

**МИНИСТАРСТВО ОДБРАНЕ  
УНИВЕРЗИТЕТ ОДБРАНЕ  
ВОЈНА АКАДЕМИЈА**



**ЗАВРШНИ РАД**

**ТЕМА: ПРИМЕНА *PYTHON* ПРОГРАМСКОГ ЈЕЗИКА ЗА  
НАПРЕДНО ПРЕТРАЖИВАЊЕ *WEB* СТРАНИЦА**

**Кадет**  
старији водник  
**(чин)**  
Мирослав Сабо  
**(име и презиме)**

**Ментор**  
пуковник, ванр. проф.  
**(чин, звање)**  
др Иван Тот, дипл. инж.  
**(име и презиме)**

**Београд, 2023. године**

# САДРЖАЈ

САДРЖАЈ.....	1
УВОД.....	3
1   Анализа захтева .....	4
1.1   Опис проблема .....	4
1.2   Анализа захтева корисника .....	4
2   Предлог приступа пројектном решењу .....	6
2.1 <i>Python</i> програмски језик .....	6
2.2 <i>PyCharm IDE</i> .....	9
2.3   Коришћене <i>Python</i> библиотеке .....	10
2.3.1 <i>Urllib.request</i> библиотека .....	10
2.3.2 <i>cyrtranslit</i> библиотека.....	11
2.3.3 <i>regex</i> библиотека.....	11
2.3.4 <i>datetime</i> библиотека.....	12
2.3.5 <i>webbrowser</i> библиотека .....	13
2.3.6 <i>os</i> библиотека.....	13
2.3.7 <i>pathlib</i> библиотека .....	14
2.3.8 <i>tkinter</i> библиотека .....	15
2.3.9 <i>tksheet</i> библиотека .....	17
2.4 <i>HTML (HyperText Markup Language)</i> .....	18
2.4.1   Означавање текста на <i>HTML</i> веб странама .....	19
2.4.2   Претраживање текста по таговима .....	19
2.5   Верзионисање софтвера .....	20
2.5.1   Алат <i>Git</i> .....	20
2.5.2 <i>GitHub</i> платформа.....	20

2.6	Предлог решења тест апликације.....	21
3	Израда тест апликације.....	23
3.1	Главни прозор.....	23
3.2	Прозор са резултатима .....	26
3.3	Прозор за потврду излаза из апликације .....	28
4	Могућности даљег проширења предложеног решења .....	30
	ЗАКЉУЧАК.....	31
	ЛИТЕРАТУРА .....	32
	СПИСАК СЛИКА.....	33

## УВОД

У данашњем времену информације су један од кључних ресурса овог света. Ко поседује више информација, он је бољи. Ово се не односи само на војску и ратне операције. У пословном свету се такође надмеће са информацијама. Пласирају се лажне информације и користе се за манипулацију и стварање лажне слике. Такође се користи и формирање јавног мњења помоћу пласирања информација. Види се значај информација у свету.

Потребно је у овом мору информација наћи информације које су од значаја. Људи углавном, у данашњем времену, користе интернет за информисање. Само претраживање интернета и тражење информација је постала свакодневница.

Информације се чак појављују и на тржишту, где се велики скупови података и информација продају у сврхе неких истраживања и статистичких прорачуна.

Такође се информације користе за прављење и подучавање вештачке интелигенције, која је све популарнија у свету. Вештачкој интелигенцији је потребан велики скуп информација, да би она научила како да се понаша у одређеним ситуацијама.

Циљ овог рада је упознавање аутора са током развоја апликације и сам развој апликације. Поред тога, циљ је био и увођење аутора у истраживачки рад.

Намена апликације је лакше претраживање веб страна путем текста који се тражи на тим веб странама.

У завршном раду је представљено решење апликације за претрагу веб страна. Представљен је проблем претраживања веб страна. Описане су технологије које су коришћене и приказана је тест апликација за претраживање веб страна.

# 1    **Анализа захтева**

У овом поглављу биће описан проблем претраживања веб страна по кључним речима и биће изнети примери јављања тог проблема. Потом ће бити описани захтеви корисника за апликацијом у виду функционалности, изгледа и могућности апликације.

## 1.1   **Опис проблема**

Приликом претраживања више веб страна по кључној речи потребно је најпре посетити сваку страницу и затим претражити у прегледачу кључну реч. За сваку страницу коју би хтели да претражимо мора се поновити тај исти поступак.

Рецимо да Управа за односе са јавношћу Министарства одбране свакодневно претражује да ли се Војска Србије појављује у медијима. Она мора сваког дана да посети исте веб стране и претражи по кључним речима значајним за Министарство одбране. Ту се јавља проблем поновног посећивања веб страна и уношења речи за претрагу.

Овај проблем може бити универзалан. Може се претраживати да ли постоје неки производи на одређеним веб странама за продају робе, где би за сваку страну морали посебно да уносимо појам за претрагу. Такође се могу претраживати огласи за посао за одређену позицију, где се опет долази до истог проблема.

Ту се јавља потреба за аутоматизацијом посећивања веб страна које се понављају и претраживање истих речи на тим веб странама. Аутоматизација доводи до бржег и ефикаснијег претраживања веб страна.

## 1.2   **Анализа захтева корисника**

Корисник апликације мора да има приступ интернету да би могао да претражује веб стране. Затим уноси све веб стране на којима жели да претражи жељени текст. То може бити било која веб страна на коју може да се приступи преко интернета и не захтева аутентификацију. Може да унесе страницу неких новина, странице неких државних служби и сл. Није битно чему је намењена та веб страна.

Корисник затим уноси реч или више речи коју жели да претражи на веб страни. Унете речи треба да се третирају као једна целина и та целина да се претражује на веб страни.

Потребно је претраживати странице и на ћириличном и латиничном писму, без обзира којим писмом су унете тражене речи.

Треба задате речи означити на веб страни да би биле уочљиве и те веб стране сачувати на локалној машини. То треба да омогући приступ претраженим веб странама и без приступа интернету.

Након претраге треба да се генерише извештај, који се налази на новом прозору, где се налазе информације да ли постоје задате речи на одређеној веб страни. Потребно је да се нагласи да ли су те речи означене. Такође је потребно омогућити приступ тим веб странама кликом на дугме.

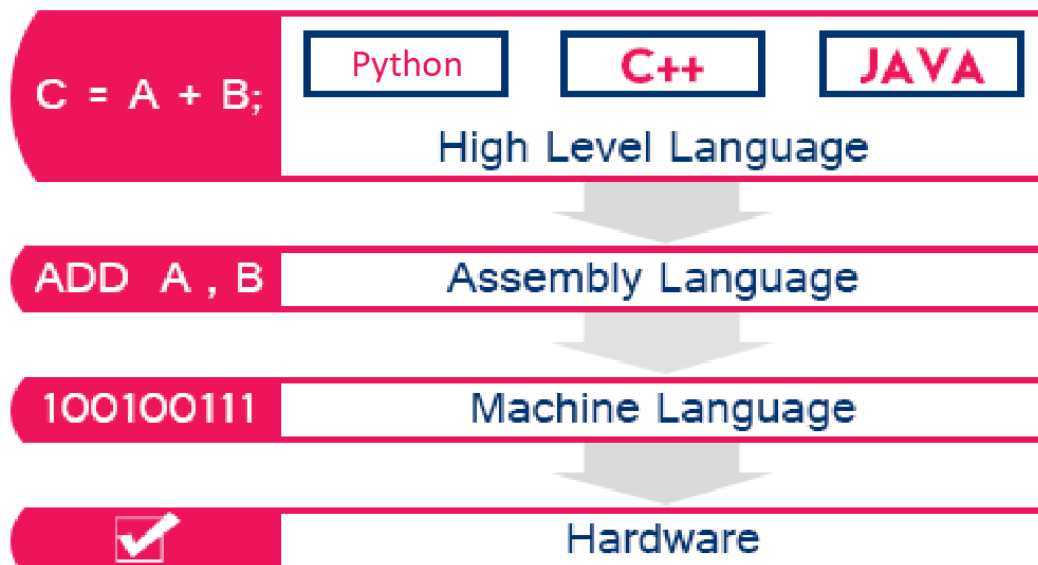
## 2 Предлог приступа пројектном решењу

У овом поглављу ће бити наведене и описане технологије које се користе за израду апликације и алати који су коришћени за примену тих технологија. Такође ће бити описана структура апликације и сама замисао рада апликације.

### 2.1 *Python* програмски језик

*Python* је интерпретирани, објектно оријентисани и функцијски, програмски језик високог нивоа<sup>1</sup>. Он је програмски језик опште намене, што значи да се у њему могу развијати било које апликације. Његове структуре података су комплексне и у комбинацији са динамичким куцањем и динамичким повезивањем, чине га веома атрактивним за брзи развој апликација, као и за писање програмских скрипти. *Python*-ова једноставна синтакса је веома читљива и стога су апликације лакше за одржавање. *Python* подржава модуле и пакете, што подстиче модуларност програма и поновну употребу кода. *Python* интерпретер<sup>2</sup> и обимна стандардна библиотека доступни су у изворном или бинарном облику без накнаде за све главне платформе и могу се слободно дистрибуирати. [1]

Програмски језици високог нивоа су језици које човек може да разуме док их чита (Слика 1).



Слика 1. Хијерархија програмских језика

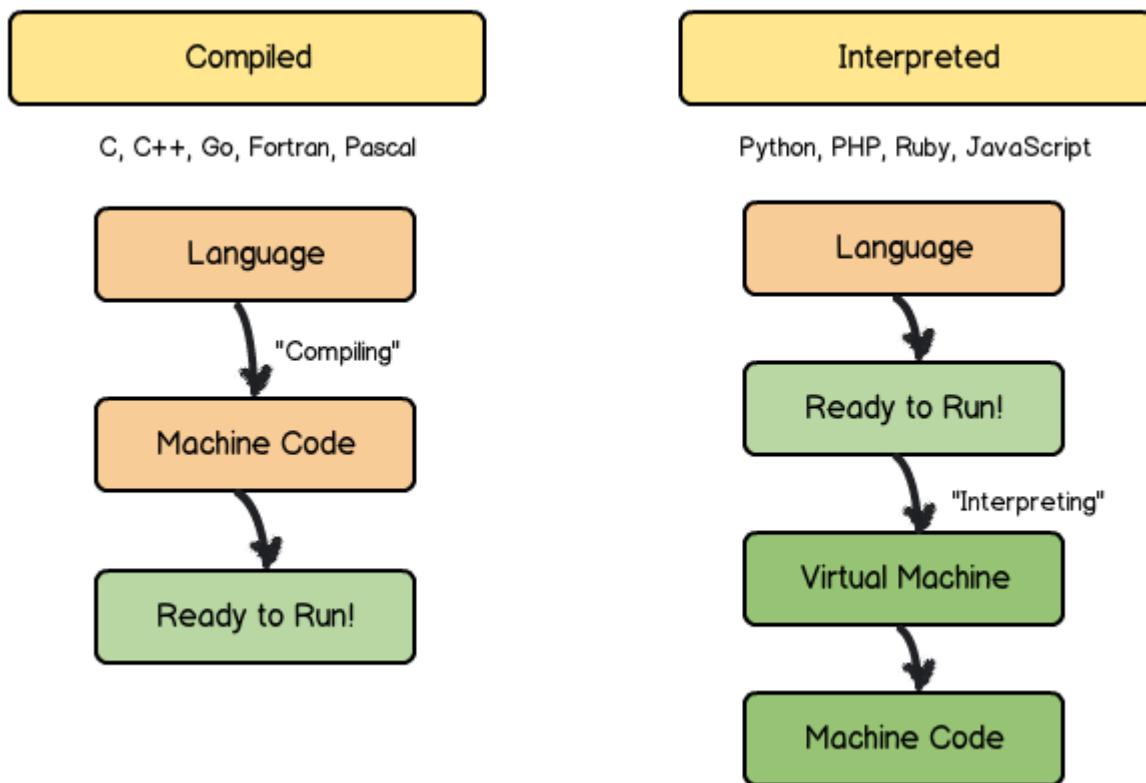
<sup>1</sup> енг. high-level

<sup>2</sup> срп. преводилац

Док се код асемблерских језика користе речи и бројеви за писање кода, он не може да се разуме самим погледом на код. Машински језик користи јединице и нуле као код и он се извршава на процесору.

