Poročilo – Sprint 4
Projekt: Sistem za samodejno evidentiranje prisotnosti z uporabo prepoznave obrazov
Ekipa: Viktor Rackov, Mario Bojarovski, Marko Milenovic
Iteracija: Sprint 4

Opravljeno delo:

V tej iteraciji smo nadaljevali z razvojem glavnega backend sistema, ki je zdaj zaključen in stabilen. Dodana je bila podpora za ustvarjanje dogodkov neposredno iz uporabniškega vmesnika – ob zagonu kamere lahko uporabnik ustvari nov dogodek, kamor se nato beležijo vsi zaznani obrazi.

Poleg prepoznave obrazov smo v sistem vključili še zaznavanje čustev. Tako se zdaj ob zaznavi osebe v prisotnosti poleg imena izpiše tudi trenutno čustveno stanje. Vsaka zaznana oseba in njeno čustvo se beležita znotraj dogodka, kar omogoča bogatejšo analizo prisotnosti.

Težave in rešitve:

Glavni izziv v tem sprintu je bil usklajevanje dveh ločenih AI modelov – za prepoznavo obrazov in za zaznavanje čustev. Zaznavanje čustev je bistveno bolj časovno zahtevno in ni omogočalo obdelave v realnem času skupaj z obrazno prepoznavo. Zato smo uvedli ločen način obdelave, kjer se vsak model izvaja samostojno in asinhrono, kar je zagotovilo stabilnost in odzivnost sistema.

Poleg tehničnih izzivov je bila ena izmed večjih ovir tudi časovna omejenost. Zaradi zgoščenega urnika v semestru in dodatnih obveznosti (službe in druge študijske zadolžitve) nekaterih članov ekipe, je bilo razporejanje časa za skupno delo zahtevno. Kljub temu smo uspeli zaključiti ključne funkcionalnosti in zagotoviti delujočo celoto.

Kompleksnost sistema:

Sistem je v tej fazi postal modularno razdeljen in obsežen. Arhitektura zdaj vključuje:

- **en frontend** (React aplikacija),
- **glavni backend** (Express + MongoDB),
- **ločen backend za upravljanje zagona in zaustavitve kamere** (Flask strežnik),
- **aplikacijo za upravljanje kamere** (Python skripta),
- **dve ločeni AI komponenti**:
    - za prepoznavo obrazov
    - za zaznavanje čustev
      (obe tečeta v ločenih Docker okoljih)

Ta modularna zasnova zahteva natančno usklajevanje med komponentami in robustno komunikacijo prek API-jev. Upoštevati je treba različne hitrosti obdelave in potencialne napake, zato smo morali posebno pozornost nameniti stabilnosti in upravljanju napak v komunikaciji med moduli.