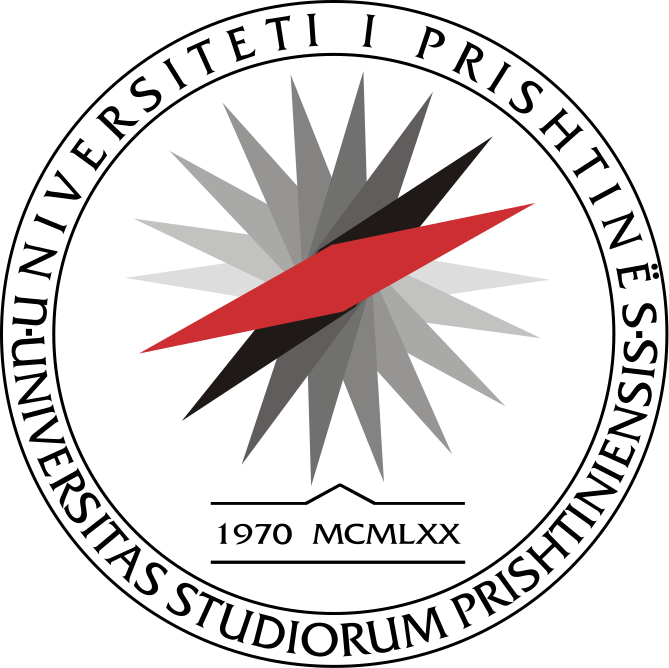
**UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”**

**FAKULTETI I INXHINERISË ELEKTRIKE DHE KOMPJUTERIKE**

**DEPARTAMENTI: INXHINIERI KOMPJUTERIKE**



**TEMA: "Realizimi i një vegle me Python për gjetjen dhe listimin e subdomeneve përmes Brute force algoritmit"**

MENTORI: KANDIDATI:

Prof. AVNI REXHEPI KENAN M. PALLUSHA

# Përmbajtja

#### Abstrakti

#### Abstrakti në gjuhën Angleze- Abstract in English

#### Hyrja

#### Motivimi

#### Përshkrimi i problemit

#### Domenët dhe Subdomenët

#### Domenët e nivelit të lartë (Top level domains)

#### Subdomenët dhe funksionaliteti i tyre

#### Gjuha programuese Python

#### Paketa PIP3 në Python

#### Gjuha programuese HTML,

#### CSS

#### Algoritmet

#### Brute Force Algoritmi

#### Elaborimi Matematikor

#### Algoritmet e Brute Force dhe Greedy: Te përbashkëtat dhe dallimet

#### Vegla për gjetjen e subdomenëve përmes Brute Force algoritmit

#### Rezulltatet

#### Referencat

Formati standard: A4, margjinat 2.5, teksti Times New Roman 12, rreshtimi ne te dy anet (Justify), distanca "single", etj...me afersisht 40-50 faqe.

Rradhitja:

- Abstrakti (shqip dhe anglisht),

- Hyrja (Hyrje, Motivimi, PërshkrimiIproblemit...),

- Kapitull i pergjithshem (ne rastin tend: algoritmet, vetite, perdorimi, karakteristika e tyre, etj...),

- Kapitull i pergjithshem (ne rastin tend: gjuhët/platformat, vetite, perdorimi, karakteristika e tyre, etj...),

- Aplikacioni/shembujt (detajet si dhe çkea është perdorur nga platforma/framework-at, rezultatet, krahasimet, etj).

- Perfundimet (shqip dhe anglisht)

- Referencat (Literatura, Lista e tabelave, lista e figurave, etj).

Në dokument inserton figura, tabela, pamje të ekranit (screen-shots), etj.

# Abstrakti

Bota e sotme është e udhëhequr nga teknologjia, Inteligjenca Artificiale, Machine Learning dhe shërbimet e rrjetit kanë bërë që rritja e mirëqenies njerëzore të rritet në mënyrë eksponenciale. Prania e web platformave ka mundësuar interface të mirë në mes të qenies njerëzore dhe Kompjuterit duke shfaqur një koncept të quajtur Komunikimi Njeri Kompjuterë, por që në vete edhe ka sjellur edhe sfidat globale sociale dhe të sigurisë së informaciont. Një aspekt kritik i theksuar është siguria kibernetike e përdoruesve është web platforma të cilës ne i japim informata çdo herë gjatë përdorimit tonë. Web faqet dhe aplikacionet e internetit janë të hostuara në domenë të ndryshme duke i mundësuar mbarë botës që ka qasje në internet të kyqet në atë domen dhe të merr- jep apo të krijoj interaksion. Domenët brenda vetës përmbajnë edhe nëndomene. Kjo temë e baçhelorit do të shtjelloj rëndësinë e domenëve dhe subdomenëve duke i gjetur dhe listuar ato përmes Brute force Algoritmit. Vegëza për kryerjen e këtij procesi është i shkruar nga gjuhët programuese Python, HTML-CSS dhe PHP. Procesi i gjetjes dhe listimit të nën-domainëve, i njohur si numërimi i nën-domenëve, një rol kyç në hulumtimet si: Siguria dhe Vlerësimi i mbrojtjes kibernetike, Testimi dhe Penetrimi (Hakeret- Etik), Menaxhimi dhe administrimi i Ueb-faqeve, Monitorimi dhe analiza e ueb-faqeve dhe e Domenëve e shumë të tjera.

Vegeza është e shkruar në gjuhën programuese Python në pjesën e integrimit të Brute Force algoritmit dhe është e lidhur me një Front-end te shkruar në HTML, CSS dhe një bazë të dhënash të shkruar në gjuhën programuese PHP.

Mjeti i plotë konceptin e Brute Force, një teknikë e cila kërkon në mënyrë algoritmike ashtu që të përcaktojë një listë emrash të mundshëm apo të kërkuar nga ana jonë të nën-domenëve. Duke përdorur multi- threading dhe programim asinkron, mjeti arrin një performancë të avancuar dhe gjetje më të shpejtë të nën-domene. Vegla të tilla mund të përdoren për vlerësime ashtu që të përdoren për matjen efikasitetin e Ueb-domenëve, testimin e gjerë në një grup të ndryshëm të domainësh, duke përfshirë faqet e internetit private dhe publike, domene korporatash dhe portalet qeveritare. Parametrat e veprimeve që merren në konsideratë gjatë procesit të zbulimit të nëndomaineve janë saktësinë e rezultateve dhe shterrja e burimeve apo shqyrtimi i të gjitha mundësive ashtu që të arrihet rezultati i kërkuar 100%. Përveç kësaj, një analizë krahasuese pas arritjes së rezultateve mund të jetë i vyeshme në shumë fusha të ndryshme. Padyshim, ndër qëllimet esenciale të veglës është që pas analizës së rezultateve Rezultatet e kësaj vegle të shërben në identifikimin e efikasitetit të nën-domaineve, duke kontribuar në përmirësimin e shërbimeve të internetit.

Si përfundim, kjo temë e baçhelorit do të shtjelloj tema të gjëra në kuadër të Domenit, Subdomenit, gjuhëve programuese si Python, PHP, HTML, dhe algoritmin Brute Force dhe do t’i analizoj të rezultatet e gjeneruara nga kërkimi i veglës. Thjeshtësia, efikasiteti dhe natyra e hapur e mjetit e bën atë të ngjeshur dhe të duket për një pamje të gjerë të skenarëve të mjekëve, duke përmirësuar përmirësimin e mbrojtjes së digjitale në një teknologji që vazhdon të evoluojë.

# 

# Abstract in English

Today's world is driven by technology, Artificial Intelligence, Machine Learning and network services have made the growth of human well-being grow exponentially. The presence of web platforms has enabled a good interface between human beings and computers, showing a concept called Human-Computer Communication, but which in itself has also brought global social and information security challenges. A critical aspect emphasized is the cyber security of the users is the web platform to which we provide information every time during our use. Web pages and Internet applications are hosted in different domains, enabling the whole world that has access to the Internet to connect to that domain and receive or create interaction. Domains within themselves also contain subdomains. This bachelor topic will elaborate on the importance of domains and subdomains by finding and listing them through Brute force Algorithm. Vegeza for performing this process is written in Python, HTML-CSS and PHP programming languages. The process of finding and listing sub-domains, known as sub-domain enumeration, a key role in research such as: Security and Cyber Defense Assessment, Penetration and Testing (Ethical Hackers), Website Management and Administration, Monitoring and analysis of web pages and domains and many others.

Vegeza is written in the Python programming language in the Brute Force algorithm integration part and is connected to a Front-end written in HTML, CSS and a database written in the PHP programming language.

The tool completes the concept of Brute Force, a technique which algorithmically searches to determine a list of possible or required sub-domain names. Using multi-threading and asynchronous programming, the tool achieves advanced performance and faster sub-domain finding. Such tools can be used for evaluations as well as for measuring the efficiency of Web domains, extensive testing on a diverse set of domains, including private and public websites, corporate domains, and government portals. The parameters of the actions that are taken into consideration during the process of discovering subdomains are the accuracy of the results and the exhaustion of resources or the examination of all possibilities in order to achieve the required result of 100%. In addition, a comparative analysis after achieving the results can be useful in many different fields. Undoubtedly, one of the essential goals of the tool is that after the analysis of the results, the results of this tool serve to identify the efficiency of the sub-domains, contributing to the improvement of Internet services.

In conclusion, this bachelor topic will elaborate on broad topics within Domain, Subdomain, programming languages such as Python, PHP, HTML, and Brute Force algorithm and I will analyze the results generated by the search tool. The simplicity, efficiency and open nature of the tool makes it compact and visible for a broad picture of clinician scenarios, improving the improvement of digital protection in a technology that continues to evolve.

# Hyrja

Kjo temë e Baçhelorit do të shtjelloj krijimin, dizajnimin dhe analizimin e një vegëze të shkruar në gjuhët programuese Python, HTML, CSS, dhe PHP ashtu që përmes një Algoritmi i quajtur Brute force që shërben për kërkim, gjetje dhe listim të subdomenëve apo subdomenëve të një domeni më të madh. Alogritmi Brute force është një algoritëm i shumë përdorur që jep rezulltate të sigurta, rrjedhimisht atë që e kërkojmë mund ta gjejmë përmes implementimit të këtij Algoritmi. Si pikë positive e ka këtë aspekt, kurse si pikë më pak të favorshme e ka efeçiencën gjatë kërkimit. Brute Force Algoritmi përdor çdo mundësi të mundshme për të arritur në rezultatin e kërkuar. Nuk ka ndonjë selektim, grupim, apo logjike më specifike për t’i anashkaluar disa mundesi apo opcione ashtu që të arrijmë deri të rezultati më shpejtë pa i kaluar të gjitha mundësitë. Kjo mbase e bënë algoritmin të përdorshëm po jo të favorshëm në aspekt të vlerësimit efeçientë të kohës dhe kalkulimit kompjuterik. Kodi i algoritmit është I shkruar në gjuhën programuese Python dhe është i lidhur me një Front-end të shkruar në gjuhën programuese HTML dhe CSS. Poashtu për listimin e e subdomenëve të një domeni të caktuar është e implementuar dhe një back-end apo database e shkruar në gjuhën programuese PHP.

Listimi i subdomenëve të një domeni përkatës mund të shërbej në disa sfera të ndryshme në shkencat kompjuterike dhe në siguri kibernetike. Listimet e tilla mund të shpien në rezultate shumë të vyeshme sa i përkët performancës së Domenit apo Webfaqes, sigurisë kibernetike apo edhe gjurmimit të dhënave.

# Motivimi

Qëllimi dhe motivimi për këtë temë, buron nga rëndësia dhe ndikimi në rritje i përdorimit të listimit të subdomainëve në fusha të ndryshme, duke përfshirë sigurinë kibernetike, zhvillimin e uebit dhe menaxhimin e rrjetit. Regjistrimi i subdomainëve luan një rol kryesor në identifikimin e dobësive të mundshme, optimizmin e infrastrukturës së uebit dhe rritjen e sigurisë së përgjithshme dixhitale. Duke kryer një studim të thelluar të teknikave të numërimit të nën-fushave, tema synonon të kontriboj me njohuri të vlefshme që mund të përfitojnë organizatat dhe individët në mënyra të shumta:

**Rritja e sigurisë kibernetike:** Në një botë ku kërcënimet kibernetike po evoluojnë vazhdimisht, të kuptuarit e teknikave të listimit të subdomenëve mund t'i ndihmojë organizatat të identifikojnë pikat e mundshme hyrëse për sulmet kibernetike. Hulumtimi i tille mund të kontribuojë në përmirësimin e pozicionit të sigurisë së sistemeve dhe faqeve të internetit duke mundësuar zbulimin në kohë dhe zbutjen e dobësive të sigurisë.

**Menaxhimi efikas i Infrastrukturës së Uebit:** Menaxhimi efektiv i infrastrukturës së uebit është thelbësor për ofrimin e përvojave pa probleme në internet. Duke studiuar numërimin e subdomenve, ju mund të ofroni strategji për të përmirësuar dhe optimizuar burimet e uebit, duke çuar në përvoja më të mira të përdoruesit dhe shpërndarjen e burimeve. Në thelb, Tema ka potencialin për të lidhur njohuritë teorike me aplikime praktike, duke adresuar sfidat e botës reale dhe duke ofruar zgjidhje që mund të ndikojnë pozitivisht në industri të ndryshme.

