**PLAGIARISM DETECTOR**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

# BACHELOR OF ENGINEERING

IN

# INFORMATION TECHNOLOGY

By

#### Roshini Peddaboina 1602-20-737-032

**Zoha Tabassum 1602-20-737-0053**

**Deeksha Gaddam 1602-20-737-010**



# Department of Information Technology Vasavi College of Engineering (Autonomous)

**ACCREDITED BY NAAC WITH 'A++' GRADE**

# (Affiliated to Osmania University and Approved by AICTE) Ibrahim Bagh, Hyderabad-31

**2022**

# Vasavi College of Engineering (Autonomous)

**ACCREDITED BY NAAC WITH 'A++' GRADE**

# (Affiliated to Osmania University and Approved by AICTE) Hyderabad-500 031

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

#### We, Roshini Peddaboina, Zoha Tabassumand Deeksha Gaddam,

bearing hall ticket numbers, **1602-20-737-32,1602-20-737-053** and **1602-20-737-010**, hereby declare that the project report entitled **"PLAGIARISM DETECTOR"** is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**Roshini Peddaboina**

**1602-20-737-032**

**Zoha Tabassum**

**1602-20-737-053**

**Deeksha gaddam**

**1602-20-737-010**

(Faculty In-Charge) (Head,Dept of IT)

**ACKNOWLEDGMENT**

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement.

We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project.

We also want to thank and convey our gratitude towards our mini project coordinators Divya Lingineni and RajyaLakshmi, for guiding us in understanding the process of project development & giving us timely suggestions at every phase.

We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions

**ABSTRACT**

The main objective of the project is to identify content similarity matches. Here checked plagiarism to between the files. We checked plagiarism for 2 files that is we have shown the plagiarism in percentage and also plagiarism between 3 files there we have shown percentages between file1, file2,file 2,file 3 and file 1 ,file 3

The framework which was used in this project was GUI using python and also using kivy.

**TABLE OF CONTENTS**

**1.Introduction**

a. Information

b. Features

**2.Technology**

a. Software Requirements

b. Hardware Requirements

**3.Proposed Work**

a. Design

b. Implementation

c. Testing

**4.Results**

**5.Outcomes apart from curriculum**

**6.Conclusion and future work**

**7. References**

**INTRODUCTION**

1. **Information**

The main objective of the project is to identify content similarity matches. Here checked plagiarism to between the files. We checked plagiarism for 2 files that is we have shown the plagiarism in percentage and also plagiarism between 3 files there we have shown percentages between file1, file2, file 2, file 3 and file 1 , file 3

The framework which was used in this project was GUI using python and also using kivy.

1. **Motivation**

The motivation behind this project is to reduce the burden for teaching staff checking student work prior to submission.

The approach encouraged more experimental learning.

**c. Features**

● In this First the user need to register to the portal.

●Then user need to give the credentials to login to the portal.

●Then the portal will open and the user will be given drag and drop option to drop files.

● It will calculate plagiarism and displays the percentage of plagiarism.

**Technology**

**a. Hardware Requirements**

The hardware components used in order to create the game are as follows

* Computer with 2GB RAM
* Internet
* External drives for backup

**b. Software Requirements**

* We used kivy and python in order to develop the code for the plagiarism detector.
* We used Sequence Matcher from difflib (which is python inbuilt library)to calculate plagiarism from the files.
* Operating System. We have chosen Windows operating system for its best support and user friendliness.

**PROPOSED WORK**

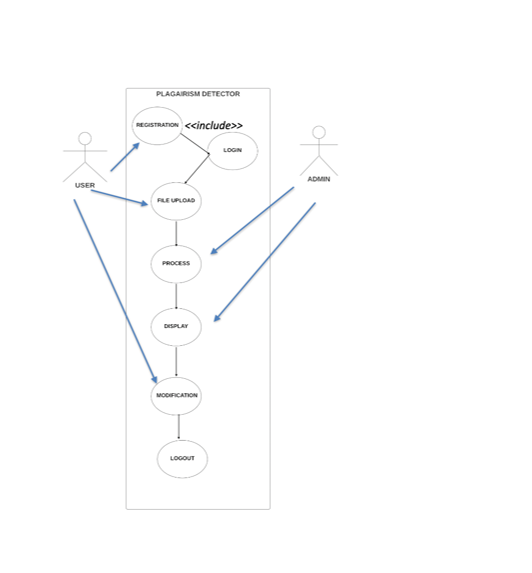
**a. Design**

**Use Case Diagram**

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

**Use Case Diagram**



**DESCRIPTION**

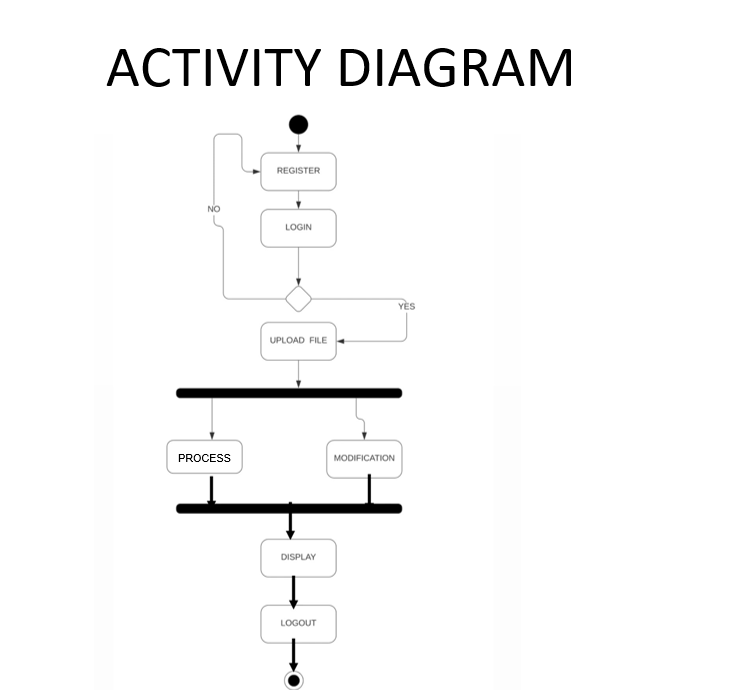
**User**

The User need to Register to the portal and then Login to the portal then the user will be given chance to drag and drop files.

