**ENCRYPZ**

**A PROJECT REPORT**

***Submitted by:***

**Vibhor Bhargava (**19BCE10010**)**

**R. Kuringi Selvan (**19BCE10433**)**

**Aman Singh (**19BCE10160**)**

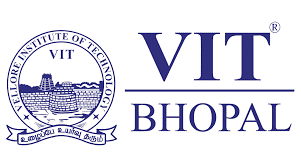
*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

# COMPUTER SCIENCE AND ENGINEERING



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRI KALAN, SEHORE**

**MADHYA PRADESH - 466114**

NOV 2020

**VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE**

**MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **ENCRYPZ** is the bonafide work of **VIBHOR BHARGAVA (19BCE10010), AMAN SINGH (19BCE10160), & R.KURINGI SELVAN (19BCE10433)** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project/research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

Dr. Ashwin M. Dr. Nitish Andola

School of Computer Science and Engineering School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on 30th April, 2021

**ACKNOWLEDGEMENT**

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr Manas Kumar Mishra, Head of the Department, School of Computer Science for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide, Mr. Nitish Andola, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| Figure 1 | Detailed Unified Modelling Language diag. | 10 |
| Figure 2 | Coding & Implementation | 12 |
| Figure 3 | Coding & Implementation | 13 |
| Figure 4 | Coding & Implementation | 14 |
| Figure 5 | Boot-up page / Main window | 16 |
| Figure 6 | Giving the required inputs | 16 |
| Figure 7 | On clicking RESULT, the converted text is displayed to the user | 17 |
| Figure 8 | Test for Decode mode | 17 |
| Figure 9 | Clicking RESET button erases values from all entry fields | 18 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE**  **NO.** | **TITLE** | **PAGE NO.** |
| Table 1 | Hardware and Software Requirements | 9 |
| Table 2 | Class Diagram | 15 |

**ABSTRACT**

EncrypZ is an encoding/decoding application that converts any given text into an encrypted form through a private key of the user’s choice, also given by the user itself. This application, built using Python, can be extremely useful for encrypted communication between 2 parties, conceal one’s private data, and much more.

Moreover, here the user remembers the key, just like a password, as the key is also given as an input. In short, EncrypZ is a simple, yet extremely efficient application for the people who do not know much about encryption.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **1.** | **INTRODUCTION**   * Objectives * Brief of existing Work | 7 |
| **2.** | **EXISTING WORK**   * Limitations of the existing work | 8 |
| **3.** | **PROPOSED WORK AND METHODOLOGY**   * A brief of the proposed approach | 8 |
| **4.** | **NOVELTY OF THE PROJECT**   * Advantages of the proposed work | 8 |
| **5.** | **SYSTEM REQUIREMENTS**   * Hardware Requirements * Software Requirements | 9 |
| **6.** | **ARCHITECTURE** | 9 |
| **7.** | **UML DIAGRAM** | 10 |
| **8.** | **LITERATURE REVIEW** | 10 |
| **9.** | **MODULE DESCRIPTION** | 11 |
| **10.** | **WORKFLOW EXPLANATION** | 11 |
| **11.** | **IMPLEMENTATION AND CODING** | 12 |
| **12.** | **CLASS DIAGRAM** | 15 |
| **13.** | **SNAPSHOTS OF THE WORK** | 16 |
| **14.** | **TESTING** | 18 |
| **15.** | **CONCLUSION** | 19 |
| **16.** | **REFERENCES** | 19 |

**Introduction**

***a*.** ***Objectives :***

**·** The objective of this project is to encode and decode messages using a common key, giving the users the freedom to enter a custom private key, that only they would know.

***b*.** ***A Brief of Existing work :***

* The altered information is referred to as ciphertext, which can be thought of as basically “digital gibberish." The information is unintelligible and essentially impossible to use for anyone without the encryption key.
* An encryption key is an indicator or identifier used to turn ciphertext into your desired output. Keys are kind of like passwords, but they’re virtually impossible to decipher without expert computational resources and decryption experience.
* Authorized recipients, on the other hand, are in possession of the key. They can easily identify themselves and gain access to the sensitive data, messages or files.

***c. Limitations of the existing work :***

. They demand Internet access.