**Fuqizimi i Hakimit Etik dhe Testimi i Penetrimit:** Hakerët etikë dhe testuesit e depërtimit (pentrimit) mbështeten në informacionin e saktë të nëndomainit për të simuluar sulmet e mundshme dhe për të vlerësuar dobësitë qe potencialisht mund të krijohen. Materiali hulumtues i kësaj teme mund t'i pajis profesionistet me mjete dhe teknika më të mira për identifikimin e pikave të mundshme të hyrjes dhe ndihmën në masat pro-aktive të sigurisë.

**Mbështetja e marketingut dixhital dhe markimit:** Nëndomainët përdoren shpesh për fushata marketingu të synuara dhe iniciativat e markës. Hulumtimi juaj mund të ofrojë njohuri se si organizatat mund të përdorin në mënyrë strategjike nënfushat për të krijuar përvoja të dallueshme në internet që angazhojnë përdoruesit në mënyrë efektive.

**Përparimi i praktikave të zhvillimit të uebit:** Zhvilluesit mund të përfitojnë nga një kuptim i thellë i teknikave të numërimit të nëndomainëve për të siguruar aplikacione të fuqishme ueb. Shtjellimi i kësaj teme mund të hedhë dritë mbi rreziqet e mundshme që lidhen me nëndomenët dhe të ofrojë udhëzime për praktikat e sigurta të zhvillimit të uebit.

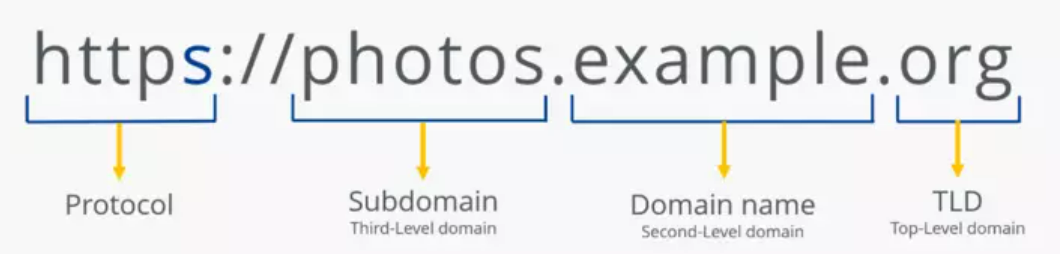
**Aktivizimi i analizës gjithëpërfshirëse të rrjetit:** Për administratorët dhe analistët e rrjetit, njohja e subdomenëve të lidhura me domenin e një organizate mund të çojë në një analizë më të plotë të rrjetit dhe në përmirësimin e vendimmarrjes.

**Kontributi akademik**: Duke u thelluar në teknikat e numërimit të nënfushave, teza juaj kontribuon në bazën e njohurive të komunitetit akademik. Ai i shton të kuptuarit kolektiv të sigurisë kibernetike, zhvillimit të uebit dhe rrjetëzimit, duke ofruar një referencë për përpjekjet e ardhshme kërkimore.

# Përshkrimi i Problemit

Gjatë përdorimit të Internetit, në shumicën e kohës jemi duke përdor domene të ndryshme. Domenët luajnë rol të rëndësishëm në strukturimin e ueb-faqes prandaj, trajtimi i tyre është i nevojshëm për të arritur një qasje më të mirë në disa aspekte sikur siguria kibernetike dhe përmirësimi i praktikës së përdorimit të tyre, të cilat i kemi cekur dhe më lartë.

# Domenët dhe Subdomenët (nëndomenët)



Domenat dhe nën-domenat janë koncepte të emërtimit në strukturën e sistemit dhe adresimit të internetit, të njohur si Sistemi i Emërimit të Domaineve (Domain Name Sytem - DNS). Ata marrin një rol kritik në identifikimin dhe gjetjen e burimeve dhe adresave në internet. Le të eksplorojmë se çfarë domenat dhe nën-domenat dhe si duhet:

Një domen është një emër i lexueshëm nga njerëzit që përkon me një adresë IP ose një grup adresash IP. Përdoret për faqet e internetit, shërbimet dhe burimet e tjera në internet. Domenat janë hierarkike dhe organizohen nga e djathta në të majtë, të ndara (pikat). Çdo pjesë e domenit quhet një etiketë (label). Për shembull, në domenin "[www.shembull.com](http://www.shembull.com)", "com" është domeni i nivelit të lartë (TLD), "shembull" është domeni i nivelit të dytë (SLD). ), dhe "www" është një nën-domen. Domenat duhet të sigurohet që emri i domenit të jetë unik, pra të mos jetë i përdorur nga asnjë përdorues tjetër.

Nën-domenat: Nën-domenat janë nën-ndarje të një domeni më të madh. Përdoren për të krijuar seksione ose degë shtesë brenda një domeni. Nën-domenat krijohen duke shtuar një etiketë në fund të domenit kryesor, të ndara me një pikë. Për shembull, në domenin "blog.shembull.com", "blog" është një nën-domen dhe "shembull.com". Në thelb, dhe nën-domenët kanë një mënyrë të strukturuar dhe intuitive për të lëvizur në internet, duke u bërë më të lehtë për të gjetur përdoruesit dhe që janë në burime në internet. Ata në organizim, markë dhe funksionimin e faqeve të internetit dhe shërbimeve në linjë.

Nën-domenët duhet të përdoret shpesh për kategoritë e përmbajtjes ose shërbimeve brenda një faqe interneti. Ata mund të punojnë drejt serverëve të ndryshëm, direktivë ose të veçanta, ndërsa janë ende pjesë e të njëjtit domen. Nën-domenat të shumëanshëm dhe të mundshëm janë për qëllime të ndryshme.

***Domenët mund të jenë të këtyre shembujve:***

Struktura Organizative Kompanitë mund të përdoren nën-domena për departamentet ose divizionet e ndryshme. Për shembull, "shitje.shembull.com" dhe "mbështetje.shembull.com".

**Shërbime**

Nën-domenat mund të jenë për të hyrë në shërbimin e "email.shembull.com" për email ose "dyqan.shembull.com" për një dyqan në internet.

**Gjuhë ose Rajon**

Faqet e internetit të përdorin nën-domena për të dhënat e rajonit në gjuhë të ndryshme ose të ndryshme të përcaktuara, si "sq.shembull.com" për përmbajtjen në shqip.

**Kampanja e Marketingut**

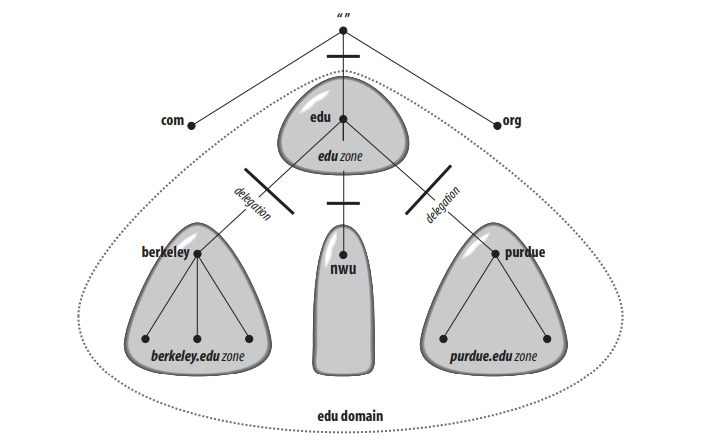
Nën-domenat mund të përdoren për fushata marketingu ose promovimi, si "verë.shitje.shembull.com".

Domain Name Server (DNS) - Sistemi i emrave të domenit është në thelb një bazë të dhënash e informacionit të hostit. Baza e të dhënave e shpërndarë e DNS indeksohet sipas emrave të domeneve. Çdo emër domain është në thelb vetëm një shteg në një pemë të madhe të përmbysur, e quajtur hapësira e emrave të domenit. Të struktura hierarkike e pemës, e paraqitur në figurën 1, është e ngjashme me strukturën e Sistemi i skedarëve Unix. Pema ka një rrënjë të vetme në krye. Në sistemin e skedarëve Unix, kjo është quhet adresa (direktoriumi) rrënjësore dhe përfaqësohet nga një prerje (/). DNS thjesht e quan atë "the rrënjë.” Ashtu si një sistem skedari, pema e DNS mund të degëzojë çdo numër mënyrash në çdo ndërthurje pika e veprimit, ose nyja. Thellësia e pemës është e kufizuar në 127 nivele (kufi që nuk jeni gjasa për të arritur).



FIG.1 Struktura e DNS name space

Shembull: Nje zonë përmban të gjithë domenët që përmbajnë domenë tjera brenda vetës me emër të njejtë, përveç nëse fushat e domenëve janë të deleguara. Për shembull, niveli i lartë domeni ***ca*** (për Kanadanë) ka nënfusha të quajtur ab.ca, on.ca dhe qc.ca, për provincat Alberta, Ontario dhe Quebec. Autoriteti për ab.ca, on.ca dhe qc.ca domenët mund t'u delegohen serverëve të emrave në çdo krahinë. Domeni ca përmban të gjitha të dhënat në ca plus të gjitha të dhënat në ab.ca, on.ca dhe qc.ca. Megjithatë, zona rreth përmban vetëm të dhënat në ca (shih Figurën 2-10), e cila ndoshta është kryesisht tregues nënfushat e deleguara. ab.ca, on.ca dhe qc.ca janë zona të ndara nga *ca* zonë.



Ndarja e Domenit të ***.edu*** në subdomenët e tij.

* Specifikimet e DNS përcaktojnë dy lloje të serverëve të emrave:

Primary Master dhe Secondary Master.

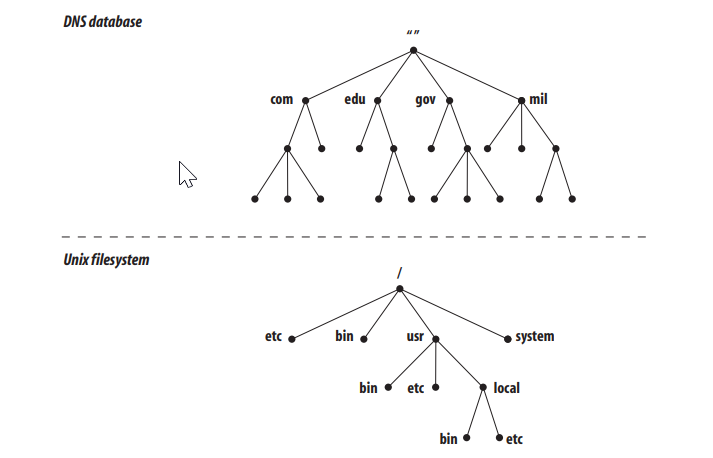


Fig.3. Dallimi në mes të Domain Name Server dhe Unix Sistemit.

Çdo domain ka një emër unik në bazë të cilit identifikohet, si çdo directory. Emri i domenit të një domeni identifikon pozicionin e tij në bazën e të dhënave, ashtu si emri absolut i rrugës së një directory specifikon vendin e tij në sistemin e filesystem. Në DNS, emri i domenit është sekuenca e etiketave nga nyja në rrënjën e domenit deri të rrënja e të gjithë pemës, me pika (.) duke ndarë etiketat.

Në file sistemin e Unix, emri absolut i rrugës së një drejtorie është lista e emrave të afërm të lexuar nga rrënja në fletë (drejtimi i kundërt nga DNS, si treguar në figurën 3), duke përdorur një të pjerrët për të ndarë emrat.

## Domenët e nivelit të lartë (Top Level Domain )

Domenët origjinale të nivelit të lartë ndanë organizatën e hapësirës së emrave të domenit të Internëtit:

* ***.com***
* ***.edu***
* ***.gov***
* ***.mil***
* ***.org***
* ***.int***
* ***.net***

*Organizatat tregtare:*

* Hewlett-Packard (hp.com),
* Sun Microsystems (sun.com)
* IBM (ibm.com).

*Organizatat arsimore:*

* Universiteti i Prishtinës “Hasan Prishtina” (uni-pr.edu),
* Oxford University (oxford.edu).

*Organizatat qeveritare*

* NASA (nasa.gov)
* National Science,
* SHBA - state.gov/

Pse ta përdorim Domain Name Server?

* Përditshmëri e gjërë në mbarë botën virtuale.
* Nëse përdorim TCP/IP – based internet
* Nëse përdorim Local Area Network apo Site Network që janë të lidhur në WAN apo MAN.

