AWS CLOUD CLUB · VIT CHENNAI

CineMood AI

Full Stack Project Submission — Web Development Domain

Submitted by: Gowreesh V T Date: 14 February 2026 Track: Full Stack

---

1 GitHub Repository Link

Repository

https://github.com/Gowreesh-VT/AWS-Web-Dev

2 Hosted Website Link

Live Deployment

https://aws-web-dev.vercel.app

3 Tech Stack Used

FRONTEND

- Next.js 15 (React 19, App Router)
- TypeScript
- Tailwind CSS 3
- Radix UI (Shadcn/ui components)
- Lucide React (Icons)

BACKEND

- Next.js API Routes (Route Handlers)
- MongoDB Atlas (Cloud Database)
- Mongoose ODM v9
- JWT Authentication (jsonwebtoken)
- bcryptjs (Password Hashing)

AI & APIS

- Google Gemini AI (Genkit)
- TMDB API (Movie Database)
- Mood-to-Genre AI Conversion

DEV TOOLS

- ESLint + TypeScript strict mode
- Turbopack (Dev Server)
- PostCSS

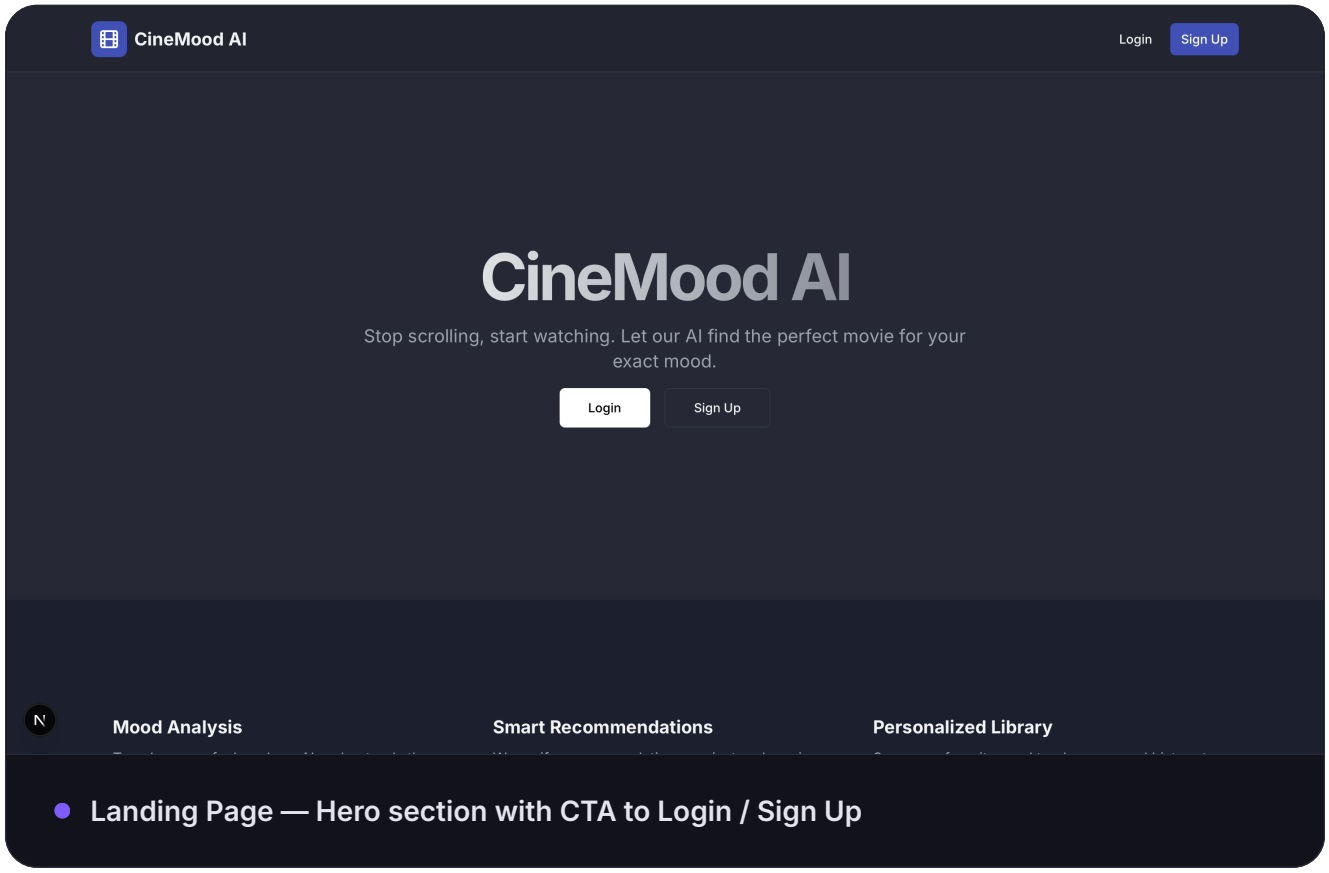
4 API Endpoints Description

METHOD	ENDPOINT	DESCRIPTION	AUTH
POST	/api/auth/signup	Register a new user with email, password, and name. Returns JWT token via HTTP-only cookie.	PUBLIC
POST	/api/auth/login	Authenticate user with email & password. Validates credentials via bcrypt and returns JWT cookie.	PUBLIC
POST	/api/auth/logout	Clears the JWT authentication cookie to log the user out.	PUBLIC
