1. Introduction

In the modern digital era, email is a widely used communication medium in personal and professional contexts. The accuracy of email addresses is critical for ensuring the delivery of important messages, avoiding bounced emails, and maintaining streamlined communication. This project, *"Email Validation with Voice Assistance,"* is an interactive Python-based application designed to validate email addresses through a graphical user interface (GUI) while providing auditory feedback to enhance user experience and accessibility.

The project leverages Python's core libraries for GUI design and text-to-speech synthesis, making it an ideal learning project for beginners and a useful utility for daily tasks.

2. Acknowledgement

I would like to express my heartfelt gratitude to my mentors, teachers, and peers who provided invaluable guidance and support during the development of this project. Their encouragement inspired me to explore innovative solutions and integrate advanced features like voice assistance. I would also like to acknowledge the Python open-source community for providing libraries like tkinter and pyttsx3, which enabled the successful implementation of this project.

3. Abstract

This project aims to create a Python-based GUI application that validates email addresses and provides immediate voice feedback. The application combines regular expressions for robust validation, the tkinter library for user-friendly interface design, and the pyttsx3 library for auditory feedback. With its simple design and interactive features, the application is accessible to all users, including those with visual impairments, making it a versatile and inclusive tool.

4. Declaration

I, Rohit Kumar, hereby declare that this project, *"Email Validation with Voice Assistance,"* is my original work. The project has been implemented using Python programming language and its libraries, adhering to ethical practices. It has not been submitted to any institution for academic or professional purposes prior to this.

5. Project Description Overview

The "Email Validation with Voice Assistance" project is designed to validate email addresses in real-time and provide users with both visual and auditory feedback. The application improves accuracy by instantly identifying valid and invalid email formats and makes the validation process more intuitive and accessible.

Key Features

1. **Real-time Email Validation**:
   Users can input an email address and immediately verify its format using a predefined pattern.
2. **Voice Feedback**:
   The application speaks out the validation result, ensuring accessibility for visually impaired users.
3. **User-Friendly Interface**:
   A clean and simple GUI design makes the tool easy to use for all age groups.
4. **Error-Free Results**:
   By using regular expressions, the application ensures precise validation of email addresses.

Tools and Technologies Used

1. **Programming Language**: Python
2. **Libraries and Modules**:
    - tkinter: For designing the graphical user interface. ○ pyttsx3: For integrating text-to-speech functionality.
    - re: For implementing the email validation logic.

6. Project Objectives

Primary Objective

To create a Python-based application that validates email addresses accurately and provides immediate feedback through both visual and auditory channels.

Secondary Objectives

- Enhance user accessibility through voice assistance.
- Demonstrate Python's versatility in GUI development and text-to-speech integration.
- Ensure the application is lightweight, fast, and easy to use.

7. Implementation Details

Workflow

1. **Input**:
   The user enters an email address in the provided text field in the GUI.
2. **Validation**:
   A regular expression checks the input against standard email formats.
3. **Feedback**:
    - **Visual Feedback**:

Displays "Valid Email Address" in green or "Invalid Email Address" in red.

- **Auditory Feedback**:
  Uses the pyttsx3 library to speak the validation result.

Code Explanation

1. **Regular Expression for Validation**:
   A regular expression (re) ensures the email follows the standard format:

```
email_regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
```

2. **GUI Creation**:
   The GUI is built using tkinter, providing an intuitive layout with:
   - An entry field for user input.
   - A button to trigger validation.
   - Labels to display the result dynamically.
3. **Voice Assistance**:
   The pyttsx3 library provides real-time voice feedback for both valid and invalid email entries:

```
engine = pyttsx3.init()
engine.say("Valid Email Address")
engine.runAndWait()
```

Full Code

```
import re
import tkinter as tk
from tkinter import ttk
import pyttsx3

# Initialize text-to-speech engine
engine = pyttsx3.init()

def speak(text):
    """Speak the provided text."""
    engine.say(text)
    engine.runAndWait()

def validate_email():
    """Validate the email address and provide voice feedback."""
    email = email_entry.get()
    email_regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
    if re.match(email_regex, email):
        result = "Valid Email Address"
        result_label.config(text=result, foreground="green")
        speak(result)
    else:
```

