Training Solutions App User Manual

Created for

MidTown IT

PROJECT REFERENCE: ICTPRG434-435_AT2
AUTHOR: Luke Wait
DATE: 01.11.23

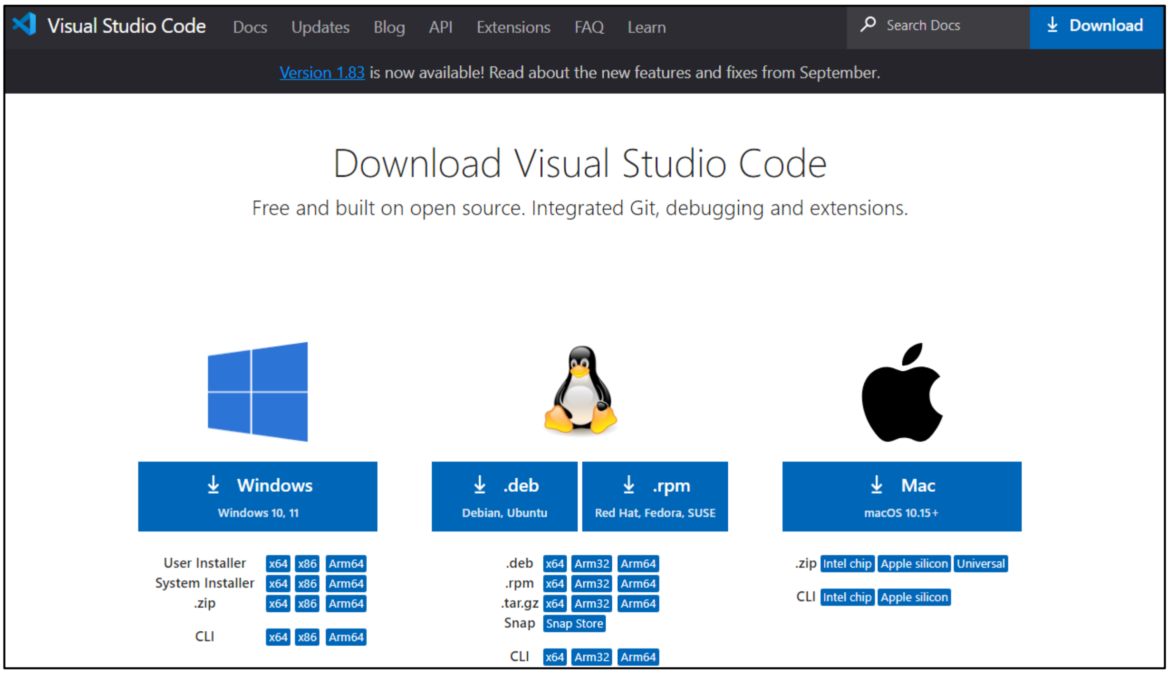
Contents

PART 1 - MidTown IT Training Solutions App

Installation and setup

Install Integrated Development Environment (IDE) and Python extensions

Open a web browser, head to https://code.visualstudio.com/download and download VS Code for the chosen operating system (Windows, Linux, or Mac).



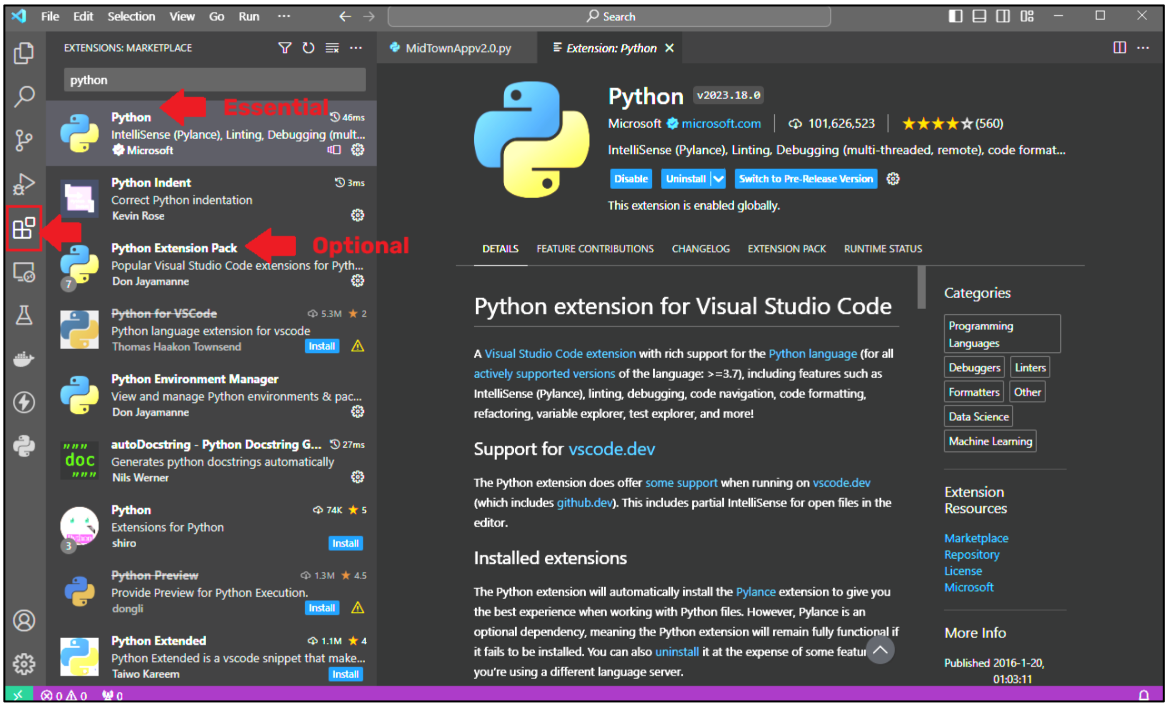
Version 1.83 is now available! Read about the new features and fixes from September.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Figure 1: Official VS Code website and download links.

Once downloaded, install and launch VS Code. Next, install required Python extensions by accessing the extensions tab on the left-hand side. Search and install the ‘Python’ extension by clicking the blue ‘Install’ button. For an improved experience you can also install the ‘Python Extensions Pack’ (optional).



Python — IntelliSense (Pylance), Linting, Debugging (mult... — Microsoft — 46ms

Python Indent — Correct Python indentation — Kevin Rose — 3ms

Python Extension Pack — Popular Visual Studio Code extensions for Pyth... — Don Jayamanne

Python for VSCode — Python language extension for vscode — Thomas Haakon Townsend — 5.3M ★ 2 — Install

Python Environment Manager — View and manage Python environments & pac... — Don Jayamanne

autoDocstring - Python Docstring G... — Generates python docstrings automatically — Nils Werner — 27ms

Python — Extensions for Python — shiro — 74K ★ 5 — Install

Python Preview — Provide Preview for Python Execution. — dongli — 1.3M ★ 4.5 — Install

Python Extended — Python Extended is a vscode snippet that make... — Taiwo Kareem — 1.1M ★ 4 — Install

Python v2023.18.0

Microsoft microsoft.com | 101,626,523 | ★★★★☆ (560)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), code format...

This extension is enabled globally.

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the Pylance extension to give you the best experience when working with Python files. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional if it fails to be installed. You can also uninstall it at the expense of some featu you're using a different language server.

Categories: Programming Languages, Debuggers, Linters, Formatters, Other, Data Science, Machine Learning

Extension Resources: Marketplace, Repository, License, Microsoft

More Info: Published 2016-1-20, 01:03:11

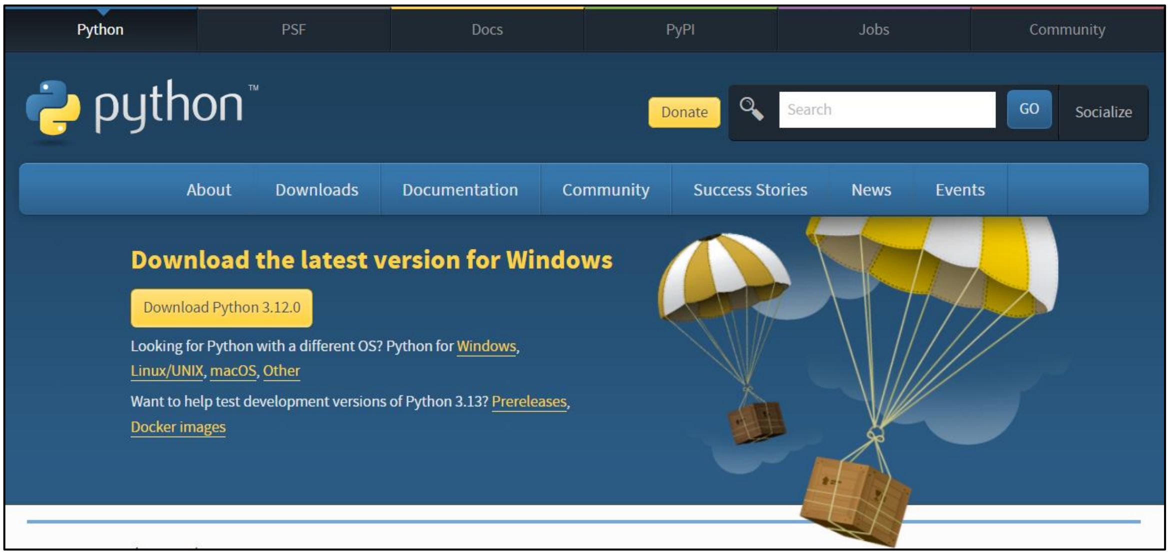
Figure 2: VS Code Extensions tab with 'python' search.

Install Python

To enable us to install additional project-specific packages and create virtual environments, Python should also be installed on the system.

Windows and Mac

Open a web browser, head to: https://www.python.org/downloads/ and download Python for your chosen operating system. Run the downloaded installer, making sure Python is added to your system PATH.



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python 3.13? Prereleases, Docker images

Figure 3: Official Python website with download links.

Linux

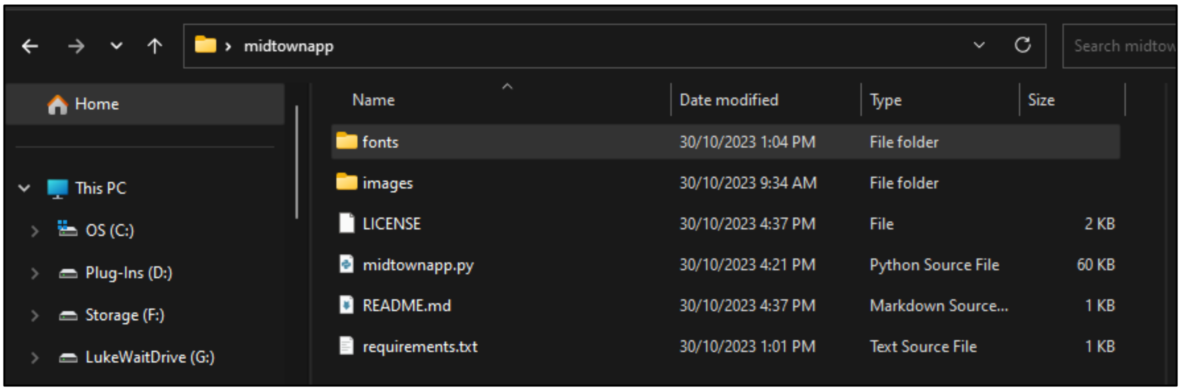
Open a terminal and execute the following commands:

- Debain: sudo apt update
  sudo apt install python3
  sudo apt install python3-pip
- Red Hat: sudo yum install python3
  sudo yum install python3-pip

Extract assets to working directory

Download the project files via: https://github.com/LukeWait/midtownapp

Unzip the file and move the internal folder 'midtownapp-main' to the desired working directory for the project (note: you can rename the folder 'midtownapp' and remove -main). You should have two folders with project assets, 'fonts' and 'images', and the other files required for the application. The project folder can be moved and renamed freely however the assets must remain in the folder for the script to locate them.



