Univerzitet u Beogradu

Elektrotehnički fakultet



Django – online video pozivi

|  |  |  |
| --- | --- | --- |
| Mentor: |  | Student: |
| Dr Ana Gavrovska,  Docent |  | Uroš Stojanović  2019/0404 |

Beograd, Jun 2023.

Sadržaj

[Uvod 3](#_Toc139035633)

[Django 4](#_Toc139035634)

[Postavka i izrada projekta 5](#_Toc139035635)

[Funkcionalnosti 5](#_Toc139035636)

[Instalacija i podešavanja 8](#_Toc139035637)

[Testiranje zakasnjenja 15](#_Toc139035638)

[Zaključak 16](#_Toc139035639)

[Literatura 17](#_Toc139035640)

# Uvod

U današnjem digitalnom dobu, razvoj veb aplikacija postao je integralni deo posla inženjera. Izgradnja dinamičnih, interaktivnih i robustnih veb sajtova i aplikacija zahteva moćne alate. Django, jedan od mnogobrojnih veb okvira za Python, danas je popularan izbor među programerima zbog svoje jednostavnosti, skalabilnosti i raznovrsnosti. Ovaj rad se bavi njegovom primenom u predelu video sistema i video striminga.

Django je prvobitno razvijen 2003. godine i pušten kao open-source softver 2005. Njegov osnovni cilj je pojednostavljenje kreiranja kompleksnih veb sajtova zasnovanih na bazama podataka. Django sledi arhitektonski obrazac Model-View-Controller (MVC), odn. Model-View-Template (MVT) u Djangu, koji razdvaja modele podataka, poslovnu logiku i slojeve prezentacije radi efikasne organizacije koda. Svojom slojevitošću podseća na Objektno orijentisano programiranje.

Neke od ključnih funkcionalnosti Django-a:

* Mapiranje objekata i relaciono modeliranje (ORM)
* Rutiranje URL-ova i funkcije prikaza
* Sistem šablona
* Autentikacija i autorizacija
* Interfejs administracije

Benefiti Django programiranja su mnogobrojni ali najvažniji su skalabilnost, raznovrsnost i brzina razvoja, koji svi proizniču iz njegovog Open-Source modela.

# Django

Django je složeni web okvir Open-code tipa baziran na Pythonu koji omogućuje brzu izradu skalabilnih i sigurnih web aplikacija. Sa svojim širokim asortimanom ugrađenih funkcionalnosti kao što su ORM za jednostavno upravljanje bazama podataka, autentikacija i autorizacija korisnika, podrška za globalizaciju i lokalizaciju, Django olakšava razvoj kompleksnih projekata. Osim toga, Django poseduje jasnu i dobro dokumentovanu strukturu projekta kojom promoviše princip DRY (Don't Repeat Yourself), čime omogućuje čist i održiv kod. Aktivna zajednica programera, veliki broj dostupnih paketa i resursa za učenje čine Django popularnim izborom među programerima širom sveta. Takođe, Django podržava integraciju sa drugim tehnologijama i ima bogat ekosistem koji olakšava izgradnju i proširivanje veb aplikacija. Nudi bogat set ugrađenih alata i funkcionalnosti za upravljanje bazama podataka, sigurnost i obradu HTTP zahteva, što ga čini popularnim izborom među programerima širom sveta.

Najvažnije karakteristike i prednosti Django programskog jezika su:

* Brza izrada veb aplikacija: Django je složeni programski jezik koji omogućuje brzu i efikasnu izradu veb aplikacija. Sadrži mnoge ugrađene funkcionalnosti i alate koji olakšavaju razvoj, što rezultuje u smanjenju vremena i truda potrebnih za izgradnju kompleksnih veb aplikacija.
* Robustnost i sigurnost: Django je osmišljen kako bi pružio sigurno i pouzdano okruženje za izradu web aplikacija. Dolazi sa ugrađenim sigurnosnim mehanizmima koji pomažu u zaštiti aplikacije od uobičajenih ranjivosti, poput CSRF napada, SQL injekcija i XSS napada. Takođe podržava autentikaciju, autorizaciju i upravljanje sesijama kako bi se osigurala sigurnost korisničkih podataka.
* Skalabilnost i fleksibilnost: Django omogućuje izgradnju skalabilnih web aplikacija koje mogu rasti s rastom prometa i zahteva. Nudi fleksibilnost u oblikovanju arhitekture aplikacije i podržava modularnost putem ponovne upotrebe aplikacijskih komponenti. To olakšava održavanje i proširivanje aplikacije kako bi se prilagodila promenjivim poslovnim potrebama.
* Obeležavanje ORM (Objektno-relacijsko mapiranje): Django uključuje moćan ORM sistem koji omogućuje rad s bazama podataka na visokom nivou apstrakcije. To omogućuje developerima da manipulišu podacima koristeći Python objekte umesto SQL upita, što olakšava rad s bazama podataka i pomaže u održavanju čistog koda.
* Razvijena zajednica i dokumentacija: Django ima aktivnu zajednicu programera. Postoji bogata dokumentacija, forumi, blogovi i resursi koji su dostupni za učenje, deljenje znanja i rešavanje problema. Ova podrška omogućuje brže rešavanje problema i pruža dodatnu podršku tokom razvoja novih projekata.
* Prijateljski za programere: Django je programerski prijateljski programski jezik. Nudi intuitivnu sintaksu, doslednost i dobro strukturiranu organizaciju projekata. Podržava princip DRY (Don't Repeat Yourself), modularnost i ponovnu upotrebu koda. Ovo olakšava razvoj, testiranje i održavanje Django aplikacija.