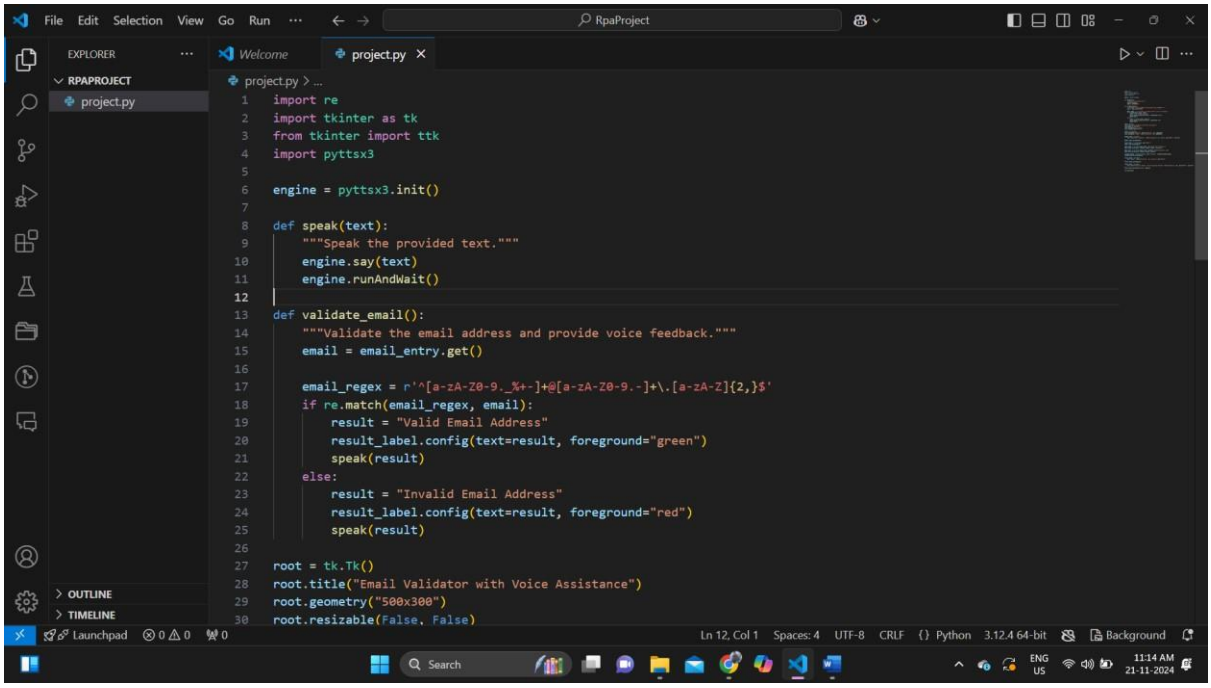
```
        result = "Invalid Email Address"
        result_label.config(text=result, foreground="red")
        speak(result)

# Create the GUI
root = tk.Tk()
root.title("Email Validator with Voice Assistance")
root.geometry("500x300")
root.resizable(False, False)
root.configure(bg="#f0f4f7")

# Styles
style = ttk.Style()
style.configure("TButton", font=("Helvetica", 12), padding=5)
style.configure("TLabel", font=("Helvetica", 12), padding=5)

# Header
header_label = tk.Label(
    root, text="Email Validator", font=("Helvetica", 18, "bold"), bg="#f0f4f7", fg="#333"
)
header_label.pack(pady=20)

# Input Frame
```