Name	Date modified	Type	Size
fonts	30/10/2023 1:04 PM	File folder	
images	30/10/2023 9:34 AM	File folder	
LICENSE	30/10/2023 4:37 PM	File	2 KB
midtownapp.py	30/10/2023 4:21 PM	Python Source File	60 KB
README.md	30/10/2023 4:37 PM	Markdown Source...	1 KB
requirements.txt	30/10/2023 1:01 PM	Text Source File	1 KB

Figure 4: Contents of unzipped GitHub download.

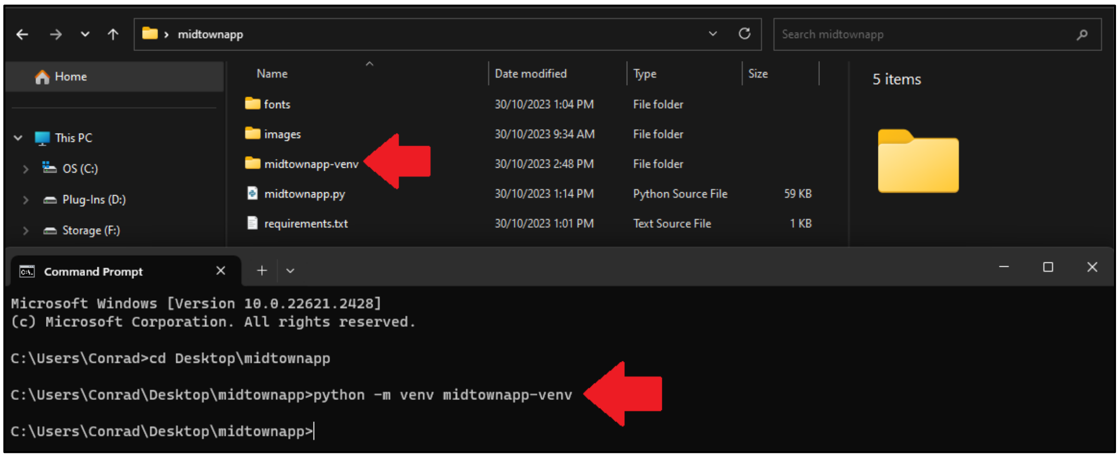
Creating a virtual environment (venv) Vs using system packages

Virtual environments in Python ensure project-specific dependencies, prevent conflicts, and maintain clean project setups. They are essential for isolated, organised, and reproducible Python development. For these reasons it is recommended to create a virtual environment and install the required packages within the venv.

Open a command prompt or terminal and navigate to the project directory. To create a virtual environment, use the following command (suggested name: midtownapp-venv):

- Windows: **python -m venv <virtual environment name>**
- Linux/Mac: **sudo apt install python3.11-venv**
  python3 -m venv <virtual environment name>

You should now see in the project directory a new folder with the chosen name.



Name	Date modified	Type	Size
fonts	30/10/2023 1:04 PM	File folder	
images	30/10/2023 9:34 AM	File folder	
midtownapp-venv	30/10/2023 2:48 PM	File folder	
midtownapp.py	30/10/2023 1:14 PM	Python Source File	59 KB
requirements.txt	30/10/2023 1:01 PM	Text Source File	1 KB

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Conrad>cd Desktop\midtownapp

C:\Users\Conrad\Desktop\midtownapp>python -m venv midtownapp-venv

C:\Users\Conrad\Desktop\midtownapp>
```

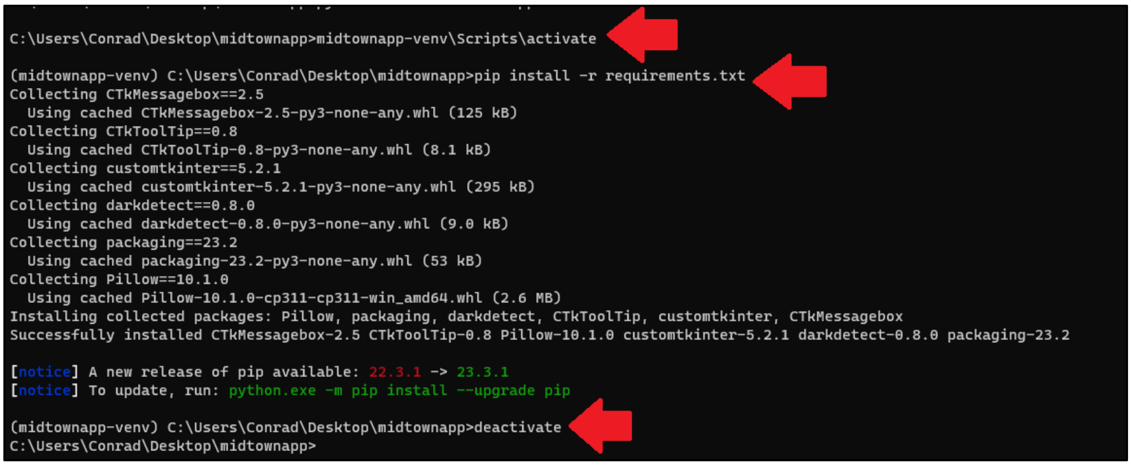
Figure 5: Creation of midtownapp virtual environment.

Now the virtual environment is created, ensure you are still in the project directory, and activate it by using the command:

- Windows: **midtownapp-venv\Scripts\activate**
- Linux/Mac: **source midtownapp-venv/bin/activate**

You'll notice (midtownapp-venv) appear before the directory, indicating it is active. While still in the project directory, we will install the required packages using the command: **pip install -r requirements.txt**

The requirements are now installed in the virtual environment and it can be closed at any time using the command: deactivate



```
C:\Users\Conrad\Desktop\midtownapp>midtownapp-venv\Scripts\activate

(midtownapp-venv) C:\Users\Conrad\Desktop\midtownapp>pip install -r requirements.txt
Collecting CTkMessagebox==2.5
  Using cached CTkMessagebox-2.5-py3-none-any.whl (125 kB)
Collecting CTkToolTip==0.8
  Using cached CTkToolTip-0.8-py3-none-any.whl (8.1 kB)
Collecting customtkinter==5.2.1
  Using cached customtkinter-5.2.1-py3-none-any.whl (295 kB)
Collecting darkdetect==0.8.0
  Using cached darkdetect-0.8.0-py3-none-any.whl (9.0 kB)
Collecting packaging==23.2
  Using cached packaging-23.2-py3-none-any.whl (53 kB)
Collecting Pillow==10.1.0
  Using cached Pillow-10.1.0-cp311-cp311-win_amd64.whl (2.6 MB)
Installing collected packages: Pillow, packaging, darkdetect, CTkToolTip, customtkinter, CTkMessagebox
Successfully installed CTkMessagebox-2.5 CTkToolTip-0.8 Pillow-10.1.0 customtkinter-5.2.1 darkdetect-0.8.0 packaging-23.2

[notice] A new release of pip available: 22.3.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(midtownapp-venv) C:\Users\Conrad\Desktop\midtownapp>deactivate
C:\Users\Conrad\Desktop\midtownapp>
```

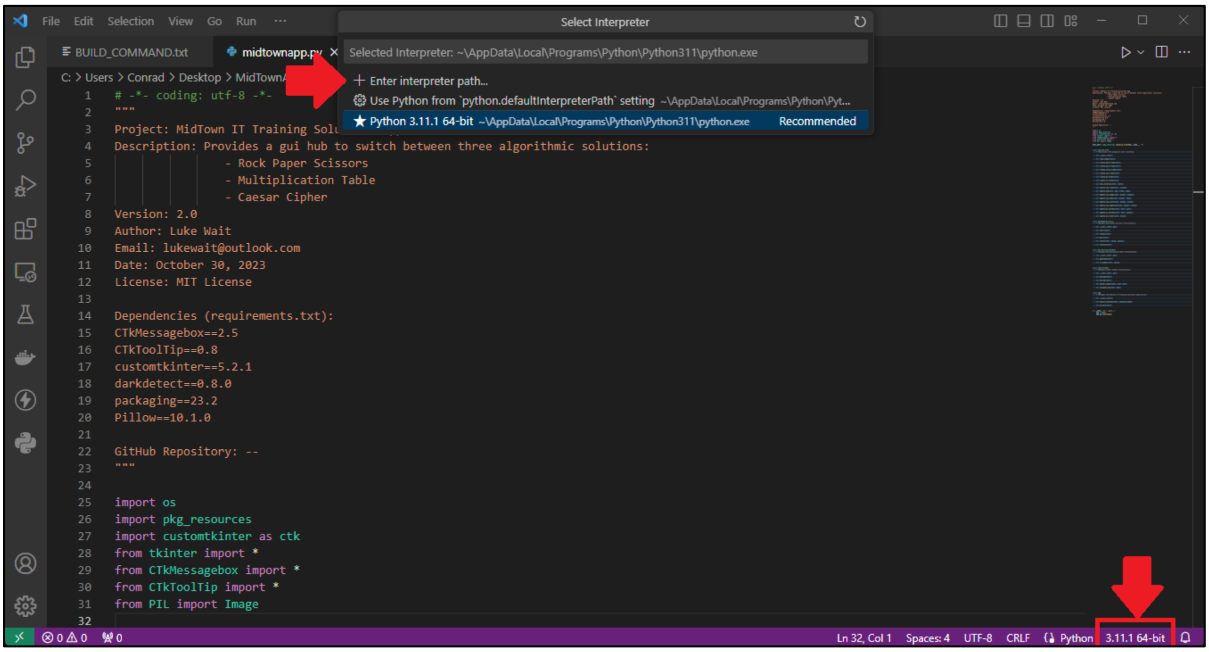
Figure 6: Installing requirements.txt in virtual environment.

Note: If you don't intend to use a virtual environment, the command **pip install -r requirements.txt** can still be used from to install the Python packages system-wide.

Interpret the script

Activate virtual environment and interpret script through VS Code IDE

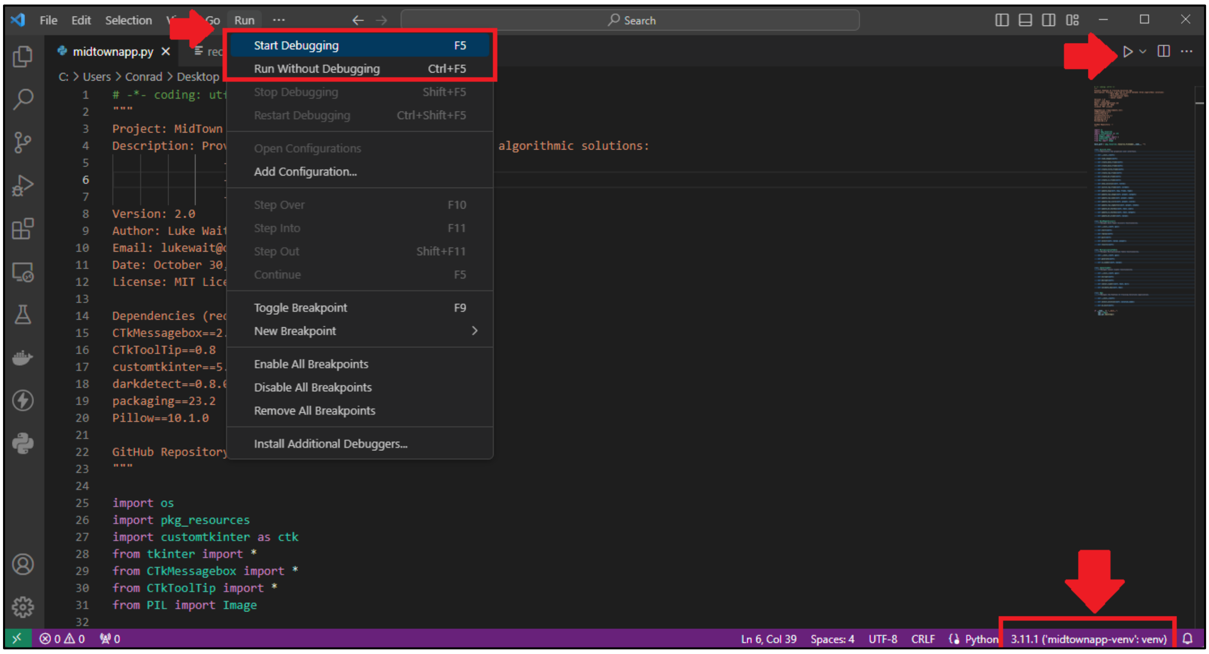
Start by opening the script with VS Code and changing the Interpreter to the newly created virtual environment in the midtownapp directory. The Interpreter being used can be seen in the bottom right corner of the VS Code window. To change this setting, hover over the indicated section and left-click. An option to ‘Enter interpreter path…’ will appear at the top. Either input the path directly or use the ‘Find…’ option to browse the file system and locate **<project_directory>\midtownapp-venv\Scripts\python.exe.**


File
Edit
Selection
View
Go
Run
Select Interpreter
BUILD_COMMAND.txt
midtownapp.py
C: > Users > Conrad > Desktop > MidTown
Selected Interpreter: ~\AppData\Local\Programs\Python\Python311\python.exe
+ Enter interpreter path...
Use Python from `python.defaultInterpreterPath` setting ~\AppData\Local\Programs\Python\Pyt...
★ Python 3.11.1 64-bit ~\AppData\Local\Programs\Python\Python311\python.exe Recommended
Ln 32, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.11.1 64-bit