Интерпретирани програмски језици су језици који користе интерпретер да би се код извршио. Интерпретер је програм који претвара код у машински језик и онда га прослеђује на извршавање. Интерпретер може да се сматра као нека виртуелна машина која се налази на локалној машини и преводи код на машински језик локалне машине. Компајлирањем се код претвара у машински језик на једној машини, као што је приказано на слици 2, и такав се прослеђује осталим машинама. Код компајлираних програмских језика може доћи до погрешног извршавања кода, што није случај код интерпретираних програмских језика.

Предност компајлираних програмских језика је што се извршавају брже, јер нема превођења кода него само извршавања.



Слика 2. Разлика између компајлираних и интерпретираних програмских језика (<https://medium.com/>)

Овај програмски језик је независан од платформе на којој се извршава. То му омогућава интерпретер. На свакој машини на којој се извршава код мора бити инсталиран интерпретер.



Он комуницира са машином и преводи код у машински језик. Тиме се постиже да код није завистан ни од хардвера ни од оперативног система те машине. Овим се решава проблем различитих верзија софтвера и различитог хардвера. Ово омогућава програмеру да не размишља о платформи за коју је намењен код и не мора да прилагођава код различитим платформама.

*Python* може да се користи и као објектно оријентисани и као функцијски језик. Код објектно оријентисаних програмских језика се користе објекти као основе за пројектовање програма. Сваки објекат представља решење неког проблема и скупом тих објеката и њихових функционалности долази се до апликације. Код функцијских програмских језика се проблеми деле на функције и подфункције, где свака функција решава одређени проблем или део проблема. Спајањем ових функција долази се до решавања целокупног проблема, односно апликације.

*Python* је бесплатан програмски језик, што значи да за коришћење његовог интерпретера и стандардне библиотеке је без накнаде. Поред стандардне библиотеке постоји и мноштво бесплатних библиотека које проширују могућности овог програмског језика. Ове библиотеке олакшавају програмеру да искористи код од неких других аутора. Омогућено је да се користе постојећа решења за одређене проблеме и тиме се штеди време. Програмер може да решава нове проблеме без задржавања на старим проблемима.

У *Python*-у се библиотеке инсталирају једноставном командом „`pip3 install naziv_biblioteke`“. Потребан је приступ интернету и цела библиотека се преузме и инсталира у програмско окружење. Након тога се мора најпре позвати у код кључном речју *import*. После тога можемо користити све функционалности те библиотеке и уштедети на времену у писању кода.

Овај програмски језик има свој *garbage collection*<sup>3</sup> који служи да води рачуна о меморији. Меморија која се више не користи се аутоматски ослобађа и враћа у опсег слободне меморије. Самим тим програмер не мора да води рачуна о меморији, него интерпретер брине да се меморија ослобађа кад је то потребно.

За израду овог рада коришћен је *Python 3.10*, који је у време израде рада био најновија верзија овог програмског језика.

---

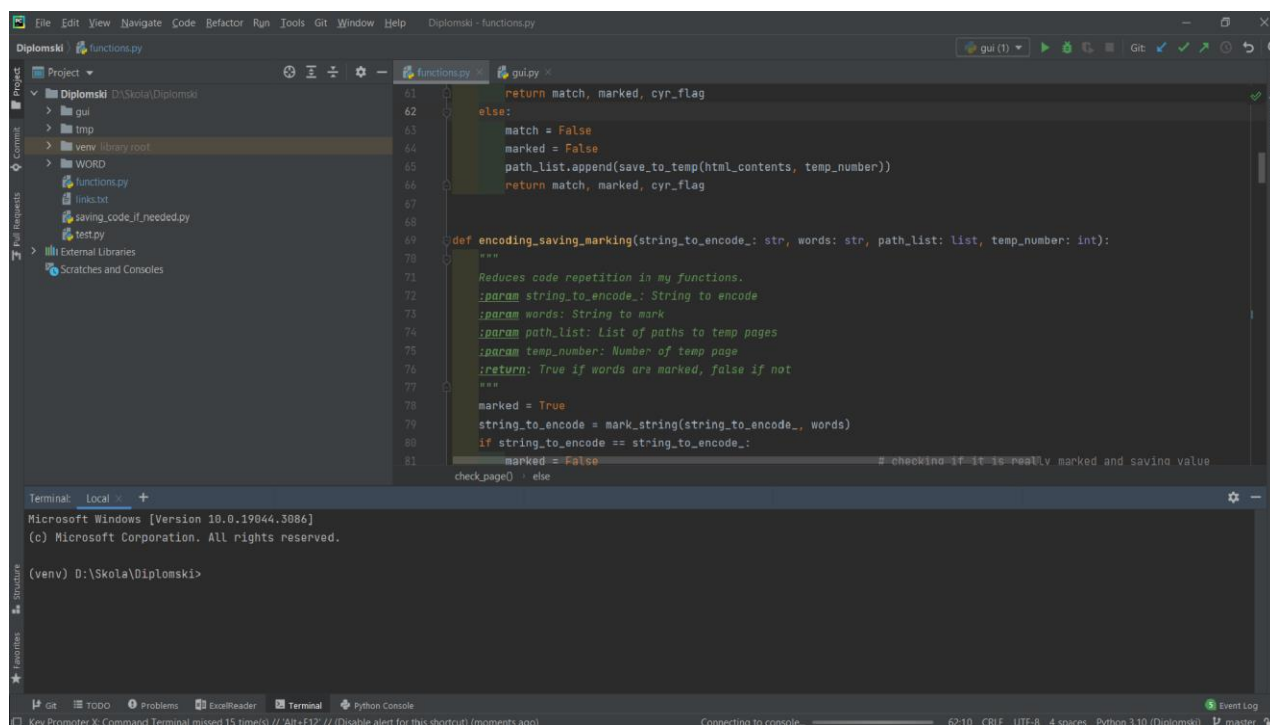
<sup>3</sup> спр. скупљање смећа

## 2.2 PyCharm IDE

PyCharm је интегрисано развојно окружење<sup>4</sup> (IDE) које се користи за развој програма у *Python* програмском језику. Оваква развојна окружења постоје за различите програмске језике и имају уграђене алате потребне за развој програма.

У овој апликацији се може писати изворни код, увозити библиотеке, инсталирати помоћне *plugin*<sup>5</sup>-е, покретати тестови апликације, „дебаговати“ код и сл. Изглед ове апликације се може видети на слици 3.

За овај рад коришћен је *PyCharm 2020.3.4 (Community Edition)*. Изабрана је ова верзија *IDE*-а, јер је ово бесплатна верзија која има довољно функционалности за једног програмера, али није компатибилна за веће пројекте. [2]



Слика 3. Изглед *PyCharm IDE*-а

<sup>4</sup> енгл. Integrated Development Environment

<sup>5</sup> срп. додатне функционалности

## 2.3 Коришћене *Python* библиотеке

У овом делу рада ће бити наведене и објашњене све библиотеке и њихове функционалности које су коришћене у изради рада.

Коришћене библиотеке у раду:

- *Urllib.request* библиотека
- *cyrtranslit* библиотека
- *regex* библиотека
- *datetime* библиотека
- *webbrowser* библиотека
- *os* библиотека
- *pathlib* библиотека
- *tksheet* библиотека
- *tkinter* библиотека

### 2.3.1 *Urllib.request* библиотека

Ова библиотека садржи функције и класе потребне за рад са *URL*<sup>6</sup>-овима (углавном *HTTP*<sup>7</sup>). Није потребно додатно скидање и инсталирање ове библиотеке, јер је она део стандардне *Python*-ове библиотеке. За њено коришћење довољно је само позивање библиотеке командом „*import Urllib.request*“.

*Urllib.request* библиотека је *open source*<sup>8</sup>, што значи да се може приступити коду који стоји иза ње. Он се такође може и модификовати и прослеђивати на даље коришћење. У суштини, може се знати шта код ради у позадини.

*HTTP* је мрежни протокол, који представља главни и најчешћи метод преноса информација на вебу и он се користи за пренос *HTML*<sup>9</sup> докумената, тј. веб страница.

Ова библиотека се користи за приступ *URL*-овима на интернету, који нас воде на жељене веб странице. Такође се користи за чување веб страница на локалну машину. [3]

За израду овог рада коришћена је уграђена библиотека из *Python 3.10* верзије.

---

<sup>6</sup> Uniform Resource Locator, срп. веб-адреса

<sup>7</sup> HyperText Transfer Protocol

<sup>8</sup> срп. отвореног кода

<sup>9</sup> HyperText Markup Language

### 2.3.2 *cyrtranslit* библиотека

*Cyrtranslit* библиотека се користи за транслитерацију из латиничног у ћирилично писмо и обрнуто. Подразумевани језик библиотеке је Српски, али она може да се користи и за транслитерацију Бугарског, Македонског, Монголског, Руског и Украјинског језика.

