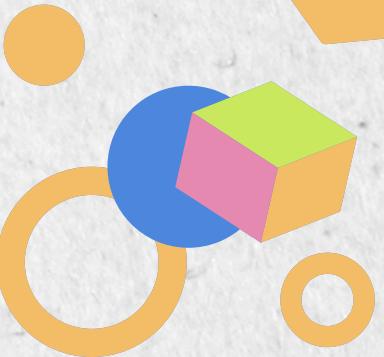
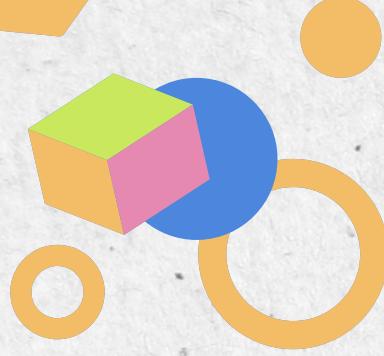


LEXI

Presented by
CS21B1030 - NOORAY MARIAM



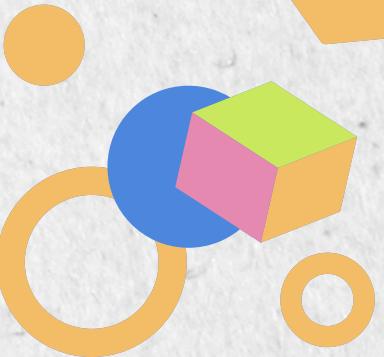
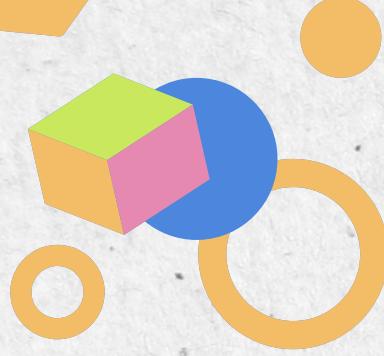
Introduction

Objective

The objective of the project is to create an application that secures data locally using encryption method and is user-friendly. It also helps in improving the writing by checking the grammar.

Motivation

With the advancement in technologies, cyber fraud such as phishing have come to a rise. It is hard to prevent a mistake. However, we can prevent the consequence. Hence, even when the security is breached and data is taken, the data is secured as it needs to be decrypted. This led to the idea of Lexi, a secured text app.



Introduction

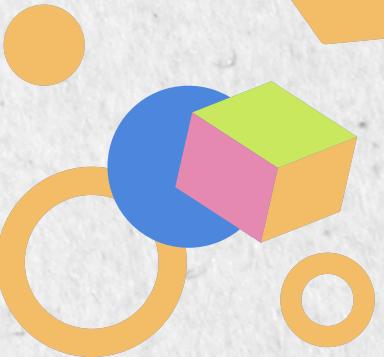
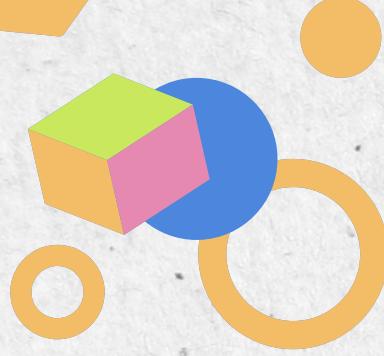
Scope

The scope of the project is the use lexi for offline basic grammar check and to improve the writing quality of all.

To prevent leaking of data through encoding methods

Problem Statement

In the growing years, with the technological advancement, it is difficult to protect our data. Hence, Lexi will aim to secure the file locally so that even if the security is breached, the data is not leaked.



Introduction

Product Features

Lexi is a secured file handler that secures the data using encryption. It has an inbuilt grammar checking tool that helps to improve the writing style. Hence, Lexi could be used for checking Grammar.