```
# -*- coding: utf-8 -*-
"""
Project: MidTown IT Training Solu
Description: Provides a gui hub to switch between three algorithmic solutions:
                    - Rock Paper Scissors
                    - Multiplication Table
                    - Caesar Cipher
Version: 2.0
Author: Luke Wait
Email: lukewait@outlook.com
Date: October 30, 2023
License: MIT License

Dependencies (requirements.txt):
CTkMessagebox==2.5
CTkToolTip==0.8
customtkinter==5.2.1
darkdetect==0.8.0
packaging==23.2
Pillow==10.1.0

GitHub Repository: --
"""

import os
import pkg_resources
import customtkinter as ctk
from tkinter import *
from CTkMessagebox import *
from CTkToolTip import *
from PIL import Image
```

Figure 7: Enabling virtual environment in VS Code IDE.

VS Code will now be using midtownapp-venv, as evidenced in the bottom right corner. The script can now interpreted either by clicking the ‘play’ icon in the top right corner, or by accessing the ‘Run’ tab and choosing the appropriate option (includes keyboard shortcuts).



```
# -*- coding: ut
"""
Project: MidTown
Description: Prov                          algorithmic solutions:



Version: 2.0
Author: Luke Wait
Email: lukewait@c
Date: October 30,
License: MIT Lice

Dependencies (rec
CTkMessagebox==2.
CTkToolTip==0.8
customtkinter==5.
darkdetect==0.8.0
packaging==23.2
Pillow==10.1.0

GitHub Repository
"""

import os
import pkg_resources
import customtkinter as ctk
from tkinter import *
from CTkMessagebox import *
from CTkToolTip import *
from PIL import Image
```

Figure 8: Checking the interpreter in use and running script.

Activate venv and interpret script directly through terminal

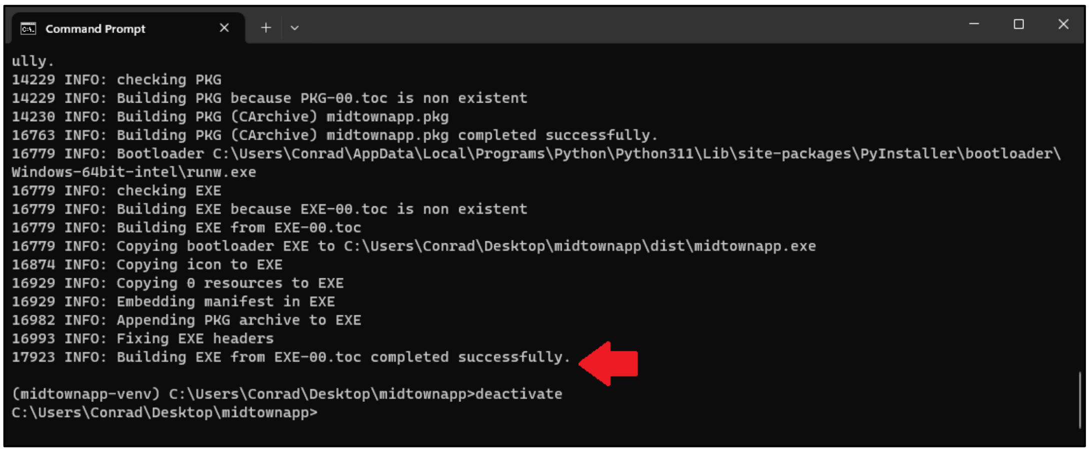
Open a terminal/command prompt and cd to the midtownapp directory.

- Activate the venv with command: **midtownapp-venv\Scripts\activate** or **source midtownapp-venv/bin/activate**
- Execute the midtownapp.py file: **python midtownapp.py** or **python3 midtownapp.py**

Build and run executable file

Another option is to create an executable file using pyinstaller. This will create one file that will be completely self-contained, meaning it won't require the virtual environment or installed packages to run - perfect for distribution. To create an executable for your chosen operating system, ensure the venv is activated and the directory is set to the project directory, then use the command:

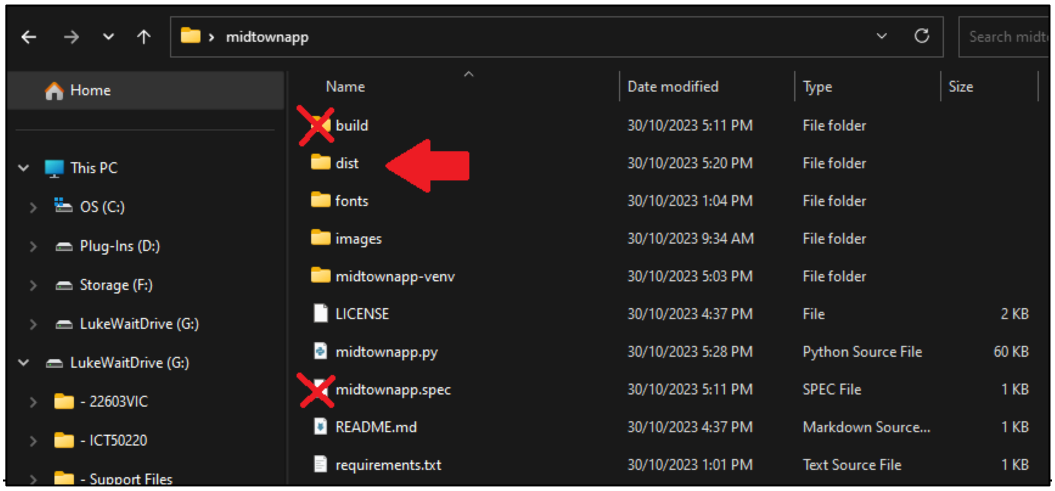
- Windows: **pyinstaller --onefile --add-data "images;images" --add-data "fonts;fonts" --noconsole midtownapp.py**
- Linux/Mac: **pyinstaller --onefile --add-data "images:images" --add-data "fonts:fonts" --noconsole midtownapp.py**



```
ully.
14229 INFO: checking PKG
14229 INFO: Building PKG because PKG-00.toc is non existent
14230 INFO: Building PKG (CArchive) midtownapp.pkg
16763 INFO: Building PKG (CArchive) midtownapp.pkg completed successfully.
16779 INFO: Bootloader C:\Users\Conrad\AppData\Local\Programs\Python\Python311\Lib\site-packages\PyInstaller\bootloader\
Windows-64bit-intel\runw.exe
16779 INFO: checking EXE
16779 INFO: Building EXE because EXE-00.toc is non existent
16779 INFO: Building EXE from EXE-00.toc
16779 INFO: Copying bootloader EXE to C:\Users\Conrad\Desktop\midtownapp\dist\midtownapp.exe
16874 INFO: Copying icon to EXE
16929 INFO: Copying 0 resources to EXE
16929 INFO: Embedding manifest in EXE
16982 INFO: Appending PKG archive to EXE
16993 INFO: Fixing EXE headers
17923 INFO: Building EXE from EXE-00.toc completed successfully.