Транслитерација је претварање текста из једног писма у друго, нпр. из ћирилице у латиницу. [4]

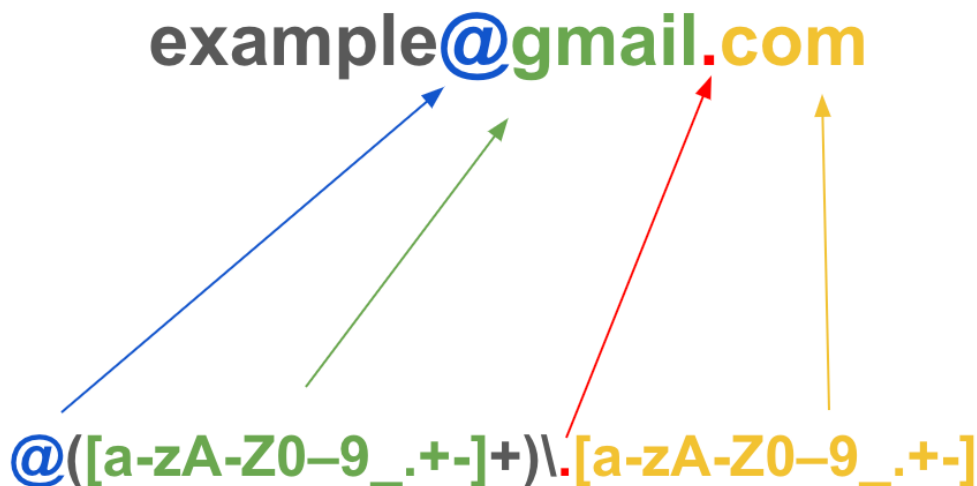
Ова библиотека је *open source*, тако да је могуће њено преузимање и модификација за жељену употребу. Такође је могуће остварити увид у рад библиотеке у позадини.

Поредно је најпре преузети ову библиотеку а затим је и инсталирати у развојно окружење, да би било могуће коришћење њених функционалности. Преузимање и инсталација се врши помоћу команде „*pip3 install cyrtranslit*“. Након завршене инсталације, за коришћење библиотеке потребно је увести библиотеку командом „*import cyrtranslit*“. Овом командом се библиотека може користити у коду.

За израду овог рада коришћена је верзија *cyrtranslit* 1.1.1, која је била најновија за време израде рада.

### 2.3.3 *regex* библиотека

*Regex* библиотека се користи за проналажење шаблона у тексту и манипулисање са тим шаблонима или модификовање тих шаблона. На пример провера да ли је унешени текст *e-mail* адреса (Слика 4).



Слика 4. Провера исправности *e-mail* адресе коришћењем *regex* израза (<https://towardsdatascience.com/easiest-way-to-remember-regular-expressions-regex-178ba518bebd> )

*Regex* је скраћеница од *regular expression*, што значи устаљени израз, и користи се у програмирању за проналажење шаблона у тексту.

*Regex* библиотека је *open source*. Она се може слободно користити и модификовати. [5]

Ова библиотека не припада стандардним библиотекама у *Python*-у. Она се мора посебно преузети и инсталирати. За то се користи команда „*pip3 install regex*“. Након завршене инсталације, за коришћење библиотеке и њених функционалности потребно је увести библиотеку командом „*import regex*“. Након ове команде је библиотека спремна за коришћење. У овом раду је ова библиотека коришћена за проналажење ћириличног писма на веб страницама. Овим се омогућило препознавање да ли је веб страна на ћирилици или латиници. Даље претраживање веб странице је олакшано јер је познато писмо на коме је веб страна. У овом раду је коришћена верзија *regex* 2023.3.22, која је била најновија за време израде рада.

#### 2.3.4 *datetime* библиотека

Ова библиотека је намењена за обраду и коришћење података о датуму и времену. Може се користити за рачунање датума и времена и разлике у временима, али основна намена јој је формирање изгледа и извлачење потребних информација из датума и времена.

Постоје уграђене класе које нам омогућавају рад са датумом, временом, комбинацијом датума и времена, временским трајањем и временским зонама (Слика 5).

<u>Classes</u>	<u>Attributes</u>	<u>Example</u>
date	year, month and day	2022-08-19
time	hour, minute, second, microsecond(tzinfo)	09:50:12
datetime	year, month, day, hour, minute, second, microsecond(tzinfo)	19/08/2022 09:50:12
timedelta	#allows us to work with time duration	T1= 10/10/2021 T2= 15/10/2021 Time_delta= 5days
tzinfo	#allows us to work with timezones	+04:00

Слика 5. Класе и атрибути класа у *datetime* библиотеци са примерима (<https://takeuforward.org/python/python-datetime> )

У свакој класи постоје атрибути којима можемо појединачно приступити и добити жељену информацију.

*Datetime* библиотека омогућава и рад кода у више временских зона и његову синхронизацију по времену. Она може сама да води рачуна о временској зони. [6]

Ова библиотека је *open source*. Могући је увид у рад ове библиотеке у позадини и модификација библиотеке по жељи корисника.

*Datetime* библиотека је уграђена у стандардну *Python*-ову библиотеку, тако да није потребно додатно преузимање са интернета и инсталација библиотеке. За коришћење библиотеке је потребно увести библиотеку командом „*import datetime*“. Овом командом омогућавамо коришћење библиотеке.

У овом раду је коришћена за добијање података о тренутном времену када се претражују веб странице. Употребљена је из стандардне библиотеке из *Python 3.10* верзије.

### **2.3.5 *webbrowser* библиотека**

*Webbrowser* библиотека се користи за рад са интернет претраживачем. Она прегледа да ли постоји подразумевани интернет претраживач на машини и користи њега. Ако он не постоји користи текстуални претраживач.

Она је врло једноставна, *open source* библиотека. Лако се користи и уграђена је у стандардну *Python*-ову библиотеку. Пре њеног коришћења потребно је увести библиотеку командом „*import webbrowser*“. [7]

Користи се тако што се проследи путања до жељене веб странице. Та веб страница може да се налази на локалној машини или на интернету.

У овом раду коришћена је за отварање преузетих веб страница и отварање веб страница са интернета.

Коришћена је *webbrowser* библиотека из стандардне библиотеке из *Python 3.10* верзије.

### **2.3.6 *os* библиотека**

Ова библиотека се користи за комуницирање са оперативним системом на ком се код извршава и коришћење функционалности оперативног система. Путем ове библиотеке може да се отвори фајл, направи нови фајл, модификује неки фајл, добије информација о путањи до фајла или тренутна путања до фолдера где се код извршава. Може да се користи за креирање фолдера и подфолдера и креирање помоћних фајлова. Могу да се прате процеси на машини, приступа системским променљивима и сл. [8]

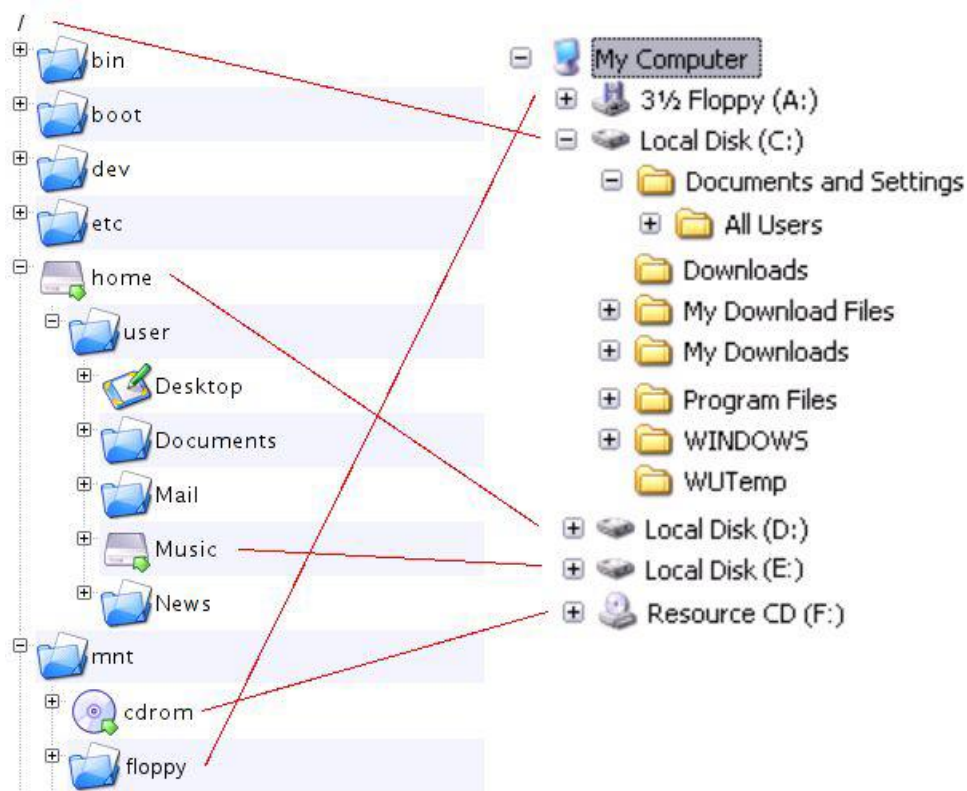
Os библиотека је *open source* библиотека. Она је уграђена у стандардну *Python*-ову библиотеку. Пре њене употребе потребно ју је увести командом „*import os*“, али могуће је увести и жељене делове ове библиотеке, уместо целе библиотеке, због њеног обима.

У овом раду *os* библиотека је коришћена за уписивање веб страница на локалну машину, мењање локације на којој се извршава програм и манипулацију са потребним фајловима. Коришћена је *os* библиотека из пакета стандардних библиотека из *Python 3.10* верзије.

### 2.3.7 *pathlib* библиотека

*Pathlib* библиотека се користи за придобијање информација о путањи до фајла и поседује класе путања за различите оперативне системе, што дозвољава превођење путања између оперативних система.

Пошто се у *Linux*-у и *Windows*-у не користе исти фајл системи (Слика 6) ни путање до фајлова неће бити исте. [9]



Слика 6. Разлика у *Linux*-овом и *Windows*-овом фајл систему  
([https://stevevincent.info/LUX205\\_2017\\_2.html](https://stevevincent.info/LUX205_2017_2.html) )

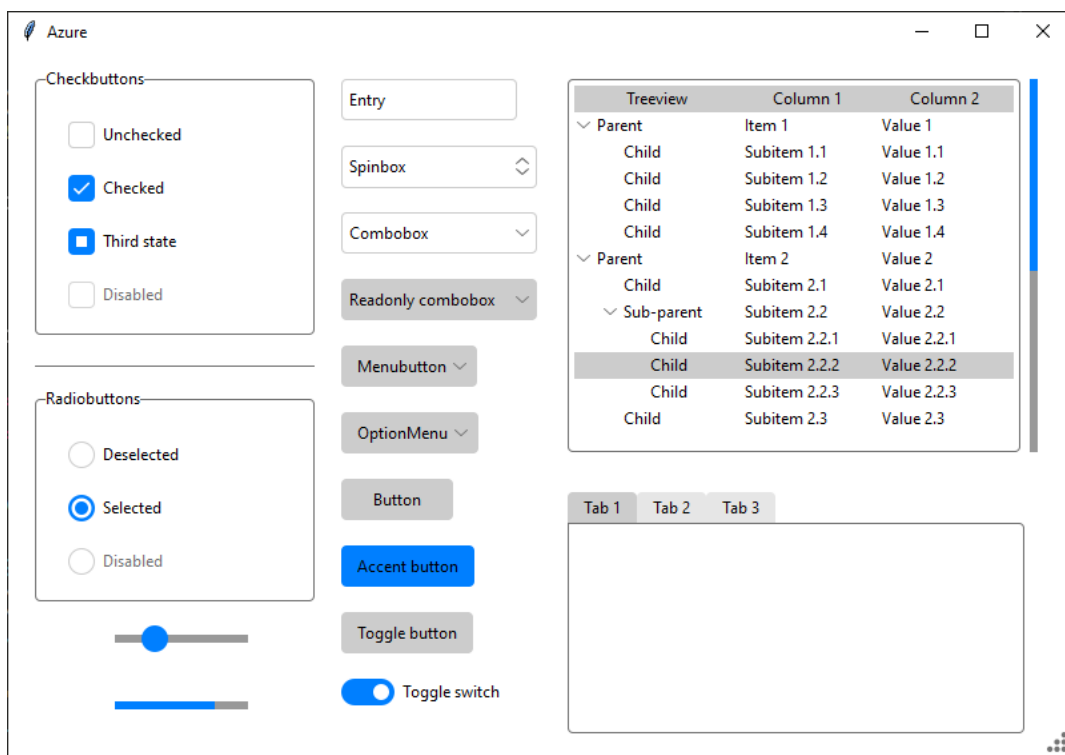
Коришћењем ове библиотеке се овај проблем превазилази. Она чува путању као такозвану „чисту путању“ и као такву може да је преведе у путању за жељени оперативни систем.