**System**

Once the user successfully drag and drop files then Plagiarism will be displayed.

**ACTIVITY DIAGRAM**

****

**MODULE WISE CODE**

**from kivy.lang import Builder**

**from kivymd.app import MDApp**

**from kivy.core.window import Window**

**from kivy.uix.screenmanager import ScreenManager, Screen**

**from difflib import SequenceMatcher**

**class Login(Screen):**

**pass**

**class Signup(Screen):**

**pass**

**class Dashboard(Screen):**

**pass**

**class Result(Screen):**

**pass**

**string="""**

**<Login>:**

**MDLabel:**

**text: "Plagiarism Detection"**

**font\_size: 50**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDLabel:**

**text: "LOGIN"**

**font\_size: 35**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.70}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDTextField:**

**id: email**

**hint\_text: "Email"**

**size\_hint\_x: 0.9**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.60}**

**required: True**

**MDTextField:**

**id: password**

**hint\_text: "Password"**

**password: True**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.45}**

**MDLabel:**

**text: "Create new account"**

**font\_size: 20**

**pos\_hint: {"center\_x": 0.65, "center\_y": 0.35}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDLabel:**

**id: err**

**text: ""**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**color: 1,0,0,1**

**pos\_hint: {"center\_x": 0.83, "center\_y": 0.25}**

**MDRaisedButton:**

**text: "Sign Up"**

**md\_bg\_color: 1, 0, 1, 1**

**pos\_hint: {"center\_x": 0.8, "center\_y": 0.35}**

**font\_size: 20**

**on\_press: app.signup()**

**MDRoundFlatButton:**

**text: "Login"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.15}**

**size\_hint\_x: 0.7**

**on\_press: app.login(email.text,password.text)**

**<Signup>:**

**MDLabel:**

**text: "SIGN UP"**

**font\_size: 35**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDTextField:**

**id: email**

**hint\_text: "Email"**

**size\_hint\_x: 0.9**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.8}**

**required: True**

**MDTextField:**

**id: password**

**hint\_text: "Password"**

**password: True**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.65}**

**MDTextField:**

**id: password1**

**hint\_text: "Re-Enter password"**

**password: True**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}**

**MDLabel:**

**id: err**

**text: ""**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**color: 1,0,0,1**

**pos\_hint: {"center\_x": 0.83, "center\_y": 0.4}**

**MDRoundFlatButton:**

**text: "Sign up"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.3}**

**size\_hint\_x: 0.7**

**on\_press: app.createaccount(email.text,password.text,password1.text)**

**MDRoundFlatButton:**

**text: "Go Back"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.08}**

**size\_hint\_x: 0.8**

**on\_press: app.goback()**

**<Result>:**

**MDLabel:**

**text: "Result"**

**font\_size: 40**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDLabel:**

**id: result**

**text: "Plagiarism is :"**

**font\_size: 40**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.7}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDRoundFlatButton:**

