**Project Report**

Title: **Bangla Plagiarism Checker**

Group Members:

1. Mehedi Hasan Raj --- 1611303042

2. Kanij Fatema ---- 1610161042

Group Number: 02

Course Code : CSE299

Section : 7

Date Of Submission : 31.08.19

**Acknowledgment:**

We have known about plagiarism checker through internet, but that was about English not for Bangla. Through a friend, named, Minhazul Islam Shanto, we have got an idea about Bangla Plagiarism Checker. He told us to do this and we are thankful to him, and special thanks to our honorable faculty Zunayeed Zahir because if he did not select it we could not implement it.

**Problem Statement:**

There are lots of available plagiarism checker software on the internet which is for English but not for Bangla language. So, when we want to check plagiarism for Bangla language or paragraph in that time, we face lots of problem and we even cannot find its proper software. To solve this problem, we have made Bangla plagiarism software which provides us a plagiarism checker that can detect Bangla plagiarism.

Table of Contents

|  |  |  |
| --- | --- | --- |
| No. | Contents | Page no. |
| 01. | Introduction | 04 |
| 02. | Literature Review | 05 |
| 03. | Timeline | 06 |
| 04. | Target User | 06 |
| 05. | User Study | 06 |
| 06. | Methodology | 07 |
| 07. | Software Requirements | 13 |
| 08. | Result | 13 |
| 09. | Conclusion | 14 |
| 10. | References | 14 |
| 11. | Appendix | 15 |

**Introduction:**

Plagiarism is the act of copying the other’s work without giving the proper Creditto them or estimating other’s work and show as yours. Now a days, Plagiarism checker software are becoming well likes because of its huge Uses. We are using this software everywhere in our life such as university, College, school and so on. In University, College, and school it is used for checking plagiarized on assignment, homework and so on. Not only for Checking plagiarized in assignment, homework even it is used for Checking plagiarized on research paper. So, it is very essential. This day, it is ready for use on internet but for English version but not for Bangla version. So if we like to check plagiarism for Bangla topics or Paragraph that time we are facing trouble and we cannot mark how much Plagiarized this topics or paragraph .To solve this puzzle out of this difficulty, we have made a software that can identify the stealing part for a given Bangla topics or paragraph from the internet and it has been an astonishing software.

**Literature Review:**

Top 10 tools or software for checking plagiarism 2019 according to this website ([www.hostingsprout.com](http://www.hostingsprout.com)). Below of these above are plagiarism tools and for English version. Using these idea or concept, we have made a Bangla plagiarism software

1. Plagiarism checker

2. Grammarly

3. Copy scape

4. White smoke

5. CopyLeaks

6. Plagiarisma

7. Plagscan

8. Plagium

9. Plagiarism

10. Unplag/Unicheck

**Timeline:**

Time that we have spent to do our project

|  |  |
| --- | --- |
| **Weeks** | **Work** |
| **1st** | **To Learn python Syntax** |
| **2nd** | **To Learn Data Structure (Queue)** |
| **3rd** | **To Make a Web Crawler (BOT)** |
| **4th** | **To Learn Beautifulsoup4 package** |
| **5th** | **To Learn Web Scraping** |
| **6th** | **To Learn Graphical User Interface Package PyQt5** |
| **7th** | **To Marge the Whole Project** |

**Target Users:**

In our Software, there are no specific users but we are mainly considering those users whose need to check plagiarism on various research paper, assignment. We are mainly targeting students, teachers, and various users.

**User Study:**

It is important to user study because our product is software based and for developing it is needed. How much user friendly our product is we can know. By User Study we can improve efficiency. But we have no features for user study for time limitation. We will develop in future and we will add this feature by rating option zero to five.

