A green and black logo

Description automatically generated

New Reporting Tool Implementation Using Python Streamlit

ICM Computer Systems Limited – VER 20240918

I7DW: new Reporting Tool Implementation Using Python Streamlit

ICM COMPUTER SYSTEMS LIMITED

2024

Table of Contents

[Project Phases: 2](#_Toc177563785)

[1. Initial Demo Phase (35-45 hours) 2](#_Toc177563786)

[2. Review and Feedback Phase (20-30 hours) 2](#_Toc177563787)

[3. Template Creation and Testing (35-45 hours) 2](#_Toc177563788)

[4. Versioning control 3](#_Toc177563789)

[5. Testing and Quality Assurance (QA) 3](#_Toc177563790)

[6. Monitoring and Logging 3](#_Toc177563791)

[Team Responsibilities: 3](#_Toc177563792)

[17 Planner 0](#_Toc177563793)

**Project Plan**: Transition from SSRS to Python Streamlit Reporting System

**Project Overview:**

The Infeed700 application is an interactive platform developed by ICMC Solutions using Streamlit. It provides 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.

**Features**

Interactive Dashboards: Visualize real-time data with interactive charts and tables.

Embedded SSRS Reports: Access SSRS reports directly within the application.

On-Premises Deployment: Ideal for clients who prefer to keep their data within their own infrastructure.

Intuitive Interface: With a simple navigation system, users can easily access different sections of the application.

**Project Structure**

The directory structure of the application is organized as follows:

Infeed700/

│

├── images/ # Directory for images and icons

│ ├── home.svg # SVG icon for the sidebar

│ └── (other images and icons)

│

├── LeftMenu/ # Directory for left menu-related files

│ ├── leftMenu.py # Contains the logic for the left menu

│ ├── expanderStyle.css # Custom styles for the left menu

│

├── styles.py # Custom styles for the menu

│

├── main.py # Main entry point for the Streamlit application

│

├── config.toml # Configuration settings for the Streamlit application

│

├── requirements.txt # List of dependencies for the application

│

├── setup.bat # Batch file for setting up the environment

│

└── .streamlit/

└── secrets.toml # Secrets for database connection

**Code Snippets**

**1. main.py**

import streamlit as st

from leftMenu.leftMenu import LeftMenu

import requests

from requests\_ntlm import HttpNtlmAuth

# Set the page layout for the Streamlit app

st.set\_page\_config(layout="wide")

# Display the left menu for navigation

LeftMenu()

# Initialize session state for the selected report if not already set

if 'selected\_report' not in st.session\_state:

st.session\_state['selected\_report'] = "Intake"

reportRDLname = st.session\_state['selected\_report']

# Database credentials and configuration settings

ipAddress = "10.202.2.22"

port = "80"

database = "Infeed700"

ReportServerName = "ReportServer"

username = "icm\\ndasilva"

password = "1984Icm022\*"

# Construct the URL for the SSRS report

ssrs\_url = f"http://{ipAddress}:{port}/{ReportServerName}/Pages/ReportViewer.aspx?%2f{database}%2f{reportRDLname}&rs:Command=Render&MinDate=2024-08-01"

# Make the request to the SSRS report using NTLM authentication

try:

response = requests.get(ssrs\_url, auth=HttpNtlmAuth(username, password), timeout=10)

# Check if the request was successful

if response.status\_code == 200:

report\_url = f"{ssrs\_url}&rs:Embed=true&rc:Parameters=Collapsed"

# Create an iframe to display the report

iframe\_code = f"""

<iframe width="100%" height="100%" style="min-height: 150vh;" src="{report\_url}" frameborder="0" allowfullscreen></iframe>

"""

st.components.v1.html(iframe\_code, height=900, scrolling=False)

else:

st.error(f"Error accessing the report: {response.status\_code}")

# Handle specific connection errors and provide user guidance

except requests.exceptions.ConnectTimeout:

error\_message = """

\*\*Connection Timeout Error\*\*

Possible reasons for this issue:

1. Verify that the provided IP address `10.202.2.22` is correct and reachable.

2. Ensure that the port `80` is open and accessible on the target server.

3. Double-check the database name `Infeed700` in your report URL.

4. Go to the `.secrets.toml` file and verify that the credentials (username and password) are correctly configured.

Please resolve these potential issues and try again.

