

Semester Project

Exam Case

Courses included:

Web Programming

Backend & Databases

Table of Contents

Introduction	2
Web Programming	3
Cinema	Fejl! Bogmærke er ikke defineret.
Frontend requirements	3
Backend requirements (admin)	3
Photo sharing platform	4
Frontend requirements	4
Backend (admin) requirements	5
Databases	5
Requirements	5
The Documentation (Report)	6
Web Programming	6
Database	7
Groups	7

Introduction

This document describes the framework for the shared semester project exam case for the courses 1) Web Programming Backend and 2) Databases.

The project is the foundation for the exam that will include both courses at the end of the semester.

Overall, the project is about developing a complete and fully functional mini Content Management System (Dynamic web page/DWP) for a selected domain.

You must choose between one of two types/themes for your project:

- A website for a Cinema. You are free to decide your product types. Examples: B-Movies, superhero movies, Art movies from all around the world, A normal cinema etc.
- A photo sharing social media platform for at selective domain. Examples: food and recipes, health and exercise, cars, pyromaniacs.

The important thing is that you find some inspiration, of which to build your webpage on, or to use as the basis for your theme.

There are no design/aesthetic criteria in this project, but it is still very nice to have!

The project will be assessed in relation to the requirements + course learning goals as well as the complexity, completeness and quality of the system, the database and the documentation.

Web Programming

General requirements

You must use vanilla PHP as the programming language/platform for the web application. When developing the platform, you must use classes/objects and describe in your report how OOP is integrated into your system.

Your dynamic web page **must** run and be located on a **live server**.

Cinema

Frontend requirements

- A contact form, which sends mails to the owner.
- A complete list of movies and Venues (loaded from database)
- A news section (loaded from database)
- The front page must contain (loaded from database):
 - o News
 - o The daily showings
 - o A presentation of the company
- A profile page for the user
- A checkout system (preferably with payment)
- A booking page for seat reservations
- A seat reservation system, as advanced as you can build it.
 - o You need to apply some rules for the reservations.
- Could be:
 - o Simple seat selection form with date, name, movie-id only 5 seat pr. customer.
 - o An advanced graphic representation of the different Venues and seat placements that you can choose from with a max of one seat between reservations.
 - o Just some sort of seat/movie reservation

Backend requirements (admin)

A dashboard/backend system for the company owner.

The criteria are:

- Security - Password protected (Session)

- Booking overview
- Invoice creator
- The company owner should at least be able to update (CRUD) these pieces of information:
 - o Description of the company
 - o Opening hours
 - o Contact information
 - o The movies/showings
 - o Daily premiere/showings (movie) on the front page
 - o The news

Photo sharing platform

Frontend requirements

The site must include

- An image overview site.
 - o Number of likes/comments per picture (loaded from database)
- A newest picture section (loaded from database)
- The front page must contain (loaded from database):
 - o Hot new pictures
 - o Upload a new image
 - o Latest comments
 - o About the platform
 - o Sticky images
- Individual image display with comments, likes and text styling and image upload
- A User profile page with all data stored in a database (as advanced as you can build it). Could be:
 - o Simple user display with email, name, birthdate and rank/access level, badges.
 - o An advanced graphic representation of the user data (think infographic dashboard).
- Must include:
 - o User editable settings and information
 - o User upload of new images

Backend (admin) requirements

A basic backend system for the site owner.

The criteria are:

- Security - Password protected (Session)
- He should at least be able to update (CRUD) these pieces of information:
 - o Description of the site
 - o Rules and regulations
 - o Contact information
 - o All user post/profiles/settings
 - o Ban/block users
 - o Hot new pictures
 - o Styling settings

Databases

Requirements

- The database must be implemented using a Relational Database Management System
- The database must be modelled / sketched using ER diagramming and transformed into a Relational Data Model
- The database must be normalized up to the 3rd Normal Form (3NF)
- You must create a minimum of 2 views that is used by the DWP
- You must create a minimum of 2 triggers that is used by the DWP
- You must implement a minimum of 2 relevant indexes
- You are free to make your database more advanced if you feel like it
- You must include an SQL script file (made by YOU and not by the PHPMyAdmin Exporter) that can create the complete database including test data, views etc. when you hand-in the project. Include the SQL file in your GitHub repo.

The Documentation (Report)

The documentation must not exceed 10 normal pages (excluding front page, intro etc.) Make sure to evenly distribute the number of pages between programming and database.

Web Programming

The report **MUST** contain:

- **Site-map:** An overview of the pages in the application. For each page, you must give a brief description of the functionality (from the users' point of view).
- **Architecture:** The architecture of the application must be explained – and focus on how you have separated the presentation and programming logic.
- **Flowchart** of the webpage (or specific function/work path)
- A **description** of the web page in general.
 - o Specific technical details: Describe the structure of a few selected (difficult) pages, classes or functions of your application.
- **Session handling:** In the section, you must describe how you handle session state.
- **Security:** describe how you handle security concerns in your application.

You have to demonstrate your knowledge and skill within OOP and integrate **at least three (3) classes** that you **instantiate** in your system and have a **section in your report** where you describe how you perform the integration.

You need to either implement or describe the use of a design pattern specific to your project.

Database

- **A description of the database:** Describe and argue for your choice of entities, relations, data types, overall data architecture etc.
- **Database diagramming model(s):** Include the ER diagram + the Relational Data Model
- **Advanced SQL query examples:** Include and explain interesting and important SQL queries used within the application
- **Code snippets** demonstrating how your DWP/CMS integrates with the database
- **Your design choices/reflections/considerations** in relation to the development of your database as well as the process. Topics you should include as a minimum:
 - o Security: Have you secured the database/DWP in order to prevent malicious users/hackers etc.?
 - o Performance: what have you done in order to optimize the database/queries etc.?
 - o Workflow: how did you (as a team or individual) plan, design, develop and implement the database?

Groups

You may form groups of 1-3.