



React Day  
Berlin

[About](#)

[Speakers](#)

[Schedule](#)

[Workshops](#)

[Sponsors](#)

[Location](#)

[FAQ](#)



Build apps, not walls

# React Day Berlin 2024

## AI, compiler etc



December 13 & 16, 2024

Berlin and online

RESERVE YOUR SPOT

ATTEND REMOTELY

800 Devs attending in-person

5K joining in remotely

45+ speakers & trainers sharing insights



# Production-ready Next.js App with Cursor AI


Maurice de Beijer  
@mauricedb



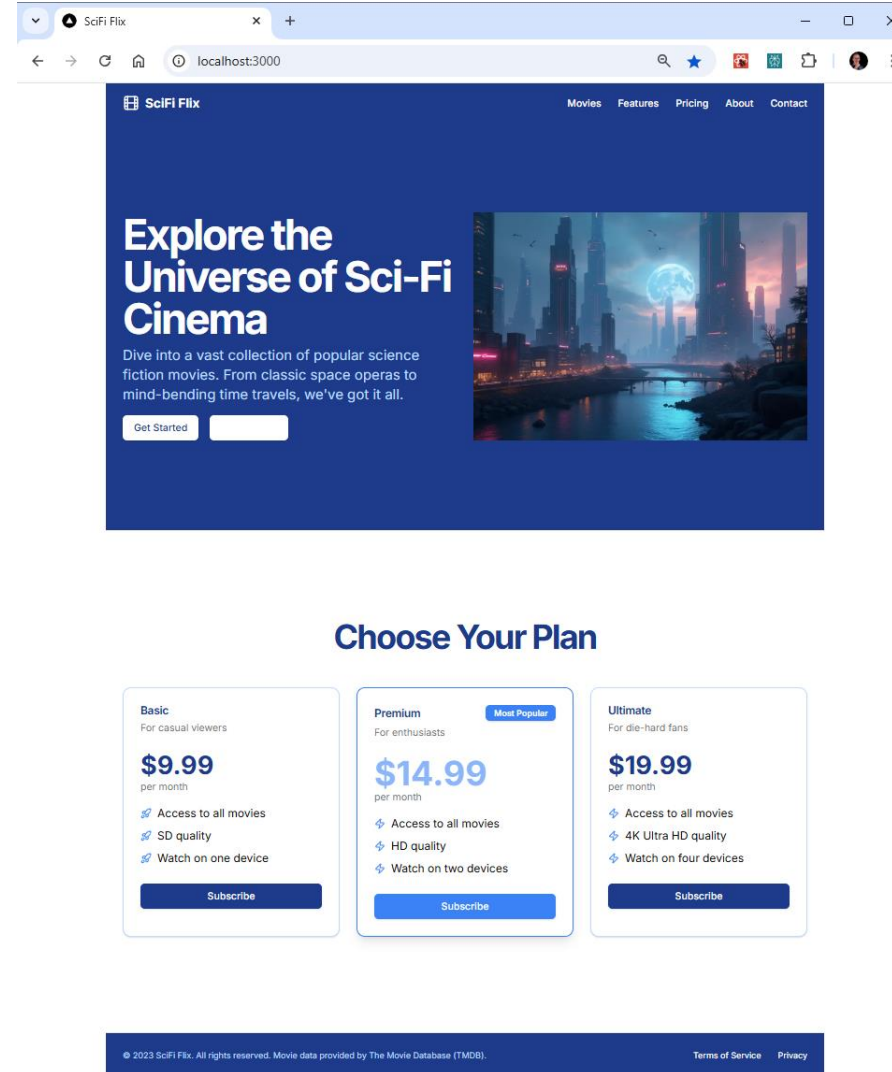
- Maurice de Beijer
- The Problem Solver
- Freelance developer/instructor
- Twitter: [@mauricedb](https://twitter.com/mauricedb)
- Web: <https://www.theproblemsolver.dev/>
- E-mail: [maurice.de.beijer@gmail.com](mailto:maurice.de.beijer@gmail.com)



# What We'll Build Today

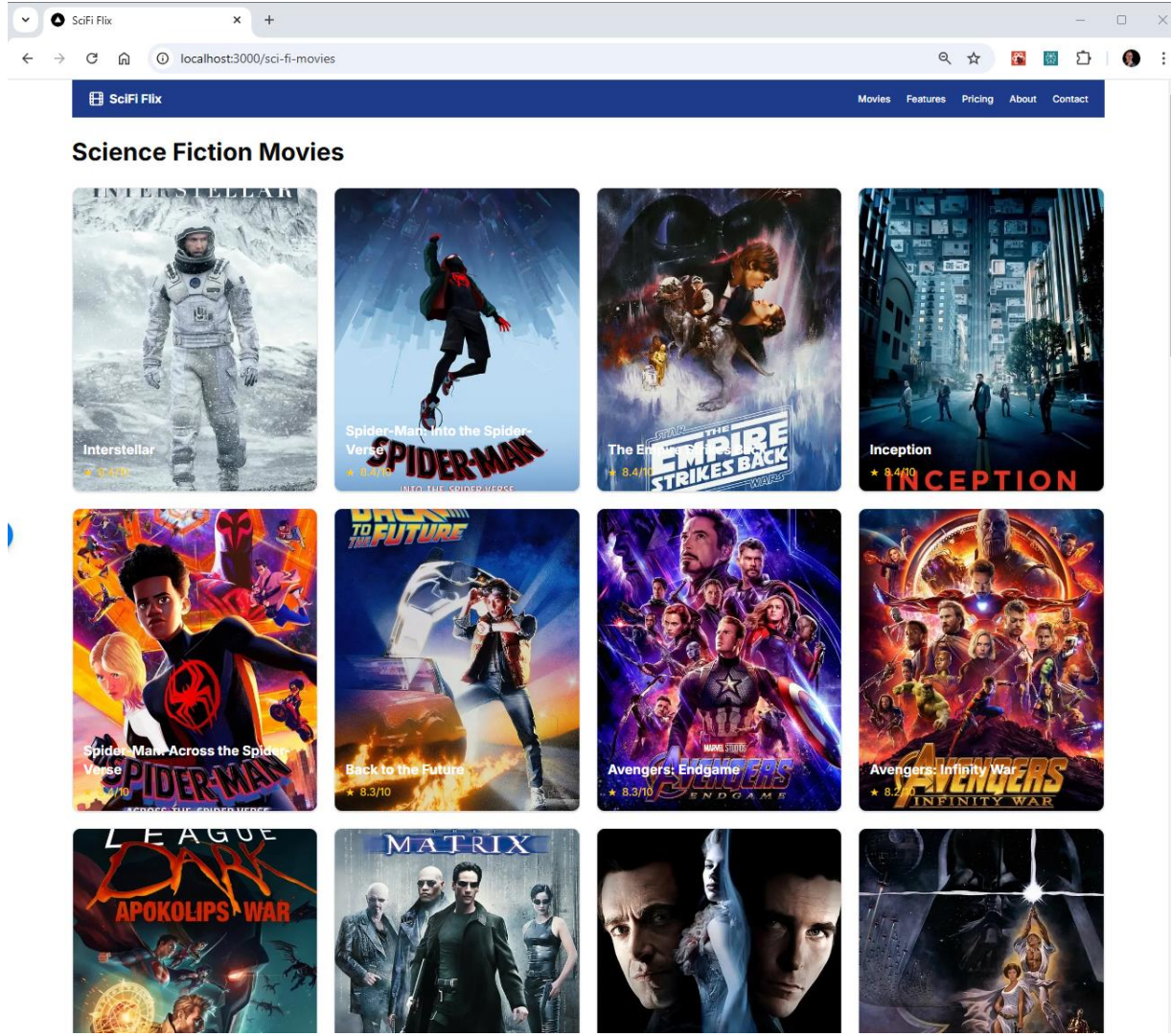
- **Sci-Fi Movie Browsing Application**
  - Landing page with hero section and pricing
  - Grid with sci-fi movies in cards
  - Movie details page
  - Movie editing functionality
- **Learning Objectives**
  - Building robust Next.js applications
  - Use the Cursor IDE and Vercel v0
  - Generate code and database using Cursor
  - How to prompt AI's to get the best results
  - Have the AI fix it's own 
  - Get AI help with code reviews

