The design and implementation of Mini Moodle: a student, course and assignment management system

Faculty of Information Technology

Hongyuan Jin

1409853G-I011-0046

1. Introduction

Mini Moodle is a web-based online grading system powered as a Single Page Application. With a user-friendly and carefully styled interface, it can efficiently help both teachers and students to manage their courses and assignment. Several modern JavaScript frameworks and Microsoft SQL Server Express 2016 are used in the development of this system. This project report is divided into the following parts: 1) Objective and solution to describe the objective of this system and the technical stack used during development, 2) Architecture and database design to illustrate the logic and data model of this system, 3) General workflow to demonstrate the elaborated usages of this system and 4) a basic user manual.

2. Objective and solution

2.1 Objective

The objective of MM is to develop a user-friendly, featured and straight-forward system for both instructors to manage their student lists, courses and assignments and for student to submit their assignments as well as checking the grading details. The basic functions include:

a. Student lists creation,
b. Courses creation and enrollment,
c. Assignment publish and grading,
d. Online assignment submission and statistics report.

2.2 Technical solutions

A light-weight and highly efficient web-based, online grading system requires a set of small-sized, flexible and modularized solutions stack. The system is a light-weight Single Page Application (SPA) which can run on all the modern browsers. During the development process, I decided to implement this system in three layers: view, model and data driver. All the frameworks and tools involved are as follows:

a. View-model layer:

- **Vue.js** (https://vuejs.org/) -- development of the dynamic interface presentation, workflow management, data calculation, fetching and posting, pre-evaluation of data validation before submitting to the server via Ajax, SPA front-end routing and local state management;

- **Sass** (http://sass-lang.com/) -- CSS extension and preprocessing tool for better organization of the display styles;
- **Webpack** (https://webpack.github.io/) -- module bundler to preprocess JavaScript and other static files to eliminate unnecessary repeated steps.

b. Data-driver layer:

- **Node.js and Express.js** (https://nodejs.org/en/) -- JavaScript runtime to provide HTTP data-fetching routing and connection with database;
- **Microsoft SQL Server Express 2016** (https://www.microsoft.com/en-us/sql-server/sql-server-editions-express) -- relational database system for data storage and management

3. Architecture and database design

3.1 System architecture design

N-tiers application is used in the development of this SPA online grading system. Our system is divided into three logical parts: view layer, model layer and data-driver layer:

Tier 1: View layer

View layer provides the system functionalities for the direct interactions with users. Data fetched from model layer are rendered and generated dynamically. In the meantime it provides form controls for users. Changes in control are sent back to model layered to be processed. Data from controls are mainly student, course and assignment information get from the user;

Tier 2: Model layer

Model layer fetch data from server side and at the same time process validations and calculations with the data sent from controls of view layer. It works like an intermediate agent between the view layer and data-driver layer;

Tier 3: Data-driver layer

This layer serves as the back-end server support for the whole system. It provides HTTP request routing, data access and data storage in database. Database queries and stored procedures are implemented here to perform any database operations. Database in this layer is relational database. This layer is not allowed to directly communicate with the view layer. All the request sent must be first validated and authenticated through model layer.

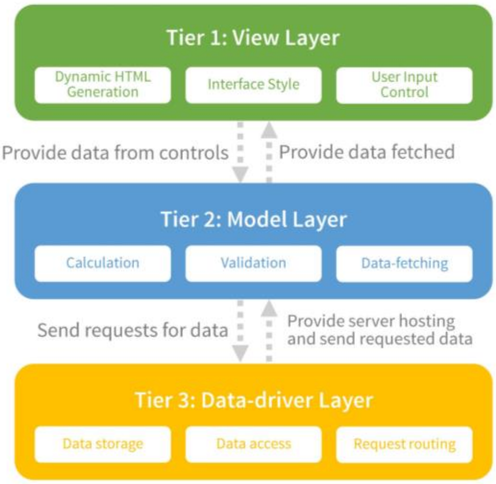


Figure.1 3-layered Mini Moodle architecture

3.2 Relational database schema design

Online Grading System data is stored in Microsoft SQL Server Express 2016. Figure 2 and 3 illustrate the ER model, database diagram and schema between tables.