"""

st.error(error\_message)

# Handle other request-related errors

except requests.exceptions.RequestException as e:

st.error(f"Error accessing the report: {e}")

2. leftMenu/leftMenu.py

import streamlit as st

from streamlit\_option\_menu import option\_menu

from styles import menu\_styles # Import custom styles

from utilities import load\_svg # Import function to load SVGs

import os # For file path manipulation

# Define headers and reports for the menu

headers = {1: "Intake", 2: "Blending", 3: "Press"}

reports = {

1: [["Intakes", "Intake"], ["Intake Tips", "TipBreakdown"]],

2: [["Blending / Batching", "Batch"], ["Blending / Run", "BatchByRunNumber"]]

}

def LeftMenu():

def local\_css(file\_name):

with open(file\_name) as f:

st.markdown(f'<style>{f.read()}</style>', unsafe\_allow\_html=True)

local\_css("leftMenu/expanderStyle.css")

svg\_file\_path = os.path.join("images", "home.svg")

with st.sidebar:

with st.expander(label='', expanded=True):

svg\_icon = load\_svg(svg\_file\_path)

selectedMenu = option\_menu(

menu\_title="Infeed700",

menu\_icon="bar-chart-fill",

options=["Dashboards", "SSRS Reports"],

icons=["pie-chart-fill", "grid-3x3-gap-fill"],

default\_index=0,

styles=menu\_styles

)

if selectedMenu == "SSRS Reports":

for headerskey, headerName in headers.items():

if headerskey in reports:

with st.expander(headerName, expanded=False):

report\_option = option\_menu(

menu\_title=None,

menu\_icon="reception-4",

options=[report[0] for report in reports[headerskey]],

icons=["table"] \* len(reports[headerskey]),

default\_index=0,

key=headerName,

styles=menu\_styles

)

for report in reports[headerskey]:

if report[0] == report\_option:

st.session\_state['selected\_report'] = report[1]

3. leftMenu/expanderStyle.css

/\* Styling for the main container of the sidebar menu \*/

.sidebar-menu {

display: flex;

flex-direction: column;

padding: 0px!important;

background-color: #ffffff;

border: 0px solid #e0e0e0;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);

width: 300px;

max-width: 100%;

margin: 0px auto;

transition: all 0.3s ease-in-out;

opacity: 0.95;

backdrop-filter: blur(5px);

}

4. .streamlit/secrets.toml

[mydb] dialect = "mssql" driver = "ODBC Driver 17 for SQL Server" username = "sa" password = "1984Icm000" host = "127.0.0.1" port = "1433" database = "infeed700ECV" instance = "MSSQLSERVER"

5. requirements.txt

streamlit>=1.38.0

pandas>=2.1.0

numpy>=1.25.0

altair>=5.0.0

openpyxl>=3.1.0

sqlalchemy>=2.0.0

pyodbc>=4.0.0

matplotlib>=3.9.2

plotly>=5.24.1

requests==2.32.3 # Corrigido para usar '=='

requests\_ntlm>=1.3.0

streamlit-option-menu>=0.3.13

**Dependency Support**

The following libraries are required for the application:

* **Streamlit**: Streamlit Documentation
* **Pandas**: Pandas Documentation
* **NumPy**: [NumPy Documentation](https://numpy.org/)
* **SQLAlchemy**: SQLAlchemy Documentation
* **PyODBC**: [PyODBC Documentation](https://github.com/mkleehammer/pyodbc/wiki)
* **Matplotlib**: [Matplotlib Documentation](https://matplotlib.org/)
* **Plotly**: [Plotly Documentation](https://plotly.com/)
* **Requests**: Requests Documentation
* **Altair**: Altair Documentation
* **Openpyxl**: [Openpyxl Documentation](https://openpyxl.readthedocs.io/en/stable/)

**Installation Instructions**

1. **Prerequisites**: Ensure that your system has **Windows** operating system and that you have administrative access to install software.
2. **Download the Installer**: Place the Python installer in the root directory alongside the application files.
3. **Dependency Installation**:
   * Place the requirements.txt file in the root directory.
   * You can either install dependencies globally or create a virtual environment.
   * To install dependencies globally, run the following command:

bash

Copy code

pip install -r requirements.txt

* + Alternatively, you can use the setup.bat file, which will automate the installation process.

1. **Database Connection**: Configure the database connection in .streamlit/secrets.toml using your database credentials.
2. **Run the Application**:
   * Navigate to the root directory in your command prompt.
   * Execute the following command:

bash

Copy code

streamlit run main.py

**Contact Information**

For support and inquiries, please reach out to **ICMC Solutions** at [ICMC Solutions](https://icmcsl.com/).

# 17 Planner