**Database set up/ design**Database 🡪 SpeedCoder

Tables:

1. codeSnippets
2. users
3. userResults

**1. Install bcrypt**

In your virtual environment:

pip install bcrypt

**2. Generate a hash for "password"**

import bcrypt

# user enters a password

plain\_pw = "password".encode("utf-8")

# generate a salt and hash

hashed\_pw = bcrypt.hashpw(plain\_pw, bcrypt.gensalt())

print(hashed\_pw.decode("utf-8")) # store this string in the DB

That prints something like:

$2b$12$KIXQ3n.YhM9MPYAXl.2u6uG6oZzJvh95gVmbiRMmkuhN8QKkhF3T6

You’d insert that into your users.password\_hash.

**3. Verifying on login**

When a user signs in with "password", you check like this:

# candidate password from login form

candidate\_pw = "password".encode("utf-8")

# hash stored in DB (string → bytes)

stored\_hash = hashed\_pw # normally you’d load this from MySQL

# if you saved it as text, convert back with stored\_hash.encode("utf-8")

if bcrypt.checkpw(candidate\_pw, stored\_hash):

print("✅ Password correct")

else:

print("❌ Wrong password")

⚠️ Important:

* bcrypt.hashpw always produces a new, unique string even for the same input (because of the salt). That’s normal — verification still works.
* The column size in your table must be at least **60 characters** (VARCHAR(60) or VARCHAR(255) is fine). Right now you used VARCHAR(25), which is **too short** — it will cut hashes.