Complex IT Systems – Practice

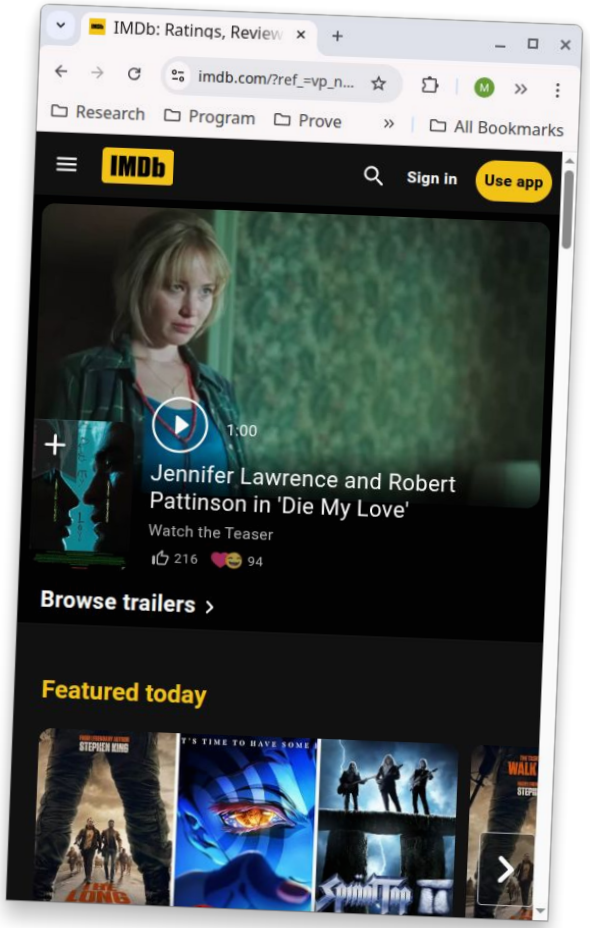
The Portfolio Project

CIT 2025

Morten Rhiger

Goal of the project

- Implement an **online database** for **movies, actors, directors**, etc. (Like IMDb.)
- **Display** (fixed) **information** (movie titles, years, person names, ages, etc.)
- **Visualize user-generated data** (history, my rating of a movie, overall rating of a movie, etc.)
- **Browse** movies, actors, etc.
- **Search** for movies, actors, etc.
- Support **bookmarking** of items.
- Allow **rating** of movies.
- Keep track of users' **search history.**



Sources of data

IMDb (The Internet Movie Database)

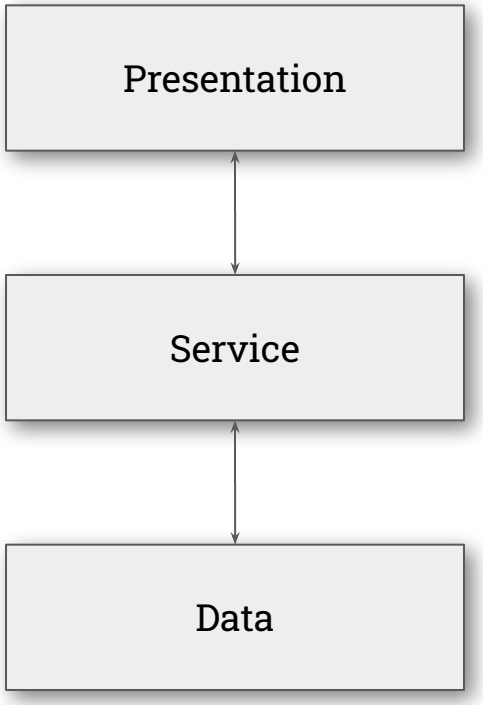
- Online database of movies, tv series, actors, directors, etc...
- The largest, most comprehensive movie database on the web: ~15 million titles, ~12 million persons.
- Launched in 1990, now owned by Amazon.
- Smaller, publicly available dataset with ~110.000 movies and ~330.000 persons.