**text: "Go Back"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.08}**

**size\_hint\_x: 0.8**

**on\_press: app.goback()**

**<Dashboard>:**

**MDLabel:**

**text: "Dashboard"**

**font\_size: 40**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDRoundFlatButton:**

**text: "Drag and Drop the required 2 text files."**

**font\_size: 30**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.50}**

**size: 75, 50**

**size\_hint: 50,50**

**on\_press: app.file\_manager\_open()**

**MDRoundFlatButton:**

**text: "Go Back"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.08}**

**size\_hint\_x: 0.8**

**on\_press: app.goback()**

**"""**

**Builder.load\_string(string)**

**sm = ScreenManager()**

**Window.size = (1000, 600)**

**c=0**

**l=[]**

**class MainApp(MDApp):**

**def build(self):**

**self.theme\_cls.theme\_style = "Light"**

**self.theme\_cls.primary\_palette = "BlueGray"**

**sm.add\_widget(Login(name='Login'))**

**sm.add\_widget(Signup(name='Signup'))**

**sm.add\_widget(Dashboard(name='Dashboard'))**

**sm.add\_widget(Result(name='Result'))**

**sm.current = 'Login'**

**return sm**

**def signup(self):**

**sm.current='Signup'**

**def login(self,email,password):**

**s = f"Email: {email}\nPassword: {password}\n"**

**if email=="admin@gmail.com" and password=="admin":**

**print("successful\n"+s)**

**sm.current='Dashboard'**

**self.file\_manager\_open()**

**else:**

**print("failed\n"+s)**

**sm.get\_screen('Login').ids.err.text="Wrong email / password"**

**def goback(self):**

**sm.current='Login'**

**def createaccount(self,email,password,password1):**

**if password==password1 and password!='':**

**sm.get\_screen('Signup').ids.err.text=""**

**s = f"Email: {email}\nPassword: {password}\n"**

**print("created: \n"+s)**

**sm.current='Login'**

**else:**

**print("Password mismatch")**

**sm.get\_screen('Signup').ids.err.text="Password mismatch"**

**def file\_manager\_open(self):**

**Window.bind(on\_dropfile=self.\_on\_file\_drop)**

**def \_on\_file\_drop(self, window, file\_path):**

**global c**

**global l**

**print(file\_path)**

**if c<2:**

**c+=1**

**l.append(str(file\_path))**

**if c==2:**

**x=l[0][2:-1].split('\\')**

**x=x[::2]**

**file\_path1="/".join(x)**

**with open(file\_path1) as f1:**

**data1 = f1.read()**

**y=l[1][2:-1].split('\\')**

**y=y[::2]**

**file\_path2="/".join(y)**

**with open(file\_path2) as f2:**

**data2 = f2.read()**

**similarity\_ratio = SequenceMatcher(None,data1,data2).ratio()**

**print(similarity\_ratio\*100 )**

**c=0**

**l=[]**

**sm.current='Result'**

**sm.get\_screen('Result').ids.result.text="Plagiarism is : "+ str(similarity\_ratio\*100) +" %"**

**print("success")**

**sm.current='Result'**

**return**

**MainApp().run()**

**PLAGIARISM DETECTOR FOR 3 FILES CODE**

**from kivy.lang import Builder**

**from kivymd.app import MDApp**

**from kivy.core.window import Window**

**from kivy.uix.screenmanager import ScreenManager, Screen**

**from difflib import SequenceMatcher**

**class Login(Screen):**

**pass**

**class Signup(Screen):**

**pass**

**class Dashboard(Screen):**

**pass**

**class Result(Screen):**

**pass**

**string="""**

**<Login>:**

**MDLabel:**

**text: "Plagiarism Detection"**

**font\_size: 50**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDLabel:**

**text: "LOGIN"**

**font\_size: 35**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.70}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDTextField:**

**id: email**

**hint\_text: "Email"**

**size\_hint\_x: 0.9**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.60}**

**required: True**

**MDTextField:**

**id: password**

**hint\_text: "Password"**

**password: True**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.45}**

**MDLabel:**

**text: "Create new account"**

**font\_size: 20**

**pos\_hint: {"center\_x": 0.65, "center\_y": 0.35}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDLabel:**

**id: err**

**text: ""**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**color: 1,0,0,1**

**pos\_hint: {"center\_x": 0.83, "center\_y": 0.25}**

**MDRaisedButton:**

**text: "Sign Up"**

**md\_bg\_color: 1, 0, 1, 1**

**pos\_hint: {"center\_x": 0.8, "center\_y": 0.35}**

**font\_size: 20**

**on\_press: app.signup()**

**MDRoundFlatButton:**

**text: "Login"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.15}**

**size\_hint\_x: 0.7**

**on\_press: app.login(email.text,password.text)**

**<Signup>:**

**MDLabel:**

**text: "SIGN UP"**

**font\_size: 35**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDTextField:**

**id: email**

**hint\_text: "Email"**

**size\_hint\_x: 0.9**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.8}**

**required: True**

**MDTextField:**

**id: password**

**hint\_text: "Password"**

**password: True**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.65}**

**MDTextField:**

**id: password1**

**hint\_text: "Re-Enter password"**

**password: True**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}**

**MDLabel:**

**id: err**

**text: ""**

**size\_hint\_x: 0.9**

**required: True**

**font\_size: 25**

**color: 1,0,0,1**

**pos\_hint: {"center\_x": 0.83, "center\_y": 0.4}**

**MDRoundFlatButton:**

**text: "Sign up"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.3}**

**size\_hint\_x: 0.7**

**on\_press: app.createaccount(email.text,password.text,password1.text)**

**MDRoundFlatButton:**

**text: "Go Back"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.08}**

**size\_hint\_x: 0.8**

**on\_press: app.goback()**

**<Result>:**

**MDLabel:**

**text: "Result"**

**font\_size: 40**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDLabel:**

**id: result**

**text: "Plagiarism is :"**

**font\_size: 40**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.7}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDRoundFlatButton:**