# Subdomenët dhe funksionaliteti i tyre

Nëndomenët janë një metode dhe praktike e shkëlqyeshme për të organizuar dhe strukturuar përmbajtjen, shërbimet ose funksionalitetin e një faqe interneti brenda hierarkisë më të madhe të emrave të domenit (DNS). Ato shërbejnë për qëllime të ndryshme dhe ofrojnë përfitime që mund të përmirësojnë përdorshmërinë, menaxhimin dhe funksionalitetin e një faqe interneti. Këtu janë disa arsye për të përdorur nëndomain dhe funksionet e tyre kryesore:

**Organizimi dhe hierarkia:** Nëndomenet ofrojnë një mënyrë të qartë dhe logjike për të organizuar seksione ose aspekte të ndryshme të një faqe interneti. Për shembull, një blog mund të strehohet në një nëndomain "blog.example.com", ndërsa një dyqan i tregtisë elektronike mund të jetë në një nëndomain "shop.example.com". Ky organizim hierarkik mund ta bëjë më të lehtë për përdoruesit të gjejnë përmbajtje specifike.

**Ndarja e përmbajtjes:** Nëndomenët ju lejojnë të mbani lloje të ndryshme të përmbajtjes të ndara nga njëri-tjetri. Kjo mund të jetë e dobishme për të shmangur konfliktet midis pjesëve të ndryshme të faqes suaj të internetit ose për izolimin e disa shërbimeve për arsye sigurie.

**Përmbajtje shumëgjuhëshe ose rajonale:** Nëse faqja juaj e internetit u përshtatet gjuhëve ose rajoneve të ndryshme, mund të përdorni nënfushat për të pritur përmbajtje specifike për secilën gjuhë ose rajon. Për shembull, "en.example.com" mund të jetë për përmbajtje në anglisht, ndërsa "es.example.com" mund të jetë për përmbajtje spanjolle.

**Segmentimi i funksionalitetit:** Nëndomenët mund të përdoren për të ndarë funksione të ndryshme të një faqe interneti, të tilla si API-të e pritjes ("api.example.com"), asetet statike ("assets.example.com") ose llogaritë e përdoruesve ("account.example". com").

**Testimi dhe zhvillimi:** Nëndomenët mund të përdoren për testimin e veçorive, modeleve ose zhvillimeve të reja pa ndikuar në faqen kryesore të internetit. Kjo lejon një mjedis të veçantë për të rregulluar ndryshimet përpara se t'i vendosë ato në domenin kryesor.

**Optimizimi i performancës:** Duke shpërndarë përmbajtje nëpër nëndomanë të ndryshëm, mund të përfitoni nga ngarkimi paralel, i cili mund të përmirësojë performancën e faqes në internet. Për shembull, imazhet mund të strehohen në një nëndomain të dedikuar për përmbajtjen statike, duke çuar në kohë më të shpejta të ngarkimit.

**Shërbimet e palëve të treta:** Nëndomenët mund të përdoren për të integruar shërbime ose platforma të palëve të treta. Për shembull, një nëndomain "login.example.com" mund të trajtojë vërtetimin përmes një shërbimi të palës së tretë.

# Protokolet e përdorura në Domen, SSL certifikata

DNS (Domain Name System): Protokoli DNS përdoret për të konvertuar emrat e domainëve në adresën e IP-së. Ky protokol është themeli i emrit dhe adresimit në internet dhe lejon qasje në burime duke u lidhur me emrat e lexueshëm nga njerëzit.

**HTTP (Hypertext Transfer Protocol)**

Protokoli HTTP përdoret për të shkëmbyer të dhëna ndërmjet klientëve dhe serverëve të uebit. Kjo është baza e komunikimit midis shfletuesve dhe faqeve të internetit.

**HTTPS (Hypertext Transfer Protocol Secure)**

HTTPS është një version i sigurt për protokollin HTTP, i cili përmban të dhëna midis klientit dhe serverit. Certifikatat e Përdoruesve SSL/TLS për të siguruar komunikimin dhe për të vërtetuar autenticitetin e faqeve.

**SMTP (Simple Mail Transfer Protocol)**

Protokoli SMTP përdoret për dërgimin e postës elektronike (email). Ai lidh serverët e emailit për të dërguar dhe pranuar mesazhe elektronike.

**IMAP (Internet Message Access Protocol)** **dhe POP3 (Post Office Protocol)**

Këto janë protokolle për të pranuar dhe lexuar postën elektronike nga serverët e emailit. IMAP lejon të mbajmë mesazhe në server, ndërsa POP3 shkarkon në kompjuterin e përdoruesit.

**Certifikata SSL/TLS:**

**Certifikatat SSL (Secure Sockets Layer) dhe TLS (Transport Layer Security)**

SSL dha TLS janë çertifikata që janë të sigurta për komunikim në internet, përdorim HTTPS. Këto certifikata duhet për të bërë enkriptimin e të dhënave ndërmjet klientit dhe serverit, duke bërë komunikimin e sigurt. Disa komponentë të certifikatës SSL/TLS janë:

* **Emri i Domainit (Domain Name)**

Çertifikatat emrin e domainit për të cilin janë lëshuar. Kjo që çertifikata të jetë vetëm për domainin e caktuar.

* **Çelësi Publik (Çelësi Publik)**

Çelësi publik përdoret për të enkriptuar të dhënat. Ky është çelësi publik mund të ndahet me të gjithë.

* **Çelësi Privat (Çelësi Privat)**

Çelësi privat përdoret për të dekriptuar të dhënat që janë të shkruara me çelësin publik. Ky çelës është konfidencial dhe nuk duhet të ndajë me të tjerët.

**Autoriteti i Çertifikatave (Autoriteti i Çertifikatës - CA)**

AK është një dokument që vërteton identitetin e pronarit të domenit dhe lëshimit të çertifikatës SSL/TLS. Shfrytzuesit e internetit ndajnë në CA për të konfirmuar autenticitetin e çertifikatave.

Çertifikatat SSL/TLS nuk janë të sigurta të sigurta në faqet e internetit dhe të mbrojnë të dhënat e përdoruesve nga skedarët e paligjshëm.

# Gjuha Programuese Python

Python është një gjuhë programimi e nivelit të lartë, e gjithanshme dhe e përdorur gjerësisht e njohur për thjeshtësinë dhe lejueshmërinë e saj. Karakteristikat dhe karakteristikat kryesore:

**Lexueshmëria:** Sintaksa e Python është krijuar për të qenë e qartë dhe e lehtë për t'u lexuar, gjë që e bën atë një zgjedhje të shkëlqyer si për fillestarët ashtu edhe për programuesit me përvojë.

**Shkathtësia:** Python mund të përdoret për lloje të ndryshme programimi, duke përfshirë zhvillimin e uebit, analizën e të dhënave, llogaritjen shkencore, mësimin e makinerive, inteligjencën artificiale, automatizimin dhe më shumë.

**Librari te gjerë:** Python vjen me një koleksion të pasur modulesh dhe Librarish të integruara, gjë që redukton ndjeshëm nevojën për të shkruar kodin nga e para për detyrat e zakonshme.

**Cross-Platform:** Python është i disponueshëm në platforma të ndryshme, duke përfshirë Windows, macOS dhe shpërndarje të ndryshme Linux.

## Paketa PIP3 ne Python 3

Menaxhimi i paketave: pip3 përdoret për të menaxhuar paketat në Python, duke e bërë të lehtë instalimin dhe përditësimin e Librarive, mjeteve dhe kornizave (framework) të ndryshme.

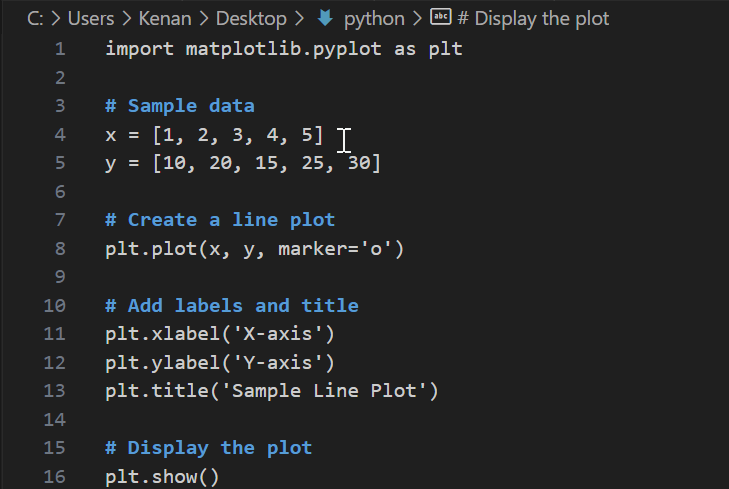
Versioni Python: pip3 është krijuar posaçërisht për të punuar me Python 3. Nëse jeni duke përdorur Python 2, mjeti ekuivalent quhet pip.

Komanda në Python për qëllime të ndryshme PIP3:

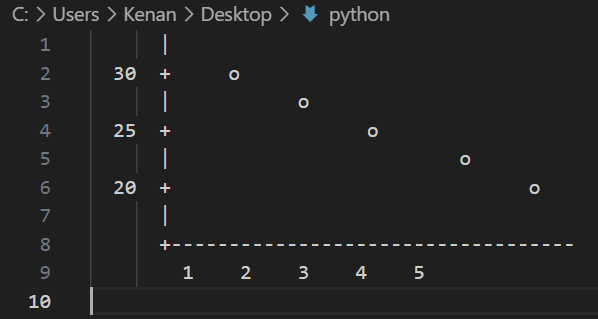
* **pip3 install numpy**
* **pip3 install package\_name**: Instalon një paketë.
* **pip3 uninstall package\_name:** Çinstalon një paketë.
* **Pip3 freeze:** Liston paketat e instaluara së bashku me versionet e tyre.
* **pip3 search query:** Kërkon për paketa në PyPI.
* **pip3 show package\_name**: Ofron informacion për një paketë specifike të instaluar.

Shembull: Përdorimi i librarisë PIP3 me paketën **matplotlib**

**Se pari e instalojmë paketën duke përdorur komandën pip3 instal matplotlib, dhe pastaj e shkruajmë kodin:**



*Kodi do të ekzekutohet si në vijim:*



# Gjuha Programuese HTML dhe CSS:

HTML është gjuha standarde e shënjimit e përdorur për të krijuar dhe strukturuar përmbajtjen në ueb. Ai siguron bazën për ndërtimin e faqeve në internet dhe përcakton strukturën dhe paraqitjen e përmbajtjes. HTML përdor etiketat për të përcaktuar lloje të ndryshme elementesh brenda një faqe interneti, të tilla si titujt, paragrafët, listat, imazhet, lidhjet, formularët dhe më shumë.

***Pikat kryesore rreth HTML:***

* **Struktura**

HTML organizon përmbajtjen duke përdorur një strukturë hierarkike të elementeve, duke përfshirë hapjen dhe mbylljen e etiketave.

* **Elementet**

Elementet HTML përbëhen nga etiketa dhe përmbajtje. Etiketat përcaktojnë llojin e elementit, dhe përmbajtja është të dhënat brenda elementit.

* **Atributet**

Elementet mund të kenë atribute që ofrojnë informacion shtesë rreth elementit, si p.sh. përcaktimi i burimeve të imazhit ose përcaktimi i destinacionëve të lidhjeve.

* **HTML semantike**

HTML5 prezantoi elemente semantike (p.sh., `<header>`, `<nav>`, `<main>`, `<footer>`) që tregojnë kuptimin dhe rolin e përmbajtjes që ata përmbajnë.

* **Hyperlinks** - HTML mundëson krijimin e hiperlidhjeve për të lidhur faqet e internetit dhe burimet së bashku. Lidhjen e me një web apo link tjetër,
* **Format**

Format HTML lejojnë hyrjen e përdoruesit dhe dërgimin e të dhënave në serverë. Elementet e formularit përfshijnë fushat e tekstit, butonat e radios, kutitë e kontrollit dhe butonat.

## CSS (Cascading Style Sheets)

CSS është një gjuhë e cila përshkruan stilin që përdoret për të kontrolluar pamjen vizuale dhe paraqitjen e elementeve HTML në një faqe interneti. Ai ndan përmbajtjen (HTML) nga prezantimi (CSS), duke i lejuar zhvilluesit të aplikojnë stile të qëndrueshme nëpër faqe dhe pajisje të shumta ueb. Së bashku, HTML dhe CSS formojnë shtyllën kurrizore të zhvillimit të uebit, me HTML që përcakton strukturën dhe përmbajtjen e faqeve të internetit dhe CSS që kontrollon prezantimin dhe paraqitjen e tyre vizuale. Ata punojnë së bashku për të krijuar faqe interneti tërheqëse dhe të dizajnuara mirë. Ndër karakteristikat kryesore të CSS janë të renditura poshtë:

* **Selektoret apo Zgjeruesit (Selectors)**

Përzgjedhësit CSS synojnë elementet HTML për të aplikuar stile. Përzgjedhësit mund të synojnë elementë sipas emrit të etiketës, klasës, ID-së ose atributeve të tjera.

* **Vetitë (Properties) dhe vlerat**

Vetitë CSS përcaktojnë aspektet e një elementi që do të ketë stil, si ngjyra, madhësia e shkronjave, diferenca, mbushja dhe më shumë. Vlerat vendosin stilin specifik për secilën pronë.

