Automatic Database Backup & Restore

Objective

This task involves automating the backup process for a specific database at regular intervals and providing an easy way to restore the database when needed. The backup files should be stored in a designated folder with timestamps for easy tracking.

Recommended Technology

For smooth implementation, we recommend the following technologies:

Database Choices

- **MySQL/MariaDB** *(Recommended for simplicity with `mysqldump`)*
- **PostgreSQL** *(Alternative with `pg_dump`)*
- **SQLite** *(For lightweight databases using file copying)*

Programming Language Choices

- **Python** *(Recommended for scripting and automation with `subprocess`)*
- **Bash** *(Ideal for Linux users with `cron` and `mysqldump` commands)*
- **PowerShell** *(For Windows users with `Task Scheduler` and `pg_dump`)*

Automation Tools

- **Linux/macOS**: `cron` for scheduling backups
- **Windows**: Task Scheduler

Task Requirements

- **Automatically backup a database** at specified intervals (e.g., daily at midnight).
- **Store backups** in a structured format with timestamps (e.g., `backup_2025-02-25.sql`).

- **Allow easy restoration** of the backup file when needed.
- **Compress backups** (optional) to save space.
- **Log backup operations** to track success/failure.
- **Provide user documentation** on how to configure and use the script.
- **GitHub Submission**: Push your final script to a GitHub repository.

Getting Started

1 Configure Database Access

- Ensure the script can access the database with the required credentials.
- Test `mysqldump` (MySQL), `pg_dump` (PostgreSQL), or appropriate backup command.

2 Implement Backup Logic

- Generate a backup file with a timestamp.
- Store it in a designated backup folder.
- Optionally compress the backup to save space.

3 Implement Restore Functionality

- Allow restoring from a selected backup file.
- Provide clear steps to restore the database.

4 Automate Execution

- **Linux/macOS:** Schedule using `cron`.
- **Windows:** Schedule using Task Scheduler.

5 Document and Submit

- Provide a README file with setup steps and restore instructions.
- Submit the script to a GitHub repository with sample backup logs.

Submission Instructions

1. **Push your code** to a GitHub repository.
2. **Include a README** with setup steps and configuration instructions.
3. **Attach sample logs** showing backup success/failure.

GitHub Repository

GitHub Repo Link (Replace with your repo)

Happy coding!