

Prototyping – Übung 6

Themen: SvelteKit, Daten in einer MongoDB verwalten

Starte entweder mit der Beispiellösung oder deiner eigenen Lösung der Übung 5.

- a) Als erstes benötigst du Demodaten in einer MongoDB. Erstelle dazu mit Compass oder auf <https://cloud.mongodb.com> eine neue Datenbank *ScreenStackDB* mit der Collection *movies*:

Füge dann via *Insert Document* Demodaten in die Collection ein. (Du findest die Beispieldaten auf Moodle.) Du kannst die Demodaten natürlich anpassen oder komplett selbst erzeugen (z.B. andere Filme etc.)

```

1 <pre>[{"_id": {"$oid": "6630e72c95e12055f661ff13"}, "title": "Das Geheimnis von Altura", "year": 2024, "length": "120 Minuten", "actors": ["Lena Herzog", "Maximilian Schröder", "Sophia Neumann"], "poster": "/images/Altura.png", "watchlist": false}, {"_id": {"$oid": "6630e72c95e12055f661ff14"}, "title": "Sternenwanderer", "year": 2025, "length": "143 Minuten", "actors": ["Hannah Fischer", "Julian Becker"]}]</pre>

```

- b) Um dich später in der Server-Anwendung auf die MongoDB zu verbinden, benötigst du den Connection String. Wähle dazu im Menü links den Eintrag *Database* → *Clusters* und klicke dann auf *Connect*.

The screenshot shows the MongoDB Atlas interface. On the left sidebar under 'DATABASE', the 'Clusters' option is selected and highlighted with a red box. In the main content area, there is a 'Clusters' card for 'Cluster0'. A 'Connect' button is located at the bottom right of this card and is also highlighted with a red box.

The screenshot shows the 'Connect to Cluster0' dialog. It consists of three steps: 'Set up connection security' (completed), 'Choose a connection method' (completed), and 'Connect' (not yet reached). The 'Choose a connection method' step is currently active, showing the 'Drivers' section which is also highlighted with a red box. A blue speech bubble on the right side of the screen contains the text: 'Kopiere diesen Connection String. Du benötigst ihn im nächsten Schritt.' (Copy this connection string. You will need it in the next step.) Below this, the connection string is displayed in a code editor-like field:

```
mongodb+srv://<db_username>:<db_password>@cluster0.gdnpu.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
```

Below the code editor, a note says: 'Replace <db_password> with the password for the <db_username> database user. Ensure any option params are URL encoded.' At the bottom of the dialog, there is a 'RESOURCES' section with links to 'Get started with the Node.js Driver' and 'Node.js Starter Sample App'.

- c) Lade die Beispiellösung der Übung "Movie Watchlist – Teil 1" von Moodle und öffne sie in Visual Studio Code. (Oder verwende deine eigene Lösung der Übung.)

Erstelle im obersten Ordner des Projekts (= der Ordner, in dem sich `package.json` befindet) eine Datei mit dem Namen **.env** (**beachte den Punkt am Beginn des Dateinamens**) und füge dort den Connection String **ohne Anführungszeichen** aus dem vorherigen Schritt ein:

```
DB_URI=mongodb+srv://...
```

Ersetze im Connection String `<db_username>` und `<db_password>` durch deinen DB-Benutzernamen und das dazugehörige DB-Passwort (**ohne die spitzen Klammern <...>**).

Hinweis: Verwende nicht deine MongoDB-Accountdaten, sondern die Zugangsdaten eines Datenbankusers. Wenn du dir nicht mehr sicher bist, kannst du auf MongoDB Atlas unter *Security → Database Access* die Liste der DB-User anschauen und verwalten.

Lade ausserdem die Datei `db.js` von Moodle und speichere sie im Ordner `lib`. In dieser Datei sind bereits die Funktionen für den Zugriff auf die MongoDB implementiert, die du benötigst.

- d) Falls du die Beispiellösung verwendest, denke daran, dass du die Abhängigkeiten einmalig mit

```
npm install
```

installieren musst.

