

# Flask + SQLAlchemy — Quick Recall Notes 🧠

Purpose: These notes are written **only for revision** so you never forget how to **run Flask apps**, **project structure**, and **create tables using SQLAlchemy**.

## 🌟 Flask Project Structure (Your Preferred Style)

```
project_folder/
    |
    ├── run.py                  # Entry point (or app.py)
    ├── instance/               # Auto-created (SQLite DB, config)
    |
    ├── Components/
    │   ├── __init__.py          # App factory + extensions
    │   ├── models.py            # SQLAlchemy models (tables)
    │   ├── routes.py            # App routes
    │   |
    │   └── templates/
    │       └── index.html      # Add + Edit form
    |
    ├── static/
    │   ├── css/
    │   └── js/
    |
    └── requirements.txt
```

📌 **Golden Rule** - `models.py` → tables - `routes.py` → logic - `run.py` → runs the app

## 💡 Virtual Environment (CMD / Terminal)

### Create venv (once)

```
python -m venv venv
```

### Activate venv

#### Windows (CMD / PowerShell)

```
venv\Scripts\activate
```

#### Linux / Mac

```
source venv/bin/activate
```

## 💡 Install Flask & SQLAlchemy

```
pip install flask flask-sqlalchemy
```

Save dependencies:

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

## ⚙️ \_\_init\_\_.py (App Factory Setup)

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

db = SQLAlchemy()

def create_app():
    app = Flask(__name__)

    app.config['SECRET_KEY'] = 'dev'
    app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
    app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

    db.init_app(app)

    from .routes import main
    app.register_blueprint(main)

    return app
```

## 🏃 run.py (How You Run Flask)

```
from Components import create_app

app = create_app()
```

```
if __name__ == '__main__':
    app.run(debug=True)
```

⬇️ Run command

```
python run.py
```

## 💡 SQLAlchemy Models (Creating Tables)

models.py

```
from . import db

class Student(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)

    def __repr__(self):
        return f"<Student {self.name}>"
```

📌 Each class = **one table**

## 💡 Creating Tables (MOST IMPORTANT PART)

### Method 1: Flask Shell (Recommended)

```
flask shell
```

Inside shell:

```
from Components import db, create_app
app = create_app()
app.app_context().push()

db.create_all()
```

🔧 This creates `database.db` + tables

## Method 2: Temporary Script (Quick)

```
from Components import create_app, db

app = create_app()

with app.app_context():
    db.create_all()
```

Run once:

```
python run.py
```

## Routes Structure (routes.py)

```
from flask import Blueprint, render_template, request, redirect, flash
from .models import Student
from . import db

main = Blueprint('main', __name__)

@main.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']

        student = Student(name=name, email=email)
        db.session.add(student)
        db.session.commit()

        flash('Student Added!')
        return redirect('/')

    students = Student.query.all()
    return render_template('index.html', students=students)
```

## Edit & Delete Pattern (Your Style)

### Edit

```
@main.route('/edit/<int:id>', methods=['GET', 'POST'])
def edit(id):
```

```

student = Student.query.get_or_404(id)

if request.method == 'POST':
    student.name = request.form['name']
    student.email = request.form['email']
    db.session.commit()
    return redirect('/')

return render_template('index.html', student=student)

```

## Delete

```

@main.route('/delete/<int:id>')
def delete(id):
    student = Student.query.get_or_404(id)
    db.session.delete(student)
    db.session.commit()
    return redirect('/')

```



## Common CMD Commands (Don't Forget)

```

# Activate venv
venv\Scripts\activate

# Run app
python run.py

# Flask shell
flask shell

# Install package
pip install package_name

```



## Common Errors & Fixes



RuntimeError: Working outside of application context

Fix:

```

with app.app_context():
    db.create_all()

```

 Tables not created

 Fix checklist: - Model imported? - `db.create_all()` executed? - App context pushed?

---

## Mental Model (Remember This)

- Flask = request handling
  - SQLAlchemy = ORM (Python → SQL)
  - Model = Table
  - Object = Row
  - `db.session.commit()` = SAVE
- 

## Final Reminder

If you remember **only this**, you're safe:

```
activate venv → python run.py → db.create_all()
```

You're not rusty — you just needed a **memory refresh** 