GET	/api/auth/me	Returns the currently authenticated user's profile (excluding password) by verifying the JWT token.	JWT
POST	/api/recommendations	Accepts a mood string, uses Gemini AI to convert it into genres, then fetches matching movies from TMDB. Saves to user history if authenticated.	JWT
GET	/api/favorites	Returns the authenticated user's saved favorite movies list.	JWT
POST	/api/favorites	Adds a movie to the user's favorites. Prevents duplicates.	JWT
DELETE	/api/favorites?id={movieId}	Removes a movie from the user's favorites by movie ID.	JWT
GET	/api/history	Returns the user's mood search history sorted by most recent first.	JWT

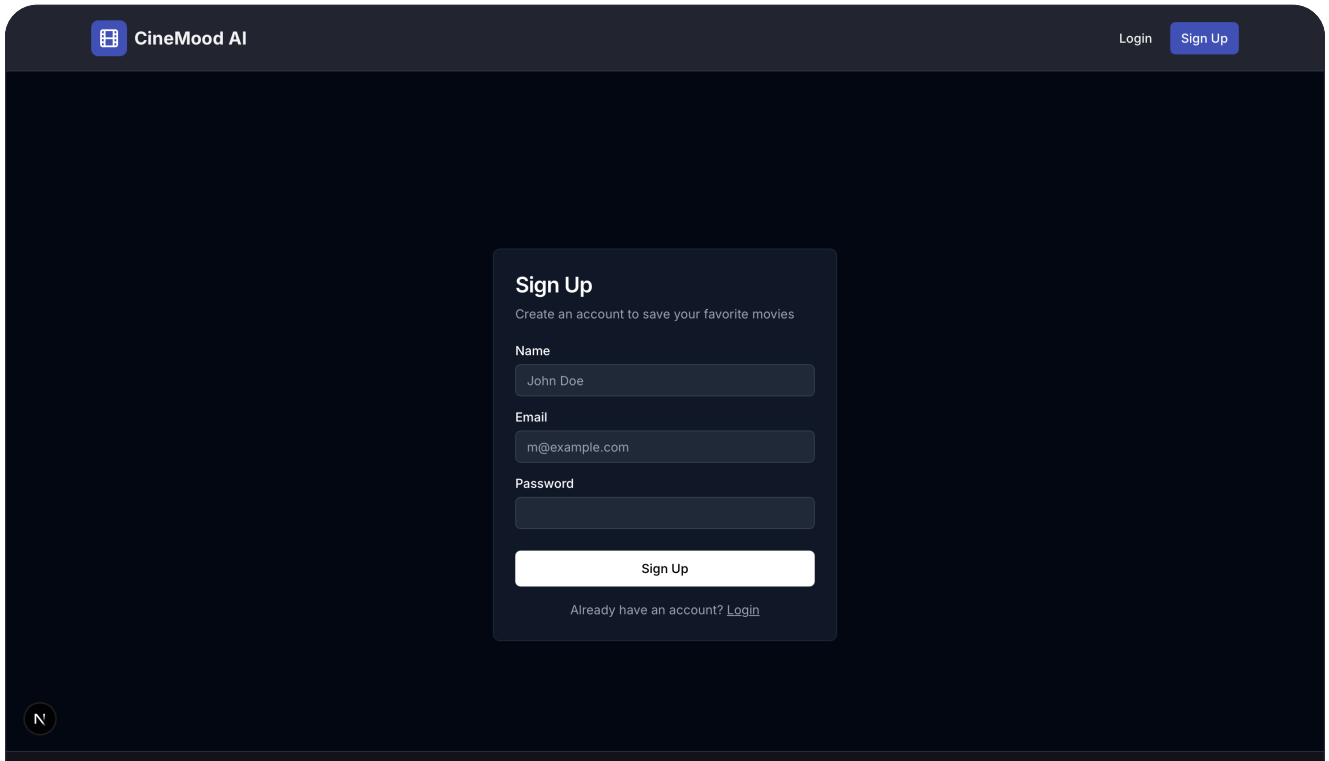
5 How to Run Locally

```
# 1. Clone the repository
git clone https://github.com/Gowreesh-VT/AWS-Web-Dev.git
cd AWS-Web-Dev

# 2. Install dependencies
npm install

# 3. Create a .env.local file with the following variables
TMDB_API_KEY=your_tmdb_api_key
GEMINI_API_KEY=your_gemini_api_key
MONGODB_URI=your_mongodb_atlas_connection_string
JWT_SECRET=your_jwt_secret

# 4. Whitelist your IP on MongoDB Atlas
# Go to MongoDB Atlas → Network Access → Add IP Address

# 5. Start the development server
npm run dev

# App runs at http://localhost:3000
```