# Landing page

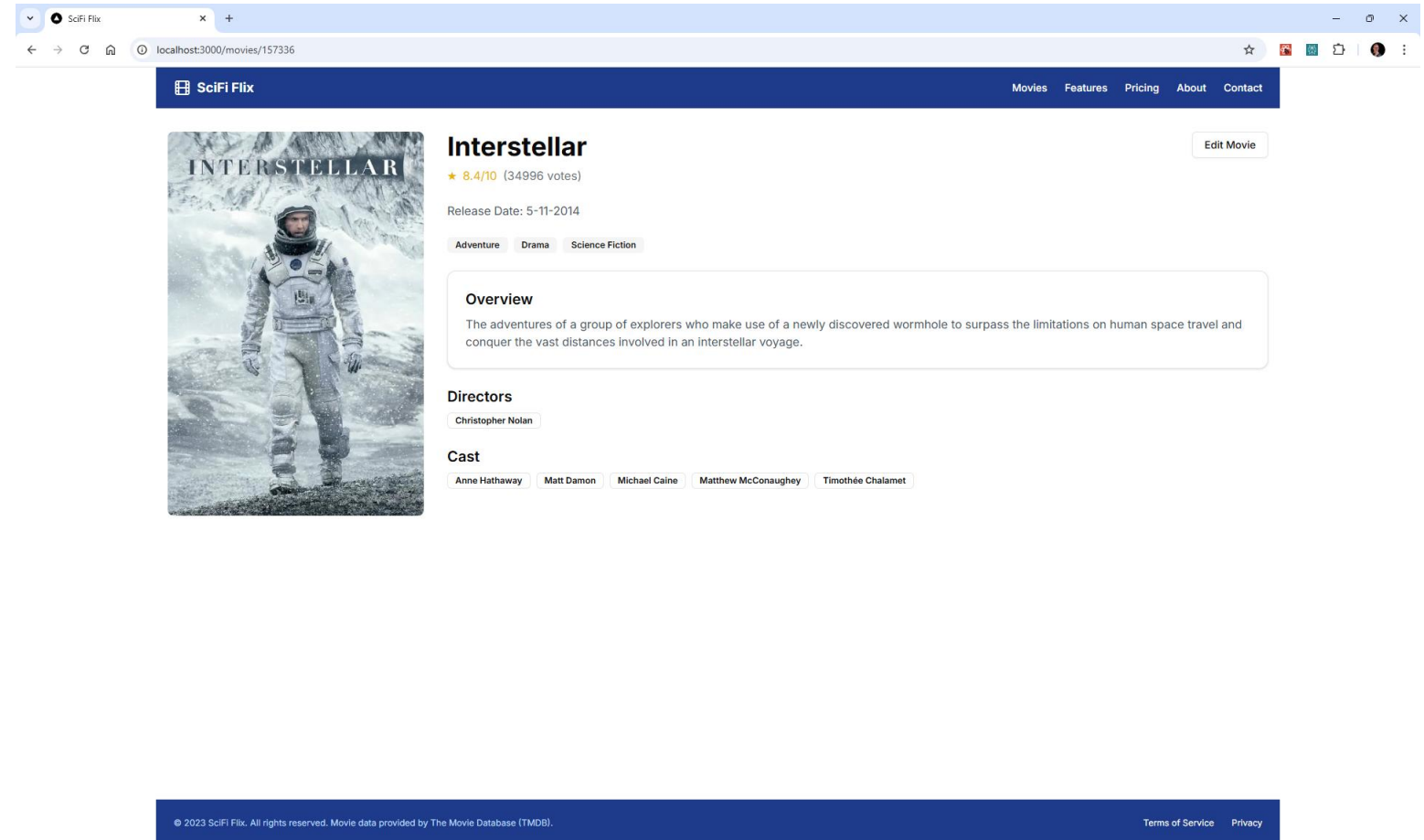




# Movies page



# Movie details page



# Movie editor page

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/movies/157336/edit'. The page title is 'SciFi Flix'. The main content area is titled 'Edit Movie' and contains the following fields and controls:

- Title:** A text input field containing 'Interstellar'.
- Overview:** A text area containing 'The adventures of a group of explorers who make use of a newly discovered wormhole to surpass the limitations on human space travel and conquer the vast distances involved in an interstellar voyage.'
- Release Date:** A date input field showing '05/11/2014' with a calendar icon.
- Genres:** A collection of buttons representing different genres. The selected genres are 'Adventure', 'Drama', and 'Science Fiction'. Other available genres include Action, Animation, Comedy, Crime, Documentary, Family, Fantasy, History, Horror, Music, Mystery, Romance, TV Movie, Thriller, War, and Western.
- Buttons:** 'Cancel' and 'Save Changes' buttons are located at the bottom right of the form.

The footer of the page contains the copyright notice '© 2023 SciFi Flix. All rights reserved. Movie data provided by The Movie Database (TMDB).' and links to 'Terms of Service' and 'Privacy'.



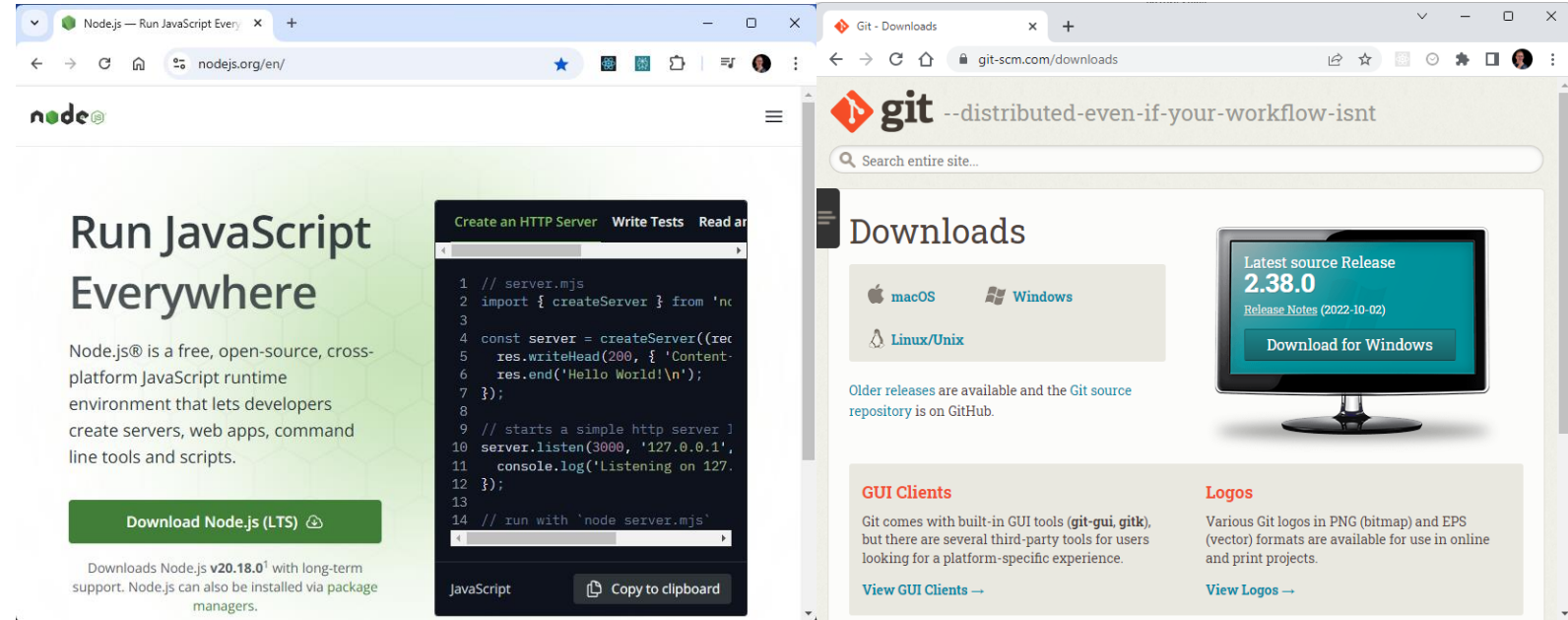


# Prerequisites

Install Node & NPM

Install the GitHub repository

# Install Node.js & NPM



```
Windows PowerShell
PS C:\demos> node --version
v20.9.0
PS C:\demos> git --version
git version 2.45.2.windows.1
PS C:\demos> npm --version
10.2.4
PS C:\demos> |
```

# Following Along

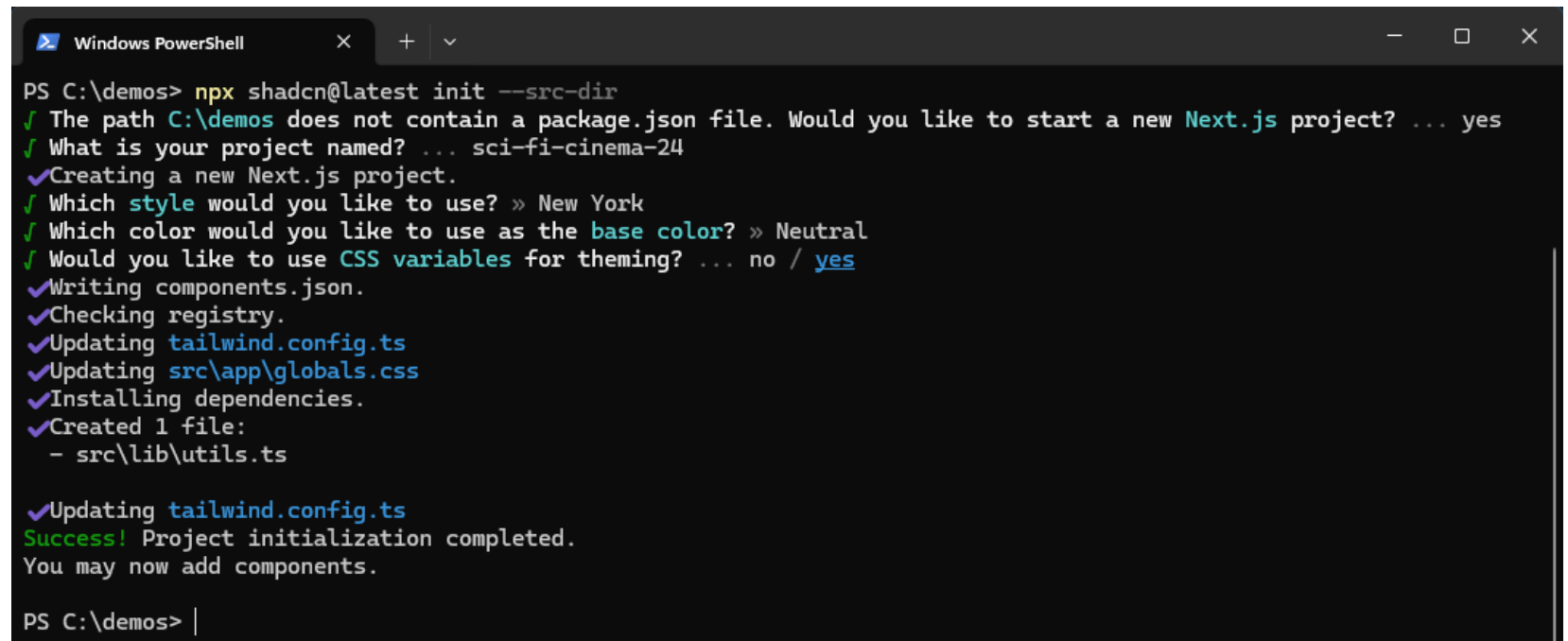


```
prisma > TS seeds > ...
37  async function main() {
89    update: {},
90    create: {
91      id: genre.id,
92      name: genre.name,
93    },
94  });
95  }
96
97  // Create movies with relationships
98  console.log('Seeding movies ... ');
99  for (const movie of moviesData) {
100    await prisma.movie.upsert({
101      where: { id: movie.id },
102      update: {},
103      create: {
104        id: movie.id,
105        backdropPath: movie.backdropPath,
106        overview: movie.overview,
107        popularity: movie.popularity,
108        posterPath: movie.posterPath,
109        releaseDate: new Date(movie.releaseDate),
110        title: movie.title,
111        voteAverage: movie.voteAverage,
112        voteCount: movie.voteCount,
113        actors: {
114          connect: movie.actorIds.map((id: number) => ({ id })),
```