· User cannot enter a key of his/her choice, and is stuck with the default key used by the algorithm.​

· Not connected to any database.

**Proposed work**

***a. A brief of proposed Approach :***

· We tried to keep our software as minimalist as possible so that it can easily run on old computers with low configurations.

· The interface is kept as simple as possible so that the user does not face any difficulties.

· Our software is to the point, it tells only the encrypted/decrypted text obtained via original message and custom key.

***b*.** ***Advantages of the proposed system :***

· EncrpyZ software helps the user secure their precious information such as payment card details, Aadhar Card No., PAN, etc. in an encrypted form.

* Only the user and authorized persons have the private key.
* Easy to use application, with a simplistic GUI, and no complexities.

**Recommended System Specifications**

**Table 1:**

|  |  |
| --- | --- |
| **Hardware Requirements** | **Software Requirements** |
| CPU: Pentium processor or above | OS: Ubuntu, Windows 7 or above |
| RAM: 128 MB minimum | Python 3 |

**Architecture Diagram**

1. Presentation- The top tier involves the user interface. In our case, the frontline window ,which appears in front of the user screen is ready to interact in user-friendly interface.​
2. When the user enters the input message, the key, and the mode ​of action on the input message (e for encode/d for decode) and clicks the RESULT button, the respective function (encode() or decode()) is called behind the scenes, and the message is operated upon by the function, to convert into the encrypted/decrypted form.
3. The converted text is then displayed in the entry field beside the RESULT button.
4. Clicking RESET button resets the values of the input message, mode, key, and result to NULL string, that is, it clears all the entry fields.
5. Clicking EXIT button calls the root.destroy() function of Tkinter in Python, and terminates the application.

**Diagram

Description automatically generated**

**Figure 1: Detailed Unified Modelling Language diag.**

**Literature review**

* First, we click on the EncrypZ executable shortcut, and the application opens.
* There are 3 input fields (Message, Key, Mode e/d), and 1 output field for encrypted/decrypted text.
* We enter our input message in the 1st field, key in the 2nd one, and ‘e’ to encode or ‘d’ to decode in the 3rd field.
* Then, we click on RESULT button to generate the converted text, and the text is displayed in the 4th field.
* We click RESET button to clear all the fields, and EXIT button to terminate the application.