* **Cascading**

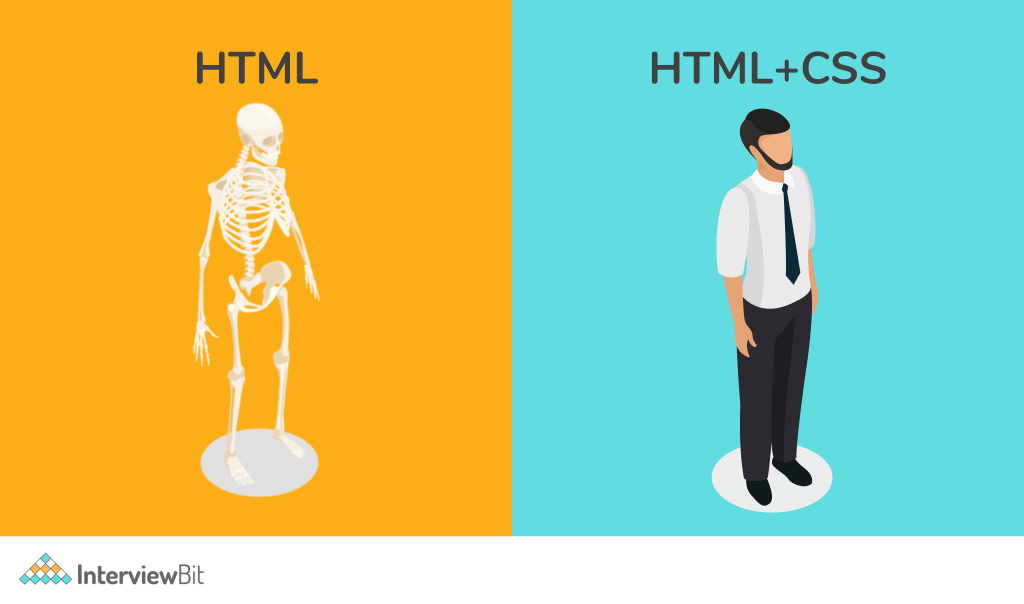
Natyra "kaskaduese" e CSS i referohet mënyrës sesi stilet aplikohen në një hierarki, me stile nga burime të ndryshme (përdorues, autor, parazgjedhje) që ndërveprojnë bazuar në specifikat dhe trashëgiminë.

* **Zgjedhuesit dhe trashëgimia**

Vetitë CSS mund të trashëgohen nga elementët prindër tek fëmijët e tyre. Disa prona trashëgohen si parazgjedhje, ndërsa të tjerat jo.

* **Dizajni i përgjegjshëm**

CSS i lejon zhvilluesit të krijojnë paraqitje të përgjegjshme që përshtaten me madhësi dhe pajisje të ndryshme të ekranit, duke përmirësuar përvojën e përdoruesit.

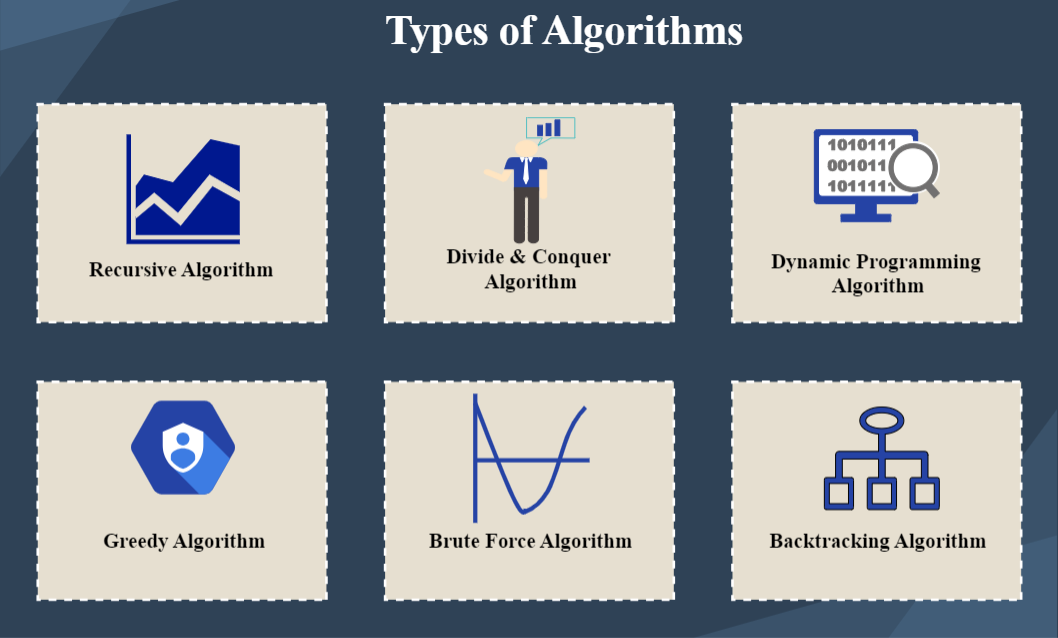


# Algoritmet

Kur shkruajmë një program kompjuterik, në përgjithësi po zbatojmë një metodë që është shpikur më parë për të zgjidhur një problem. Kjo metodë është shpesh pavarësisht nga gjuha e caktuar e programimit që përdoret - ka të ngjarë të jetë po aq i përshtatshëm për shumë kompjuterë dhe shumë gjuhë programimi. Është metodë, në vend të vetë programit kompjuterik, që specifikon hapat që ne mundemi marrë për të zgjidhur problemin. Termi algoritëm përdoret në shkencën kompjuterike për të përshkruar një metodë e fundme, përcaktuese dhe efektive e zgjidhjes së problemeve, e përshtatshme për zbatim si një program kompjuterik. Algoritmet janë lëndë e shkencës kompjuterike: ato janë objektet qendrore të studimit në terren. Ne mund të përcaktojmë një algoritëm duke përshkruar një procedurë për zgjidhjen e një problemi në a gjuhë natyrore, ose duke shkruar një program kompjuterik që zbaton procedurën.

Termi "algoritëm" rrjedh nga emri i matematikanit dhe studiuesit persian *Muhamed ibn Musa al-Khwarizmi*, i cili jetoi gjatë shekullit të 9-të. Al-Khwarizmi dha një kontribut të rëndësishëm në matematikë dhe prezantoi konceptin e zgjidhjes së ekuacioneve lineare dhe kuadratike, të cilat hodhën bazat për algjebër dhe të menduarit algoritmik.

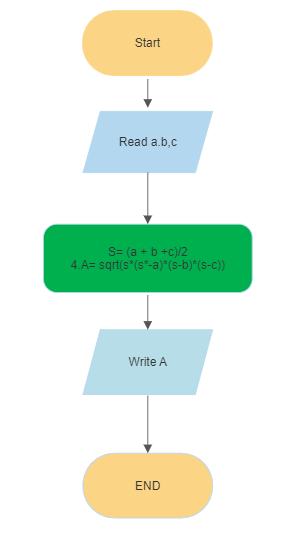
Disa lloje të algoritmeve janë të përshkruara në figurën e mëposhtëme.



Algoritmet mund të paraqiten edhe në formë të flowchart apo diagrameve, kjo metode është e përdorur dhe thjeshton tejmase zbërthimin e algoritmit. Prandaj për ta kuptuar më mirë do të marrim një shembull me Algorimin e përshkruam në formë të thjeshtë, diagramit dhe kodit.

Nëse e marrim një shembull nga jeta e përditshme, gjatë përdorimit të rrjeteve sociale, përdoruesit dhe platforma që në këtë rast është rrjeti social krijon një interakcion. Ky interaksion bëhet me vetëdijen tonë por shumë shpesh edhe pa vetëdijen tonë. Si përdorues të zakonshëm e “ushqejmë” me informata të shumta platformën e rrjeteve sociale, e cila për qëllime marketingu apo joshje për zgjatje të përdorimit të platformës së vet, përdor algoritme të ndryshme shumë komplekse ashtu që neve na merr vëmendjen duke na servuar me materiale që neve na ngjall një interesim dhe kureshtje për të vazhduar dhe zgjatur kohën e kaluar me përdorimin e platformës së rrjeteve sociale. Shpesh këto algoritme shumë komplekse dhe shumë të sofistikuar krijojnë një profilizim të përdoruesve të tyre ashtu që pastaj ato i përdor për qëllime të ndryshme marketingu. Algoritmet janë shtylla kryesore e zhvillimit të inteligjencës artificiale që sot për fat të mirë jemi duke e përjetuar një kohë të shkëlqimit të saj.

***Algorithm në flowchart diagram***



1. Start
2. Read a,b,c
3. S= (a + b +c)/2
4. A= sqrt(s\*(s\*-a)\*(s-b)\*(s-c))
5. Print or Disply A
6. Stop

Kodi:

#include<iostream>

#include <cmath>

Unsing namespace std;

Int main ()

{

Int a,b,c;

Double s, A;

Count << “Enter values of a, b, c” <<endl;

cin >> a>>b>>c;

s= (a+b+c)/2;

A= sqrt (s\*(s-a)\*(s-b)\*(s-c));

Cout << “Area of triangle is:”<<A;

Return 0;

}

# Brute Force Algoritmet

Algoritmi i Brute Force është një qasje e drejtpërdrejtë dhe shteruese për zgjidhjen e problemeve që përfshin përpjekjen e të gjitha zgjidhjeve të mundshme për të gjetur atë që plotëson një kriter të caktuar. Megjithëse nuk është metoda më efikase, ajo garanton një zgjidhje të saktë duke eksploruar sistematikisht çdo opsion të mundshëm.

## Përshkrimi algoritmik:

Algoritmi i forcës brutale përfshin përsëritjen e të gjitha zgjidhjeve të mundshme dhe testimin e secilës për të përcaktuar nëse e zgjidh problemin. Është njësoj si të kërkosh për një gjilpërë në një kashtë duke ekzaminuar çdo pjesë të barit.

**Raste të perdorimit:**

Algoritmet e forcës brutale përdoren zakonisht në skenarë ku metodat e tjera më të optimizuara janë jopraktike ose të panevojshme.

Këtu janë disa raste përdorimi:

* **Thyerja e fjalëkalimit**

Forca brutale mund të përdoret për të thyer fjalëkalimet duke provuar të gjitha kombinimet e mundshme derisa të gjendet e duhura.

* **Kërkim shterues**

Kur kërkoni për një artikull ose zgjidhje specifike me një hapësirë të kufizuar kërkimi, si p.sh. zgjidhja e një enigme ose gjetja e një sekuence lëvizjesh në një lojë.

* **Thyerja e enkriptimit**

Në rastet kur metodat e kriptimit kanë dobësi, forca brutale mund të përdoret për të provuar të gjithë çelësat e mundshëm të deshifrimit.

* **Probleme kombinuese**

Disa probleme në matematikë dhe shkenca kompjuterike, mund të zgjidhen duke përdorur forcën brutale për madhësi të vogla të dhënash.

# Elaborimi matematikor:

Le të marrim një problem matematikor për të ilustruar qasjen e forcës brutale: gjetja e vlerës minimale të një funksioni.

Problem: Gjeni vlerën minimale të funksionit: ***f(x) = x2 - 4x + 4.***

Qasja e Brute Force algoritmit:

Zgjidhni një domen për x, p.sh., ***-10 deri në 10.***

Llogaritni ***f(x)*** për çdo ***x*** në interval.

Mbani gjurmët e ***f(x)*** më të vogël të hasur.

***f(x)*** më e vogël i përgjigjet vlerës minimale të funksionit.

***Shpjegimi matematikor:***

Në këtë rast, algoritmi i forcës brutale vlerëson funksionin për çdo vlerë të mundshme të x në diapazonin e dhënë. Duke provuar sistematikisht të gjitha vlerat, siguron që asnjë minimum potencial të mos humbasë. Megjithatë, kjo qasje mund të marrë kohë dhe mund të thuhet se është joefikase për funksione më komplekse ose diapazon dhe domen më të madh.

## Avantazhet dhe disavantazhet:

***Përparësitë:***

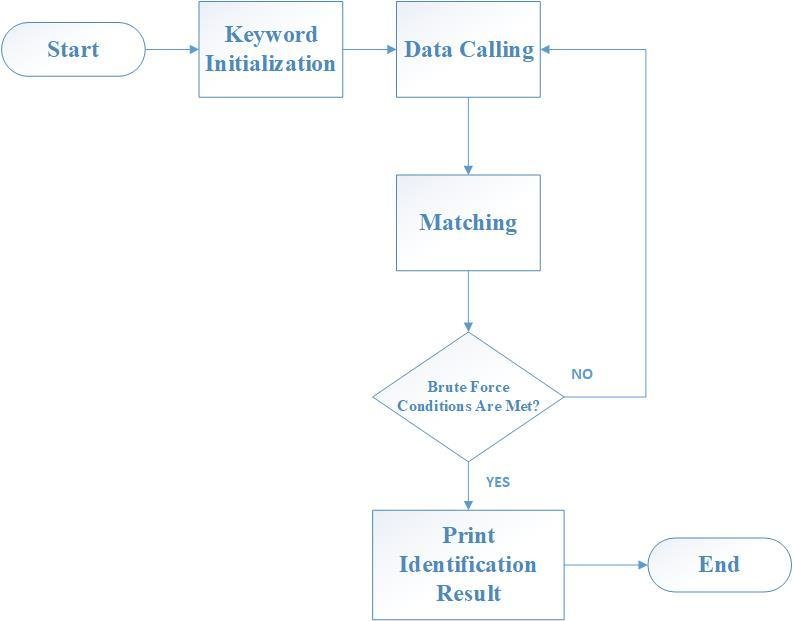
* Garanton një zgjidhje të saktë për kërkesën konkrretë.
* E thjeshtë dhe e lehtë për t'u zbatuar.
* I përshtatshëm për madhësi të vogla të hyrjes ose skenarë ku optimizmi nuk është prioritet.