- Repo: <https://github.com/mauricedb/sci-fi-cinema-24>
- Slides: <https://www.theproblemsolver.dev/docs/react-day-berlin-2024.pdf>

# What have I done so far?

- Create a new Next.js app with shadcn/ui
  - `npx shadcn@latest init --src-dir`
- Add sample movie JSON data



```
Windows PowerShell
PS C:\demos> npx shadcn@latest init --src-dir
✓ The path C:\demos does not contain a package.json file. Would you like to start a new Next.js project? ... yes
✓ What is your project named? ... sci-fi-cinema-24
✓ Creating a new Next.js project.
✓ Which style would you like to use? » New York
✓ Which color would you like to use as the base color? » Neutral
✓ Would you like to use CSS variables for theming? ... no / yes
✓ Writing components.json.
✓ Checking registry.
✓ Updating tailwind.config.ts
✓ Updating src\app\globals.css
✓ Installing dependencies.
✓ Created 1 file:
  - src\lib\utils.ts

✓ Updating tailwind.config.ts
Success! Project initialization completed.
You may now add components.

PS C:\demos> |
```

# The changes



Commits · mauricedb/sci-fi-cine x +

github.com/mauricedb/sci-fi-cinema-24/c...

mauricedb / sci-fi-cinema-24

Code Issues Pulls Actions Projects Wiki Settings

### Commits

main All users All time

Commits on Nov 29, 2024

<b>Added project setup instructions</b> mauricedb committed 3 hours ago	a498a88	📄 <>
<b>Limit width</b> mauricedb committed 3 hours ago	7b1b1d7	📄 <>
<b>Movie Edit Page</b> mauricedb committed 3 hours ago	38271ee	📄 <>
<b>Movie Details Page Frontend</b> mauricedb committed 3 hours ago	2ef22d4	📄 <>
<b>Movie Details Page Backend</b> mauricedb committed 3 hours ago	34c0736	📄 <>
<b>Navigation Menu Refactor</b> mauricedb committed 3 hours ago	eb24242	📄 <>
<b>Movie cards</b> mauricedb committed 3 hours ago	3aff8df	📄 <>
<b>Refactoring with a prompt file</b> mauricedb committed 3 hours ago	7825cf9	📄 <>
<b>Composer Notepads and Prompt files</b> mauricedb committed 3 hours ago	cfa6d5b	📄 <>
<b>Movies List Page</b> mauricedb committed 3 hours ago	a2f283a	📄 <>

# Clone the GitHub Repository

```
Windows PowerShell
PS C:\demos> git clone https://github.com/mauricedb/sci-fi-cinema-24.git
Cloning into 'sci-fi-cinema-24'...
remote: Enumerating objects: 170, done.
remote: Counting objects: 100% (170/170), done.
remote: Compressing objects: 100% (102/102), done.
remote: Total 170 (delta 50), reused 166 (delta 49), pack-reused 0 (from 0)
Receiving objects: 100% (170/170), 442.49 KiB | 4.25 MiB/s, done.
Resolving deltas: 100% (50/50), done.
PS C:\demos> |
```



# Install NPM Packages

```
Windows PowerShell
PS C:\demos> cd .\sci-fi-cinema-24\
PS C:\demos\sci-fi-cinema-24> npm install
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated @humanwhocodes/config-array@0.13.0: Use @eslint/config-array instead
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead
npm WARN deprecated eslint@8.57.1: This version is no longer supported. Please see https://eslint.org/version-support for other options.

added 376 packages, and audited 377 packages in 25s

137 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\demos\sci-fi-cinema-24> |
```

# Start branch

- Start with the **00-start** branch
  - `git checkout --track origin/00-start`

# Start the application

```
next-server (v14.2.16) x + v
PS C:\demos\sci-fi-cinema-24> npm run dev

> sci-fi-cinema-24@0.1.0 dev
> next dev

  ▲ Next.js 14.2.16
  - Local:      http://localhost:3000

✓ Starting...
✓ Ready in 1626ms
○ Compiling / ...
✓ Compiled / in 2.6s (546 modules)
GET / 200 in 2865ms
✓ Compiled in 225ms (254 modules)
|
```

# The application



## NEXT.JS

1. Get started by editing `src/app/page.tsx`.
2. Save and see your changes instantly.

▲ Deploy now

Read our docs

 Learn  Examples  Go to [nextjs.org](https://nextjs.org) →



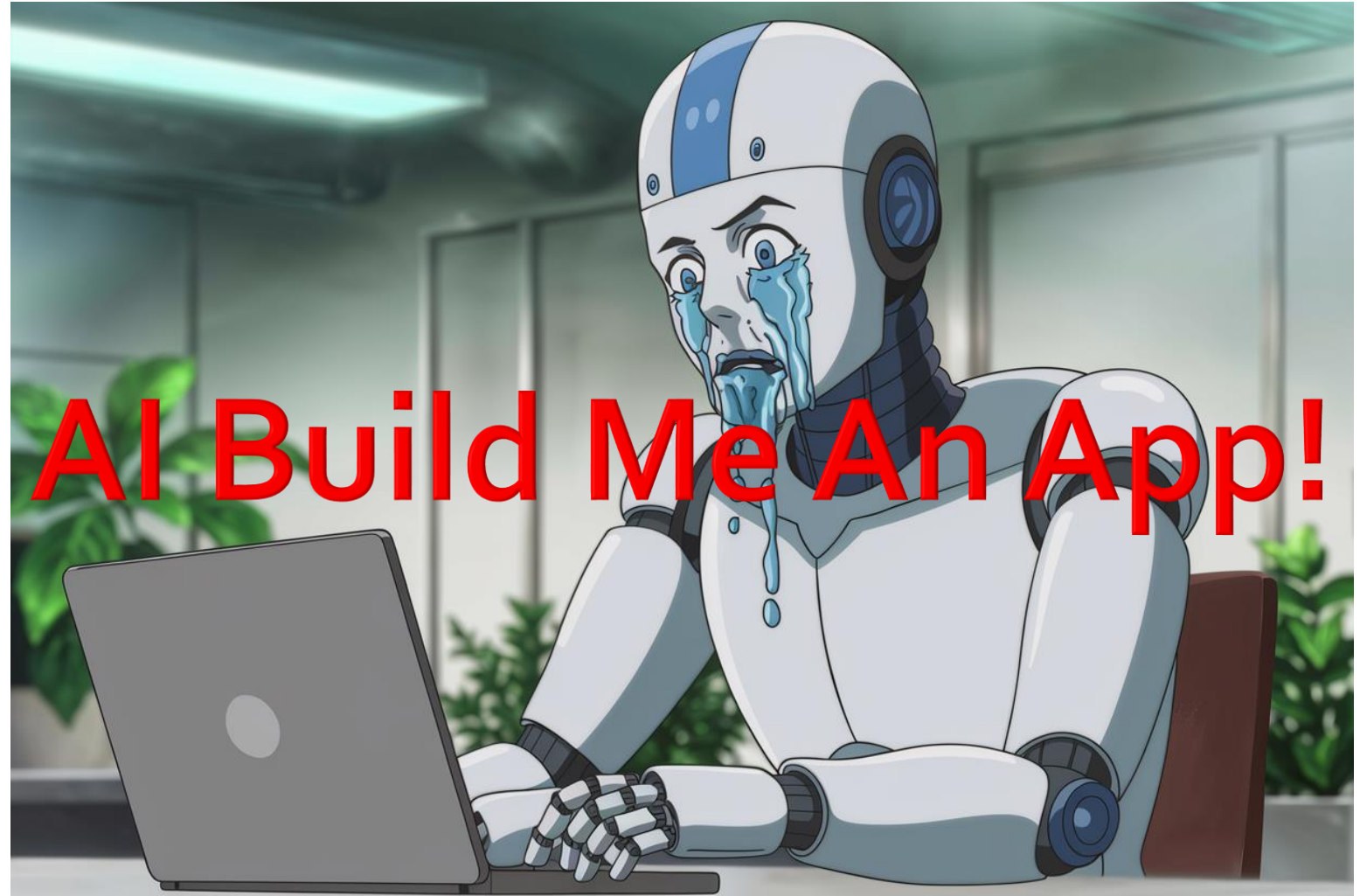
# What is AI-Driven App Development?

# What is AI-Driven App Development?

- **Leveraging AI capabilities** to streamline and enhance the app development process
- Focus on **creativity, innovation, and higher-level problem-solving** over manual coding



# What is AI-Driven App Development?



# The Cursor IDE

- **AI-enhanced code editor** that integrates seamlessly with existing development workflows
- **Key capabilities:**
  - Code prediction
  - Real-time suggestions
  - Architectural comprehension

# Alternatives

- Visual Studio Code plug-ins
  - [GitHub Copilot](#)
  - [Cline](#)
- Editors
  - [Aider](#)
  - [Bolt](#)
  - [Windsurf Editor](#)

