Esercizio: Bacheca delle Idee – Mini API CRUD con Express e MongoDB

Obiettivo

Costruire una piccola **API REST** che gestisce una *bacheca virtuale di idee creative*:
un luogo digitale dove ogni utente può lasciare la sua idea, leggerne altre, modificarla o cancellarla.

L'obiettivo è applicare i concetti di **CRUD** (Create, Read, Update, Delete) in un contesto più realistico e coinvolgente.

Scenario narrativo

Immagina che la tu stia progettando una **piattaforma collaborativa per startup o inventori**.
Ogni persona può:

- proporre una nuova idea
- leggere le idee degli altri
- aggiornare o migliorare la propria idea
- cancellare quella che non serve più

Costruirai **il backend (l'API)** che permetterà tutto questo.

Struttura del progetto

```
ideas-api/
│
├── models/
│   └── idea.js
│
├── routes/
│   └── ideas.js
│
├── .env
├── package.json
├── server.js
└── README.md
```

Entità principale: "Idea"

Ogni idea conterrà i seguenti campi principali:

Campo	Tipo	Descrizione
`author`	String	Nome dell'autore dell'idea
`title`	String	Titolo breve dell'idea
`description`	String	Testo che descrive l'idea
`category`	String	Categoria (es. tecnologia, educazione, ambiente…)

Campo	Tipo	Descrizione
`createdAt`	Date	Data di creazione (gestita automaticamente)

Requisiti tecnici

Il progetto deve includere:

- Un server **Express**
- Connessione a **MongoDB Atlas** tramite variabili in `.env`
- Un modello Mongoose `Idea`
- Un router `/api/ideas` con le seguenti rotte:

Metodo	Rotta	Azione
POST	`/api/ideas`	Crea una nuova idea
GET	`/api/ideas`	Mostra tutte le idee
GET	`/api/ideas/:id`	Mostra una singola idea
PATCH	`/api/ideas/:id`	Aggiorna titolo o descrizione
DELETE	`/api/ideas/:id`	Cancella un'idea

Indicazioni per lo sviluppo

1 **Crea il progetto e installa le dipendenze**

```
mkdir ideas-api && cd ideas-api
npm init -y
npm install express mongoose cors dotenv
npm install -D nodemon
```

2 **Crea la struttura delle cartelle e i file principali**

3 **Implementa le 5 rotte CRUD** nel file `routes/ideas.js`

4**T esta l'API** con Postman o con i comandi `curl`:

- Crea una nuova idea
- Leggi tutte le idee
- Aggiorna un titolo
- Cancella un'idea

Suggerimento creativo

Puoi aggiungere un campo extra, ad esempio:

- `likes` (numero di "mi piace") → tipo `Number`, default `0`
- `status` ("in corso", "approvata", "scartata") → tipo `String`

Così la tua API diventa la base di una futura **app di brainstorming o startup lab**.

Obiettivi didattici

- Consolidare la logica CRUD completa
- Capire la struttura base **MVC** di un backend Node.js
- Usare **Postman** come strumento di test
- Riconoscere e risolvere errori di connessione, validazione e sintassi

Attività extra (facoltativa)

Prova a creare un semplice file `index.html` con un form che invia i dati tramite `fetch()` alla tua API.
Vedrai come **frontend e backend** possono comunicare tra loro in modo dinamico.