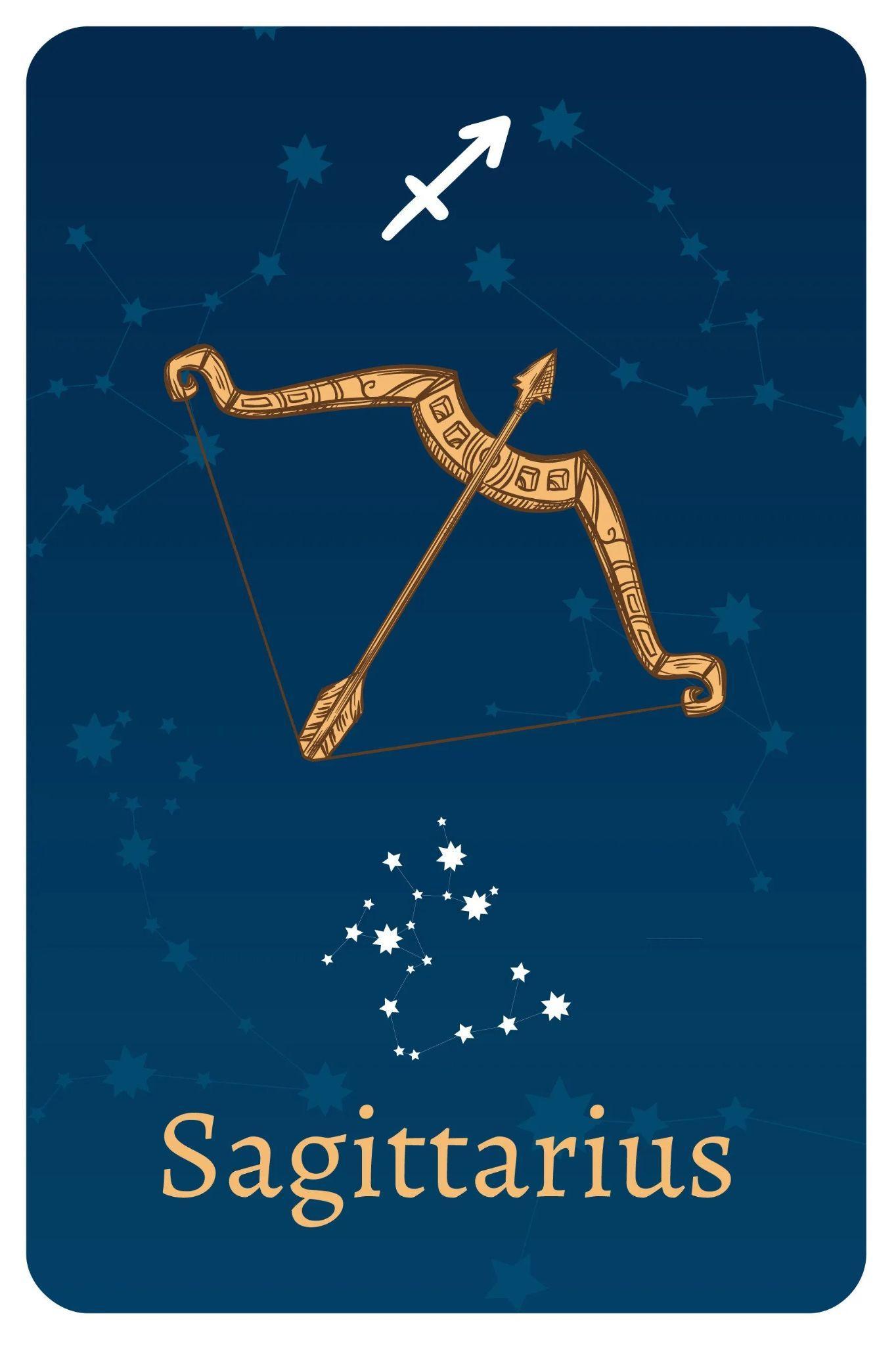
****

TEAM MEMBERS

Shivansh Shukla

Nathan Lee

Edwin Kulakkattolikel

Contents

[PURPOSES AND GOALS 3](#_heading=h.gjdgxs)

[HOW TO INSTALL THE PROGRAMS 4](#_heading=h.30j0zll)

[DEPENDENCIES 4](#_heading=h.1fob9te)

[HOW TO RUN THE PROGRAMS 4](#_heading=h.3znysh7)

[DESIGN ARCHITECTURE 5](#_heading=h.2et92p0)

[PROCESS & WORKFLOW 5](#_heading=h.tyjcwt)

[TEST DATA 5](#_heading=h.3dy6vkm)

[VIDEO RECORDINGS 6](#_heading=h.1t3h5sf)

[CONCLUSIONS 6](#_heading=h.4d34og8)

[REFERENCES 6](#_heading=h.2s8eyo1)

# PURPOSES AND GOALS

The purpose of the lab is for students to learn how to combine Python, GUI development, Excel for database administration, email, and text messaging in order to create a workable business solution. The objective is to develop an appointment scheduling system for practical applications, enhancing their understanding and use of these technologies.

