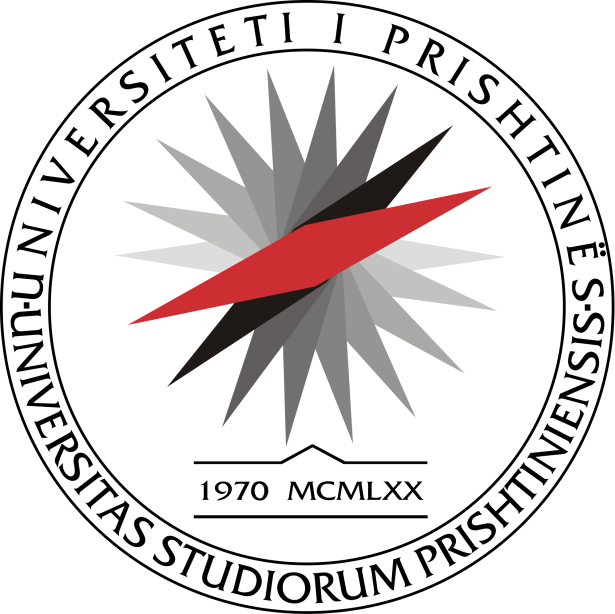
Universiteti i Prishtinës  
"Hasan Prishtina"

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Departamenti Inxhinieri Kompjuterike



RAPORT MBI PROJEKTIN E PARË NË LENDËN RRJETA KOMPJUTERIKE

TEMA E PROJEKTIT : DIZAJNIMI KLIENT – SERVER

Studentja: Profesori i lendës:  
Rrezarta Sallauka Prof.Asoc.Dr.Blerim Rexha

Asistent i lendës:  
 MSc. Haxhi Lajqi

Prishtinë, 2018

Hyrje

Projekti eshte realizuar me gjuhen programuese PYTHON (versioni i përdorur 3.6.3 ) duke bere dizajnimin Klient-Server. Socket është një nga pikat përfundimtare të një lidhjeje të dyanshme të komunikimit midis dy programeve që funksionojnë në rrjet. Foleja lidhet me një numër porti në mënyrë që shtresa TCP të identifikojë aplikacionin që të dhënat janë të destinuara për tu dërguar. Një endpoint është një kombinim i një adrese IP dhe një numri porta.

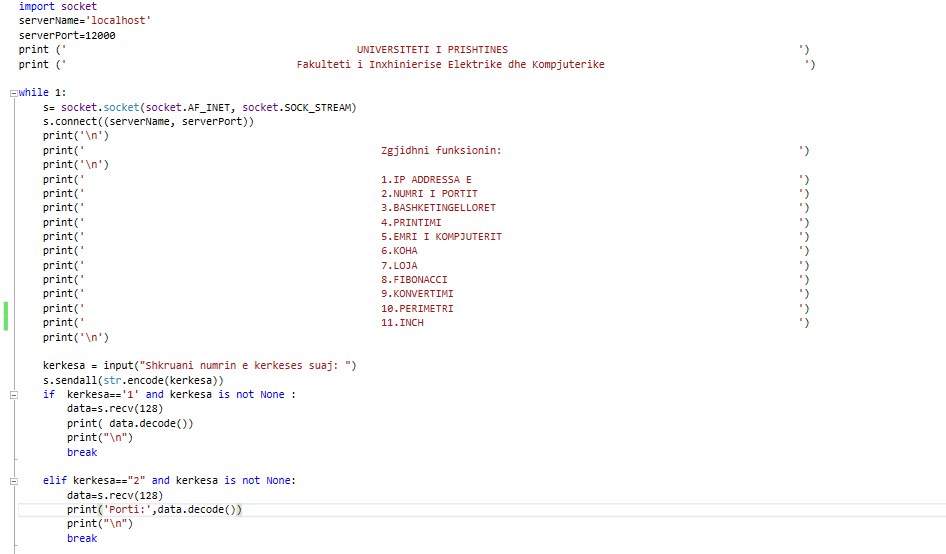
Soketat përdorin dy protokolle të komunikimit, protokollin TCP dhe atë UDP. Ky projekt është realizuar me protokollin FIEK i cili përmban versionin në dy protokollet, FIEK –TCP dhe FIEK- UDP . Te dy këto protokollet kane pjesën përkatëse te klientit dhe te serverit.

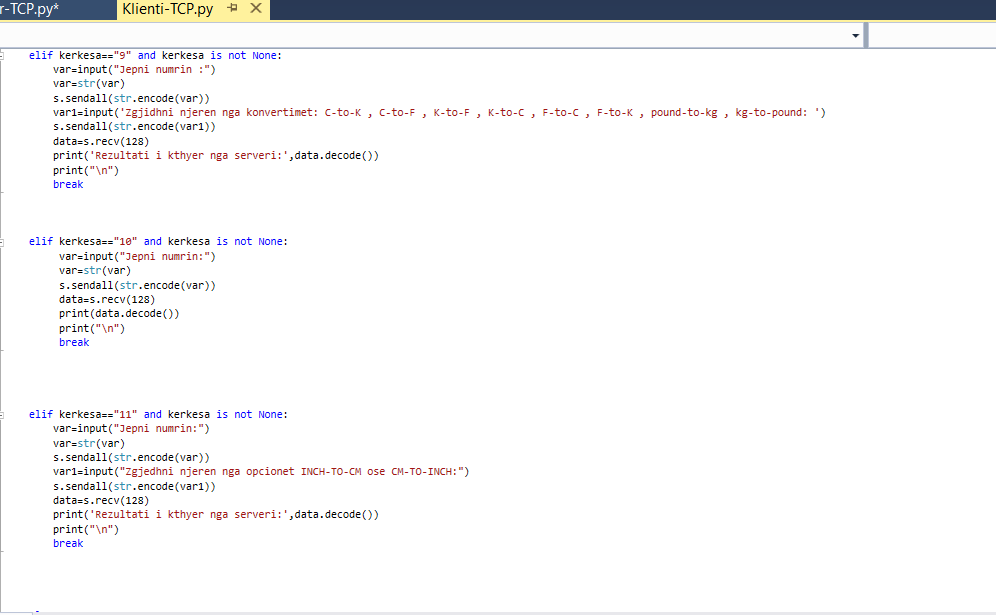
Serveri në të dy protokollet ka përdorur hostin "localhost" dhe portin 12000.