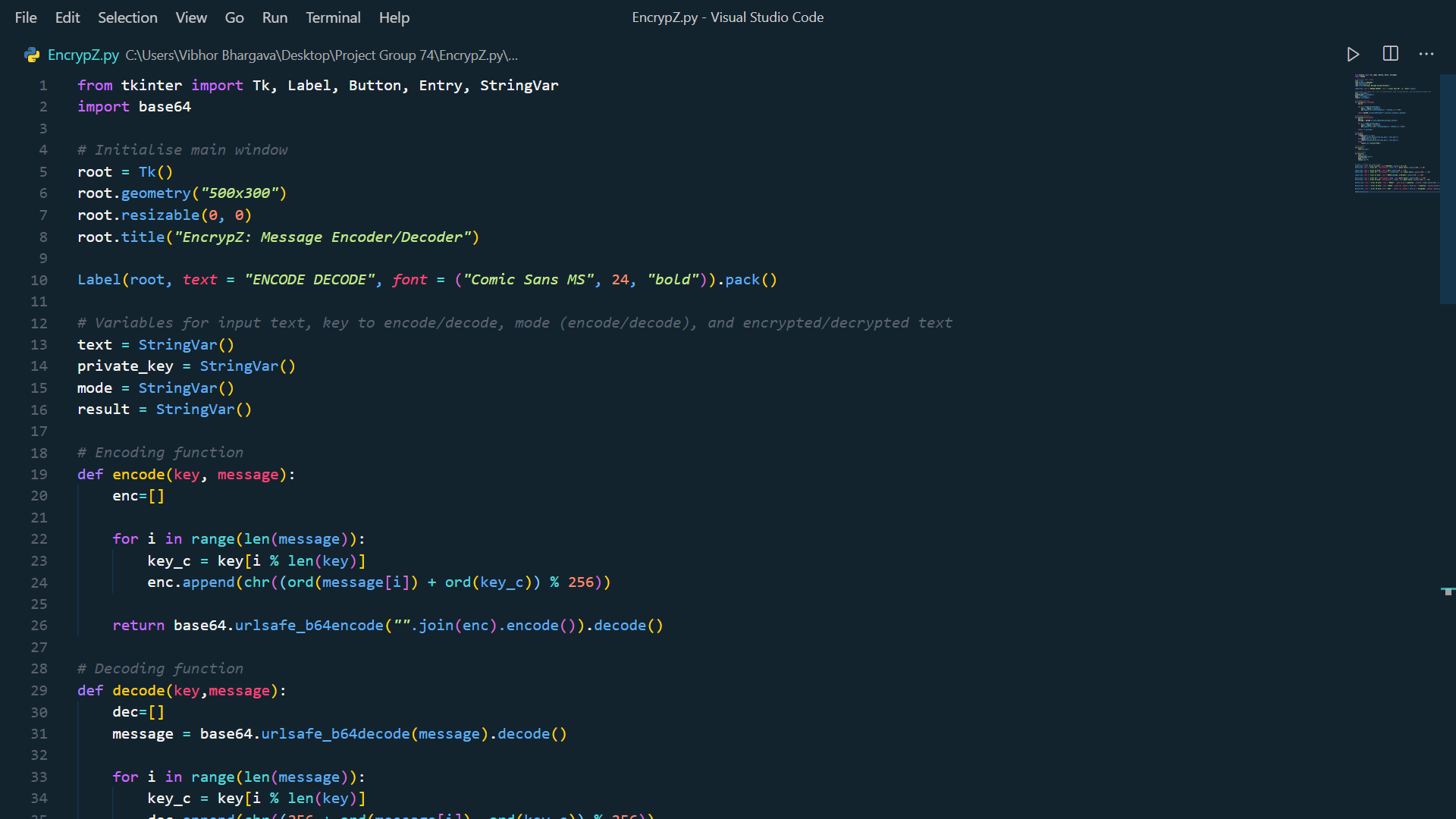
**Module description**

* Tkinter – It provides us with the classes of widgets that are used to build a GUI application in Python, like Label(), Button(), Entry(), etc.
* Base64 -- This module provides functions for encoding binary data to printable ASCII characters and decoding such encodings back to binary data. It provides encoding and decoding functions for the encodings specified in [**RFC 3548**](https://tools.ietf.org/html/rfc3548.html), which defines the Base16, Base32, and Base64 algorithms, and for the de-facto standard Ascii85 and Base85 encodings.