*Pathlib* библиотека је *open source* и налази се у пакету стандардних *Python*-ових библиотека. За увођење ове библиотеке потребно је користити команду „*import pathlib*“. Након увођења библиотеке она се може користити једноставним позивом жељене класе.

У овом раду се *pathlib* библиотека користила за чување путање до привремених фајлова у којима се чувају претражене веб странице. Ова библиотека је коришћена из пакета стандардних библиотека из *Python 3.10* верзије.

### 2.3.8 *tkinter* библиотека

Ова библиотека је стандардна библиотека у *Python*-у, која се користи за израду *GUI*<sup>10</sup> апликација. *GUI* је део апликације који приказује кориснику шта апликација ради и како изгледа и омогућава лако коришћење апликације (Слика 7). Он је уствари прозор који се види када се апликација покрене и помаже кориснику да лакше користи апликацију преко графичких објеката.



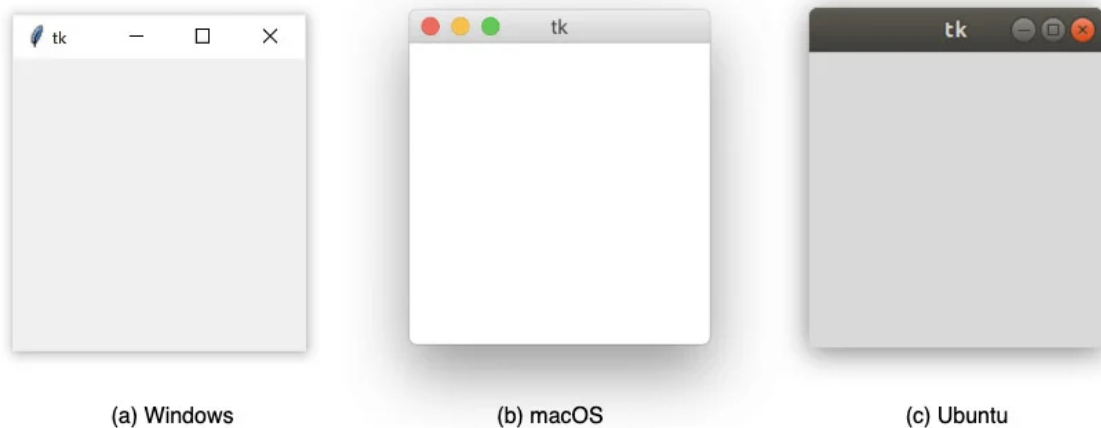
Слика 7. Једноставан *GUI* прозор направљен помоћу *tkinter* библиотеке (<https://pythonprogramming.altervista.org/nice-gui-graphic-for-tkinter-with-ttk-and-azure-theme-from-this-guy> )

<sup>10</sup> Graphical User Interface, срп. графички кориснички интерфејс



Tkinter библиотека је лака за коришћење и омогућава брзу израду *GUI* апликација. Она је конципирана као објектно-орјентисана библиотека. Тиме се постиже да сваки објект на екрану има своју класу унутар библиотеке и дозвољава нам лак приступ атрибутима унутар класе и модификацију објекта на жељени начин.

*Tkinter* библиотека је компатибилна са *Linux*, *Windows* и *MacOS* оперативним системима (Слика 8). Ова компатибилност дозвољава да *GUI* ради на овим платформама без додатног програмирања за различите оперативне системе. Она комуницира са оперативним системом и тако користи његове функционалности за манипулацију са *GUI* прозорима. [10]



Слика 8. Главни *GUI* прозор у различитим оперативним системима  
(<https://realpython.com/python-gui-tkinter> )

Ова библиотека је уграђена у пакет стандардних *Python*-ових библиотека. За њено коришћење потребно ју је најпре увести командом „*import tkinter*“. Након тога је она спремна за коришћење и израду *GUI* апликација.

*Tkinter* библиотека је *open source*. То омогућава модификацију свих њених објеката. Тако може да се направи јединствени изглед и понашање *GUI* апликације.

За овај рад коришћена је *Tkinter* библиотека из пакета стандардних библиотека из *Python 3.10* верзије.

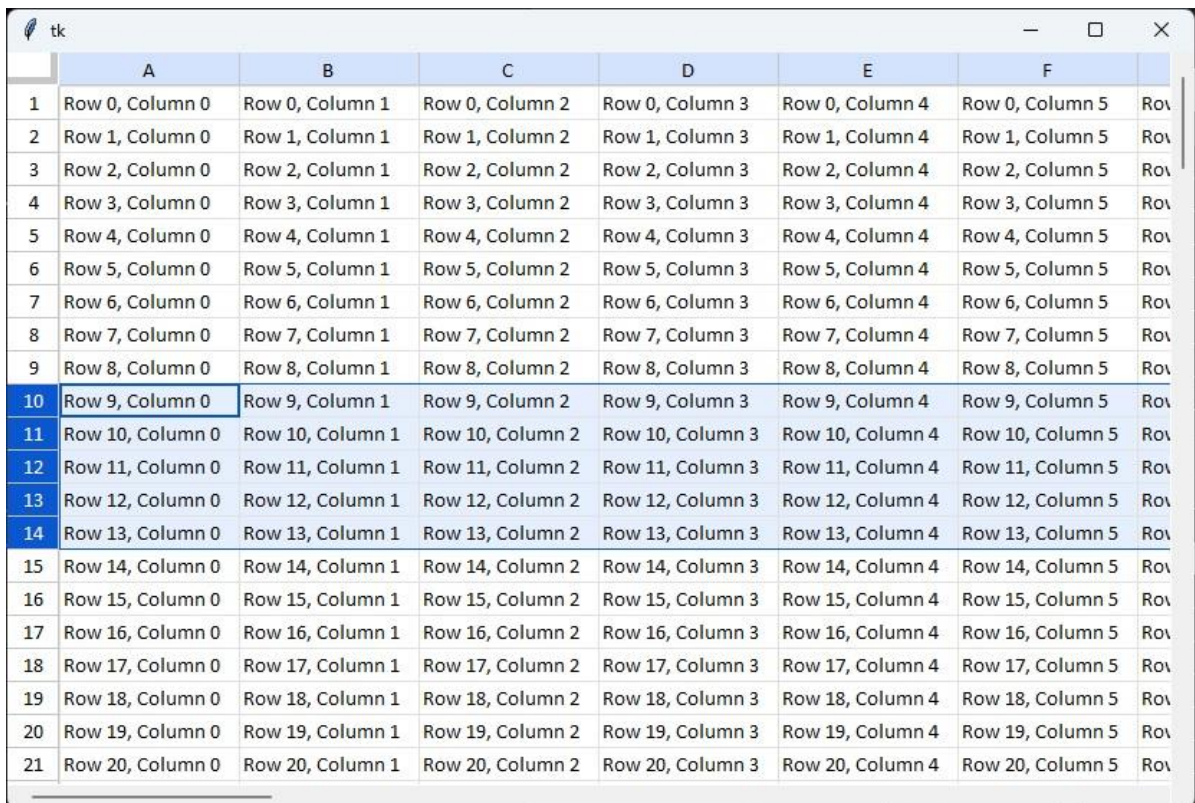
У овом раду коришћене су следеће функционалности ове библиотеке:

- *Tk* - главни прозор апликације (Слика 8)
- *Canvas* - позадина прозора на коју могу да се стављају други објекти
- *PhotoImage* - убацивање слике у прозор
- *Entry* - поље за унос кратког текста
- *Text* - поље за унос дугог текста

- *Button* - дугме
- *Progressbar* - трака за читавање
- *Checkbutton (checkbox)* - поље за потвду
- *Toplevel* - помоћни прозор који припада главном прозору
- *Label* - текст на прозору
- *Messagebox* - прозор за потврђивање

### 2.3.9 *tksheet* библиотека

Tksheet библиотека се користи за табуларни приказ података на *tkinter GUI* прозорима (Слика 9). Она прослеђене податке смести у табелу и конфигурише ту табелу за приказ на екрану. За њен рад је потребно имати *tkinter GUI* прозор. [11]



	A	B	C	D	E	F
1	Row 0, Column 0	Row 0, Column 1	Row 0, Column 2	Row 0, Column 3	Row 0, Column 4	Row 0, Column 5
2	Row 1, Column 0	Row 1, Column 1	Row 1, Column 2	Row 1, Column 3	Row 1, Column 4	Row 1, Column 5
3	Row 2, Column 0	Row 2, Column 1	Row 2, Column 2	Row 2, Column 3	Row 2, Column 4	Row 2, Column 5
4	Row 3, Column 0	Row 3, Column 1	Row 3, Column 2	Row 3, Column 3	Row 3, Column 4	Row 3, Column 5
5	Row 4, Column 0	Row 4, Column 1	Row 4, Column 2	Row 4, Column 3	Row 4, Column 4	Row 4, Column 5
6	Row 5, Column 0	Row 5, Column 1	Row 5, Column 2	Row 5, Column 3	Row 5, Column 4	Row 5, Column 5
7	Row 6, Column 0	Row 6, Column 1	Row 6, Column 2	Row 6, Column 3	Row 6, Column 4	Row 6, Column 5
8	Row 7, Column 0	Row 7, Column 1	Row 7, Column 2	Row 7, Column 3	Row 7, Column 4	Row 7, Column 5
9	Row 8, Column 0	Row 8, Column 1	Row 8, Column 2	Row 8, Column 3	Row 8, Column 4	Row 8, Column 5
10	Row 9, Column 0	Row 9, Column 1	Row 9, Column 2	Row 9, Column 3	Row 9, Column 4	Row 9, Column 5
11	Row 10, Column 0	Row 10, Column 1	Row 10, Column 2	Row 10, Column 3	Row 10, Column 4	Row 10, Column 5
12	Row 11, Column 0	Row 11, Column 1	Row 11, Column 2	Row 11, Column 3	Row 11, Column 4	Row 11, Column 5
13	Row 12, Column 0	Row 12, Column 1	Row 12, Column 2	Row 12, Column 3	Row 12, Column 4	Row 12, Column 5
14	Row 13, Column 0	Row 13, Column 1	Row 13, Column 2	Row 13, Column 3	Row 13, Column 4	Row 13, Column 5
15	Row 14, Column 0	Row 14, Column 1	Row 14, Column 2	Row 14, Column 3	Row 14, Column 4	Row 14, Column 5
16	Row 15, Column 0	Row 15, Column 1	Row 15, Column 2	Row 15, Column 3	Row 15, Column 4	Row 15, Column 5
17	Row 16, Column 0	Row 16, Column 1	Row 16, Column 2	Row 16, Column 3	Row 16, Column 4	Row 16, Column 5
18	Row 17, Column 0	Row 17, Column 1	Row 17, Column 2	Row 17, Column 3	Row 17, Column 4	Row 17, Column 5
19	Row 18, Column 0	Row 18, Column 1	Row 18, Column 2	Row 18, Column 3	Row 18, Column 4	Row 18, Column 5
20	Row 19, Column 0	Row 19, Column 1	Row 19, Column 2	Row 19, Column 3	Row 19, Column 4	Row 19, Column 5
21	Row 20, Column 0	Row 20, Column 1	Row 20, Column 2	Row 20, Column 3	Row 20, Column 4	Row 20, Column 5

Слика 9. Tksheet табела у *tkinter* прозору  
(<https://pypi.org/project/tksheet/> )

Ова библиотека је *open source*, тако да је могуће њено преузимање, преглед рада саме библиотеке и модификација за жељену употребу.