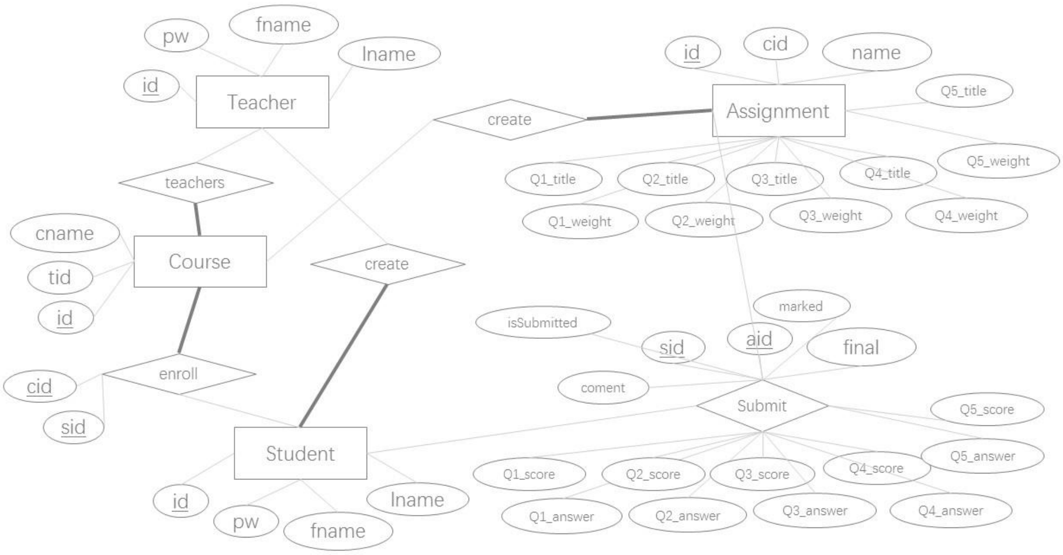


Figure.2 ER model of database design

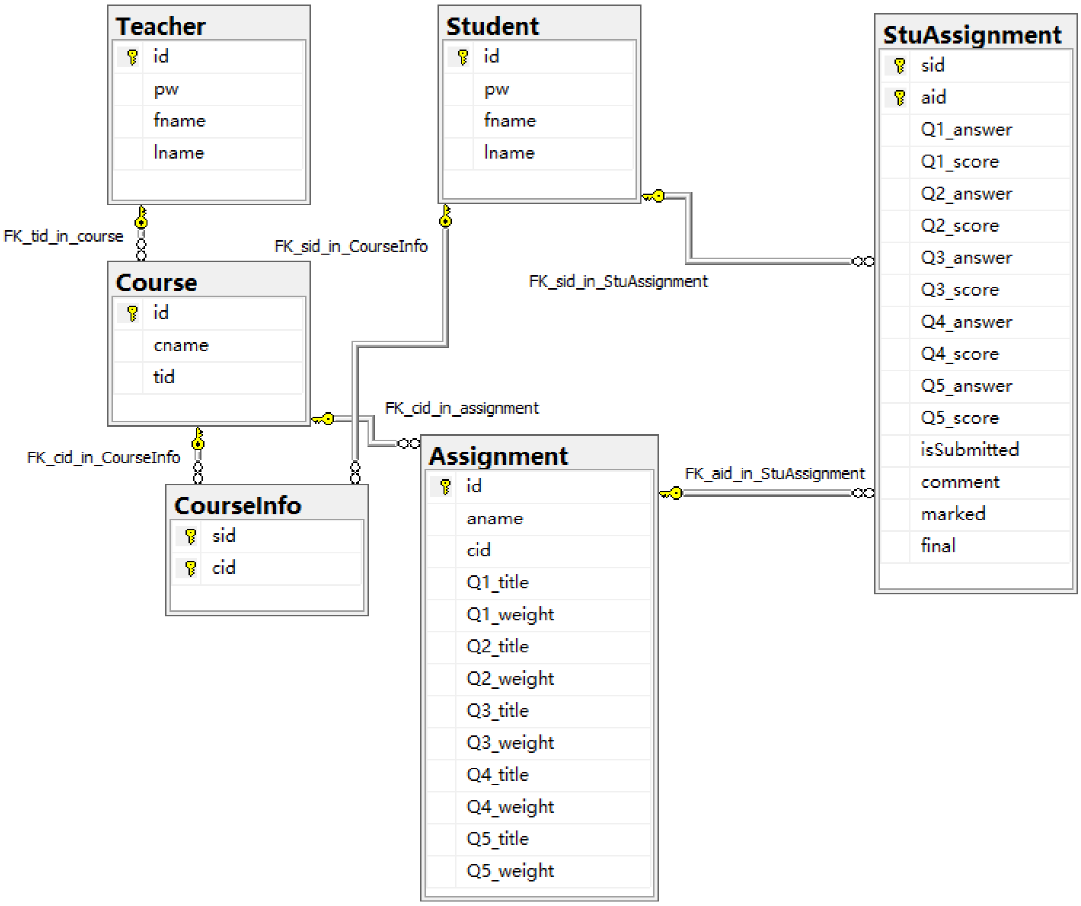


Figure. 3 Class diagram of database design

4. General workflow of Mini Moodle

4.1 General log in/out mechanism

The entrance portal of MM is a login panel for users that have not signed in into the system. User can choose his/her log in mode (student/teacher) according to their own identity. ID and password are preset in the database which will be examined of their validation with the help of Ajax before actual redirection to the chosen system page. If the password is not a valid match to the relative ID, alert will be shown to indicate the user for another try.