OMDB (The Open Movie Database)

TMDB (The Movie Database)

- Community-built movie and tv database.
- Includes additional images of actors and directors.
- Free online access via public API.

3-layer architecture



Single-page web application.
Runs in a browser.

Application logic.
interface to the presentation layer through web services.

Encapsulates storage and retrieval of data.
Provides an interface to the service layer by a set of functions and procedures.

Three portfolio subprojects

- **Portfolio subproject 1 : Database**
  - Design and implement a database
  - Fixed data from IMDB and OMDB
  - Additional structures to support users (signing up, bookmarking, rating, etc.)
  - Provide an API to the backend
- **Portfolio subproject 2 : Backend (server)**
  - Design and implement web services to access and manipulate the database
  - Provide a RESTful web service interface to the frontend
- **Portfolio subproject 3 : Frontend (client)**
  - Design and implement a graphical user interface to the application
  - Access data from TMDB via its API.

The portfolio subprojects

To be handed in:

- **Programs** implementing parts of the system
- A **report** documenting these programs
- (Once, along with PP3) An **individual reflection** discussing a concept from the course, and relate to own product.

What to write? How?

- Each portfolio project is an implementation **exercise**, not a **software development project.**
- How to document these programs?
  (After all, it is given what they must do, which libraries, frameworks, programming languages they may use, when they fulfill the requirements, etc.)

On the content of the reports

- For each subtask, procedure, functions, class, method, component, etc: **analyze, design, implement, evaluate,** and **reflect.**
- Avoid being (just) descriptive. **Explain, motivate, justify.**
- The report is **not diary of activities** documenting what you did when. Present the product, not how you got there.
- The report is **not a list of things** (programs, classes, tables, methods, components, procedures, etc.).
- Avoid stating what is obvious, common knowledge, or already given by the requirements.

On the (writing) style of the reports

- Be **precise**.
- Be **consistent.**
- Be **concise.**
- Be **formal.**
- Be **systematic.**
- Make the report **readable.**
- Write in english! Write whole sentences. Use english spelling and grammar.
- Use bullets, numbered lists, tables, diagrams, sections, etc. cleverly.
- The reports don’t have to be long: don’t add stuff just because you have that stuff.

On the format of the reports

- **Format consistently.** A change in format conveys information.
- **Number all sections.**
- **Number all pages.**
- Add a **table of contents.**
- Use a **monospace** font for code.
- **Don't copy-paste code** from an IDE (VSCode, Visual Studio)!
- **Never show screenshots of code!**
- Have many figures? Put them in appendices.

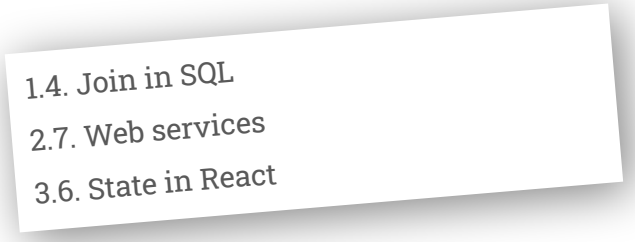
On the process of writing the reports

- **Spend time** on the report!
- Think about the choices you make. (You may diverge from these suggestions, but at least make an informed choice.)
- Don't write the report shortly before deadline.
- Write the report **iteratively: Draft** a section or paragraph, then **revise** and **refine** it. Let other group members **review** a what you wrote.
- Proofread for grammar, spelling, and formatting errors. Use a spell checker.
- (Read or skim the set of requirements for all three portfolio projects before getting started on the first.)

Exams

Complex it systems – Theory

- **Oral exam**
- **Individual**
- No preparation
- Pick snippet with three topics, then present these topics



Complex it systems – Practice

- **Oral exam**
- With whole **group**
- Based on portfolio projects
- Each student present own individual reflection
- We all discuss the details of the implementation