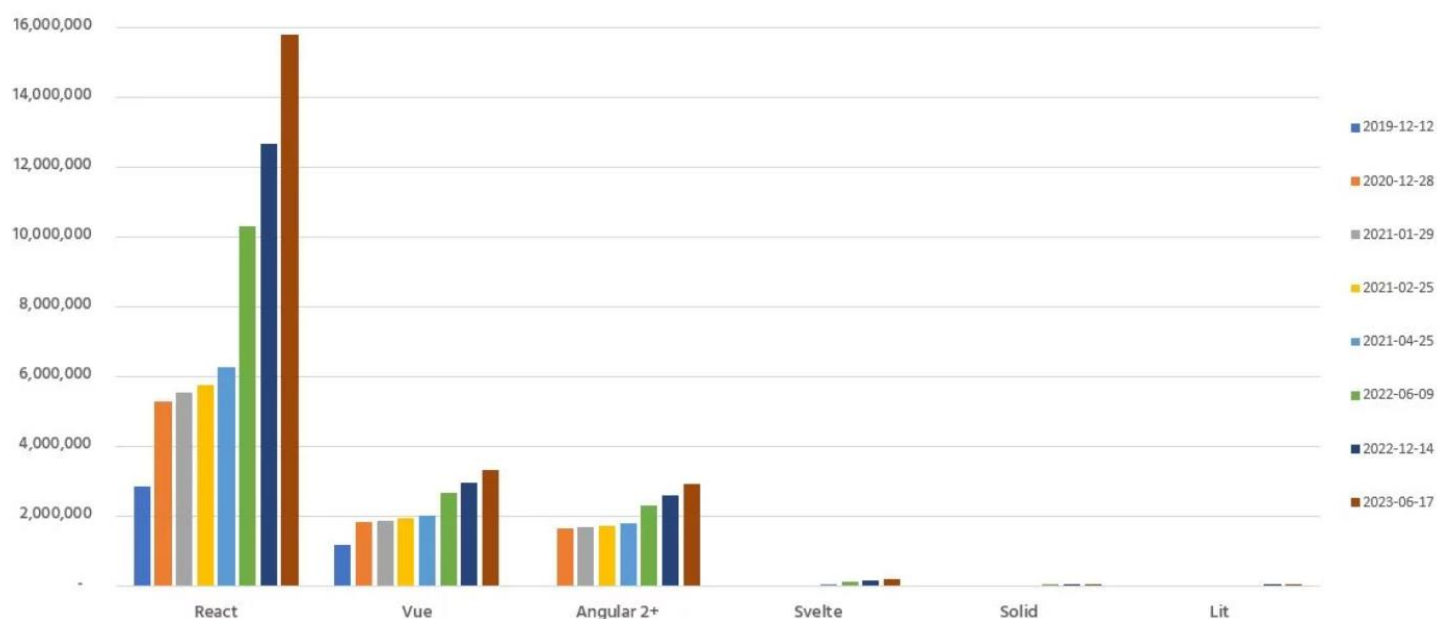
VueJS se može efikasno implementirati u kontekstima kao što su:

1. Poboljšanje statičkog HTML-a;
2. Ugradnja kao web komponente na bilo koju stranicu;
3. Dinamičke aplikacije s jednom stranicom (engl. *Single-Page Application*);
4. SSR (engl. *Server-Side Rendering*);
5. ... [8]

Prema trenutnoj statistici, 318.435 web stranica koristi VueJS, a prema slici na kojoj je prikazan broj repozitorija na GitHub-u, vidljivo je da je jedan od najpopularnijih okvira web dizajna. Od tri najpopularnija okvira, React, Angular i Vue, potonji je najmlađi. AngularJS

u javnost je izašao 2010. godine, ReactJS 2013. godine, , a VueJS u veljači 2014. godine.
[9]

Slika 4: Grafički prikaz broja repozitorija koji koriste web okvire na GitHub-u [11]



3.3. Django

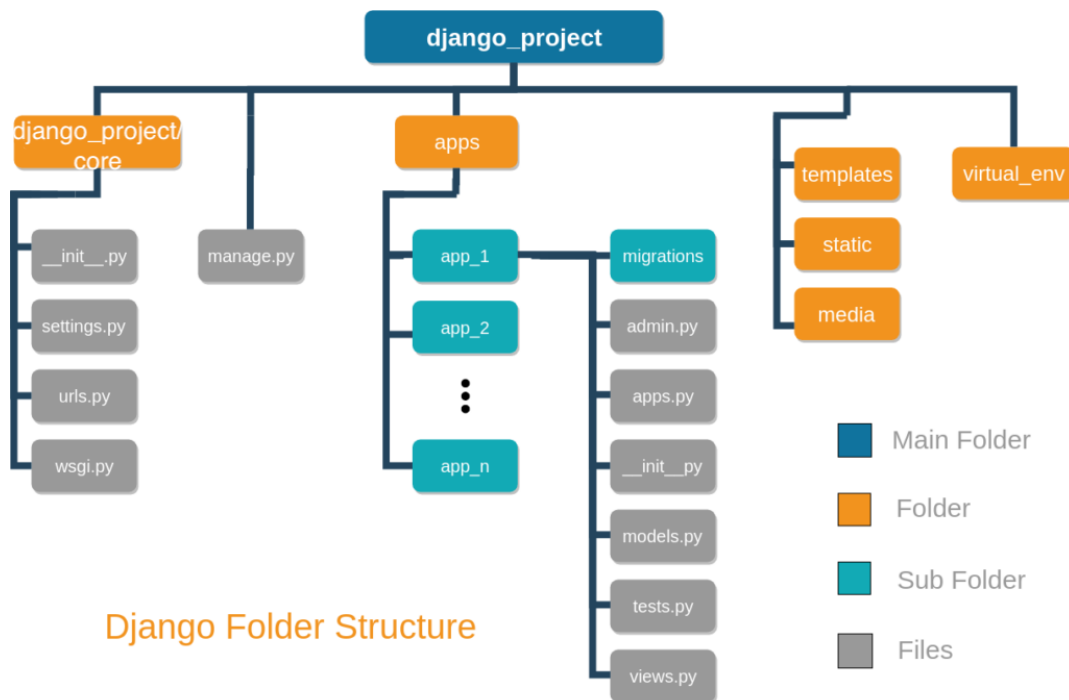
Django je besplatan web okvir otvorenog koda koji se temelji na programskom jeziku Python, a radi na web poslužitelju te slijedi arhitektonski obrazac MTV (engl. *Model-template-views*). Primarni cilj Django je olakšati stvaranje složenih web stranica s bazom podataka. Okvir omogućava ponovnu upotrebu komponenti, pisanje manje koda, brz

razvoj i načelo “*ne ponavljaj se*”. Python se koristi čak i za postavke, datoteke i modele podataka. Također nudi administrativno sučelje za čitanje, ažuriranje, brisanje i kreiranje koje se generira dinamički i konfigurira putem administratorskih modela.

Django ima definiranu nomenklaturu, ali osnova njegovog okvira može se promatrati kao MVC arhitektura. Sastoji se od sloja objektno-relacijskog mapera koji se može koristiti za interakciju s podacima iz različitih relacijskih baza podataka kao što su SQLite, PostgreSQL i MySQL, a ujedno i omogućuje dodavanje, brisanje, modificiranje i postavljanje upita objektima koristeći API (engl. *Application Programming Interface*) koji se zove ORM (engl. *Object Relational Mapping*). [10]

Django se isporučuje s raznim dodatnim, opcionalnim alatima koji rješavaju uobičajene probleme web razvoja. Taj dio koda se nalazi u `django/contrib` Django distribucije. Neki od tih alata su sljedeći:

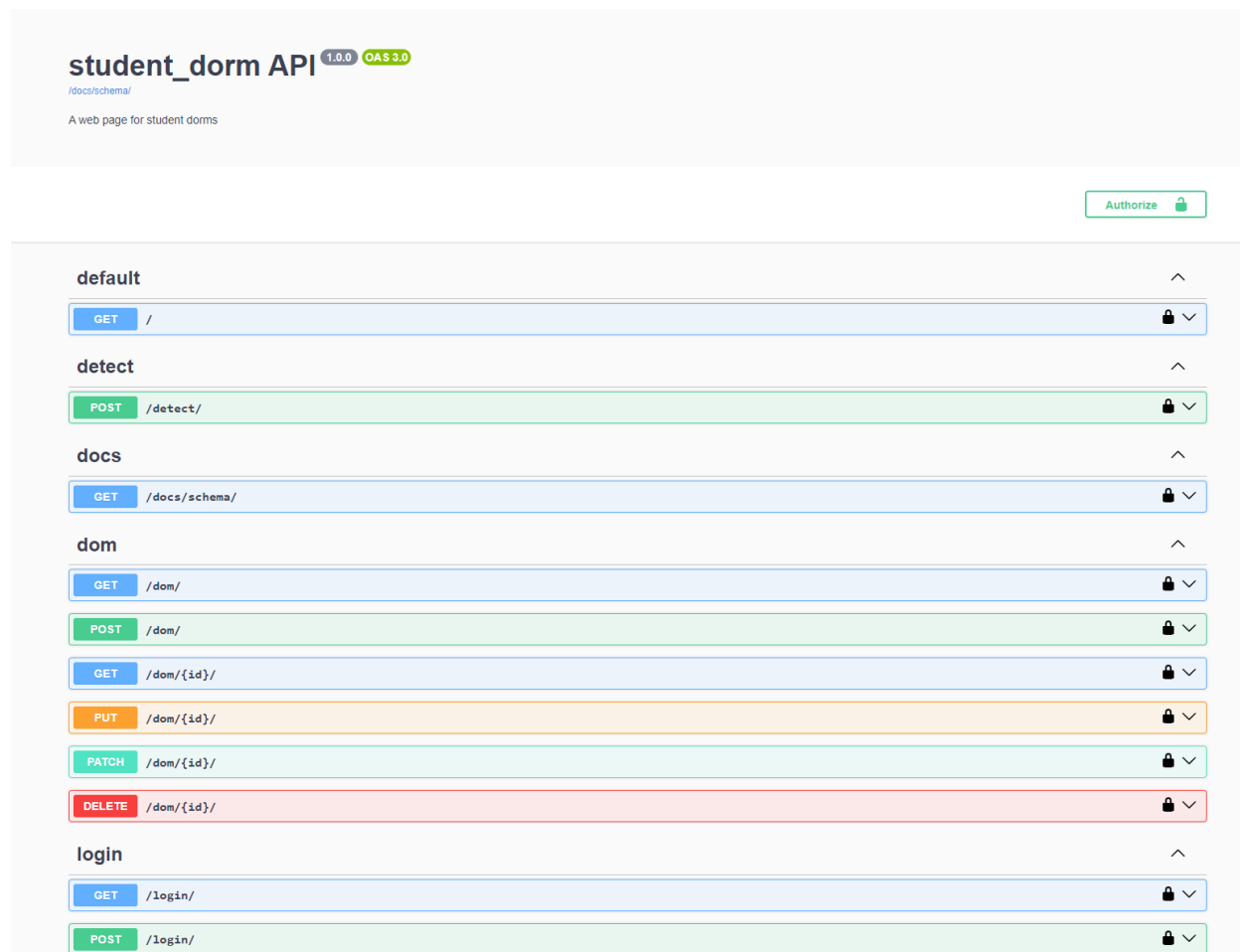
1. Autentifikacijski sustav;
2. Dinamičko administrativno sučelje;
3. “*Sites*” alat koji jednoj Django instalaciji omogućuje pokretanje više web stranica.



Slika 5: Struktura Django projekta [11]

SwaggerUI je alat otvorenog koda koji se može koristiti u različitim slučajevima kao što su razvoj, interakcija s API-jem i dokumentacija te može automatski generirati dokumentaciju na temelju samog koda.

Na URL-u “localhost:8000/docs” nalazi se interaktivna dokumentacija web stranice koja je predmet ovog rada, a koja je generirana od strane alata “SwaggerUI”. Spomenuti alat generira detaljnu dokumentaciju za API, uključujući sve rute, zahtjeve tih ruta (kao što su GET, POST, PATCH), parametre koji se šalju u zahtjevu i parametre koje odgovor šalje.



Slika 6: Dokumentacija generirana pomoću SwaggerUI

SwaggerUI omogućava testiranje zahtjeva na jednostavan način. Proširenjem retka u kojem se nalazi primjerice POST zahtjev dobije se modal kao na slici 6. Upisivanjem odgovarajućih vrijednosti i izvršavanjem zahtjeva na efikasan način se može provjeriti funkcionira li taj zahtjev. [12]

GET /dom/{id/}

Parameters Cancel

Name	Description
id * required integer (path)	A unique integer value identifying this dom.

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/dom/42/' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8000/dom/42/
```

Server response

Code Details

200

Response body

```
{
  "name": "Put Stanova ia",
  "address": "Put Stanova ia, 23000 Zadar",
  "city": "Zadar",
  "description": "Studentski dom raspola\u017ee sa 129 soba, od \u00e7ega 32 trokrevetne, 91 dvokrevetna i 6 jednokrevetnih za osobe s invaliditetom. Dom je opremljen sa tri \u00e7ajne kuhinje, praonicom rubij
a i u\u00e7ionicom.",
  "capacity": 284,
  "satellite_image": "http://localhost:8000/static/temp_aud8nez.jpeg",
  "image": "http://localhost:8000/static/placePutOsijek.jpg",
  "bounding_boxes": [
    {
      "x": 0.30846873228832367,
      "y": 0.26518359604505815,
      "name": "1866151",
      "width": 0.3338815789473684,
      "height": 0.47594501710213056,
      "scaleX": 0.8763846174146744,
      "scaleY": 1.01753963520766,
      "stroke": "red",
      "rotation": -5.010380829705892,
      "draggable": true,
      "paviljoni": {
        "number": "1",
        "kategorija": "3",
        "description": "Studentski dom raspola\u017ee sa 129 soba, od \u00e7ega 32 trokrevetne, 91 dvokrevetna i 6 jednokrevetnih za osobe s invaliditetom. Dom je opremljen sa tri \u00e7ajne kuhinje, praonicom
rublja i u\u00e7ionicom."
      }
    },
    {
      "strokeWidth": 2
    }
  ]
}
```

Response headers

```
allow: GET,PUT,PATCH,DELETE,HEAD,OPTIONS
content-length: 967
content-type: application/json
cross-origin-opener-policy: same-origin
date: Tue, 06 Feb 2024 18:37:40 GMT
referrer-policy: same-origin
server: WSGIServer/0.2 CPython/3.11.2
vary: Accept,origin,Cookie
x-content-type-options: nosniff
x-frame-options: DENY
```

Responses

Code	Description	Links
200	Media type <input type="text" value="application/json"/> Controls Accept header. Example Value Schema	No links