After successfully logging in, basic user information including the login ID, login type, user first and last names will be stored in the client side as local storage, which will preserve the login state and perform as the key to fetch data every time there is a redirection request. Therefore, accidental operations such as refreshing/closing the webpage itself or enter the page with only the root path will still successfully lead the user to the desired dashboard.

To log out, simply click the "logout" button on the left-hand side of the dashboard and all the data stored in the client side are wiped out and the system will automatically redirect you back to the log in page.

4.2 For instructors

There are three components in the instructor section including student management, course management and assignment management. Student management enables an instructor to add new accounts for students, while course management provides a convenient way for instructors to add course entries and select the students enrolled at the same time. Both student and course management support one-time creation of multiple entries. In terms of the assignment management section, courses and related assignments overview is presented in an organized way, and functions like assignment addition and grading are supported in this section.

4.2.1 Student Management

Student management enables an instructor to add new accounts for student. After entering the basic information, the instructor can choose to add another entry or directly create this student entry into the database. Before adding new entries or creating student records, the MM system will firstly check the data validation of the last item to prevent repeated student id and incomplete student information. Instructor can also choose to delete an undesired student entry which has not been created into the database. After creation, student record editing panel will be set to the default one entry.

4.2.2 Course Management

Course management basically shares the same adding/deleting and creation function as Student Management, yet what makes it unique is that instructor can choose the students enrolled in this course in the meantime. The chosen students will be assigned with the assignments published in this course.

4.2.3 Assignment Management

One of the spotlights featured in Assignment Management is that the assignment creation process is extremely straight-forward. Instructors can first choose a course and then define the assignment ID and title, five questions and weight respectively and the assignment is distributed to the students enrolled in this course. Same ID validation mechanism is also employed here. If there's a repeated assignment ID, the creation will not be completed until the instructor assign a new ID to this assignment.

As for assignment grading, after selecting the assignment to mark, the system will display a list of assignment submitted by students. The details of each student's answer can be examined on the click of the button "detailed". Instructor can then mark each of the questions can even add an overall comment for the performance of the student in this particular homework. After confirmation of grading, a tag of "marked" will be attached to the related student assignment entry, helping the instructor to specify the marked questions from the unmarked ones. On confirming the grading, the final score for this assignment is calculated automatically and stored into the database at the same time.

4.3 For students

The functionalities of MM for students are simple, neat but still powerful. Dashboard lists panels of assignment overviews per course. At the end of the line of each assignment is tags showing the grading or submitted info. If marked, students can click on the "details" button to check their grade, final score as well as the comment from the instructor. If not submitted, the "answer" button is where the student can click and find the entrance to view the questions for the assignment, put down their own answers and submit. After submitted, the assignment will be shown in the instructor's marking panel.

4 **Brief user manual**

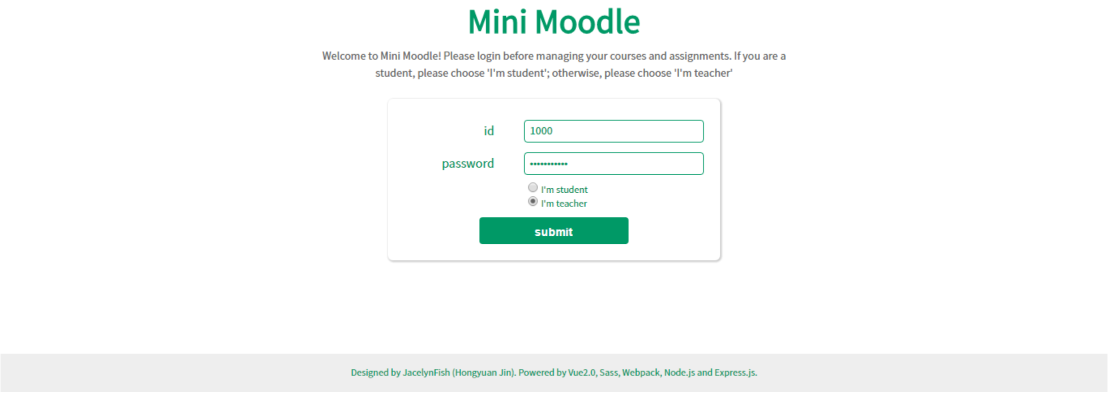
a. Modify the database configuration file “dbconfig.js” to set up your database host name, id and password. Create the database and tables using “database_creation_query.sql”,

b. Please go to Node.js’s official website (https://nodejs.org/en/) and download, install the latest version. After installation, open the cmd under the project’s directory. Enter the following command:

```
npm install
cd bin
node www
```

And the system server will be running on port 3000.

c. Login: enter id and password and choose your login type before submitting.