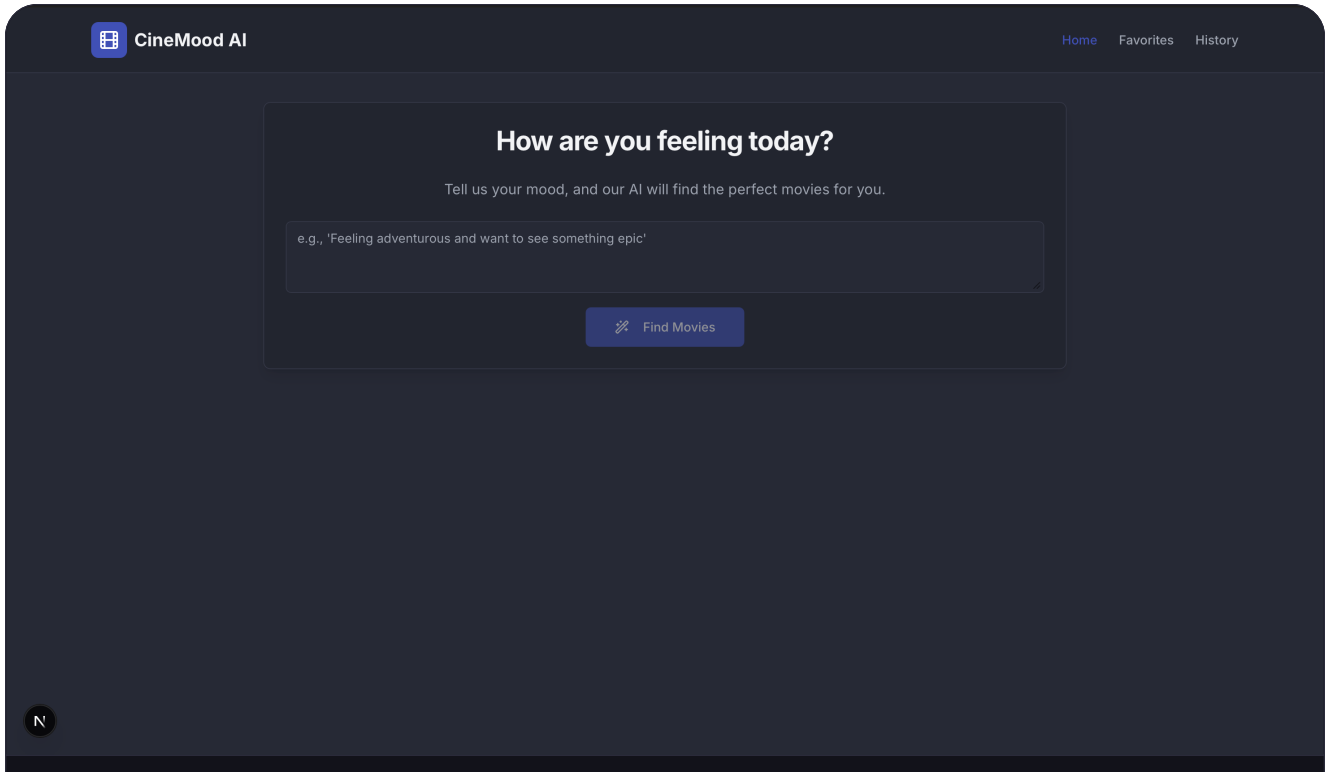
6 Screenshots of All Webpages