Поредно је најпре преузети ову библиотеку а затим је и инсталирати у развојно окружење, да би било могуће коришћење њених функционалности. Преузимање и инсталација се врши

помоћу комаде „*pip3 install tksheet*“. Након завршене инсталације, за коришћење библиотеке потребно је увести библиотеку командом „*import tksheet*“. Овом командом се библиотека може користити у коду.

За израду овог рада коришћена је верзија *tksheet* 6.0.0., која је била најновија за време израде рада.

## 2.4 *HTML (HyperText Markup Language)*

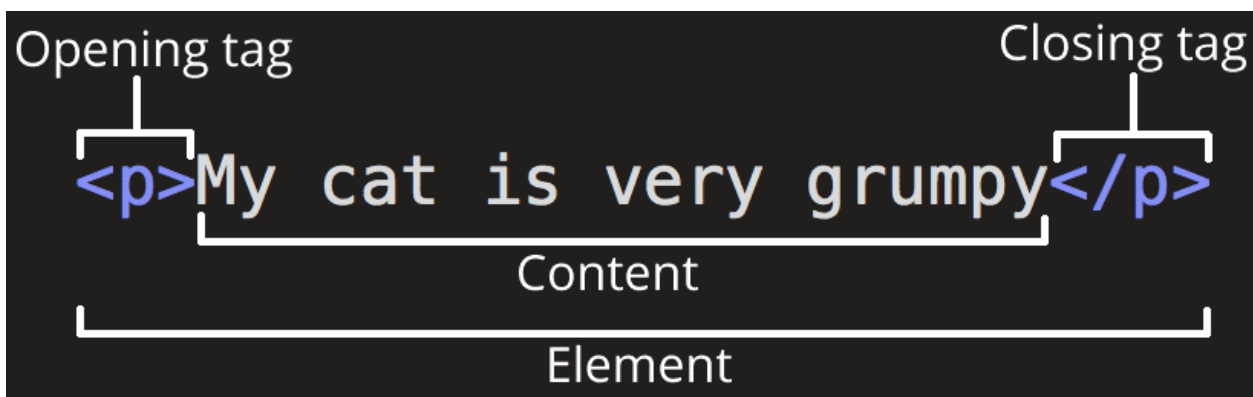
*HTML* је језик за израду веб страница. *HyperText* означава међусобно повезивање једне или више веб страница. Њиме се дефинише структура веб странице и елементи који се налазе на самој веб страници. Он комуницира са интернет претраживачем и распоређује елементе по претраживачу. Њиме се дефинише намена објеката на страници, нпр. слика, линк, наслов и сл. *HTML* елемент је одвојен од другог текста у документу помоћу "тагова", који се састоје од назива елемента окруженог са "<" и ">". Име елемента унутар ознаке не разликује велика и мала слова. То јест, може се писати великим, малим словима или мешавином. На пример, ознака „<title>“ може бити написана као „<Title>“, „<TITLE>“ или на било који други начин. Међутим, конвенција и препоручена пракса је писање ознака малим словима. [12]

Ознака отварања тага се састоји од имена елемента умотаног у угаоне заграде за отварање и затварање. Ово наводи где елемент почиње или почиње да ступа на снагу.

Завршна ознака тага је иста као и почетна ознака, осим што укључује косу црту испред имена елемента. Ово наводи где се елемент завршава. Заборављање додавања завршне ознаке је једна од стандардних почетничких грешака и може довести до чудних резултата.

Садржај елемента се налази између ових тагова.

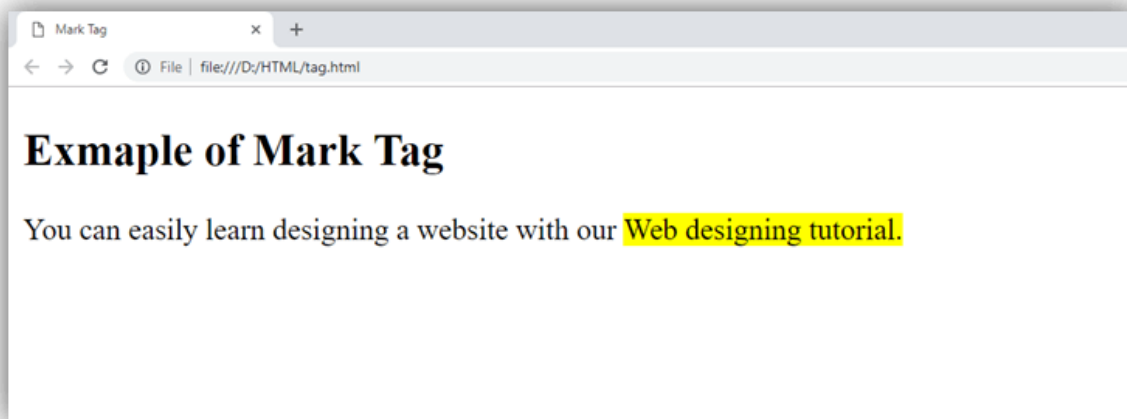
Ознака за отварање, ознака за затварање и садржај заједно чине елемент. (Слика 10)



Слика 10. Структура *HTML* елемента  
([https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics))

#### 2.4.1 Означавање текста на *HTML* веб странама

У овом раду је коришћен *HTML* за означавање пронађених речи или скупа речи на веб страници. Коришћењем тага „`<mark>`“ и „`</mark>`“ омогућено је означавање текста на веб страни жутом бојом (Слика 11). Текст који је означен жутом бојом се налази између тагова „`<mark>`“ и „`</mark>`“.



Слика 11. Пример коришћења тага за означавање  
(<https://www.javatpoint.com/html-mark-tag>)

#### 2.4.2 Претраживање текста по таговима

Да би претраживање текста на *HTML* страници било лакше, коришћено је претраживање по одређеним таговима. Тиме се убрзао рад самог програма јер је скраћен обим претраге на тагове који садрже текст и наслове а не и све остале тагове који се налазе у *HTML* документу.

## 2.5 Верзионисање софтвера

Приликом израде неког програма, када се дода неко побољшање или промена у програму одређује се нова верзија тог програма. Углавном се те верзије означавају бројевима, нпр. *tksheet* 6.0.0. Бројеви са лева на десно означавају величину промене у том програму. Ако се мења леви број онда је то нека велика промена у том софтверу, а ако је десни број онда је то нека мала промена у функционалности или је отклоњен неки проблем. [13]

Постоје разни алати за контролисање верзионисања софтвера. У овом делу рада је обрађен најпопуларнији алат за верзионисање, који је коришћен у овом раду. Овај алат је *Git*.

### 2.5.1 Алат *Git*

*Git* је алат за контролу верзионисања софтвера. Он је намењен за развој софтвера у већем окружењу, што значи да већи број људи ради на истом софтверу. Он сваком програмеру доставља његову копију кода на којој он може да ради. Након завршетка рада он чува верзију пре промене кода или неког фајла и након промене кода и фајла, где свака верзија има своју ознаку. Он води рачуна о променама у сваком фајлу и фолдеру, а не само о променама у коду. Дозвољава да сваки човек који ради на пројекту остави опис промене коју је унео у пројекат. [14]

### 2.5.2 *GitHub* платформа

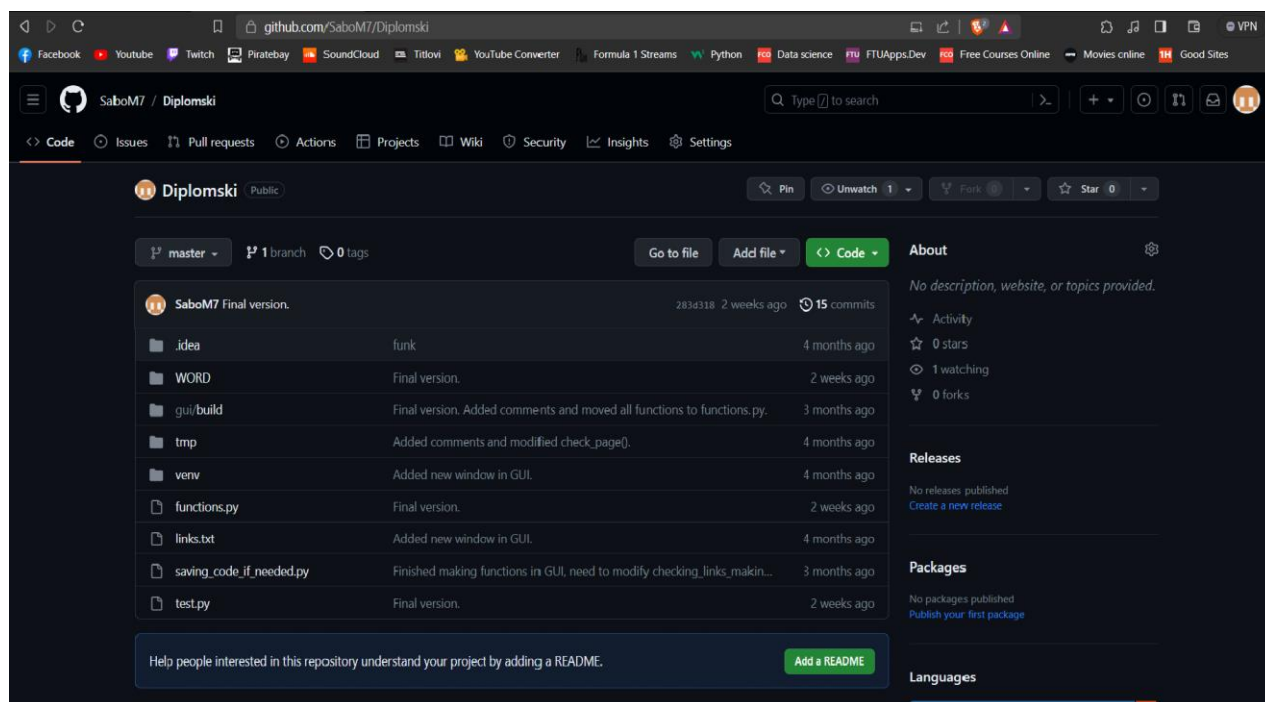
*GitHub* платформа је сервис у облаку<sup>11</sup> која је намењена за дељење и складиштење верзија софтвера направљених у *Git* алату. Она може да служи и за складиштење разних фајлова, не мора да се користи само за верзионисање софтвера. Може да се користи и за чување верзија књига или пројеката из других области. [15]

Ова платформа има интуитиван кориснички интерфејс (Слика 12) и лака је за коришћење. Због тога је постала много популарна за верзионисање пројеката.