***Disavantazhet:***

* I paefektshëm për hapësira të mëdha kërkimi(Domene të mëdha të kërkimit).
* Konsumon kohë për probleme komplekse (Nuk i eliminon mundësitë sipas një logjike të caktuar por i provon të gjitha mundësitë duke kërkuar rezultatin e saktë gjithmonë).
* Jo i përshtatshëm për probleme me një numër të madh zgjidhjesh të mundshme.

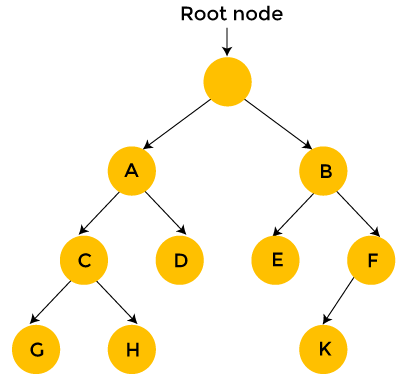
***Konkluzioni:***

Algoritmi i forcës brutale është një metodë e drejtpërdrejtë dhe e lehtë po ashtu edhe besueshme për zgjidhjen e problemeve duke provuar në mënyrë shteruese të gjitha zgjidhjet e mundshme. Ndërsa mund të mos jetë qasja më efikase në shumë raste, ajo shërben si një bazë për të kuptuar algoritme më të avancuara dhe optimizmit e tyre. Ka përdorim të gjërë në raste kur domenet e kërikimit janë të kufizuara, ashtu që të arrihet rezultati në një kohë më të shkurtër. Kurse në rastin në rastin kur domeni i kërkimit është i madh, atëherë preferohet që të përdorën algoritme tjera alternative të cilat e shkurtojnë kohën e kërkimit.



Flowchart diagrami i paraqitur më lartë përshkruan në mënyrë të thjeshtë funksionimin e Brute force algoritmit. Si fillim deklarohen dhe inicializohen variablat dhe pastaj thirren të dhënat e nevojshme apo mund ta quajmë dhe si domeni i veprimit te algorimit. Nëse ka rezultat të kërkuar sipas kushtit të dhënë të kërkuara nga ana e programit atëherë algoritmi vjen me një rezultat dhe programi merr fund, por nëse nuk gjindet rezultat me herën e parë, algoritmi do të hyj në loop apo në unazë që do të riekzekutoj kushtin duke marrë të dhëna nga domeni deri në momentin që do të arrij në rezultat.

Pasiqë Brute Force algoritmi është i bazuar në rezultat dhe gjithmonë na jep një rezultat të saktë pasiqë i provon të gjitha mundësitë ,por që në aspekt të vlerësimit të kohës brute force algoritmi mbetet i mangët në krahasim me algoritmet tjera.

Shembull: Supozoni se e kemi problemin në formën e pemës së treguar si më poshtë.

Kërkimi përmes Brute Force algoritmit do të merr në konsideratë çdo gjendje apo mundesi të një peme, dhe gjendja përfaqësohet në formën e një nyje. Për sa i përket pozicionit fillestar, ne kemi dy zgjedhje, Do të thotë që, një state (gjendje) A dhe një shtet B. Ne mund të gjenerojmë ose gjendjen A ose gjendjen B. Në rastin e gjendjes B, ne kemi dy gjendje, d.m.th., gjendjen E dhe F. Në rastin e kërkimit me Brute force algoritmit, çdo state (gjendje) konsiderohet një nga një. Siç mund të vërejmë në pemën e mësipërme se kërkimi i forcës brutale kërkon 12 hapa për të gjetur zgjidhjen.

Nga ana tjetër, backtracking, i cili përdor kërkimin Depth - First, merr parasysh gjendjet e mëposhtme vetëm kur gjenja ofron një zgjidhje të mundshme. Merrni parasysh pemën e mësipërme, filloni nga nyja rrënjë, pastaj kaloni të nyja A dhe më pas nyja C. Nëse nyja C nuk ofron zgjidhjen e realizueshme, atëherë nuk ka kuptim të merren parasysh gjendjet G dhe H. Ne kthehemi nga nyja C në nyja A. Më pas kalojmë nga nyja A në nyjën D. Meqenëse nyja D nuk ofron zgjidhjen e mundshme, ne e hedhim poshtë këtë gjendje dhe kthehemi nga nyja D në nyjën A. Kalojmë në nyjen B, pastaj kalojmë nga nyja B në nyjën E. Kalojmë nga nyja E në nyjën K; Meqenëse **K është një zgjidhje**, kështu që duhen 10 hapa për të gjetur zgjidhjen. Në këtë mënyrë, ne eliminojmë një numër më të madh gjendjesh në një përsëritje të vetme. Prandaj, mund të themi se kthimi prapa është më i shpejtë dhe më efikas se qasja e forcës brutale.

# Algoritmet e Brute Force dhe Greedy: të përbashkëtat dhe dallimet

Algoritmet Brute force dhe Greedy janë dy qasje të dallueshme për zgjidhjen e problemeve në shkencën kompjuterike. Ndërsa ata kanë disa të përbashkëta, ato gjithashtu ndryshojnë në strategjitë dhe karakteristikat e tyre. Le të shqyrtojmë aspektet dhe dallimet e tyre të përbashkëta.

***Aspekte të përbashkëta:***

* **Kërkimi shterues**:

Të dy algoritmet e babëzitur dhe të forcës brutale përfshijnë eksplorimin e të gjitha zgjidhjeve ose opsioneve të mundshme brenda një hapësire të caktuar kërkimi.

* **Qëllimet e optimizmit**

Të dyja qasjet synojnë të gjejnë një zgjidhje optimale, pavarësisht nëse është zgjidhja më e mirë e mundshme ose një zgjidhje që plotëson kritere specifike.

* **Procesi përsëritës**

Të dyja qasjet zakonisht përfshijnë një proces përsëritës ku hapat ndërmerren në mënyrë sekuencave dhe çdo hap kontribuon në përsosjen e zgjidhjes.

***Dallimet:***

**Brute Force**

Algoritmet e forcës brutale eksplorojnë sistematikisht të gjitha zgjidhjet e mundshme, të cilat mund të jenë joefikase për madhësi më të mëdha të problemeve për shkak të natyrës së tyre shteruese.

**Greedy**

Algoritmet e babëzitur bëjnë zgjedhje lokale optimale në çdo hap, duke shpresuar se këto zgjedhje do të çojnë në një zgjidhje optimale globalisht. Ata marrin vendimin më të mirë në çdo hap pa marrë parasysh pasojat e ardhshme.

***Kompleksiteti:***

Brute Force shpesh kanë kompleksitet të lartë kohor dhe hapësinor, veçanërisht për problemet me hapësira të mëdha kërkimi apo domene me te mëdha.

Greedy algoritmet pritet të kenë kompleksitet më të ulët kohor sepse nuk marrin parasysh të gjitha mundësitë. Megjithatë, ata nuk mund të gjejnë gjithmonë zgjidhjen më të mirë.

***Optimaliteti:***

***Brute Force***

Garantojnë gjetjen e zgjidhjes optimale sepse eksplorojnë të gjitha mundësitë. Ato janë të besueshme, por mund të jenë jopraktike për probleme të mëdha.

***Greedy***

E gjejnë gjithmonë zgjidhjen optimale globalisht, sepse ato fokusohen në optimizmin lokal në çdo hap. Ato mund të çojnë në zgjidhje jo optimale në disa raste.

Aplikimet e dy algoritmeve janë të ngjashme por sa i përket Brute Force algoritmit përdoret shpesh kur hapësira e kërkimit është e vogël ose kur optimizimi nuk është kritik. Është i përshtatshëm për probleme ku numri i mundësive është i menaxhueshëm.

Greedy - përdoren kur një problem mund të ndahet në nënprobleme më të vogla dhe optimizimi lokal kontribuon në optimizimin global. Ato janë të zakonshme në problemet e optimizimit.

***Marrja e Vendimeve:***

***Brute Force -*** Ndjekin një proces sistematik të vlerësimit të të gjitha mundësive, pa marrë parasysh nëse një zgjedhje e veçantë është më e mirë se të tjerat.

***Greedy*** – Marrin vendime bazuar në përfitimin e menjëhershëm, pa marrë parasysh ndikimin e mundshëm afatgjatë në zgjidhje.

Shembull:

Për një shembull konkret, merrni parasysh *"Problemin e kthimit të kusurit me sasi sa më të pakët të kartëmonedhave apo monedhave metalike"* ku duhet të jepni një sasi të caktuar kusuri duke përdorur sa më pak monedha të mundshme. Një qasje e Brute Force do të përfshinte përpjekjen e të gjitha kombinimeve të monedhave për të gjetur më të mirën. Qasja tjetër e algoritmit Greedy mund të përfshijë gjithmonë zgjedhjen më të madhe të monedhës në dispozicion në çdo hap. Ndërsa qasja greedy algoritmit mund të funksionojë në disa raste, ajo mund të çojë në zgjidhje jo optimale, ndërsa qasja e forcës brutale garanton zgjidhjen më të mirë. Të dy algoritmet synojnë të gjejnë zgjidhje optimale, ato ndryshojnë në strategjitë, efikasitetin dhe optimalitetin e tyre. Algoritmet e pangopur bëjnë zgjedhje lokale optimale, ndërsa algoritmet e Brute Force kërkojnë në mënyrë shteruese të gjitha mundësitë. Zgjedhja midis të dyjave varet nga karakteristikat e problemit dhe kërkesat e optimizmit.

# Vegla për gjetjen e subdomeneve përmes Brute Foce algoritmit

Vegla për gjetjen e subdomenëve përmes përdorimet të algoritmit Brute Force është e shkruar në disa gjuhe programuese. Algoritmi i Brute Force algoritmit është i shkruar në gjuhen programuese Python duke i përdorur dhe libraritë shoqëruese relevante. Pjesa e kodit që përshkruan algoritmin është e lidhur me një front end të shkruar në gjuhën programues të HTML5 dhe CSS. Përmes Front end mundësohet qasje më e mirë vizuele ndaj programit dhe do të paraqiten rezultatet e gjeneruara përmes searchit. Të dhënat ngarkohen përmes butonit “Ngarko” me që rast aty do të ngarkohet një listë e siguruar nga interneti dhe kjo listë ngarkohet në back-end (PHP database). Pasi që të ngarkohet lista, shtypim butonin search dhe pas një kohe të caktuar programi do të gjeneroj një përgjigje apo ndryshe do t’i listoj të disa subdomene të një domeni përcaktues ashtu që të rezultatet do të shfaqen në front end dhe poashtu në back-end PHP database.

Sa i përket rezultateve do t’i mund të krijojmë edhe një mendim të përgjithshëm rreth domenit duke parë fushëveprimin dhe ndarjen e domenit në nën domene. Kjo është e nevojshme në shumë fusha të studimit të cilat i kemi potencuar më parë.

Për të startuar programin për ekzekutim duhet që të instalohet programi XAMPP.

XAMPP është një paketë softuerësh falas, me burim të hapur që ofron një mjedis zhvillimi lokal për aplikacionet në ueb. XAMPP përdoret për Web Local development dhe për testime të platformave që përdorin databazë dhe vegza tjera qe i përkrah XAMPP.

Akronimi "XAMPP" qëndron për:

X: I referohet ndonjë prej sistemeve operative (Windows, Linux, macOS).