```
{
  "id": 0,
  "name": "string",
  "address": "string",
  "city": "string",
  "description": "string",
  "capacity": 2147483647,
  "satellite_image": "string",
  "image": "string",
  "bounding_boxes": "string"
}
```

Slika 7: GET zahtjev izvr\u0161en unutar SwaggerUI

3.4. PostgreSQL

PostgreSQL besplatan je sustav za upravljanje bazama podataka. Otvorenog je koda, a naglasak je na proširivost i usklađenost sa SQL-om. Razvijen je od strane PostgreSQL Global Development Group, zajednice programera i korisnika koji su radili na razvoju i održavanju ovog sustava za upravljanje bazama podataka. Sami početci su krenuli na Berkeleyju, na sveučilištu Kalifornija (engl. *University of California*). Prva verzija, postgres, bila je istraživački projekt, u kojem se eksperimentiralo s konceptima objektno-orijentiranog modeliranja, tipova podataka i podrške za transakcije, a prva javno dostupna verzija PostgreSQL-a izašla je u javnost 29. siječnja 1997. godine.

PostgreSQL se pridržava svojstava atomarnost, konzistentnosti, izolacije i izdržljivosti (ACID) te sadrži transakcije s automatski ažuriranim pogledima, materijaliziranim pogledima, okidačima, stranim ključevima i pohranjenim procedurama. ACID svojstva osiguravaju da skup operacija baze podataka ostavi bazu podataka u konzistentnom stanju, čak i u slučaju neočekivanih pogrešaka.

1. Atomarnost osigurava da se cijela transakcija prekine ako bilo koja operacija s bazom u sklopu transakcije završi pogreškom.
2. Konzistentnost omogućava da transakcije mogu dovesti bazu podataka iz jednog ispravnog stanja u drugo ispravno stanje, pri čemu ispravno ne znači da su podaci točno uneseni.
3. Svojstvo izolacije omogućuje da dvije istovremeno izvršene različite transakcije daju isti rezultat kao da su te iste transakcije izvedene jedna iza druge.
4. Izdržljivost jamči da se, jednom kada je transakcija potvrđena, njezini učinci zadržavaju čak i u slučaju kvarova sustava, poput prekida napajanja ili rušenja. Potvrđene promjene trajno su pohranjene u bazi podataka i nisu izgubljene,

osiguravajući da se baza podataka može oporaviti u dosljedno stanje nakon neuspjeha.

PostgreSQL upravlja svojom internom sigurnošću na temelju uloga s tim da uloga može biti korisnik ili grupa. Dopuštenja se mogu dodjeljivati za bilo koji objekt do razine stupca, a dopušta se ili sprječava čitanje, stvaranje, izmjena i brisanje objekata na razini baze podataka, sheme, tablice i redaka.

Značajne organizacije i proizvodi koji koriste PostgreSQL kao primarnu bazu podataka su:

1. Microsoft;
2. Myspace;
3. OpenStreetMap;
4. Sony Online;
5. Reddit;
6. Skype;
7. Instagram;
8. TripAdvisor;
9. OpenAI. [13]

Get Data / Dorm List

Dorm List

OPTIONS

GET

GET /dom/

HTTP 200 OK

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

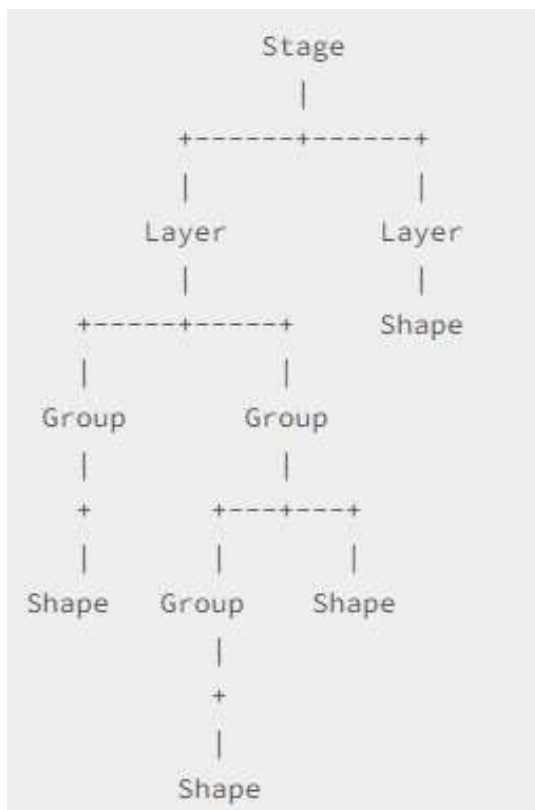
```
[
  {
    "id": 47,
    "name": "Dubrovnik",
    "address": "Ul. Marka Marojice 28, 20000 Dubrovnik",
    "city": "Dubrovnik",
    "description": "Dom se nalazi u samom centru grada Dubrovnika i na pješačkoj udaljenosti odjelima Sveučilišta u Dubrovniku, ali i ostalim visokim učilištima u Dubr",
    "capacity": 503,
    "satellite_image": "http://localhost:8000/static/temp_zlhgzNM.jpeg",
    "image": "http://localhost:8000/static/placeDubrovnik.jpg",
    "bounding_boxes": [
      {
        "x": 0.8582554517133957,
        "y": 0.14492753623188406,
        "name": "55180",
        "width": 0,
        "height": 0,
        "stroke": "red",
        "draggable": true,
        "strokeWidth": 2
      }
    ],
    {
      "x": 0.16288157413503518,
      "y": 0.19835766650796052,
      "name": "77456",
      "width": 0.6806853582554517,
      "height": 0.6992753623188406,
      "scaleX": 0.9748962025797667,
      "scaleY": 0.6887220918438512,
      "stroke": "red",
      "rotation": 9.034168781895682,
      "draggable": true,
      "paviljoni": {
        "number": "1",
        "kategorija": "2",
        "description": "Sobe u domu u Dubrovniku imaju vlastitu kupaonicu i čajnu kuhinju na svakom katu. Od ostalih sadržaja ima teretanu, praonice, zajedničk",
      },
      "strokeWidth": 2
    }
  }
]
```

Slika 8: Primjer ntorke u relaciji "dom"

3.5. Konva.js

Konva je HTML5 Canvas JavaScript okvir koji proširuje standarde mogućnosti HTML kanvasa omogućavanjem interaktivnosti platna za desktop i mobilne aplikacije. Ona omogućuje animacije visokih performansi, prijelaze, slojevitost, filtriranje, rukovanje događajima i drugo.

Konva.Stage sadrži nekoliko korisničkih slojeva, od kojih svaki sloj ima dva prikazivača (engl. *Renderer*) te svaki sloj može sadržavati oblike, grupe oblika ili grupe drugih grupa. Hijerarhija čvorova prikazana je na slici 8. [14]



Slika 9: Hijerarhija čvorova alata Konva [14]

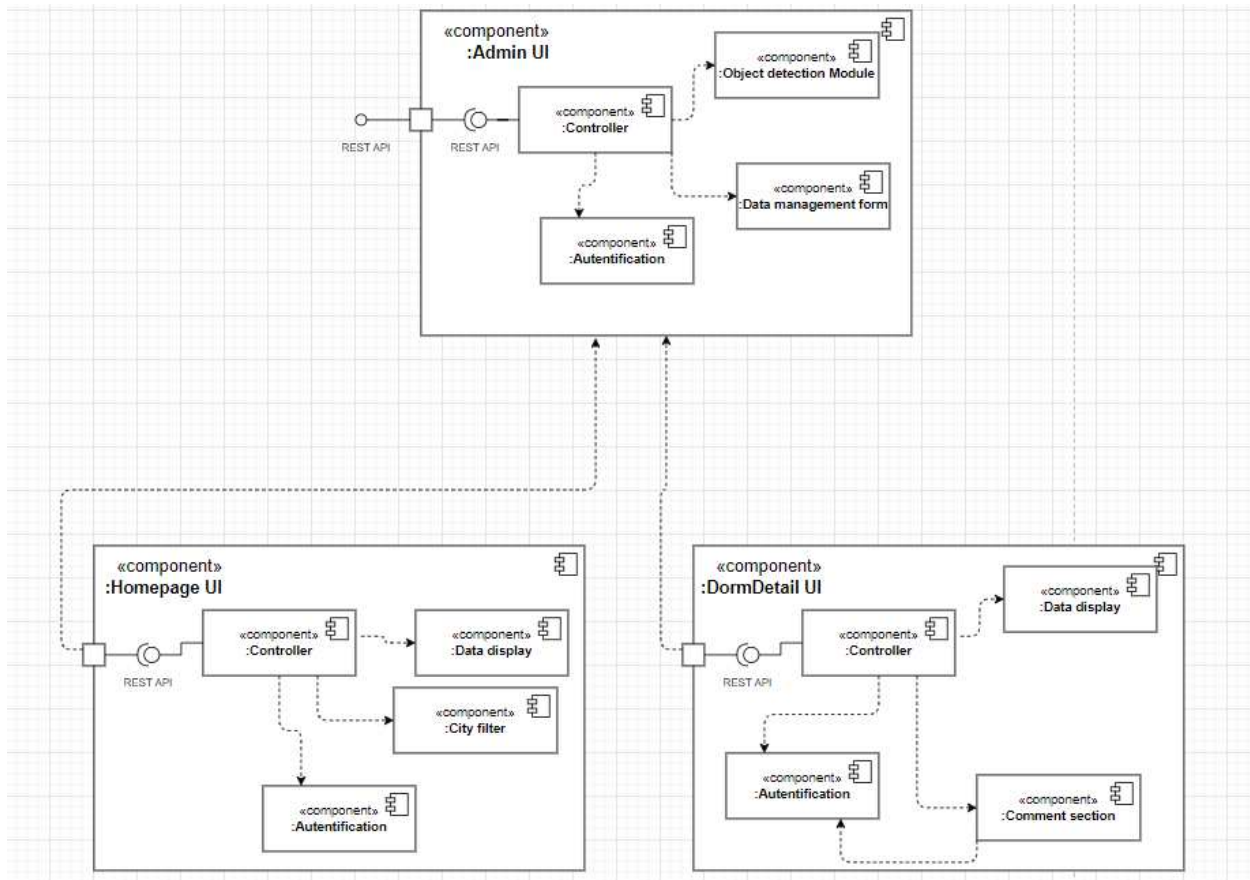
Sljedeći kôd ilustrira kako se u web okviru Vue3 može postaviti pozornica (engl. *stage*) alata Konva:

```
<v-stage ref="stage"
  :config="canvas"
  @mousedown="handleStageMouseDown"
  @touchstart="handleStageMouseDown"
  @mouseup="handleMouseUp"
  @mousemove="handleMouseMove"
  @click="selectPaviljon"
  >
  <v-layer>
    <v-image v-if="stageImage.image" v-bind="stageImage" />
    <v-rect
      v-for="item in boxes"
      v-bind="item"
      @dragend="setBoxPosition($event, item)"
      @transformend="setBoxPosition($event, item)"
    />
    <v-transformer
      ref="transformer"
      v-bind="transformerProps"
    />
  </v-layer>
</v-stage>
```

Na pozornici su postavljeni različiti događaji miša koji pozivaju različite funkcije. Svaki pravokutnik generira se iz liste pravokutnika s pomoću ugrađene funkcionalnosti kreiranja objekta. Za svaki tako generiran objekt moguće je postaviti različite događaje koji definiraju što se dogodi kada korisnik prestane povlačiti pravokutnik po platnu.

4. Opis rada sustava

Ovo poglavlje pruža detaljan uvid u aplikaciju, opis funkcionalnost, slike stranica i opis kôda. Na slici 10 prikazan je komponentni dijagram koji definira konceptualne module sustava i njihovu povezanost.



Slika 10: Komponentni dijagram

4.1 Model za detekciju objekata

Prvi korak u generiranju modela za detekciju objekata jest priprema slika. Da bi se iz prethodno treniranog modela dobio model za detekciju željenih objekata, treba pripremiti skup slika. U tu svrhu korišten je skup od 143 slike, a one su podijeljene na skup za treniranje i skup za testiranje u omjeru 80:20.

Drugi korak je označavanje slika u skupu za treniranje. U tu svrhu korišten je alat “*Labellmg*” u Pythonu. Na svakoj slici ručno su označeni i anotirani paviljoni, a označene slike spremljene su u posebnu mapu sa XML datotekom.

Treći korak koji je potreban za treniranje modela je konverzija XML datoteka u CSV zbog toga što je to tip datoteke koji TensorFlow očekuje. Iz CSV-a se generira jednostavan format za pohranu niza binarnih zapisa (engl. *Tfrecords*).

Konverzija iz XML-a u CSV je realizirana sljedećom programskom funkcijom:

```
def xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                    int(root.find('size')[0].text),
                    int(root.find('size')[1].text),
                    member[0].text,
                    int(member[4][0].text),
                    int(member[4][1].text),
                    int(member[4][2].text),
                    int(member[4][3].text)
                    )
            xml_list.append(value)
    column_name = ['filename', 'width', 'height', 'class', 'xmin', 'ymin',
                  'xmax', 'ymax']
    xml_df = pd.DataFrame(xml_list, columns=column_name)
    return xml_df
```

U ovom je koraku potrebna mapa oznaka koja sadrži imena oznaka i jedinstveni identifikator svake oznake. Ta mapa koristit će se kao ulazna datoteka pri učenju modela.

S obzirom na to da se koristi prethodno trenirani model, potrebno je pokrenuti skriptu za preuzimanje modela koji se nalazi na službenom TensorFlow Github repozitoriju. Preuzeti model ima “.config” datoteku u kojoj je potrebno promijeniti broj klasa, broj slika koje model istovremeno trenira, tip kontrolne točke te ulazne datoteke, odnosno slike za testiranje i slike za treniranje.

Posljednji korak prije korištenja novog modela jest treniranje modela. U ovom se koraku generira datoteka u kojoj se nalazi model s težinama. U tom je trenutku model spreman za korištenje.

4.2. Početna stranica

Početnoj stranici mogu pristupiti sve vrste korisnika: anonimni, prijavljeni i admin korisnik. Početna stranica se sastoji od dvije komponente – jedna komponenta je navigacija i banner slika koju vidimo na slici 9., a druga je lista svih domova u Republici Hrvatskoj vidljiva na slici 10.

Anonimni i prijavljeni korisnici u navigaciji imaju dvije poveznice. Jedna poveznica vodi korisnike na početnu stranicu, dok ih druga vodi na profil. Admin korisnik ima i treću poveznicu koja vodi na admin stranicu za unošenje novog studentskog doma u bazu podataka.



Slika 11: Prva komponenta početne stranice

Druga komponenta sadrži listu svih domova unesenih u bazu podataka, a prikazana je slika doma i ime doma. Osim toga, postoji i funkcionalnost filtriranja prema gradu unutar Republike Hrvatske.

Studentski domovi

Svi domovi ▾



Dubrovnik



Stjepan Radić "Sava"



Cvjetno naselje



Kampus Dr. Franjo Tuđman



Dom A



I. G. Kovačić



Put Stanova 1a



K. P. Svačića



Trsat

Slika 12: Druga komponenta početne stranice

Klikom na bilo koji od studentskih domova na početnoj stranici otvara se nova stranica sa svim informacijama o tom konkretnom domu.

4.3. Stranica pojedinog studentskog doma

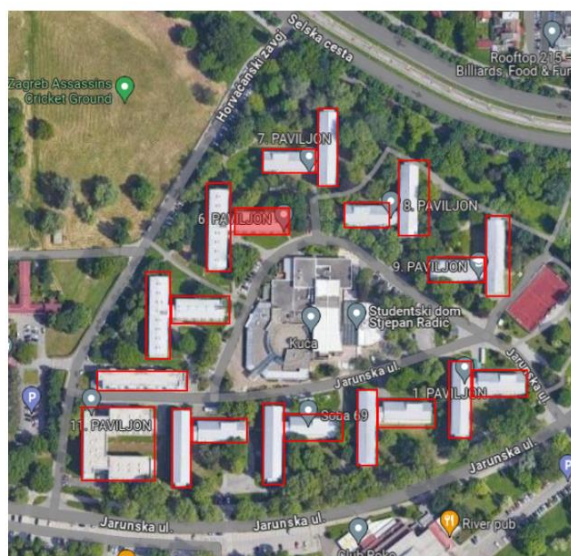
Na stranici svakog studentskog doma prikazano je ime studentskog doma, adresa, kratak opis svakog doma te satelitska snimka studentskog doma s istaknutim paviljonima i/ili zgradama doma unutar pravokutnika generiranih modelom za detekciju objekata na slikama.



Stjepan Radić "Sava"

Jarunska 2, 10000 Zagreb

Studentski dom Stjepan Radić, sastoji se od 12 paviljona, broji preko 4000 kreveta. Od popratnog sadržaja ima fitness centar, kino dvoranu, učionice, pastoralni ured, liječničku i stomatološku ordinaciju, plesnu dvoranu, dvije menze, pizzeriu te caffe Slastičarnu.



Slika 13: Stranica studentskog doma Sava

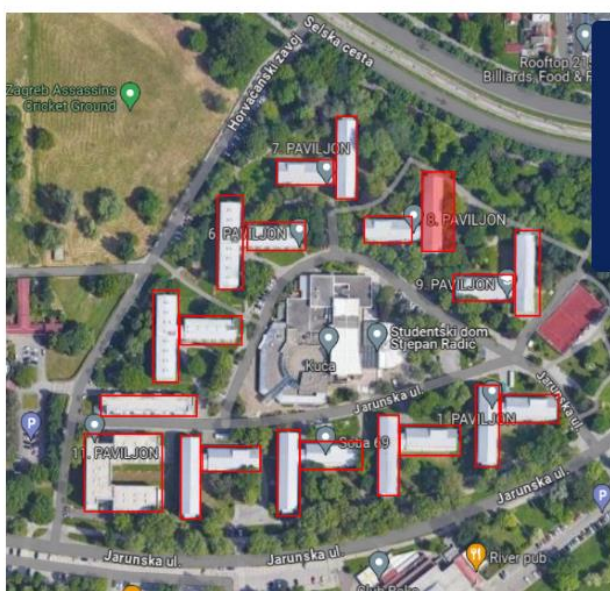
Slika je postavljena unutar kanvas HTML elementa što omogućuje crtanje pravokutnika i različite događaje (engl. *Event*) miša. Prelaskom miša preko ucrtanih pravokutnika mijenja se boja na način da se pravokutnik na kojem je miš pozicioniran istakne.

Pravokutnici su klikabilni, te se klikom miša otvara modal u kojem su prikazani detalji paviljona, kao što je prikazano na slici 12.

Stjepan Radić "Sava"

Jarunska 2, 10000 Zagreb

Studentski dom Stjepan Radić, sastoji se od 12 paviljona, broji preko 4000 kreveta. Od popratnog sadržaja ima fitness centar, kino dvoranu, učionice, pastoralni ured, liječničku i stomatološku ordinaciju, plesnu dvoranu, dvije menze, pizzeriu te caffe Slastičarnu.



Paviljon broj: 8

Opis paviljona: U ovom paviljonu nalaze se dvokrevetne sobe sa zajedničkim tuševima, toaletima i kuhinjama po katu. Označeno je veliko krilo osmog paviljona.

Kategorija paviljona: 3

Slika 14: Modal sa opisom paviljona

