

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

**Antonio Labinjan**

COMPUTER VISION ATTENDANCE SYSTEM

Dokumentacija

Pula, 2024.

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

## COMPUTER VISION ATTENDANCE SYSTEM

Dokumentacija

**Antonio Labinjan**

JMBAG: 0303106891 redovan student

Kolegij: **Web-aplikacije**

Mentor: **doc. dr. sc. Nikola Tanković**

Pula, prosinac 2024.

Tu ću hitit sadržaj kad ga buden napisa

## 1. Sažetak

Computer vision attendance system je aplikacija za evidentiranje prisutnosti pomoću prepoznavanja lica.

Ova aplikacija koristi computer vision za pojednostavljenje evidencije prisutnosti u obrazovnim institucijama. Studenti mogu bilježiti svoju prisutnost putem prepoznavanja lica u unaprijed definiranim intervalima, čime se uklanja potreba za ručnim ili papirnatim metodama. Profesori imaju intuitivno sučelje za upravljanje učenicima, predmetima i rasporedima te generiranje detaljnih izvještaja i statistika o prisutnosti kako bi pratili angažiranost učenika. Sustav osigurava točnost, smanjuje administrativni teret i pruža uvid u trendove prisutnosti. Razvijen s naglaskom na jednostavnost i učinkovitost, ovaj sustav integrira suvremenu tehnologiju kako bi unaprijedio iskustvo obrazovanja za studente i profesore.

Postoje 2 tipa korisnika; običan student koji unutar aplikacije može samo prijaviti prisutnost skeniranjem lica i profesor koji ima dodatne ovlasti koje će biti detaljnije razrađene u idućim poglavljima.

Omogućena je registracija profesora u sustav pomoću emaila i passworda. Nakon registracije, profesor može postaviti predmet koji se trenutno izvodi sa predviđenim datumom i vremenskim intervalom. Zatim, implementirane su funkcionalnosti dodavanja novih studenata u sustav pomoću ručnog uploada slika, ili pomoću izvlačenja frameova iz live snimke, ovisno o potrebi. Nadalje, omogućeno je automatizirano slanje obavijesti putem maila za svaku uspješnu prijavu prisustva korištenjem SendGrid API-ja.

Kod svake prijave prisustva također se provjerava se je li pred kamerom prava osoba ili slika pomoću jednostavne anti-spoofing mehanike. (postoje i neki napredniji API-ji za liveness check, ali su poprilično skupi).

Nakon postavljanja predmeta i dodavanja studenata, moguće je prijaviti prisutnost tako što aplikacija najprije provjeri koje je trenutno vrijeme i pripada li ono nekom od intervala za postavljene predmete. Zatim, kada student stane pred kameru, kreće složeni proces prepoznavanja koristeći OPENAI-jev CLIP model i Facebook-ov AI Similarity Search.

Što se tiče implementacije same aplikacije, korišten je Pythonov web-development framework Flask za backend, HTML za frontend te Sqlite za bazu podataka.

## 2. Uvod i motivacija

Ova aplikacija nastala je zbog želje za povezivanjem više različitih područja poput baza podataka, web-developmenta, računalnog vida i umjetne inteligencije u kompletan "proizvod" i zbog toga što me navedena područja izrazito zanimaju i definitivno ću se njima (bar se nadam) baviti u budućnosti. Odabran je Python kao jezik izrade aplikacije zato što se iz mog dosadašnjeg iskustva čini kao najbolji izbor za bilo kakve zadatke vezane uz AI. Također, posebno je motivirajuća činjenica da je ova aplikacija prilično inovativna. (Naravno, nije prva aplikacija koja implementira computer vision za prepoznavanje ljudi i evidenciju prisutnosti, ali je prva koja koristi kombinaciju CLIP-a i FAISS-a).

Nadalje, ono što je zanimljivo, su svakako prilično dobri rezultati koje ona postiže. Na iznimno dobrim referentnim slikama, postiže se točnost od oko 100%. Korištenjem nešto manje kvalitetnih slika, rezultati opadaju (~73%), no možemo tvrditi da je aplikacija pri primarnom use-caseu gotovo savršena. Ako bismo uzeli 30-ak ljudi (ekvivalent jednom školskom razredu) i par njihovih slika za feed-anje u model, aplikacija bi ih identificirala u 100% slučajeva.

