

Installation Guide

This guide will walk you through setting up the application using Python 3.13.4. It includes instructions to install Python, create a virtual environment, and commands to launch the project.

Step 0: Open Powershell

- Press **Windows key + R**, type **powershell**, and press **Enter**. Now, Powershell will be opened.

Step 1: Verify Python Installation

Before proceeding, ensure that Python **3.13.4** is installed on your system.

To check your Python version, open your terminal or command prompt and execute the following command:

```
py -3.13 -V
```

If Python 3.13.4 is installed, the output should display Python 3.13.4. If you have a different version or Python is not found, please install the correct version before moving to the next step.

Step 2: Install Python 3.13.4 (If Needed)

This project requires Python 3.13.4 to run.

If Step 1 indicated you have a different version or Python is not found, please proceed with the installation below or else you can skip this step and step 3.

Download Python

Download the appropriate Python 3.13.4 installer for your operating system from the official Python website or from the links given below.

Your Computer	File to Download	Link
Windows 64-bit	python-3.13.4-amd64.exe	Click to Download
Windows 32-bit	python-3.13.4.exe	Click to Download

If you're unsure, most modern computers are 64-bit.

Install Python

Follow these steps to install Python:

1. Double-click the downloaded file to start installation.
2. **Important:** Check the box that says **"Add Python to PATH"**.
3. Click "Install Now" and wait for the setup to finish.

After installing python, it's always a good practice to close the current powershell terminal and open a new powershell terminal using **Win + r** shortcut method mentioned above.

Step 3: Verify Python Installation

After installing Python, it's essential to confirm that it's correctly recognized by your system and that the correct version (3.13.4) is accessible.

Check Python Version

Once PowerShell is open, type the following command and press **Enter**:

```
py -3.13 -V
```

Expected Output:

If Python 3.13.4 was installed correctly and added to your system's PATH, you should see output similar to this:

```
Python 3.13.4
```

Step 4: Create a Virtual Environment

It's highly recommended to create a virtual environment for this project. This isolates the project's dependencies from your system-wide Python installation, preventing conflicts and ensuring a clean development environment.

Navigate to the Project Directory

Use the `cd` (change directory) command to navigate to the folder where you have stored all the Planning Tool files. This is the root directory of the project.

Example:

If your `planningTool` folder is inside `C:\MyProjects`, you would type:

```
cd C:\MyProjects\censustract_app
```

Creating a venv

Now, let's create a venv, we will name this virtual environment as 'censustract_app'. Use:

```
py -3.13 -m venv censustract_app
```

This command uses your Python 3.13 installation to create a new virtual environment named `censustract_app` within your current project directory.

Step 5: Activate the Virtual Environment

After creating your virtual environment, you need to activate it. Activating the environment ensures that all Python packages you install will be contained within `censustract_app`, keeping your system's global Python installation clean.

Allow Script Execution (Only Once Per PowerShell Session)

Before activation, PowerShell may block scripts. Run this command to allow it:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process
```

Activate the Environment

Run the following command to activate the virtual environment:

```
.\censustract_app\Scripts\Activate.ps1
```

What You Should See

Your PowerShell prompt will change to show the environment is active:

```
(censustract_app) PS C:\Your\Project\>
```

This prefix (`censustract_app`) means that the virtual environment is now active.

Step 6: Install Required Packages

This step helps you **install all necessary Python libraries**. To keep everything clean and contained, we will store the required packages **inside the virtual environment** folder.

Make sure your virtual environment is **activated** before performing the steps below.

Install Packages using pip and requirements.txt

Now install the packages using the below command:

```
(censustract_app) PS C:\Your\Project\> pip install -r path\to\requirements.txt
```

Example:

```
(censustract_app) PS C:\Your\Project\> pip install -r  
censustract_app\requirements.txt
```

Step 7: Start the FASTAPI Server

Once packages are installed, Start the FASTAPI Server:

```
(censustract_app) PS C:\Your\Project\> python path\to\main.py
```

Example

```
(censustract_app) PS C:\Your\Project\> python backend\main.py
```

IMPORTANT: Keep this terminal running. Do not close it, as the backend server must remain active for the app to work.

Step 8: Run the Streamlit Application

As now the FASTAPI server is started. Now, **open a new PowerShell window(Step 0) and repeat the earlier steps:**

- Navigate to your project folder using `cd`.
- Activate the virtual environment with (Step 5)
- Then start the Streamlit app:

```
(censustract_app) PS C:\Your\Project\> streamlit run path\to\script_file.py
```

Example

```
(censustract_app) PS C:\Your\Project\> streamlit run ui\improved_ui.py
```

That's it! Your application should now launch in a browser window.

Note: To stop the application, after using it, press **Ctrl + c** in the powershell terminal.

If you get an error like `ModuleNotFoundError`, double-check:

- Your virtual environment is active.
- All packages were installed properly.