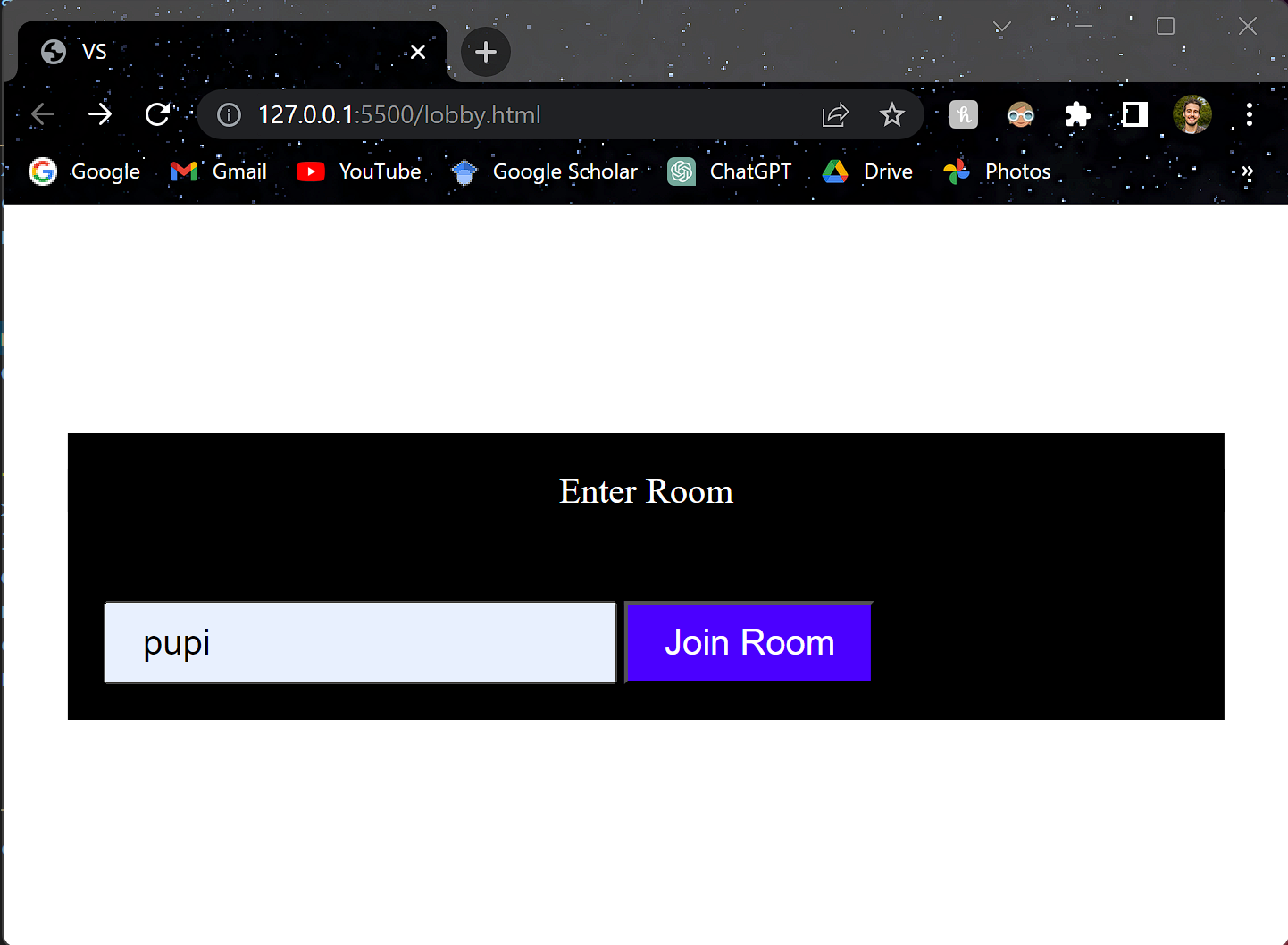
# Postavka i izrada projekta

Cilj ovog projekta je razviti veb aplikaciju koristeci Django okvir. Aplikacija treba da omogući korisnicima da se međusobno povežu i komuniciraju putem video poziva u stvarnom vremenu. U osnovi je kucan kod korišćenjem programskih jezika html i css za front-end, a korisceni su java skrip i python uz django ekstenziju za back-end.

Svi kodovi su kucani u Visual Studio Code IDE okruženju uz ekstenzije Live Server, Python i Pylance.

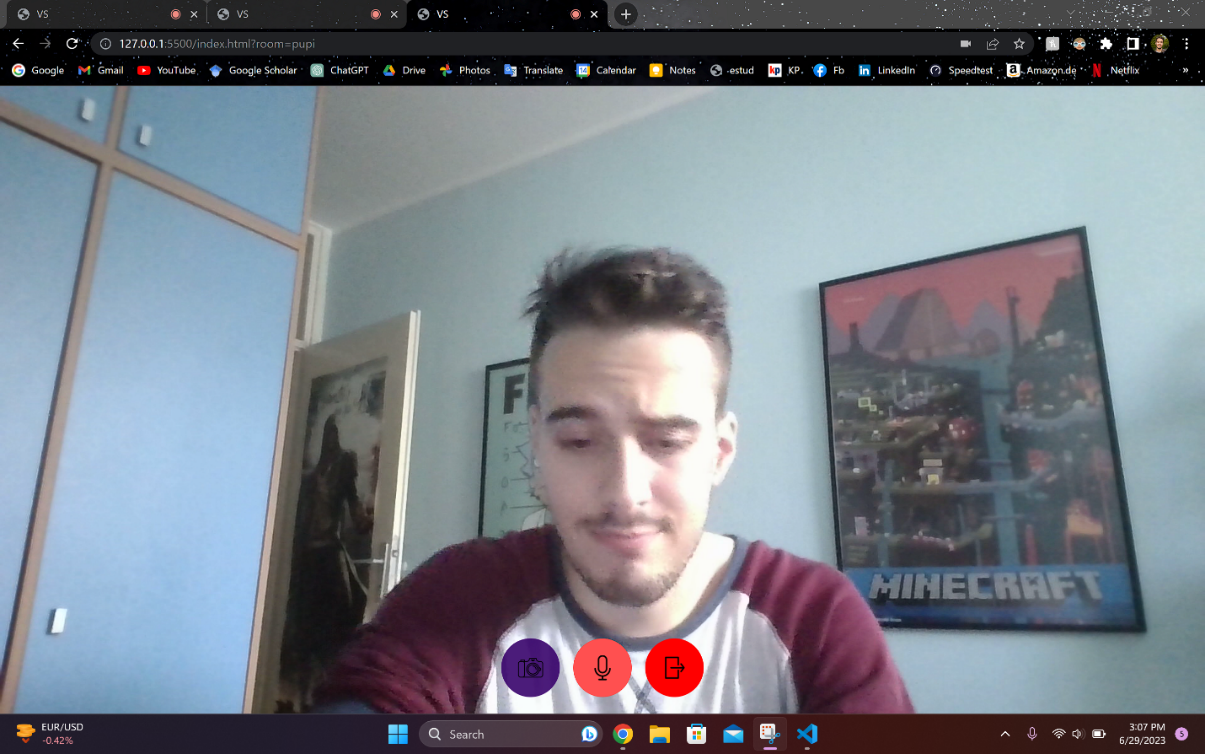
## Funkcionalnosti

Projekat je video-chat aplikacija preko interneta. Aplikacija ima lobby koji dozvoljava korisnicima da formiraju nove pozive ili ulaze u već postojeće. Unosom naziva prostorije koja ne postoji u textbox kreira se nova soba.