**text: "Go Back"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.08}**

**size\_hint\_x: 0.8**

**on\_press: app.goback()**

**<Dashboard>:**

**MDLabel:**

**text: "Dashboard"**

**font\_size: 40**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}**

**halign: 'center'**

**size\_hint\_y: None**

**padding\_y: 15**

**MDRoundFlatButton:**

**text: "Drag and Drop the required 3 text files."**

**font\_size: 30**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.50}**

**size: 75, 50**

**size\_hint: 50,50**

**on\_press: app.file\_manager\_open()**

**MDRoundFlatButton:**

**text: "Go Back"**

**font\_size: 25**

**pos\_hint: {"center\_x": 0.5, "center\_y": 0.08}**

**size\_hint\_x: 0.8**

**on\_press: app.goback()**

**"""**

**Builder.load\_string(string)**

**sm = ScreenManager()**

**Window.size = (1000, 600)**

**c=0**

**l=[]**

**class MainApp(MDApp):**

**def build(self):**

**self.theme\_cls.theme\_style = "Light"**

**self.theme\_cls.primary\_palette = "BlueGray"**

**sm.add\_widget(Login(name='Login'))**

**sm.add\_widget(Signup(name='Signup'))**

**sm.add\_widget(Dashboard(name='Dashboard'))**

**sm.add\_widget(Result(name='Result'))**

**sm.current = 'Login'**

**return sm**

**def signup(self):**

**sm.current='Signup'**

**def login(self,email,password):**

**s = f"Email: {email}\nPassword: {password}\n"**

**if email=="admin@gmail.com" and password=="admin":**

**print("successful\n"+s)**

**sm.current='Dashboard'**

**self.file\_manager\_open()**

**else:**

**print("failed\n"+s)**

**sm.get\_screen('Login').ids.err.text="Wrong email / password"**

**def goback(self):**

**sm.current='Login'**

**def createaccount(self,email,password,password1):**

**if password==password1 and password!='':**

**sm.get\_screen('Signup').ids.err.text=""**

**s = f"Email: {email}\nPassword: {password}\n"**

**print("created: \n"+s)**

**sm.current='Login'**

**else:**

**print("Password mismatch")**

**sm.get\_screen('Signup').ids.err.text="Password mismatch"**

**def file\_manager\_open(self):**

**Window.bind(on\_dropfile=self.\_on\_file\_drop)**

**def \_on\_file\_drop(self, window, file\_path):**

**global c**

**global l**

**print(file\_path)**

**if c<3:**

**c+=1**

**l.append(str(file\_path))**

**if c==3:**

**print(l)**

**x=l[0][2:-1].split('\\')**

**x=x[::2]**

**file\_path1="/".join(x)**

**with open(file\_path1) as f1:**

**data1 = f1.read()**

**y=l[1][2:-1].split('\\')**

**y=y[::2]**

**file\_path2="/".join(y)**

**with open(file\_path2) as f2:**

**data2 = f2.read()**

**z=l[2][2:-1].split('\\')**

**z=z[::2]**

**file\_path3="/".join(z)**

**with open(file\_path3) as f3:**

**data3 = f3.read()**

**similarity\_ratio1 = SequenceMatcher(None,data1,data2).ratio()**

**similarity\_ratio2 = SequenceMatcher(None,data1,data3).ratio()**

**similarity\_ratio3 = SequenceMatcher(None,data2,data3).ratio()**

**#print(similarity\_ratio\*100 )**

**c=0**

**l=[]**

**sm.current='Result'**

**sample= "file 1 - 2 : " + str(similarity\_ratio1\*100)+"%\n"+"file 1 - 3 : " + str(similarity\_ratio2\*100)+"%\n"+"file 2 - 3 : " + str(similarity\_ratio3\*100)+"%"**

**sm.get\_screen('Result').ids.result.text="Plagiarism is : \n\n"+ sample**

**print("success")**

**sm.current='Result'**

**return**

**MainApp().run()**

**c. Testing**

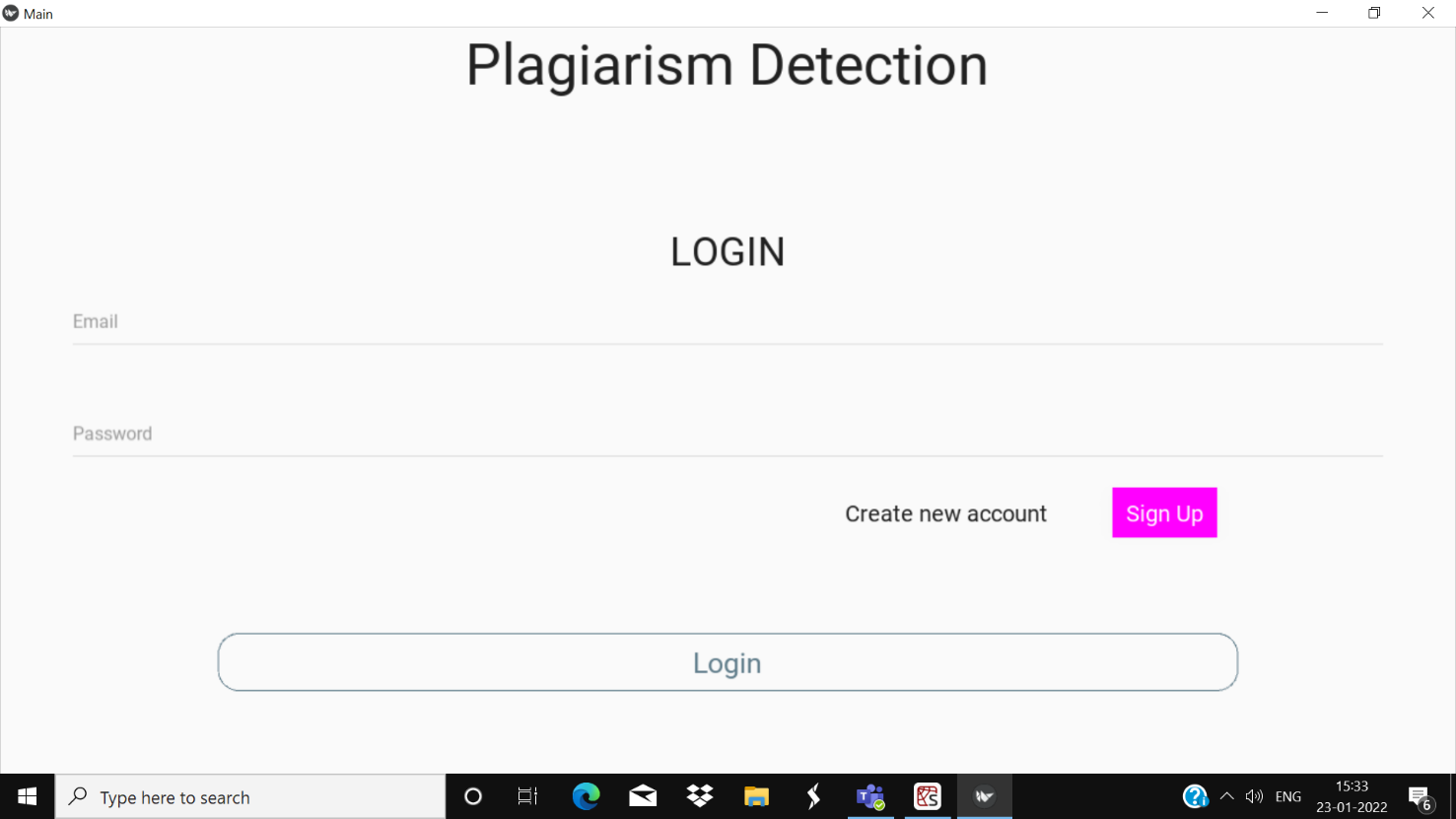
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC01 | | **Use Case ID:** UC01 | |
| **Test Case Title:** Registration | |  | |
| **Test Case Description:**  Verify whether user is Registered to the portal. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.First the portal is displayed.  2. The user need to give the Credentials.  3.The system verifies the user is Registered or not | The user gets access to the portal if the user is already registered | | The user successfully accessed the portal  (pass). |