Naravno, nije sve bilo savršeno iz prvog pokušaja i bilo je potrebno puno eksperimentiranja, istraživanja, pokušaja i pogrešaka, te u trenutku pisanja dokumentacije još uvijek tražimo idealnu kombinaciju parametara koja će dati maksimalne rezultate i za lošiji dataset slika.

### 2.1. SWOT analiza

#### Strengths:

- **Inovativna tehnologija:** Korištenje computer visiona i modela poput CLIP-a donosi modernu i naprednu tehnologiju u obrazovni sektor i ostale ustanove gdje je evidencija prisutnosti od ključne važnosti
- **Automatizacija procesa:** Eliminira ručno vođenje evidencije o prisutnosti, čime štedi vrijeme nastavnicima i administraciji
- **Preciznost:** Modeli mogu prepoznati učenike čak i u velikim grupama, uz minimalne greške
- **Jednostavna integracija:** Može se lako prilagoditi različitim školskim sustavima
- **Smanjenje varanja:** Prepoznavanje lica smanjuje mogućnost lažnog prijavljivanja prisutnosti

#### Weaknesses

- **Ovisnost o kvaliteti dataseta:** Loši podaci smanjuju preciznost modela, što može dovesti do misklasifikacija i frustrirati korisnike
- **Privatnost i sigurnost:** Mogući problemi s GDPR-om i zaštitom osobnih podataka učenika, usprkos činjenici da se podaci strogo čuvaju i teoretski ne bi trebali izlaziti izvan granica sustava
- **Infrastrukturni zahtjevi:** Potreba za kamerama visoke kvalitete i stabilnom mrežom u ustanovama što u nekim slučajevima može predstavljati velik problem
- **Ograničenja u nepovoljnim uvjetima:** Loše osvjetljenje, pokretni subjekti i slabe kamere mogu utjecati na točnost

- **Troškovi:** Ustanove s ograničenim budžetom mogu imati poteškoća s implementacijom sustava

## Opportunities

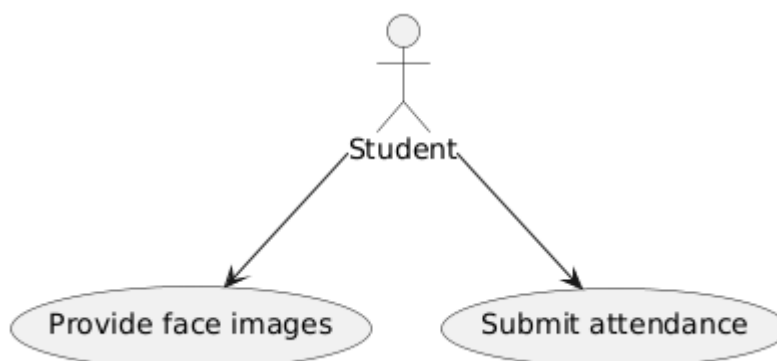
- **Širenje na druga tržišta:** Fakulteti, korporacije (praćenje zaposlenika), industrijske primjene (sigurnost i autorizacija)
- **Povećanje funkcionalnosti:** Integracija s dnevnicima, sustavima ocjenjivanja ili aplikacijama za roditelje
- **Prilagodba za druge namjene:** Npr. sigurnosne provjere, kontrola ulaza u objekte ili praćenje emocionalnog stanja učenika
- **Suradnja s vladinim institucijama:** Kao dio digitalizacije obrazovnog sustava
- **Razvoj freemium modela:** Osnovne funkcije besplatne, dok napredne zahtijevaju pretplatu

