A green and black logo

Description automatically generated

New Reporting Tool Implementation Using Python Streamlit

ICM Computer Systems Limited – VER 20241118

I7DW: new Reporting Tool Implementation Using Python Streamlit

ICM COMPUTER SYSTEMS LIMITED

2024

Table of Contents

[1. Push the Git Code to AzureDevops 1](#_Toc182824987)

[Step 1: Generate a New GitHub Personal Access Token (PAT) 1](#_Toc182824988)

[Step 3: Set Up Your New Project and Repository in Azure DevOps 1](#_Toc182824989)

[Step 4: Add Azure DevOps as a Remote in Your Local Git Repository 1](#_Toc182824990)

[Step 5: Push Your Code to Azure DevOps 1](#_Toc182824991)

[Step 6: Verify in Azure DevOps 1](#_Toc182824992)

[11. Streamlit Application 0](#_Toc182824993)

[http://localhost:8501 0](#_Toc182824994)

A screenshot of a computer

Description automatically generated

# 1. Push the Git Code to AzureDevops

## Step 1: Generate a New GitHub Personal Access Token (PAT)

Go to Settings > Developer settings > Personal access tokens in GitHub.

Click Generate new token.

Name the token (e.g., "Azure DevOps Migration").

Select the repo scope.

Click Generate token and copy it for later use.

Step 2: Clone Your GitHub Repository with the New Token

Open a terminal or Visual Studio Code.

Clone your private GitHub repository using your GitHub username and new PAT in the URL:

git clone https://<your-username>:<your-PAT>@github.com/nicolasSilvaBR/Infeed700Plus.git

Replace <your-username> with your GitHub username and <your-PAT> with the token you just created.

      PAT = Token

## Step 3: Set Up Your New Project and Repository in Azure DevOps

Log in to Azure DevOps with your company account.

Create a New Project in Azure DevOps:

Click New Project and name it.

Set the visibility and select Git for version control.

Create a Repository:

In your new project, go to Repos > Files.

If a repository isn’t automatically created, click New Repository.

## Step 4: Add Azure DevOps as a Remote in Your Local Git Repository

In the terminal, navigate to your cloned GitHub repository folder:

cd Infeed700Plus

Add the Azure DevOps repository as a second remote:

git remote add azure <http://icmdevops1:8080/tfs/DefaultCollection/Infeed700%20Python%20%20SSRS/_git/Infeed700%20Python%20%20SSRS>

## Step 5: Push Your Code to Azure DevOps

Push all branches from your local repository to the new Azure DevOps repository:

git push azure --all

git push azure –tags

## Step 6: Verify in Azure DevOps

Go to Repos > Files in your Azure DevOps project to confirm that all code and branches were successfully uploaded.

# 11. Streamlit Application

## <http://localhost:8501>