NTNU

Kunnskap for en bedre verden

PROJECT REPORT

Group report by:

Ragnhild Alstadsæter
Sondre Berntsen
Elisabeth Medlien
Mads Øigård

SPRING 2018

GJØVIK

Organization

We started working on the project after several other due projects, so we had a time constraint of two weeks. We decided to meet and work on the project every day to make up for lost time.

The first few days were spent designing and constructing the database, after which we made some simple design frameworks for different views. When we had a general plan for how the different pages were going to look, we spent one day looking at Laravel to decide if we wanted to use that as a framework, or make our own. Laravel seemed overwhelming for the short timeframe we had, so we decided to make our own framework in the end.

One person was in charge of the HTML and CSS, and code implementation was mostly divided by views.

Design

The interface is easy to understand, without excessive information. All group members know bootstrap well and it is being used on all views of the webpage making the design look consistent and mobile friendly.

Database design

Our database 'secondhand' consists of five tables; user, category, item, message and messagethread. In the development of the database we created an EER diagram and a relational model to visualize how the database was formatted.

Performance

We used a lot of Ajax calls to reduce the necessary page reloads. Clicking various items throughout the page queries the DB and provides useful data if for example the user is trying to log in with incorrect data. Clicking category items or message threads provides dynamically loaded content without reloading.

Our SQL queries are indexed to reduce redundant data.

Security

We are using PDO for all our SQL queries, which provides good security with prepared statement. We have also hashed all password.

Frameworks

Due to time limitations, we decided to make our own PHP framework rather than learning to use an existing one. Following the MVC-pattern, we have controllers for each page providing all available user options, which talk to the model classes if a function is called. We also have separate JS pages for each page, along with a "main.js" file for functions required on multiple pages.