Welcome to the Self Serving Kiosk Terminal Application

steps to install the application

Operating the Self-Serving Kiosk Terminal Application

1. **Open a Terminal Window:**
   - On your computer, open a terminal window. This can typically be done through your operating system's search or application launcher.
2. **Navigate to the Application Directory:**
   - Use the `cd` command to navigate to the directory where the `RUN_SSKiosk.sh` script is located. For example:

     ```
     cd ./GustavoJimenez_T1A3/src
     ```
3. **Run the Script:**
   - Execute the following command to run the self-serving kiosk application:

     ```
     ./RUN_SSKiosk.sh
     ```
4. **Script Execution:**
   - The script will perform the following actions:
     - **Check Python Installation:**
       - Verify if Python is installed on your system.
       - If Python is not installed, the script will exit with an error message prompting you to install Python.
     - **Check venv Installation:**
       - Confirm the availability of the `venv` module.
       - If `venv` is not installed, the script will exit with an error message providing guidance on installation.
     - **Create and Activate Virtual Environment:**
       - Create a virtual environment named `.kiosk` using `python3 -m venv .kiosk`.
       - Activate the virtual environment using `source .kiosk/bin/activate`.
     - **Install Dependencies:**
       - Install Python dependencies listed in `requirements.txt` using `pip3 install -r requirements.txt`.
     - **Run the Kiosk Application:**

            - Execute the self-serving kiosk application by running `python3 ordering_kiosk.py`.
        - **Deactivate Virtual Environment:**
            - Deactivate the virtual environment after the application completes running by using the `deactivate` command.

5. **Troubleshooting:**

    - If you encounter any issues during the script execution, carefully read the error messages for guidance on resolution.
    - Additionally, refer to the following resources for help:
        - Official virtualenv Documentation
        - Python Packaging User Guide

6. **Enjoy Your Kiosk Experience:**

    - Once the script has successfully executed, enjoy using the self-serving kiosk application for your needs.

Dependencies required by the application to operate

While the dependencies will be installed automatically by the .sh file, here is a list of the python libraries used in this application

```
about-time==4.2.1
alive-progress==3.1.5
grapheme==0.6.0
markdown-it-py==3.0.0
mdurl==0.1.2
Pygments==2.17.2
rich==13.7.0

iniconfig==2.0.0
packaging==23.2
pluggy==1.3.0
pytest==7.4.3
```

Any system/hardware requirements

Your provided system and hardware requirements for running a basic Python terminal application are quite reasonable. However, let's break it down further and add more specificity:

System Requirements:

Operating System:

- Windows 10, macOS, or Linux

Python Version:

- Python 3.2 and above

Python Package:

- `virtualenv` package installed

Hardware Requirements:

Processor:

- Intel Core i3 or AMD Ryzen 3

Memory (RAM):

- 4 GB RAM

Storage:

- 256 GB SSD

Display:

- Minimum resolution of 1366 x 768

Additional Considerations:

- **Disk Space:** Ensure that there is sufficient free disk space to accommodate the Python interpreter, virtual environments, and any additional dependencies or datasets your application may require.

- **Internet Connection:** While not explicitly mentioned, an internet connection may be necessary during the initial setup or when installing Python packages.

- **GPU:** For basic terminal applications, a dedicated GPU is not a strict requirement. However, if you are working with computationally intensive applications or large datasets that can benefit from GPU acceleration, consider having a compatible GPU.

- **Environmental Variables:** Make sure that the necessary environmental variables are set up correctly, especially those related to Python and virtual environments.

- **Terminal Emulator:** Ensure that the operating system has a functional terminal emulator. On Windows, this could be Command Prompt, PowerShell, or a third-party terminal emulator like Git Bash. On macOS and Linux, the default terminal applications should be sufficient.

Recommendations:

- Regularly update the operating system, Python, and installed packages to benefit from the latest features, improvements, and security patches.

- When working with Python, consider using a version manager like `pyenv` (on macOS/Linux) or `pyenv-win` (on Windows) to manage different Python versions.

- If you plan to deploy your application on various platforms, test it on each to ensure compatibility.

These specifications should be suitable for running this terminal application.

How to use any command line arguments made for the application

No need to fuss with command line arguments; this application elegantly guides the client through the registration and login process, ensuring a personalised journey, from step one using the kiosk. As the users register, their names become the key to unlocking exclusive benefits, seamlessly integrated into the menu selection and purchase experience. Use with ease of ordering and the pleasure of personalised recommendations—all effortlessly accessible with a touch. Whether you're a returning client, a first-time diner or a once in a lifetime visitor, the SS_kiosk brings the extraordinary fusion of technology and hospitality. Welcome to a dining adventure where your personal details and purchase history takes center stage, unlocking a world of culinary offering tailored just for you!