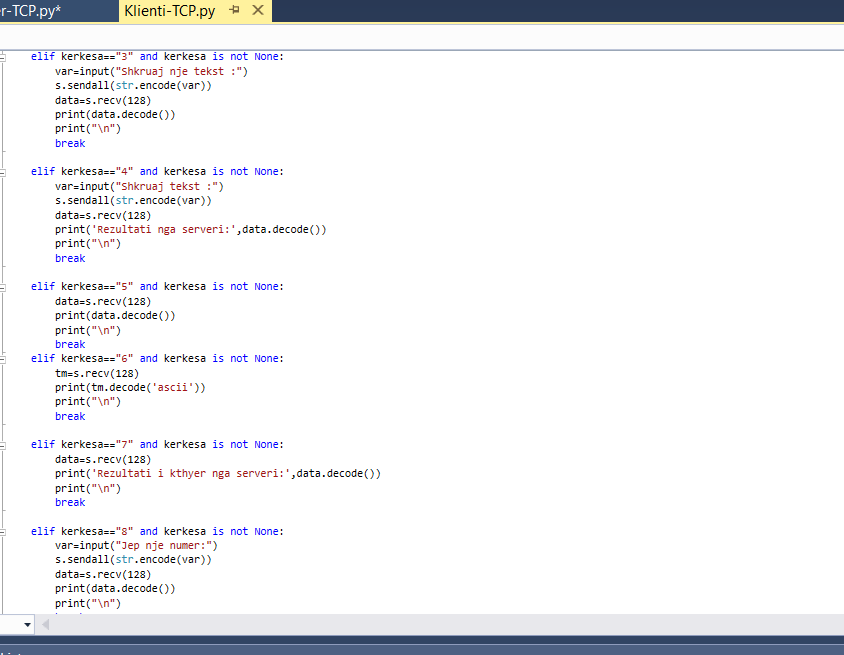
FIEK-TCP-Klienti

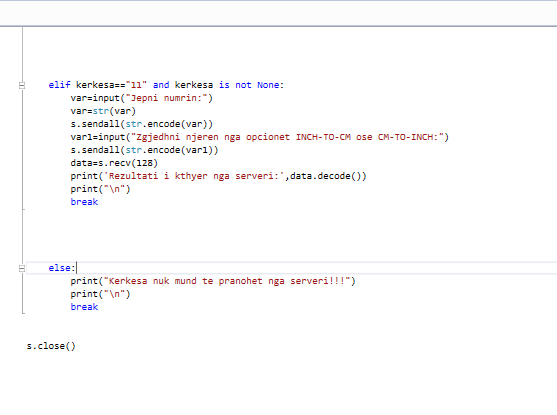
Fillimisht kemi dhenë emrin e portit , pastaj kemi krijuar dhe inicializuar nje soket me protokollin TCP dhe kemi krijuar lidhjen me hostin dhe u deklaruar një variabël e cila merret nga inputi dhe paraqet kërkesën e klientit. Përderisa kjo lidhje ekziston klienti dërgon të dhëna te enkoduara sipas kodit ASCII kurse serveri i pranon ato duke i dekoduar përmes të të njëjtit kod. Pasi klienti zgjedh numrin e kerkeses atehere prej serverit kthehet rezultati.