Ова платформа нуди више пакета за коришћење њених услуга. Постоје бесплатни пакети за кориснике и компаније који имају мање могућности за контролу пројеката и мањи меморијски простор за складиштење фајлова. Код пакета који се плаћају постоје веће могућности за контролу тока пројекта и већи простор за складиштење фајлова. Такође је могуће чешће верзионисање у односу на бесплатну верзију.

---

<sup>11</sup>eng. Cloud-service



Слика 12. Изглед платформе *GitHub*

У овом раду је коришћено верзионисање софтвера помоћу алата *Git* који може да се користи кроз *PyCharm IDE*, што олакшава рад са самим алатом. Код је чуван на платформи *GitHub*, поред чувања и на локалној машини. То је омогућило да рад буде сачуван и у случају отказивања локалне машине.

Верзионисање апликације је такође вршено на платформи *GitHub*. То је помогло при праћењу додавања функционалности и описивању промена у коду и пројекту.

Коришћена је верзија *Git 2.39.1.windows.1* алата за верзионисање и веб страница *GitHub* (<https://github.com/>) за чување кода и верзија. За коришћење платформе *GitHub* потребно је имати интернет конекцију.

## 2.6 Предлог решења тест апликације

Тест апликација треба да прихвата унос жељених веб страница и текста за претрагу на тим веб странама. Ове податке треба да унесе корисник. Ако ови подаци нису унети, апликација не треба да настави даљи рад. У апликацији треба да постоје одређене веб стране које могу опционо да се изаберу за претрагу, попут неких дневних листова.

Апликација треба да претражује појам по страницама без обзира да ли је он унет великим или малим словима. Такође треба да провери да ли су унети појам и свака веб страна написани у ћириличном или латиничном писму и сходно томе изврши транскрипцију траженог појма.

За рад апликације потребан је приступ интернету. Након претраживања задатог појма веб страна треба да се сачува на локалној машини. Треба да се омогући приступ веб страни на локалној машини и на интернету кликом на дугме.

Након претраге потребно је да се тражени текст означи ако је нађен и да се саопшти кориснику да ли је на тој веб страни пронађен тражени појам.

Потребно је сачувати и приказати тачно време претраге појма по веб странама.

### 3 Израда тест апликације

У овом делу рада биће приказан изглед тест апликације и биће описане све функционалности апликације.

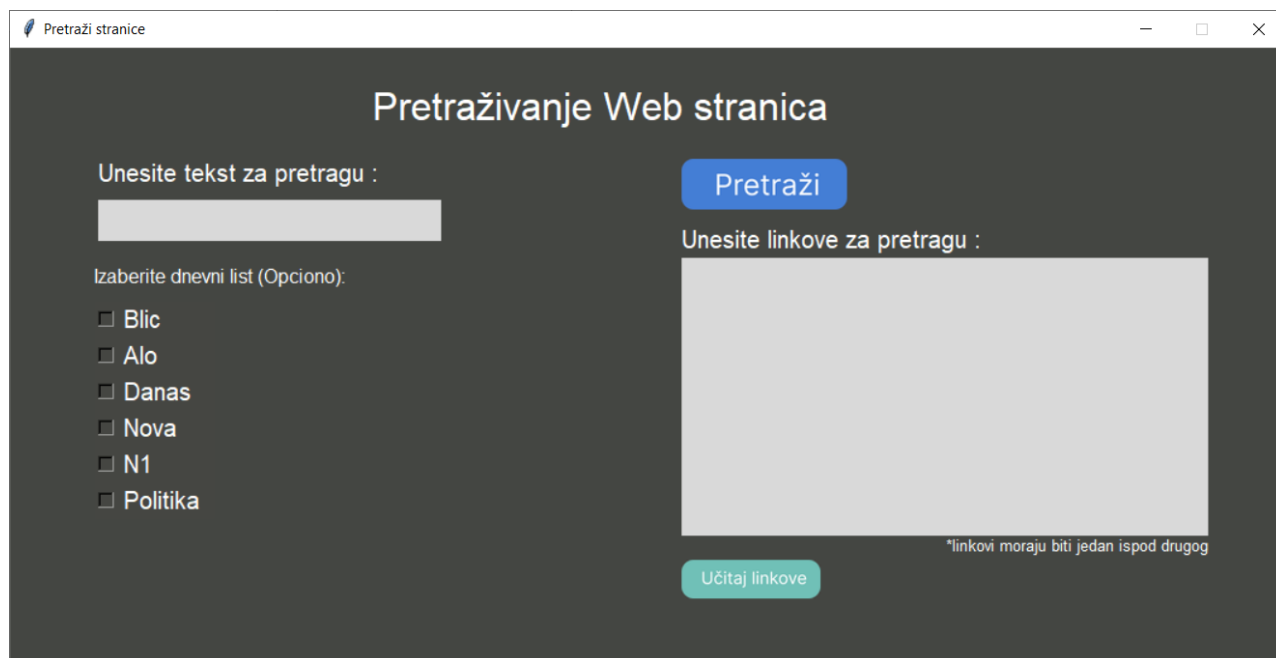
У изради апликације коришћене су следеће технологије, које су описане у претходном поглављу:

- *Python 3.10*
- *HTML*
- *Git*

За употребу ових технологија коришћен је *PyCharm 2020.3.4 (Community Edition) IDE*. Он у себи има интегрисане алате за коришћење ових технологија.

#### 3.1 Главни прозор

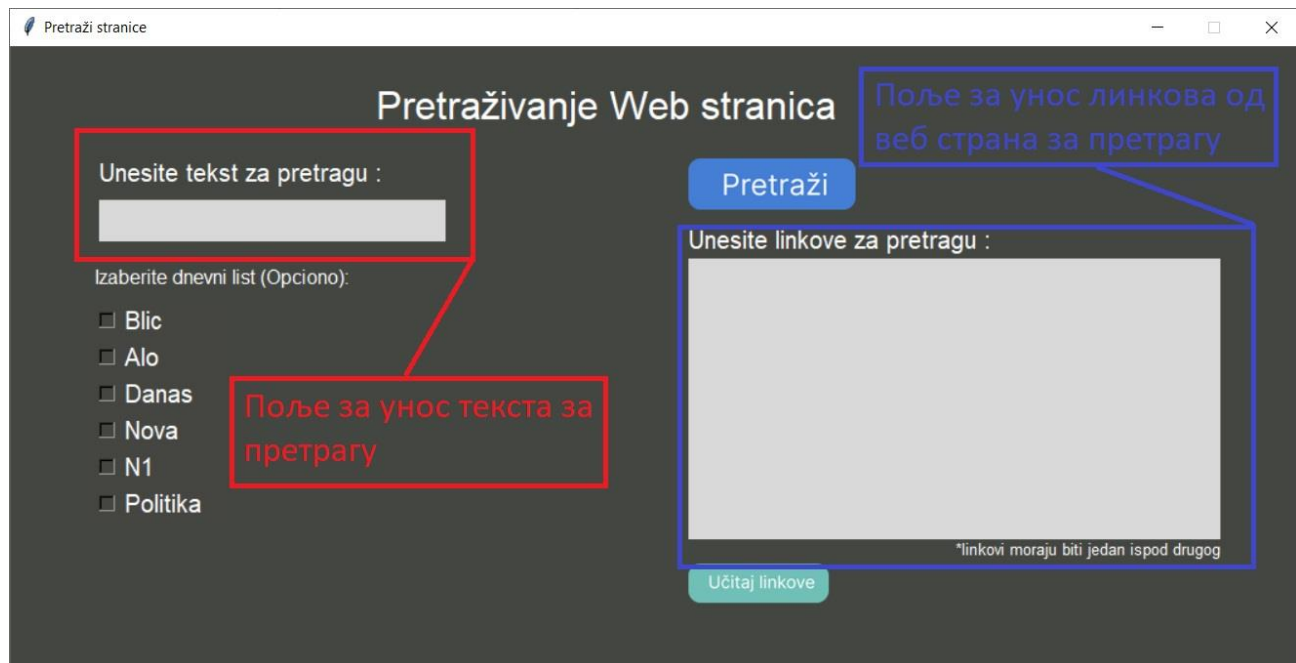
Главни прозор ове апликације се појављује одмах макон покретања апликације. Његов изглед је приказан на слици 13.



Слика 13. Изглед главног прозора тест апликације



На главном прозору се налазе поља за унос текста за претрагу и линкова од веб стана на којима треба извршити претрагу (Слика 14).



Слика 14. Поља за унос текста за претрагу и унос веб страна на којим се врши претрага

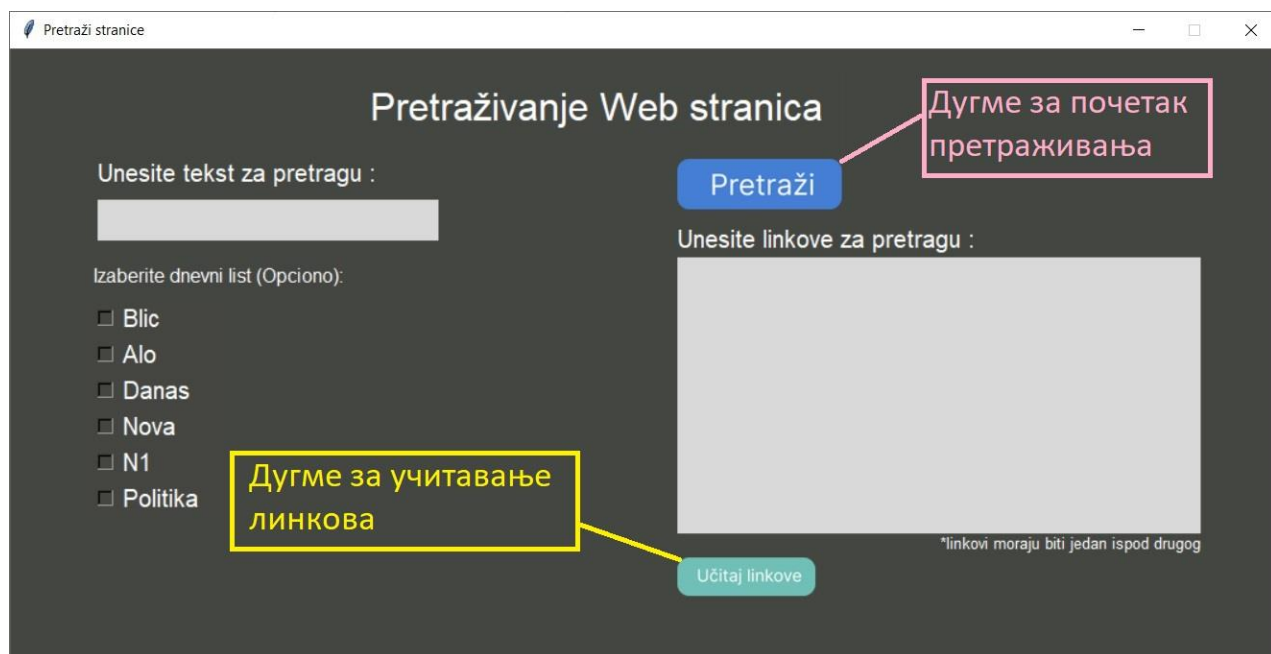
У поље за унос линкова потребно је уносити линкове један испод другог да би апликација радила правилно. Ако се у једно од поља не унесе ништа онда апликација не почиње са претраживањем и чека унос у поље.