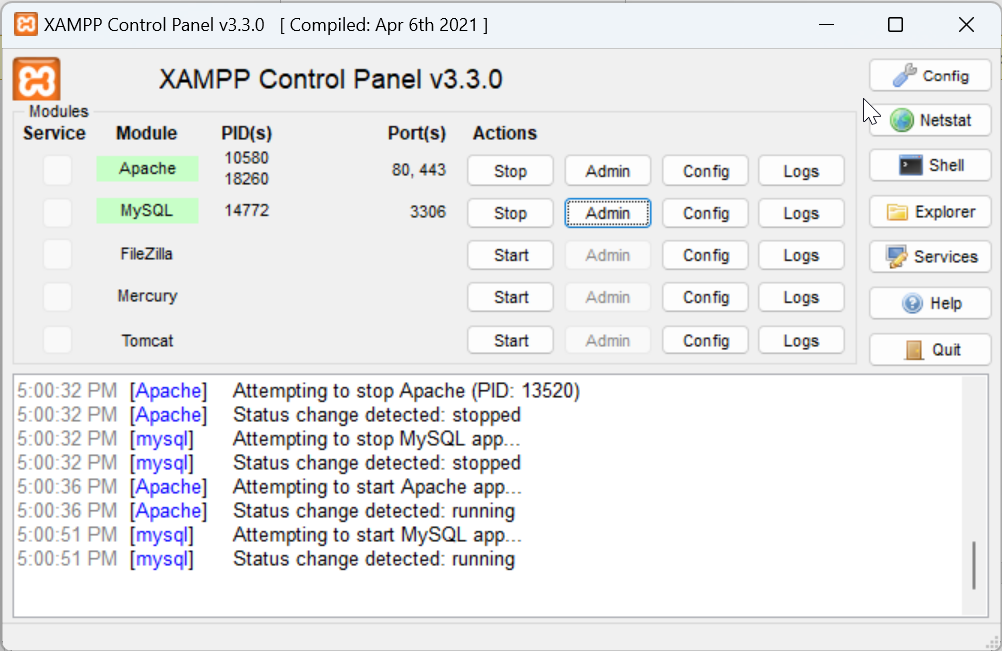
Përgjigje: Serveri HTTP Apache

M: MySQL ose MariaDB (sistemet e menaxhimit të bazës së të dhënave)

P: PHP

P: Perl

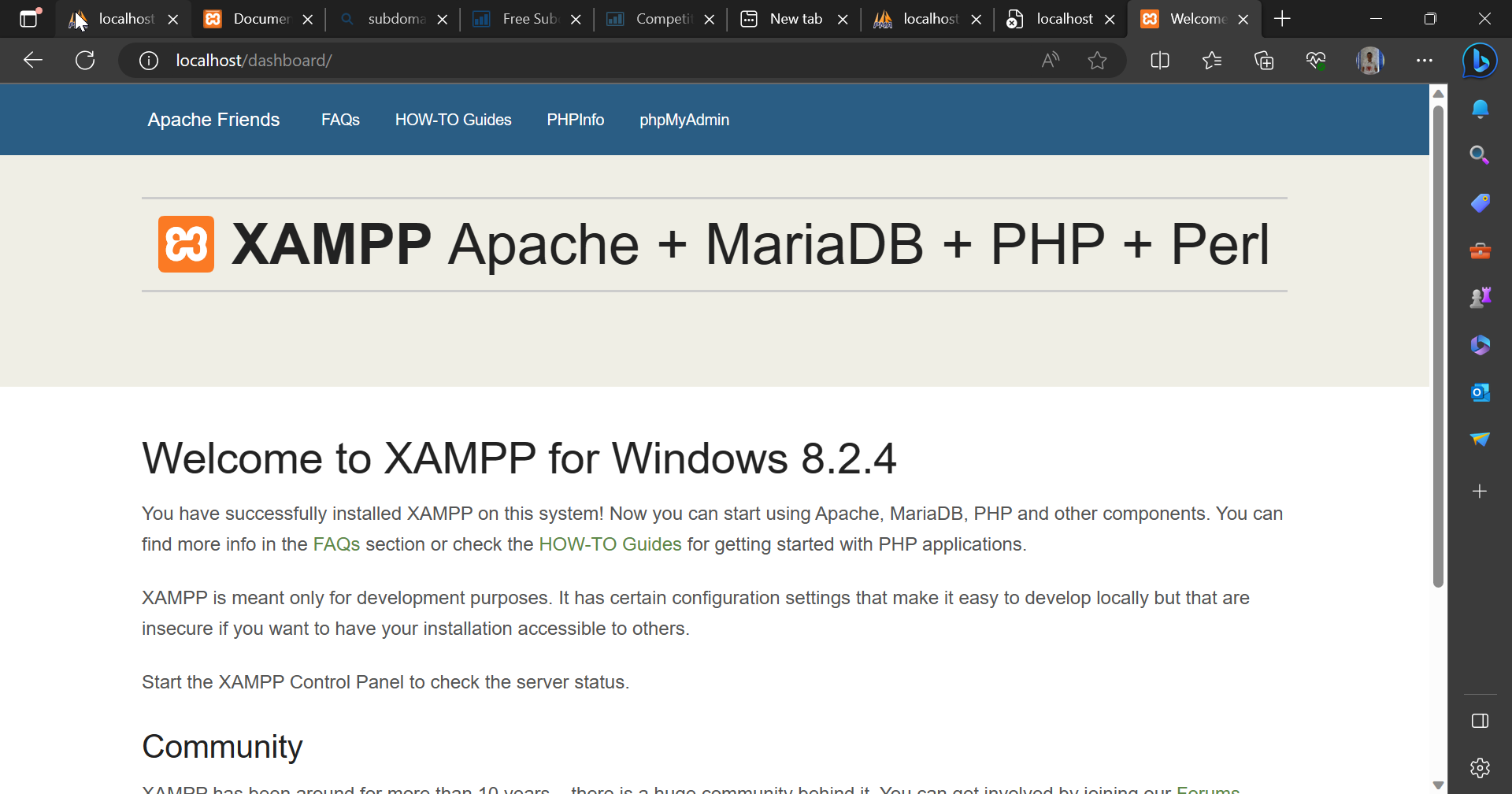
XAMPP i bashkon këta komponentë së bashku, duke e bërë të lehtë për zhvilluesit të konfigurojnë një mjedis server lokal për testimin dhe zhvillimin e aplikacioneve në ueb pa pasur nevojë të instalojnë secilin komponent veç e veç. XAMPP në vete përmban Apache HTTP Server, MySQL/ MairaDB, PHP, PHP myAdmin.



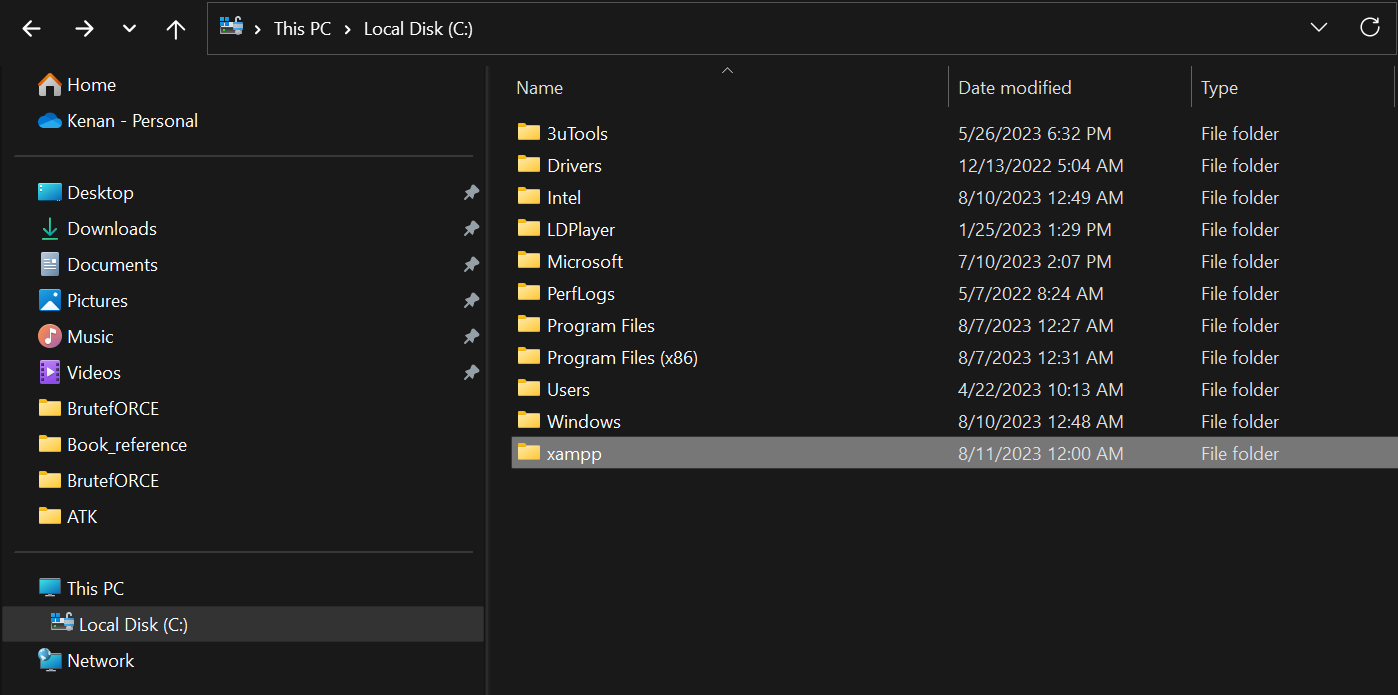


Pikë së pari duhet të startohet APACHE dhe pastaj duhet të startohet edhe MySQL.

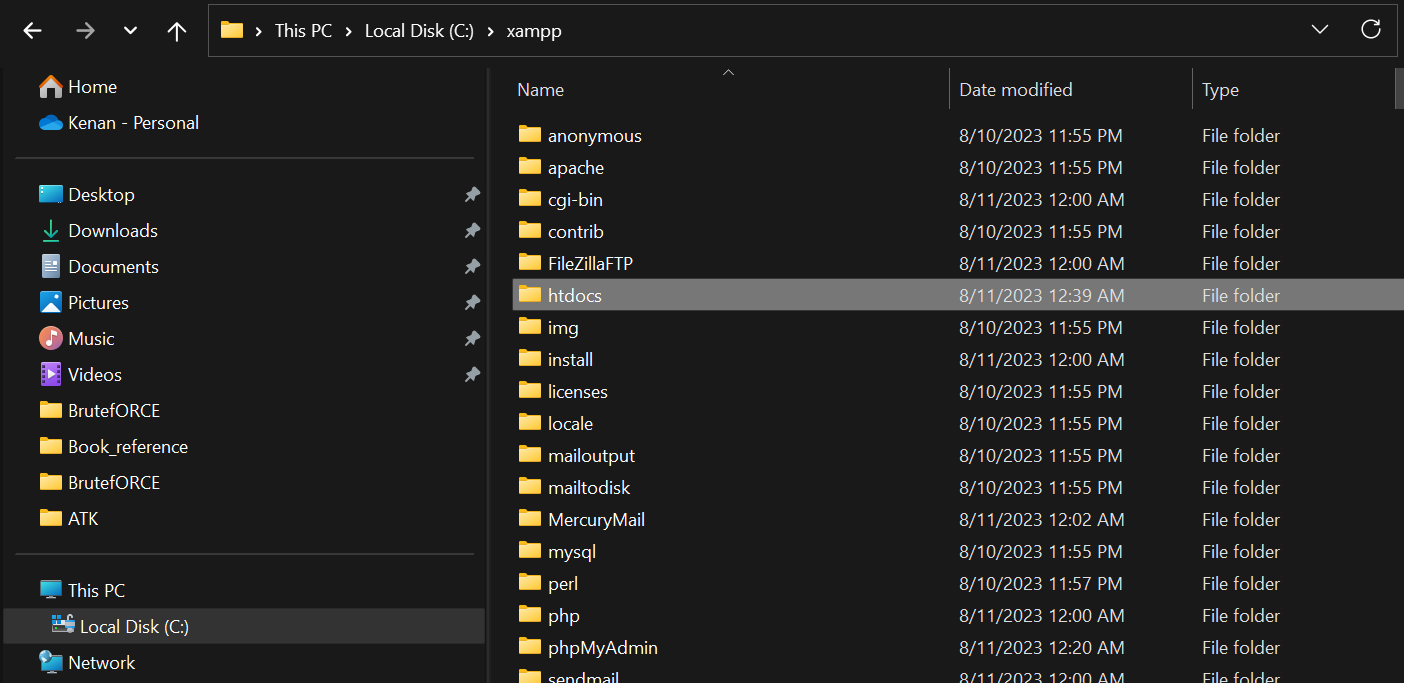
Pastaj duhet të startohet një Browser mund të përdoret Google Chrome apo Microsoft Edge, dhe në të kërkohet në pjesën e search bar “localhost”, në rast se është çdo gjë në rregull dhe programi është i konfiguruar ashtu siç duhet, në browser do të shfaqet, si me poshtë:



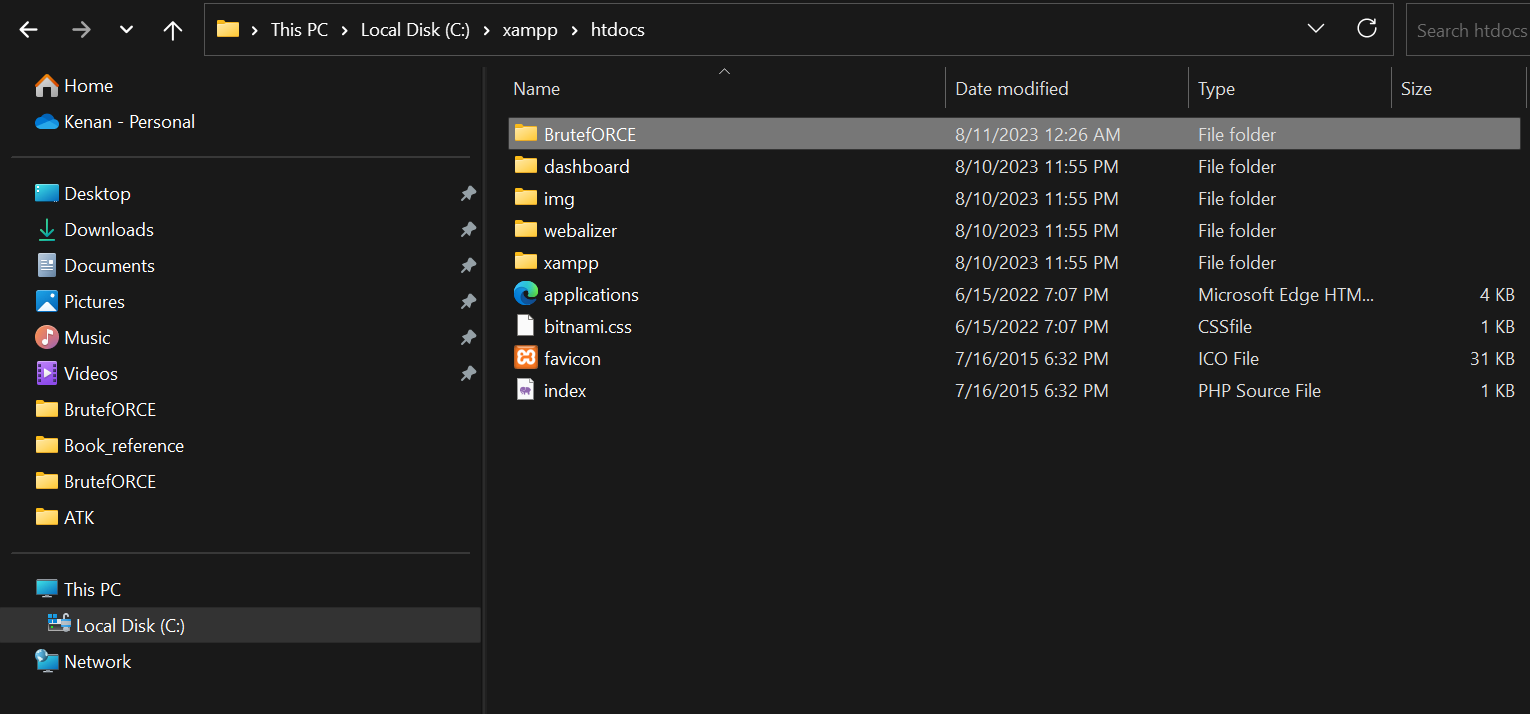
Hapi i mëtutjeshëm është që ta vendosim filet e programit në folderin burimor të XAMPP ashtu që të thirret programi dhe të vazhdohet me inicializim të mëtutjeshëm. Prandaj i ndjekim hapat të përshkruar si më poshtë:



Drejtohuni në “C” apo në “local disk”-un në të cilin e keni të instaluar programin XAMPP. Pastaj duhet të zgjidhet folderi me emrin “xampp”.



Selektoni folderin “htdoocs”, dhe vendosni tek fajllat e programit të bashkuara në një folder të vetëm.



Duhet të kopjohet emri i folderit të projektit, që në këtë rast është ***“BrutefORCE”.***

Pastaj vazhdojmë me search bar të browserit. ***“localhost/BrutefORCE”*** dhe do të gjenerohet front end i vegëzes i lidhur me databazë.

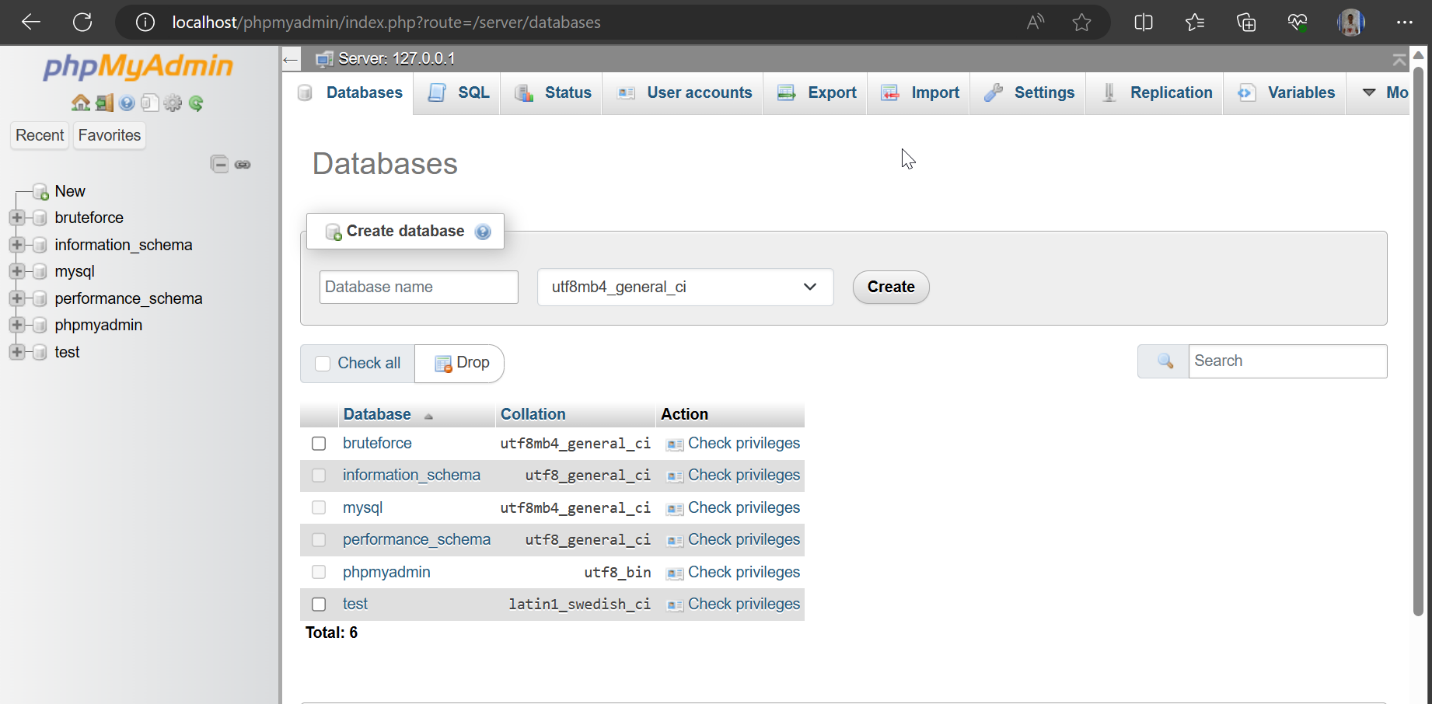


Interface për përdorimin e vegezes është paraqitur në pamjen më sipër. Në momentin kur shtypet butoni “Upload” uploadohet file me lista të subdomeneve, rrjedhimisht ngarkohet në databazë. Në pjesën e search mund t’i kërkojmë subdomenët të cilat janë pjesë e listës së ngarkuar në databazë.

 Search bar për Kerim te subdomenave përkatës.

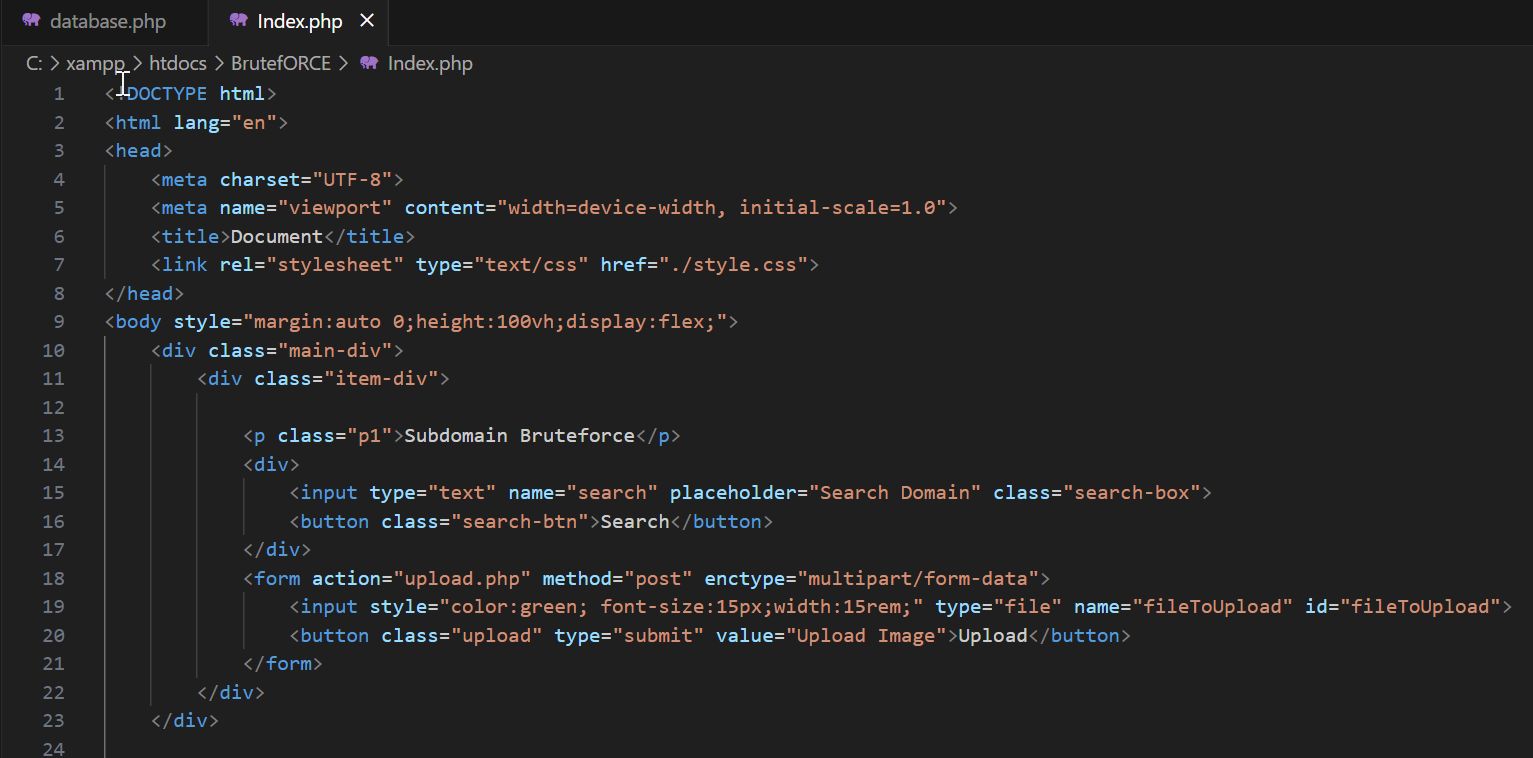
 Butoni për ngarkim të fajllit, që do të përdoret për analizim.

 Ekzekutimi i komandës për ngarkim të fajllit.



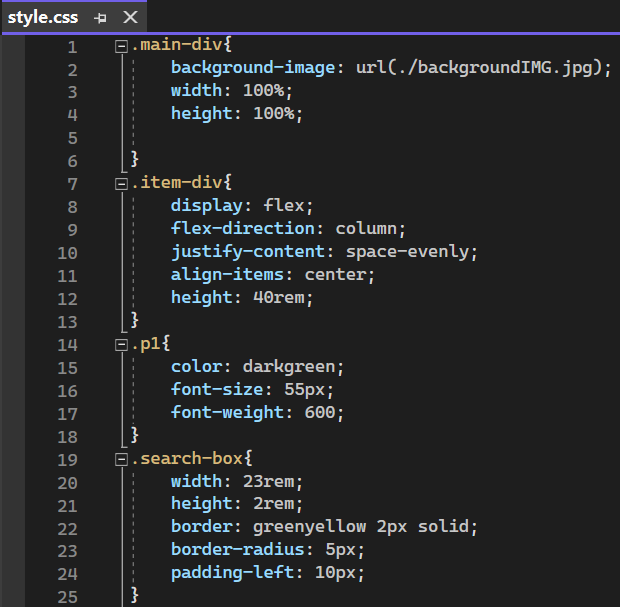
Pamje nga “PHPMyAdmin”

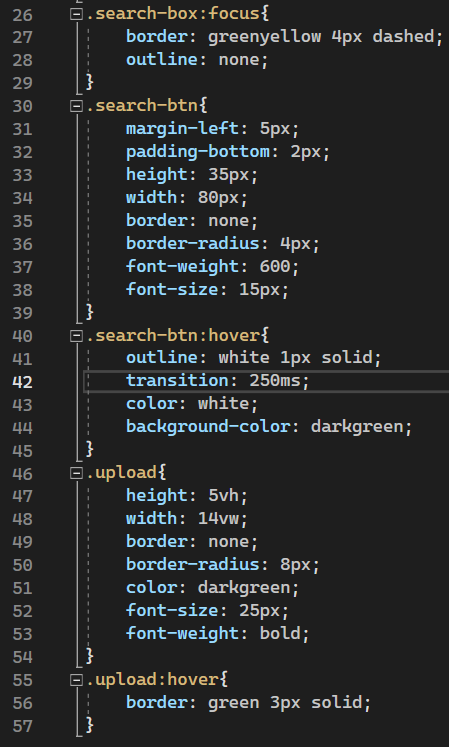
Pjesa e kodit e shkruar në gjuhën programuese HTML 5