Kodi i TCP klientit



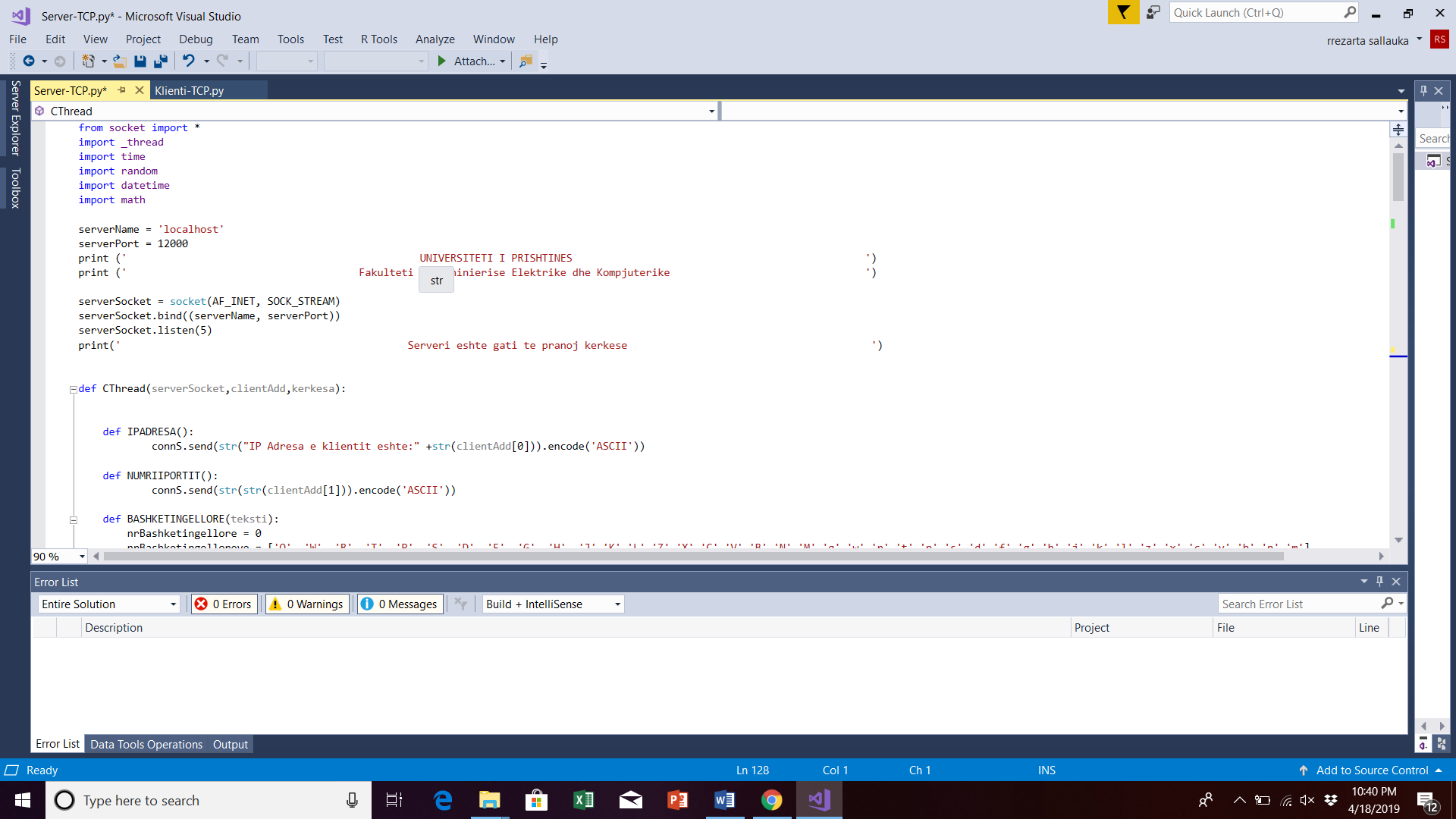




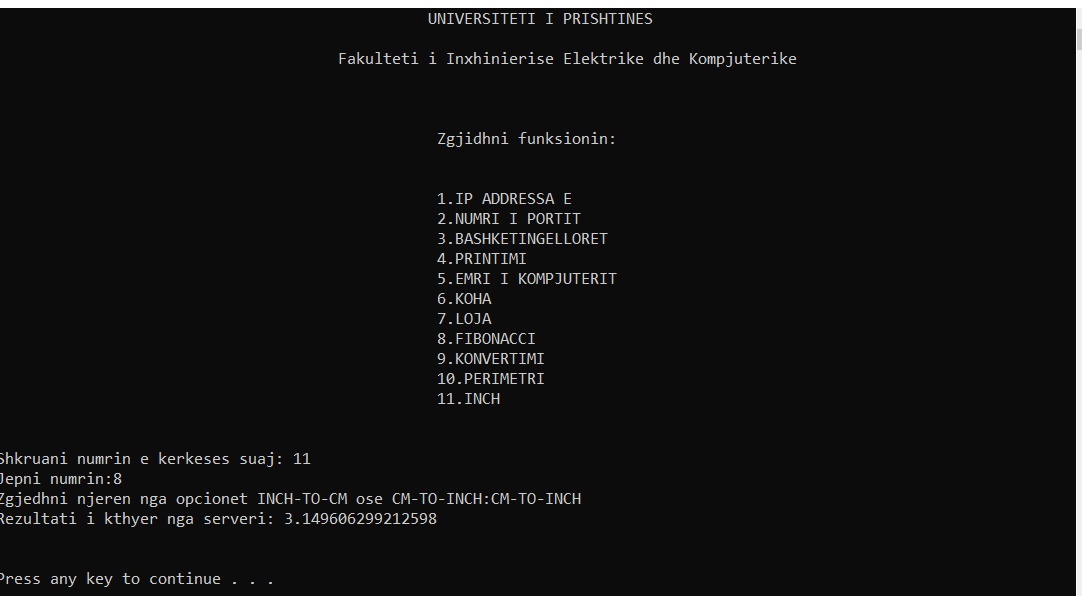
