Experimental LabStore

Description

Overview

The repository is a Streamlit-based application named “Experimental LabStore.” The project aims to provide tools for laboratory management and data analysis. The README describes the idea and basic usage steps. After installing the dependencies, you start the Streamlit server and open the interface in a browser.

Core modules

config.py – Defines the page_config helper. This sets global page options and draws a sidebar with links to all the pages in the app.

lab.py – The main entry point. The PlayStoreApp class invokes page_config and displays links to each subpage, arranged in columns for quick navigation.

Pages (pages/ directory) – Each submodule is a standalone Streamlit script with its own UI logic:

Calculadora_FTC.py – A frequency/cycle/time calculator with a form-based interface.

Data_Explorer.py – Lets you upload a CSV, plot columns, view stats and correlations, and save filtered results.

Rotations.py – Generates rotated versions of measurement columns in Excel files according to user parameters.

Calibraciones.py – A large calibration tool that generates charts, computes errors, and can build detailed PDF reports (via report_generator.py).

Cálculo_Tracción_KC-390.py – Calculates traction force from diameter and hardness values, displaying results and generating PDFs.

Other pages include a time calculator and demo/test pages. There is also a wip/ folder with work-in-progress modules.

Utility modules – utils.py contains a couple of helpers for st.session_state management. report_generator.py houses the heavy logic for building calibration reports in PDF form using ReportLab.

Data and resources – The data/ folder provides sample spreadsheets and CSV files. The resources/ folder holds assets like RK_CR.csv (the Rockwell hardness table) and images used in the UI.

Dependencies – requirements.txt lists the packages required, including Streamlit, pandas, matplotlib, seaborn, and ReportLab.

Pointers for further exploration

Streamlit architecture – Understanding how Streamlit manages state and forms will help you modify the UI. Check the utility functions in utils.py.

ReportLab – If you need custom PDF reports, inspect report_generator.py for formatting and data insertion logic.

Data handling – Several pages read Excel/CSV files using pandas. Explore the data directory to test the explorers.

Add new pages – config.py centralizes page links; extending the application involves creating new modules under pages/ and adding them in page_config.

Work in progress – The pages/wip/ directory contains experimental modules like Graficador.py and Instrumentos.py. These illustrate additional features (e.g., database queries or multi-file plotting) and could be expanded.

Overall, the codebase is a modular Streamlit project with separate interactive tools. Familiarity with Streamlit workflows, pandas data manipulation, and the ReportLab library will help you extend or adapt these applications.

Prerequisites

- Python 3.10.11
- pip package manager
- A modern web browser

Installation

To install Experimental LabStore, follow these steps:

1. Clone the repository: `git clone https://github.com/your_username/Experimental-LabStore.git`
2. Navigate to the project directory: `cd Experimental-LabStore`
3. Install necessary packages: `pip install -r requirements.txt`

Usage

1. Start the server: `python -m streamlit run .\lab.py`
2. Open your web browser and navigate to `http://localhost:5000`
3. You can now interact with the system, add inventory items, track usage, and automate restocking routines. For more details, please refer to the user manual included in the project documentation.

Para solucionar error "Error on ghost text request: FetchError: The pending stream has been canceled (caused by: self signed certificate in certificate chain)" en Github Copilot:

$env:NODE_TLS_REJECT_UNAUTHORIZED="0"