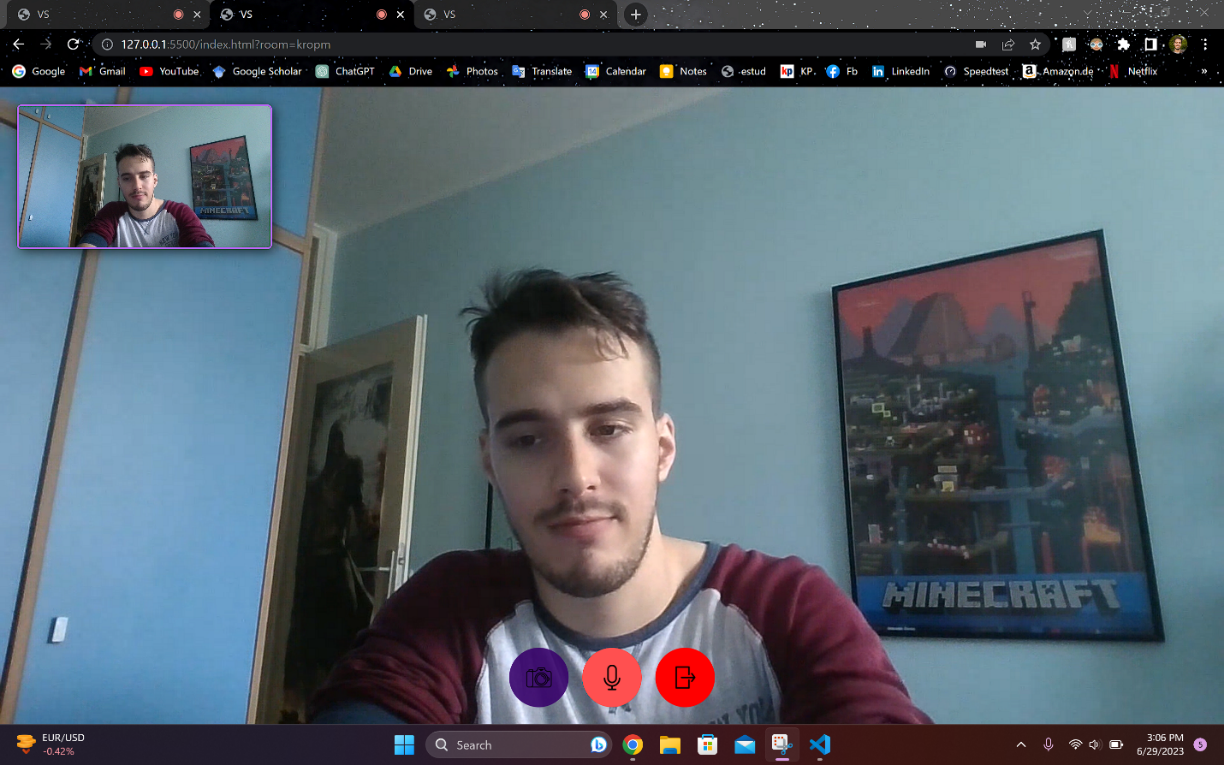
Slika 2.1. Lobby VS cet aplikacije

Na ovoj stranici korisnik dobija upit da li dozvoljava pristup svojoj kameri i mikrofonu od strane sajta. Ništa od ovih podataka se ne čuva u serveru, samo u lokalnim cookies-ima. Nakon toga, korisnik vidi svoj prikaz moze da čuje sebe preko zvučnika.



Slika 2.2. Poziv sa jednim korisnikom u VS cet aplikaciji

Pri ulasku drugog korisnika na isti link ili kucanjem naziva već postojeće prostorije , tj. poziva, u textbox lobby-a putem Agora.io servera inicira se handshake poziv koji obaveštava prvi uređaj o ulasku drugog na server i počinje video poziv. Konekcija je uspostavljena na liniji PTP (Peer-to-Peer) i nije potreban nikakav server za održavanje veze. Svaki od korisnika može da vidi video drugog korisnika preko celog ekrana i manji prikaz svog video sadrzaja u gornjem levom uglu.

