2024

New Reporting Tool Implementation Using Python Streamlit

I7DW: NEW REPORTING TOOL IMPLEMENTATION USING PYTHON STREAMLIT

ICM COMPUTER SYSTEMS LIMITED

Table of Contents

Python Streamlit Instalation Steps

ZIP folder

Begin by gathering necessary files in a ZIP format

STEP 1

Installing Python

Set up Python, the required programming language

STEP 2

Run the Setup Batch File

Execute the setup file to configure the environment

STEP 3

Config.toml and secrets.toml

Config the credentials for database and SSRS.

STEP 4

Access the Application

Open and run the Streamlit application

STEP 5

Automatic Background Execution

Ensure the application runs continuously in the background.

STEP 6

1. Overview

The Infeed700 application is an interactive platform developed by **ICM Computer Systems Limited** using Streamlit. It provides embedded dashboards and SSRS (SQL Server Reporting Services) reports, allowing users to access and visualize data efficiently in an on-premises environment. This application is designed to serve multiple clients, offering a user-friendly and responsive interface.

2. Installation Instructions

Follow these steps to set up the Infeed application on your local machine:

2.1 Choose an Installation Directory:

2.1.1 Decide where you want the application files to be installed.

This could be a specific directory, such as ***C:\Program Files\Infeed700*** or a custom folder like ***C:\MyProjects\Infeed700***.
If the directory doesn't exist, you may need to create it.

2.1.2 Move and Extract the ZIP Folder:

Copy or move the ZIP folder to your chosen directory.
Right-click on the ZIP folder and select "Extract All..." (or use an extraction tool if you have one).
Choose the extraction location (if prompted) and confirm.

2.2.3 Navigate to the Extracted Folder:

Once unzipped, navigate to the extracted folder where you'll find the necessary files to proceed with the installation or setup.

3. Installing Python:

Navigate to the directory where the **Infeed700Plus** files are located, find the folder named ***python_offline_installer***, and inside, locate the Python ***3.12.5*** offline installer (***python-3.12.5-amd64***). Double-click the installer and follow the steps to complete the installation.

If you need to install from the internet, follow these steps:

Visit the official Python website: https://www.python.org/downloads
Download Python ***version 3.12.5*** (make sure to choose the installer compatible with your operating system).
Run the installer and check the box that says **"Add Python to PATH"** during the installation.

4 Set up the Config.toml and Screts.toml file

Navigate to the directory where you can find all the files, then go to the ***.streamlit*** folder.

4.1 Open the secrets.toml file.

Open with notepad or notepad++.
Update the values for [site_info] and [secrets_config] with the relevant site information.

secrets.toml

```
[secrets_config]
secrets_name = "secrets_das"    # site TLA e.g secrets_das, secrets_kmg

[site_info]
site_name = 'DAS'                          # site TLA e.g DGL, KMG, DAS

[browser]
gatherUsageStats = false
```

4.2 Update the secrets_tla.toml

secrets_tla.toml

```
[mydb]
# Connection details for the SQL Server database using to Produce Python Reports/ Dashboards
dialect = "mssql"
driver = "ODBC Driver 17 for SQL Server"                # Specify the ODBC driver
username = "sa"                                         # Database username
```

```
password = "1984Icm000"                    # Database password
host = "192.168.5.182"                     # IP address of the SQL Server
port = "1433"                              # Default SQL Server port
database = "Infeed700"                     # Infeed700 database e.g Infeed700DAS
enecoms_database = "Enecoms"               # Enecoms database e.g EnecomsDAS
instance = "MSSQLSERVER"                   # Optional: Include the instance name if needed

[ssrs_config]
# Connection details for the SSRS server
ipAddress = "192.168.5.182"
port = "80"
database = "Infeed700"
database-enecoms= 'Enecoms'
ReportServerName = "Reportsmanager"
username = "DASMILL\\Reports"
password = "Reporting123"

[streamlit_server]
ip_address = "localhost"
port = "8501"
```

5.Run the Setup Batch File:

Navigate to the ***Infeed700*** directory, locate the ***setup.bat*** file, right-click it, and select Run as administrator. A CMD window will appear, and it will begin installing all necessary libraries in offline mode. Ensure that the folder named ***libs*** is in the same directory as setup.bat

6. Access the Application:

After running the setup, the **Streamlit** application should automatically launch in your web browser. If it doesn't, you can manually open your browser and go to **http://localhost:8501**

By following these steps, you will have the **Infeed700** application set up and running on your local machine. If you encounter any issues, please refer to the troubleshooting section of this documentation or seek assistance from your IT support team.

Documentation for support on **http://localhost:8502**

7. Setting Up of Streamlit Apps on Windows

This section explains how to set up your Streamlit applications to run in the background automatically when your PC starts or restarts using a batch script and the Windows Task Scheduler.