Requirements

User Requirements : Basic knowledge of using computers and handling buttons in an application

System Requirements : RAM - 4GB, Processor - Intel Pentium or above, Python preinstalled

Concepts Used

1. Modules

- a. base64 for Encoding and Decoding
- b. os for handling the directories
- c. language_tool_python for checking Grammar
- d. dbm for managing database
- e.time to create default filename and for saving in database
- f. webbrowser for opening a visit page
- g.tkinter for handling the GUI in python

2. Logic

- a. The data is saved by encoding
- b. The data is retrieved only by Lexi by decoding
- c. The grammar check is done on the whole text

3. GUI

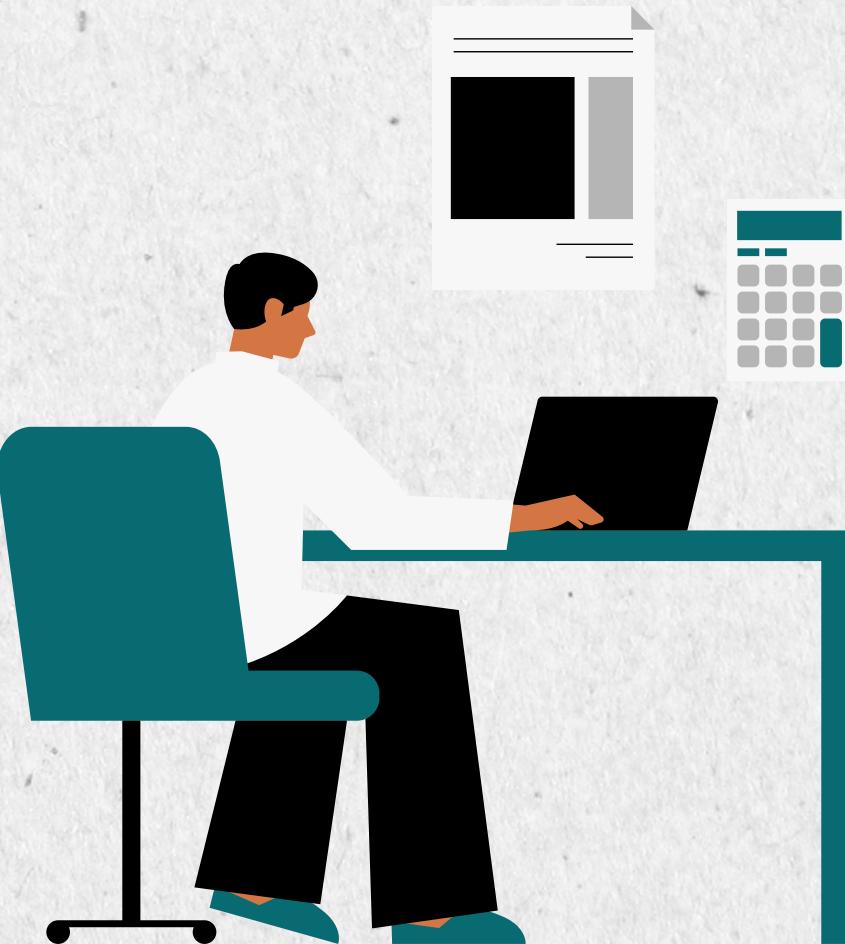
- a. Tkinter is used for GUI
- b.Text widget is used to get data and to display Grammar check
- c.Buttons are used for Open, Close, Save, etc.

4. OOPs

- a.The entire data is framed under a Window using classes
- b. all the widgets are inherited from their parent widgets
- c.The instance of class, root is an Object of tkinter

5. Database

- a. dbm, data base management, is used as it saves data in the form of dictionary
- b. The credentials are stored as key value and the data is stored as pair value. Hence easy to retrieve