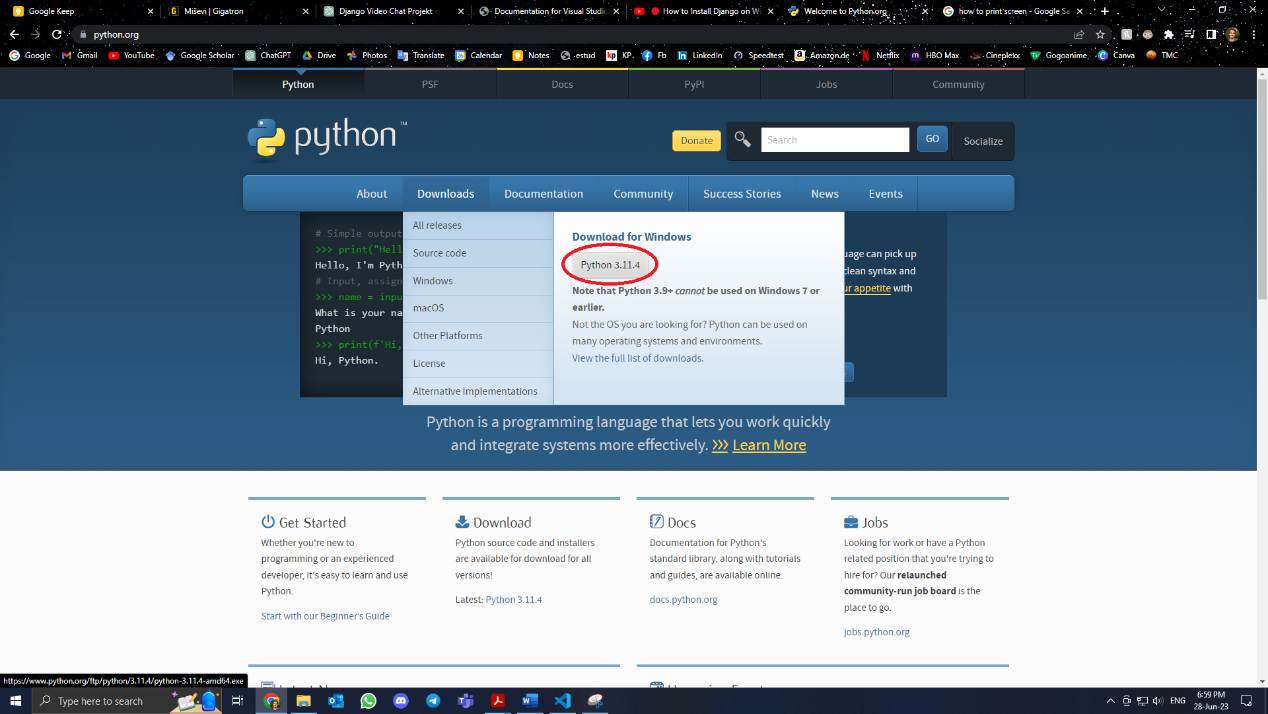


Slika 2.3. Poziv sa dva korisnika u VS cet aplikaciji

Svaki od korisnika takodje ima mogućnost da isključi svoj mikrofon ili kameru pojedinačno. Dugmići koji su ljubičasti označavaju aktivne komponente, dok crveni označavaju neaktivne.Treće dugme se odnosi na napuštanje poziva i vraća nas na početnu stranu. Izlaskom sa stranice na bilo koji drugi način takođe oslobađa drugu poziciju u pozivu. U slučaju da su oba korisnika izašla, prostorija se automatski briše i oslobađa memoriju.

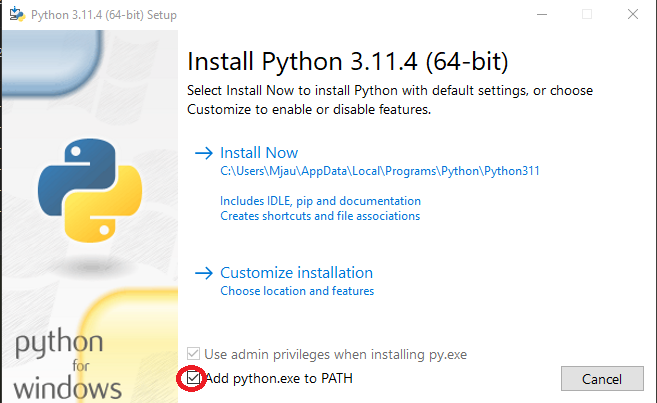
# Instalacija i podešavanja

Prvo je potrebno skinuti i instalirati najnoviju verziju pythona sa sajta [www.python.org](http://www.python.org).



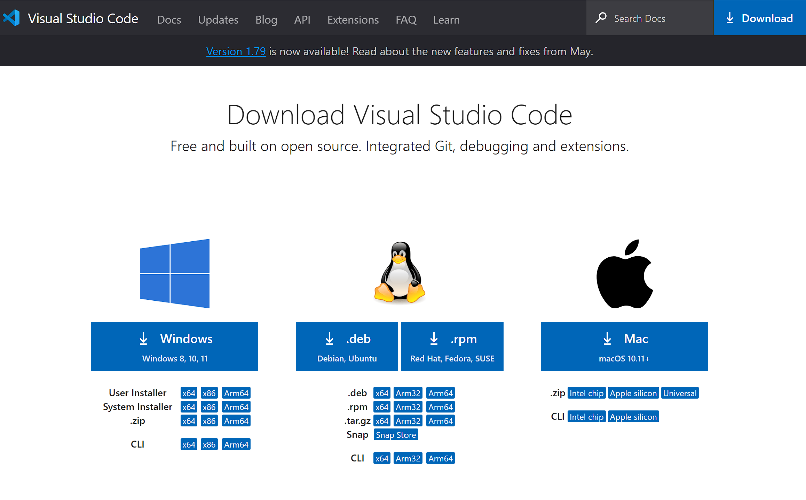
Slika 3.1. Skidanje Python programskog jezika

Pri instalaciji Pythona, neophodno je štiklirati “Add Python.exe to PATH” pre pritiskanja “Install Now”. Odatle će program biti instaliran i možemo da pređemo na instalaciju Visual Studio Code-a.



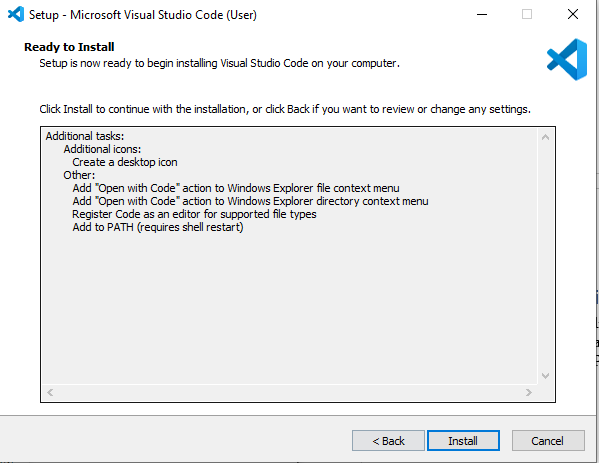
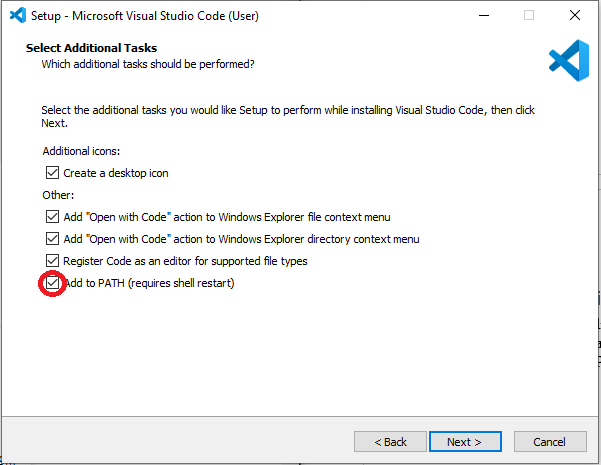
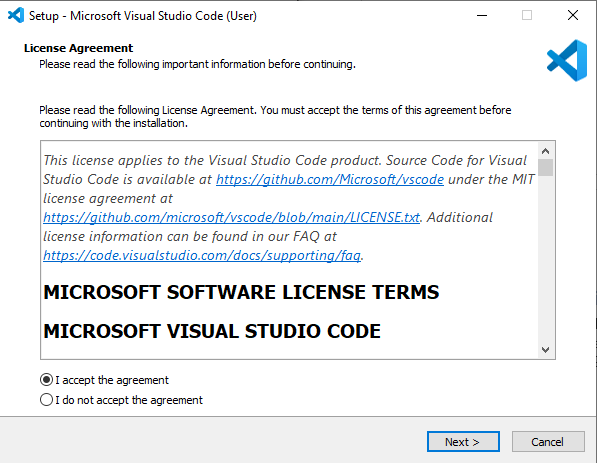
Slika 3.2. Instalacija Python programskog jezika