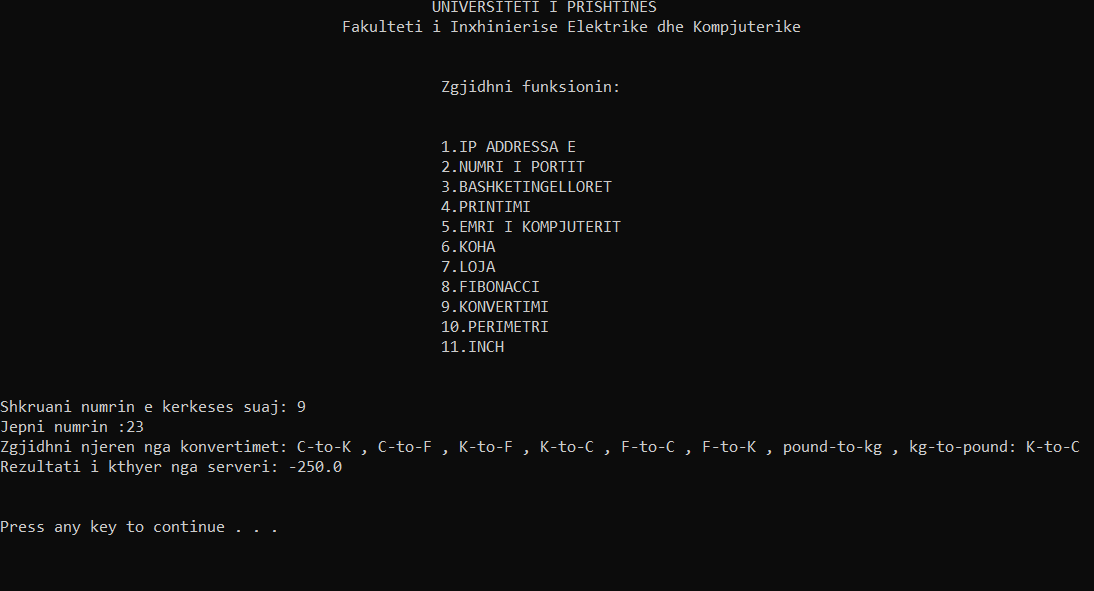
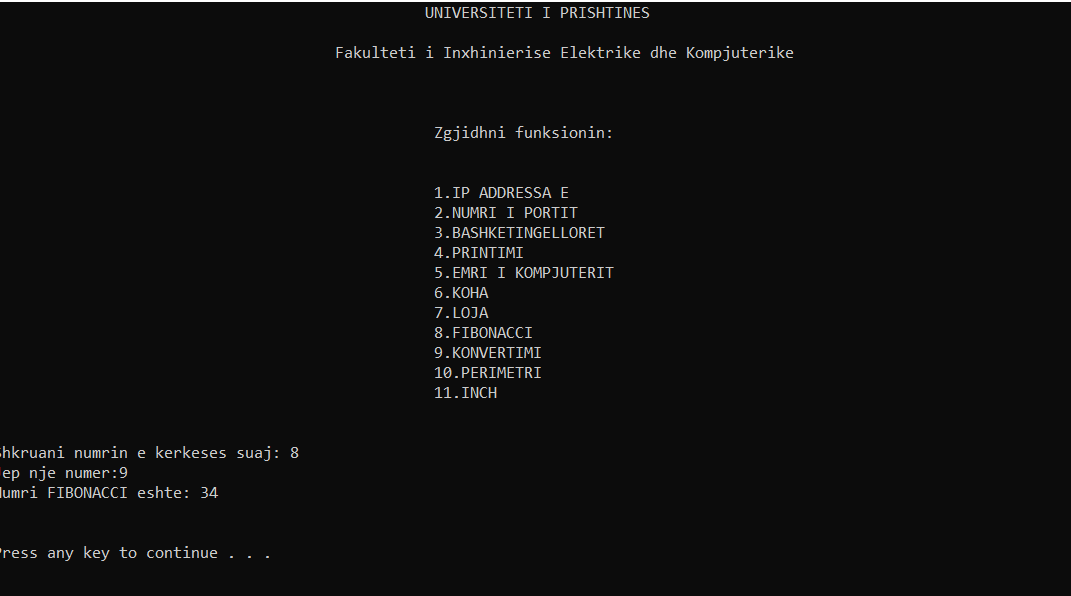
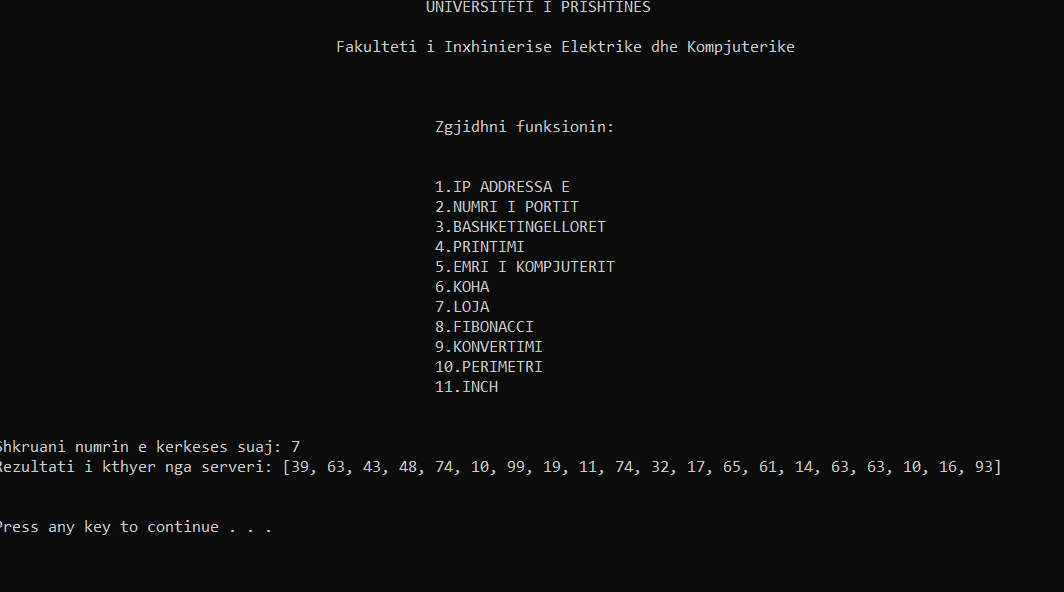