Prikazani modal ima postavljenu CSS vrijednost svojstva pozicije ljepljivo (engl. *Position sticky*). Navedeno omogućava da modal bude vidljiv korisniku bez obzira na visinu slike zbog toga što će modal biti pozicioniran pokraj slike, unutar prozora za prikaz (engl. *Viewport*).

Na ovoj stranici prijavljeni korisnik ima mogućnost komentiranja što je vidljivo na slici 15.



Slika 15: Komentari

4.4. Registracija i prijavljivanje u sustav

Kada anonimni korisnik klikne na poveznicu za profil unutar navigacije, korisnik se preusmjerava na registracijsku formu prikazanu na slici 13. Kako bi se registrirao, korisnik treba upisati korisničko ime, email adresu, ime, prezime i lozinku te je pritom važno da korisničko ime i email adresa budu jedinstveni. Klikom na gumb “*Login*” korisnik se preusmjerava na stranicu za prijavu.

Sign Up

With Your Email

Username

Email

First Name

Last Name

Password

Re-Type Password

Slika 16: Registracija u sustav

Korisnik koji već ima kreiran korisnički račun u sustav se može prijaviti preko forme za prijavu i to tako da upiše svoje korisničko ime i lozinku. Svaki korisnik ima mogućnost označiti polje “Zapamti me” kako se ne bi trebao svakog puta iznova prijavljivati na portal, dok će u suprotnome korisnikova sesija trajati dok ne zatvori pretraživač.

The image shows a login form with a teal border. It contains the following elements:

- Username:** A text input field with an envelope icon on the left and the placeholder text "e.g. bobsmith@gmail.com".
- Password:** A text input field with a lock icon on the left and placeholder text "*****".
- Remember me:** A checkbox followed by the text "Remember me".
- Login:** A green button with the text "Login".

Slika 17: Prijava u sustav

4.5. Korisnički profil

Nakon prijave u sustav, korisnik ima pristup stranici profila koja sadrži korisnikove osnovne informacije kao što su korisničko ime, ime, prezime i email adresa. Korisnik može promijeniti svoju sliku profila, svoje ime i prezime klikom na gumb "Uredi".

The image shows a user profile page for 'CROSTUDENT'. The page has a dark blue header with the logo 'CROSTUDENT' on the left and navigation icons (home, mail, profile) on the right. The main content area is divided into two sections:

- Profile Picture:** A circular placeholder for a profile picture with a green 'Odjava' button below it.
- Profil:** A table of user information with a green 'Uredi' button at the bottom right.

Username	admin
First name	Bruno
Last name	Šegvić
Email	admin@admin.admin

Slika 18: Profil korisnika

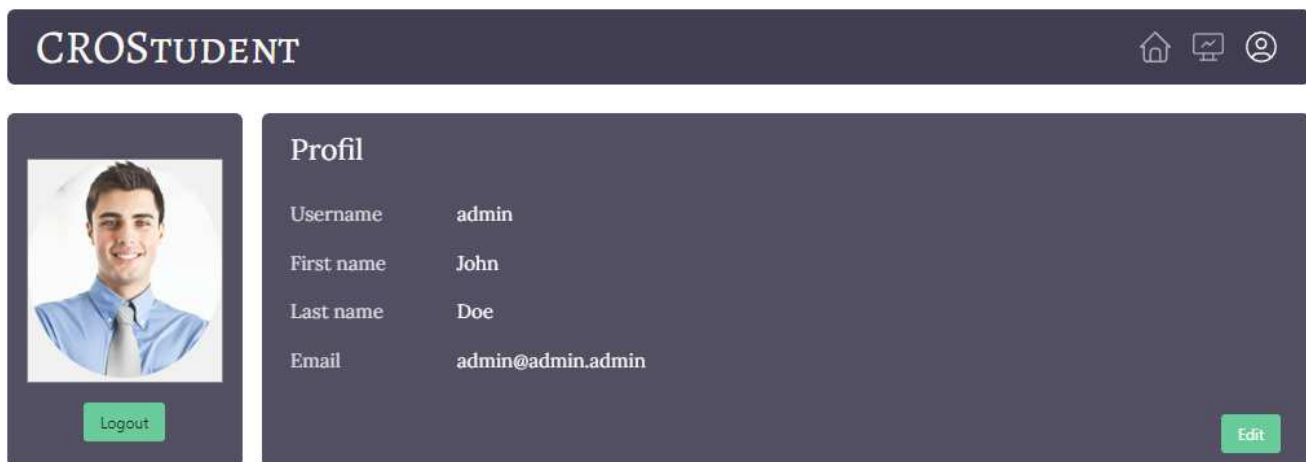
Klikom na gumb “Uredi” otvara se modal za promjenu podataka. Korisniku, u ovom slučaju adminu, je prikazano trenutno ime i prezime te on, ako tako želi, može unijeti novo ime, prezime ili jedno od toga. Osim toga, klikom na profilnu fotografiju korisnik ju može zamijeniti s nekom drugom fotografijom.



The screenshot shows a dark-themed modal window titled "CROSTUDENT" in the top left corner. In the top right corner, there are three icons: a home icon, a monitor icon, and a user profile icon. The modal contains two input fields: "Ime" (Name) with the value "Bruno" and "Prezime" (Surname) with the value "Segvić". Below the input fields is a green button labeled "SPREMI" (Save). A close button (X) is located in the top right corner of the modal.

Slika 19: Modal za promjenu imena i prezimena

Nakon izmjene podataka o imenu, prezimenu i promjeni profilne fotografije, profil korisnika izgleda kao na slici 18.



The screenshot shows the user profile page in the CROSTUDENT application. The top navigation bar includes the "CROSTUDENT" logo and three icons: home, monitor, and user profile. The profile section is divided into two columns. The left column features a profile picture of a man in a blue shirt and tie, with a green "Logout" button below it. The right column is titled "Profil" and contains a table of user information:

Username	admin
First name	John
Last name	Doe
Email	admin@admin.admin

A green "Edit" button is located at the bottom right of the profile information section.

Slika 20: Profil admina

4.6. Administracijsko sučelje

4.6.1. Izmjena podataka

Administrator ima mogućnost unošenja, brisanja i konfiguriranja podataka o domovima. Ako se dogodi pogreška pri unosi ili je potrebno ažurirati neke podatke, admin to napravi na stranici za pojedini studentski dom gdje ime gumb za uređivanje podataka studentskog doma i gumb za uređivanje podataka paviljona.

