# USER MANUAL FOR THE CUSTOME **SANDBOX** EXECUTION TOOL

## TRUSTWORTHY COMPUTING

SAGHAR FADAEI
STUDENT NO: 202329649

# TABLE OF CONTENTS

INTRODUCTION ..... 3

SYSTEM REQUIREMENT.....4

INSTALLATION .....5

USAGE INSTRUCTION .....6

HIGHLIGHTS .....8

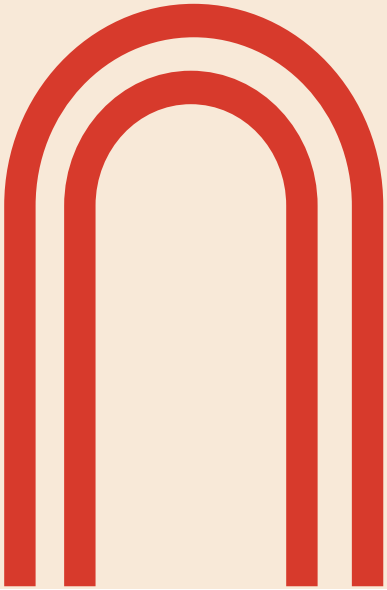
CONCLUSION.....9

# INTRODUCTION

The Custom Sandbox Execution Tool is an innovative utility designed to provide faculty members in computer science departments with a secure and isolated environment for running and testing student-submitted code. This tool utilizes the robust features of Windows Sandbox, a lightweight virtual machine environment integrated within Windows 10 and Windows 11 Pro and Enterprise editions, to execute code without risking the integrity of the host system. The tool is versatile, supporting a range of command-line arguments for custom execution settings, and includes a user-friendly graphical interface to streamline the setup process. With the capability to disable network access, read from specified input files, and write to designated output files, the tool offers a comprehensive solution for evaluating student code in a controlled and safe manner. Whether you are processing hundreds of code submissions or exploring untested applications, the Custom Sandbox Execution Tool ensures a seamless, secure, and efficient user experience.

# SYSTEM REQUIREMENTS

- Windows 10 Pro or Enterprise, or Windows 11
- Virtualization enabled in BIOS
- At least 4GB of RAM (8GB recommended)
- At least 1 GB of free disk space (SSD recommended)



# INSTALLATION

Enable Windows Sandbox:

- Navigate to 'Turn Windows features on or off' in Control Panel.
- Check 'Windows Sandbox', click OK, and restart your computer.

Prepare Execution Scripts:

- Place the provided PowerShell script (`GenerateSandboxConfig.ps1`) in a designated project directory.

.

# USAGE INSTRUCTIONS

- **Step 1: Compile C# Code**
  - Use the provided `Hello.cs` or any other program to compile a .NET executable.
  - Use `csc Hello.cs` or Visual Studio to compile the executable.

- **Step 2: Running the tool**
  - **Starting the GUI:**
    Right-click on GenerateSandboxConfig.ps1 and choose "Run with powershell" and launch the tool's graphical interface. If prompted, allow PowerShell to run the script.

  - **Configuring Execution Parameters:**
    *Executable Path*: Use the "Browse" button to select the .exe file you wish to run in the sandbox.
    *Output File Name*: Optionally, you can specify a name for the output file to capture the execution results of the program. Please note that if an output file is specified, the designated program will not automatically run in a Command Prompt (cmd) window within the sandbox. The program's output will be redirected to the specified file instead.
    *Custom Command:* If needed, enter a custom command for execution. For example you can write this command to get input text and print output a custom text in custom output file :

    ```
    "C:\Users\WDAGUtilityAccount\Desktop\output\Hello.exe" | Out-File -FilePath "C:\Users\WDAGUtilityAccount\Desktop\output\GUI-result.txt"; echo `customText` | Out-File -FilePath "C:\Users\WDAGUtilityAccount\Desktop\output\GUI-result.txt" -Append
    ```

*Disable Network:* Check this option to block network access in the sandbox environment.
*Read-Only:* Enable this to make the sandbox filesystem read-only.

- **Executing the Program:** Click "Run" to start the execution. The tool will generate a .wsb file based on your settings and launch Windows Sandbox accordingly.

- **Step 3: Viewing Results**
  - **Standard Output:** By default, the output of the program is displayed in the sandbox's console window. If no custom command or output file name is specified, your program will automatically run, and you can view its output directly in this window. Look for this output in the `Desktop > Output` directory within the sandbox.
  - **Output file:** If an output file name is specified, the program's output is redirected to this named file within the sandbox environment. This file will be accessible in the mapped directory on your host system after you close the sandbox. Note that specifying an output file will prevent the program from automatically running in a Command Prompt window within the sandbox.
  - **Automatic Execution**: Your program will automatically be executed upon launching the sandbox if no custom command is provided and no output file name is specified. This allows for an immediate and hassle-free observation of the program's behavior.
  - **File Persistence:** Any file created or modified within the sandbox, including the specified output file, will be mirrored in the corresponding mapped directory on your host machine. This ensures that you have access to all relevant files and results even after the sandbox session is closed.

# HIGHLIGHTS

### ISOLATED EXECUTION ENVIRONMENT

Our tool leverages the robust capabilities of Windows Sandbox, ensuring that all code executions occur within a completely isolated environment. This approach safeguards the host system from potential threats and unwanted changes, providing a secure testing ground for student-submitted code.

### GUI FOR EASE OF USE

We understand the importance of accessibility, especially for educators with diverse technical backgrounds. Our intuitive Graphical User Interface (GUI) greatly simplifies the process of configuring and managing the sandbox environment, making it straightforward to use without compromising on functionality.

.

### CUSTOM CONFIGURATION

Our tool offers advanced customization options, allowing users to tailor sandbox configurations to meet the unique requirements of each code submission. From network isolation to adjustable file access permissions, educators have the flexibility to create the most appropriate testing conditions, ensuring a thorough and relevant evaluation of student work.

# CONCLUSION

The Custom Sandbox Execution Tool offers a secure and versatile environment for testing and executing student-submitted code. Its user-friendly GUI and advanced features like custom command execution and output redirection make it an invaluable asset for educators and developers alike.