□ Protected To-Do List GUI Application

This project provides a simple, interactive **To-Do List application** built with Python's Tkinter library. It demonstrates how to create a desktop GUI and protect the source code using **Pyarmor** for obfuscation and **PyInstaller** for bundling it into a standalone executable.

□ Features

- □ **Add Tasks**: Easily add new to-do items.
- □ **View Tasks**: Displays all current tasks with their completion status.
- □ **Mark Complete**: Mark tasks as completed.
- □ **Delete Tasks**: Remove tasks from the list.
- □ **Persistent Storage**: Saves tasks to a `todos.json` file.
- □ **Basic Obfuscation**: Uses Pyarmor to make the source code harder to read.
- □ **Standalone Executable**: Built with PyInstaller so no Python install is needed for end-users.

□ Important Note on Source Code Access

> *If this project is hosted on a public platform like GitHub, **anyone can access the original, un-obfuscated Python source code** (`todo_list.py`).*
>
> *The protection steps in this README are intended for securing the **distributed executable**, not the public repository.*
>
> *To fully protect your code, distribute only the **PyInstaller-generated** `.exe`, not the raw `.py` files.*

□ Prerequisites

- **Python 3.x** – [Download Python]
- **Command Line Interface** – Terminal / Command Prompt

> □ *During Python installation, ensure **"Add Python to PATH"** is checked.*

□ Setup & Run (Unprotected Version)

1. Create Project Folder & Virtual Environment

```bash
mkdir project7
cd project7
python -m venv venv

# Activate the virtual environment
# On Windows:
.\venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate
```

2. Create the Application Script

Create a file named todo_list.py and paste your full Python code into it. □ Refer to the original code block above or insert your code directly here.

3. Run the Unprotected GUI App

```
python todo_list.py
```

□ Protection Steps

Apply protection with Pyarmor (obfuscation) and PyInstaller (executable bundling).

1. Install Protection Tools

```
pip install pyarmor pyinstaller
```

2. Obfuscate with Pyarmor

```
mkdir pyarmor_output
pyarmor gen --output pyarmor_output todo_list.py
This creates pyarmor_output/ with the obfuscated script and runtime files.
```

3. Create Executable with PyInstaller

Navigate back to the main project directory:

```
cd ..
Then build the executable:
pyinstaller --onefile --windowed --name ProtectedTodoListGUIApp --add-data "pyarmor_output;pyarmor_output" todo_list.py
```

□ Explanation of Flags

- onefile: Bundle everything into a single .exe.
- windowed: No terminal/console window will appear with the GUI.
- name: Sets the output filename.
- add-data: Includes obfuscated code and runtime in the bundle.
- Format: source_path;destination_path_inside_executable

□ Running the Protected Application

Navigate to the dist/ directory:

```
cd project7\dist
.\ProtectedTodoListGUIApp.exe  # On Windows
# ./ProtectedTodoListGUIApp    # On macOS/Linux
```

□ Project Structure

```
project7/
├── build/                    # PyInstaller temp files
├── dist/                     # Final executable
├── pyarmor_output/           # Obfuscated files
├── todo_list.py              # Original source (optional after packaging)
├── todos.json                # Created on runtime
└── venv/                     # Python virtual environment
```

Conclusion

You now have a fully functional and protected To-Do List application with:

- A Tkinter GUI
- Persistent task storage
- Basic source code protection via Pyarmor
- A standalone executable for easy sharing and distribution