Sledeća je instalacija Visual Studio Code, IDE okruženja za ovaj projekat, u kome je rađeno sve programiranje i sve nadogradnje i ekstenzije. Od ekstenzija su nam bile potrebni najnoviji Python(v2023.10.1), Pylance(v2023.6.30) i Live Server(v5.7.9). Visual Studio Code skidamo sa zvaničnog sajta [www.code.visualstudio.com/download](http://www.code.visualstudio.com/download), a potom pokrećemo instalaciju.



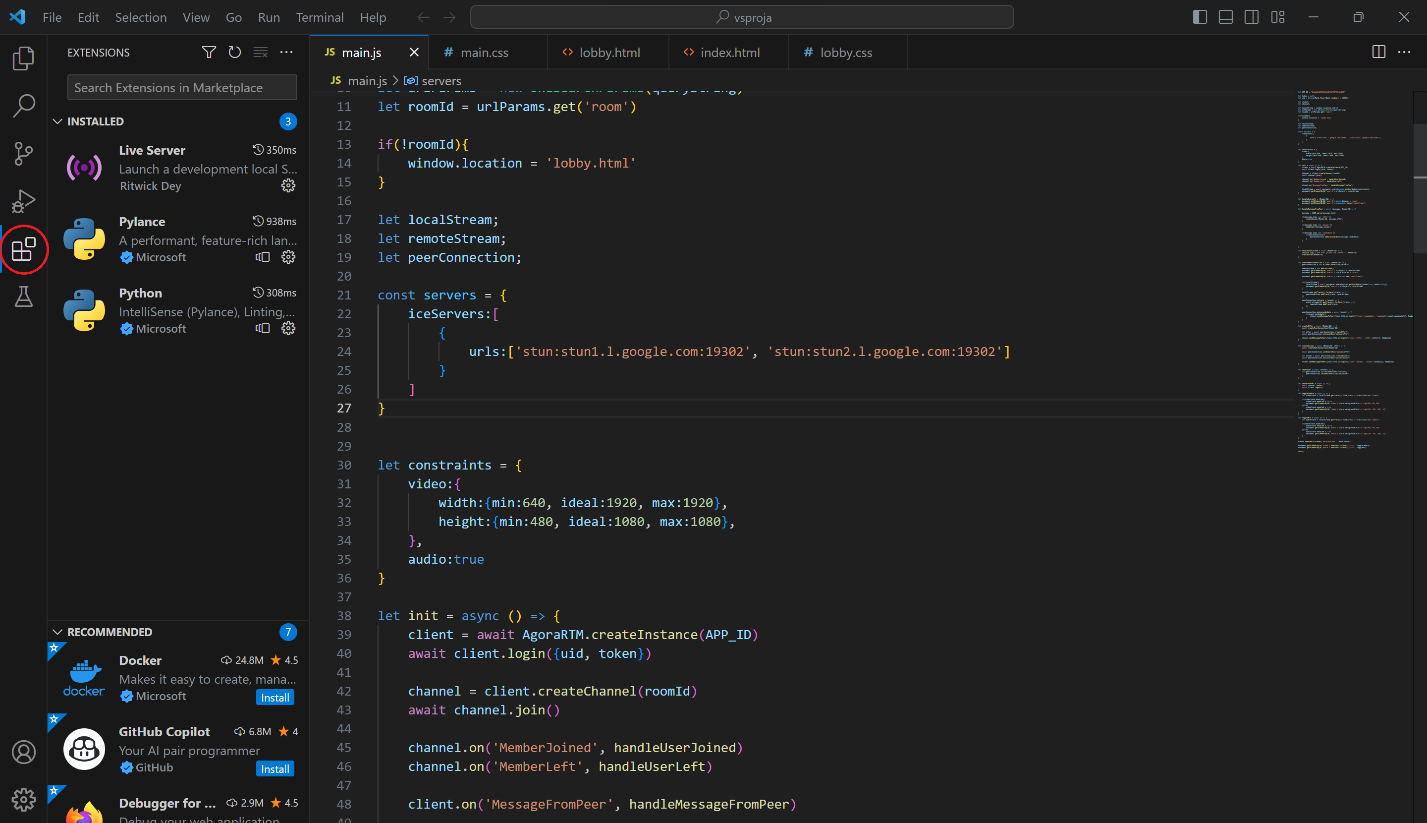
Slika 3.3. Skidanje Visual Studio Code IDE okruzenja

Ovde je takodje bitno štiklirati “Add to PATH” .



