

Flask Study Tracker - Setup Instructions & Codebase

PROJECT: Flask + Bootstrap Study Tracker

This guide includes complete setup steps and code with inline comments.

SETUP INSTRUCTIONS

1. Create a new folder:

```
mkdir study_tracker && cd study_tracker
```

2. Create a virtual environment:

```
python3 -m venv .venv
source .venv/bin/activate    (Mac/Linux)
.venv\Scripts\activate      (Windows)
```

3. Install dependencies:

```
pip install Flask SQLAlchemy
```

4. Create these files/folders:

```
- app.py
- models.py
- /templates/
  - base.html
  - index.html
  - edit.html (optional)
- /static/
  - style.css (optional)
```

5. Run the Flask app:

```
flask run
```

6. Visit:

```
http://127.0.0.1:5000/
```

DATABASE

First run will generate 'database.db' in your folder.

It uses SQLite and SQLAlchemy.

You can explore the DB using tools like DB Browser for SQLite.

File: app.py

```
from flask import Flask, render_template, request, redirect, url_for

# Imports core Flask modules

from models import db, StudyProject

# Imports the database object and model

app = Flask(__name__)

# Create the Flask app instance

app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'

# Set database location

app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# Disable track modifications

db.init_app(app)

# Initialize DB with app

with app.app_context():

    # Push app context for DB ops

    db.create_all()

    # Create tables if they don't exist

@app.route('/')

# Homepage route

def index():

    # Controller for main page

    projects = StudyProject.query.all()

    # Fetch all projects

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

    # Pass to template

@app.route('/add', methods=['POST'])

# POST route to add project

def add_project():

    name = request.form['name']

    # Get name

    description = request.form['description']

    # Get description

    goal = request.form['goal']

    # Get goal

    project = StudyProject(name=name, description=description, goal=goal)

    # Create instance

    db.session.add(project)

    # Add to session

    db.session.commit()

    # Save to DB

    return redirect(url_for('index'))

    # Return home

@app.route('/delete/<int:id>')

# Route to delete project
```

```
def delete_project(id):
    project = StudyProject.query.get_or_404(id)
    # Find by ID or 404
    db.session.delete(project)
    # Delete
    db.session.commit()
    # Commit delete
    return redirect(url_for('index'))

@app.route('/update/<int:id>', methods=['POST'])
    # Update progress route
def update_progress(id):
    project = StudyProject.query.get_or_404(id)
    # Get project
    project.progress = int(request.form['progress'])
    # Update field
    db.session.commit()
    # Save change
    return redirect(url_for('index'))

if __name__ == '__main__':
    # Main entry point
    app.run(debug=True)
    # Run server
```

File: models.py

```
from flask_sqlalchemy import SQLAlchemy

# Import SQLAlchemy

db = SQLAlchemy()

# Initialize DB object

class StudyProject(db.Model):

    # Model for study projects

    id = db.Column(db.Integer, primary_key=True)

    # ID field

    name = db.Column(db.String(100), nullable=False)

    # Name of project

    description = db.Column(db.Text, nullable=True)

    # Optional description

    goal = db.Column(db.String(100), nullable=True)

    # Goal of project

    progress = db.Column(db.Integer, default=0)

    # Progress percent
```

File: templates/base.html

```
<!DOCTYPE html>

# Document type declaration

<html>

# Start HTML

  <head>

# Head section

    <title>Study Projects</title>

# Page title

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet"> # Bootstrap CDN

  </head>

# Close head

  <body>

# Body start

    <div class="container py-4">

# Main container

      <h1 class="mb-4 text-center">Study Projects Tracker</h1>

# Page heading

      {% block content %}{% endblock %}

# Block for dynamic content

    </div>

# Close container

  </body>

</html>
```

File: templates/index.html

```
{% extends 'base.html' %}

# Inherit from base template

{% block content %}

# Start content block

<form method="POST" action="/add" class="mb-4">

# Form for adding projects

    <div class="row g-2">

# Grid row

        <div class="col-md-3"><input name="name" class="form-control" placeholder="Project
Name" required></div> # Name input

        <div class="col-md-4"><input name="description" class="form-control"
placeholder="Description"></div> # Description input

        <div class="col-md-3"><input name="goal" class="form-control"
placeholder="Goal"></div> # Goal input

        <div class="col-md-2"><button class="btn btn-primary w-100">Add</button></div>

# Submit button

    </div></form>

# Close form

    <table class="table table-bordered align-middle">

# Project table

<thead><tr><th>Name</th><th>Description</th><th>Goal</th><th>Progress</th><th>Actions</th></tr></thead> # Table headers

    <tbody>

# Start table body

        {% for project in projects %}

# Loop through projects

            <tr>

# Start row

                <td>{{ project.name }}</td>

# Project name

                <td>{{ project.description }}</td>

# Description

                <td>{{ project.goal }}</td>

# Goal

                <td><form method="POST" action="{{ url_for('update_progress', id=project.id) }}"
class="d-flex"> # Progress form

                    <input name="progress" type="number" min="0" max="100" value="{{
project.progress }}" class="form-control form-control-sm me-2" style="width: 70px;"> # Progress
input

                    <button class="btn btn-sm btn-success">Update</button></form>

# Submit update

                    <div class="progress mt-1"><div class="progress-bar" style="width: {{
```

```
project.progress }}%;">{{ project.progress }}%</div></div> # Progress bar
    </td>
    <td><a href="{{ url_for('delete_project', id=project.id) }}" class="btn btn-sm
btn-danger">Delete</a></td> # Delete link
</tr>

# End row
{% endfor %}

# End loop
</tbody></table>

# Close table
{% endblock %}

# End content block
```

File: templates/edit.html

<!-- edit.html not used in this version -->