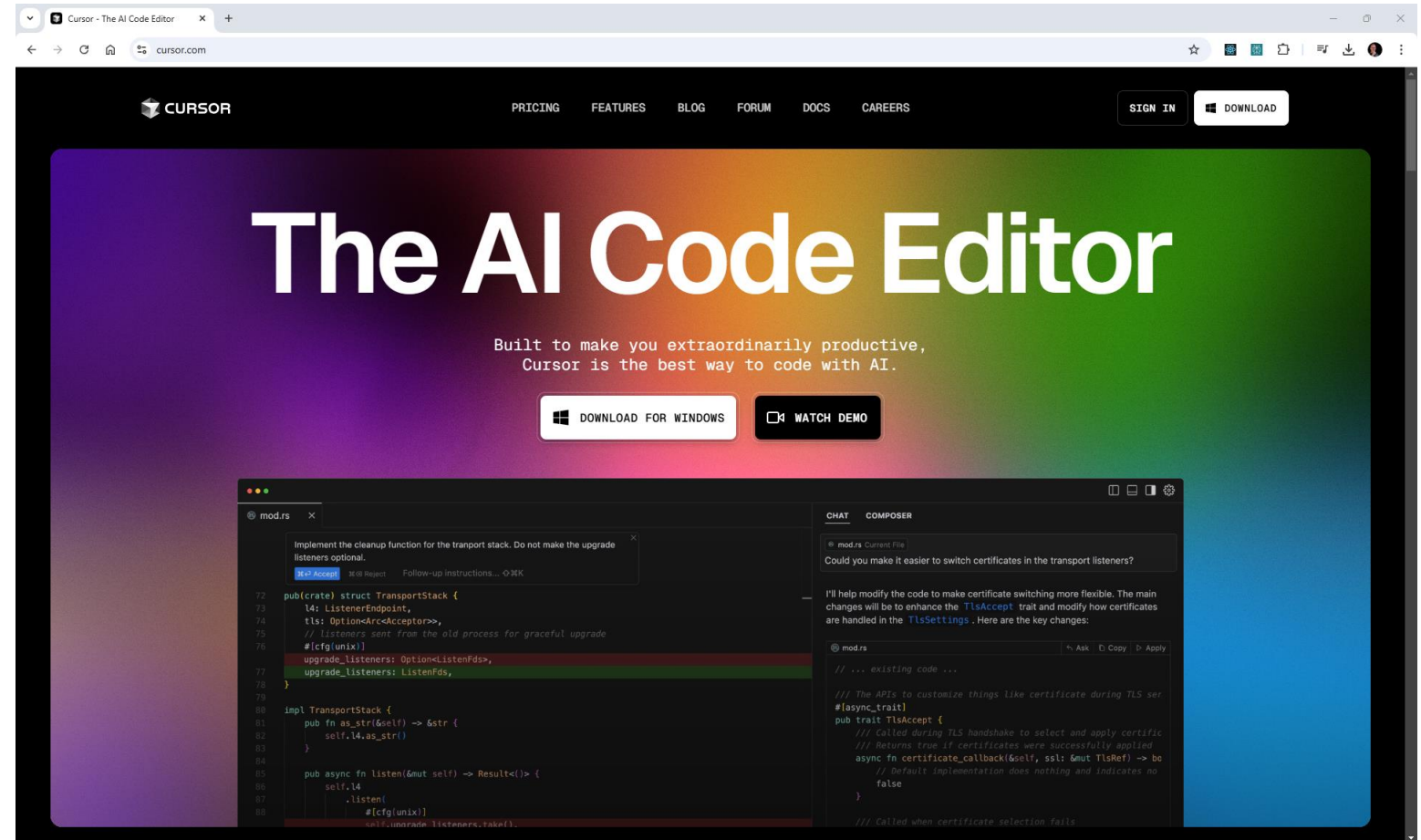
# Benefits of AI-Driven App Development

- **Reduced coding effort** and repetitive tasks
- **Faster prototyping** and iteration
- Improved **developer productivity** and creativity
- Consistent, **high-quality code** generation



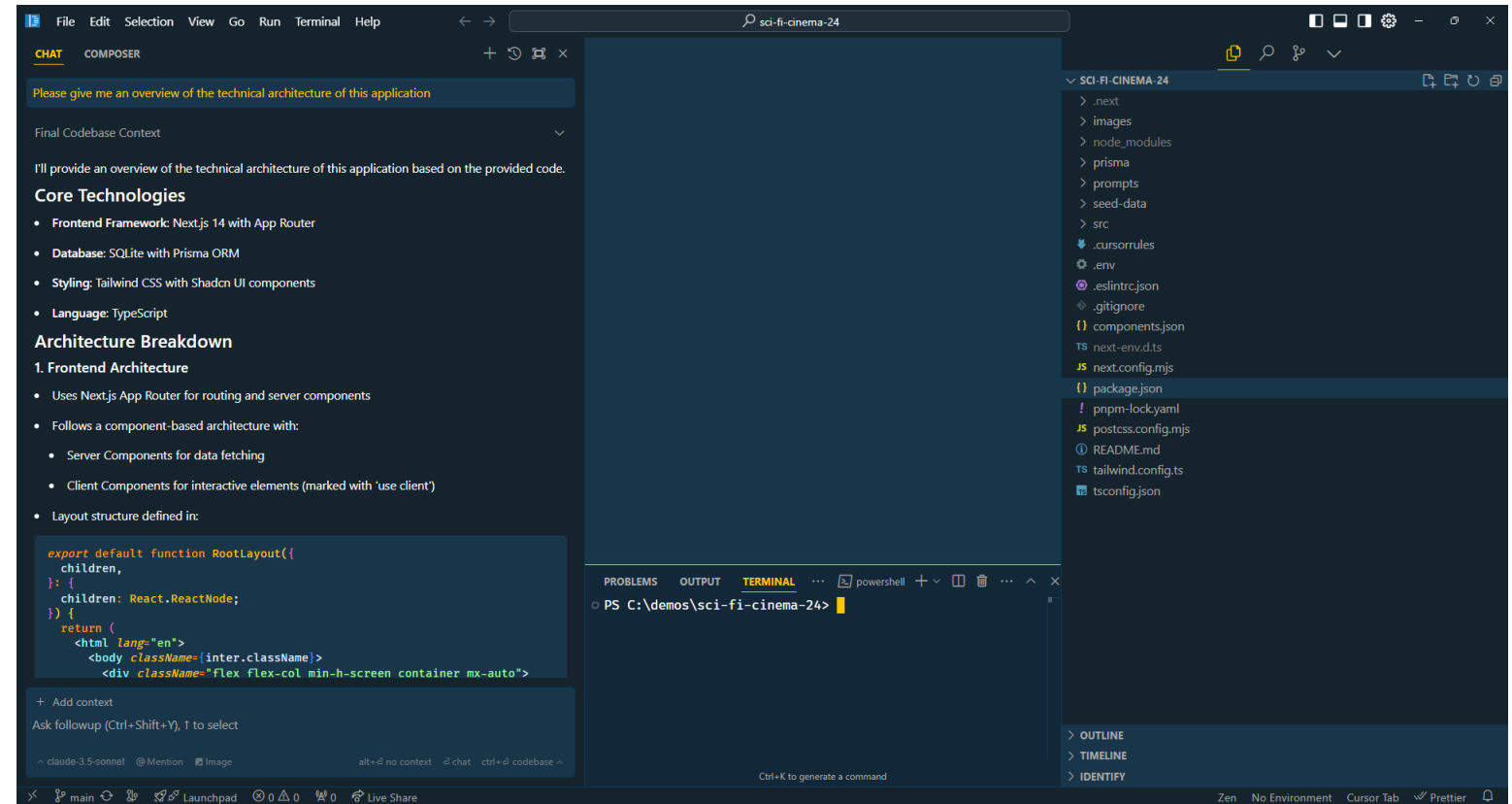
# Installing Cursor IDE

# Installing Cursor IDE





# Installing Cursor IDE



# Installing Cursor IDE



**CHAT** COMPOSER

① README.md File

Update the @README.md with this information

Here's an updated version of the README.md that incorporates the setup instructions while following standard README best practices:

① README.md ↶ Ask ↷ Copy ↵ Apply

```
# Movie Database Application

A modern web application built with Next.js 14, Prisma

## Tech Stack

- **Framework:** Next.js 14 (App Router)
- **Database:** SQLite with Prisma ORM
- **Styling:** Tailwind CSS + Shadcn UI
- **Language:** TypeScript

## Prerequisites

- Node.js 18+
- npm or pnpm

## Getting Started

1. Clone the repository:
```bash
git clone [repository-url]
cd [project-name]
```

2. Install dependencies:
```bash
npm install
# or
pnpm install
```
```

+ Add context

Ask followup (Ctrl+Shift+Y), 1 to select

~ claude-3.5-sonnet @ Mention Image ↶ chat ↵ ctrl+⇧ codebase ~

① README.md M x

```
# Movie Database Application
## Getting Started
git clone [repository-url]
cd [project-name]
```

2. Install dependencies:

```bash
npm install
# or
pnpm install
```

3. Initialize Shadcn UI:

```bash
npx shadcn@latest init --src-dir
```
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS GITLENS ...

PS C:\demos\sci-fi-cinema-24>

Message (Ctrl+⇧ to commit on "main")

✓ Commit

Changes

① README.md ↶ ↷ + M

SOURCE CONTROL GRAPH

GITLENS

Ctrl+K to generate a command

Zen Ln 17, Col 1 Spaces: 4 UTF-8 CRLF () Markdown Cursor Tab Prettier

# Adding a landing page

With <https://v0.dev/>

# Adding a landing page

- Vercel <https://v0.dev/> is great at generating more complex UI's
  - Has a preference for Tailwind and Shadcn components

# The prompt

*Can you create me an appealing landing page for an application to view popular science fiction movies.*

*Include the following:*

- *Header with a nav menu*
- *Hero section with an image*
- *Section where the user can choose the wanted subscription with three appropriate prices.*
- *Make the middle price stand out as most popular*
- *Footer with info that the movie data comes from <https://www.themoviedb.org/>*
- *Use a blue theme*

# Adding a landing page



```
landing-page.tsx
src > components > landing-page.tsx > ...
18 export function LandingPageComponent() {
19   return (
20     <div className="flex flex-col min-h-screen ■bg-blue-50">
21       <header className="px-4 lg:px-6 h-14 flex items-center □bg-blue-900 ■text-white">
22         <Link className="flex items-center justify-center href="#">
23           <Film className="h-6 w-6" />
24           <span className="ml-2 text-lg font-bold">SciFi Flix</span>
25         </Link>
26       <nav className="ml-auto flex gap-4 sm:gap-6">...
27     </nav>
28   </header>
29   <main className="flex-1">
30     <section className="w-full py-12 md:py-24 lg:py-32 xl:py-48 □bg-blue-900 ■text-white">
31       <div className="container px-4 md:px-6">
32         <div className="grid grid-cols-1 lg:grid-cols-2 gap-6 items-center">
33           <div className="flex flex-col items-center lg:items-start space-y-4 text-center lg:
34             <div className="space-y-2">
35               <h1 className="text-3xl font-bold tracking-tighter sm:text-4xl md:text-5xl lg:t
36                 Explore the Universe of Sci-Fi Cinema
37               </h1>
38               <p className="mx-auto max-w-[700px] ■text-blue-200 md:text-xl">
39                 Dive into a vast collection of popular science fiction
40                 movies. From classic space operas to mind-bending time
41                 travels, we&apos;ve got it all.
42               </p>
43             </div>
44             <div className="space-x-4">
45               <Button className="■bg-white □text-blue-900 ■hover:bg-blue-100">
```