## Threats

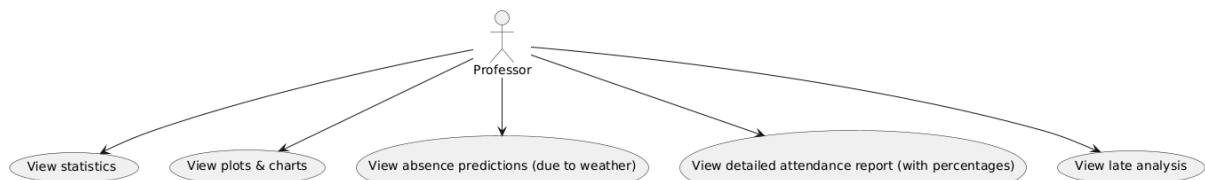
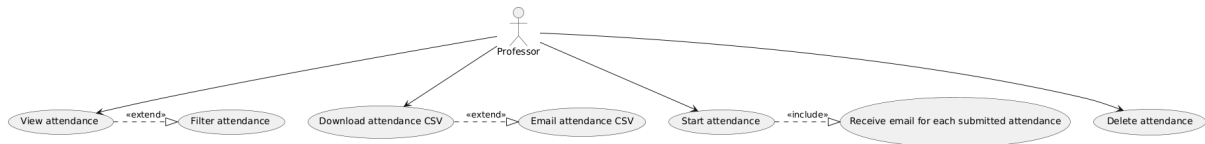
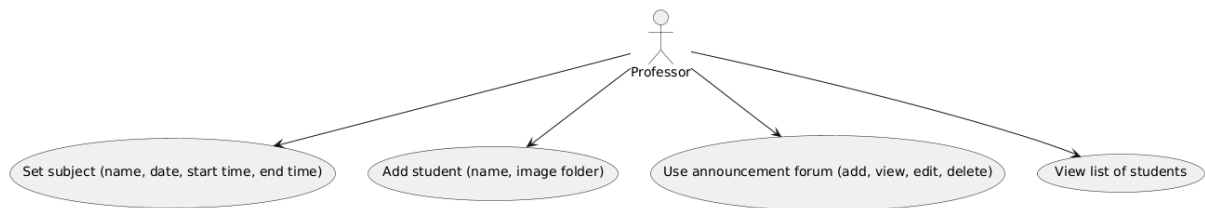
- **Regulacije i zakoni:** Stroge regulative o privatnosti podataka mogu ograničiti implementaciju
- **Ovisnost o tehnologiji:** Tehnički kvarovi, zastarijevanje modela ili sigurnosne rupe mogu imati loš utjecaj na prihvaćanje aplikacije
- **Otpor korisnika:** Strah ili skepticizam prema tehnologiji prepoznavanja lica, posebno kod roditelja
- **Troškovi održavanja:** Aplikacija zahtijeva stalna ažuriranja i održavanje kako bi bila konkurentna

## 3. Use Case dijagram sustava

Sustav ima 2 tipa korisnika: studenta koji nema praktički nikakve ovlasti unutar aplikacije osim prijavljivanja prisutnosti i profesora koji ima pune administratorske ovlasti, što je detaljnije razrađeno na sljedećim dijagramima (iako su i student i profesor dio istog sustava, razdvojeni su u 2 dijagrama radi preglednosti te je dodatno dijagram za profesora razdvojen u 3 dijela).



Use case za studenta - prilaže slike za evidenciju u sustav i prijavljuje prisutnost