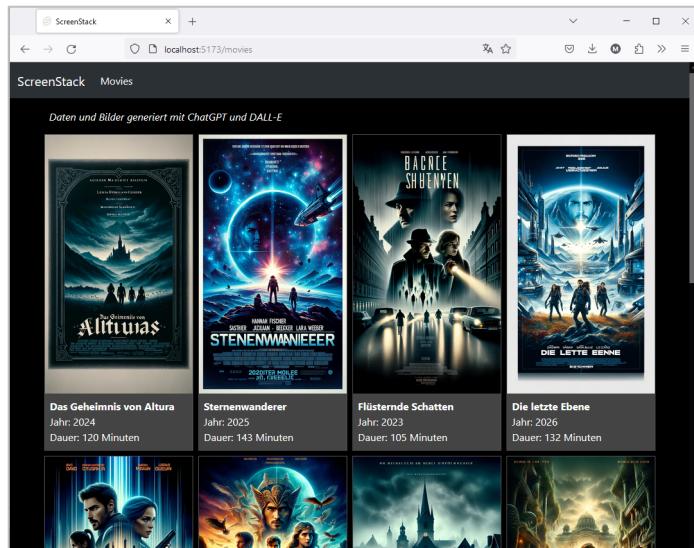
Du benötigst ausserdem das `mongodb`-Package. Installiere es mit

```
npm install mongodb
```

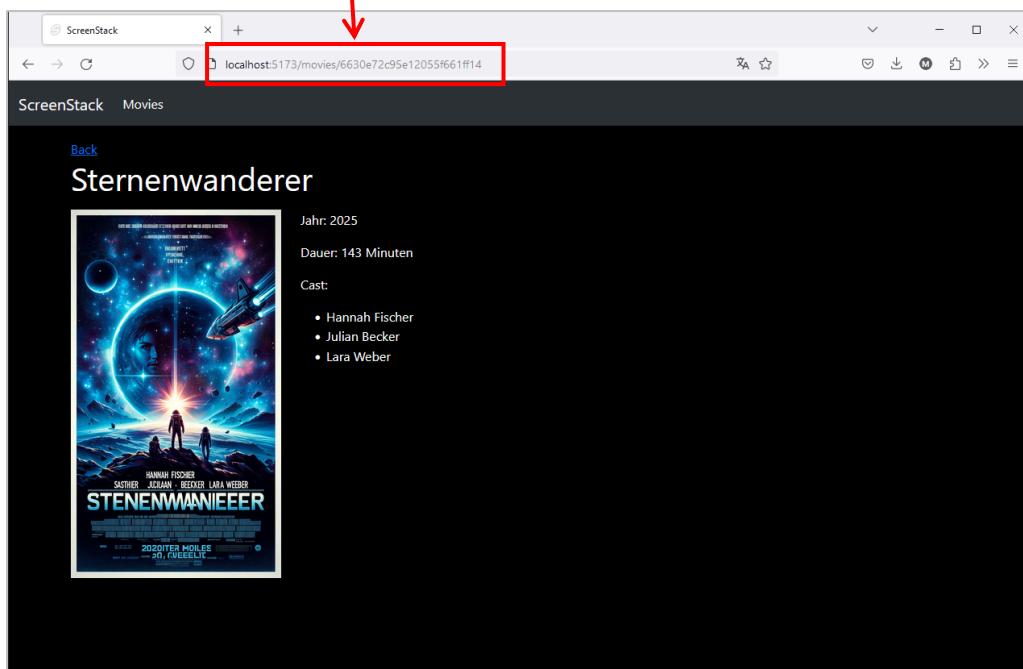
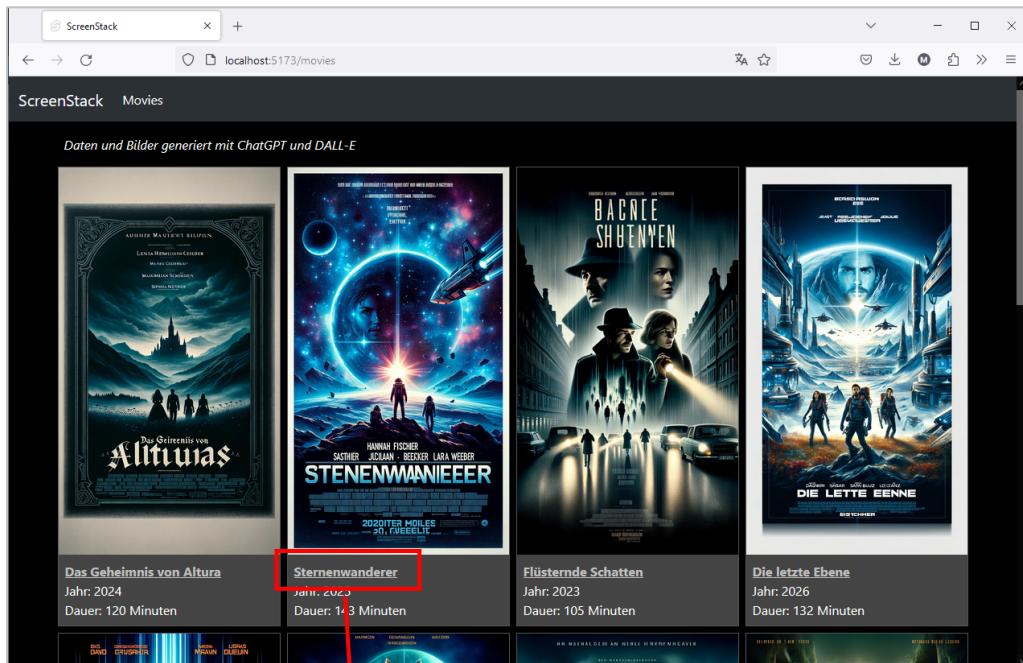
Starte die App dann wie gewohnt mit

```
npm run dev
```