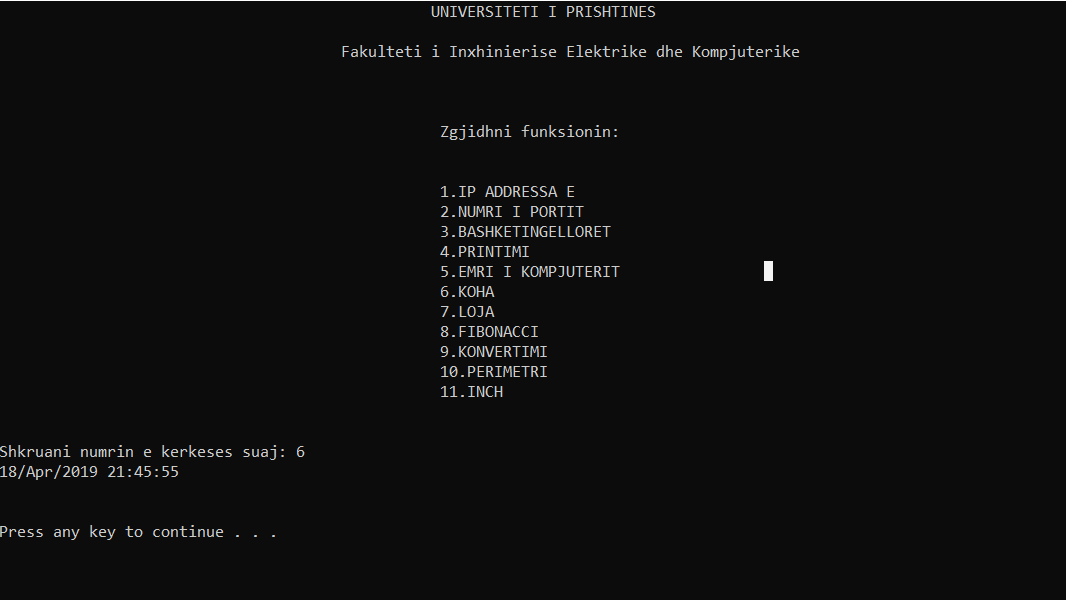
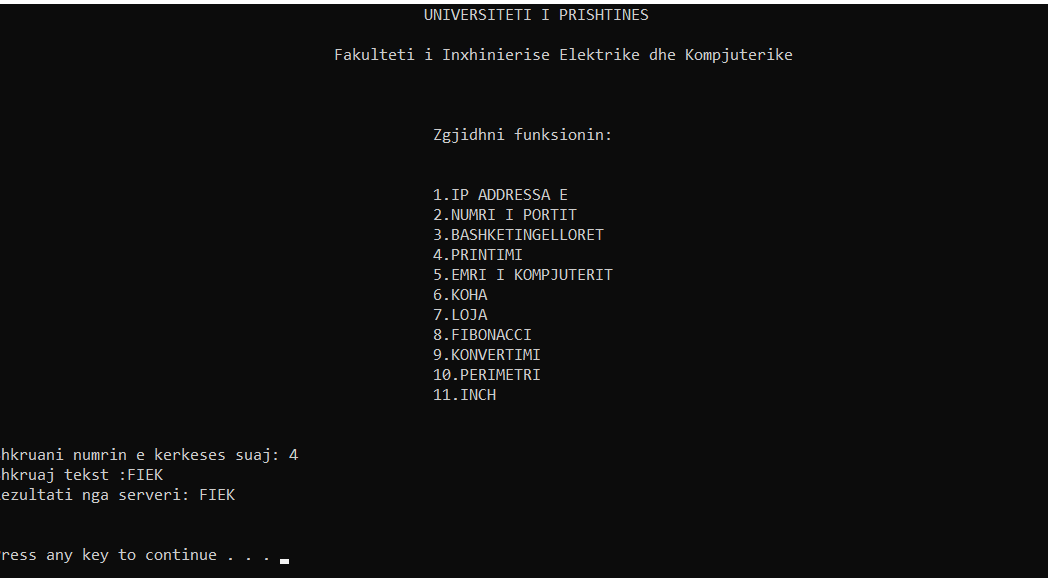
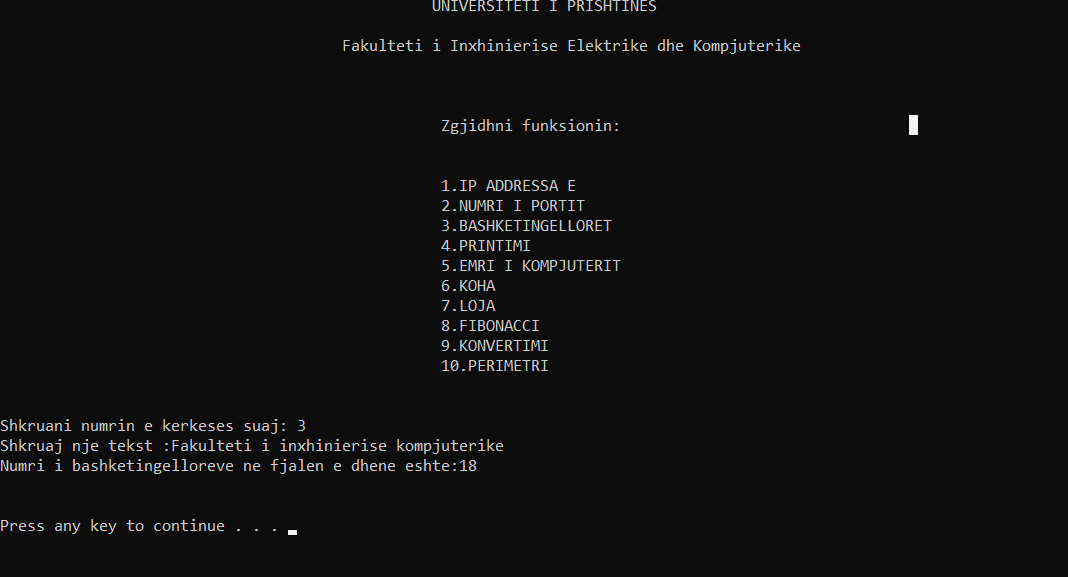
