# Flask App Database Setup Guide (with Flask-Migrate)

## 📁 Project Overview

This guide explains how to initialize and manage the SQLite database schema for a Flask app using Flask-Migrate, which is built on top of Alembic.

## 🧰 Requirements

Install required Python packages:

pip install flask flask\_sqlalchemy flask\_migrate flask\_cors werkzeug

## 🏗️ Step-by-Step Setup

1. 1. Set FLASK\_APP Environment Variable

* For Linux/macOS:

export FLASK\_APP=app.py

* For Windows (CMD):

set FLASK\_APP=app.py

1. 2. Initialize Migration Directory (Once Only)

flask db init

This creates a migrations/ folder to store version history of database schema.

1. 3. Create Initial Migration Script

flask db migrate -m "Initial DB migration"

This scans the models in app.py and generates a migration script.

1. 4. Apply the Migration to Create Tables

flask db upgrade

This applies the migration to the SQLite database (pythonlogin.db), creating all necessary tables.

## 🔄 Making Schema Changes Later

Whenever you update the Account model or add a new model:

flask db migrate -m "Describe your schema change"  
flask db upgrade

## ❌ Important Note

Avoid calling db.create\_all() in production code when using Flask-Migrate. Migrations give you more control and safety over your schema changes.

## ✅ Done!

Your Flask app now uses a reliable and professional approach for managing database changes!