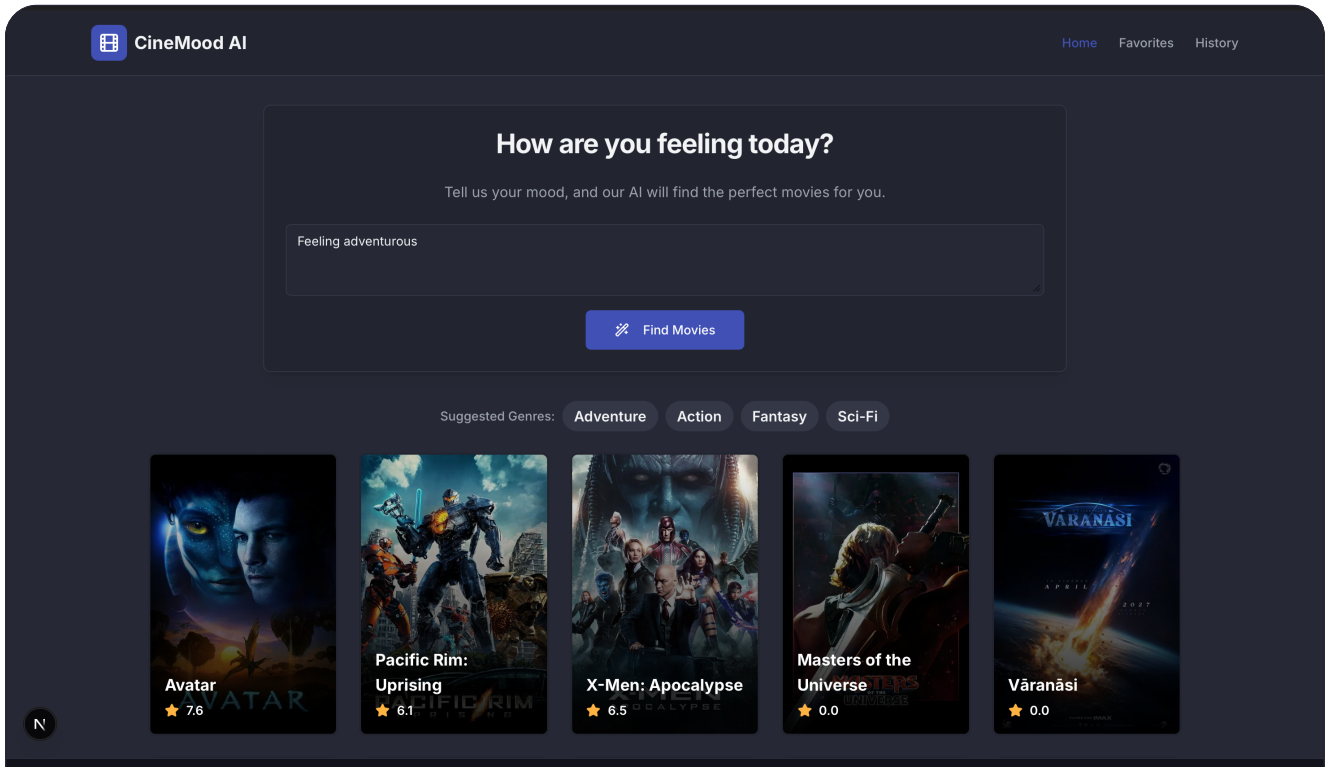
- Landing Page — Hero section with CTA to Login / Sign Up