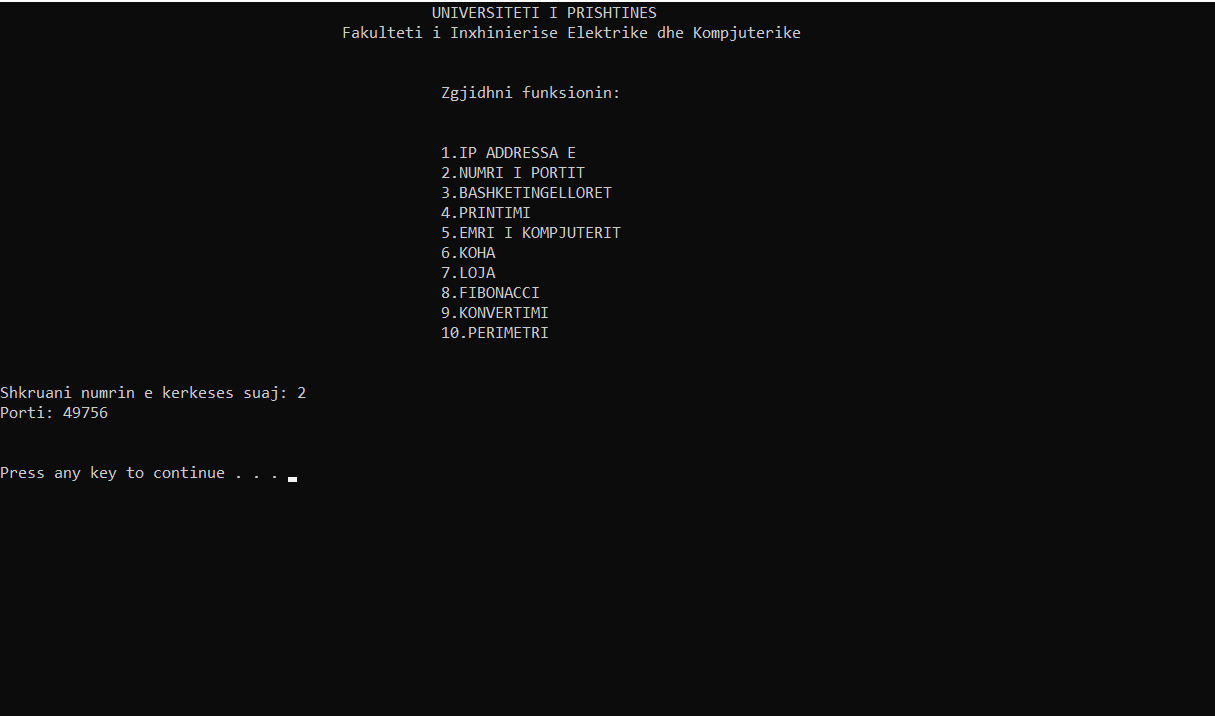
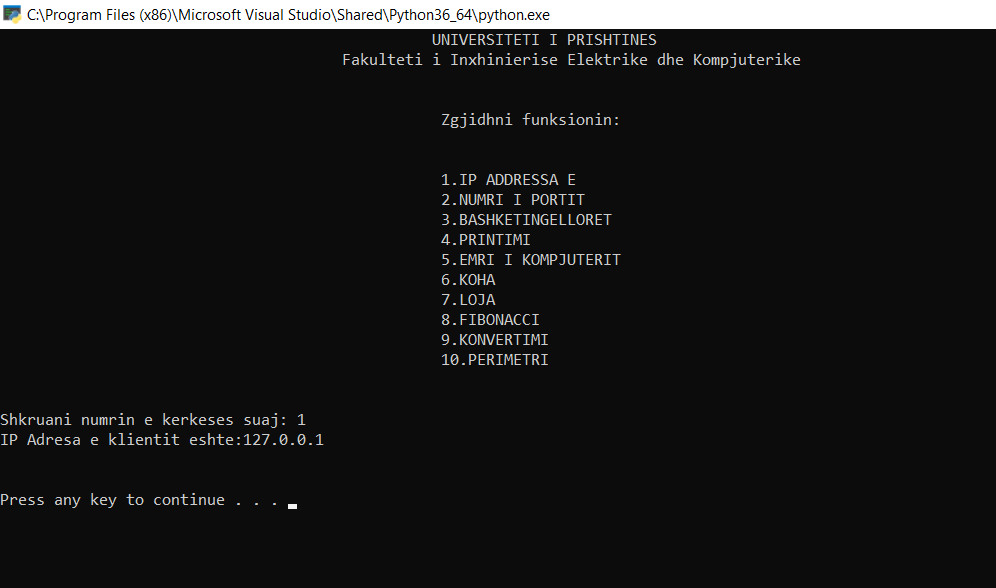
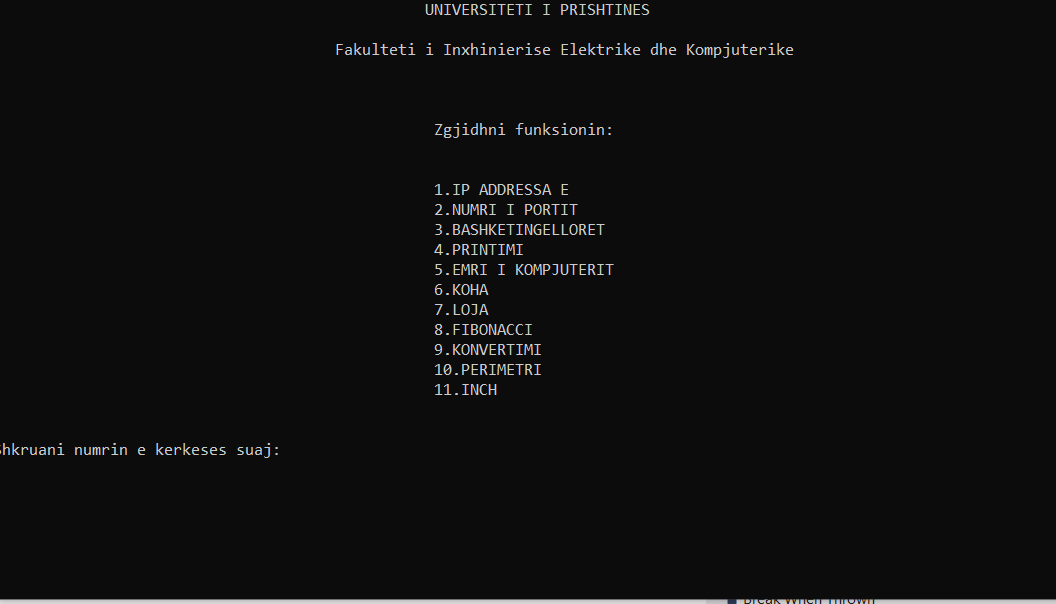