(midtownapp-venv) C:\Users\Conrad\Desktop\midtownapp>deactivate
C:\Users\Conrad\Desktop\midtownapp>
```

Figure 9: Building an executable with pyinstaller.

When the process is complete you will notice some new files and folders in the project directory. The 'build' folder and '.spec' file can be deleted as they are only used in the compilation process. The exe will be available is the 'dist' folder and can be moved, renamed, and launched from any location.



Name	Date modified	Type	Size
build	30/10/2023 5:11 PM	File folder	
dist	30/10/2023 5:20 PM	File folder	
fonts	30/10/2023 1:04 PM	File folder	
images	30/10/2023 9:34 AM	File folder	
midtownapp-venv	30/10/2023 5:03 PM	File folder	
LICENSE	30/10/2023 4:37 PM	File	2 KB
midtownapp.py	30/10/2023 5:28 PM	Python Source File	60 KB
midtownapp.spec	30/10/2023 5:11 PM	SPEC File	1 KB
README.md	30/10/2023 4:37 PM	Markdown Source...	1 KB
requirements.txt	30/10/2023 1:01 PM	Text Source File	1 KB

Figure 10: Locating executable and removing unnecessary files.

Navigating the MidTown IT Training Solutions App

Once the script is run, either through the venv, IDE, terminal, or executable file, you will see the fully resizable GUI window that will enable you to navigate between the three algorithmic solutions: Rock Paper Scissors, Multiplication Tables, and Caesar Cipher.

To move between the solutions, simply click on the title in the left-hand sidebar. The state of a solution will persist, meaning you can move freely between them without losing data.

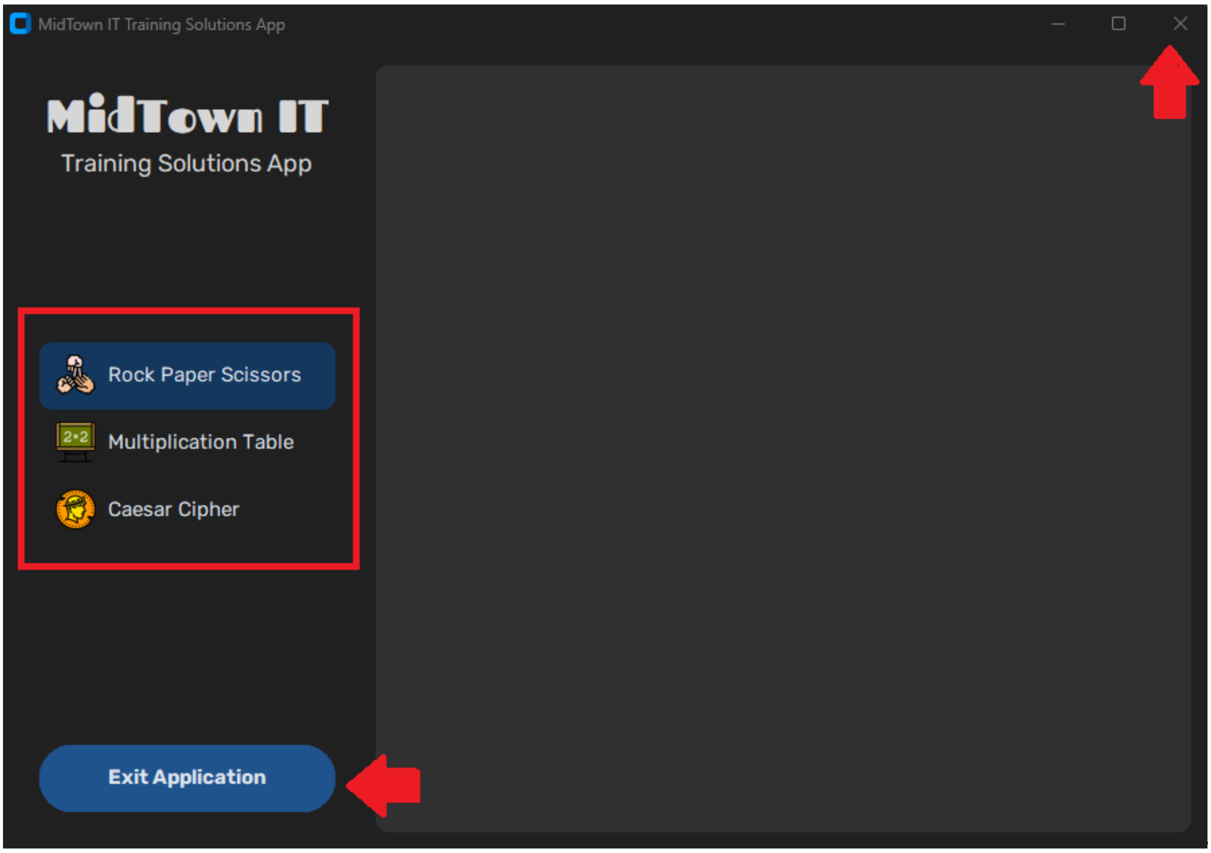
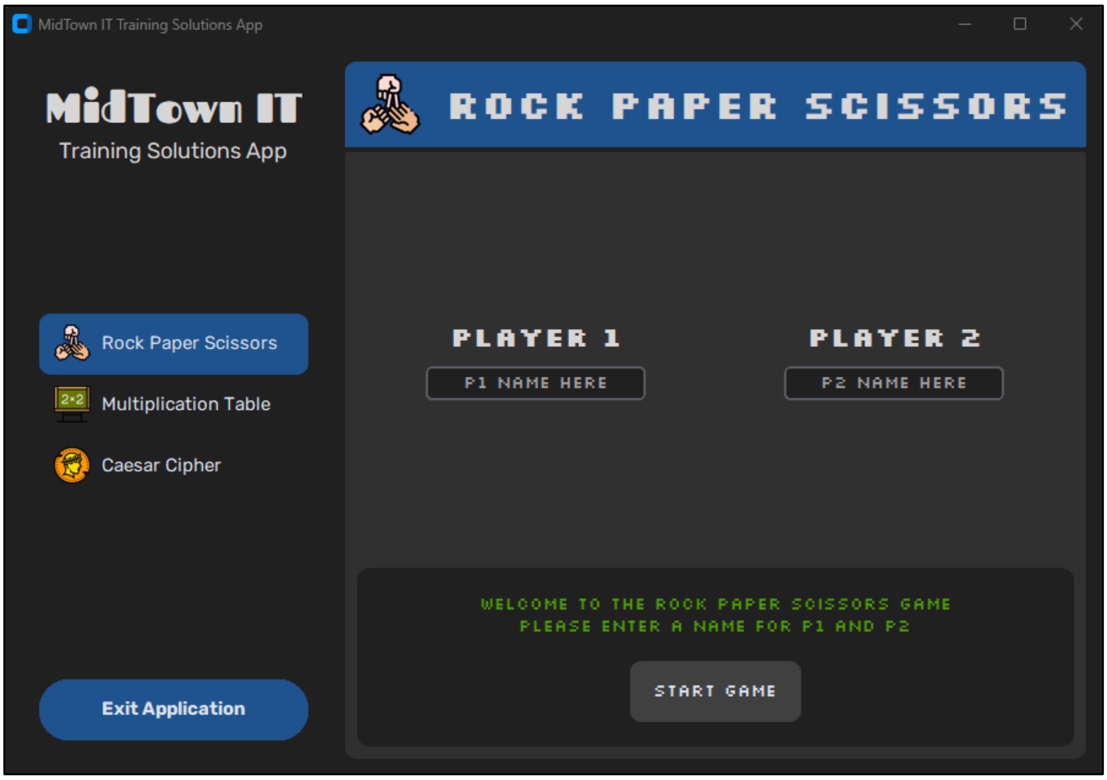


Figure 11: MidTown IT Training Solutions App launch screen.

The application can be closed, and the solutions terminated gracefully, at any time by using the ‘Exit Application’ button or by clicking the X in the top right corner.

PART 2 - Rock Paper Scissors

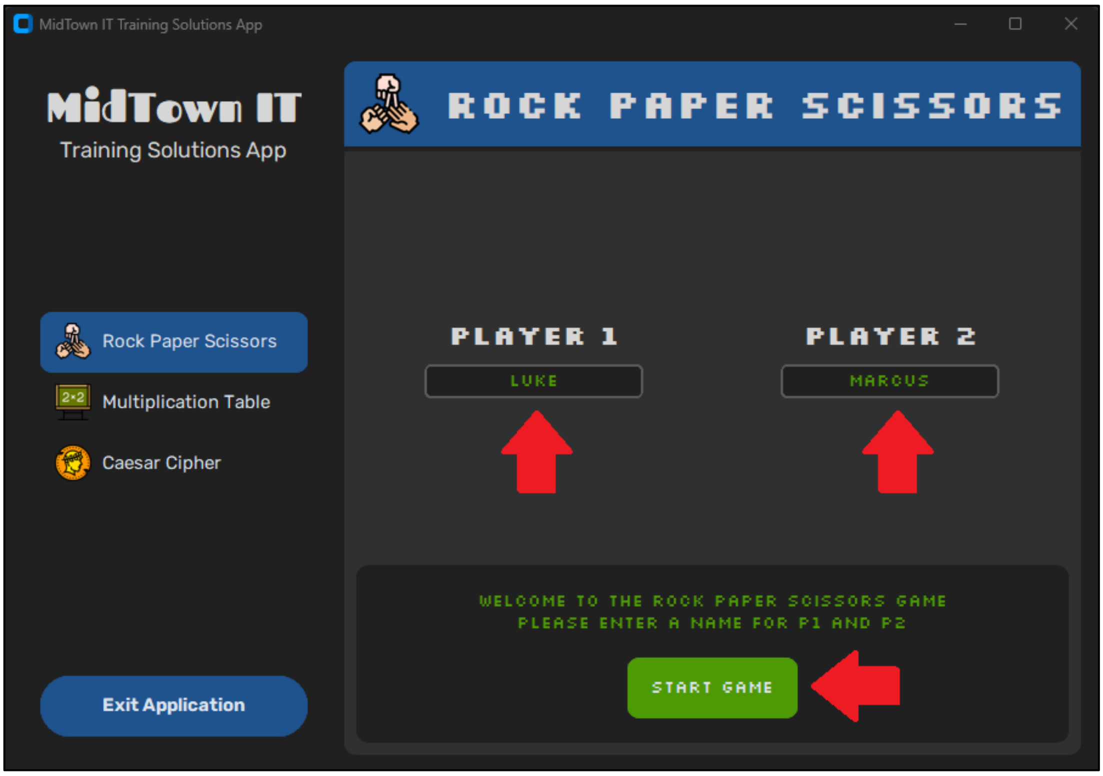
Select Rock Paper Scissors from the side-menu to display the title screen and welcome message.



WELCOME TO THE ROCK PAPER SCISSORS GAME
PLEASE ENTER A NAME FOR P1 AND P2

Figure 12: Rock Paper Scissors title screen.