lexi LEXI

- □ X

Open

Exit

Visit

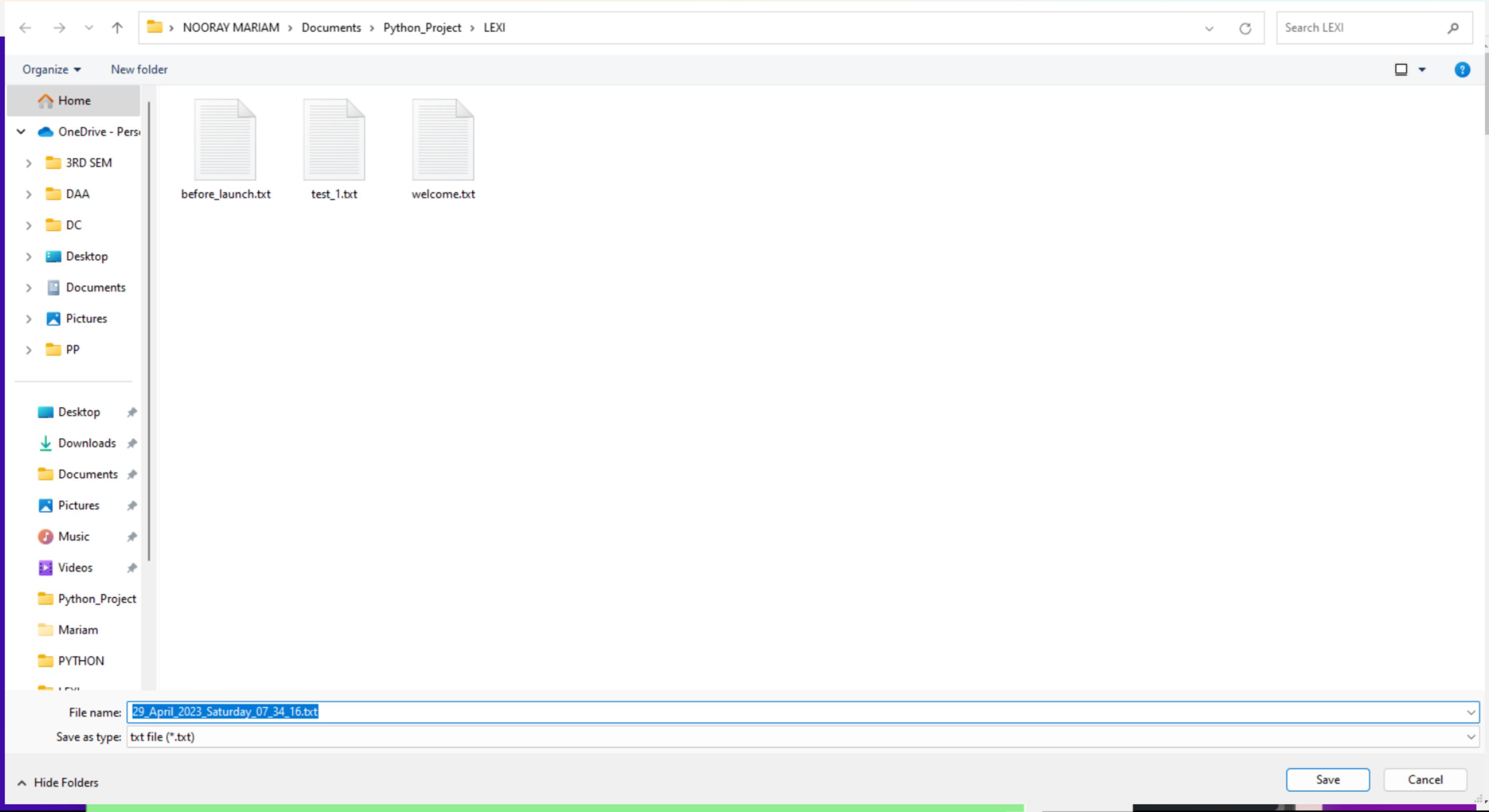
GrammarCheck

SAVE AS

Save

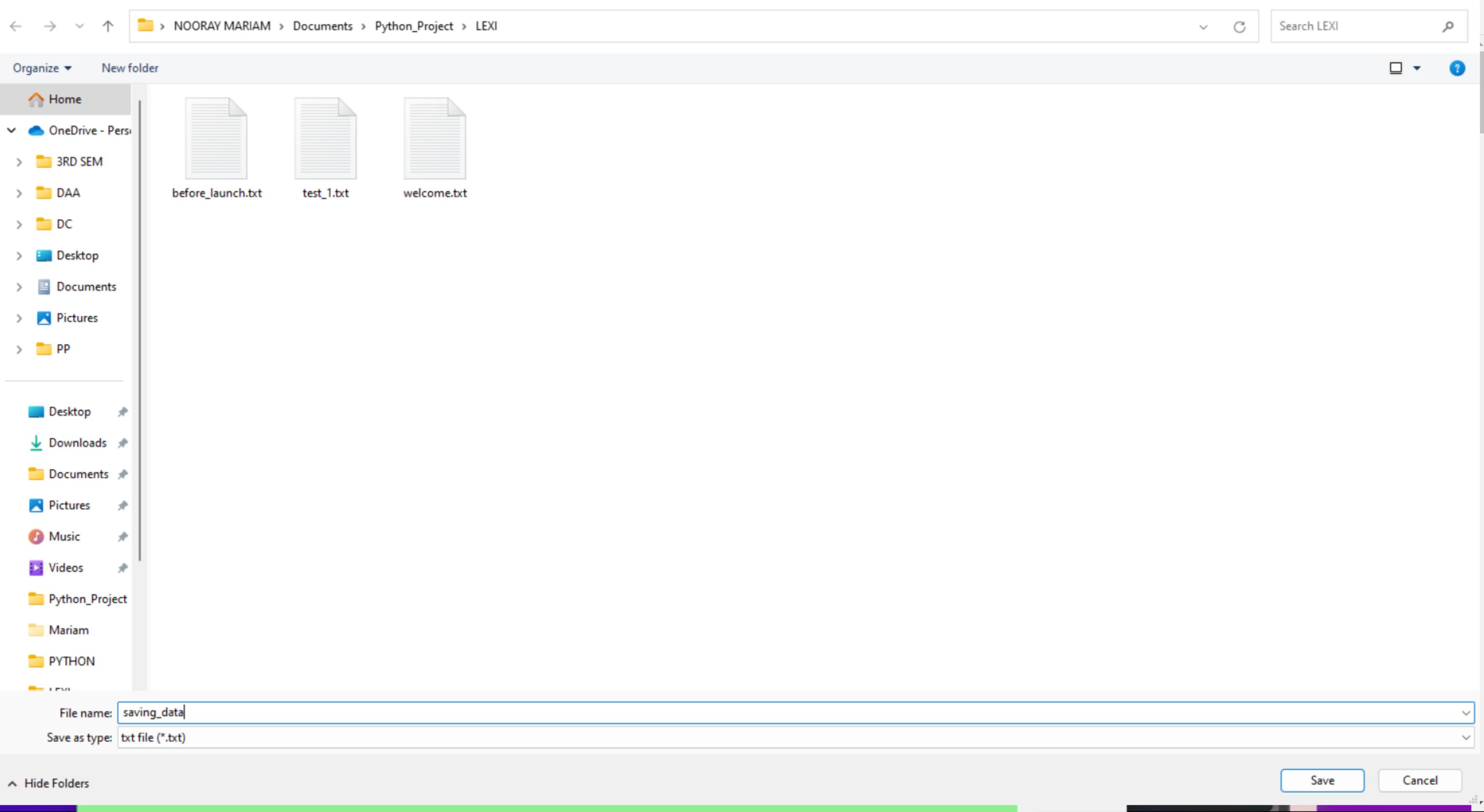
Save File as

X

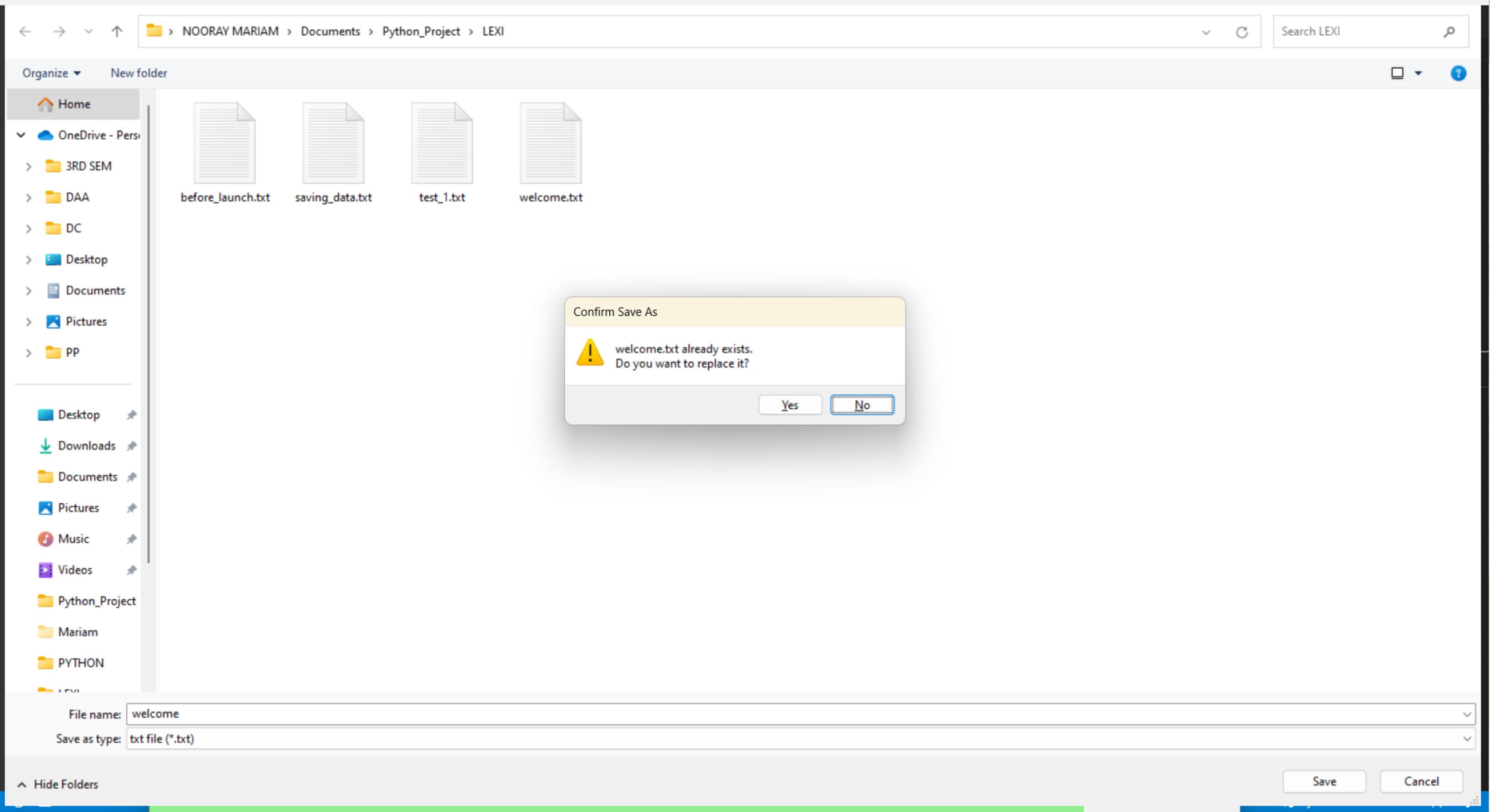


Save File as

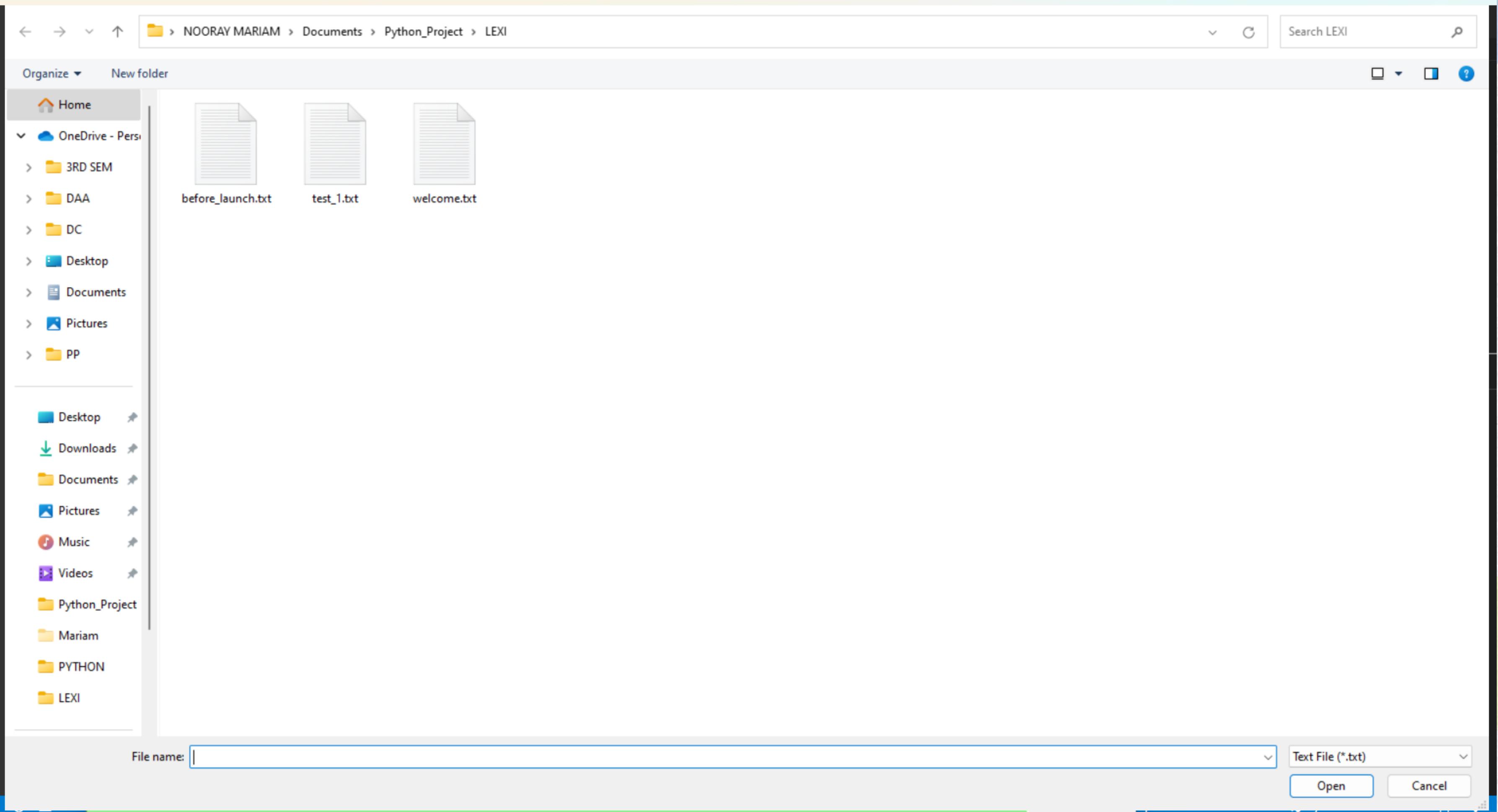
X



Save File as



Select a text file or all file.



Browser Products Privacy Resources Advertise

brave

Download Brave Try Brave Search

The best privacy online

Browse privately. Search privately. And ditch Big Tech.

Download Brave

33,135
Trackers & ads blocked

34.5 MB
Bandwidth saved