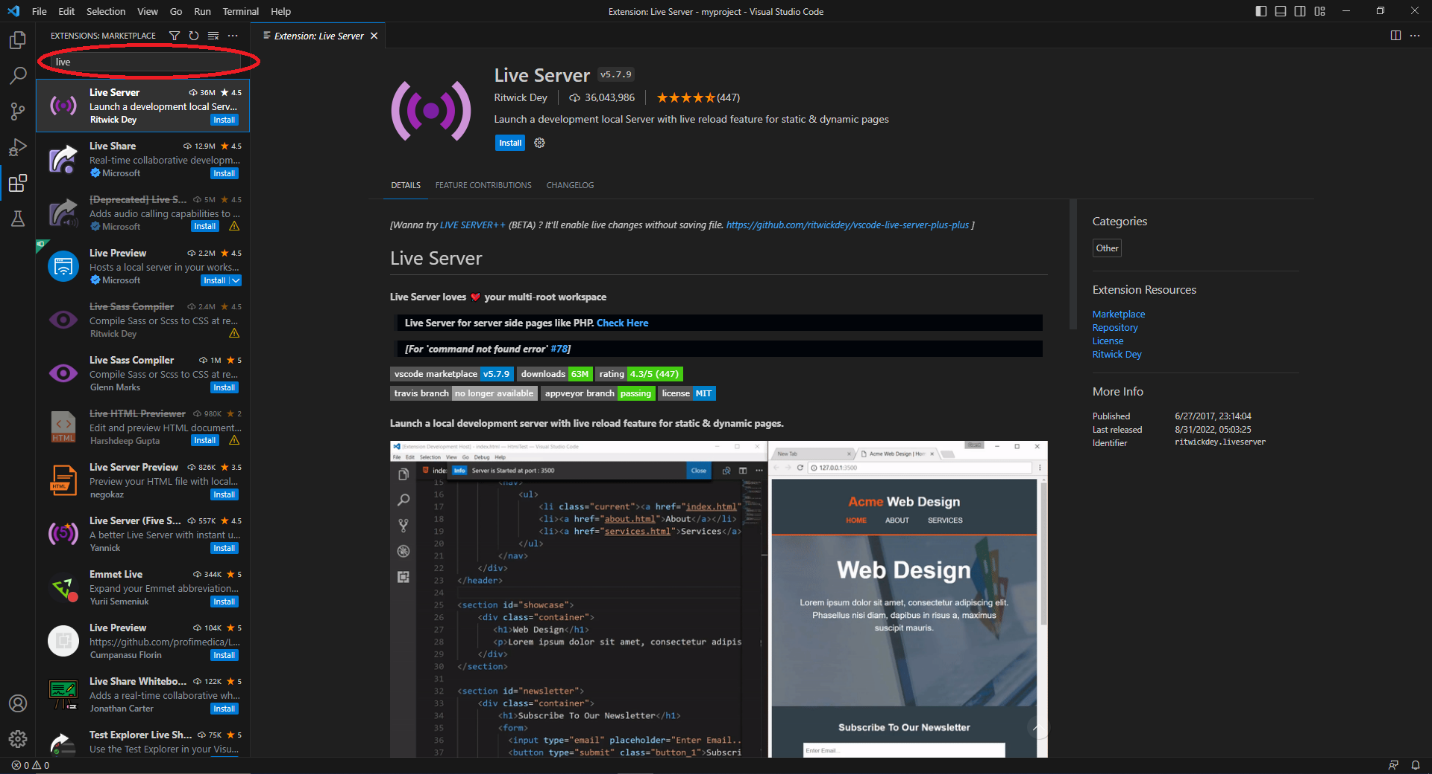
Slika 3.4. Instalacija Visual Studio Code IDE okruzenja

Nakon pokretanja aplikacije i inicijalne postavke, prelazimo na instalaciju ekstenzija. Otvaranjem odeoka za ekstenzije sa leve strane (Zaokruženo crvenom bojom) dobijamo spisak trenutno instaliranih ekstenzija na nasem sistemu i mogucnost dodavanja drugih. Za ovaj projekat su korišćene Python (v2023.10.1), Pylance (v2023.6.30) i Live Server (v5.7.9). Ekstenzije za JS, HTML i CSS mogu biti korisne zbog lakšeg kucanja i predloga sintakse, ali nisu nužne.



Slika 3.5. Dodavanje ekstenzija u Visual Studio Code okruzenju

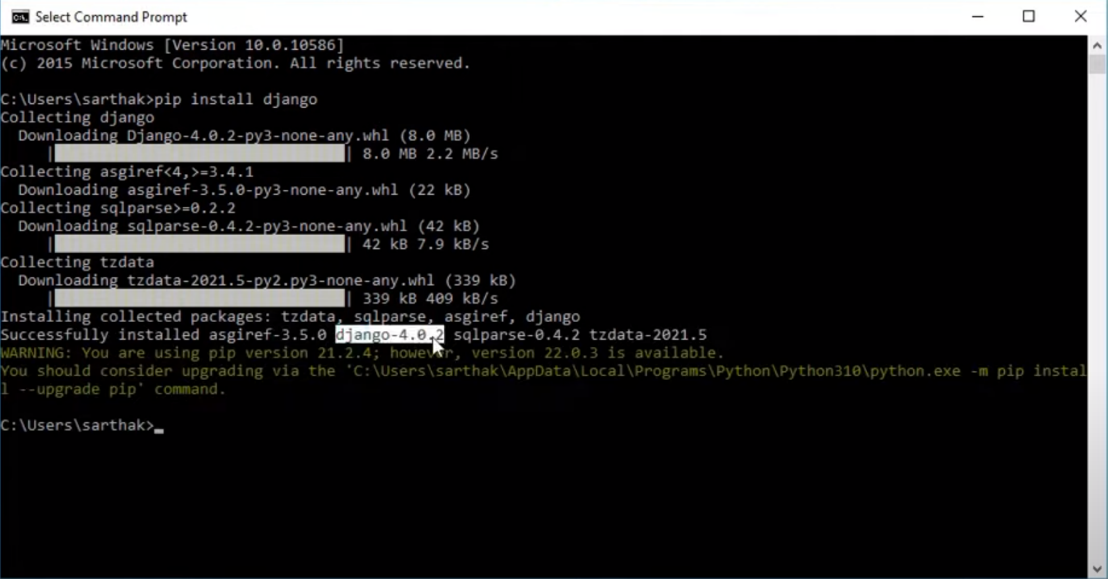
Ekstenziju dodajemo kucanjem njenog naziva u Search-bar i odabiranjem one koja nam odgovara.



Slika 3.6. Pretraga ekstenzija u Visual Studio Code okruzenju

Nakon dodavanja svih ekstenzija prelazimo na instalaciju Django-a. U Command Promptu kucamo sledeće komande:

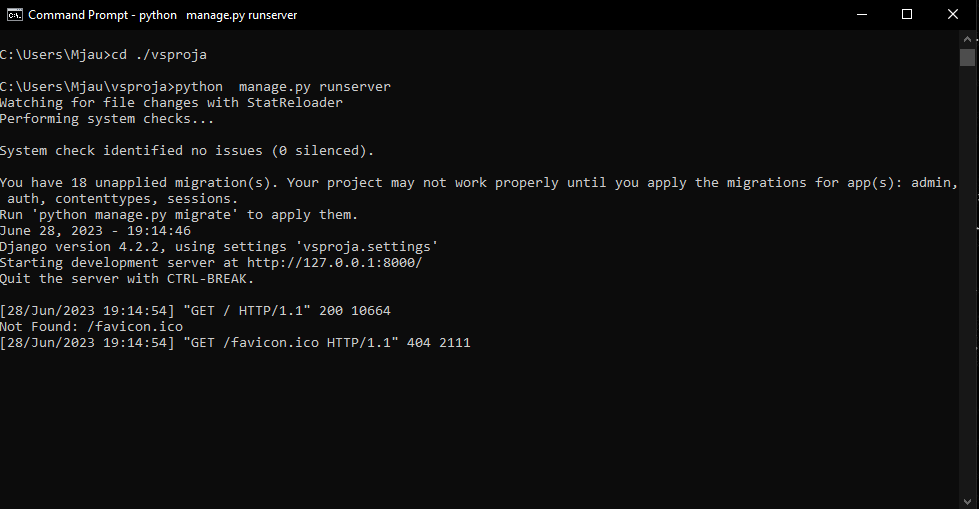
* Za instalaciju: **pip install django**
* Da proverimo da li je Django pravilno skinut: **python -m django --version**
* Za kreiranje projekta: **django-admin startproject vsproja**



