Embedd content in website

* Install Flask
  + pip install flask
* App Structure
  + /flask\_app/
  + │── app.py (Main Flask app)
  + │── templates/
  + │ ├── index.html (Main page)
  + │ ├── edit.html (Edit lesson page)
  + │── static/
  + │ ├── styles.css (CSS styles)
* App Code
  + from flask import Flask, render\_template, request, redirect, url\_for
  + import json
  + app = Flask(\_\_name\_\_)
  + # Fake database (replace with SQLite, Firebase, or PostgreSQL)
  + LESSON\_FILE = "lesson.json"
  + def save\_lesson(lesson\_data):
  + """Saves the lesson to a JSON file."""
  + with open(LESSON\_FILE, "w", encoding="utf-8") as f:
  + json.dump(lesson\_data, f, indent=4)
  + def load\_lesson():
  + """Loads the lesson from a JSON file."""
  + try:
  + with open(LESSON\_FILE, "r", encoding="utf-8") as f:
  + return json.load(f)
  + except FileNotFoundError:
  + return {"text": "No lesson available yet.", "tables": "", "images": [], "sources": []}
  + @app.route("/")
  + def index():
  + """Main page displaying the lesson."""
  + lesson = load\_lesson()
  + return render\_template("index.html", lesson=lesson)
  + @app.route("/edit", methods=["GET", "POST"])
  + def edit():
  + """Edit the lesson content."""
  + lesson = load\_lesson()
  + if request.method == "POST":
  + updated\_text = request.form["lesson\_text"]
  + lesson["text"] = updated\_text
  + save\_lesson(lesson)
  + return redirect(url\_for("index"))
  + return render\_template("edit.html", lesson=lesson)
  + if \_\_name\_\_ == "\_\_main\_\_":
  + app.run(debug=True)
* Store and view lessons
  + from flask import Flask, render\_template, request, redirect, url\_for
  + import json
  + import os
  + import time
  + app = Flask(\_\_name\_\_)
  + LESSON\_DIR = "lessons" # Folder to store lessons
  + if not os.path.exists(LESSON\_DIR):
  + os.makedirs(LESSON\_DIR) # Create directory if it doesn't exist
  + def generate\_lesson\_name():
  + """Generates a unique lesson filename based on timestamp."""
  + return f"lesson\_{int(time.time())}.json" # Example: lesson\_1708492031.json
  + def save\_lesson(lesson\_name, lesson\_data):
  + """Saves the lesson to a JSON file with a unique name."""
  + lesson\_path = os.path.join(LESSON\_DIR, lesson\_name)
  + with open(lesson\_path, "w", encoding="utf-8") as f:
  + json.dump(lesson\_data, f, indent=4)
  + def load\_lesson(lesson\_name):
  + """Loads a lesson from a JSON file."""
  + lesson\_path = os.path.join(LESSON\_DIR, lesson\_name)
  + try:
  + with open(lesson\_path, "r", encoding="utf-8") as f:
  + return json.load(f)
  + except FileNotFoundError:
  + return {"text": "No lesson available yet.", "tables": "", "images": [], "sources": []}
  + def list\_lessons():
  + """Lists all saved lessons."""
  + return [f for f in os.listdir(LESSON\_DIR) if f.endswith(".json")]
  + @app.route("/")
  + def index():
  + """Homepage listing all lessons."""
  + lessons = list\_lessons()
  + return render\_template("index.html", lessons=lessons)
  + @app.route("/lesson/<lesson\_name>")
  + def view\_lesson(lesson\_name):
  + """Displays a selected lesson."""
  + lesson = load\_lesson(lesson\_name)
  + return render\_template("lesson.html", lesson=lesson, lesson\_name=lesson\_name)
  + @app.route("/edit/<lesson\_name>", methods=["GET", "POST"])
  + def edit\_lesson(lesson\_name):
  + """Allows editing of an existing lesson."""
  + lesson = load\_lesson(lesson\_name)
  + if request.method == "POST":
  + updated\_text = request.form["lesson\_text"]
  + lesson["text"] = updated\_text
  + save\_lesson(lesson\_name, lesson)
  + return redirect(url\_for("view\_lesson", lesson\_name=lesson\_name))
  + return render\_template("edit.html", lesson=lesson, lesson\_name=lesson\_name)
  + @app.route("/create", methods=["POST"])
  + def create\_lesson():
  + """Creates a new lesson with unique name."""
  + lesson\_name = generate\_lesson\_name()
  + lesson\_data = {
  + "text": request.form["lesson\_text"],
  + "tables": request.form["tables"],
  + "images": request.form.getlist("images"),
  + "sources": request.form.getlist("sources")
  + }
  + save\_lesson(lesson\_name, lesson\_data)
  + return redirect(url\_for("view\_lesson", lesson\_name=lesson\_name))
  + if \_\_name\_\_ == "\_\_main\_\_":
  + app.run(debug=True)

List Lessons

* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>AI Lessons</title>
* </head>
* <body>
* <h1>📚 AI-Generated Lessons</h1>
* <ul>
* {% for lesson in lessons %}
* <li><a href="{{ url\_for('view\_lesson', lesson\_name=lesson) }}">{{ lesson }}</a></li>
* {% endfor %}
* </ul>
* <h2>Create New Lesson</h2>
* <form action="{{ url\_for('create\_lesson') }}" method="post">
* <textarea name="lesson\_text" rows="5" cols="50" placeholder="Enter new lesson..."></textarea><br>
* <button type="submit">Generate Lesson</button>
* </form>
* </body>
* </html>

Lesson Page

* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>Lesson</title>
* </head>
* <body>
* <h1>📖 Lesson: {{ lesson\_name }}</h1>
* <p>{{ lesson["text"] }}</p>
* <a href="{{ url\_for('edit\_lesson', lesson\_name=lesson\_name) }}">✏️ Edit Lesson</a>
* <br>
* <a href="{{ url\_for('index') }}">⬅️ Back to Lesson List</a>
* </body>
* </html>

Edit Website

* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>Edit Lesson</title>
* </head>
* <body>
* <h1>✏️ Edit Lesson: {{ lesson\_name }}</h1>
* <form action="{{ url\_for('edit\_lesson', lesson\_name=lesson\_name) }}" method="post">
* <textarea name="lesson\_text" rows="10" cols="50">{{ lesson["text"] }}</textarea><br>
* <button type="submit">Save Changes</button>
* </form>
* <a href="{{ url\_for('view\_lesson', lesson\_name=lesson\_name) }}">⬅️ Back to Lesson</a>
* </body>
* </html>

Style Website

* body {
* font-family: Arial, sans-serif;
* margin: 20px;
* padding: 20px;
* background-color: #f4f4f4;
* }
* h1 {
* color: #333;
* }
* pre {
* background: #fff;
* padding: 10px;
* border: 1px solid #ddd;
* }
* img {
* border-radius: 5px;
* margin: 10px;
* }

Run App

* python app.py
* <http://127.0.0.1:5000/>

Resources

* <https://flask.palletsprojects.com/en/stable/installation/#install-flask>
* <https://flask.palletsprojects.com/en/stable/quickstart/>