- e) Verwende nun – wie in der Vorlesung gezeigt – die `load`-Funktion, um die Daten aus der MongoDB zu laden. Überprüfe, ob die Daten auf der Page `/movies` korrekt angezeigt werden:



- f) Nutze die in der Vorlesung kennengelernten Konzepte, um eine Detailansicht für Filme zu implementieren. Bei einem Klick auf einen Filmtitel soll zur Detailseite navigiert werden:

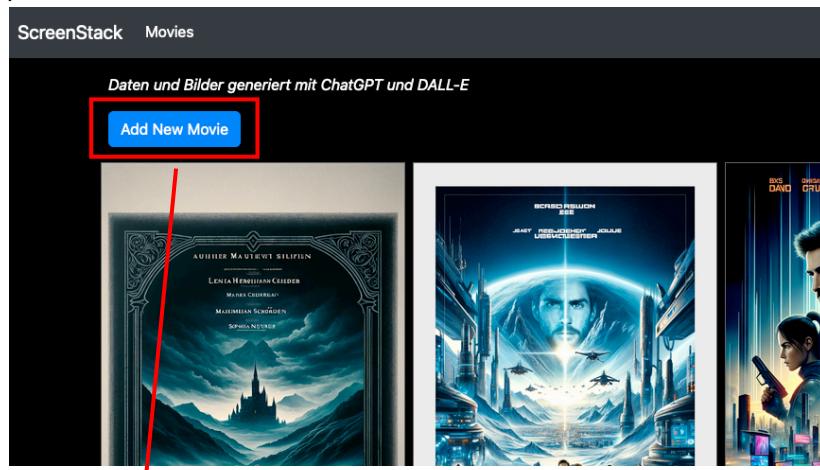


Hinweise:

- Verwende die ID des Movies als Pfadparameter.
- Nutze in der Load-Funktion der Detailseite die in `db.js` implementierte Funktion `getMovie(id)`, um die Filminformationen zu laden.
- Die Seite muss nicht exakt so wie in den Screenshots aussehen. Du kannst beliebige Anpassungen vornehmen.

- g) Erstelle eine Seite mit einem Formular, um neue Filme zu erfassen. Orientiere dich am in der Vorlesung gezeigte Beispiel.