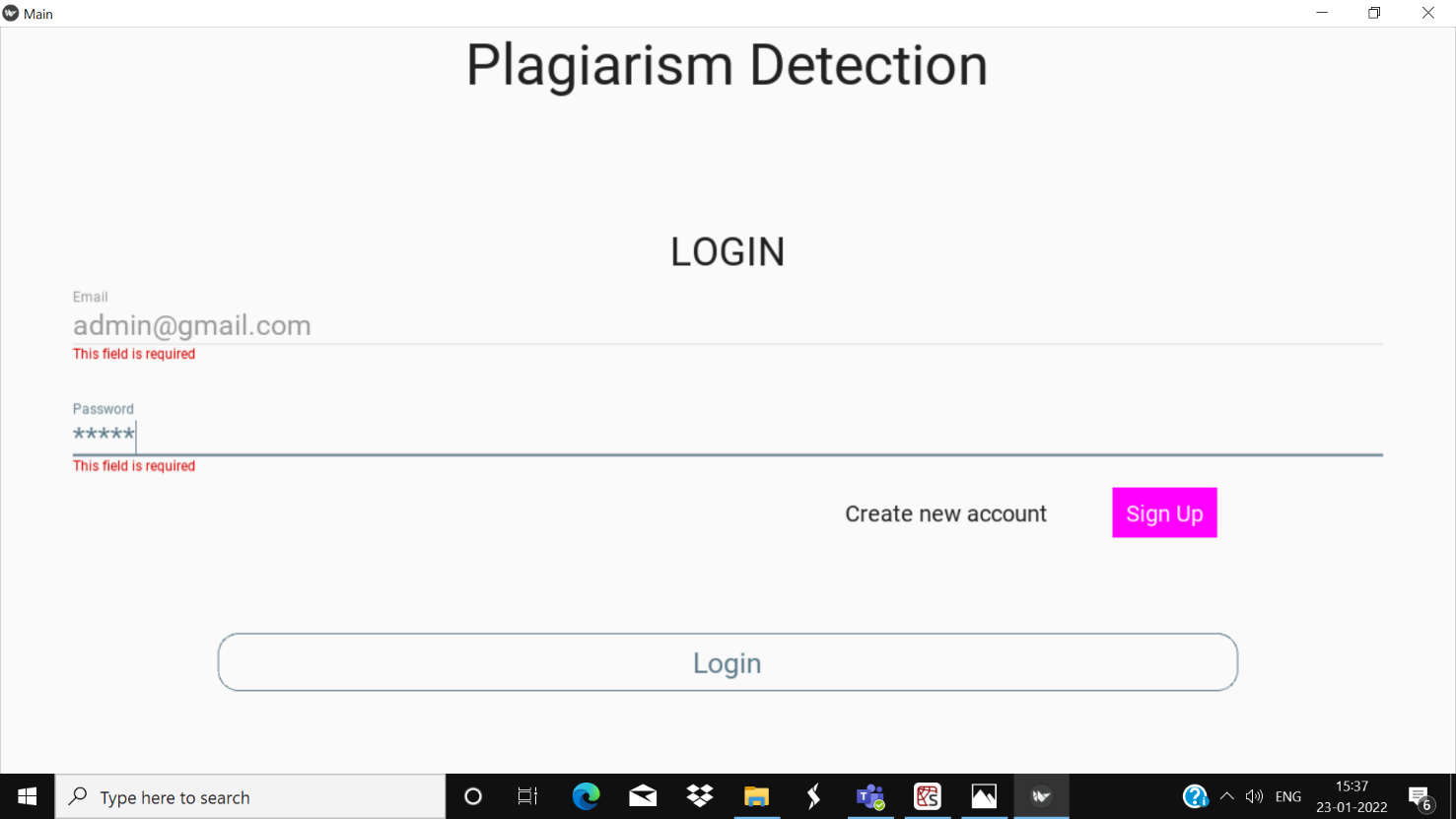
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC02 | | **Use Case ID:** UC02 | |
| **Test Case Title:** Login | |  | |
| **Test Case Description:**  Verify whether user can access to portal. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.user logins to the portal. | The user gets access to the portal if he gives the login credentials correctly. | | The user successfully accesses the portal  (Pass). |