Enter a username in the provided entry fields for PLAYER 1 and PLAYER 2 before clicking on the ‘START GAME’ button, or using the Enter key while the cursor is in either entry field.



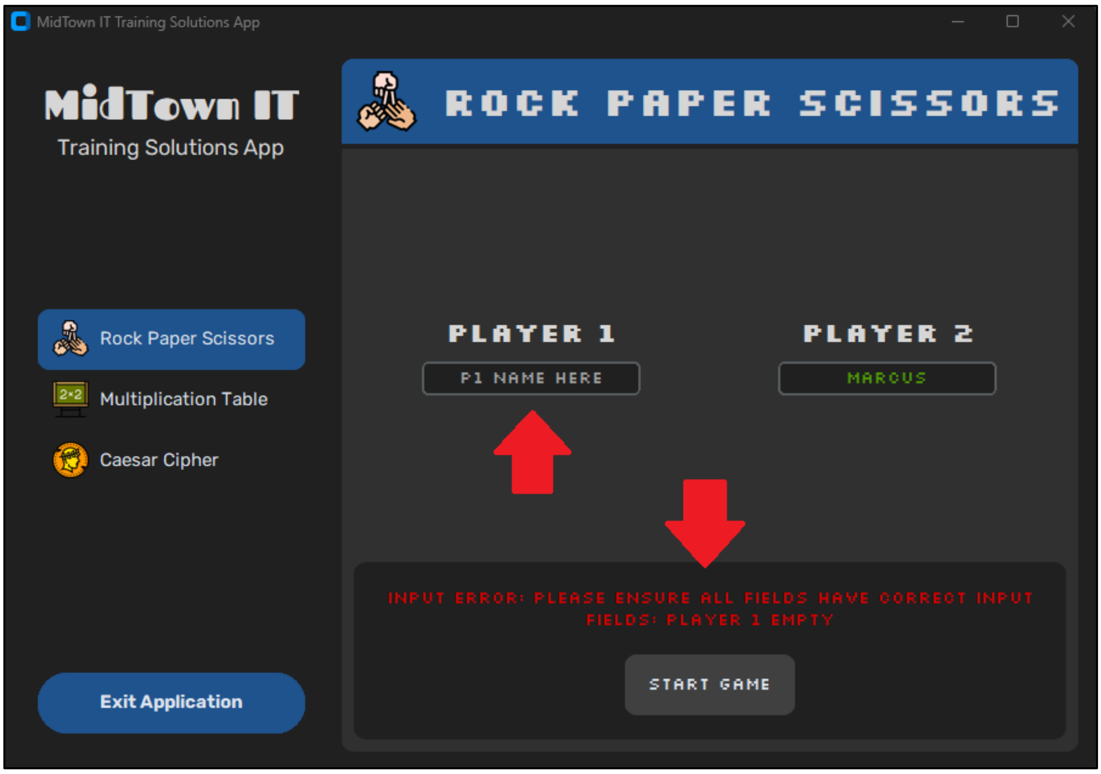
PLAYER 1: LUKE

PLAYER 2: MARCUS

WELCOME TO THE ROCK PAPER SCISSORS GAME
PLEASE ENTER A NAME FOR P1 AND P2

Figure 13: Entry field requirements for starting game.

If either of the fields are empty an error message will display notifying the user which fields are empty. Note: There is an input limit of 12 characters, however any combination of characters can be used, including symbols and space.



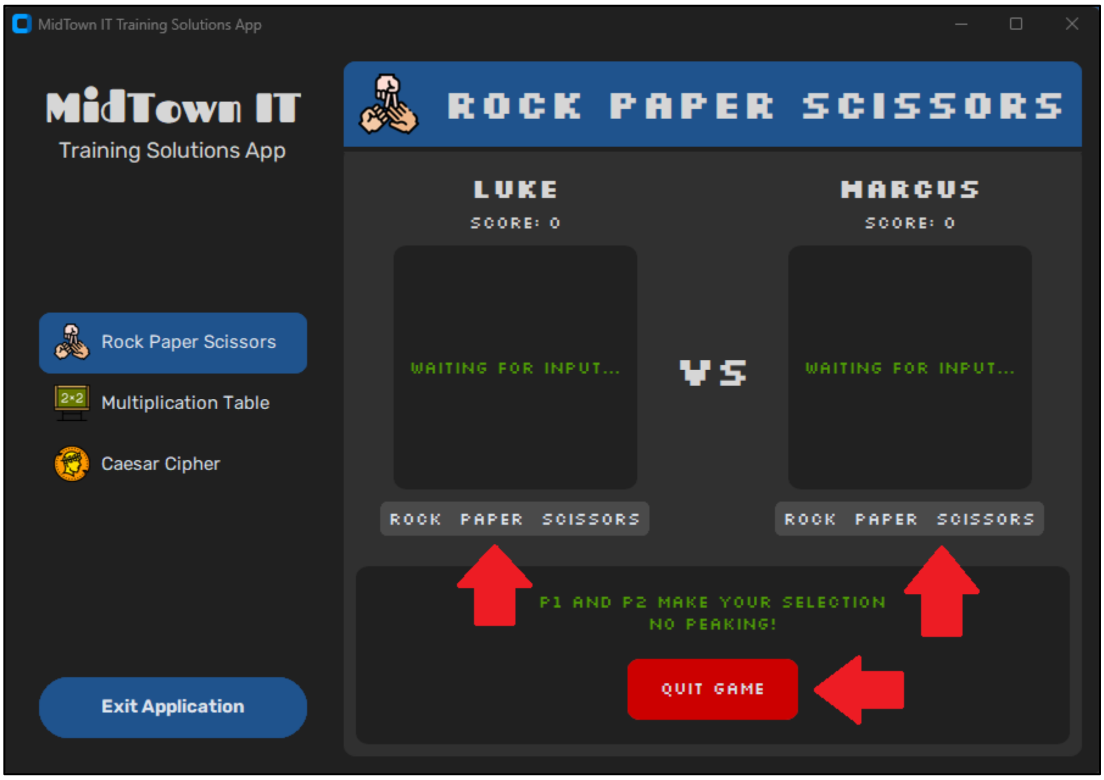
P1 NAME HERE

MARCUS

INPUT ERROR: PLEASE ENSURE ALL FIELDS HAVE CORRECT INPUT
FIELDS: PLAYER 1 EMPTY

Figure 14: Input error showing specific fields in need of remedy.

Once the game has begun, you will have the option for a player (either player can go first) to make their selection with a segmented button. Note: You can also quit the game and return to the title screen at any time with the 'QUIT GAME' button.



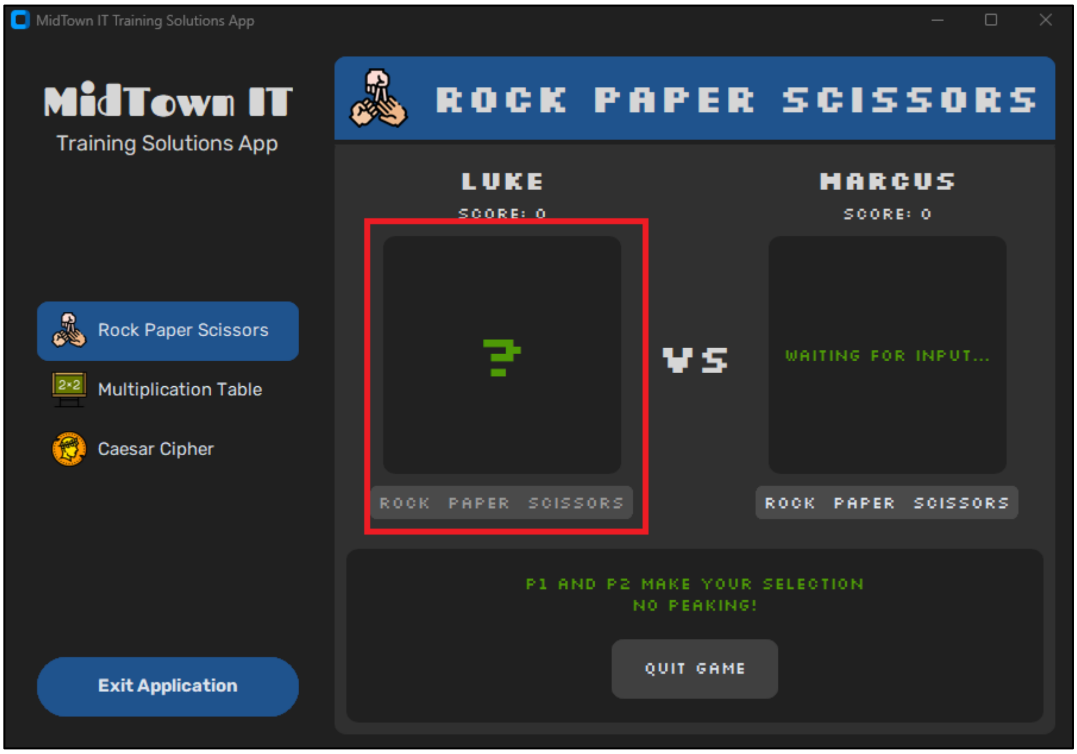
WAITING FOR INPUT...

WAITING FOR INPUT...

P1 AND P2 MAKE YOUR SELECTION
NO PEAKING!

Figure 15: Game buttons for player choices and 'quit game' button.

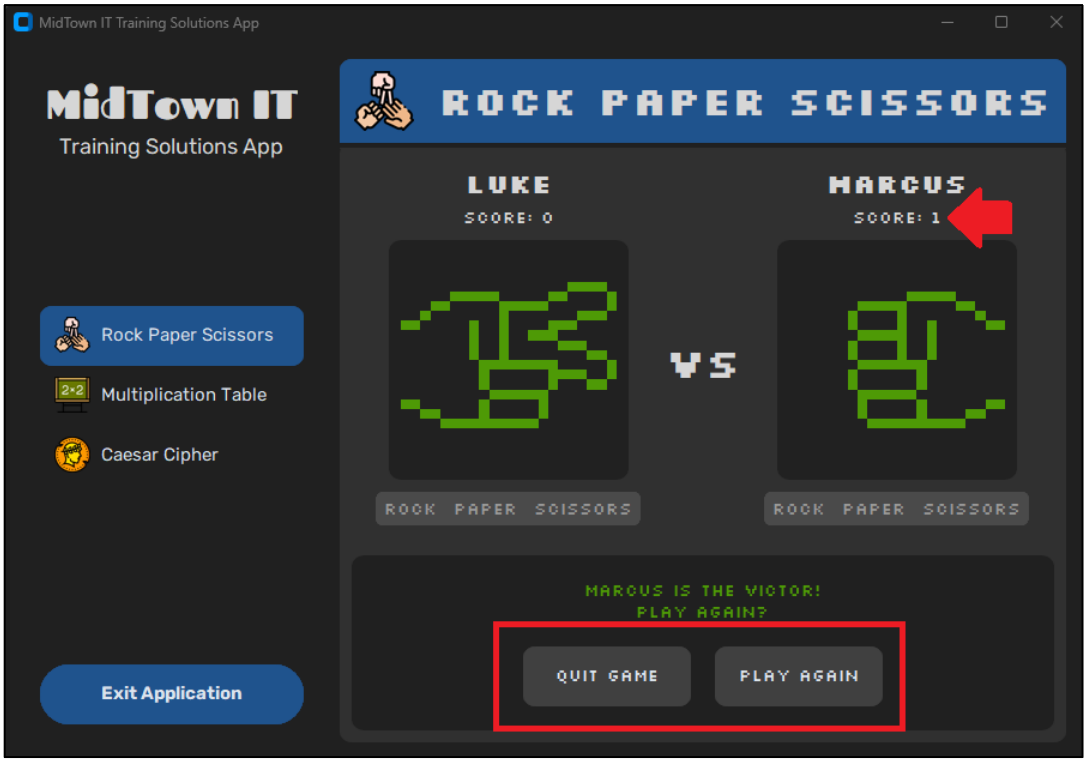
When a player makes a selection, the image will change to indicate that they have chosen, and the segmented button will become disabled.



P1 AND P2 MAKE YOUR SELECTION
NO PEAKING!

Figure 16: Updates to GUI following first player to input selection.

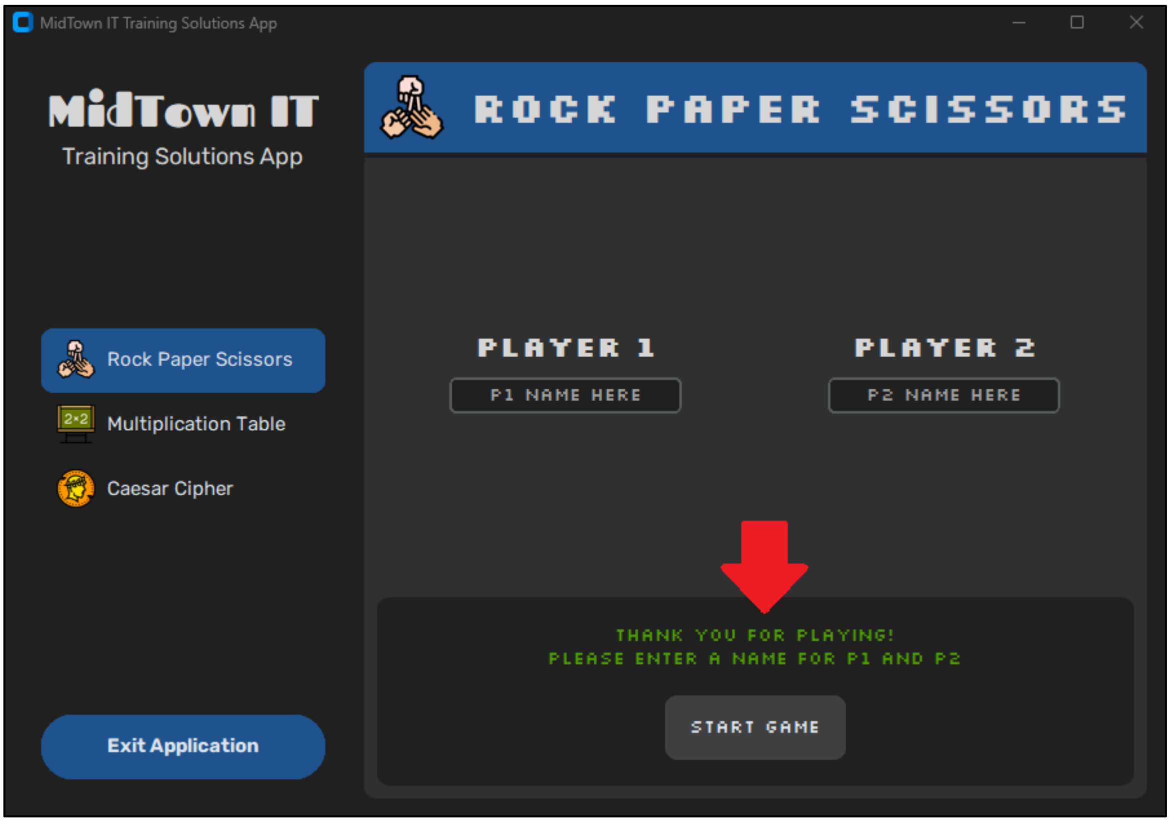
Once the second player has made their selection the images are updated to show the choices made and the result is displayed in the bottom text area. The winners score will also be incremented by one, with a tie resulting in no points being awarded. The option to 'QUIT GAME' or 'PLAY AGAIN' will be made available. Choosing to play again will reset the images and re-enable the segmented buttons so more rounds can be played.



MARCUS IS THE VICTOR!
PLAY AGAIN?

Figure 17: Post-game screen with updated score and option to play again.