# The Cursor AI Tools

# The Cursor AI Tools/Panels

- Cursor **Tab**
- Cursor **Prompt Bar** (Command K)
- Cursor **Chat** (Command L)
- Cursor **Composer** (Command I)

# Cursor Tab

- **Cursor Tab** enhances coding with **AI-powered autocomplete**
  - Like GitHub Copilot but multi line, code changes and predictions
- **Key Features:**
  - Predicts your next edit, allowing you to breeze through changes
  - Lets you tab through suggested code completions
  - Provides intelligent autocomplete functionality
- **When to Use:**
  - During active coding sessions
  - When you want to speed up your coding process
  - For quick code completions and predictions

# Cursor Prompt Bar

- The **Prompt Bar** is for **quick AI-assisted code generation and editing**
- **Key Features:**
  - Allows inline code generation and editing
  - Provides context-aware suggestions
  - Supports follow-up instructions for refinement
  - Enables quick questions about selected code
- **When to Use:**
  - For rapid code generation without switching context
  - To make AI-assisted edits to existing code
  - When you need quick answers about specific code sections
  - To generate terminal commands in plain English

# Cursor Chat

- The **Chat panel** in Cursor serves as an **AI-powered coding assistant**
- **Key Features:**
  - Allows you to get answers from your codebase
  - Enables you to refer to files or documentation
  - Offers the ability to use the model's code suggestions with one click
- **When to Use:**
  - When you need clarification on your codebase
  - To quickly access relevant documentation
  - For getting AI-generated code suggestions

# Cursor Composer

- The **Composer panel** in Cursor empowers you to **write and modify code using natural language instructions**
- **Key Features:**
  - Lets you write code using simple prompts or instructions
  - Enables updating entire classes or functions with a single prompt
  - Understands your codebase context
- **When to Use:**
  - When you want to make broad changes to your code
  - For generating new code sections based on descriptions
  - To refactor or update existing code structures quickly



## Pro tip

- **Break large tasks up into smaller ones**
  - Just like you divide an epic into more manageable user stories
- **Tell Cursor to ask you questions about what to do**
  - Just like you would during a refinement
- **Create frequent Git commits**
  - Makes it easy to go back to a previous state
- **Consider adding Playwright E2E test**
  - Sometimes the AI agent makes unintended changes



# Adding Prisma



# The prompt

*Add a SQLite database with the Prisma ORM*  
*Create the required tables to store the data in*  
*Seed the database with the data in @seed-data*

# Adding Prisma

```
schema.prisma  TS prisma.ts  TS seed.ts
prisma > schema.prisma > ...
4  generator client {
5    provider = "prisma-client-js"
6  }
7
You, 3 days ago | 1 author (You)
8  datasource db {
9    provider = "sqlite"
10   url       = env("DATABASE_URL")
11 }
12
You, 3 days ago | 1 author (You)
13 model Movie {
14   id           Int      @id
15   backdropPath String?
16   overview     String
17   popularity   Float
18   posterPath   String?
19   releaseDate  DateTime
20   title        String
21   voteAverage  Float
22   voteCount    Int
23   genres       Genre[]
24   actors       Actor[]
25   directors    Director[]
26 }
27
You, 3 days ago | 1 author (You)
28 model Actor {
```

# Adding Prisma

```
schema.prisma  TS prisma.ts  TS seed.ts
src > lib > TS prisma.ts > ...
You, 3 days ago | 1 author (You)
1  import { PrismaClient } from '@prisma/client';
2
3  const globalForPrisma = global as unknown as { prisma: PrismaClient };
4
5  export const prisma = globalForPrisma.prisma || new PrismaClient();
6
7  if (process.env.NODE_ENV !== 'production') globalForPrisma.prisma = prisma;
```

# Adding Prisma



```
schema.prisma  TS prisma.ts  TS seed.ts  X
prisma > TS seed.ts > ...
37  async function main() {
89      update: {},
90      create: {
91          id: genre.id,
92          name: genre.name,
93      },
94  });
95  }
96
97  // Create movies with relationships
98  console.log('Seeding movies ... ');
99  for (const movie of moviesData) {
100      await prisma.movie.upsert({
101          where: { id: movie.id },
102          update: {},
103          create: {
104              id: movie.id,
105              backdropPath: movie.backdropPath,
106              overview: movie.overview,
107              popularity: movie.popularity,
108              posterPath: movie.posterPath,
109              releaseDate: new Date(movie.releaseDate),
110              title: movie.title,
111              voteAverage: movie.voteAverage,
112              voteCount: movie.voteCount,
113              actors: {
114                  connect: movie.actorIds.map((id: number) => ({ id })),
```

10-minute break

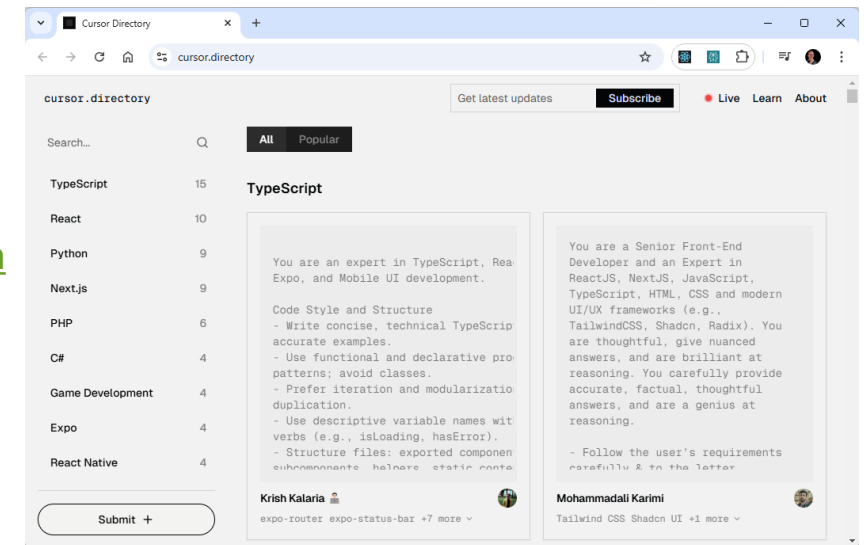




# Using Rules for AI & .cursorrules files

# Using Rules for AI

- Rules for AI instructions are **standard instructions** to the AI agent
  - On a per user basis
- Add **project-specific instructions** to a `.cursorrules` file
  - Committed in GitHub
  - See:
    - <https://cursor.directory>
    - <https://cursorlist.com>
    - <https://dotcursorrules.com>



# The prompt

*Add a section about using Prisma and using Prisma best practices with Next.js and SQLite*  
*Also add the fact that we are using the Next.js `./src` folder*  
*@Prisma @NextJS*



# Using Rules for AI



```
.cursorrules M x
.cursorrules
1 You are an expert in TypeScript, Node.js, Next.js App Router, React, Shadcn UI, Radix UI and
  Tailwind.
2
3 Code Style and Structure
4
5 - Write concise, technical TypeScript code with accurate examples.
6 - Use functional and declarative programming patterns; avoid classes.
7 - Prefer iteration and modularization over code duplication.
8 - Use descriptive variable names with auxiliary verbs (e.g., isLoading, hasError).
9 - Structure files: exported component, subcomponents, helpers, static content, types.
10 - Use the ./src directory for all source code.
11
12 Naming Conventions
13
14 - Use lowercase with dashes for directories (e.g., components/auth-wizard).
15 - Favor named exports for components.
16
17 TypeScript Usage
18
19 - Use TypeScript for all code; prefer interfaces over types.
20 - Avoid enums; use maps instead.
21 - Use functional components with TypeScript interfaces.
22
23 Syntax and Formatting
24
25 - Use the "function" keyword for pure functions.
26 - Avoid unnecessary curly braces in conditionals; use concise syntax for simple statements.
```



# Movies List Page

# The prompt

*Add a new movies list route `/sci-fi-movies`.*

*Display a list of all science fiction movies from the database.*

*Ask any questions you might have.*

# Movies List Page

```
page.tsx x landing-page.tsx M JS next.config.mjs M
src > app > sci-fi-movies > page.tsx > ...
4  async function getSciFiMovies() {
5    const movies = await prisma.movie.findMany({
6      where: {
7        genres: {
8          some: {
9            id: 878, // Science Fiction genre ID
10          },
11        },
12      },
13      include: {
14        genres: true,
15        directors: true,
16      },
17      orderBy: {
18        voteAverage: 'desc',
19      },
20    });
21    return movies;
22  }
23
24  export default async function SciFiMoviesPage() {
25    const movies = await getSciFiMovies();
26
27    return (
28      <main className="py-8">
29        <h1 className="text-4xl font-bold mb-8">Science Fiction Movies</h1>
30        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6">
```

# Movies List Page

```
page.tsx U  landing-page.tsx M X  JS next.config.mjs M
src > components > landing-page.tsx > ...
18 export function LandingPageComponent() {
68   <div className="space-x-4">
69     <Link href="/sci-fi-movies">
70       <Button className="bg-white text-blue-900 hover:bg-blue-100">
71         Get Started
72       </Button>
73     </Link>
```

Error:  
Invalid src prop



```
page.tsx U  landing-page.tsx M  JS next.config.mjs M X
JS next.config.mjs > ...
You, 14 minutes ago | 1 author (You)
1  /** @type {import('next').NextConfig} */
2  const nextConfig = {
3    images: {
4      remotePatterns: [
5        {
6          protocol: 'https',
7          hostname: 'image.tmdb.org',
8          pathname: '/t/p/**',
9        },
10     ],
11   },
12 };