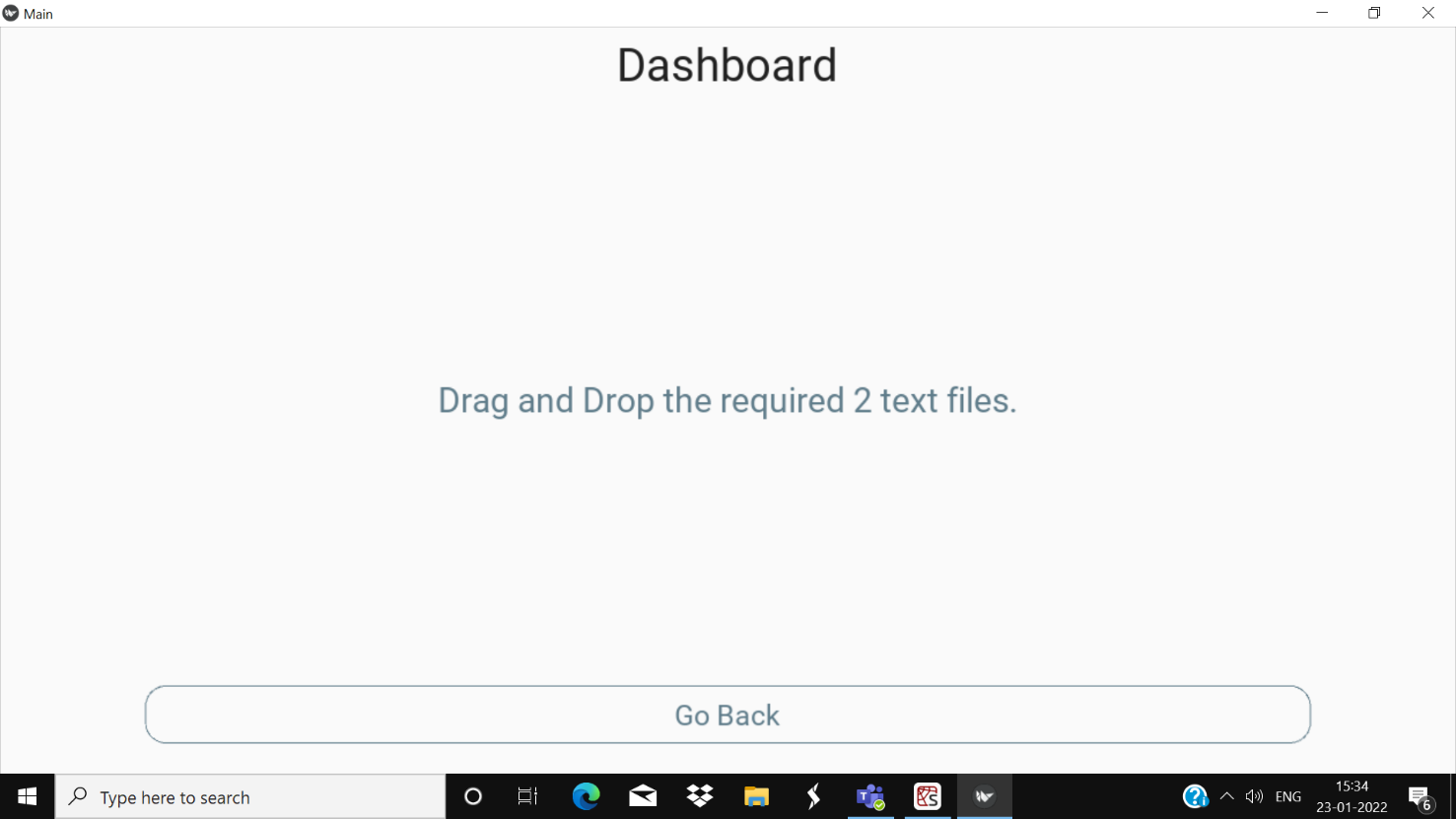
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC03 | | **Use Case ID:** UC03 | |
| **Test Case Title:** Process | |  | |
| **Test Case Description:**  Here user will be given allowance to drag and drop files. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.The user will be given allowance to successfully drag and drop files. | The user can successfully drag and drop files. | | The user successfully drag and drop files (pass). |