7.1 Step 1: Create a Batch Script to Run Streamlit Apps in the Background

7.1.1 Creating the .bat file:

Open a text editor (such as Notepad) and paste the following content

*Only follow this step if there is no **setup.bat** file in the directory*.

```
@echo off
start /b "" streamlit run main.py --server.port 8501
start /b "" streamlit run functions/documentation.py --server.port 8502
exit
```

7.1.2. Save the file.

Save this file as `run_streamlit_apps.bat` in the same directory where all the `.py` files are located**.**

7.2 Step 2: Set Up Task Scheduler to Run the Script at Startup

Once the `.bat` file is created, we will configure the Windows **Task Scheduler** to run it automatically at startup.

7.2.1. Opening the task Scheduler.

Press Win + R to open the Run dialog.
Type ***taskschd.msc*** and press Enter to open Task Scheduler.

7.3 Step 3: Create a New Task

1. In **Task Scheduler**, click **Create Task**.
2. In the **General** tab:
   - Name the task, e.g., ***Run Streamlit Apps***.
   - Select **Run whether user is logged on or not**.
   - Check ***Do not store password*** if you want the task to run without needing a password.

7.4 Step 4: Configure Triggers

1. In the ***Triggers*** tab, click ***New***.
2. In the ***Begin the task*** dropdown, select ***at startup***.
3. Click ***OK*** to save the trigger.

7.5 Step 5: Configure Actions

1. In the ***Actions*** tab, click ***New***.
2. Select ***Start a program*** from the ***Action*** dropdown.
3. In ***Program/script***, click ***Browse*** and select the ***run_streamlit_apps.bat*** file.
4. In ***Start in (optional),*** enter the full directory path where the `.bat` and `.py` files are located. For example:

   - Program/script: `C:\path\to\your\directory\run_streamlit_apps.bat`
   - Start in: `C:\path\to\your\directory\`

5. Click ***OK*** to save the action.

7.6 Step 6: Test the Task

1. In ***Task Scheduler***, right-click on the task and select ***Run*** to test whether the Streamlit apps start correctly.
2. Restart your computer to confirm that the task runs automatically on startup.

By following these steps, your Streamlit apps will start running in the background automatically each time the computer starts or restarts.

Notes:

- If the `.bat` file and the Python scripts are in the **same directory**, you don't need to specify the full paths to the `.py` files in the script.
- Make sure to enter the correct path in the **Start in (optional)** field when creating the task in **Task Scheduler** to ensure it runs in the correct directory.

8. Configure Inbound Rules in Windows Firewall

Open Windows Firewall

Press Win + S to open the search bar.
Type "Windows Defender Firewall" and select "Windows Defender Firewall with Advanced Security" from the results.

Access Inbound Rules

In the left-hand panel, click "Inbound Rules".
Create a New Rule

In the right-hand panel, select "New Rule..." to open the New Inbound Rule Wizard.
Select Rule Type

In the wizard, select "Port" and click "Next".
Specify the Port
Choose "TCP" (commonly used for Streamlit).
Under "Specific local ports", enter the port number your Streamlit app uses (e.g., 8501).
Click "Next".
Allow the Connection

Select "Allow the connection" and click "Next".
Apply the Rule to Network Types

Choose the network types where the rule should apply:
Domain: Applies to computers within a domain.
Private: Applies to trusted private networks (e.g., home or office).
Public: Applies to untrusted public networks (e.g., coffee shop Wi-Fi).
Select as appropriate for your use case and click "Next".
Name the Rule

Give your rule a descriptive name (e.g., "Streamlit App Port 8501").
Optionally, add a description for future reference.
Click "Finish".

9. Required Software and Libraries

To run the batch file you provided, the following software and libraries are required:

9.1. Python:

- Version: Python should be installed on the machine. The batch file checks if Python is installed using `python --version`. You can download Python from https://www.python.org/
- Ensure Python is in your system's PATH**: So it can be accessed from the command line. If it's not, the batch file will fail with the error message "Python is not installed or not found in the PATH."

9.2 pip (Python Package Installer):

- Installed with Python: Pip comes installed by default with Python. It is used to install the `.whl` files (Python wheel files) found in the `libs` folder. The command `pip install --no-index --find-links="%LIBS_DIR%" "%%f"` is used to install these dependencies without fetching from the internet.

9.3. Pre-downloaded Python Dependencies (.whl files):

- You need to have pre-downloaded Python wheel (`.whl`) files for all required packages stored in the `libs` directory. These `.whl` files must be compatible with your Python version and system architecture. The batch file installs each `.whl` file found in the `libs` folder using `pip`.

9. 4. Streamlit:

- Python package: The batch file assumes that your project uses Streamlit, a Python-based web application framework. One of the `.whl` files in the `libs` folder should be `streamlit`. You will also need any other dependencies required by your Streamlit application (e.g., `pandas`, `numpy`, etc.).
- Running the Streamlit application: The command `streamlit run main.py` is used to start the Streamlit app, so Streamlit must be installed in the environment.