Slika 3.7. Instalacija Django nadogradnje

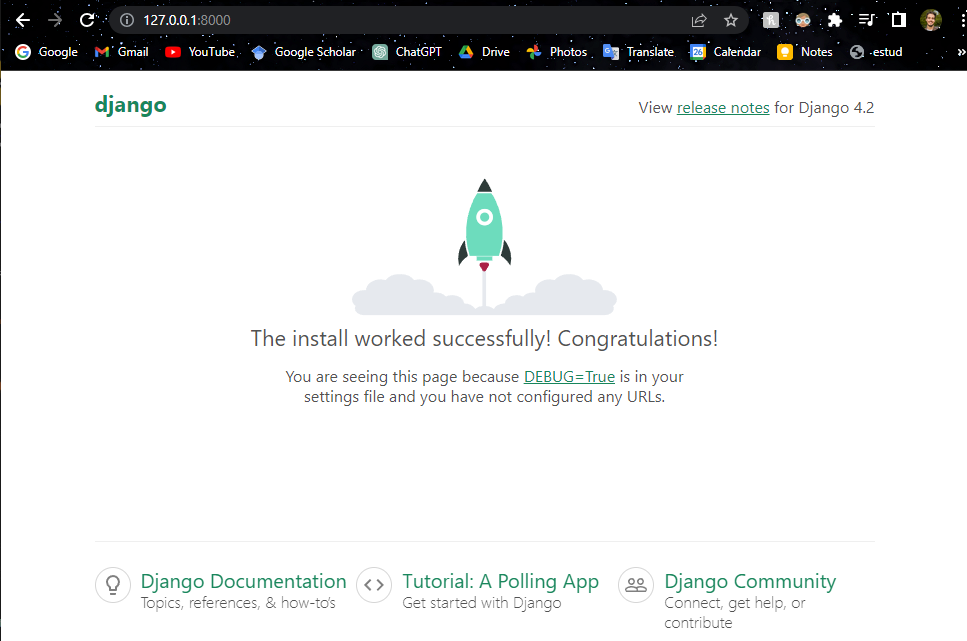
Nakon instalacije i kreiranja prvog projekta potrebno je promeniti direktorijum da bi ušli u projekat i pokrenuli ga. Ovo radimo korišćenjem komandi:

* Za promenu direktorijuma: **cd ./vsproja**
* Za pokretanje projekta: **python manage.py runserver**



Slika 3.8. Pokretanje prve Django web aplikacije

Odatle dobijamo izlaz koji izgleda kao gore na slici i kopiranjem linka servera, u ovom slučaju <http://127.0.0.1:8000/>, i pejstovanjem u Google dobijamo sajt koji nam verifikuje da smo osposobili server.