```



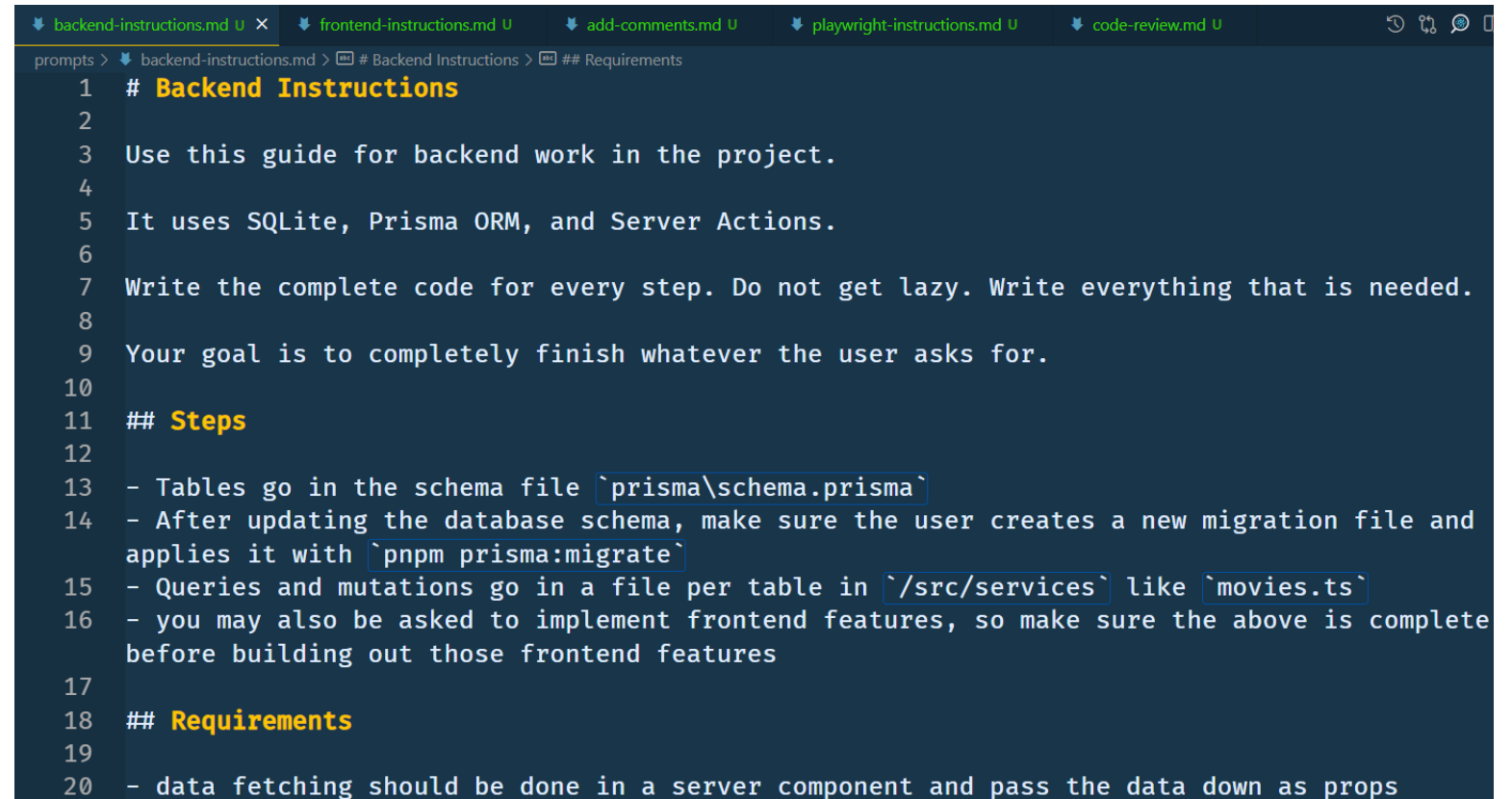
# Composer Notepads

# Composer Notepads

- **Use Notepads to manage context** for specific areas of your project:
  - Create separate notepads for different components or features
  - Tag relevant files within each notepad for easy reference
- **Reference Notepads** directly in chat or Composer prompts to provide context for your queries or code generation tasks
- Using **Prompt files is an alternative**
  - Shared between developers through GitHub



# Composer Notepads and Prompt files



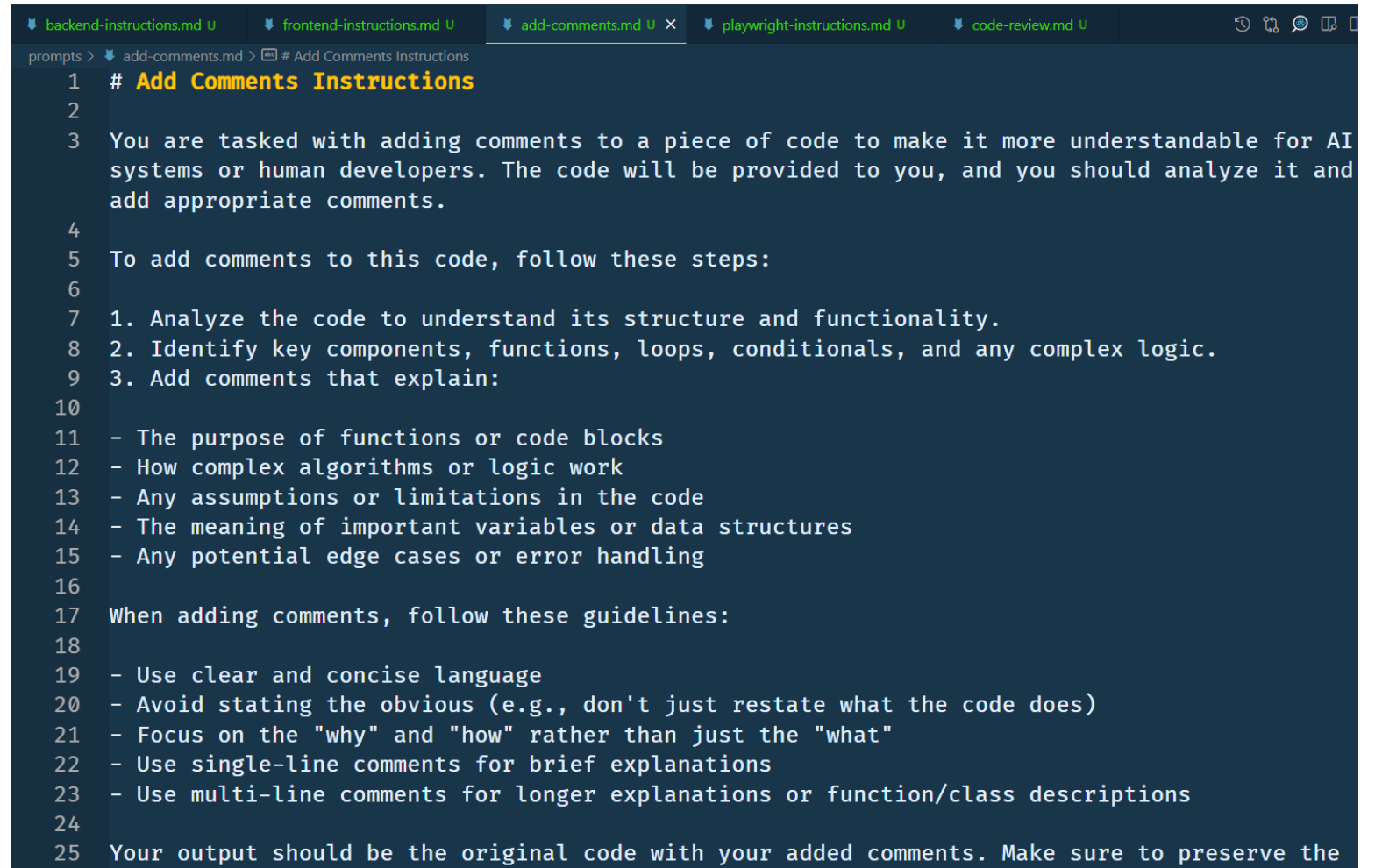
The screenshot shows a code editor with several tabs at the top: 'backend-instructions.md', 'frontend-instructions.md', 'add-comments.md', 'playwright-instructions.md', and 'code-review.md'. The active tab is 'backend-instructions.md'. The editor content shows a prompt for a backend developer, starting with a title '# Backend Instructions' and followed by several paragraphs of instructions. The instructions cover the use of SQLite, Prisma ORM, and Server Actions, and emphasize writing complete code for every step. A section titled '## Steps' lists specific tasks: creating a schema file, updating the database schema, and creating migration files. Another section titled '## Requirements' lists a requirement for data fetching in a server component.

```
prompts > backend-instructions.md > # Backend Instructions > ## Requirements
1  # Backend Instructions
2
3  Use this guide for backend work in the project.
4
5  It uses SQLite, Prisma ORM, and Server Actions.
6
7  Write the complete code for every step. Do not get lazy. Write everything that is needed.
8
9  Your goal is to completely finish whatever the user asks for.
10
11 ## Steps
12
13 - Tables go in the schema file `prisma/schema.prisma`
14 - After updating the database schema, make sure the user creates a new migration file and
  applies it with `pnpm prisma:migrate`
15 - Queries and mutations go in a file per table in `/src/services` like `movies.ts`
16 - you may also be asked to implement frontend features, so make sure the above is complete
  before building out those frontend features
17
18 ## Requirements
19
20 - data fetching should be done in a server component and pass the data down as props
```

# Composer Notepads and Prompt files

```
backend-instructions.md U frontend-instructions.md U X add-comments.md U playwright-instructions.md U code-review.md U
prompts > frontend-instructions.md > # Frontend Instructions > ## Requirements
1  # Frontend Instructions
2
3  Use this guide for frontend work in the project.
4
5  It uses Next.js, Tailwind, and Shadcn.
6
7  Write the complete code for every step. Do not get lazy. Write everything that is needed.
8
9  Your goal is to completely finish whatever the user asks for.
10
11 ## Steps
12
13 - All new components should go in `~/src/components` and be named like `example-component.tsx`
  unless otherwise specified
14 - All new pages go in `~/src/app`
15
16 ## Reminders
17
18 - We use the Next.js app router
19 - Try and group components into folders
20
21 ## Requirements
22
23 - All data fetching should be done in a server component and pass the data down as props
24 - Client components (useState, hooks, etc) require that 'use client' is set at the top of the
  file
25 - useRouter should be imported from next/navigation
```