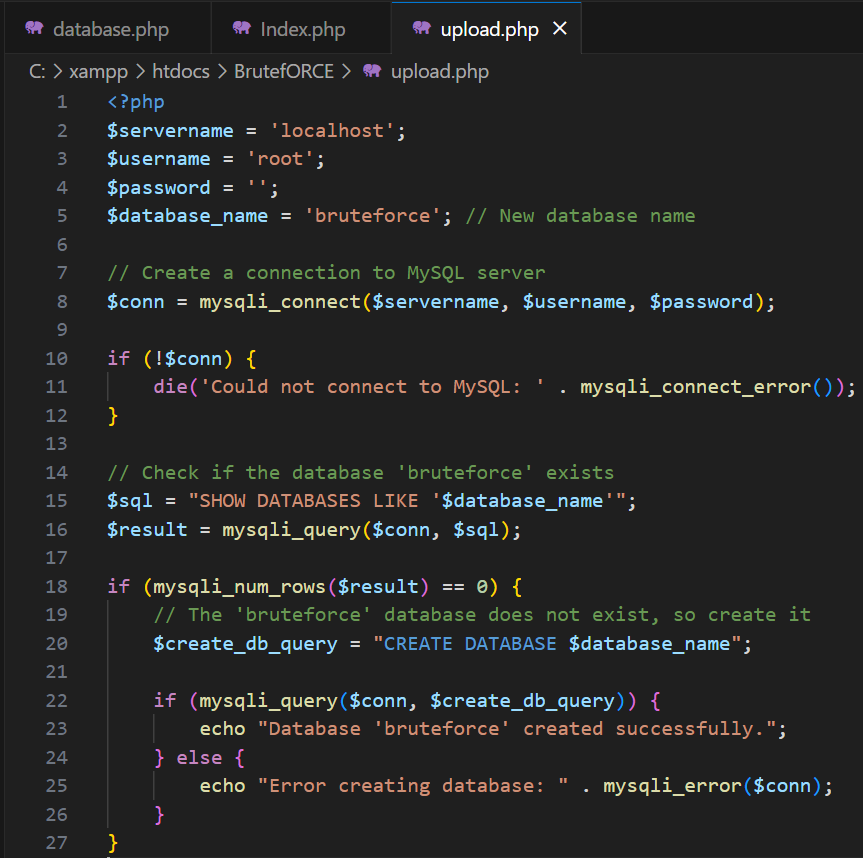
das

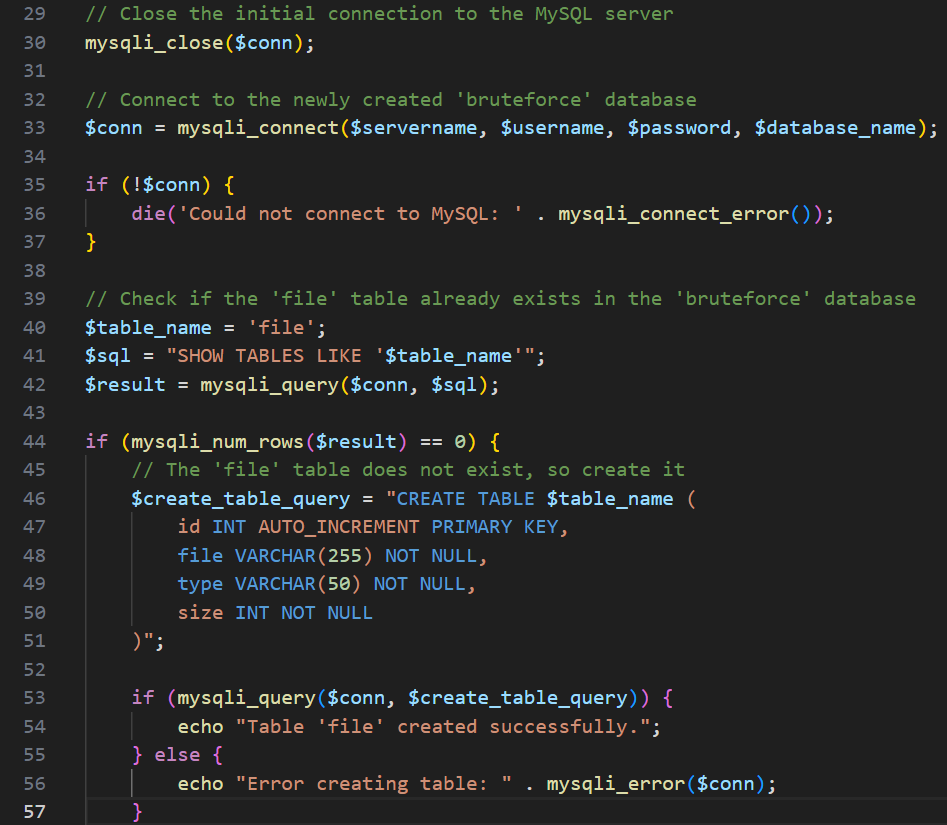
**Kodi dhe parametrat në CSS.**

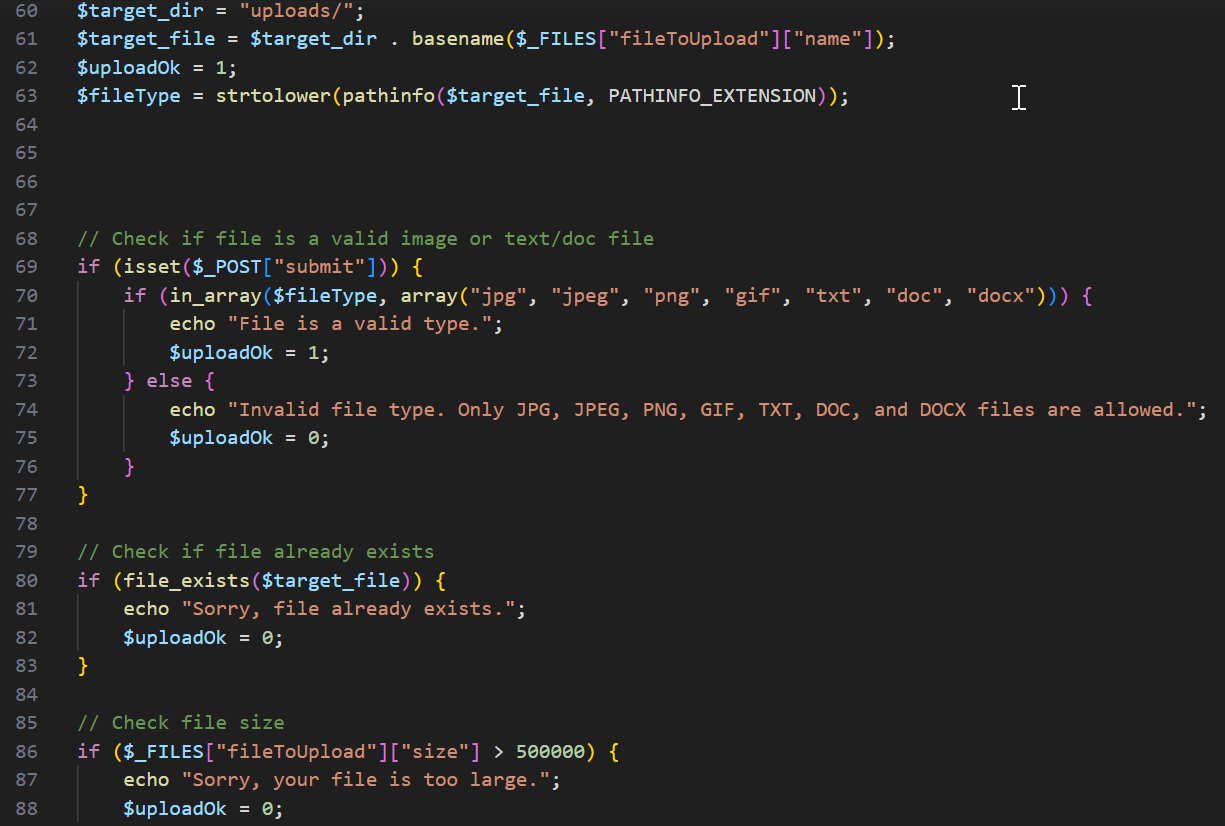


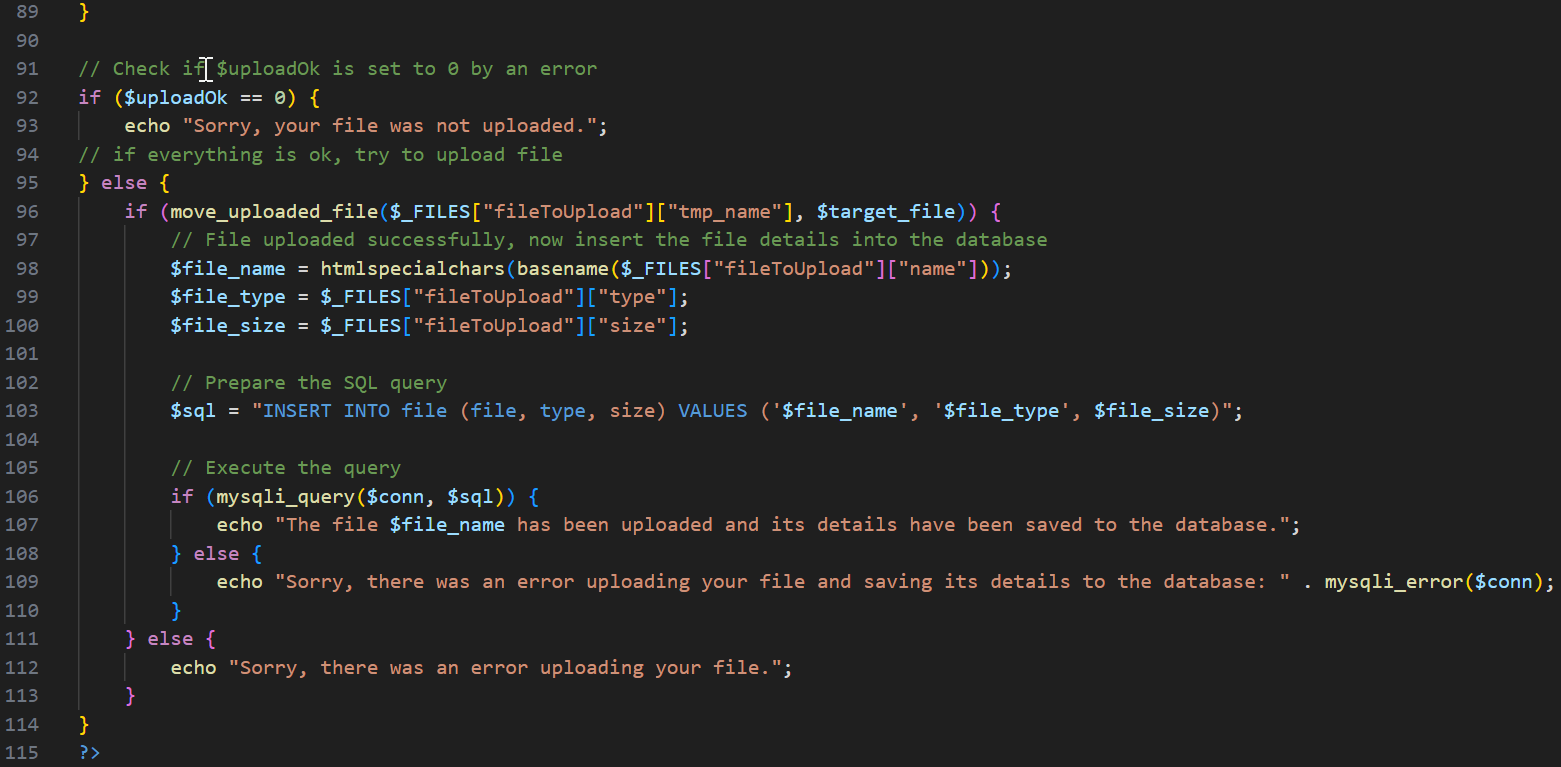


**Kodi në PHP dhe thirrja e bazës së dhënave**









**Rezultet**

# Referencat

1. “Algorithm Design” by Jon Kleinberg, Eva Tardos
2. “Algorithms” FOURTH EDITION Robert Sedgewick and Kevin Wayne
3. “Understanding brute force” Daniel J. Bernstein
4. “Data Structures and Algorithms in Python” by Michael T Goodrich, Roberto Tamassia, and Michael H Goldwasser
5. “Introduction to Algorithms” Fourth Edition by Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein
6. “Python Data Structures and Algorithms” by Bejamin Baka
7. “Python ESSENTIAL REFERENCE” by David Beazley
8. “DNS and BIND” by Cricket Liu and Paul Albitz

Lista e Figurave :

Lista e tabelave:

Dasdasd çdoherë