FIEK-TCP-Serveri

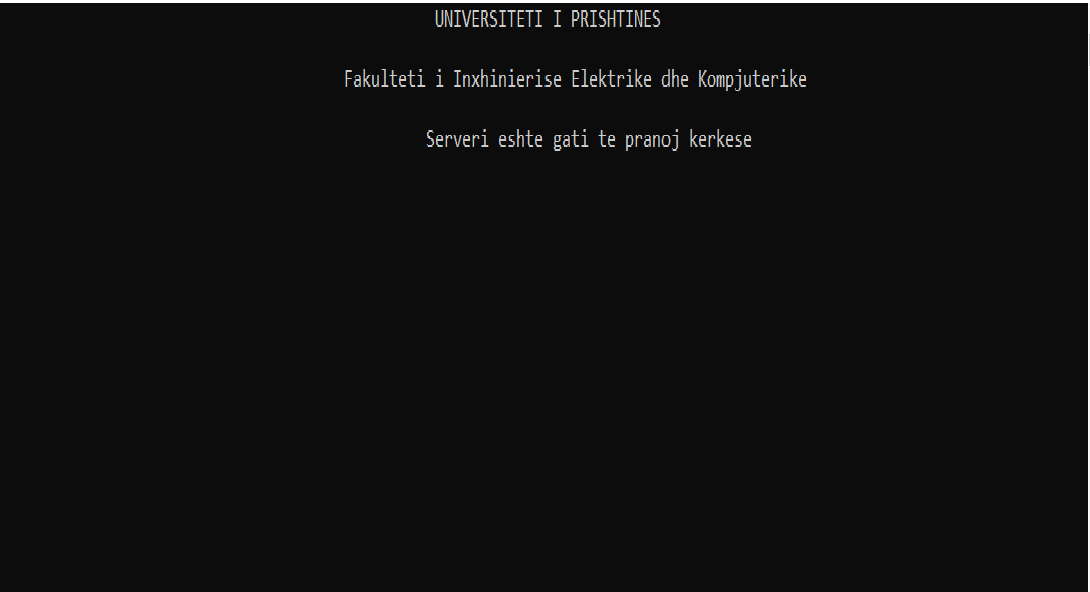
Në këtë protokoll fillimisht kemi importuar disa librari të gatshme që na i ofron gjuha programuese Python te cilat janë përdorur gjatë implementimit te metodave.  
Pastaj kemi krijuar soketin qe lidhet me server ne portin e dhënë për kërkesat qe vijnë. Serveri pret kërkesën dhe përderisa te ekzistoje lidhja mes serverit dhe klientit ai pranon te dhëna.



Ekzekutimi i kerkesave te klientit.



Me ekzekutimin e serverit do te fitojmë nje dritare te tille :



Disa nga metodat e krijuara ne pjesën e Serverit.

Kërkesat te cilat mund te ekzekutohen janë:   
IPADDR - e cila si rezultat paraqet IP adresën e klientit.  
NUMRIPORT - e cila si rezultat paraqet numrin e portit te klientit.  
PRINTIMI – e cila paraqet një fjale qe eshte dhene si tekst hyrës nga klienti.  
EMRIIKOMPJUTERIT – e cila paraqet emrin e hostit.  
TIME – e cila paraqet kohën aktuale në server. Meqense eshte importuar nga libraria math atëherë kemi përdorur formatin e gatshëm (‘%d/%b/Y %X’) , d paraqet ditën , b paraqet muajin, Y paraqet vtin, kurse %X paraqet kohen te dhene ne ore,minuta, sekonda.  
LOJA – e cila paraqet 7 numra te rastesishem nga rangu 1-99, numrat e rastësishëm i marrim përmes librarisë random e cila eshte importuar.  
FIBONACCI – e cila paraqet numrin FIBONACII si rezultat i parametrit të dhënë hyrës,  
KONVERTIMI – e cila paraqet konvertimin e një vlere në njërin nga opcionet : CelsiusToKelvin, CelsiusToFarenheit, KelvinToFarenheit, KelvinToCelsius, FarenheitToCelsius, FarenheitToKelvin, PoundToKilogram, KilogramToPound.  
Si metoda shtesë janë përdorur metoda NRPRIMAR() e cila tregon nëse nje numër i dhënë është primar apo jo , dhe metoda PERIMETRI() e cila llogarit perimetri e një rrethi duke pas si parametër hyrës vlerën e rrezes.  
Në rastin e pranimit të kërkesës për të kthyer rezultatin e saj metodat i kemi ndarë në bazë të pjesës ose "numrit të parametrave" që ato përdorin.