# Composer Notepads and Prompt files



The screenshot shows a code editor with a dark theme. At the top, there are several tabs: 'backend-instructions.md', 'frontend-instructions.md', 'add-comments.md' (which is active and has a close button), 'playwright-instructions.md', and 'code-review.md'. Below the tabs, the editor displays the content of 'add-comments.md'. The text is as follows:


```
prompts > add-comments.md > # Add Comments Instructions
1  # Add Comments Instructions
2
3  You are tasked with adding comments to a piece of code to make it more understandable for AI
4  systems or human developers. The code will be provided to you, and you should analyze it and
5  add appropriate comments.
6
7  To add comments to this code, follow these steps:
8
9  1. Analyze the code to understand its structure and functionality.
10 2. Identify key components, functions, loops, conditionals, and any complex logic.
11 3. Add comments that explain:
12
13 - The purpose of functions or code blocks
14 - How complex algorithms or logic work
15 - Any assumptions or limitations in the code
16 - The meaning of important variables or data structures
17 - Any potential edge cases or error handling
18
19 When adding comments, follow these guidelines:
20
21 - Use clear and concise language
22 - Avoid stating the obvious (e.g., don't just restate what the code does)
23 - Focus on the "why" and "how" rather than just the "what"
24 - Use single-line comments for brief explanations
25 - Use multi-line comments for longer explanations or function/class descriptions
26
27 Your output should be the original code with your added comments. Make sure to preserve the
```

# Composer Notepads and Present files



```
page.tsx M x
src > app > sci-fi-movies > page.tsx > ...

4  /**
5   * Fetches science fiction movies from the database
6   * Uses TMDB genre ID 878 to identify sci-fi movies
7   * Returns movies sorted by vote average (highest rated first)
8   * Includes related genre and director data for each movie
9   */
10 > async function getSciFiMovies() {...
28 }
29
30 /**
31 * Server Component that displays a responsive grid of sci-fi movies
32 * Features:
33 * - Responsive grid layout (1-4 columns based on screen size)
34 * - Movie cards with poster images, ratings, and details
35 * - Hover animations for interactive feedback
36 * - Fallback for missing poster images
37 */
38 export default async function SciFiMoviesPage() {
39   const movies = await getSciFiMovies();
40
41   return (
42     <main className="py-8">
43       <h1 className="text-4xl font-bold mb-8">Science Fiction Movies</h1>
44       {/* Responsive grid layout using Tailwind's grid system */}
45       <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6">
46         {movies.map((movie) => (
47           // Movie card with hover animation
```



# Refactoring with a prompt file

# The prompt

*Refactor this file and move the `getSciFiMovies()` function to the proper location.*

*@backend-instructions.md*

# Refactoring

```
page.tsx M x TS movies.ts U
src > app > sci-fi-movies > page.tsx > ...
1 import { getSciFiMovies } from '@services/movies';
2 import Image from 'next/image';
3
4 /**
5  * Server Component that displays a responsive grid of sci-fi movies
6  * Features:
7  * - Responsive grid layout (1-4 columns based on screen size)
8  * - Movie cards with poster images, ratings, and details
9  * - Hover animations for interactive feedback
10 * - Fallback for missing poster images
11 */
12 export default async function SciFiMoviesPage() {
13   const movies = await getSciFiMovies();
14
15   return (
16     <main className="py-8">
17       <h1 className="text-4xl font-bold mb-8">Science Fiction Movies</h1>
18       { /* Responsive grid layout using Tailwind's grid system */ }
```

# Refactoring



```
page.tsx M TS movies.ts U X
src > services > TS movies.ts > ...
1 import { prisma } from '@lib/prisma';
2
3 /**
4  * Fetches science fiction movies from the database
5  * Uses TMDB genre ID 878 to identify sci-fi movies
6  * Returns movies sorted by vote average (highest rated first)
7  * Includes related genre and director data for each movie
8  */
9 export async function getSciFiMovies() {
10   const movies = await prisma.movie.findMany({
11     where: {
12       genres: {
13         some: {
14           id: 878, // Science Fiction genre ID from TMDB
15         },
16       },
17     },
18     include: {
19       genres: true,
20       directors: true,
21     },
22     orderBy: {
23       voteAverage: 'desc',
24     },
25   });
26   return movies;
27 }
```



10-minute break





# Responsive Grid & Movie cards

# The prompt

*Update the movie card to look like this image*

*The 'Read More' should only be visible when the mouse hovers over the card.*

*Also move the movie card to a separate component and use the Shadcn Card component*

*Ask any questions you might have.*

*@frontend-instructions.md*

The second  
prompt as it  
was using  
useState()

*Make this a RSC and use @Tailwind CSS to do the hover instead of the component using React state*

# Responsive Grid & Movie cards

```
page.tsx M x movie-card.tsx U
src > app > sci-fi-movies > page.tsx > ...
You, 7 minutes ago | 1 author (You)
1 import { getSciFiMovies } from '@services/movies';
2 import { MovieCard } from '@components/movies/movie-card';
3
4 /**
5  * Server Component that displays a responsive grid of sci-fi movies
6  * Features:
7  * - Responsive grid layout (1-4 columns based on screen size)
8  * - Movie cards with poster images and ratings
9  * - Hover animations with Read More button
10  */
11 export default async function SciFiMoviesPage() {
12   const movies = await getSciFiMovies();
13
14   return (
15     <main className="py-8">
16       <h1 className="text-4xl font-bold mb-8">Science Fiction Movies</h1>
17       <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6">
18         {movies.map((movie) => (
19           <MovieCard key={movie.id} movie={movie} />
20         ))}
21       </div>
22     </main>
23   );
```

# Responsive Grid & Movie cards



```
src > components > movies > movie-card.tsx > ...
9   export function MovieCard({ movie }: MovieCardProps) {
10     .
29   <div className="w-full h-[450px] ■bg-gray-200 flex items-center justify-center">
30     No Image Available
31   </div>
32   }
33
34   { /* Movie Info Overlay */ }
35   <CardContent className="absolute bottom-0 left-0 right-0 p-4 ■text-white z-20">
36     <h2 className="text-xl font-bold mb-2">{movie.title}</h2>
37     <div className="flex items-center gap-2 ■text-yellow-400">
38       <span>★</span>
39       <span>{movie.voteAverage.toFixed(1)}/10</span>
40     </div>
41   </CardContent>
42
43   { /* Read More Button */ }
44   <CardFooter
45     className="absolute inset-0 flex items-center justify-center transition-opacity duration
46     opacity-0 group-hover:opacity-100"
47   >
48     <button className="■bg-red-600 ■text-white px-6 py-2 rounded-md font-semibold ■hover
49       READ MORE
50     </button>
51   </CardFooter>
52 </Card>
53   );
```



# Navigation Menu Refactor

# The prompt

*Move the header and footer into separate components and update the usage so they are visible on all pages.*

*Also add the route to the sci-fi movies to the navigation bar in the header*

*@frontend-instructions.md*



# Navigation Menu Refactor

```
landing-page.tsx | header.tsx | footer.tsx | layout.tsx
src > components > landing-page.tsx > ...
18 export function LandingPageComponent() {
19   return (
20     <div className="flex flex-col ■bg-blue-50">
21       <section className="w-full py-12 md:py-24 lg:py-32 xl:py-48 □bg-blue-900 ■text-white">
22         <div className="container px-4 md:px-6">
23           <div className="grid grid-cols-1 lg:grid-cols-2 gap-6 items-center">
24             <div className="flex flex-col items-center lg:items-start space-y-4 text-center lg:
25               <div className="space-y-2">
26                 <h1 className="text-3xl font-bold tracking-tighter sm:text-4xl md:text-5xl lg:
27                   Explore the Universe of Sci-Fi Cinema
28                 </h1>
29                 <p className="mx-auto max-w-[700px] ■text-blue-200 md:text-xl">
30                   Dive into a vast collection of popular science fiction movies.
31                   From classic space operas to mind-bending time travels,
32                   we&apos;ve got it all.
33                 </p>
34             </div>
35             <div className="space-x-4">
36               <Link href="/sci-fi-movies">
37                 <Button className="■bg-white □text-blue-900 ■hover:bg-blue-100">
38                   Get Started
39                 </Button>
40               </Link>
41               <Button
42                 variant="outline"
43                 className="■text-white ■border-white ■hover:bg-white □hover:text-blue-90
44             >
```

# Navigation Menu Refactor

```
landing-page.tsx | header.tsx | footer.tsx | layout.tsx
src > components > layout > header.tsx > ...

3 import Link from 'next/link';
4 import { Film } from 'lucide-react';
5
6 export function Header() {
7   return (
8     <header className="px-4 lg:px-6 h-14 flex items-center bg-blue-900 text-white">
9       <Link className="flex items-center justify-center" href="/">
10         <Film className="h-6 w-6" />
11         <span className="ml-2 text-lg font-bold">SciFi Flix</span>
12       </Link>
13       <nav className="ml-auto flex gap-4 sm:gap-6">
14         <Link
15           className="text-sm font-medium hover:underline underline-offset-4"
16           href="/sci-fi-movies"
17         >
18           Movies
19         </Link>
20         <Link
21           className="text-sm font-medium hover:underline underline-offset-4"
22           href="#features"
23         >
24           Features
25         </Link>
26         <Link
27           className="text-sm font-medium hover:underline underline-offset-4"
28           href="#pricing"
29         >
```

# Navigation Menu Refactor

```
landing-page.tsx  header.tsx  footer.tsx  layout.tsx
src > components > layout > footer.tsx > ...
1  import Link from 'next/link';
2
3  export function Footer() {
4    return (
5      <footer className="flex flex-col gap-2 sm:flex-row py-6 w-full shrink-0 items-center">
6        <p className="text-xs text-blue-200">
7          © 2023 SciFi Flix. All rights reserved. Movie data provided by{' '}</p>
8        <a
9          href="https://www.themoviedb.org/"
10         className="hover:underline underline-offset-2"
11         target="_blank"
12         rel="noopener noreferrer"
13       >
14         The Movie Database (TMDB)
15     </a>
16     .
17   </p>
18   <nav className="sm:ml-auto flex gap-4 sm:gap-6">
19     <Link className="text-xs hover:underline underline-offset-4" href="#">
20       Terms of Service
21     </Link>
22     <Link className="text-xs hover:underline underline-offset-4" href="#">
23       Privacy
24     </Link>
25   </nav>
26 </footer>
```