/movies



/movies/create

Add a Movie

Name
Barbie

Year
2023

Length
114 Minuten

Add Movie

Hinweise:

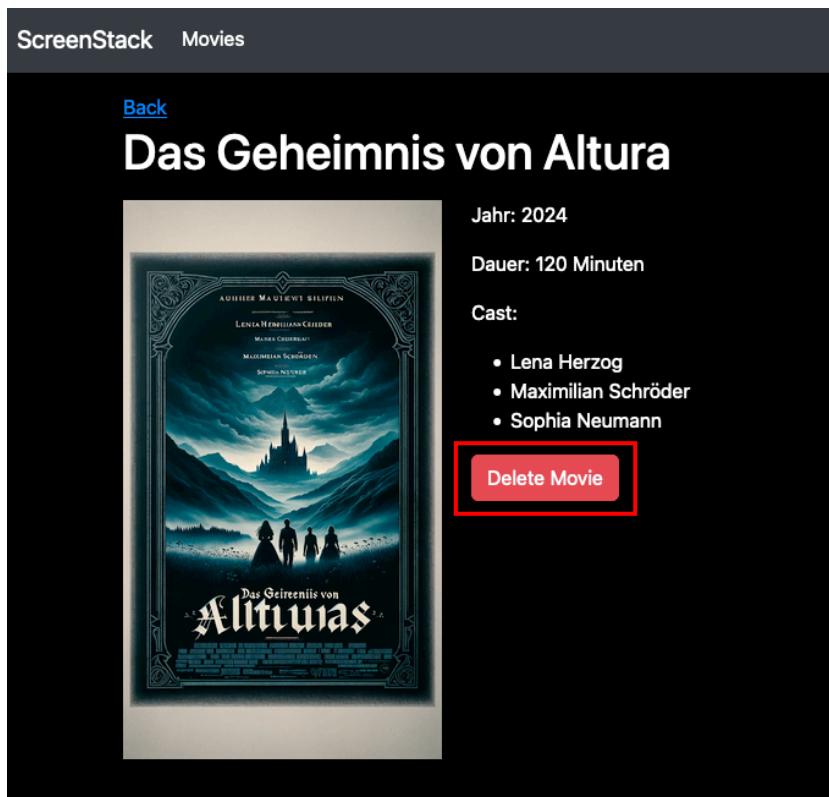
- Beschränke dich beim Erfassen von Filmen auf Namen, Jahr und Dauer. Für das Bild soll lediglich die URL erfasst werden. Du musst keinen Bild-Upload implementieren. Die Datenbankfunktion `createMovie(...)` ergänzt die fehlenden Angaben mit leeren Daten bzw. einem Platzhalterbild.

Beispiel für den Aufruf der Funktion:

```
let movie = {
  title: "Barbie",
  year: 2023,
  length: "114 Minuten"
};
await db.createMovie(movie);
```

- Das oben erwähnte Platzhalterbild muss den Namen `placeholder.jpg` haben und sich im Ordner `/static/images` befinden. Du kannst entweder ein eigenes Platzhalterbild erstellen oder das auf Moodle zur Verfügung gestellte Bild verwenden.

- h) **Optional:** Implementiere auf der Detailseite die Möglichkeit, einen Film aus der Datenbank zu löschen.



Hinweis: Nach dem Löschen des Films soll automatisch zur Übersichtsseite navigiert werden.
Dies ist mit einem *redirect* möglich:

```
import { redirect } from "@sveltejs/kit";
// ...
export const actions = {
  delete: async ({ request }) => {
    // ...
    redirect(303, "/movies");
  },
};
```

Ausblick

In der nächsten Woche werden wir in der Grossklasse einige optionale Erweiterungen implementieren, z.B. Movies aktualisieren oder Movies einer Watchlist hinzufügen.

Abgabe

Gib dein Projekt als Zip-Datei auf Moodle ab. Achte darauf, dass der Ordner `node_modules` **nicht** in der Zip-Datei enthalten ist.