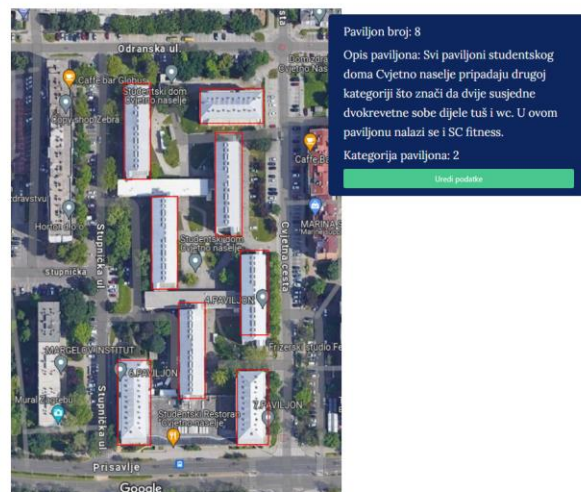


Cvjetno naselje

Odranska 8, 10000 Zagreb

Studentski dom Cvjetno naselje ima 8 paviljona i preko 1800 kreveta. Od popratnog sadržaja ima fitness centar, učionice, liječnicu i stomatološku ordinaciju, polivalentnu dvoranu, dvije menze te cafe slastičarnu.

UREĐI PODATKE



Slika 21: Administratorsko sučelje pojedinog studentskog doma

Administrator ima mogućnost klika na gumb za uređivanje podataka o studentskom domu i na gumbu za uređivanje podataka paviljona.

Na slici broj 19 prikazana su oba modala.

Cvjetno naselje

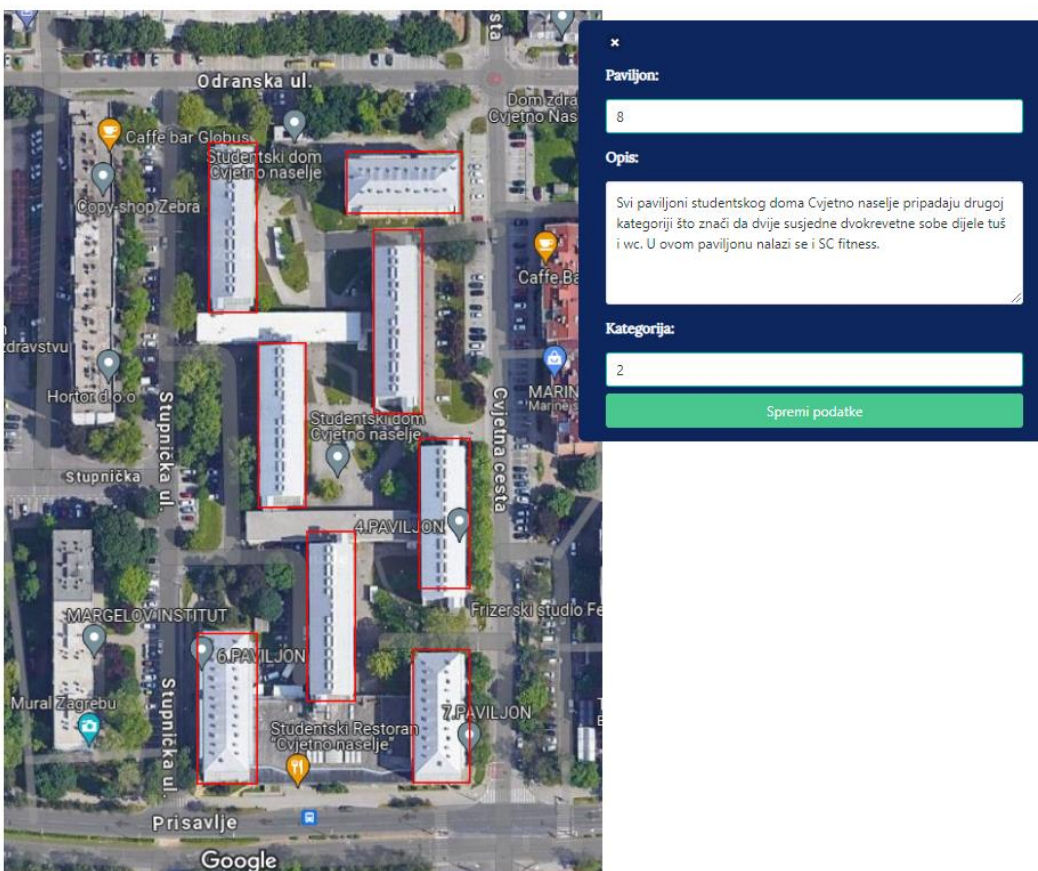
Odranska 8, 10000 Zagreb

Zagreb

Studentski dom Cvjetno naselje ima 8 paviljona i preko 1800 kreveta. Od popratnog sadržaja ima fitness centar, učionice, liječnicu i stomatološku ordinaciju, polivalentnu dvoranu, dvije menze te caffe slastičarnu.

1200

SPREMI PODATKE



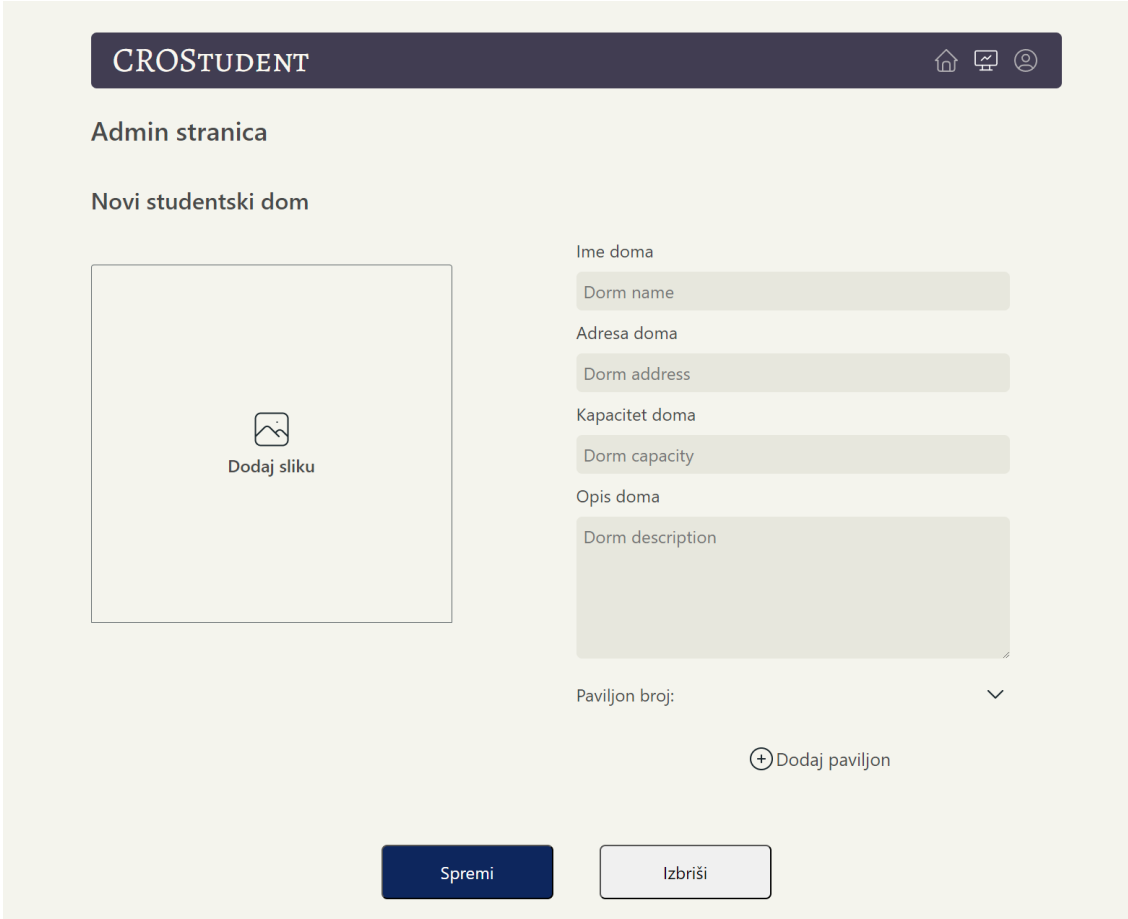
Slika 22: Modali za uređivanje podataka doma i paviljona

Nakon izmjene, podaci se spremaju u bazu podataka i novi podaci su sada vidljivi svim korisnicima. U slučaju da je korisnik istovremeno bio na stranici studentskog doma čiji su

podaci mijenjani, novi podaci bit će mu biti prikazani tek nakon ponovnog učitavanja stranice.

4.6.2. Unos novih podataka

Administrator ima pristup stranici za unos novog studentskog doma u bazu podataka. Kada dođe na tu stranicu, administrator ima opciju da odabere satelitsku snimku studentskog doma i polja za unos informacija o domovima i paviljonima. Dok ne odabere fotografiju, sva polja za unos su onemogućena.




The screenshot shows the 'Admin stranica' (Admin page) for 'Novi studentski dom' (New student dorm) in the CROSTUDENT system. The interface includes a header with the logo and navigation icons. The main content area features a large image placeholder with a camera icon and the text 'Dodaj sliku' (Add image). To the right, there are several input fields: 'Ime doma' (Dorm name), 'Adresa doma' (Dorm address), 'Kapacitet doma' (Dorm capacity), and 'Opis doma' (Dorm description). Below these is a dropdown menu for 'Paviljon broj:' (Pavilion number) with a 'Dodaj paviljon' (Add pavilion) button. At the bottom, there are two buttons: 'Spremi' (Save) and 'Izbrisi' (Delete).

Slika 23: Administracijsko sučelje za unos

U trenutku kada administrator odabere fotografiju, model za detekciju objekata izvršava detekciju paviljon na toj slici te taj proces traje pet do petnaest sekundi. Nakon izvršavanja administrator dobije obrađenu sliku, kao što je prikazano na slici 21. Za potrebe demonstracije u ovom je radu odabran studentski dom “Cvjetno Naselje” u Zagrebu.

Klikom na polje “Dodaj paviljon”, generira se nova forma za paviljon. Tako administrator može dodati koliko paviljona ima i spremati podatke o paviljonima pojedinačno. Kroz CSS su dodane transformacije rotacije i skaliranja elementa slike strelice i slike za spremanje podataka o paviljonu, respektivno.

Broj paviljona 

Slika 24: Prikaz prije rotacija elementa slike strelice

Broj paviljona 

Broj paviljona

Kategorija paviljona

Kategorija paviljona

Opis paviljona

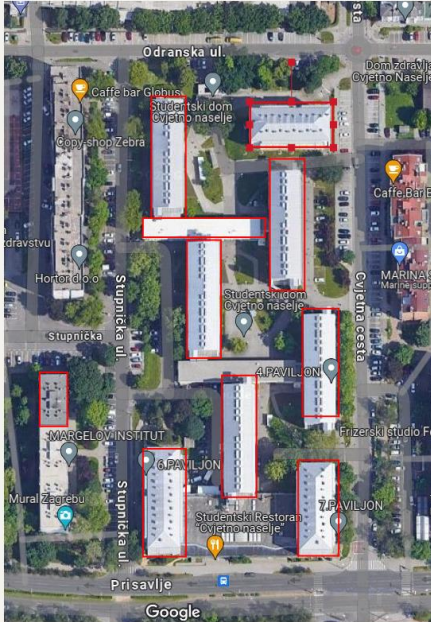
Opis paviljona



Slika 25: Prikaz poslije rotacije elements slike strelice i skaliranja slike za spremanje

Admin stranica

Novi studentski dom



Ime doma

Adresa doma


Kapacitet doma

Opis doma

Broj paviljona 

Kategorija paviljona

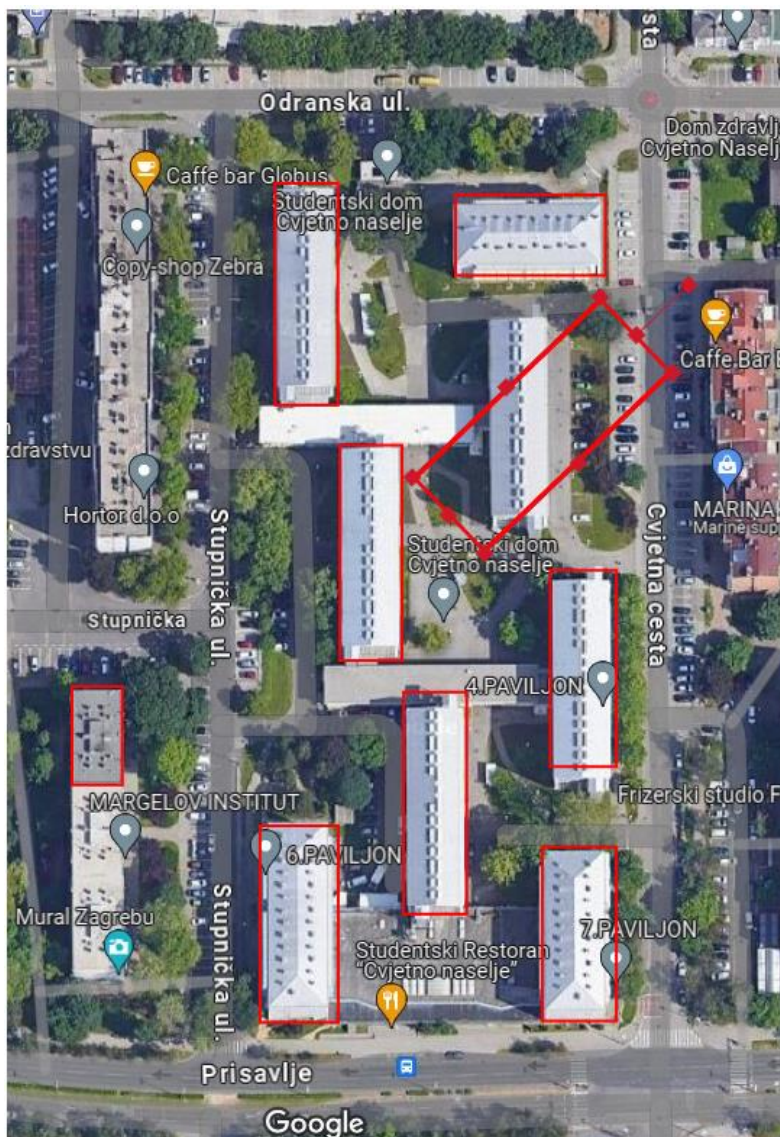
Opis paviljona

Broj paviljona  Dodaj paviljon

Slika 26: Izlaz generiran modelom za detekciju

S obzirom na to da model za detekciju ima prostora za usavršavanje, administrator sustava ima opciju brisanja, rotiranja, mijenjanja veličine i stvaranja novih pravokutnika. Opcija brisanja ostvaruje se odabirom pravokutnika zatim klikom na gumb "Izbriši".

Ostatak navedenih opcija može se vidjeti na slici 24.



Slika 27: Prikaz mogućnosti na kanvasu

Nakon što admin spremi podatke pojedinog studentskog doma u bazu podataka, taj je studentski dom vidljiv i drugim korisnicima. Podaci o paviljonima u bazi podataka spremljeni su kao lista objekata. Primjer jednog objekta iz takve liste dan je u nastavku:

```
"x":0.16288157413503518,
"y":0.19835766650796952,
"name":"7456",
"width":0.6806853582554517,
"height":0.6992753623188406,
"scaleX":0.9748962025797667,
"scaleY":0.6887220918438512,
"stroke":"red",
"rotation":9.034168781895682,
"draggable":true,
"paviljoni":{
  "number":"1",
  "kategorija":"2",
  "description":"Sobe u domu u Dubrovniku imaju vlastitu kupaonicu i čajnu kuhinju na svakom katu. Od ostalih sadržaja ima teretanu, praonice, zajedničke prostorije za učenje i druženje, restoran i podzemnu garažu."
},
  "strokeWidth":2
}
```

U objektu atributi “x” i “y” predstavljaju normaliziranu gornju lijevu koordinatu pravokutnika. Te koordinate množe se sa širinom i visinom prikazane slike na stranici pojedinog studentskog doma. Atribut “name” predstavlja jedinstveno ime pravokutnika. Normalizirana širina i visina pravokutnika predstavljene su atributima “width” i “height”.

Funkcija za normalizaciju prikazana je u nastavku:

```
function normaliseBoxes() {
  for (let i = 0; i < boxes.value.length; i++) {
    boxes.value[i].x = boxes.value[i].x / stageImage.value.image.width;
    boxes.value[i].y = boxes.value[i].y / stageImage.value.image.height;
    boxes.value[i].width = boxes.value[i].width /
stageImage.value.image.width;
    boxes.value[i].height =
      boxes.value[i].height / stageImage.value.image.height;
    boxes.value[i].rotation = boxes.value[i].rotation;
  }
  saveIntoDatabase(boxes.value);
}
```

Funkcija prolazi kroz petlju po svim vrijednostima koordinata x, y, širine i visine svih pravokutnika i te vrijednosti dijeli s trenutnom širinom slike. Nakon što petlja prođe kroz sve pravokutnike, poziva se funkcija koja sprema te vrijednosti u bazu podataka.

Atribut “stroke” i “strokeWidth” definiraju boju i veličinu granice pravokutnika. Rotacija je bitan atribut koji označava je li pravokutnik rotiran i, ako je, za koliko stupnjeva. Svi pravokutnici pri kreiranju doma imaju atribut “draggable” što omogućava adminu da može namjestiti pravokutnik, ako je potrebno, na pravu lokaciju. Isti atribut nije dostupan običnom korisniku.

5. Zaključak

Trend porasta broja stanovnika koji se odlučuju za studiranje u Republici Hrvatskoj proteklih godina vodi do sve veće potrebe za sustavom koji će proces odabira studentskog doma učiniti brzim i jednostavnim. Iz tog je razloga izrađena web aplikacija “CroStudent” koja za glavni cilj ima sumiranje svih relevantnih podataka o studentskim domovima u Republici Hrvatskoj na jednom mjestu.

Aplikacija sadrži vizualni prikaz svakog studentskog doma unutar Republike Hrvatske s opisom, adresom, kapacitetom i detaljnim informacijama o svakom paviljonu, uključujući kategoriju, broj i opis. Sama aplikacija koristi prepoznavanje objekta (engl. *Object recognition*) pri unosu slika te nudi opciju prijave kroz tri različite uloge: kao administrator sustava, kao anonimni korisnik i kao korisnik koji je prijavljen na portal. Prilikom prijave na portal koristi se jedinstveno korisničko ime i lozinka koje aplikacija može zapamtiti ili ne, ovisno o preferencijama korisnika. Administrator ima zaseban profil i mogućnosti unosa novih podataka, izmjene postojećih ili brisanje podataka koji više nisu relevantni. Osim izmjene podataka o određenom studentskom domu, administrator ima mogućnost izmjene i cijelih domova (dodavanje novih ili brisanje postojećih iz sustava).

Aplikacija CroStudent napravljena je tako da bude maksimalno prilagodljiva i jednostavna za korištenje te pruža velik potencijal za rast i druge primjene.

Literatura

1. Sivasankaran, S. (2022). A Basic Introduction to Object Detection [Online]. Available: <https://www.analyticsvidhya.com/blog/2022/03/a-basic-introduction-to-object-detection/>
2. Technoarch Software. (2020). Django Request-Response Cycle [Online]. Available: <https://www.technoarchsoftwares.com/blog/django-request-response-cycle/>
3. Amazon Web Services. (n.d.). Three-tier architecture overview [Online]. Available: <https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/three-tier-architecture-overview.html>
4. Educative. Shubham Singh Kshatriya (n.d.). What is MVT structure in Django? [Online]. Available: <https://www.educative.io/answers/what-is-mvt-structure-in-django>
5. TensorFlow. (n.d.). TensorFlow [Online]. Available: <https://www.tensorflow.org/>
6. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions [Online]. Available: <https://arxiv.org/abs/1506.01497v3>
7. Davis, Q. (2018, May 18). Between the Wires: An Interview with Vue.js Creator Evan You [Online]. Available: <https://medium.com/free-code-camp/between-the-wires-an-interview-with-vue-js-creator-evan-you-e383cbf57cc4>
8. Vue.js. (n.d.). Vue.js [Online]. Available: <https://vuejs.org/>
9. SimilarTech. (n.d.). JavaScript Technologies. SimilarTech [Online]. Available: <https://www.similartech.com/categories/javascript>
10. BlueBird International. (n.d.). Frontend Frameworks [Online]. Available: <https://bluebirdinternational.com/frontend-frameworks/>
11. Rai-Shahnawaz, A. (n.d.). Creating a Django Project the Right Way [Online]. Available: <https://rai-shahnawaz.medium.com/creating-a-django-project-the-right-way-14d230358d72>
12. Swagger. (n.d.). Swagger UI. Swagger [Online]. Available: <https://swagger.io/tools/swagger-ui/>

13. PostgreSQL. (n.d.). PostgreSQL. Wikipedia [Online]. Available:
<https://en.wikipedia.org/wiki/PostgreSQL>
14. Konva. (n.d.). Konva Overview [Online]. Available:
<https://konvajs.org/docs/overview.html>

Web application for student housing based on advanced two-dimensional maps

ABSTRACT

The specificities and differences of student dormitories in Croatia make finding comprehensive information about dormitories in different cities complicated. Considering the increasing popularity and need for accommodation in student dormitories, a web application has been developed to provide all necessary information about dormitories within Croatia.

The developed application includes a visual representation of each student dormitory within Croatia with descriptions, addresses, capacities, and detailed information about each pavilion, including category, number, and description, utilizing object detection when uploading images. The application itself consists of a database, server, and user application. The following techniques were used in the development of the application: TensorFlow, Vue3, Django, PostgreSQL, and Konva.js. The application can be accessed from three different user roles - as an administrator with the ability to modify data, as a user who can log into the system with unique user data and password or as an anonymous user.

Given that the goal of the application is to expedite and simplify the process of searching and selecting student dormitories in Croatia, and considering that machine learning used in the development process has almost unlimited possibilities if it contains a sufficiently large training dataset, this application has the potential for broader application beyond the territory of Croatia.

KEYWORDS: Web application; machine learning; student dorms; object detection

Web aplikacija za studentske domove temeljena na naprednim dvodimenzionalnim kartama

SAŽETAK

Specifičnosti i različitosti studentskih domova u Republici Hrvatskoj čine pronalazak sveobuhvatnih informacija o studentskim domovima u različitim gradovima kompliciranim, a s obzirom na sve veću popularnost i potrebu za smještajem u studentskim domovima, napravljena je web aplikacija koja pruža sve potrebne informacije o domovima unutar Republike Hrvatske.

Razvijena aplikacija sadrži vizualni prikaz svakog studentskog doma unutar Republike Hrvatske s opisom, adresom, kapacitetom i detaljnim informacijama o svakom paviljonu, uključujući kategoriju, broj i opis te koristi prepoznavanje objekta (engl. *Object detection*) pri unosu slika. Sama aplikacija se sastoji od baze podataka, poslužitelja i korisničke aplikacije. Pri izradi aplikacije korištene su sljedeće tehnike: TensorFlow, Vue3, Django, PostgreSQL i Konva.js. Aplikaciji se može pristupiti s dva gledišta - kao administrator koji ima mogućnost izmjene podataka ili kao korisnik koji se u sustav može prijaviti s jedinstvenim korisničkim podacima i lozinkom, ali može i ostati anonimn.

S obzirom na to da je cilj aplikacije ubrzati i pojednostaviti proces pretraživanja i odabira studentskih domova u Republici Hrvatskoj, a imajući na umu da je korišteno strojno učenje s gotovo neograničenim mogućnostima ako sadrži dovoljno veliki skup podataka za treniranje, ova aplikacija ima potencijal primjene širi od područja Republike Hrvatske.

KLJUČNE RIJEČI: Web aplikacija; strojno učenje; studentski domovi; detekcija objekata