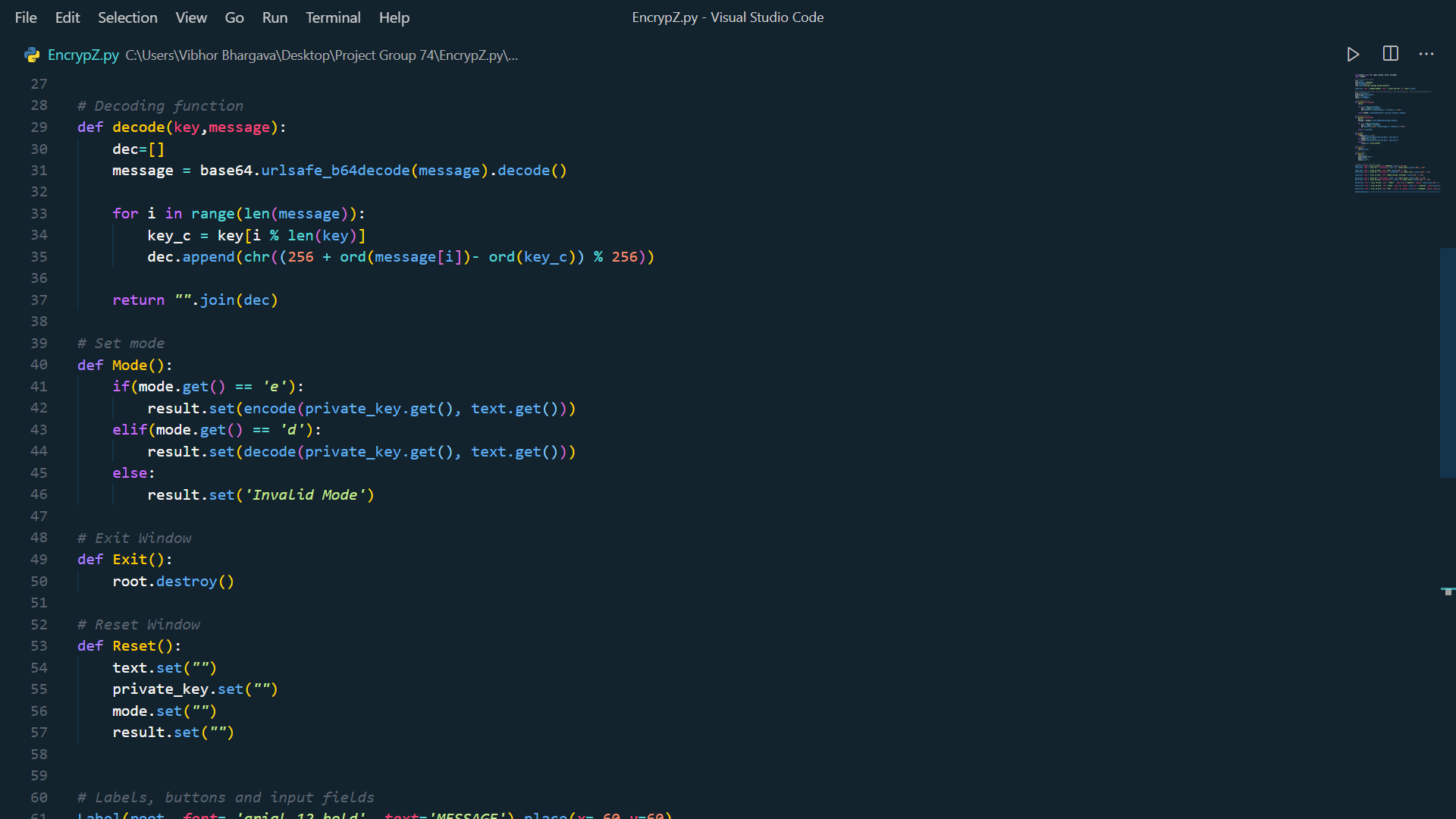
**Workflow explanation**

* First, the user clicks the EncrypZ executable shortcut to open the app, and the main window of the application opens up, with all the fields blank.
* User enters the input message, the custom key of their choice, and the mode of action in the respective entry fields.
* On clicking RESULT button, the converted text is displayed in the 4th entry field.
* RESET button erases values in all the fields, so that the user can convert some other text again.
* EXIT button terminates the application.