**Methodology:**

1. **Python:** As we have done our whole project in python language so we had to

Learn python language. For learning python, we have used [www.learnpython.org](http://www.learnpython.org) website. We choose python language for our project because of some advantages, and these are given below

**i. Readable and Maintainable Code:**

While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

**ii. Multiple Programming Paradigms:**

Like other modern programming languages, Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

**iii. Compatible with Major Platforms and Systems:**

At present, Python is supported many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

**iv. Robust Standard Library:**

Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

**v. Many Open Source Frameworks and Tools:**

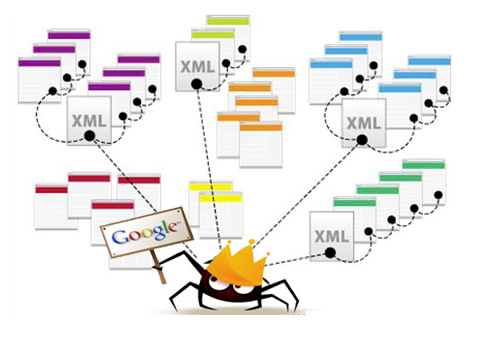
As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, you can accelerate desktop GUI application development using python GUI framwork and toolkits like PyQT5, PyJs, PyGUI, Kivy, PyGTK and WxPython.

**vi. Simplify Complex Software Development:**

Python is a general purposes programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many python developers even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

**2.Web Crawler:**

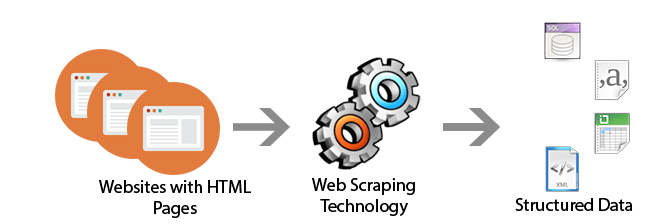
A Web crawler, sometimes called a spider or spider bot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing. – By searching from Wikipedia. A Crawler is an automated Script which means all of its actions are predefined. It is a program that visits different websites and reads their Web pages and other information in order to create entries for a Search Engine Index. – By searching from google. Firstly, we have made a web crawler. There is a spider robot who traverses one web page to another page in other way, we can say, it traverses one link to another link from a base link. In our software there are two list and their names are 1. links and 2. Traversed link. In link list we have kept that type of website’s list that need to be traversed that means links which have not been traversed yet , and another list named traversed list where we have kept that type of links that has been traversed because we do not want to traverse a link which already been traversed.

 Figure no 1: Web Crawler Concept

**3.Web Scraping:**

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. --By searching from Wikipedia.

So far, we discussed about web crawler. We have traversed the website by web crawler. For our project we have extracted data from website while traversing. This is why we should have the knowledge of web scraping. There are lots of python built in packages for web scraping and BeautifulSoup4 is one of them. This package is fantastic package for web scraping and easy to use. By web scraping we have got data from website and we have cleaned it to use because there are lots of unnecessary things are available too, we should clean it. Then we can use our input data for comparing with clean data and thus we have got result.

 Figure No 2: Web Scraping Concept

**4.Graphical User Interface:**

We have made a desktop application So, we need to have a user interface so that user can feel comfortable to use it. Python has a package named PyQt5. PyQt5 is used for making graphical user interface (GUI). We have made our graphical user interface by the help of PyQt5 package. Graphical use interface’s demo is given below

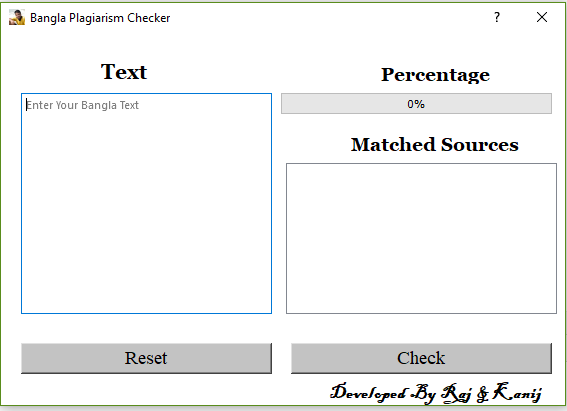


Figure: Graphical User Interface (GUI)

**Software Requirement/Need to Have Knowledge:**

1. Python Basic Knowledge specially syntax
2. Web Crawler knowledge and spider making knowledge
3. Web Scraping Concept
4. IDE PyCharm Professional 2009
5. Queue Data Structure Concept
6. User Interface Package PyQt5
7. Web Scrapping Package Beautifulsoup4
8. Internet connection

**Result:**

Our Bangla Plagiarism checker software is a plagiarism checker where taking Bangla sentence or paragraph as input and it matches how much plagiarism are there. Finally gives two result one is how much percentages have been copied and second is matched sources where from data is copied.