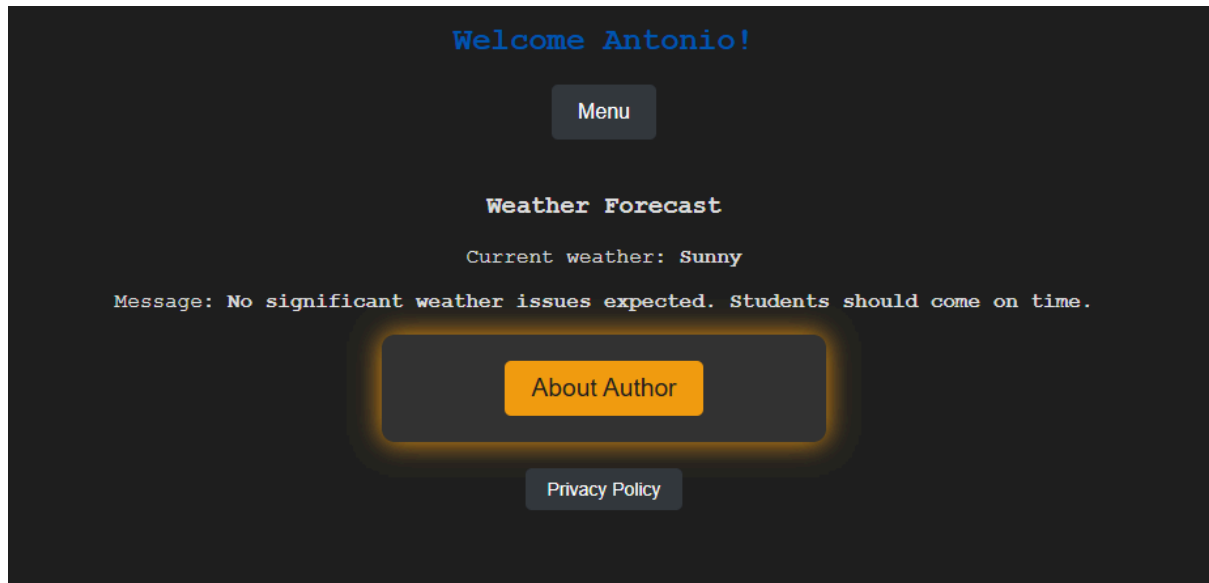
Use case za profesora - nakon prijave u sustav može postaviti trenutni predmet, njegov start time i end time, dodavati nove studente u sustav, uređivati objave na forumu, pregledavati popis svih studenata, pregledavati podatke o prisustvu s opcijom brisanja i filtriranja, preuzeti izvješće o prisustvima u csv formatu i dijeliti ga e-mailom, započeti interval bilježenja prisutnosti i zaprimiti email s potvrdom svake prijave i po potrebi obrisati neki od zapisa o prisutnosti. Može i pregledavati razne statističke analize, predikcije, izvještaje i grafove vezane uz podatke o prisutnostima.

BITNO=> NAKNADNO UBACIT CLASS DIJAGRAM

#### 4. Razrada funkcionalnosti

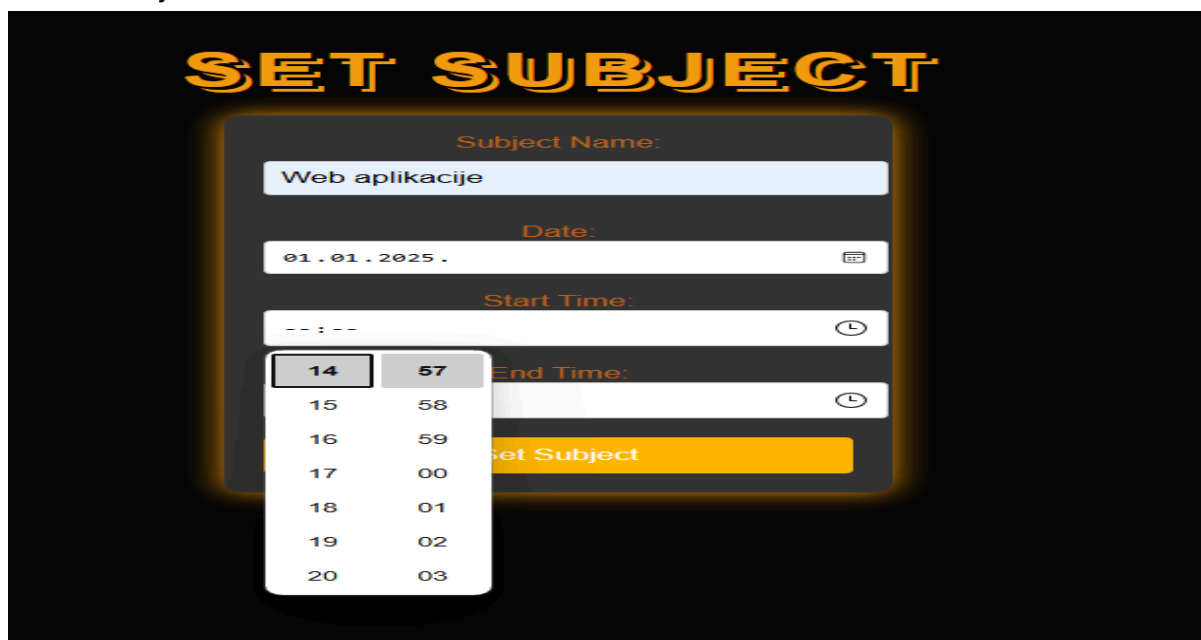
Aplikacija se sastoji od 2 dijela - frontenda napisanog u HTML-u i backenda napisanog u Flasku uz dodatak SQL baze podataka te CLIP modela i FAISS-a. U nastavku će biti detaljnije opisani svi dijelovi aplikacije uz napomenu da sposobnosti dizajna definitivno nisu nešto čime se mogu pohvaliti .

