Project Documentation

Project Overview

This project encompasses four primary components:

1. HTML URL Extraction from GitHub API

2. Calculator UI

3. Stop Watch UI

4. School Database Management System (DBMS)

---

1. HTML URL Extraction from GitHub API

Description

This component retrieves the HTML URLs for all issues present in the Pandas GitHub repository.

Functionality

- Makes a GET request to the specified API.

- Extracts and prints the `html_url` of each issue if the request is successful.

---

2. Calculator UI

Description

This is a simple calculator application built using Tkinter, a standard GUI toolkit in Python.

Functionality

- Provides a graphical user interface for performing basic arithmetic operations.
- Supports addition, subtraction, multiplication, and division.
- Includes a clear function and evaluates expressions.

---

3. Stop Watch UI

Description

This is a stopwatch application built using Tkinter, providing basic stopwatch functionality.

Functionality

- Allows users to start, stop, and reset a stopwatch.
- Displays elapsed time in a user-friendly format.

---

4. School Database Management System (DBMS)

Description

This component manages a school database with functionalities to add, update, view, and delete student records.

Database Schema

- Student Table: Stores student details such as name, age, sex, class, fees, rank, and marks.

- Teacher Table: Contains teacher information including name, age, sex, salary, and class.
- Principal Table: Includes details of the principal.
- Admin Table: Manages admin login credentials.

Functionality

- Admin Login: Validates admin credentials.
- CRUD Operations: Allows the admin to create, read, update, and delete student records.
- Navigation: Offers a menu for navigating between student, teacher, and principal tables.

User Interface

- Prompts admin for login credentials.
- Displays options for CRUD operations.

Conclusion

This project effectively integrates various functionalities, from API interaction and basic calculations to a comprehensive database management system. Each component is modular, making it easy to maintain and expand.

Installation and Usage

1. Install the required packages: `requests`, `tkinter`, and `mysql-connector`.
2. Create the MySQL database and tables as outlined above.
3. Run each Python script to use the respective functionality.

Future Improvements

- Implement error handling for database operations.
- Enhance UI with better design and additional features.
- Include user authentication for other roles (teacher, principal).