Choosing to quit the game session will reset the score and usernames so new players can start a game. Upon quitting, the title screen will now display a ‘thank you for playing’ message.

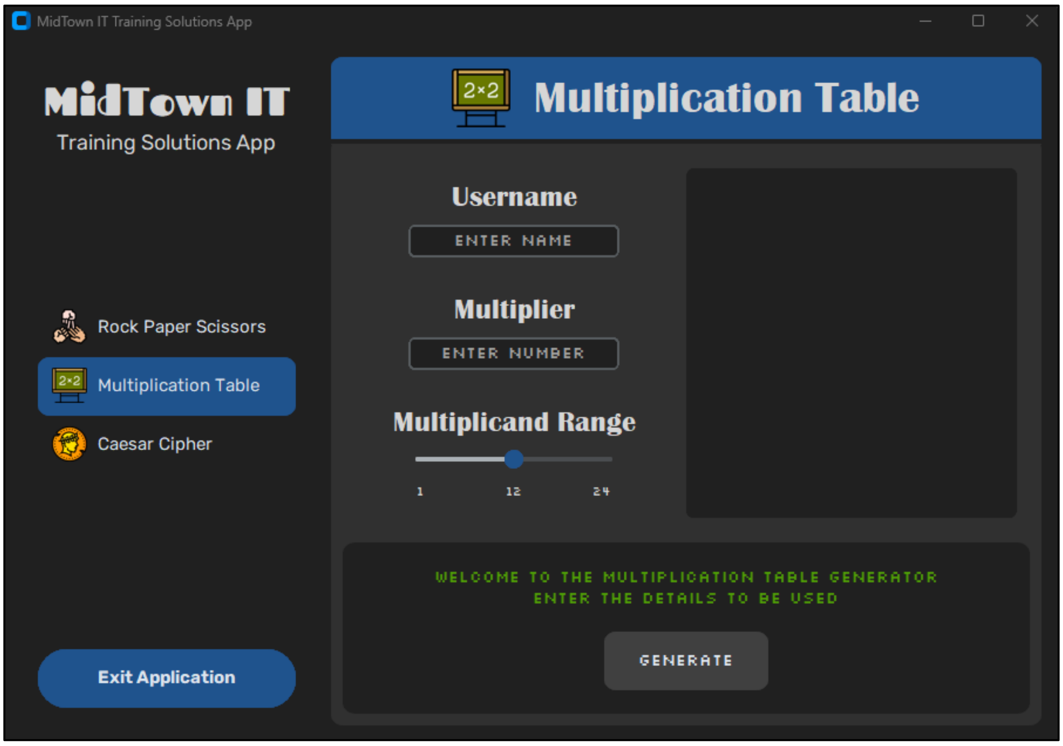


THANK YOU FOR PLAYING!
PLEASE ENTER A NAME FOR P1 AND P2

Figure 18: Modified title screen after a game has been quit.

PART 3 - Multiplication Tables

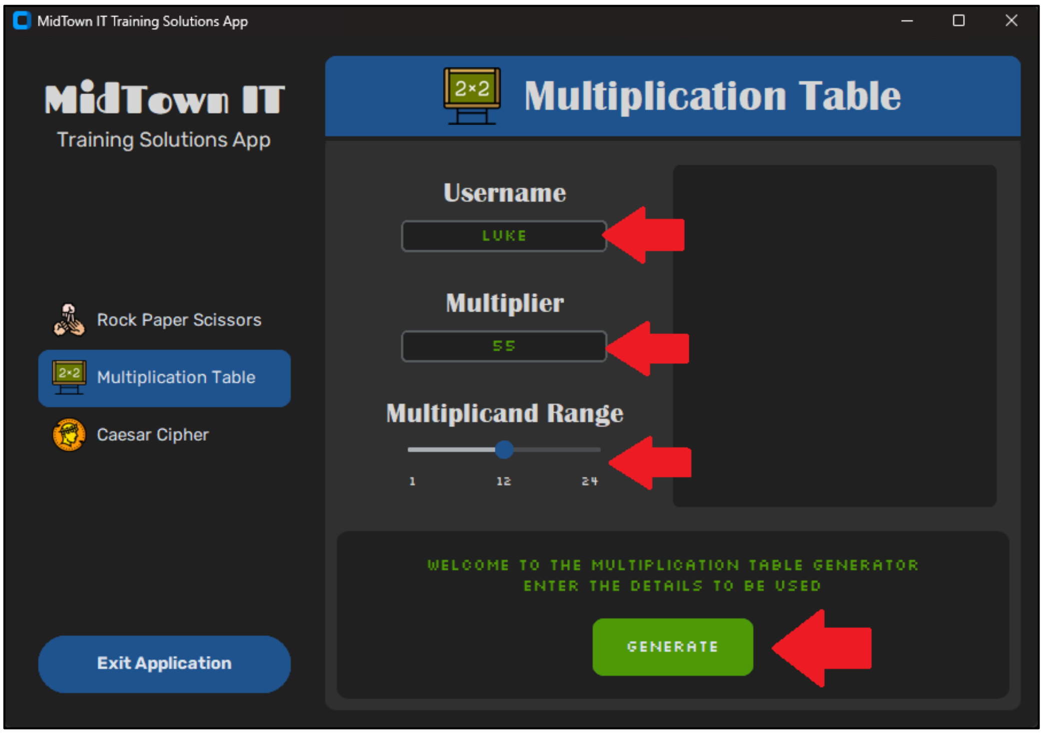
Select Multiplication Table from the side-menu to display the title screen and welcome message.



WELCOME TO THE MULTIPLICATION TABLE GENERATOR
ENTER THE DETAILS TO BE USED

Figure 19: Multiplication Table title screen.

Enter a username and multiplier into the provided entry fields for Username and Multiplier. Choose a Multiplicand Range using the slider (1-24. This indicates the max multiplicand to be used, with the min preset to 1) before clicking on the ‘Generate’ button, or using the Enter key while the cursor is in either entry field..



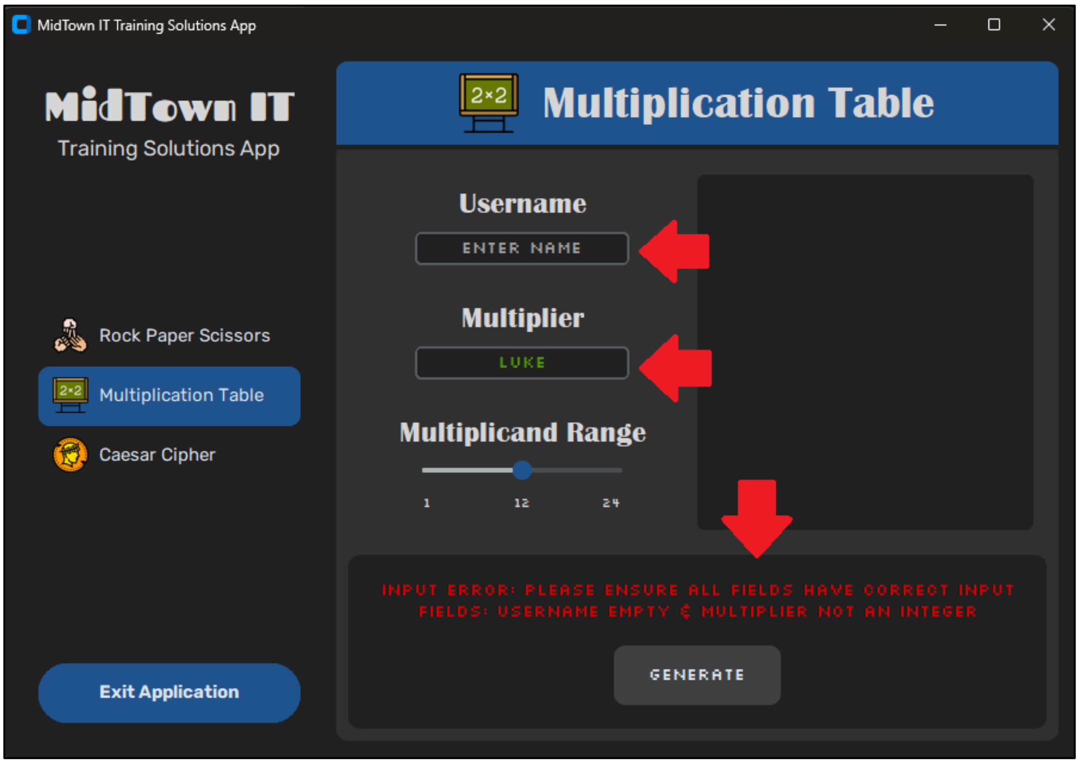
Username: LUKE

Multiplier: 55

WELCOME TO THE MULTIPLICATION TABLE GENERATOR
ENTER THE DETAILS TO BE USED

Figure 20: Entry field requirements for generating a multiplication table.

If either of the fields are empty, or the Multiplier field contains a non-numeric value (negative numbers are ok), an error message will display notifying the user which fields don’t meet the requirements. Note: There is an input limit of 12 characters placed on the entry fields. Also, a failed table generation will cause the textbox to be cleared of a pre-existing table.



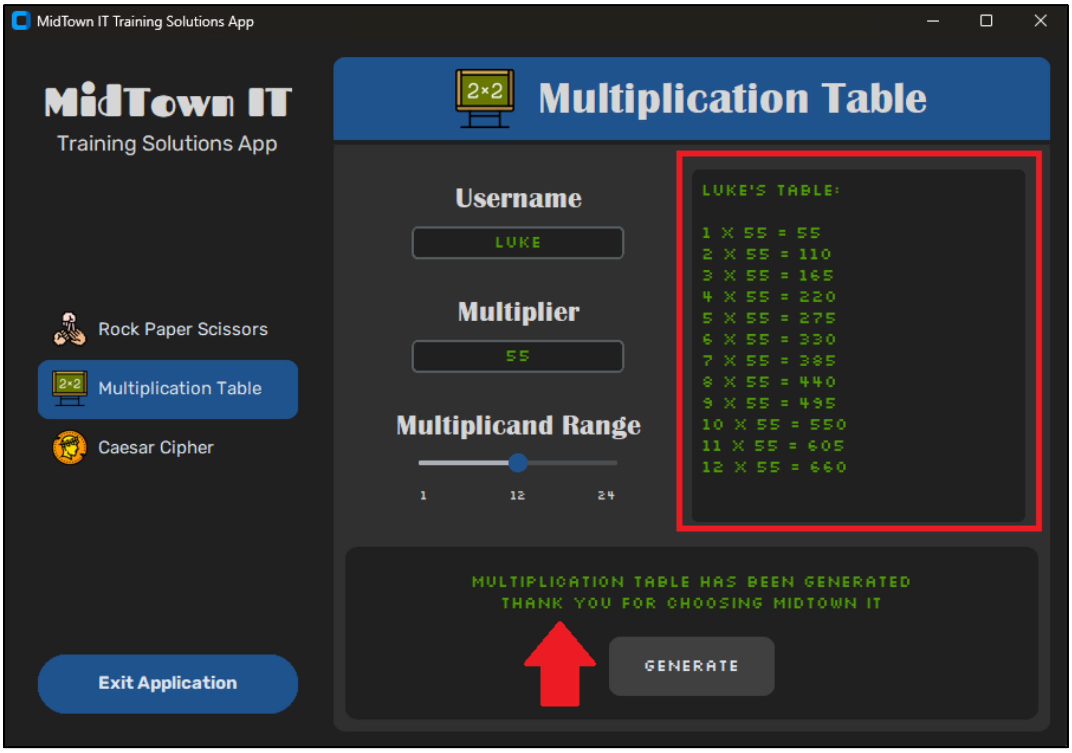
Username: ENTER NAME

Multiplier: LUKE

INPUT ERROR: PLEASE ENSURE ALL FIELDS HAVE CORRECT INPUT
FIELDS: USERNAME EMPTY & MULTIPLIER NOT AN INTEGER

Figure 21: Input error showing specific fields in need of remedy.

Once valid inputs have been input and the ‘Generate’ button clicked, the requested multiplication table will be displayed in the textbox with the username. Successfully table generation will also have a thank you message display at the bottom.



Username: LUKE

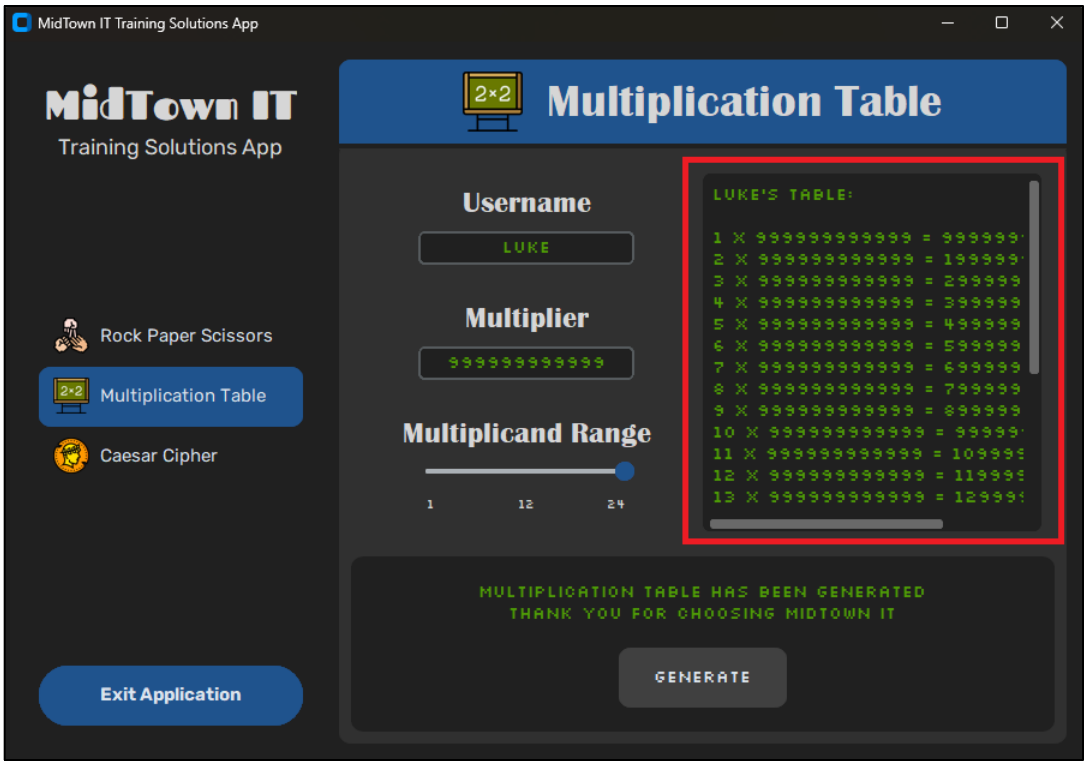
Multiplier: 55