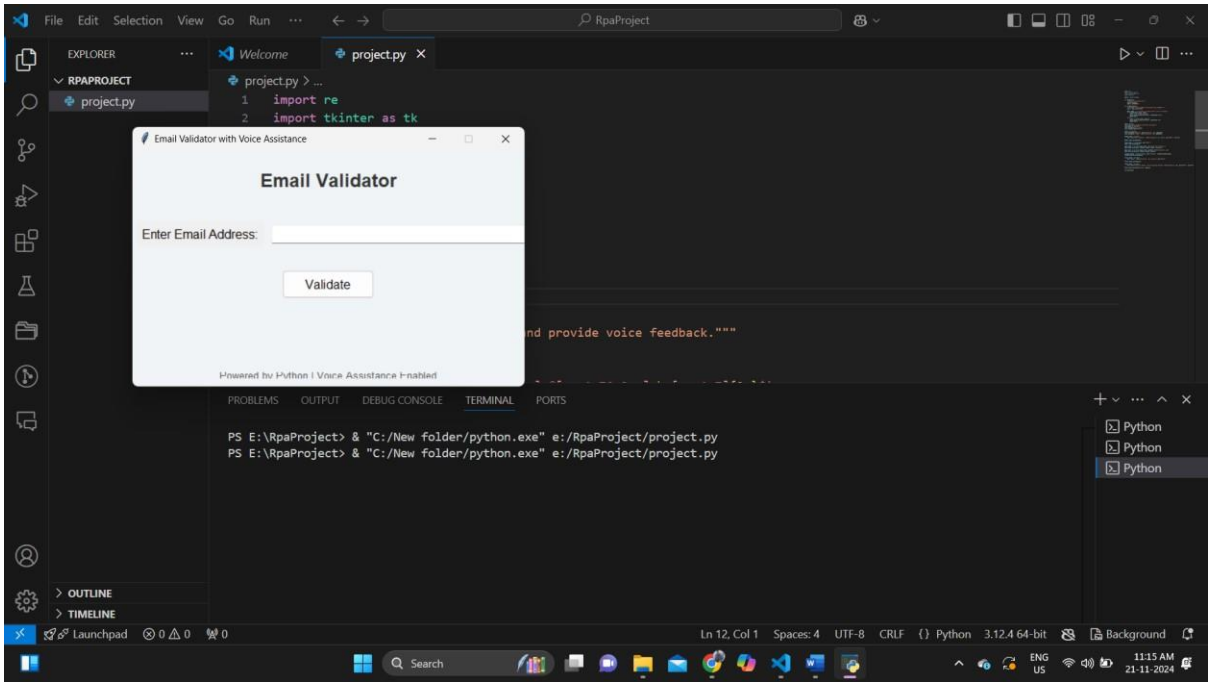
Copy code

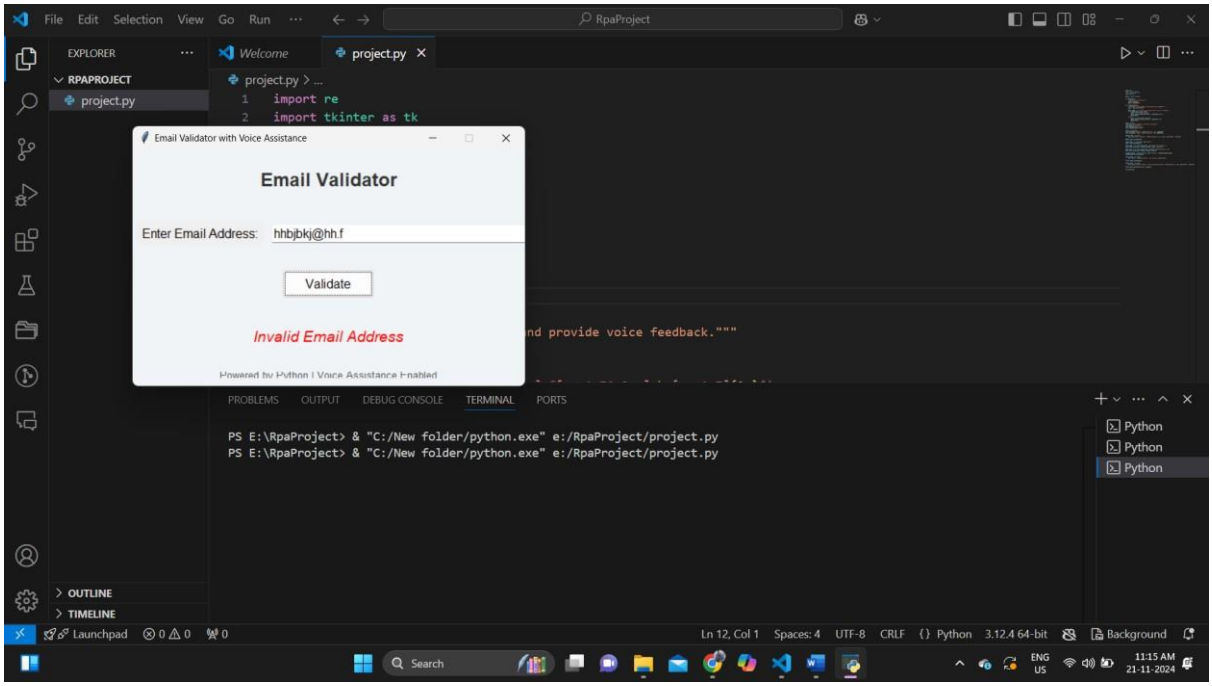
```
# Input Frame
input_frame = tk.Frame(root, bg="#f0f4f7")
input_frame.pack(pady=10)

# Input Widgets
email_label = ttk.Label(input_frame, text="Enter Email Address:")
email_label.grid(row=0, column=0, padx=5, pady=5, sticky="w")

email_entry = ttk.Entry(input_frame, width=40, font=("Helvetica", 12))
email_entry.grid(row=0, column=1, padx=5, pady=5)

# Validate Button
validate_button = ttk.Button(root, text="Validate", command=validate_email)
validate_button.pack(pady=15)

# Result Label
result_label = tk.Label(root, text="", font=("Helvetica", 14, "italic"), bg="#f0f4f7")
result_label.pack(pady=20)

# Footer
footer_label = tk.Label(
    root, text="Powered by Python | Voic ssistance Enabled", font=("Helvetica", 10), bg=
)
```