#### 4.1. Landing page



Nakon što se korisnik uspješno prijavi u sustav, dočekat će ga landing page koji sadrži welcome poruku s dinamički dohvaćenim korisničkim imenom, tipkom za dropdown menu koja se aktivira pri hoveru miša, tipkom za rutiranje na about stranicu i tipkom za pregled privacy policy-ja koji objašnjava kako referentne slike korištene za prepoznavanje unutar aplikacije ni u kojem slučaju neće izaći izvan sustava. Također, prikazuje se i trenutna vremenska prognoza za Pulu dohvaćena putem vanjskog API-ja.

#### 4.2. Set subject

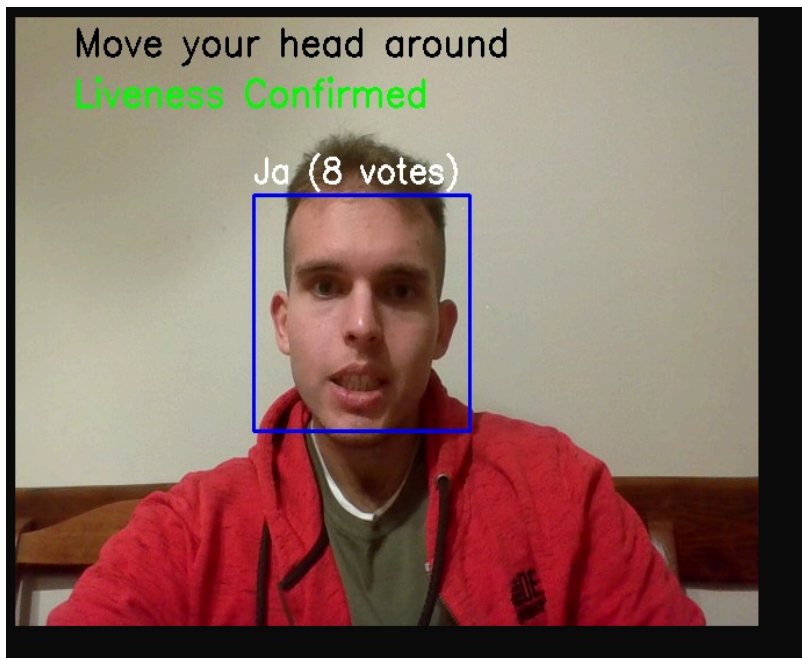


Komponenta za postavljanje trenutnog predmeta je točka od koje kreće svaka evidencija prisutnosti. U tom dijelu aplikacije profesor definira za koji se predmet trenutno bilježi prisutnost. Nadalje, definira se datum evidentiranja prisutnosti koji se automatski postavlja na današnji datum, no moguće ga je promijeniti ukoliko je to potrebno. Na kraju, još je