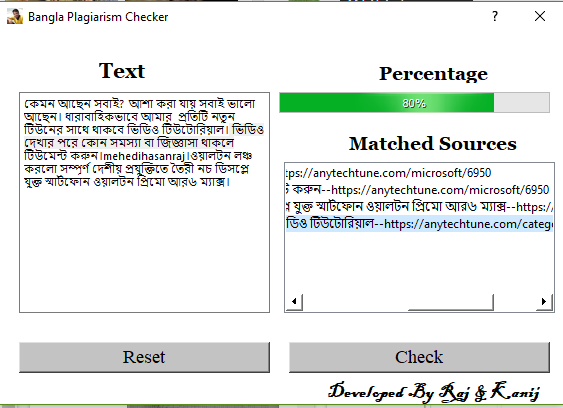


Figure: Sample result for text value

**Conclusion:**

Our project is not much big and it takes some times to run but it gives the result perfectly, and it has some unique features like matched sources that makes it awesome. The main point is that we get result for Bangla text.

**References:**

1. <https://www.learnpython.org/>
2. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
3. <https://pypi.org/project/PyQt5/>
4. <https://www.geeksforgeeks.org/stack-queue-python-using-module-queue/>

**Appendix:**

**Code:** This is code for our project

**Main Module:**

**import** Window **as** window *#for importing function and classes from Windows file***import** WebSystem  
**import** sys  
  
*# Running Main traversing and searching data  
#---------------------------------------------------------------------------------------------------------------------  
#---------------------------------------------------------------------------------------------------------------------  
# Showing windows  
#-------------------------------------------------------------------------------------------------------------------*app = window.QApplication(window.sys.argv)  
desktop\_window = window.ApplicationWindow()  
desktop\_window.show()  
  
**def** check\_clicked():  
  
 input\_data\_from\_desktop\_window = desktop\_window.getInputValue() *#getting input value from window* input\_with\_space = input\_data\_from\_desktop\_window.split(**'।'**)  
 inputs =[]  
 **for** new **in** input\_with\_space:  
 **if** new!=**''**:  
 inputs.append(new)  
  
 *#print(inputs)* url = **"https://anytechtune.com/"** *# this is base url and it can be anything* links =WebSystem.link\_Finder(url)  
 links.append(url)  
 counter =0  
 desktop\_window.listWidget.clear()  
 desktop\_window.progressBar.setProperty(**'value'**, 0)  
 input\_length = len(inputs)  
  
 **for** link **in** links:  
 contents =WebSystem.content\_finder(link)  
 **for** content **in** contents:  
 **for** input **in** inputs:  
 **if** len(inputs) == 0:  
 **break  
 if** input **in** content **or** content **in** input:  
 counter = counter+1  
 x =str(counter)+**'. '**+input+**'--'**+link  
 inputs.remove(input)  
 desktop\_window.listWidget.addItem(x)  
 **if** len(inputs)==0:  
 **break  
 if** len(inputs) == 0:  
 **break  
 if** len(inputs) == 0:  
 **break** result =int(((input\_length - len(inputs))/input\_length)\*100)  
 *#print(result)* desktop\_window.progressBar.setProperty(**'value'**,result)  
  
  
  
  
  
desktop\_window.check.clicked.connect(check\_clicked)  
sys.exit(app.exec())

**WebSystem Module:**

**from** urllib.request **import** urlopen  
  
**import** requests  
**from** bs4 **import** BeautifulSoup  
  
  
 *# Finding links from a website  
 #------------------------------------------------------------------------------------------------------------------------***def** link\_Finder(url):  
 html = urlopen(url)  
 html = html.read()  
 page = html.decode(**"UTF-8"**)  
 *# print(page)* soup = BeautifulSoup(page, **'html.parser'**)  
  
 href\_links\_list = set()  
 clean\_links = []  
  
 *#print(beautify\_contents.find\_all('p'))  
  
 #finding href contents with https links and none https link* **for** link **in** soup.find\_all(**'a'**):  
 x = link.get(**'href'**)  
 href\_links\_list.add(link.get(**'href'**))  
  
 *#this is for removing link without https contents* **for** newlist **in** href\_links\_list:  
 **if** newlist[0] == **'h' and** newlist[1] == **'t' and** newlist[2] == **'t' and** newlist[3] == **'p' and** newlist[4] == **'s' and 'facebook.com' not in** newlist **and 'youtube.com' not in** newlist:  
 clean\_links.append(newlist)  
  