Такође се на почетном прозору налазе дугмићи за почетак претраживања страница и учитавање линкова који су били унети при изласку из апликације (Слика 15).

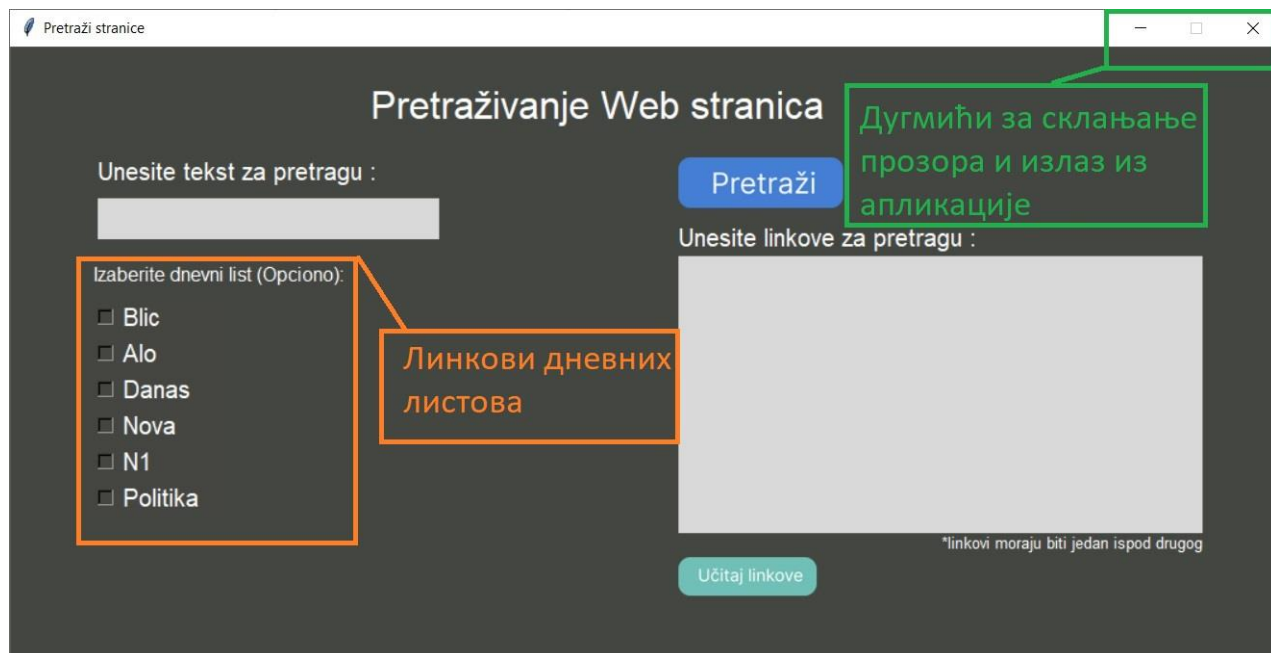
Приликом покретања апликације се учитавају линкови који су били унети у поље претраге приликом последњег напуштања апликације.

На почетном прозору се налазе и опционе веб стране дневних листова (Слика 16), које можемо изабрати кликом на назив дневног листа или квадрат поред назива дневног листа. Ове веб стране није обавезно изабрати.

У горњем десном углу главног прозора налазе се класични дугмићи за склањање прозора апликације и излаз из апликације (Слика 16).

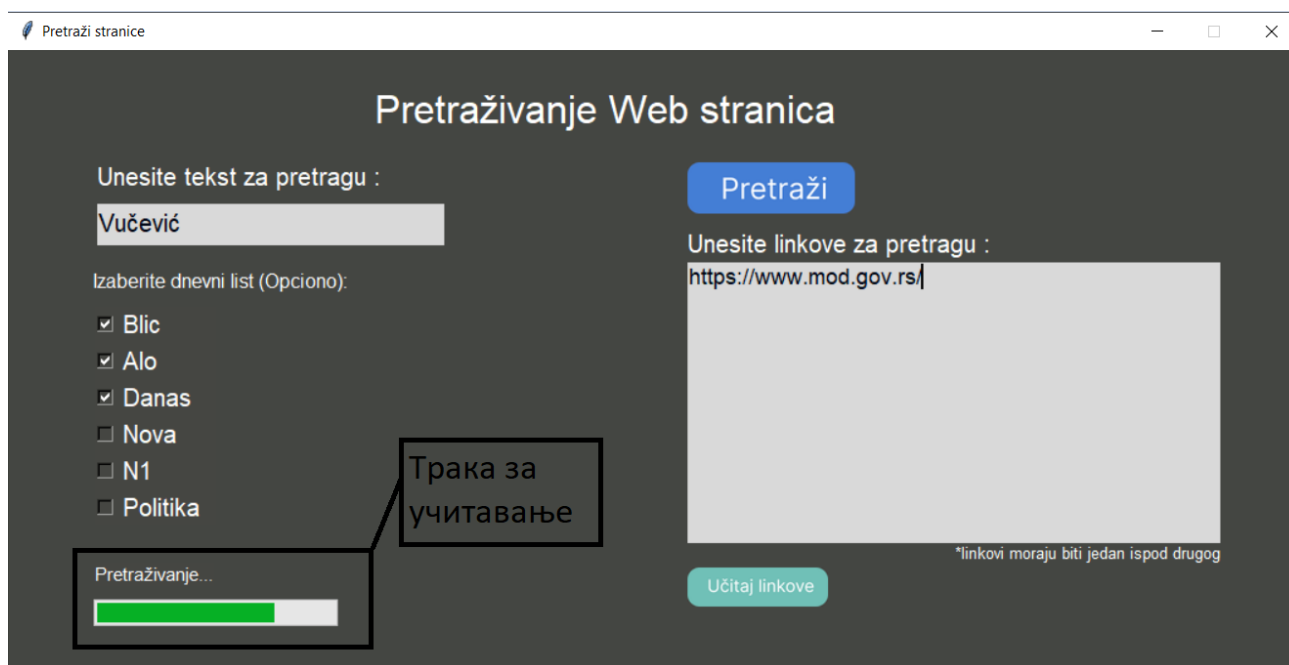


Слика 15. Дугмићи за почетак претраживања страница и учитавање линкова



Слика 16. Дугмићи за склањање прозора и излаз из апликације и линкови дневних листова

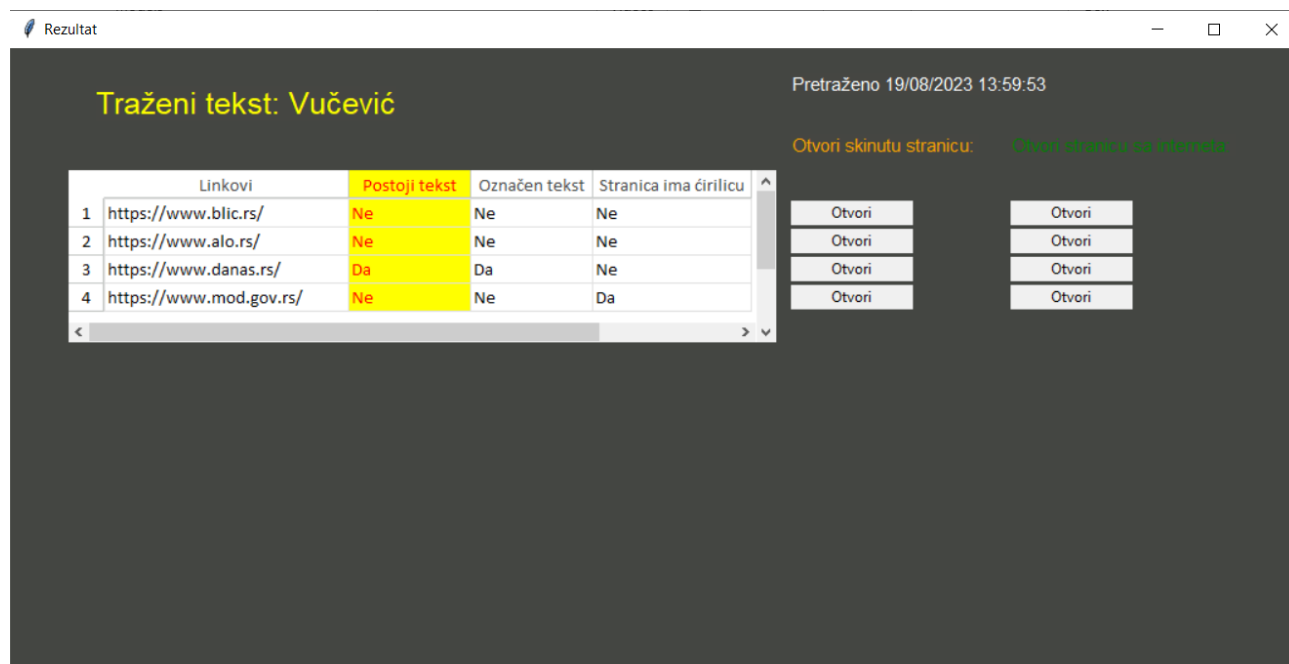
Након покретања претраживања јавља се трака за учитавање, која показује ток извршавања претраживања (Слика 17). У овом примеру се тражи текст „*Višević*“ на веб странама дневних листова „Ало“, „Блиц“, „Данас“, и на сајту министарства одбране војске србије.



Слика 17. Трака за учитавање

### 3.2 Прозор са резултатима

Прозор са резултатима садржи резултате претраге, време претраге и дугмиће за приступ веб странама. Након претраге јавља се нови прозор са резултатима претраге веб страна (Слика 18).