# Navigation Menu Refactor



```
landing-page.tsx  header.tsx  footer.tsx  layout.tsx  X
src > app > layout.tsx > ...

14  export default function RootLayout({
15    children,
16  }: {
17    children: React.ReactNode;
18  }) {
19    return (
20      <html lang="en">
21        <body className={inter.className}>
22          <div className="flex flex-col min-h-screen">
23            <Header />
24            <main className="flex-1">{children}</main>
25            <Footer />
26          </div>
27        </body>
28      </html>
29    );
30  }
```

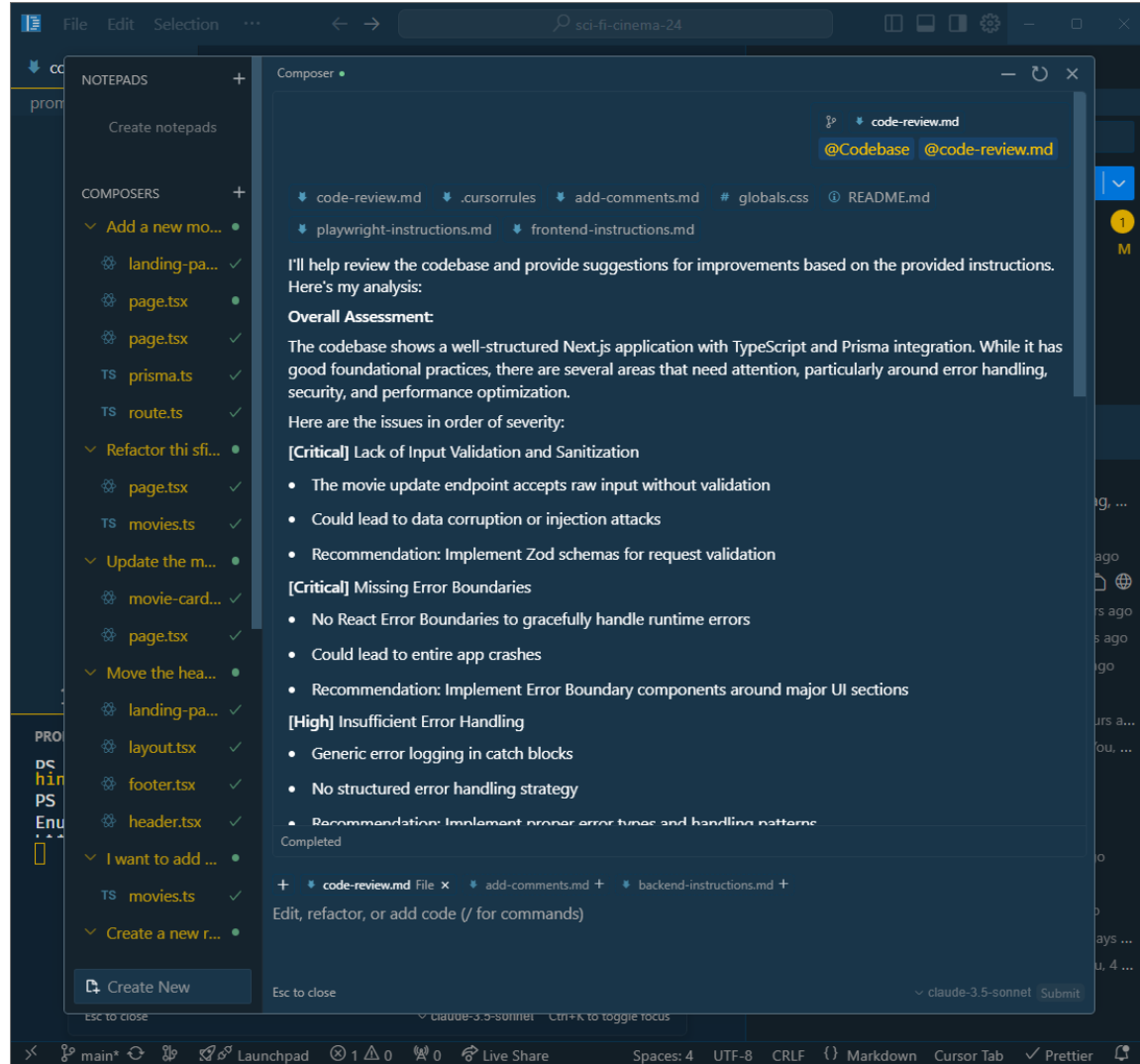


# Code Reviews

# Code Reviews

- **Review Diff's** with Main Branch, Commit, Working State or Codebase
- Use a **Prompt File/Notepad** like `code-review.md`

# Code Reviews



10-minute break







# Movie Details Page

# The prompt

*I want to add a route to inspect all the details available in the database for a single movie.*

*Create a query I can use to load the object in page I will create next.*

*@backend-instructions.md*

# Movie Details Page Backend



```
TS movies.ts M x schema.prisma
src > services > TS movies.ts > ...

30 export async function getMovieById(id: number) {
31   try {
32     const movie = await prisma.movie.findUnique({
33       where: {
34         id: id,
35       },
36       include: {
37         genres: true,
38         directors: true,
39         actors: true,
40       },
41     });
42
43     if (!movie) {
44       throw new Error('Movie not found');
45     }
46
47     return movie;
48   } catch (error) {
49     console.error('Error fetching movie:', error);
50     throw error;
51   }
52 }
53
54 // Type for the returned data
55 export type MovieWithDetails = Prisma.PromiseReturnType<typeof getMovieById>;
```

# The prompt

*Create a new route to inspect all the details available in the database for a single movie. Use the `getMovieById` function to retrieve all the movie data.*

*Update the @movie-card.tsx to link to this route from Read More*

*Ask any questions you might have.*

*@frontend-instructions.md*

# Movie Details Page Frontend

```
page.tsx U x movie-card.tsx M
src > app > movies > [id] > page.tsx > ...
7 interface MoviePageProps {
8   params: {
9     id: string;
10  };
11 }
12
13 export default async function MoviePage({ params }: MoviePageProps) {
14   try {
15     const movie = await getMovieById(parseInt(params.id));
16
17     return (
18       <main className="container mx-auto py-8 px-4">
19         <div className="grid md:grid-cols-[350px_1fr] gap-8">
20           { /* Movie Poster */ }
21           <div className="relative h-[525px]">
22             {movie.posterPath ? (
23               <Image
24                 src={`https://image.tmdb.org/t/p/w500${movie.posterPath}`}
25                 alt={movie.title}
26                 fill
27                 className="object-cover rounded-lg"
28                 priority
29               />
30             ) : (
31               <div className="w-full h-full bg-gray-200 rounded-lg flex items-center"
32                 No Image Available
33               </div>
34             )}
35         </div>
36       </main>
37     );
38   } catch {
39     // Handle error
40   }
41 }
```

# Movie Details Page Frontend



```
page.tsx U  movie-card.tsx M X
src > components > movies > movie-card.tsx > ...
10  export function MovieCard({ movie }: MovieCardProps) {
44      /* Read More Button */
45      <CardFooter
46          className="absolute inset-0 flex items-center justify-center transition-
47              opacity-0 group-hover:opacity-100"
48      >
49          <Link href={`/${movies}/${movie.id}`}>
50              <button className="bg-red-600 text-white px-6 py-2 rounded-md font
51                  READ MORE
52              </button>
53          </Link>
54      </CardFooter>
55  </Card>
56  );
```



# Movie Edit Page

# The prompt

*Create a new route to edit the details for a movie available in the database. Add a React Server Action to do the update when the RSC form is submitted.*

*Ask any questions you might have.*

*@backend-instructions.md @frontend-instructions.md @Codebase*



# Movie Edit Page



SciFi Flix

localhost:3000/movies/157336/edit

SciFi Flix

Movies Features Pricing About Contact

### Edit Movie

Title

Interstellar

Overview

The adventures of a group of explorers who make use of a newly discovered wormhole to surpass the limitations on human space travel and conquer the vast distances involved in an interstellar voyage.

Release Date

05/11/2014

Genres

Action Adventure Animation Comedy Crime Documentary Drama Family Fantasy History Horror Music Mystery Romance Science Fiction TV Movie Thriller War Western


Cancel Save Changes

© 2023 SciFi Flix. All rights reserved. Movie data provided by The Movie Database (TMDB). Terms of Service Privacy

But there is more 😊

But there is  
more 😊

- Use AI to create **product requirements**
- Use AI to **estimate the effort**
- Use AI to design the **architecture**
- Use AI to create a **project plan**
- Use AI to write **user stories** and acceptance criteria
- Use AI to **write README** files
- Use AI to **draw diagrams**
  - 👉 Use Mermaid syntax to directly embed in GitHub Markdown
- Use AI to generate Playwright **end to end tests**
- Use AI to create ...



# Recommendations with AI-Driven App Development

# Recommendations

- The AI is a **junior software engineer** at best
  - Give it small focused steps for best results
- Give the AI as **much context** as possible
  - Using a `.cursorrules` and prompt files
- **Break larger steps up into small focused ones**
- Let Cursor know **it can ask you questions**
- **Commit to Git** after each small step you are happy with
- **Review the AI generated code** like a junior developer
  - But remember it doesn't learn from a chat conversation

# When not to use AI?

- Don't treat AI as a **silver bullet**
  - Some tasks are easier to just do manually
- Be careful with generating **critical code**
  - An AI might not consider all implications
- When dealing with **legacy system**
  - The AI is probably not aware about the system and will hallucinate

# Conclusion

- Using AI tools will make **software engineers more productive**
  - But won't replace them any time soon (if ever)
- **AI tools are like a junior software engineer** at the moment
  - It needs a lot of guidance and assistance

Thank you for joining

[Share your thoughts](#)