 **return** clean\_links  
  
  
  
  
  
*# finding content from website  
#--------------------------------------------------------------------------------------------------------------------------------***def** content\_finder(url):  
 all\_html\_content = requests.get(url) *# all source code contents* beautify\_contents = BeautifulSoup(all\_html\_content.content,**'html.parser'**) *# for simplifying sourcecode and operation* dirty\_contents = []  
 clean\_contents = []  
  
  
 *# finding href contents with https links and none https link* **for** link **in** beautify\_contents.find\_all(**'p'**):  
 dirty\_contents.append(link.text)  
  
 *#removing empty contents from list* **for** data **in** dirty\_contents:  
 **if** data !=**''**:  
 clean\_contents.append(data)  
  
 **return** clean\_contents  
  
  
  
  
  
  
  
  
  
**'''  
url = 'https://anytechtune.com/'  
links = link\_Finder(url)  
  
links.append(url)  
x =input()  
  
for link in links:  
 contents = content\_finder(link)  
 #print('--------------------------data for this url-----'+link+' --------------------')  
 for content in contents:  
 if x in content or content in x:  
 print('got '+x+' 111'+content)  
 #print(contents)  
  
'''**

**UserInterface Module:**

*# -\*- coding: utf-8 -\*-  
  
# Form implementation generated from reading ui file 'interface.ui'  
#  
# Created by: PyQt5 UI code generator 5.13.0  
#  
# WARNING! All changes made in this file will be lost!***from** PyQt5 **import** QtCore, QtGui, QtWidgets  
  
  
**class** Ui\_windows(object):  
 **def** get\_input\_data(self):  
 x = self.inputText.toPlainText() *# this for getting input data from input box  
 #print(x)* **return** x  
 **def** setupUi(self, windows):  
 windows.setObjectName(**"windows"**)  
  
 windows.resize(564, 372)  
 icon = QtGui.QIcon()  
 icon.addPixmap(QtGui.QPixmap(**"c.ico"**), QtGui.QIcon.Normal, QtGui.QIcon.Off)  
 windows.setWindowIcon(icon)  
 windows.setAutoFillBackground(**False**)  
 windows.setStyleSheet(**"background-color: rgb(255, 255, 255);"**)  
 windows.setSizeGripEnabled(**False**)  
 self.inputText = QtWidgets.QPlainTextEdit(windows)  
 self.inputText.setGeometry(QtCore.QRect(20, 60, 251, 221))  
 font = QtGui.QFont()  
 font.setFamily(**"Nirmala UI"**)  
 self.inputText.setFont(font)  
 self.inputText.setObjectName(**"inputText"**)  
 self.text = QtWidgets.QLabel(windows)  
 self.text.setGeometry(QtCore.QRect(100, 30, 51, 16))  
 font = QtGui.QFont()  
 font.setFamily(**"Georgia"**)  
 font.setPointSize(16)  
 font.setBold(**True**)  
 font.setWeight(75)  
 self.text.setFont(font)  
 self.text.setObjectName(**"text"**)  
 self.percentage = QtWidgets.QLabel(windows)  
 self.percentage.setGeometry(QtCore.QRect(380, 30, 121, 21))  
 font = QtGui.QFont()  
 font.setFamily(**"Georgia"**)  
 font.setPointSize(14)  
 font.setBold(**True**)  
 font.setWeight(75)  
 self.percentage.setFont(font)  
 self.percentage.setObjectName(**"percentage"**)  
 self.label\_3 = QtWidgets.QLabel(windows)  
 self.label\_3.setGeometry(QtCore.QRect(350, 100, 181, 20))  
 font = QtGui.QFont()  
 font.setFamily(**"Georgia"**)  
 font.setPointSize(14)  
 font.setBold(**True**)  
 font.setWeight(75)  
 self.label\_3.setFont(font)  
 self.label\_3.setObjectName(**"label\_3"**)  
 self.label\_4 = QtWidgets.QLabel(windows)  
 self.label\_4.setGeometry(QtCore.QRect(310, 340, 231, 41))  
 self.label\_4.setStyleSheet(**"font: 14pt \"Blackadder ITC\";\n"  
"color: rgb(0, 0, 0);"**)  
 self.label\_4.setObjectName(**"label\_4"**)  
 self.reset = QtWidgets.QPushButton(windows)  
 self.reset.setGeometry(QtCore.QRect(20, 310, 251, 31))  
 font = QtGui.QFont()  
 font.setFamily(**"Times New Roman"**)  
 font.setPointSize(14)  
 font.setBold(**False**)  
 font.setWeight(50)  
 self.reset.setFont(font)  
 self.reset.setAutoFillBackground(**False**)  
 self.reset.setStyleSheet(**"background-color: rgb(195, 195, 195);"**)  
 self.reset.setObjectName(**"reset"**)  
 self.check = QtWidgets.QPushButton(windows)  
 self.check.clicked.connect(self.get\_input\_data)  
 self.check.setGeometry(QtCore.QRect(290, 310, 261, 31))  
 font = QtGui.QFont()  
 font.setFamily(**"Times New Roman"**)  
 font.setPointSize(14)  
 self.check.setFont(font)  
 self.check.setStyleSheet(**"background-color: rgb(195, 195, 195);"**)  
 self.check.setObjectName(**"check"**)  
 self.progressBar = QtWidgets.QProgressBar(windows)  
 self.progressBar.setGeometry(QtCore.QRect(280, 60, 271, 21))  
 self.progressBar.setLayoutDirection(QtCore.Qt.LeftToRight)  
 self.progressBar.setProperty(**"value"**, 24)  
 self.progressBar.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignVCenter)  
 self.progressBar.setInvertedAppearance(**False**)  
 self.progressBar.setObjectName(**"progressBar"**)  
 self.listWidget = QtWidgets.QListWidget(windows)  
 self.listWidget.setGeometry(QtCore.QRect(285, 130, 271, 151))  
 self.listWidget.setObjectName(**"listView"**)  
 self.retranslateUi(windows)  
 self.reset.clicked.connect(self.inputText.clear)  
 self.reset.clicked.connect(self.progressBar.reset)*#new edit* self.reset.clicked.connect(self.listWidget.clear)*#new edit* QtCore.QMetaObject.connectSlotsByName(windows)  
  
