Why Hardcoded Passwords Are a Problem & Setting Up python-dotenv

AiCE Software Engineering Foundations — MySQL Programming

Right now, app.py has a real password sitting in plain text — password="your_password". That's fine for a five-minute experiment, but it's a habit worth breaking immediately. If you ever push this code to GitHub, share it with a classmate, or reuse it in a bigger project, that password goes with it. This lesson moves it somewhere safer.

1. Starting a Fresh, Secured Project

Open your terminal and go into your repository. Create a new project folder:

```
mkdir secured_aice_books_app
cd secured_aice_books_app
```

Create and activate a virtual environment, exactly as before. On PowerShell:

```
.\venv\Scripts\Activate.ps1
```

On Ubuntu:

```
python3 -m venv venv
source venv/bin/activate
```

With (venv) active, install the MySQL connector again, as you did in the last lesson:

```
pip install mysql-connector-python
```

2. What Is an Environment Variable?

An environment variable is simply a value that lives outside your code, in the environment your program runs in, rather than typed directly into the file. Your code asks for it by name when it runs, instead of having it hardcoded.

You'll manage yours using a package called python-dotenv, which reads variables from a plain text file named .env and loads them for you. With (venv) active, install it:

```
pip install python-dotenv
```

Document both packages in requirements.txt:

```
pip freeze > requirements.txt
```

3. Creating the .env File

Create the .env file with vim:

```
vim .env
```

Add your database settings, one per line, with no quotes and no spaces around the equals sign:

```
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=your_password
DB_NAME=aice_books
```

DB_PASSWORD is your real MySQL root password. Save and quit once it's in place.

4. Never Commit .env to Source Control

This is the entire point of the exercise — never commit .env to Git. Create a .gitignore file with vim, and list the folder and file you don't want pushed to GitHub: the venv directory and the .env file itself.

```
venv/
.env
```

Save and quit.

If .env Is Never Committed, How Does the Password Get to the Server?

It doesn't travel with your code at all — each machine running the project (yours, a teammate's, a production server) gets its own local .env file, created once and never pushed. The code that reads it is identical everywhere; only the secret values differ.

Key Takeaways

A hardcoded password in app.py travels with your code everywhere it goes — every push, every clone, every share. An environment variable moves that value outside your code entirely, into a .env file that python-dotenv can read, and that .gitignore makes sure never reaches GitHub. With mysql-connector-python and python-dotenv installed, and DB_HOST, DB_USER, DB_PASSWORD, and DB_NAME safely stored in .env, the next lesson writes the Python code that actually loads these values and uses them to connect — finishing the job of keeping secrets where they belong.