**Implementation and Coding**



**Figure 2**



**Figure 3**

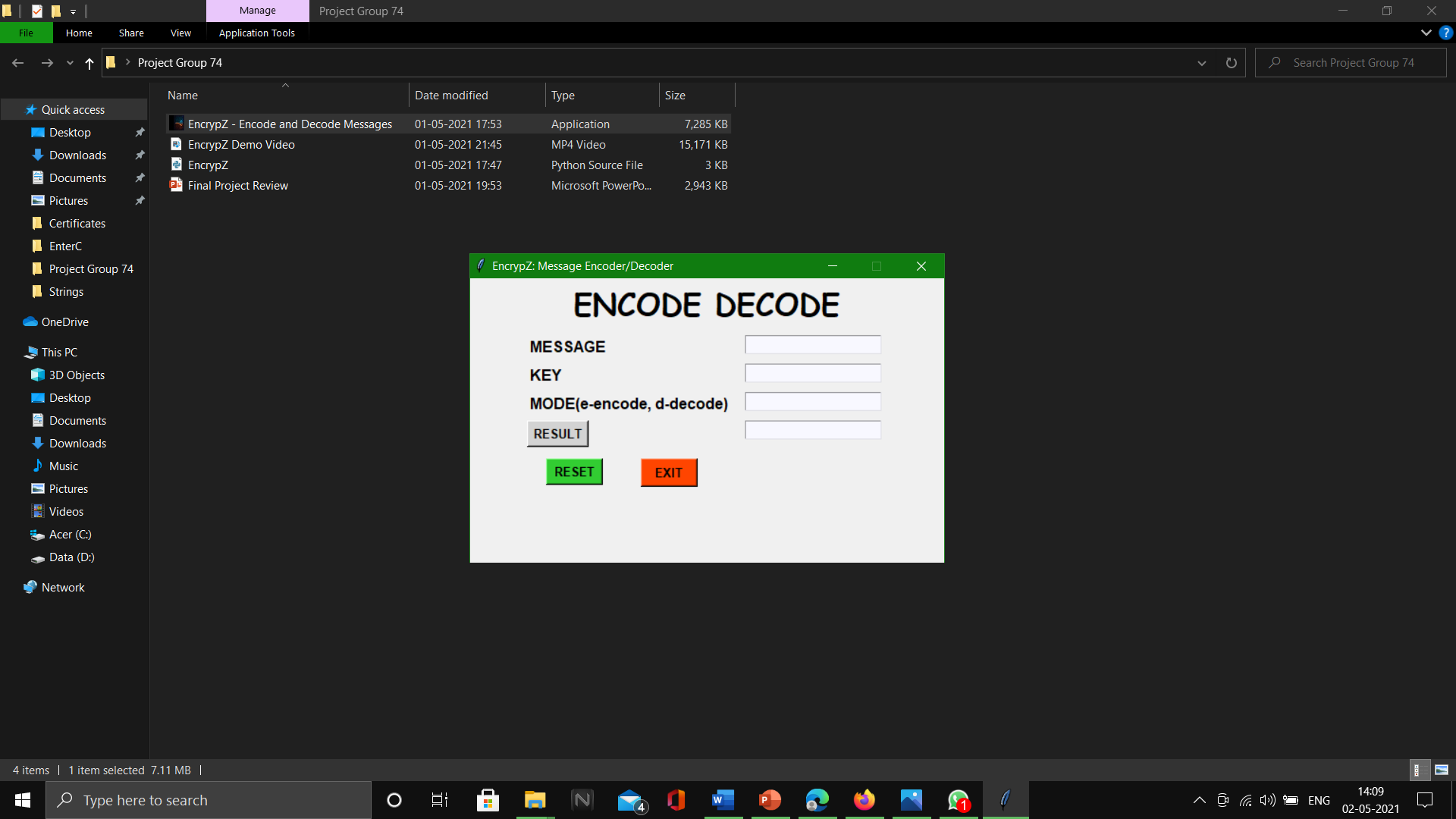


**Figure 4**

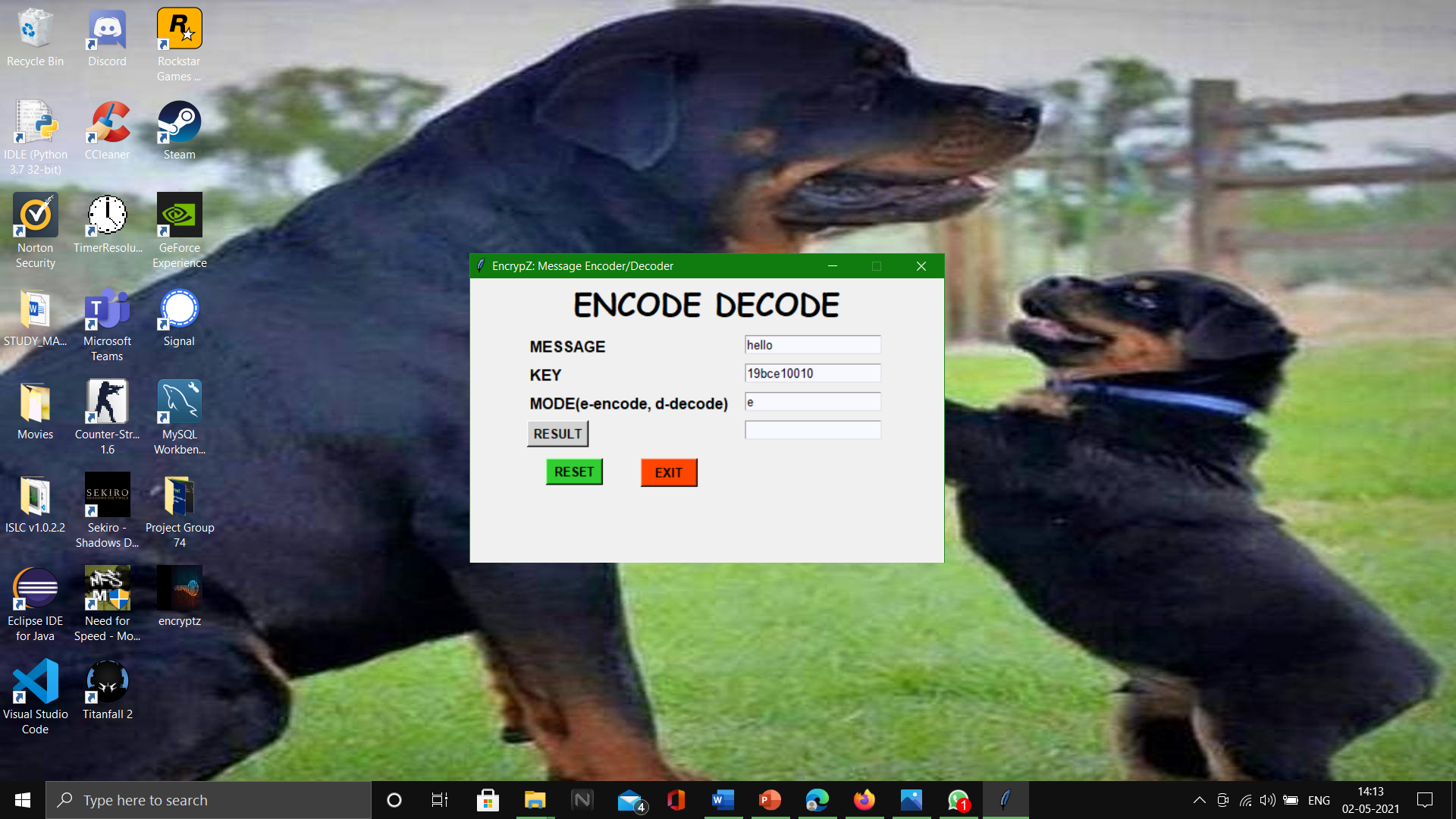
**Class Diagram (Table 2)**

|  |  |
| --- | --- |
| **Tk()** | The Tk class is instantiated without arguments. This creates a top-level widget of Tk which usually is the main window of an application. |
| **Label()** | This class implements a display box where you can place text or images. The text displayed by this widget can be updated at any time you want. |
| **Entry()** | The Entry class is a Tkinter widget used to enter or display a single line of text. |
| **Button()** | The Button class is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button. |
| **StringVar()** | StringVar class helps you manage the value of a widget such as a Label or Entry more effectively. |