10. Python Naming Conventions and Best Practices

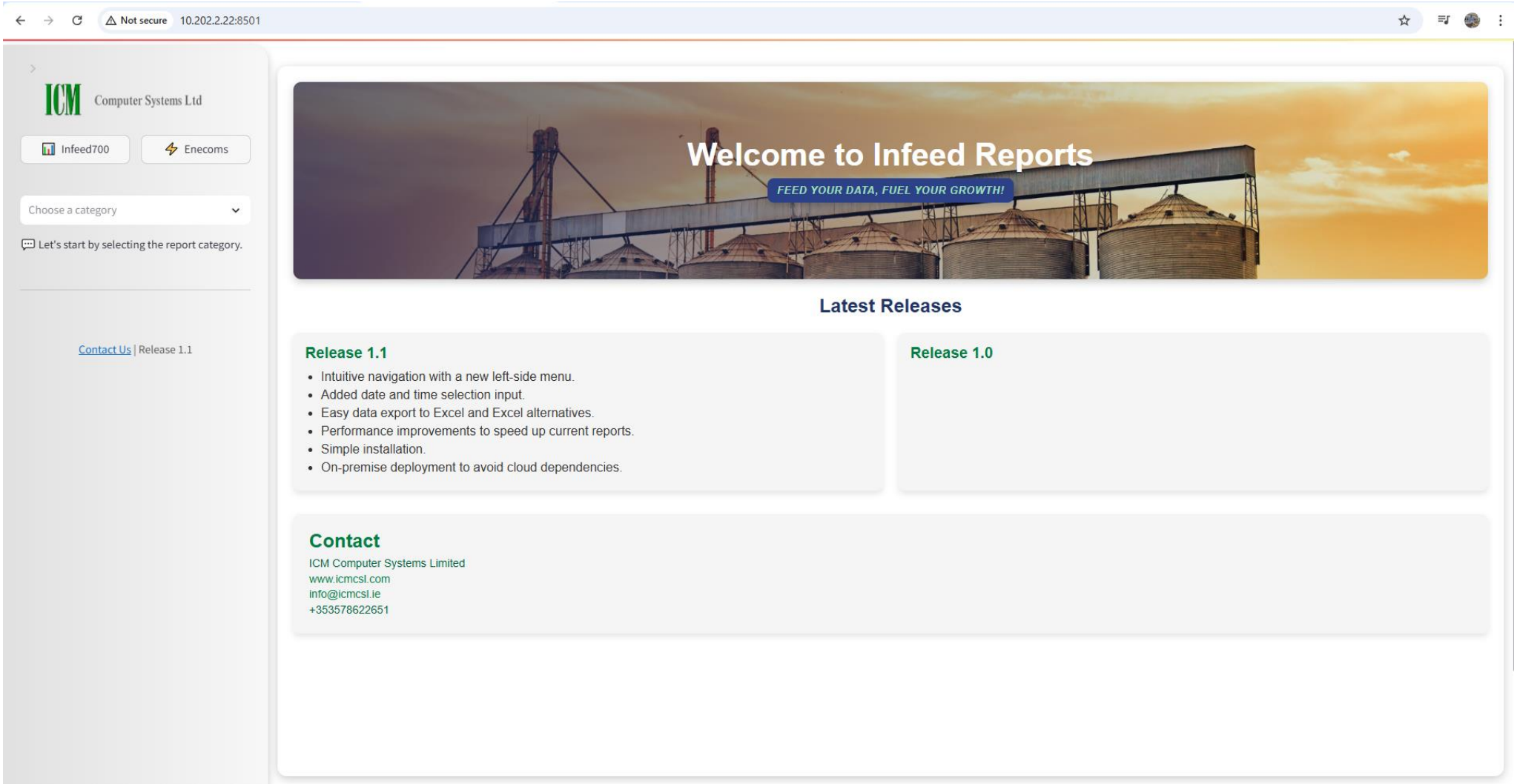
This document outlines Python naming conventions and best practices for naming files, functions, variables, classes, and constants in Python projects. Following these conventions ensures consistency, readability, and maintainability across your codebase. The guidelines are based on **PEP 8**, the official style guide for Python code.

10.1 Summary of Naming Conventions

Entity	Naming Convention	Example
Files	snake_case	database_connection.py
Functions	snake_case()	fetch_data()
Variables	snake_case	total_weight
Classes	PascalCase	UserProfile
Constants	UPPERCASE_WITH_UNDERSCORES	MAX_RETRIES
Private Methods	underscore_prefix()	connect_to_database()

11. Streamlit Application

http://localhost:8501



12. Documentation

http://localhost:8502

username:admin
password:1984Icm000