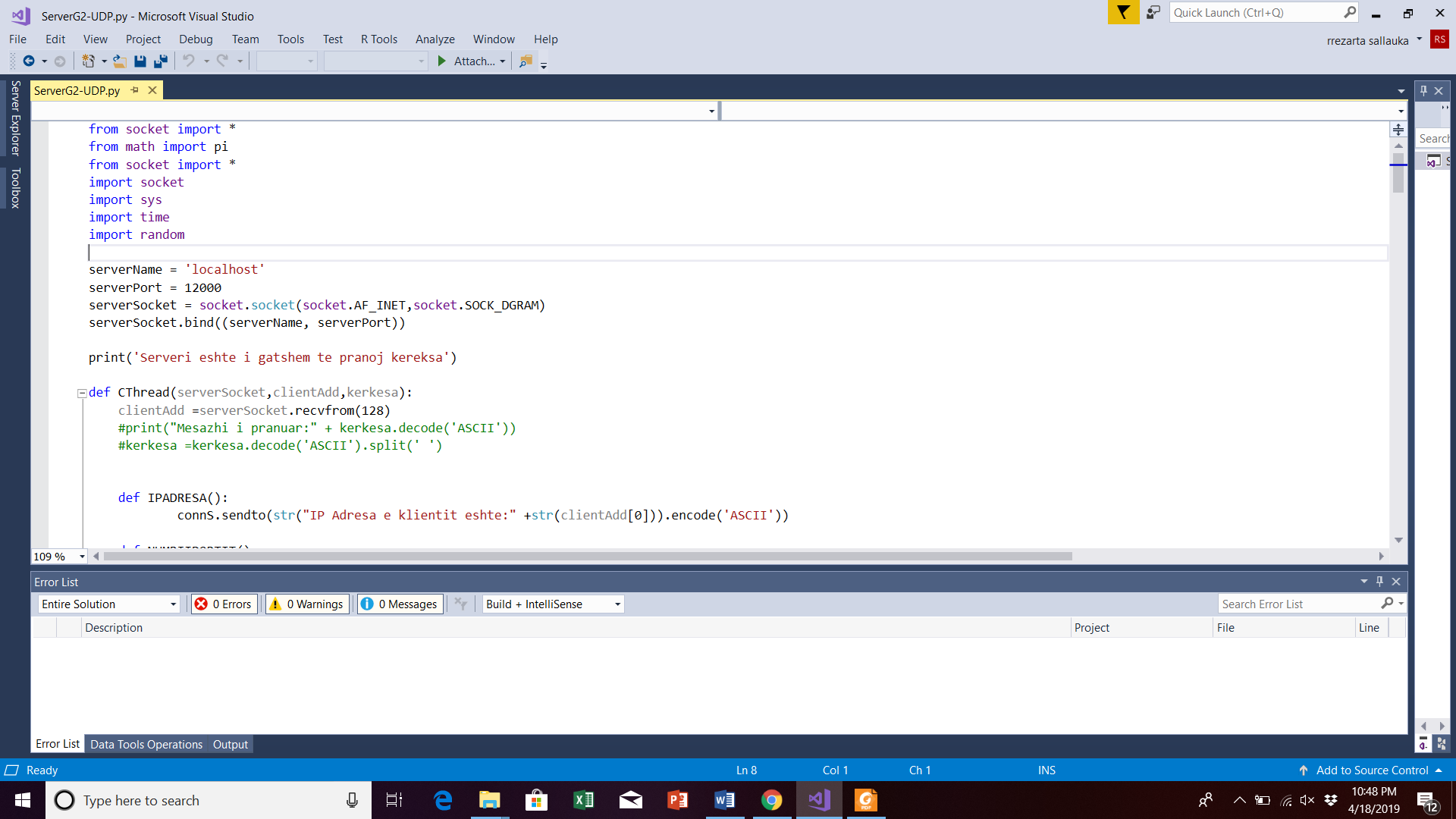
FIEK-UDP-Klienti

Sikur se tek protokolli TCP edhe ne protokollin UDP krijohet lidhja mes klientit dhe serverit, me dallim se këtu lloji i soketes eshte DATAGRAM , kemi nje lidhje me te shpejte por jo aq te sigurt sikur lidhja TCP. Pra, klienti e dërgon kërkesën vetëm përmes UDP datagram, ku mund të dërgohet vetëm një kërkesë.



FIEK-UDP-Serveri

Klienti UDP dallon ne baze te soketit me klientin TCP edhe serverët dallojnë ne baze te tij, pra edhe këtu si tip te soketit zgjedhim SOCK\_DGRAM. Tek ky server përdorim metodën sendto dhe ne fund te qdo pergjigjje shtojmë dhe adresën e klientit. Libraritë e importuara janë te njëjta me libraritë e importuara ne Serverin TCP dhe poashtu perdoret edhe porti i njëjtë për komunikim. Nëse kërkesa e dërguar nga klienti është valide , serveri e kthen pergjigjen ne një UDP datagram.



Kur ekzekutohet e serveri dhe klienti fitojmë këto dy dritare, ku shkruarja e kërkesave behet njejte sikur tek TCP Klienti, po ashtu edhe rezulatet janë te njëjta.

KONKLUSIONI

Ne kete projekt kemi arritur krijimin e nje lidhje ndermjet klientit dhe serverit.

Kur varsishte prej kerkesave dhe nevojave te klientit serveri kthen pergjigjje dhe realizon funksionet te cilat nevojten per ato kerkesa.

. Është arritur krijimi i disa metodave me ç’rast kemi nxjerr rezultatet e pritura nga to, pra nuk ka pasur pengesa ne thirrjen dhe përdorimin e tyre.

REFERENCAT

<https://sites.google.com/site/haxhilajqi/rrjetat-kompjuterike-1/ushtrime-rrjeta>

[https://docs.python.org/2/library/time.html](https://l.facebook.com/l.php?u=https%3A%2F%2Fdocs.python.org%2F2%2Flibrary%2Ftime.html&h=ATOeTjpYo3jPFcSxFoyY8tL1ZPd0_EN1Y4M5110-8WOVhwxkcGrzXAeK_eOpn6VFqz5b4klX2Lws1O2v8uSflPTMyO9zvUoWFwNgimC3mJFxsyvGhf80)