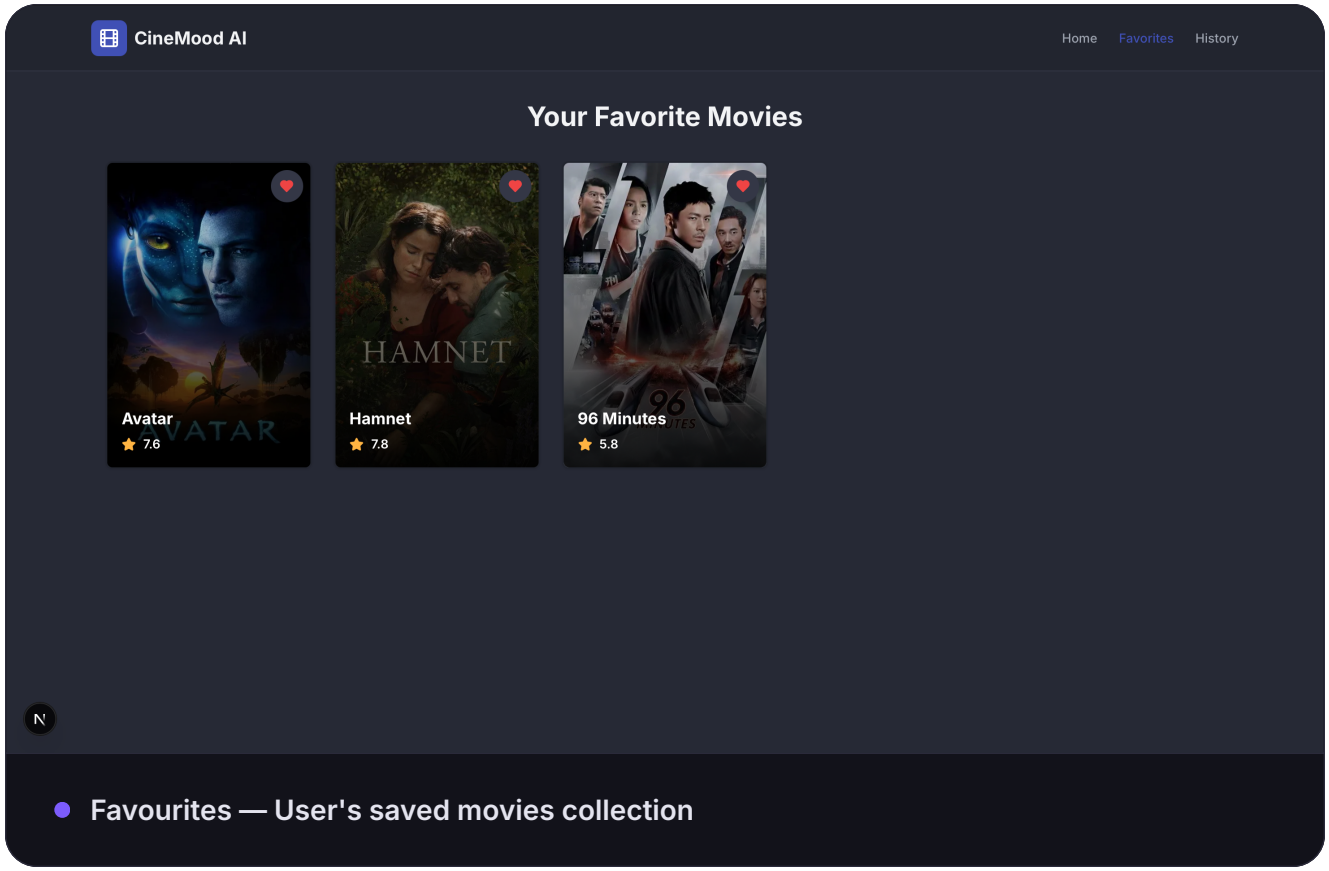
- Login Page — Email & password authentication form