27 minutes
Time saved

Brave Basicatt Yubico

11:11
Search or enter address
Home New Tab

Privacy Stats
33.1k Trackers & ads blocked
34.5 MB Bandwidth saved
27m Time saved

LEXI

checking error one

- Open
- Exit
- Visit
- GrammarCheck
- SAVE AS
- Save

[Offset 0, length 8, Rule ID: UPPERCASE_SENTENCE_START
Message: This sentence does not start with an uppercase letter.
Suggestion: Checking
checking error one
^ ^ ^ ^ ^ ^ ^ }

Test Cases

Saving File:

1. Save + Filename + save = Save successfull
2. Save (after first save) = Save successfull
3. Save + existingFilename + save = save
invalid
4. SaveAs + Filename + save = Save successfull
5. SaveAs + existingFilename + save = save
invalid

OpenFile:

1. Open + Filename(lexi) = successfull
2. Open + Filename(other than lexi) =
unsuccessful

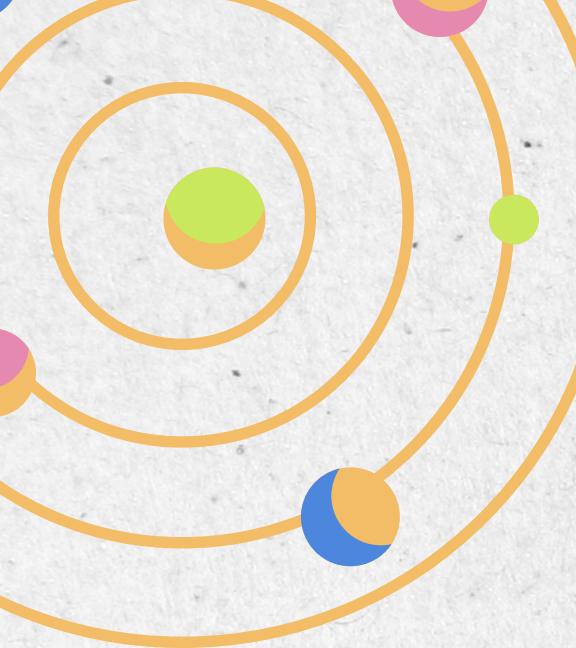
Visit Page:

1. visit + active internet = visit successfull
2. visit + no internet = visit unsuccessfull

Grammar Check:

1. text_with_error + Grammar = Grammar
check successfull
2. text_with_no_error + Grammar = Grammar
check successfull and no error reported





Conclusion

Conclusion

in conclusion, a crisp and confident writing cuts a lot of deal than an average writing. Hence a little help from Lexi might change your probabitly of success. Also it is better to integrate securing data on the spot as a precaution rather than loosing data and handling loss. Lexi is just the beginning.

Future Scope

- In future, the project could explore by integrating various features such as Login/Register page, Backup to Cloud Space and multi-user paradigm
- In future, more secured encryption logic could be invented



THANK YOU