**Screenshots of the Work**



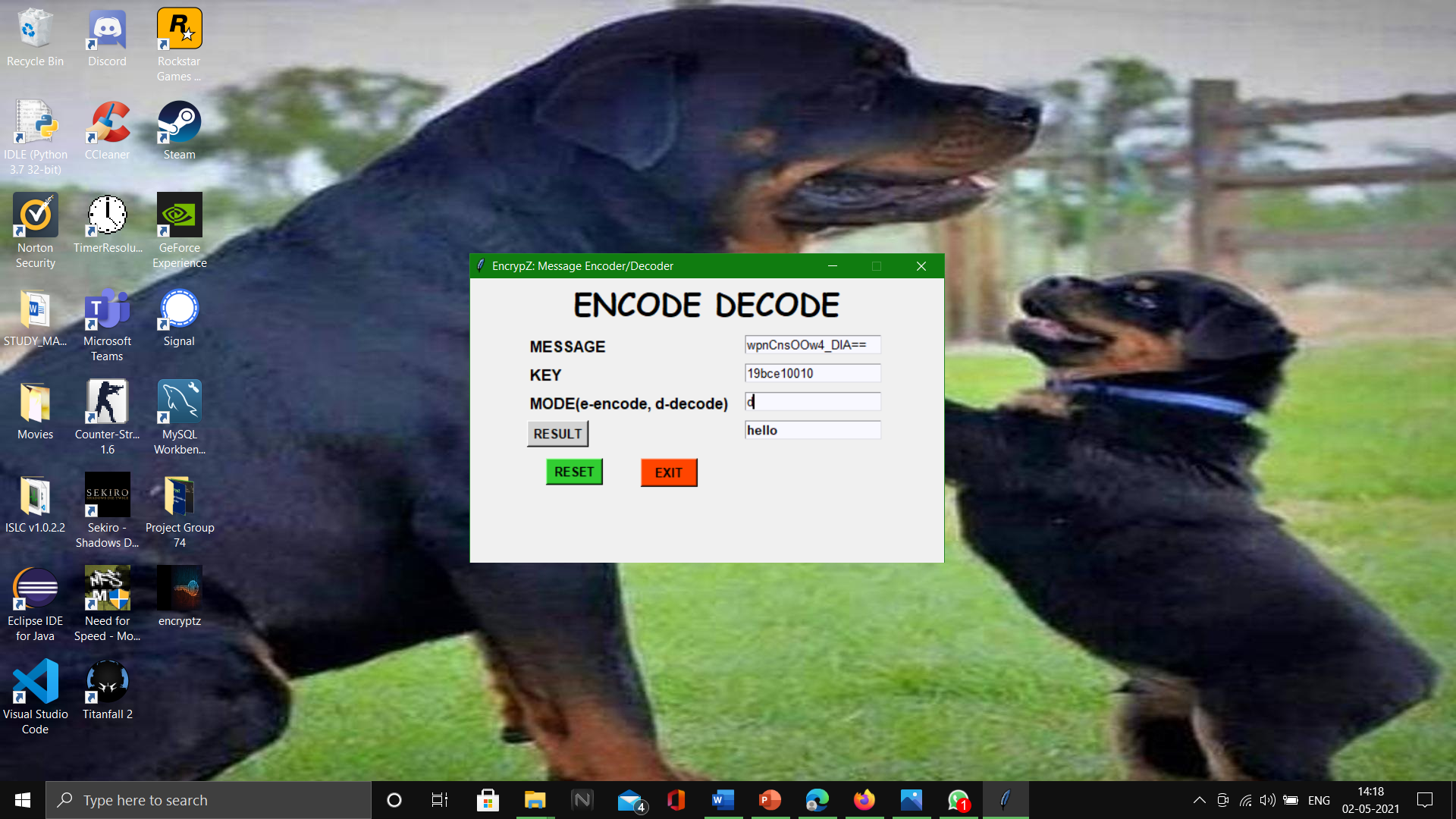
**Figure 5: Boot-up page / Main window**