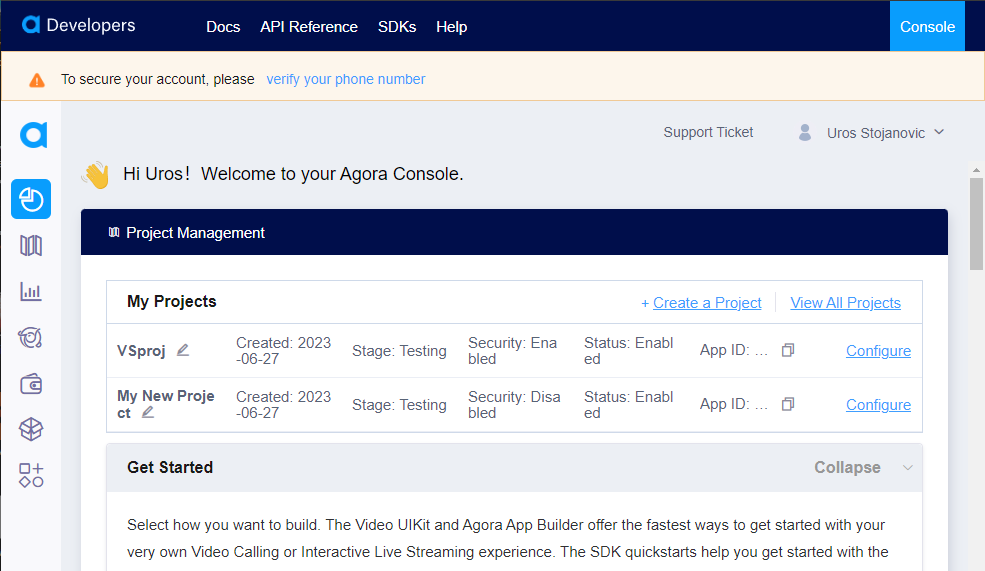


Slika 3.9. Uspesno pokrenuta prva Django web aplikacija

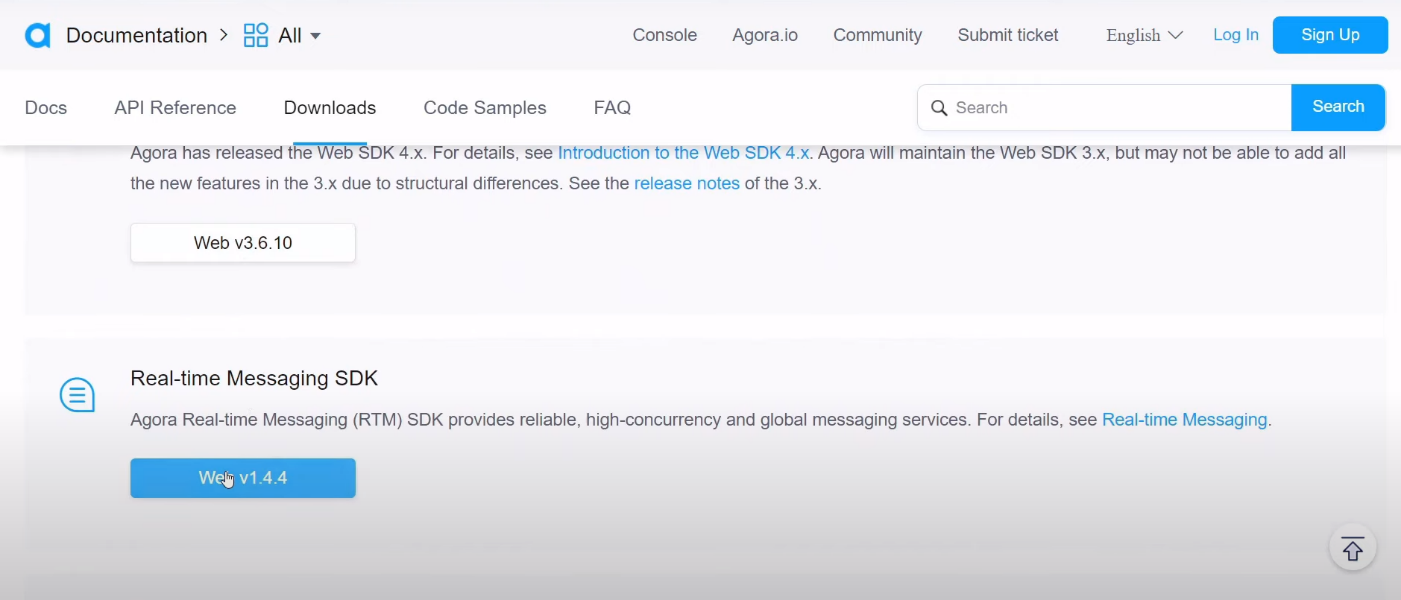
Nakon ove provere možemo da ugasimo Command Prompt i Google Chrome i pokrenemo Visual Studio code. Prevlačenjem foldera u kome smo napravili Django projekat u VS Code ga dodajemo u Edditor. Odavde pravimo nove foldere i podfoldere za sve naše statičke elemente, CSS, HTML, JS i Python kododve.

Za naš sistem video poziva implementirali smo WebRTC protokole koristeći Agora SDK. WebRTC protokoli se svode na korišćenje PTP veze i zaobilaženje servera nakon inicijalnog hand-shake-a. Ovo zahteva da kreiramo nalog na sajtu <https://www.agora.io/en/developer-resources/>.

Odavde kreiramo projekat i skidamo SDK za Web => Real-time Messaging.



Slika 3.10. Kreiranje projekta u Agora.io



Slika 3.11. Skidanje RTM SDK sa sajta Aurora.io

Dodajemo SDK u projekat i prelazimo na programiranje.

# Testiranje zakasnjenja

Nakon povezivanja i testiranja video chat funkcionalnosti moguce je testirati performanse ovog sistema. Neke od tih performansi ukljucuju: protok, zagusenja, broj izgubljenih frameova, broj izgubljenih poziva i kasnjenje. U ovom radu smo se samo bavili testiranjem kasnjenja.

Kasnjenje smo merili pomocu kamere na mobilnom uredjaju S22 Ultra koja ima sposobnosti da snima do 960FPS na rezoluciji od 1280x720. Nakon 5 snimanja objekta u pokretu ispred ekrana na kome se nalazi video stream obavljeno je jos 5 snimanja istog tog objekta u pokretu na isti nacin dok se na ekranu nalazila kamera aplikacija pokrenuta na istom racunaru. Ovo je uradjeno jer sama kamera moze vec da unosi neku kolicinu zakasnjenja i to moramo uzeti u obzir pri testiranju sistema koji se nadovezuju na same funkcionalnosti masine. Naknadno su snimci prebaceni na racunar i pomocu alata FFMPEG prebaceni u pojedinacne slike. Analizom tih slika dobili smo rezultare da je kasnjenje

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Broj merenja | Frame Web | Frame Loc | VremeWeb[ms] | VremeLoc[ms] |
| 1. | 120 | 38 | 125 | 40 |
| 2. | 123 | 44 | 128 | 46 |
| 3. | 140 | 40 | 146 | 42 |
| 4. | 129 | 44 | 134 | 46 |
| 5. | 117 | 50 | 122 | 52 |

Tabela 1. Merenje zakasnjenja

Tlsr=45ms

Twsr=131ms

Kao sto vidimo iz rezultata nas sistem unosi oko 90ms dodatnog zakasnjenja. Takodje je bitno uracunati i ogranicenja opreme, gde je kamera na laptopu ogranicena na 30FPS, samim time unosi dodatnu gresnu od 30ms. Ova ogranicenja nam govore o funkcionalnosti naseg softvera i mogucnosti primena. Za primene za koje je potrebno zakasnjenje od svega nekoliko desetina milisekundi ili manje nas softver nije prihvatljivo resenje. Daljim testiranjem protoka, zagusenja, broja izgubljenih frameova i broja izgubljenih poziva se moze utvrditi efektivnost i skalabilnost ovog rada na visem nivou.

# Zaključak

Django programiranje značajno je transformisalo pejzaž veb razvoja. Njegova elegancija, jednostavnost i sveobuhvatni skup funkcionalnosti omogućavaju programerima da lako izgrade sofisticirane, sigurne i skalabilne veb aplikacije. Odabirom Django-a, programeri se mogu oslanjati na maltene automatizovani back-end uz minimalno znanje povezivanja i samo kreirati frontend veb aplikacije. Django je Open-source što mu je omogućilo nivo i brzinu ravoja koju postiže danas. Uz to dolazi i veliki broj foruma i drugih iskusnih korisnika koji su već testirali i rešili mnogobrojne probleme na koje može novi korisnik da naleti. Obradu paketa, nezavisno od tipa (audio, video, data), vrši uz već ugradjene funkcije i eventualno dodatne SDK nadogradnje. Video Chat aplikacija je samo jedan od mnogobrojnih primena u kojima ga možemo implementirati.

# Literatura

1. The Definitive Guide to django – Web Development Done Right – Adrian Holovaty, Jacob Kaplan-Moss
2. Django for beginners- Build websites with Python and Django – William S. Vincent
3. Traversy Media - https://www.youtube.com/@TraversyMedia – youtube kanal