```
LUKE'S TABLE:

1 X 55 = 55
2 X 55 = 110
3 X 55 = 165
4 X 55 = 220
5 X 55 = 275
6 X 55 = 330
7 X 55 = 385
8 X 55 = 440
9 X 55 = 495
10 X 55 = 550
11 X 55 = 605
12 X 55 = 660
```

MULTIPLICATION TABLE HAS BEEN GENERATED
THANK YOU FOR CHOOSING MIDTOWN IT

Figure 22: Successful multiplication table generation.

When inputting large numbers, the resulting multiplication table may extend beyond the boundaries of the textbox. In this event, scroll bars will become available. Alternatively, the MidTown IT Training Solutions window can be resized/maximized to increase the size of the textbox.



Username: LUKE

Multiplier: 999999999999

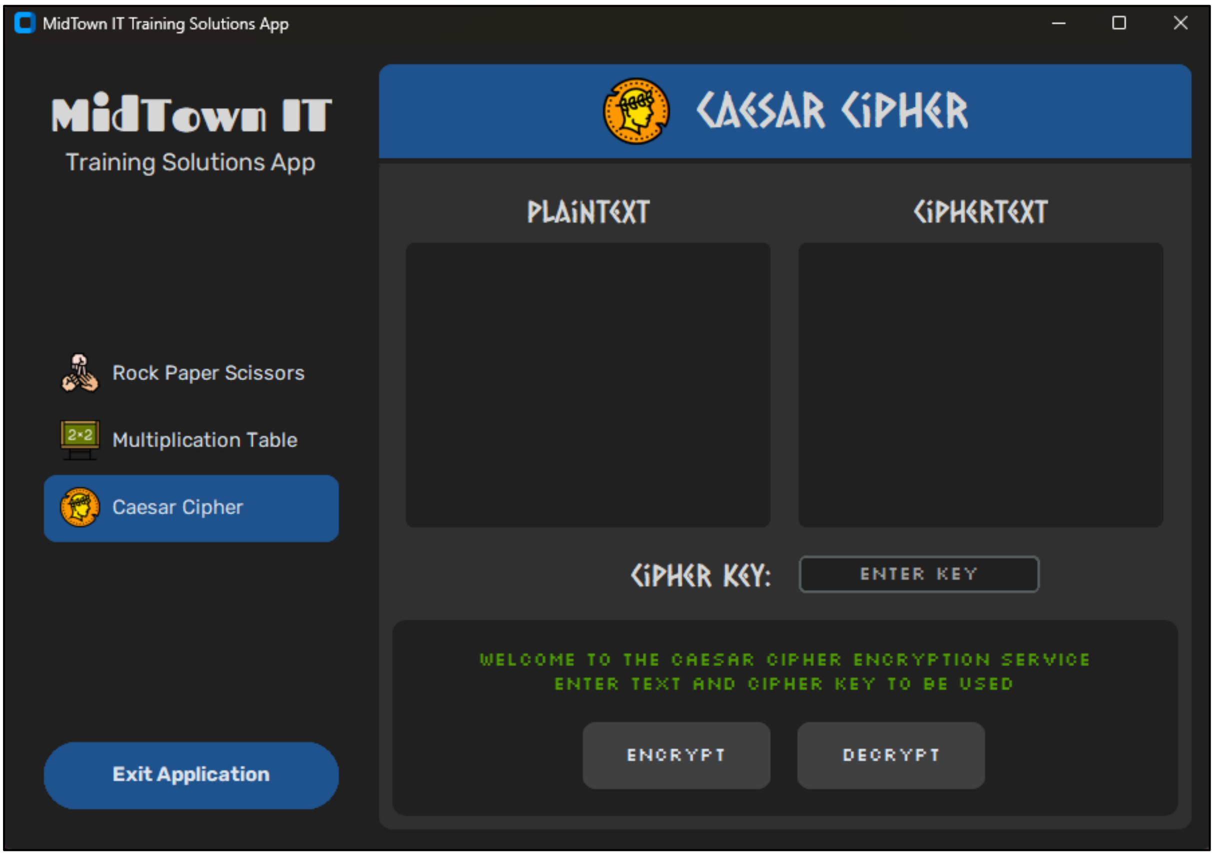
```
LUKE'S TABLE:

1 X 999999999999 = 999999
2 X 999999999999 = 199999
3 X 999999999999 = 299999
4 X 999999999999 = 399999
5 X 999999999999 = 499999
6 X 999999999999 = 599999
7 X 999999999999 = 699999
8 X 999999999999 = 799999
9 X 999999999999 = 899999
10 X 999999999999 = 99999
11 X 999999999999 = 10999
12 X 999999999999 = 11999
13 X 999999999999 = 12999
```

MULTIPLICATION TABLE HAS BEEN GENERATED
THANK YOU FOR CHOOSING MIDTOWN IT

Figure 23: Multiplication table textbox does not word-wrap to keep operations in line.

PART 4 - Caesar Cipher

Select Caesar Cipher from the side-menu to display the title screen and welcome message.

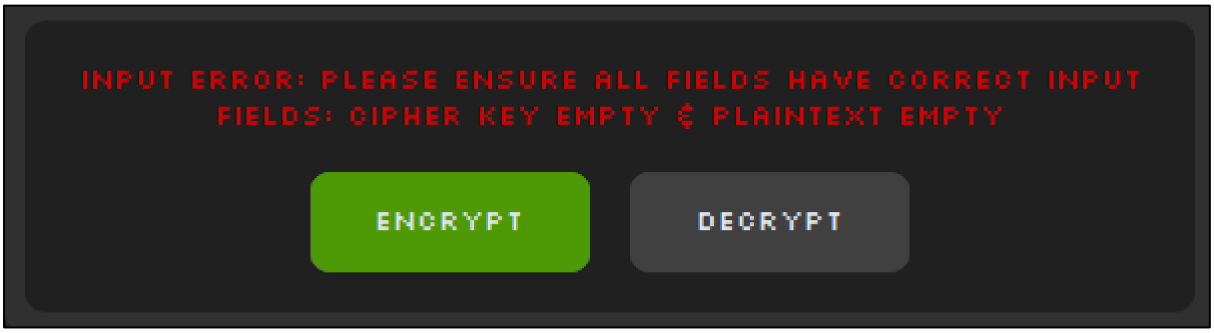


WELCOME TO THE CAESAR CIPHER ENCRYPTION SERVICE
ENTER TEXT AND CIPHER KEY TO BE USED

Figure 24: Caesar Cipher solution title screen.

To encrypt and decrypt text messages, ensure a CIPHER KEY has been entered into the provided entry field. Note: There is an input limit of 12 characters, however any combination of characters can be used, including alpha-numeric characters and symbols, as they will be converted to ascii values. This ensures any input can be used and the Caesar Cipher will function as intended.
You will also need to enter a message into the relevant textbox: PLAINTEXT when using the ‘ENCRYPT’ button, and CYPHERTEXT when using the ‘DECRYPT’ button.

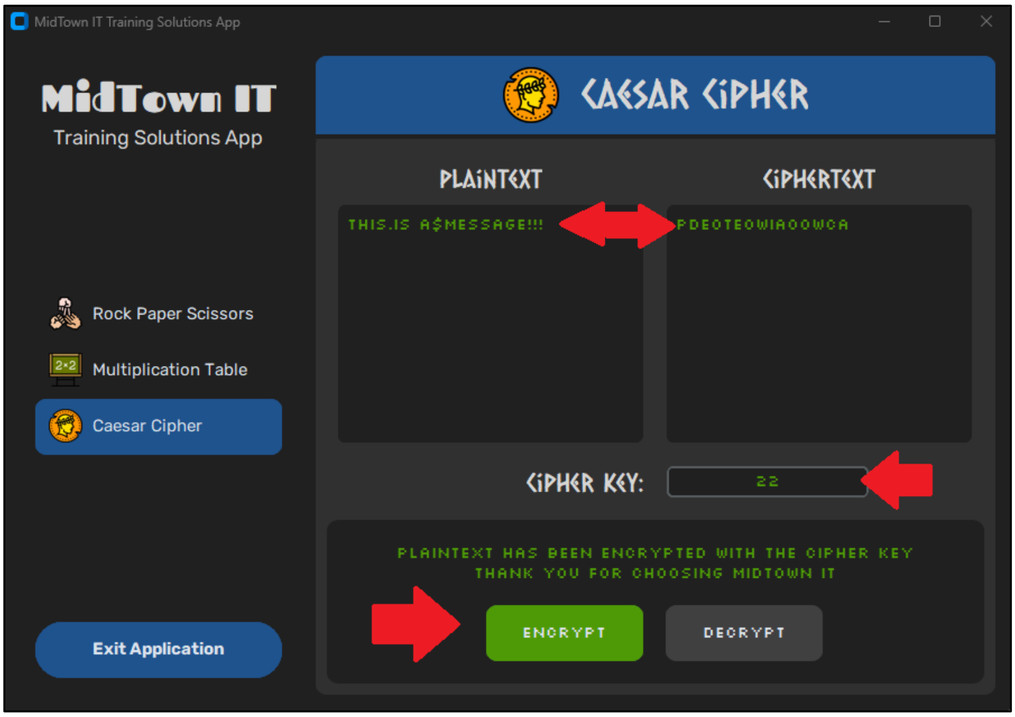
If any required fields are missing when clicking either of the buttons, an error message will display informing the user which missing fields are still required to perform the selected process.



INPUT ERROR: PLEASE ENSURE ALL FIELDS HAVE CORRECT INPUT
FIELDS: CIPHER KEY EMPTY & PLAINTEXT EMPTY

Figure 25: Error messages showing specific entry field requirements.

If a message exits in the PLAINTEXT field and a CIPHER KEY has been entered when the user clicks the ‘ENCRYPT’ button, the message will be encrypted and displayed in the CIPHERTEXT field. Note: While symbols and spaces are accepted as valid input, they are removed for the sake of the cipher. In addition, all alpha characters are converted to uppercase (though this will not be noticeable as the chosen font always shows uppercase) and full stops are converted to ‘X’ before encryption.



PLAINTEXT

THIS.IS A$MESSAGE!!!

CIPHERTEXT

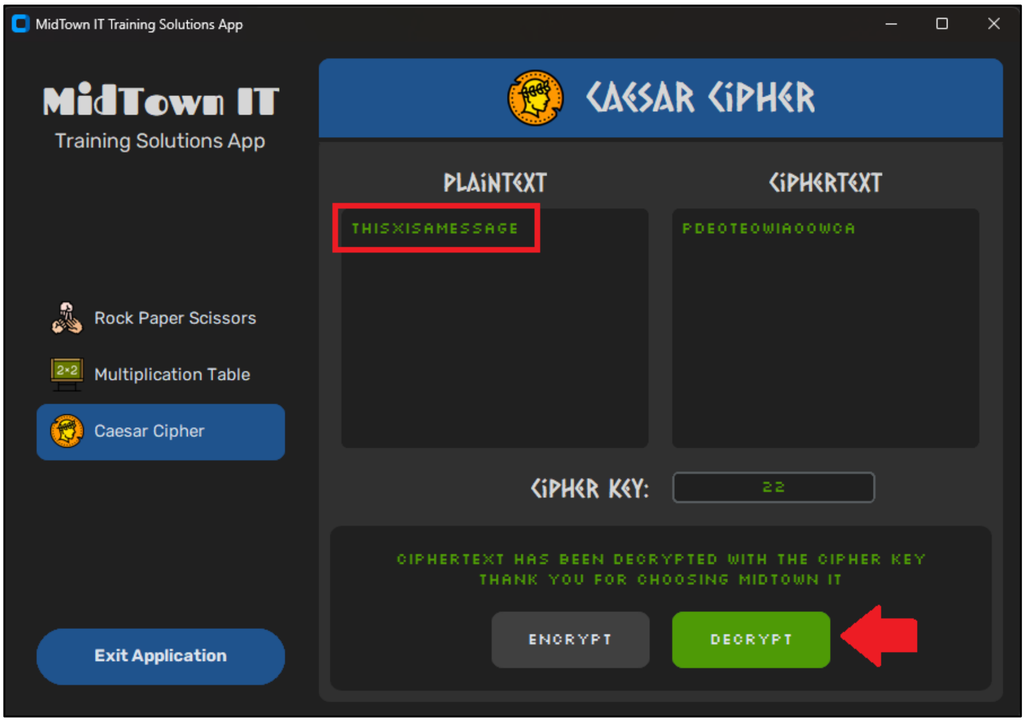
PDEOTEOWIAOOWCA

CIPHER KEY: 22

PLAINTEXT HAS BEEN ENCRYPTED WITH THE CIPHER KEY
THANK YOU FOR CHOOSING MIDTOWN IT

Figure 26: A plaintext message successfully converted to ciphertext.

When a CIPHERTEXT message is decrypted using the same CIPHER KEY, the decrypted message will be shown in the PLAINTEXT field. Notice the symbols and spaces are removed, and the full stops replaced with ‘X’.



PLAINTEXT

THISXISAMESSAGE

CIPHERTEXT

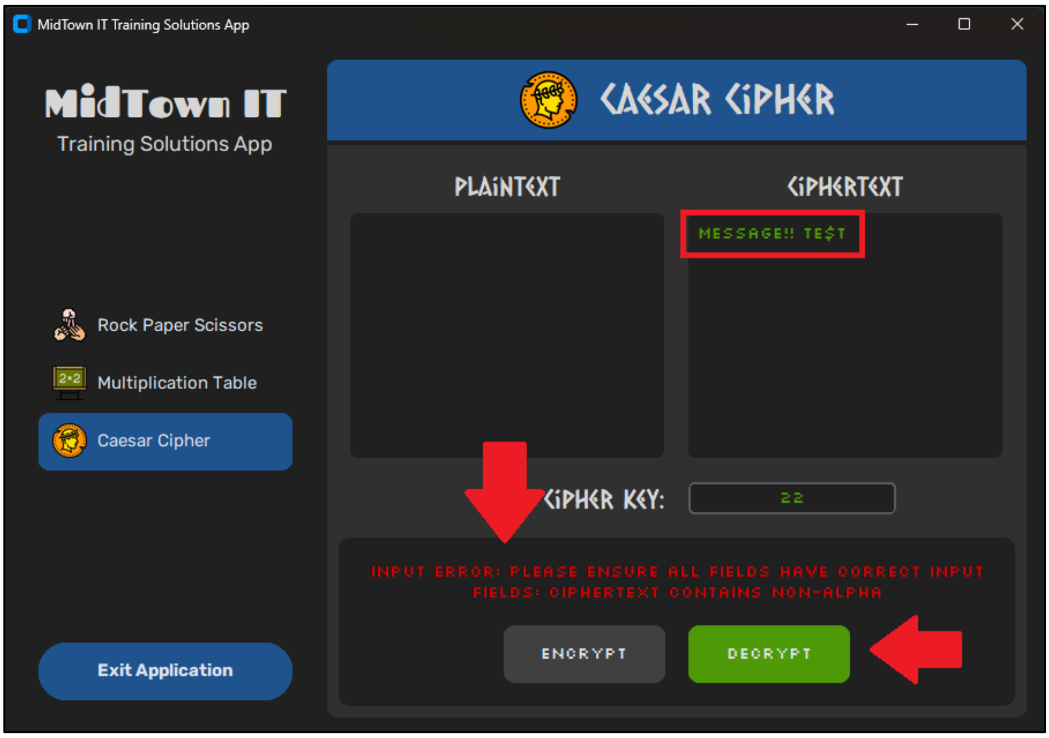
PDEOTEOWIAOOWCA

CIPHER KEY: 22

CIPHERTEXT HAS BEEN DECRYPTED WITH THE CIPHER KEY
THANK YOU FOR CHOOSING MIDTOWN IT

Figure 27: Decrypted ciphertext using 'X' as a full stop.

If the user attempts to decrypt CIPHERTEXT that contains any non-alpha characters, an error message will be displayed. Note: Whenever an invalid encryption/decryption occurs, the opposing text field will be cleared.



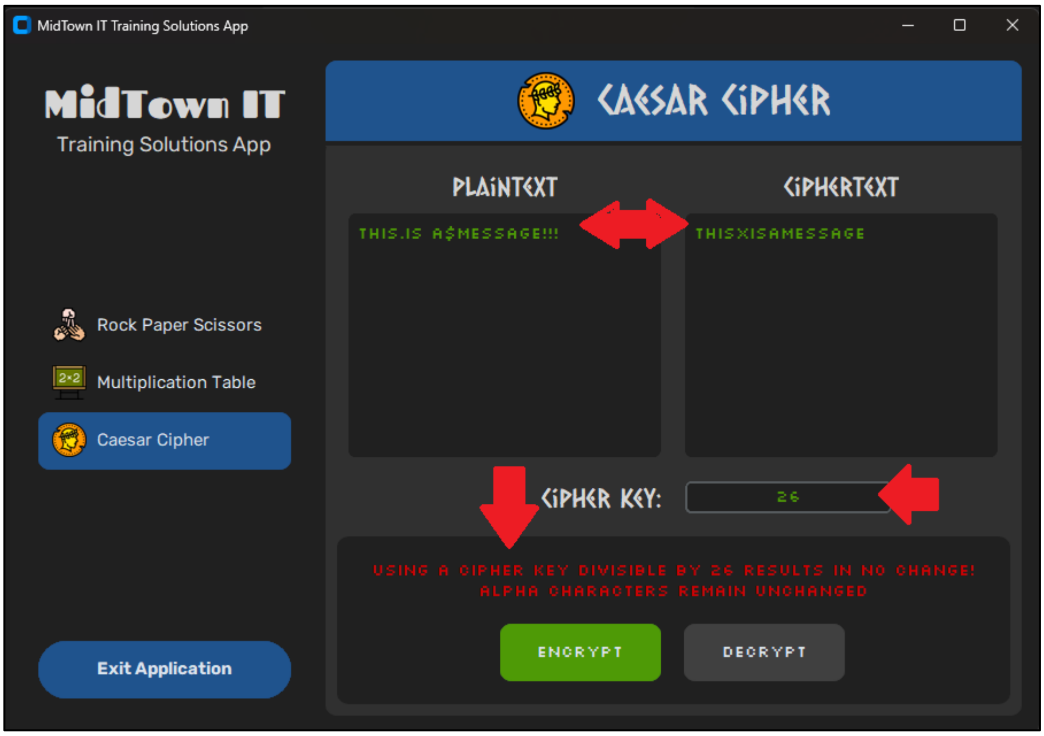
MESSAGE!! TE$T

22

INPUT ERROR: PLEASE ENSURE ALL FIELDS HAVE CORRECT INPUT
FIELDS: CIPHERTEXT CONTAINS NON-ALPHA

Figure 28: Decrypting invalid ciphertext.

When a user encrypts/decrypts a message with a CIPHER KEY divisible by 26 an anomaly occurs, and the resulting ciphertext/plaintext will have the same alpha characters. The solution will still encrypt/decrypt the message, however, an error message will notify the user of the situation. Note: When encrypting, spaces and symbols will still be removed, and full stops replaced with 'X'.



THIS.IS A$MESSAGE!!!

THISXISAMESSAGE

26

USING A CIPHER KEY DIVISIBLE BY 26 RESULTS IN NO CHANGE!
ALPHA CHARACTERS REMAIN UNCHANGED

Figure 29: Encryption/decryption with a cipher key divisible by 26.

Appendix

A: Project files – GitHub repository

https://github.com/LukeWait/gui-app-switcher-midtownapp
Includes:

- Finalized Python script with internal documentation
- LICENSE & README
- requirements.txt (for installing dependencies)
- ‘fonts’ and ‘images’ asset folders and associated files