 **def** retranslateUi(self, windows):  
 \_translate = QtCore.QCoreApplication.translate  
 windows.setWindowTitle(\_translate(**"windows"**, **"Bangla Plagiarism Checker"**))  
 self.inputText.setPlaceholderText(\_translate(**"windows"**, **"Enter Your Bangla Text"**))  
 self.text.setText(\_translate(**"windows"**, **"Text"**))  
 self.percentage.setText(\_translate(**"windows"**, **"Percentage"**))  
 self.label\_3.setText(\_translate(**"windows"**, **"Matched Sources"**))  
 self.label\_4.setText(\_translate(**"windows"**, **"<html><head/><body><p align=\"right\"><span style=\" font-weight:600;\">Developed By Raj &amp; Kanij</span></p></body></html>"**))  
 self.reset.setText(\_translate(**"windows"**, **"Reset"**))  
 self.check.setText(\_translate(**"windows"**, **"Check"**))  
  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 **import** sys  
 app = QtWidgets.QApplication(sys.argv)  
 windows = QtWidgets.QDialog()  
 ui = Ui\_windows()  
 ui.setupUi(windows)  
 windows.show()  
 sys.exit(app.exec\_())

**Window Module:**

**from** PyQt5.QtCore **import** \*  
**from** PyQt5.QtGui **import** \*  
**from** PyQt5.QtWidgets **import** \*  
**from** PyQt5 **import** QtWidgets  
**import** sys  
  
**import** UserInterface  
  
**class** ApplicationWindow(UserInterface.Ui\_windows,QDialog):  
  
 *# set progress value* **def** setProgressValue(self,new\_value):  
 self.progressBar.setProperty(**"value"**,new\_value)  
  
 **def** getInputValue(self):  
 x = self.inputText.toPlainText() *# this for getting input data from* **return** x  
  
  
 **def** \_\_init\_\_(self):  
 QDialog.\_\_init\_\_(self)  
 self.setupUi(self)  
 self.progressBar.setProperty(**"value"**,0)  
  
  
  
  
  
**'''  
  
for showing windows  
------------------  
 app = QApplication(sys.argv)  
 interface = ApplicationWindow()  
 #interface.check.clicked.connect(method) for click option  
 interface.show()  
 app.exec\_()  
  
'''**