Слика 18. Прозор са резултатима претраге

На прозору се види тражени текст, који је у жутој боји, и датум и време претраге, који су у белој боји. Са леве стране се налази табела у којој су смештени резултати претраге. У табели постоје колоне:

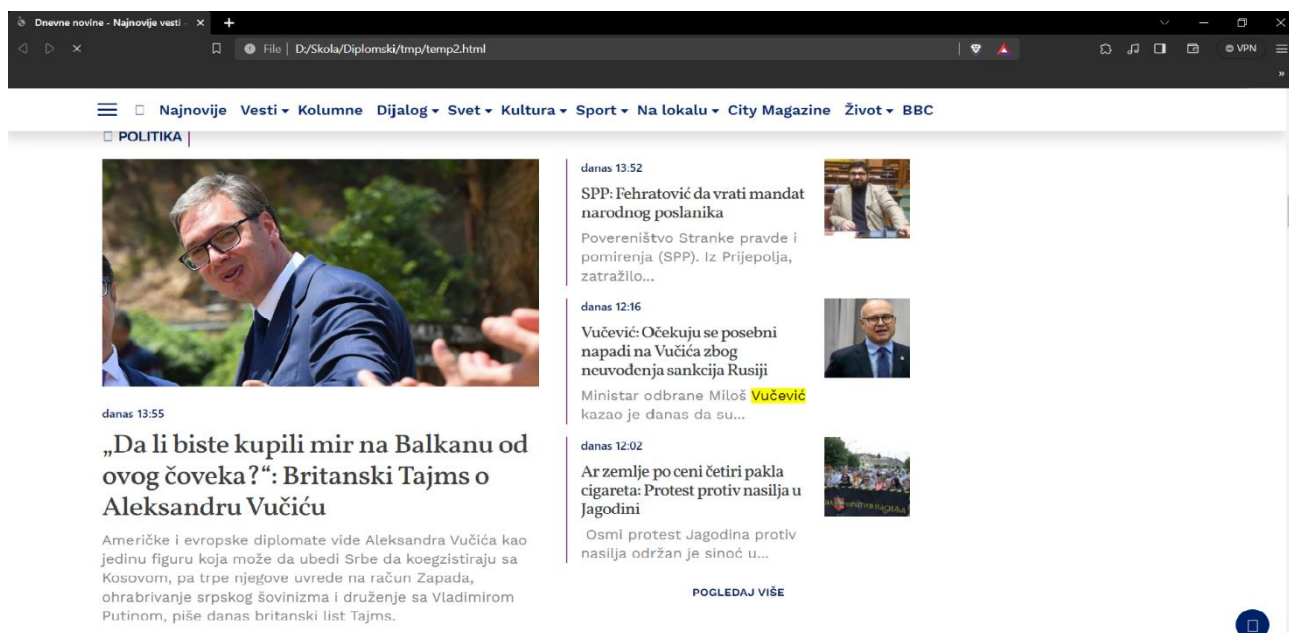
- *Linkovi* - Ова колона садржи линкове од страница које су претражене
- *Postoji tekst* - У овој колони се налази резултат претраге. Она је обојена жутом бојом, а текст у њој је у црвеној боји. У њој се могу налазити вредности „*Da*“ и „*Ne*“, што значи да ли је текст пронађен или није пронађен на веб страници, респективно.
- *Označen tekst* - Ова колона означава да ли је текст који се тражио означен на преузетој веб страни. Ова веб страна се налази на локалној машини. У колони се могу налазити вредности „*Da*“ и „*Ne*“, што значи да ли је на преузетој страни означен или није означен тражени текст, респективно.
- *Stranica ima ćirilicu* - У овој колони се налази податак да ли се на веб страни налази ћирилично писмо. Ова колона је значајна јер је могуће да је унети текст за претрагу на латиничном писму, а веб страна на ћириличном писму. Ова колона отклања забуну ако се отвори веб страна на којој је пронађен унети текст који је на латиничном писму, а сама веб страна је на ћириличном писму.

На десној страни се налазе дугмићи за отварање веб страна. Са леве стране се налази дугме за отварање веб стране која је преузета на локалну машину и на којој постоји могућност да је тражени текст означен. Са десне стране се налази дугме које отвара ту веб страну на интернет претраживачу преко интернета. Отвара се веб страна која је доступна у тренутку клика на то дугме.

Табела и дугмићи се аутоматски додају и проширују у зависности од броја веб страна за претраживање. За отварање ових веб страна користи се подразумевани интернет претраживач на лакој машини.

Овај прозор је могуће оставити са стране и извршити другу претрагу и након тога поредити резултате. Након друге претраге није више могуће приступити веб странама на локалној машини из прве претраге, јер се оне преписују са веб странама из друге претраге.

У овом примеру текст „*Vićević*“ је пронађен само у дневном листу „*Данас*“ и означен је у преузетој веб страници (Слика 19). На слици се види путања до веб стране која се налази на локалној машини у *D:* партицији.



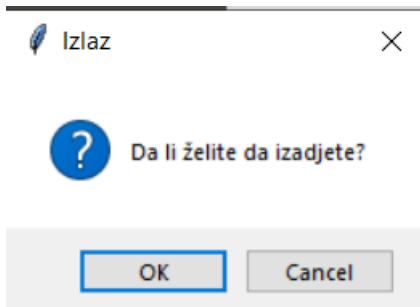
Слика 19. Претражена и означена веб страна на локалној машини

У табели са резултатима претраге (Слика 18) се такође види да веб сајт министарства одбране војске србије садржи ћирилично писмо.

Могуће је да веб стране не одговарају на захтеве приступа ако се у кратком року приступа истим веб странама и самим тим долази до грешке у претраживању. Тако се веб стране штите од могућих напада, иако то у овој апликацији није случај.

### 3.3 Прозор за потврду излаза из апликације

Приликом клика на дугме за излаз из апликације, које се налази у горњем десном углу главног прозора (Слика 16), јавља се прозор за потврду излаза из апликације (Слика 20). На њему се тражи потврда да ли корисник жели да напусти апликацију. Кликот на дугме „ОК“ корисник



Слика 20. Прозор за потврду излаза из апликације

напушта апликацију и линкови који се у том тренутку налазе у пољу се чувају. Кликом да дугме „*Cancel*“ или на дугме „X“, у горњем десном углу, наставља се рад у апликацији.

## 4 Могућности даљег проширења предложеног решења

Постојеће решење тест апликације се може унапредити у више смерова, што би увело коришћење других технологија.

Могла би се додати функционалност чувања претраге. Ту се мисли на чување претраживаних текстова и чување претражених веб страна које су на локалној машини. За ову функционалност било би потребно увести коришћење релационе или нерелационе базе података, у којој би се чували претраживани текстови и обележене веб стране са датумима претраге. Поред базе података било би потребно и изградити нови прозор, којиме би корисник приступао тој бази података. Такође би било потребно повезати базу података са апликацијом која већ постоји, ради чувања, брисања и прегледања претходних претраживања веб страница. Ова функционалност би додала могућност чувања стања веб страница у одређеним тренуцима, јер се садржај веб страна временом мења. Овим би апликација временом заузимала све већи меморијски простор, због чувања свих претражених страница.

Могуће побољшање би било да се веб стране на којима није пронађен тражени текст одмах бришу и не чувају у меморији локалне машине. У овом решењу тест апликације се све претражене веб стране чувају у меморији.

Постоји и решење да се ова апликација, која је десктоп апликација, пребаци у *Web* апликацију. Ту би се она учитавала у интернет претраживачу и одатле би се вршило претраживање страница. Ово решење захтева познавање других технологија које нису коришћене у овом раду.

## ЗАКЉУЧАК

У овом раду је коришћењем *Python* програмског језика израђена десктоп апликација. Тест апликација је софтверско решење за напредно претраживање веб страна. *Python* је релативно лак за програмирање и самим тим је стекао популарност у свету програмирања. Он је потпуно бесплатан и *open source*, што додатно доприноси његовој популарности. Постоји прегршт уграђених библиотека које олакшавају рад у њему. Такође је могуће придодати библиотеке које су нам потребне, које се могу наћи на интернету.

Циљеви овог рада су били упознавање аутора са током развоја апликације и сам развој апликације и увођење аутора у истраживачки рад и искуство у писању научних радова.

Овај рад описује апликацију која је замишљена као помоћ при претраживању веб страница са одређеним текстом. Она би честа претраживања одређених веб страна олакшала кориснику.

Ова апликација не захтева средства за њену реализацију, јер су све технологије, које су коришћене за развој ове апликације, бесплатне.



## ЛИТЕРАТУРА

- [1] <https://www.python.org/doc/essays/blurb/>
- [2] <https://www.jetbrains.com/pycharm/>
- [3] <https://docs.python.org/3/library/urllib.request.html>
- [4] <https://pypi.org/project/cyrtranslit/>
- [5] <https://pypi.org/project/regex/2023.3.22/>
- [6] <https://docs.python.org/3/library/datetime.html>
- [7] <https://docs.python.org/3/library/webbrowser.html>
- [8] <https://docs.python.org/3/library/os.html>
- [9] <https://docs.python.org/3/library/pathlib.html>
- [10] <https://docs.python.org/3/library/tkinter.html>
- [11] <https://github.com/ragardner/tksheet/wiki/Version-6>
- [12] <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [13] <https://cpl.thalesgroup.com/software-monetization/software-versioning-basics>
- [14] <https://www.atlassian.com/git/tutorials/what-is-git>
- [15] <https://www.techtarget.com/searchitoperations/definition/GitHub>

## СПИСАК СЛИКА

Слика 1. Хијерархија програмских језика.....	6
Слика 2. Разлика између компајлираних и интерпретираних програмских језика.....	7
Слика 3. Изглед <i>PyCharm IDE</i> -а.....	9
Слика 4. Провера исправности <i>e-mail</i> адресе коришћењем <i>regex</i> израза.....	11
Слика 5. Класе и атрибути класа у <i>datetime</i> библиотеци са примерима .....	12
Слика 6. Разлика у <i>Linux</i> -овом и <i>Windows</i> -овом фајл систему .....	14
Слика 7. Једноставан <i>GUI</i> прозор направљен помоћу <i>tkinter</i> библиотеке .....	15
Слика 8. Главни <i>GUI</i> прозор у различитим оперативним системима .....	16
Слика 9. <i>Tksheet</i> табела у <i>tkinter</i> прозору .....	17
Слика 10. Структура <i>HTML</i> елемента.....	19
Слика 11. Пример коришћења тага за означавање .....	19
Слика 12. Изглед платформе <i>GitHub</i> .....	21
Слика 13. Изглед главног прозора тест апликације.....	23
Слика 14. Поља за унос текста за претрагу и веб страна на којим се врши претрага .....	24
Слика 16. Дугмићи за склањање прозора и излаз из апликације и линкови дневних листова..	25
Слика 15. Дугмићи за почетак претраживања страница и учитавање линкова .....	25
Слика 17. Трака за учитавање.....	26
Слика 18. Прозор са резултатима претраге .....	26
Слика 19. Претражена и означена веб страна на локалној машини .....	28
Слика 20. Прозор за потврду излаза из апликације.....	28