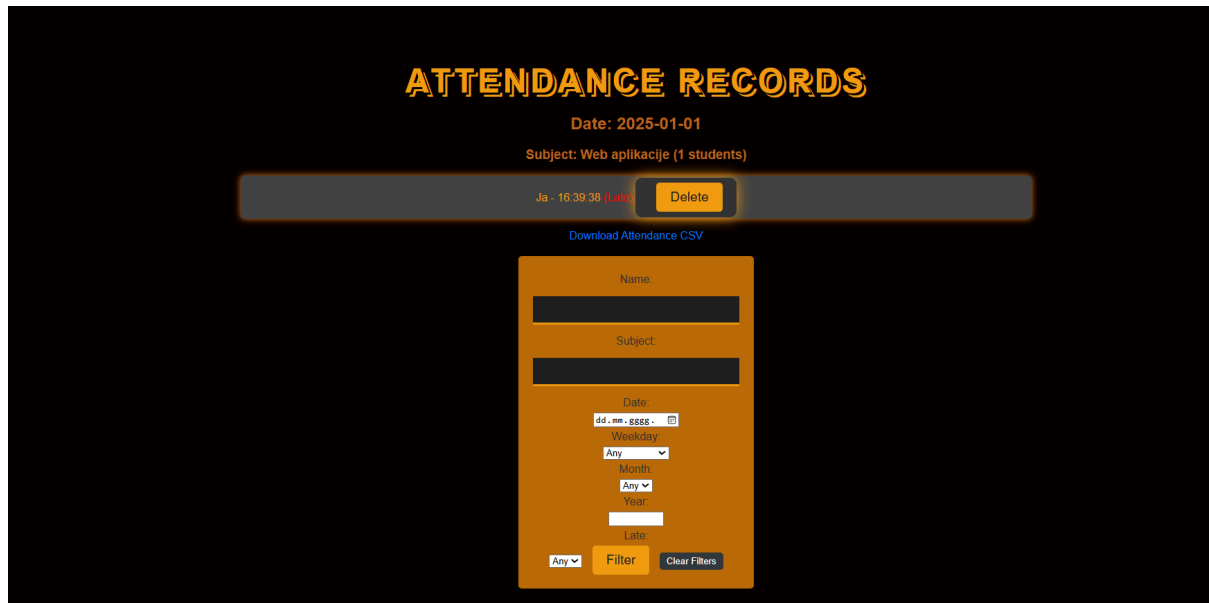
potrebno definirati start time i end time za evidenciju kako bi se prijava prisutnosti ograničila na definirani vremenski interval.

#### 4.3. Attendance logging



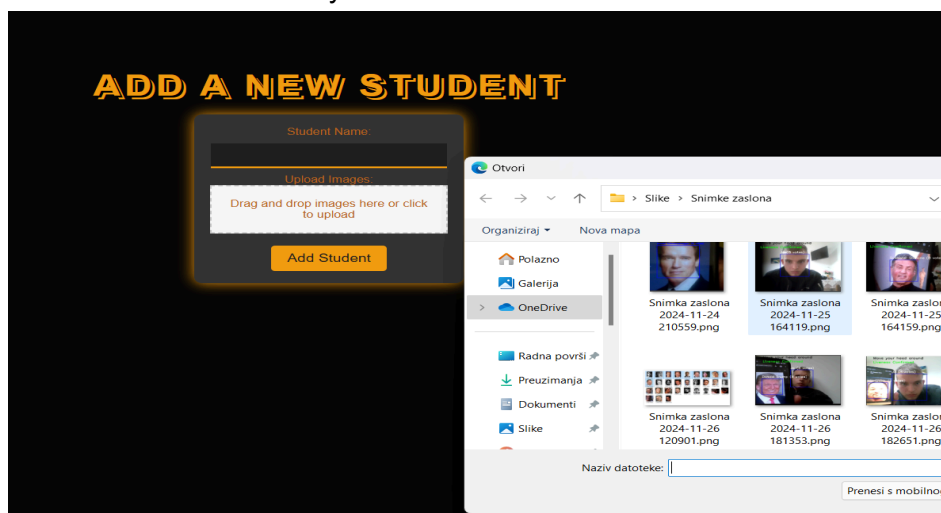
Najsloženiji dio aplikacije svakako se odnosi na modul za live prijavu prisustva. Pomoću cv2 biblioteke, otvara se kamera koja najprije traži lica u dohvaćenim frameovima koristeći haarcascade. Za svako prepoznato lice, kreira se bounding box. Nakon toga, provodi se jednostavna liveness detekcija koja provjerava kreće li se lice i registrira treptaje (kao što je već spomenuto, postoje bolja rješenja za liveness provjeru, ali naplaćuju se). Ukoliko lice prođe liveness check, kreće proces konverzije slike u embedding koristeći clip. Ono što se događa u pozadini je zapravo pretvorba featuresa slike u tensor. Nakon što se tensor dodatno normalizira, kreće usporedba dobivenog tensora sa tensorima svih poznatih lica koji su pohranjeni u faiss indexu. Faiss index omogućava indeksiranu pretragu na temelju sličnosti. Takva je potraga implementirana zato što su lica najprije bila pohranjivana u običan embedding space, no to se nije pokazalo funkcionalnim za veći broj poznatih lica jer je došlo do preklapanja embeddinga i misklasifikacije. Faiss je omogućio dohvat top-k rezultata, odnosno uzimanje input tensora sa kamere i dohvat k slika koje najviše sliče inputu. Nakon dohvata k slika, provodi se majority voting koji provjerava s kojom se klasom input najviše podudara. Radi dodatne optimizacije, koriste se k1 i k2 koji omogućuju da se ne pretražuju sva lica, već se na samom početku iz razmatranja izbace ona lica koja su previše različita. Također, postoji i parametar thresholda koji definira koliko % input mora sličiti nekom od lica kako bi se oni smatrali validnim parom.

