**config.py** file.  
This file is used to **store application settings** (like secrets and database credentials) so they don’t sit inside app.py or routes.  
where-It should be in the **root of your project**

**CONFIG.py**

import os

from dotenv import load\_dotenv

# Load values from .env file

load\_dotenv()

# Secret key for Flask session cookies

SECRET\_KEY = os.getenv("SECRET\_KEY", "dev-secret-key")

# Database connection settings

DB\_HOST = os.getenv("DB\_HOST", "localhost")

DB\_USER = os.getenv("DB\_USER", "root")

DB\_PASSWORD = os.getenv("DB\_PASSWORD", "your\_password")

DB\_NAME = os.getenv("DB\_NAME", "training\_center")

 **load\_dotenv()** → loads your .env file into environment variables.

 **SECRET\_KEY** → used by Flask for security (sessions, cookies, CSRF protection).

 **DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME** → used later in models/db.py to connect to MySQL.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++config.py should contain:

Secret key,Database credentials,(Later, you could add debug flags, email server config, etc.)

So in short:

* config.py = central place for settings.
* .env = stores the **real secret values**.
* models/db.py = uses those values to connect to MySQL.

**models/db.py**

This file will handle **connecting to your MySQL database**.

## What this does

* Connects to MySQL using details from .env
* Returns a connection object you can use in models
* If connection fails, it prints an error

## Correct Flow

Step-by-step order should be:

1️⃣ DB setup (users table ✅ done)  
2️⃣ User model (create\_default\_admin + validate\_user ✅ done) 3️⃣ \*\*Auth blueprint\*\* (auth/routes.py) with Flask routes for login/logout 4️⃣ \*\*Templates (HTML)\*\* for login page (templates/auth/login.html`)  
5️⃣ Test login → admin logs in → dashboard (later)