[A] CRUD Code Snippets

In this assignment, you will create a web application for persistent handling of programming code snippets using an application framework and an object data modeling library for MongoDB.

Goals for this assignment

These are the main goals for this assignment.

- Learn to build a website using Node and Express.
- Learn how to connect to a database from Node.
- Practice writing a CRUD web application.
- Practice writing code according to the MVC design pattern.
- Practice working with sessions for authetication and flash messages.

Grading

This is an individual assignment.

This assignment is graded as Fail (U) or Pass (G) and it is worth 2.5 credit/hp.

Preconditions

You have access to the course environment on GitLab.

You have knowledge of programming JavaScript with Node, Express and Mongoose/MongoDB.

Get going

Get going with the assignment by cloning and preparing the assignment repo.

acronym in the course at GitLab. Start of by cloning the repo to your computer ("xxx" is your lnu-username):

```
# Using ssh
git clone git@gitlab.lnu.se:1dv528/student/xxx/b2-crud.git

# Move into the repo
cd b2-crud
```

Add the development tools

Setup the repo with the same development environment used. The document "Development environment" contains the details on how to do that.

Requirements

The application in Node.js will use Express as the application framework and Mongoose as the object modeling library. The application must have full CRUD functionality regarding snippets, whereby a user must be able to create, read, update and delete snippets.

Users must be able to register and login to the application after entering a username and a password. A user cannot register an already existing username because the username must be unique to the application. A logged in user must be able to log out of the application.

Anonymous users should only be able to view snippets. Authenticated users, in addition to view snippets, must also be able to create, edit and delete their snippets. **No one but the authenticated user should be able to create, edit and delete their snippets.** Because of this, the application must support some basic authentication and authorization. On the server-side you may only use plain session storage, using the express-session package, to implement authentication and authorization. **You must not use any packages such as Passport, etc., to authenticate or authorize.**

When writing and presenting snippets, **the application must support multiline text**, enabling the user to write real code snippets, not just a one-line text string.

The application should be easy to understand, meaning that the users should get clear notifications on what is going on in the application (eg. using flash messages).

As far as possible, the application must be protected from vulnerable attacks.

Code style

Requirements related to the code style and how to run the application.

- The only command that the examiner will need to execute to run your application after cloning it from GitLab is `npm install` and `npm start`.
- The linters should pass using `npm run lint`.
- Use JavaScript/TypeScript, Node, Express.
- Use an MVC structure to structure your code.
- Make several committs to yout Git repo.

Setup database

You will need a database connection. You are free to use Docker or a connection-string to mLab (see the database connection text for more information)

Extra features [optional]

For those who want to expand functionality in the application, feel free to do that. Some examples of this could be:

- Add support for tagging each snippet by one or more tags.
- Add support for just showing snippets that's belongs to one tag or/and one user.

Presentation

You are to record a video presentation of your project where you show how your project implements the features and fullfills the requirements. You should also present a few highlights from your source code and elaborate on why you feel the solution/code is good or how it could be enhanced.

The video should be 5-7 minutes.

Submission

This is how to submit this assignment.

1. Add a tag `v1.0.0` to your repo when you are done. If you make updates then add another tag like `v1.0.1` or `v1.1.0` and so on.

2. Ensure you have committed and pushed all your changes, including the tags, to GitLab.

3. Create an issue on the repo and answer the following questions in it. Write freely with 15 to 30 sentences of text.

    1. Describe the architecture of your application. How have you structured your code and what are your thoughts behind the architecture? What are your thoughts on MVC?

    2. How was it to work with persistent data in this assignment, can you relate to other databases you might have learned before?

    3. Are you satsified with your application? Does it have some improvement areas? What are you especially satisfied with?

    4. What is your TIL for this course part?

4. Ensure that the README.md contains any essential details for installing, starting, configuring your submission.

5. When you are done, assign the issue to the teacher, to show that you are ready for grading. The teacher will be notified by mail.

[L] Access control