#### 4.4. Attendance overview



Nakon uspješne evidencije prisutnosti, u bazu podataka se pohranjuje zapis o prisutnosti koji se sastoji od trenutnog predmeta za koji bilježimo prisutnost, vremena evidentiranja, datuma, imena i prezimena studenta i late flaga. Late flag označava kasni li student ili je došao na vrijeme. Taj se dio evidentira provjeravajući koje je definirano početno vrijeme za postavljeni predmet i koje je vrijeme evidentiranja prisutnosti. Ukoliko je prisutnost evidentirana 15 ili više minuta kasnije od početnog vremena, taj se zapis definira kao kašnjenje. (Postoji akademska četvrt). Ime i prezime se dohvaća preko naziva klase u koju je skenirano lice svrstano. Dodatno, postoji mogućnost brisanja zapisa i filtriranje po imenu studenta, predmetu, datumu, danu u tjednu, mjesecu, godini i late flagu. Na samom kraju, radi lakše obrade podataka omogućeno je preuzimanje zapisa o prisutnosti u csv formatu.

#### 4.5. Add student manually



Moguće je dodati novog studenta u bazu poznatih lica ručno, koristeći slike iz lokalne pohrane. Potrebno je unijeti ime i odabrati slike iz računala ili ih drag&droppati na predviđeno mjesto. Minimalni unos je 1 slika, dok gornje granice nema. No, kako takav pristup nije naročito praktičan, omogućeno je i live dodavanje pomoću kamere.

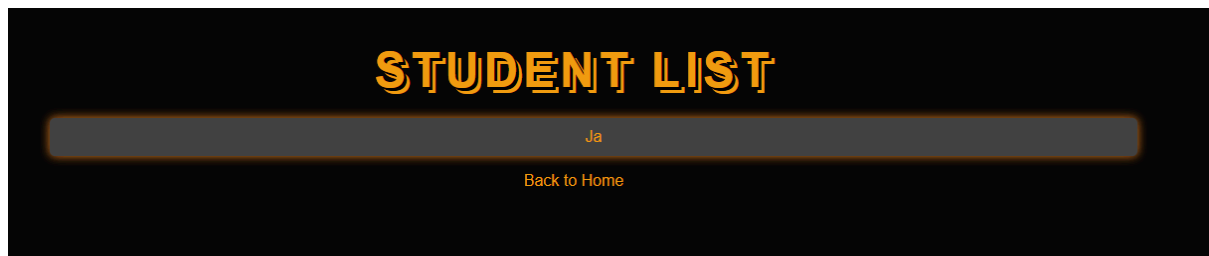
#### 4.6. Add student using live feed

## Add New Student via Live Feed

Student Name:

Dodavanje pomoću kamere funkcionira na prilično jednostavan način. Potrebno je unijeti ime i prezime studenta i samo pritisnuti na tipku. Tada student treba gledati u kameru nekoliko sekundi, kako bi se ulovilo 25 frameova i pohranilo ih se u novi folder s njegovim imenom. Ova mogućnost olakšava prikupljanje slika za proširivanje baze poznatih lica kako ne bi bilo potrebno za svakog studenta uploadati slike ručno i značajno ubrzava proces.

#### 4.7. Students overview



Pri prvotnoj implementaciji spremanja prisutnosti došlo je do malih poteškoća u provjeri poznatih studenata, pa je tako nastala ova, inicijalno, debug ruta koja dohvaća sve studente iz svih zapisa o prisutnosti, neovisno o predmetu i datumu. Provodi se sql upit koji dohvaća sve unique studente koji su ikada spremljeni u bazu, odnosno evidentirani kao prisutni.