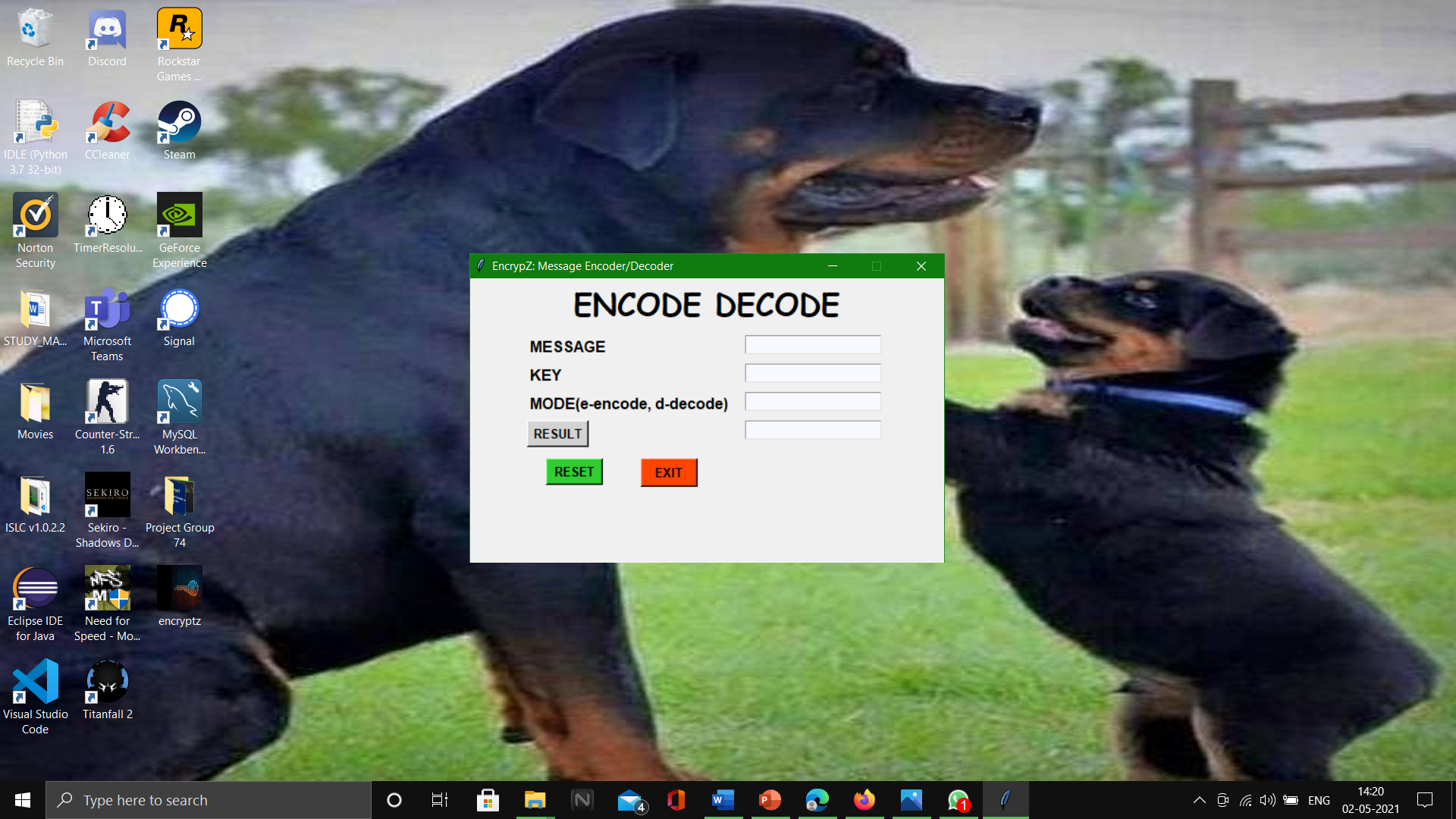
**Figure 6: Giving the required inputs**



**Figure 7: On clicking RESULT, the converted text is displayed to the user**



**Figure 8: Test for Decode mode**



**Figure 9: Clicking RESET button erases values from all entry fields**

**Testing**

* Initially, we conducted testing at phase level during development cycle or at module level in program code.​
* We tried to validate our software as per user experience and that it fully satisfies the need of the user.​
* We tried to verify that our software is as per our plan. Main targets of our test were errors, faults and failures.​
* During the testing of the encode() and decode(), it was kept in mind that whatever test case is run for Encode mode, using the encrypted text in the test for Decode mode, as input message with the same key should return the input message of the previous test case for Encode.
* For example, in 1 test, a user enters values of input message ‘hello’, key ‘19bce10010’, and mode ‘e’ for encrypting the text, and the converted text turns out to be ‘wpnCnsOOw4\_DlA==’. Then, in another test with inputs ‘wpnCnsOOw4\_DlA==’, key ‘19bce10010’, and mode ‘d’ for decoding the given text should return ‘hello’ as the decoded text.
* This way, we know that our application works flawlessly, without any loopholes.

**Conclusion**

* Encryption may seem unnecessary to many people, but in many cases, it is extremely essential in the contemporary world.
* For example – Concealing one’s credit/debit card details, Aadhar Card No., etc; an organisation wanting to store some confidential data.
* As we have seen in this application, it is impossible to relate or decode an encrypted message into its original meaning, without the key and the algorithm.
* Therefore, using EncrypZ, people can easily secure their information without using any complicated software, that they would not understand.

**REFERENCES**

1. For Tkinter module - [Python GUI's With TKinter - YouTube](https://www.youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV)
2. For base64 module - [base64 — Base16, Base32, Base64, Base85 Data Encodings — Python 3.9.4 documentation](https://docs.python.org/3/library/base64.html)
3. <https://www.geeksforgeeks.org/python-programming-language/>
4. <https://docs.python.org/3/>