

WEEK 3 PROGRESS-TO-DATE

Backend (Django + DRF)

TMDB API integration with helper `_tmdb_get` to handle keys, errors, caching.

Core endpoints:

- `/api/movies/search/` → search movies by title.
- `/api/movies/trending/` → trending movies of the week.
- `/api/movies/now_playing/` → in-cinema releases.
- `/api/movies/streaming_trending/` → trending on streaming
- `/api/movies/providers/` → list available streaming providers by region.
- `/api/movies/person_movies/` → movies by actor/actress.
- `/api/movies/{id}/` → movie detail with credits, trailers, and providers (cached).
- `/api/movies/poster_palette/` → extracts poster colour palette using “ColorThief”.
- User system: registration endpoint wired with RegisterSerializer.

Frontend (React + Vite)

- Routing setup with React Router DOM.
- API helpers (Axios-based) for calling backend endpoints.

Home page features:

- Movie search with live suggestions.
- Trending / Now Playing sections.
- Person search (actors).
- Movie detail page:
 - Poster, title, overview, release info, runtime, genres.
 - Dynamic gradient background (poster-derived).
 - Meta chips for runtime, genres, rating.
 - Trailer embed (YouTube).
 - Top Cast strip with scrollable cards.
- Where to Watch block:
 - Glass-effect panel.
 - Compact, scrollable, clickable provider chips linking to Netflix, Disney+, Prime, etc.
- Basic login and logout - private/auth only pages to be set up

Summary of what has been done to date:

- Search movies and people.
- Browse trending / now playing / streaming sections.
- Themed movie detail pages with cast, trailer, and streaming availability.
- UI styled with dynamic gradients and glass cards for a modern look.
- Backend caches and abstracts TMDB API for efficiency and safety.