The lab's goals are to create a user-friendly GUI, manage databases, automate communications like text reminders, and integrate tools such as Python, GUIs, Excel databases, email, and text messaging.

# HOW TO INSTALL THE PROGRAMS

## DEPENDENCIES

pip install twilio

pip install yagmail

create Twilio account at link <https://www.twilio.com/try-twilio>

create test email

# HOW TO RUN THE PROGRAMS

In order to run the COVID database program, the user needs “pip install Twilio” and “pip install yagmail” to send notifications to their number and email. They also need the program file that includes all of the Excel files, PNG pictures, and the Python code application. With this, the user is able to run the program.

# DESIGN ARCHITECTURE

In this lab, a system for registering appointments is created by connecting hardware and software components. A computer running Python, a graphical user interface (GUI) for user interaction, Excel as a database, Python scripts for email and text messaging, and a mobile phone for text message reception are important components.

Hardware Building Blocks: A laptop and a phone.

Software Logical Blocks: Text Method (Twilio), Email Method (Gmail and yagmail), GUI (Tkinter), Excel (database), and Python (core language).

User input is enabled by the GUI, data processing is handled by Python scripts, an Excel database is updated, and email and SMS reminders are sent. The mobile phone receives SMS reminders, and external services (such as Gmail and Twilio) manage email and text communication.

By automating appointment reminders, this integrated system demonstrates how several tools and technologies work together to deliver a useful business solution.

# PROCESS & WORKFLOW

TestTwilio.py - The provided code utilizes the Twilio API to automate the process of sending text messages. It begins by setting up the Twilio client with the necessary credentials and then constructs a message template that includes placeholders for the recipient's first and last name. In this specific case, it is designed to send a COVID-19 vaccination reminder to a recipient named "Shivansh Shukla" on a specified date. The recipient's first and last names are assigned to variables, and the message template is filled in with this information. The code then sends the customized message from a designated Twilio number to the recipient's phone number. Once the message is sent, the code prints a confirmation message containing the Service Identifier (SID) for tracking. This process streamlines the workflow for sending personalized text messages, making it efficient and convenient.

TestYagmail.py - This code uses the yagmail library to send an email from a Gmail account to another email address. It starts by setting up the sender's email and a secure password. The recipient's email, subject, and email content (including attachments) are defined. Then, it connects to Gmail's email system and sends the email. If successful, it prints a confirmation message. This code simplifies sending emails with attachments through Gmail.

COVID database - This code uses the Yagmail and Twilio libraries along with some other ones necessary for operating the database. The code uses multiple user-defined functions to create a GUI-style database to simulate a COVID-19 vaccination record application. It starts by making the user register for an account which is like an employee getting on board. Next, the user logs in using their credentials and the person taking the vaccination has to fill out a general registration form that records their name, date of birth, address, current date, gender, etc. in an Excel database. Then the application sends reminders via text message and email for the first vaccination shot completed and sends another reminder for the second vaccination shot 21 days later.

# TEST DATA

|  |  |  |
| --- | --- | --- |
| Test Name | Code Results | Succesful |
| TestTwilio.py |  | Yes |
| TestYagmail.py |  | Yes |
| Covid database |  | Yes |

# VIDEO RECORDINGS

|  |  |  |
| --- | --- | --- |
| **Recording Title** | **URL** | **Notes** |
| Lab 2 Demo | <https://youtu.be/CSDgJQstbf8> | The demo for Lab 2, includes everything |

# CONCLUSIONS

In this lab, our team learned the process of Yagmail and Twilio which can help send updates to users in a set database. We also learned how to incorporate Excel spreadsheets into the program file that is able to store inputted info when users are creating a new account, as well as receiving information in Excel spreadsheets for users to enter in their personal information to get COVID-19 vaccination updates.

The lessons we have learned in this lab include making sure the Python script is organized and readable so that it would be easier to find bugs and to add a few features to the program. We also learned how to better research the web on how to execute specific commands on Python to make our program work well and successfully.

# REFERENCES

* What Is Agile Reporting? (Definition and How To Complete) <https://www.indeed.com/career-advice/career-development/agile-reporting>
* Test Report <https://strongqa.com/qa-portal/testing-docs-templates/test-report>
* Image import <https://bobbyhadz.com/blog/tkinter-tclerror-couldnt-recognize-data-in-image-file>
* Yagmail [https://www.geeksforgeeks.org/send-email-using-yagmail-in-python/#](https://www.geeksforgeeks.org/send-email-using-yagmail-in-python/)