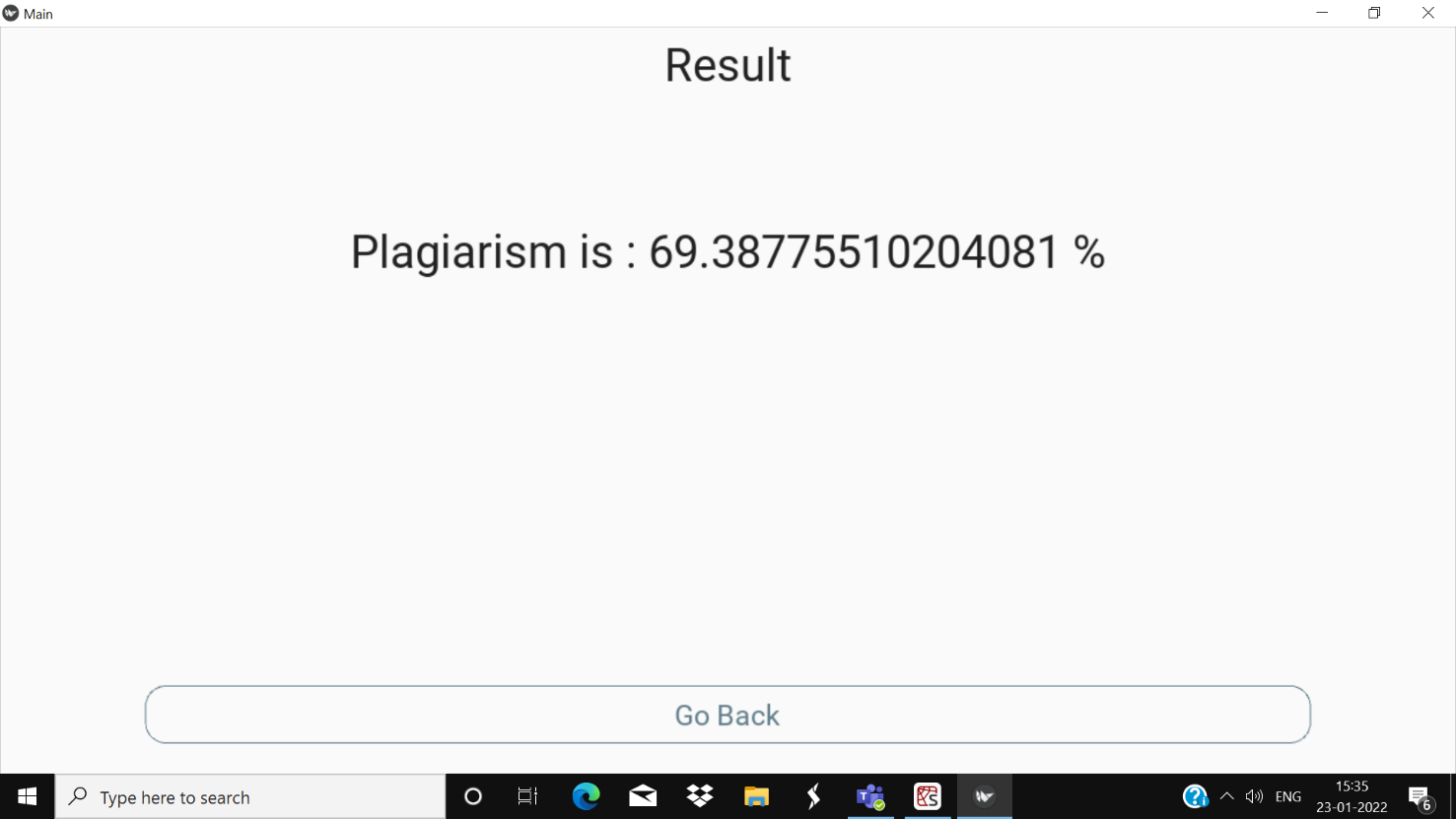
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC04 | | **Use Case ID:** UC04 | |
| **Test Case Title:** Result | |  | |
| **Test Case Description:**  Here the process happens and plagiarism percentage is displayed. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.The System undergoes process and plagiarism percentage is displayed. | The plagiarism percentage is displayed after successful drag and drop of files. | | The system successfully displays the plagiarism percentage.(pass). |

