How to Run This Project

This project is a **full-stack app**: a **GraphQL API** (Apollo Server + MongoDB) and a **Next.js frontend** (IMDB Chat) that talks to it. Both need to run. Everything below assumes you start from the **project root** — the folder that contains `graphql` and `frontend`.

Project layout

```
<project root>   ← you are here
├── graphql/      ← backend (Apollo Server, port 4000)
│   ├── package.json
│   ├── index.js
│   └── .env      (optional; for MongoDB URL)
└── frontend/     ← frontend (Next.js, port 3000)
    ├── package.json
    └── app/
```

What you need

- **Node.js** (v16 or newer). Check: `node -v`
- **MongoDB** — either:
    - Local MongoDB, or
    - A MongoDB Atlas (or other) connection string for the backend

---

Step 1 — Backend (GraphQL API)

1. Open a terminal and go into the backend folder:

    ```
    cd graphql
    ```

2. Install dependencies:

```
npm install
```

3. **(Optional)** If you use **MongoDB Atlas** (or any remote URL), create a file named `.env` inside `graphql` with:
```
MONGODB_URL=your_connection_string_here
```

   If you use **local MongoDB**, you can skip this; the app will use `mongodb://127.0.0.1:27017/imdb`.

4. Start the backend:

```
npm start
```

   Leave this terminal running. You should see something like **"Server running at http://localhost:4000"** and **"MongoDB Connected"**.

Step 2 — Frontend (Next.js)

1. Open a **second** terminal and go into the frontend folder:

```
cd frontend
```

2. Install dependencies:

```
npm install
```

3. **(Optional)** If your GraphQL API is **not** at `http://localhost:4000`, create a `.env.local` in the `frontend` folder with:
```
GRAPHQL_URL=http://your-api-url-here/
```

   For local runs with the backend on 4000, you can skip this.

4. Start the frontend:
```
npm run dev
```

   Leave this terminal running. You should see something like **"Ready on http://localhost:3000"**.

Step 3 — Use the app

- **Main app (chat UI):** open **http://localhost:3000** in your browser. You can ask about movies and actors; the frontend calls the GraphQL API.
- **GraphQL API / Sandbox:** open **http://localhost:4000** to use Apollo's Sandbox (or Playground) and run queries/mutations directly.

Order matters: start the **backend first** (port 4000), then the **frontend** (port 3000). The frontend expects the API at `http://localhost:4000` unless you set `GRAPHQL_URL`.

LLMs and tools used for this assignment

- **In-app LLM:** This project uses **Ollama** with the **gemma3:4b** model (see `graphql/ai/ollamaService.js`). The backend sends natural-language requests to a local Ollama server at `http://localhost:11434`; that model parses user intent and returns structured actions (e.g. query, create, update, delete) for the GraphQL API.