create flashcard app using mongodb, fastapi, react

**ChatGPT said:**

Creating a flashcard app using MongoDB, FastAPI, and React involves multiple steps. Below is a high-level overview of how you can implement it, including the key components and the code structure.

**Key Components:**

1. **Backend**: FastAPI to handle the API endpoints.
2. **Database**: MongoDB for storing flashcards.
3. **Frontend**: React for the user interface.

**Steps to Create the Flashcard App:**

**Step 1: Setup MongoDB**

Ensure you have MongoDB installed and running. You can install MongoDB locally or use a service like [MongoDB Atlas](https://www.mongodb.com/cloud/atlas) for a cloud-hosted instance.

* **Database Structure**: Each flashcard can have a front side (question) and a back side (answer).
* **Collection**: flashcards
* **Document**: Each document in flashcards will look like this:

json

Copy

{

"\_id": ObjectId(),

"question": "What is 2 + 2?",

"answer": "4"

}

**Step 2: Backend - FastAPI**

**Install FastAPI and MongoDB Dependencies**:

bash

Copy

pip install fastapi uvicorn pymongo

1. **Create the FastAPI backend (main.py)**: This will define the API endpoints to get, create, and delete flashcards.

python

from fastapi import FastAPI, HTTPException

from pydantic import BaseModel

from pymongo import MongoClient

from bson import ObjectId

from typing import List

# Initialize FastAPI app

app = FastAPI()

# MongoDB connection

client = MongoClient("mongodb://localhost:27017/")

db = client["flashcard\_db"]

flashcards\_collection = db["flashcards"]

# Pydantic model for Flashcard

class Flashcard(BaseModel):

question: str

answer: str

class FlashcardInDB(Flashcard):

id: str

# Endpoint to create a new flashcard

@app.post("/flashcards/", response\_model=FlashcardInDB)

async def create\_flashcard(flashcard: Flashcard):

result = flashcards\_collection.insert\_one(flashcard.dict())

new\_flashcard = flashcard.dict()

new\_flashcard['id'] = str(result.inserted\_id)

return new\_flashcard

# Endpoint to get all flashcards

@app.get("/flashcards/", response\_model=List[FlashcardInDB])

async def get\_flashcards():

flashcards = []

for flashcard in flashcards\_collection.find():

flashcards.append({"id": str(flashcard["\_id"]), "question": flashcard["question"], "answer": flashcard["answer"]})

return flashcards

# Endpoint to get a flashcard by id

@app.get("/flashcards/{flashcard\_id}", response\_model=FlashcardInDB)

async def get\_flashcard(flashcard\_id: str):

flashcard = flashcards\_collection.find\_one({"\_id": ObjectId(flashcard\_id)})

if flashcard is None:

raise HTTPException(status\_code=404, detail="Flashcard not found")

return {"id": str(flashcard["\_id"]), "question": flashcard["question"], "answer": flashcard["answer"]}

# Endpoint to delete a flashcard by id

@app.delete("/flashcards/{flashcard\_id}")

async def delete\_flashcard(flashcard\_id: str):

result = flashcards\_collection.delete\_one({"\_id": ObjectId(flashcard\_id)})

if result.deleted\_count == 0:

raise HTTPException(status\_code=404, detail="Flashcard not found")

return {"message": "Flashcard deleted successfully"}

1. **Run the FastAPI server**:

bash

Copy

uvicorn main:app --reload

* This will start the FastAPI backend on http://localhost:8000.

**Step 3: Frontend - React**

**Create a React app** using create-react-app:

bash

Copy

npx create-react-app flashcard-app

cd flashcard-app

npm start

Install dependencies for HTTP requests (Axios):

bash

Copy

npm install axios

1. **Create a Flashcard Component**: In src/components/Flashcard.js, create a simple component to display the flashcards.

javascript

Copy

import React, { useState, useEffect } from 'react';

import axios from 'axios';

const FlashcardApp = () => {

const [flashcards, setFlashcards] = useState([]);

const [question, setQuestion] = useState("");

const [answer, setAnswer] = useState("");

useEffect(() => {

// Fetch flashcards on load

axios.get('http://localhost:8000/flashcards/')

.then(response => {

setFlashcards(response.data);

})

.catch(error => {

console.error("There was an error fetching the flashcards!", error);

});

}, []);

const addFlashcard = () => {

axios.post('http://localhost:8000/flashcards/', { question, answer })

.then(response => {

setFlashcards([...flashcards, response.data]);

setQuestion('');

setAnswer('');

})

.catch(error => {

console.error("There was an error adding the flashcard!", error);

});

};

const deleteFlashcard = (id) => {

axios.delete(`http://localhost:8000/flashcards/${id}`)

.then(() => {

setFlashcards(flashcards.filter(card => card.id !== id));

})

.catch(error => {

console.error("There was an error deleting the flashcard!", error);

});

};

return (

<div>

<h1>Flashcard App</h1>

<input

type="text"

placeholder="Question"

value={question}

onChange={(e) => setQuestion(e.target.value)}

/>

<input

type="text"

placeholder="Answer"

value={answer}

onChange={(e) => setAnswer(e.target.value)}

/>

<button onClick={addFlashcard}>Add Flashcard</button>

<h2>Flashcards</h2>

<ul>

{flashcards.map(flashcard => (

<li key={flashcard.id}>

<strong>{flashcard.question}</strong> - {flashcard.answer}

<button onClick={() => deleteFlashcard(flashcard.id)}>Delete</button>

</li>

))}

</ul>

</div>

);

};

export default FlashcardApp;

1. **Update App.js** to use FlashcardApp component:

javascript

Copy

import React from 'react';

import './App.css';

import FlashcardApp from './components/Flashcard';

function App() {

return (

<div className="App">

<FlashcardApp />

</div>

);

}

export default App;

1. **Run the React App**: In the flashcard-app directory, run:

bash

Copy

npm start

* This will run the React app on http://localhost:3000.

**Final Structure:**

* **Backend** (FastAPI):
  + Handles API endpoints for CRUD operations.
  + MongoDB for storing flashcards.
* **Frontend** (React):
  + Displays flashcards and allows users to add or delete them.
  + Uses Axios to communicate with the FastAPI backend.

**Running the Application:**

1. Run MongoDB locally (or use MongoDB Atlas).
2. Start the FastAPI backend server using uvicorn main:app --reload.
3. Start the React app using npm start in the React project directory.

Now, you should have a basic flashcard app where you can:

* Add new flashcards.
* View all flashcards.
* Delete flashcards.

Let me know if you need more details on any specific part of the process!