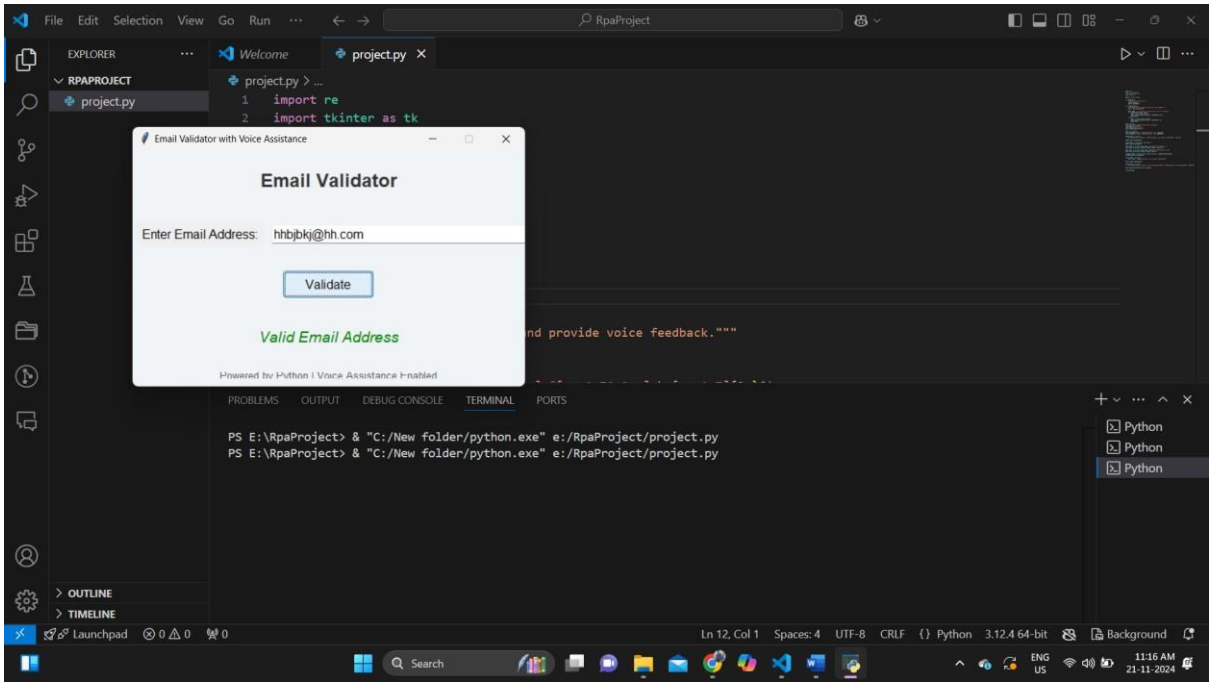
```
footer_label.pack(side="bottom", pady=10)

# Run the GUI
root.mainloop()
```

8. Screenshot









9. Conclusion

The *"Email Validation with Voice Assistance"* project showcases how Python can be used to develop user-friendly and accessible applications. The integration of GUI and voice assistance creates a seamless experience for users while emphasizing the importance of validating email addresses accurately.

This project lays the foundation for future enhancements, such as:

- Adding real-time domain verification.
- Supporting multiple languages for voice assistance.
- Extending the functionality to validate bulk email addresses.

With its simplicity and functionality, the project achieves its objectives and highlights Python's versatility in solving real-world problems.