

DBexecuter

Here is the documentation for the provided Streamlit code:

Overview

This Python script uses Streamlit to create a web application for database backup and upgrade operations. Users can select between "Backup" and "Upgrade" tabs to perform corresponding actions. The script uses `subprocess` to execute shell commands for backup and SQL file execution.

Dependencies

- `streamlit` : For creating the web interface.
- `os` : For file and directory operations.
- `subprocess` : For executing shell commands.
- `datetime` : For timestamping backup files.

Functionality

The application provides two main functionalities:

1. **Backup**: Allows users to create a backup of a MySQL database.
2. **Upgrade**: Allows users to execute SQL files to upgrade a MySQL database.

User Inputs

For Backup:

- **Username**: MySQL username for backup.
- **Password**: MySQL password for backup.
- **Database**: Name of the database to back up.
- **Backup Folder Path**: Directory where the backup file will be saved.

For Upgrade:

- **Username:** MySQL username for executing SQL files.
- **Password:** MySQL password for executing SQL files.
- **Database:** Name of the database to upgrade.
- **Folder Path:** Directory containing `.sql` files to be executed.

Global Variables

- **option:** Determines which tab ("Backup" or "Upgrade") is currently selected.

Functions

`execute_backup()`

- **Purpose:** Creates a backup of the specified database using `mysqldump`.
- **Command:** `mysqldump -u {backup_username} -p{backup_password} {backup_database} > "{backup_file_path}"`
- **Error Handling:** Logs error messages if the backup fails.

`execute_sql_file(file_path)`

- **Purpose:** Executes a SQL file on the specified database using the `mysql` command.
- **Command:** `mysql -u {upgrade_username} -p{upgrade_password} {upgrade_database} < "{file_path}"`
- **Error Handling:** Logs error messages if the SQL execution fails.

Workflow

1. Backup Tab:

- Users input required details and click "Backup".
- The script creates a timestamped backup file in the specified folder.
- If successful, it prints a success message; otherwise, it logs errors.

2. Upgrade Tab:

- Users input required details and click "Upgrade".

- The script executes all `.sql` files in the specified directory.
- If SQL files are executed successfully, it prints a success message; otherwise, it logs errors.

Notes

- **Error Handling:** Errors are printed to the console but are not displayed in the Streamlit UI.
 - **Directory Creation:** Ensures that the backup folder exists or creates it if necessary.
-