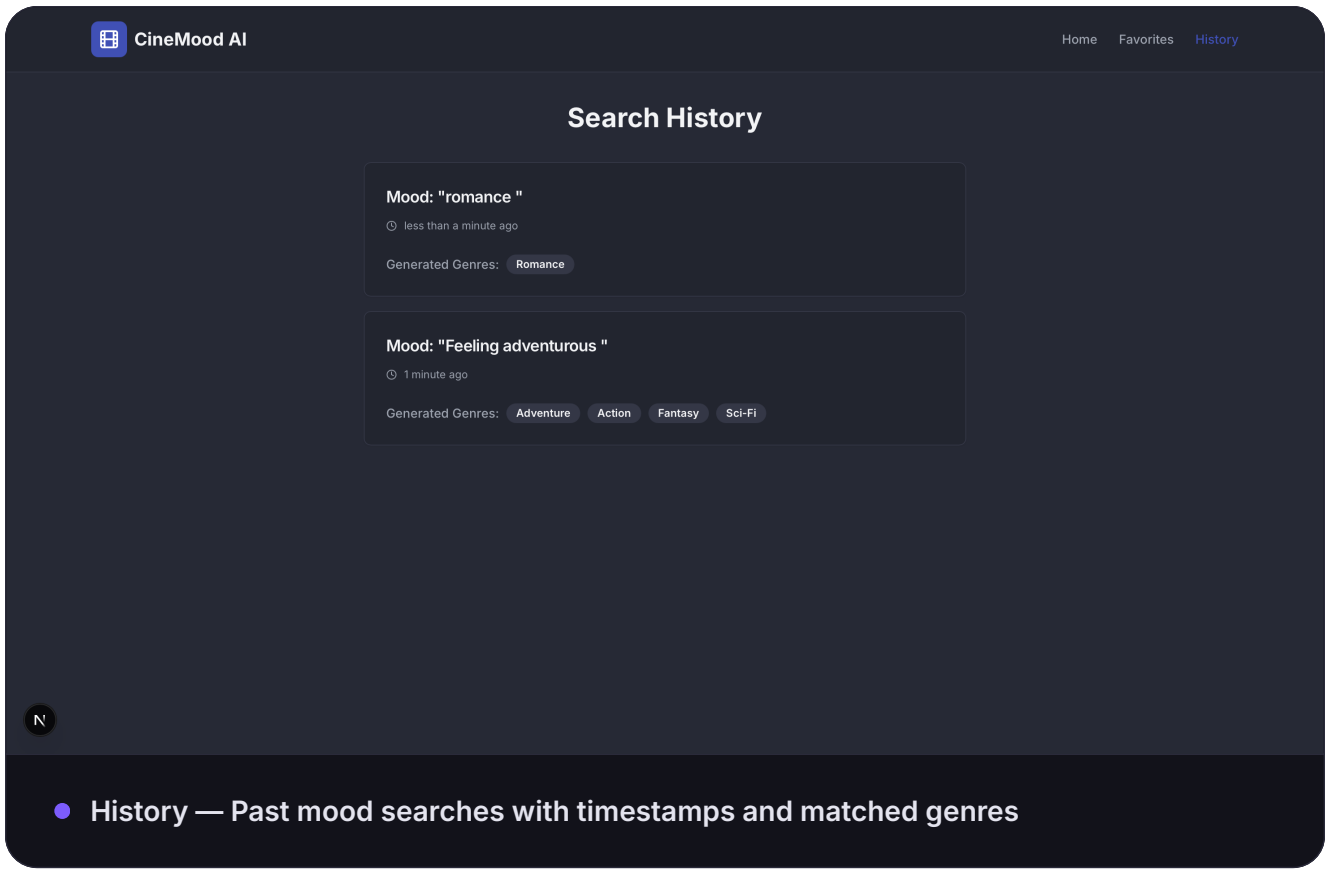
- Dashboard — Mood input form where users describe how they feel



- Search Results — AI-generated genre pills + TMDB movie recommendations



- Favourites — User's saved movies collection



- History — Past mood searches with timestamps and matched genres

7 Design Theme & Rationale

🎬Dark Cinematic Theme

CineMood AI uses a **dark-mode-first cinematic design** inspired by modern streaming platforms like Netflix and Letterboxd. The dark background (#0a0a0f) reduces eye strain during extended browsing sessions and makes movie poster artwork pop with vibrant contrast — exactly what users expect in a movie discovery app.

The color palette centers around **deep blacks, muted grays, and purple-violet accents**, creating a premium, immersive feel that mirrors the atmosphere of a movie theater. Typography uses clean, modern Inter font with gradient text effects on hero elements to add visual sophistication.

Why This Design?

🌙Dark Mode First

Movie posters and artwork stand out best against dark backgrounds, creating an immersive browsing experience similar to cinema.

🧠AI-Centric UX

The mood input is front and center on the dashboard — a single text field that makes the AI interaction feel natural and effortless.

🎯Minimal & Functional

Clean card-based layouts with generous spacing. No visual clutter — users focus purely on discovering movies that match their mood.

📱Responsive Grid

Movie cards use adaptive grid layouts (2–5 columns) that reflow gracefully across all screen sizes from mobile to desktop.
