

Exam Assignment: Development of a Data Warehouse as a Fullstack Application

1 Project Description

Develop a complete fullstack application in the programming language of your choice that functions as a data warehouse. The objective is to build a functional prototype that facilitates the management and analysis of public datasets. The focus is not on the content or complexity of the data itself but on designing a robust architecture and implementation that demonstrates modern data management practices.

2 Frontend Requirements

2.1 User-Friendly Interface

Develop a frontend that presents a structured master-detail view. The application should provide intuitive navigation components, allowing users to view, create, modify, and delete records.

2.2 Technology Choice

You are free to choose between:

- Server-side rendering using a template engine or HTMX, or
- A client-side framework such as React (or an equivalent technology).

Your choice of technology and its advantages must be clearly justified in your report.

2.3 API Interaction

The frontend must interact with the provided API to perform all CRUD operations.

3 Backend, API, and Database Integration Requirements

3.1 Server and API

Develop a server that connects your application with the frontend via an API (RESTful, GraphQL, or another approach). The API must support fundamental CRUD functionalities (Create, Read, Update, Delete) and include a search function with pagination to

enable efficient dataset navigation and filtering. You must justify your choice of API style and the implementation of the search functionality, including pagination, in your report.

3.2 Database Integration

The approach to database integration is flexible—you may opt for an object-relational model (ORM) or another method. Your implementation must be thoroughly documented and justified in the scientific report.

4 Data Model and Sample Data

4.1 Choice of Dataset

Select one of the following publicly available datasets as the basis for your data model:

- Gene Expression Omnibus
- Ensembl Biomart
- UniProt KB
- Human Protein Atlas
- Protein Data Bank (PDB)

4.2 Database Schema

Design an appropriate schema that represents the structure and relationships within the selected dataset. It is sufficient to include a few representative records—the primary focus is on system architecture and integration.

5 Scientific Report

Prepare a comprehensive scientific report (in PDF format) that documents all aspects of your project. The report should include the following sections:

Introduction

Provide an overview of the background, motivation, and objectives of your project.

Materials

Provide a detailed list of the technologies, tools, frameworks, and libraries used in this project. Specify the version numbers to ensure reproducibility. Additionally, include information about the chosen dataset, its origin, and sample records integrated into the system. This section should enable a reader to replicate the development environment and verify the consistency of the results.

Methods

- **Frontend:** Explain the implementation of the master-detail view, the user interaction mechanisms, and the reasoning behind the selection of your frontend technology. Describe how components communicate with the backend and any optimizations applied for usability and performance.
- **API & Server:** Outline the interaction between the frontend and backend, including the reasoning behind the chosen API type (REST, GraphQL, or alternative). Detail the implementation of CRUD operations and the search functionality with pagination. Additionally, provide a step-by-step guide on how to start the server, configuration files, and execution commands.
- **Database Integration:** Describe the database schema, including the structure and relationships between tables. Justify the choice of integration approach (e.g., ORM or direct queries) and explain how data persistence and retrieval are handled.
- **Software Design:** Present an overview of the application's overall architecture, including major components and their interactions. Discuss key design decisions, performance considerations, and potential scalability aspects.

Results

Present examples, screenshots, and functional evidence demonstrating the application's features.

Discussion

Evaluate the strengths and weaknesses of the chosen solutions, discuss encountered challenges, and propose alternative approaches.

Conclusion

Summarize key findings and suggest possible enhancements or extensions.

Formal Requirements

Ensure proper scientific citation and include a declaration of originality.

6 Group Work

- **Teamwork Allowed:** This assignment may be completed in groups. However, the individual contributions of each team member must be clearly distinguishable. For example, in a team of two, one member could focus on frontend development while the other handles backend implementation.
- The workload distribution must be equitable, ensuring that a frontend-specific role is as technically demanding and thorough as a backend-focused role.

- In group projects, the report may be tailored to emphasize each team member's specific role, but it must also provide a comprehensive explanation of the interactions between all components. For instance, if responsible for the frontend, your report should primarily focus on that aspect while referencing the API when explaining its integration.

7 Submission and Evaluation

7.1 Code

Submit the complete, well-structured, and documented source code. The code should clearly illustrate the interaction between different components.

7.2 Scientific Report

The detailed PDF report will serve as the primary documentation for your project.

7.3 Deadline

The final submission of this assignment is due by August 31, 2025. All required materials, including the source code and scientific report, must be submitted by this date. Late submissions may be subject to penalties as outlined in the module handbook.

Additional Notes

Consider integrating validation and testing strategies (e.g., unit tests, end-to-end tests) early in development to ensure application robustness. Evaluating different technologies and architectural choices is encouraged—critical reflection and well-founded reasoning are crucial to the scientific discourse of this project.

This assignment offers you the opportunity to refine your skills in frontend and backend development while applying a holistic approach to building modern data warehouses.