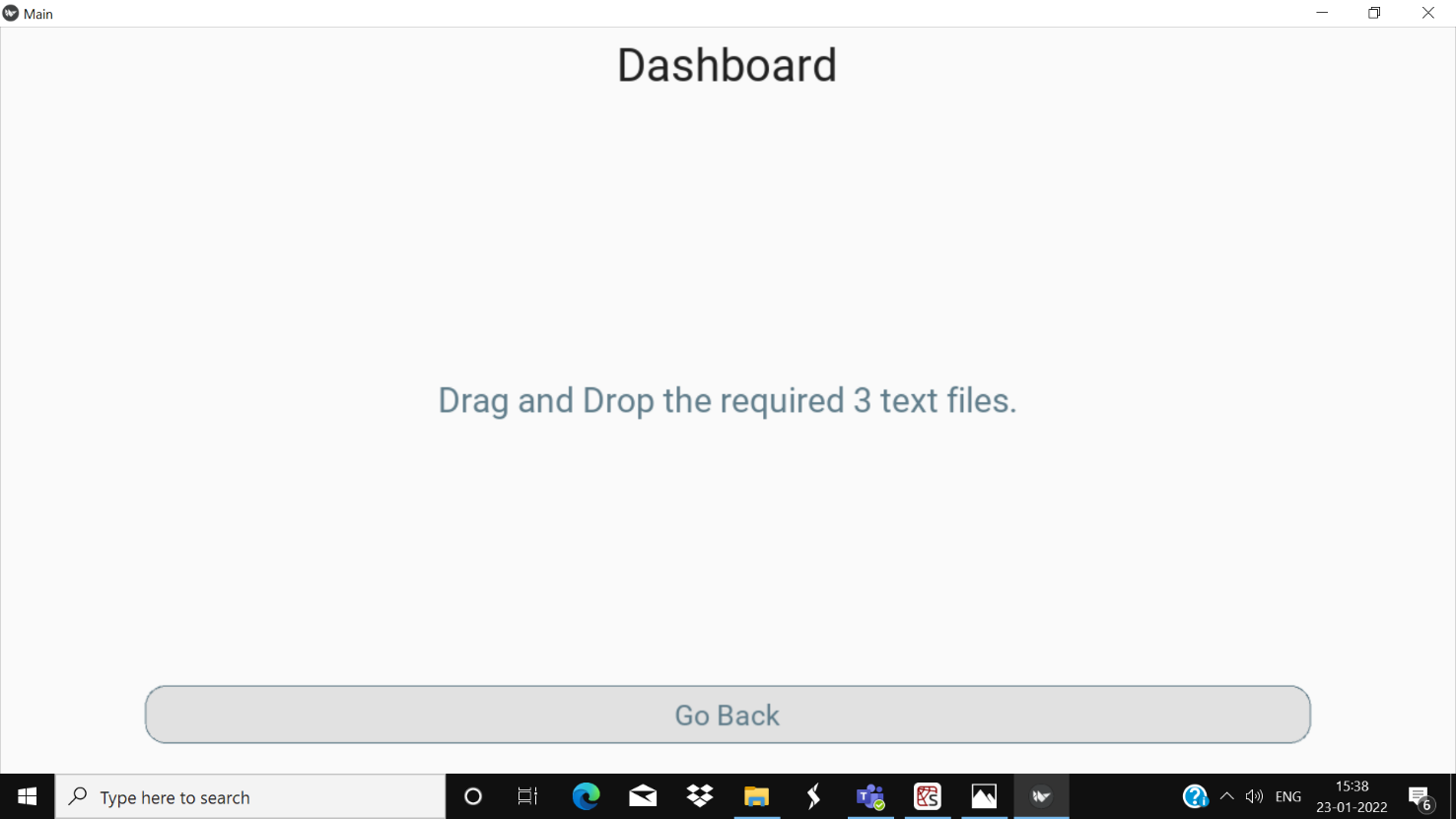
**RESULTS**

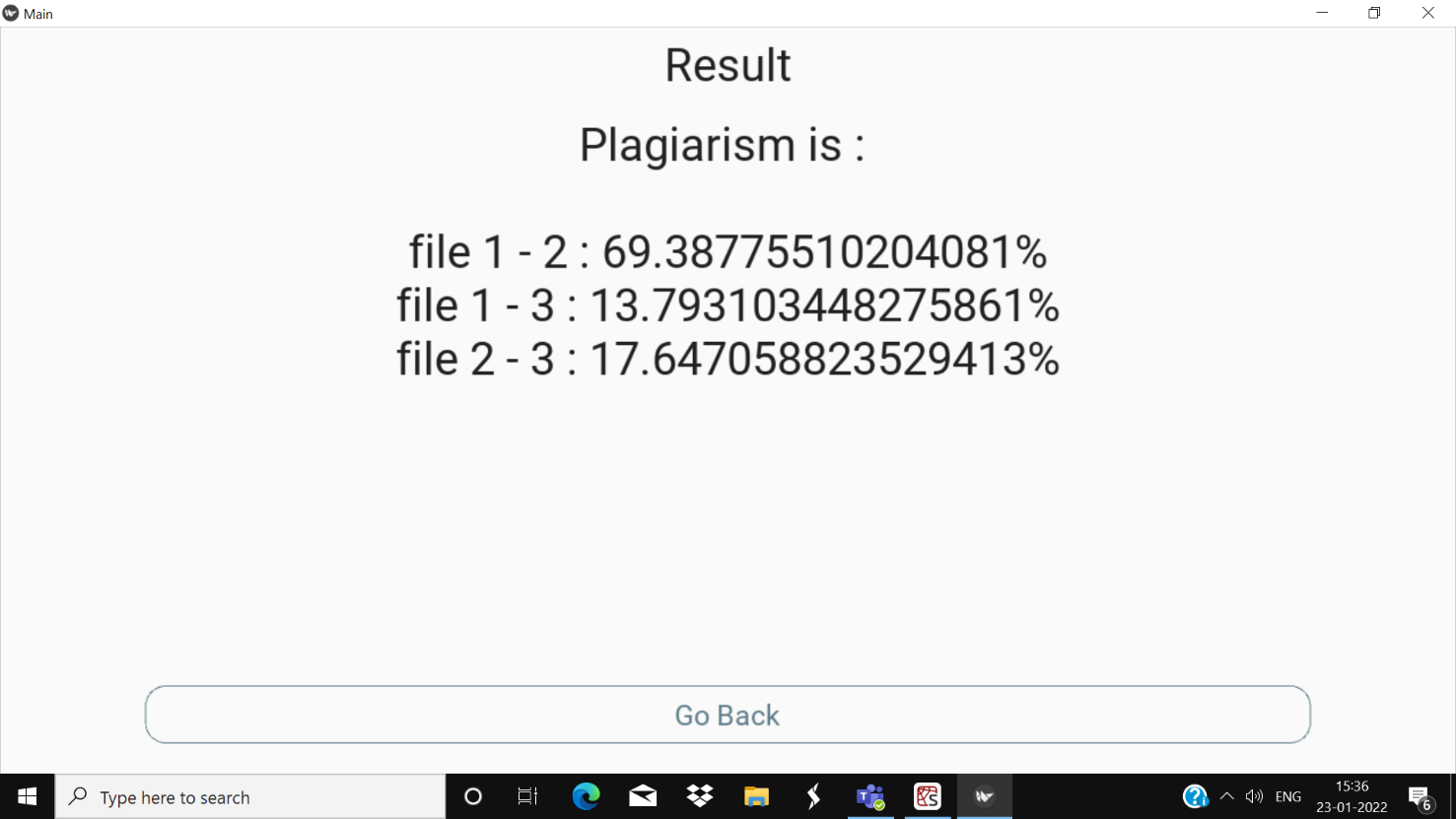
****

****

****

****

****

****

**Outcomes from the project apart from curriculum**

● I have got to know how to use different kinds of modules.

●I have discovered different kinds of libraries and modules and got to know their implementation and working.

●I have got to know about difflib from python.

●I have learnt how to build GUI using kivy and python.

●I have got to know how to handle events in GUI and about the components in them.

**CONCLUSION AND FUTURE WORK:**

**Future work**

● Including a database for the project.

● Improve GUI look and feel.

● Introduce options to select the files.

● Take user at which particular line plagiarism can be detected.

The project is successfully implemented with all the features and steps of Plagiarism detector as per the requirements.

**11. REFERENCES**

● Allen Downey,” Think Python: How to Think Like a Computer Scientist”,

O’Reilly publications,2nd Edition.

● nptel courses: <https://nptel.ac.in/noc/courses/noc22/SEM1/noc22>