Mini Moodle

Welcome to Mini Moodle! Please login before managing your courses and assignments. If you are a student, please choose 'I'm student'; otherwise, please choose 'I'm teacher'

id: 1000

password: ••••••••••••

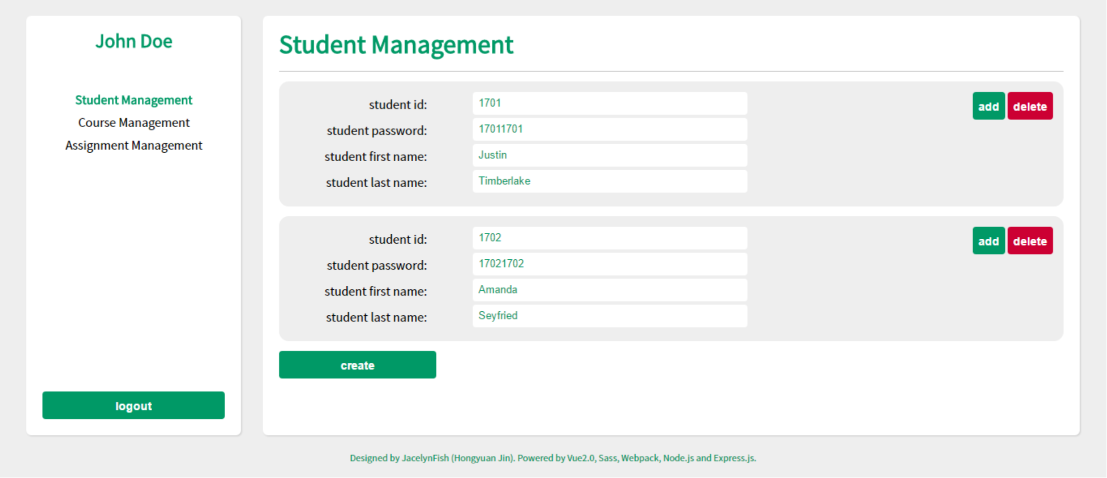
○ I'm student
● I'm teacher

submit

Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.

d. For teachers: on the left hand side navigation bar, choose the preferred functions that you want to use.

In student management, student information must be all completed before adding another entries. If you don’t want to create a particular student you have added, simple click the "delete" button.



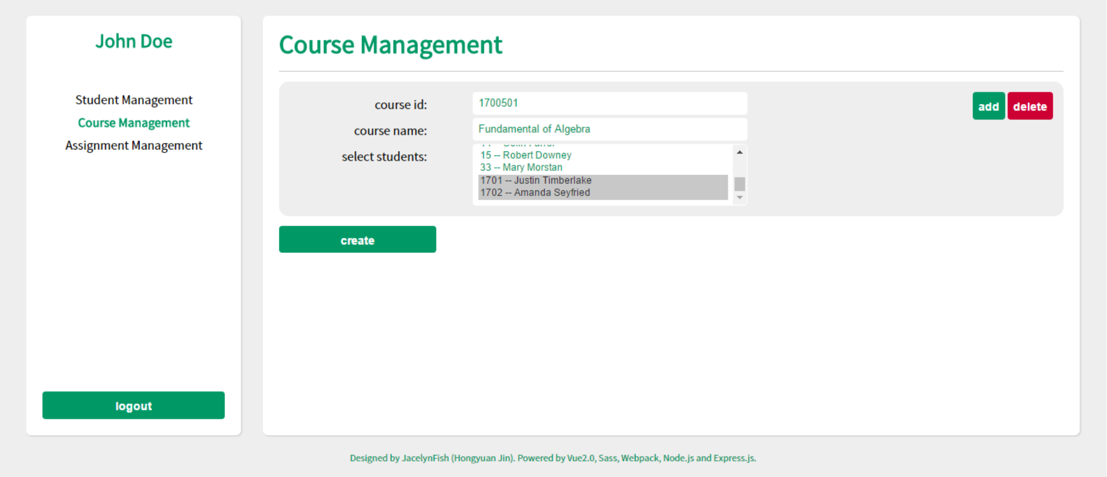
Student Management

student id: 1701
student password: 17011701
student first name: Justin
student last name: Timberlake

student id: 1702
student password: 17021702
student first name: Amanda
student last name: Seyfried

Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.

In course management, student enrollment list support multiple selection.

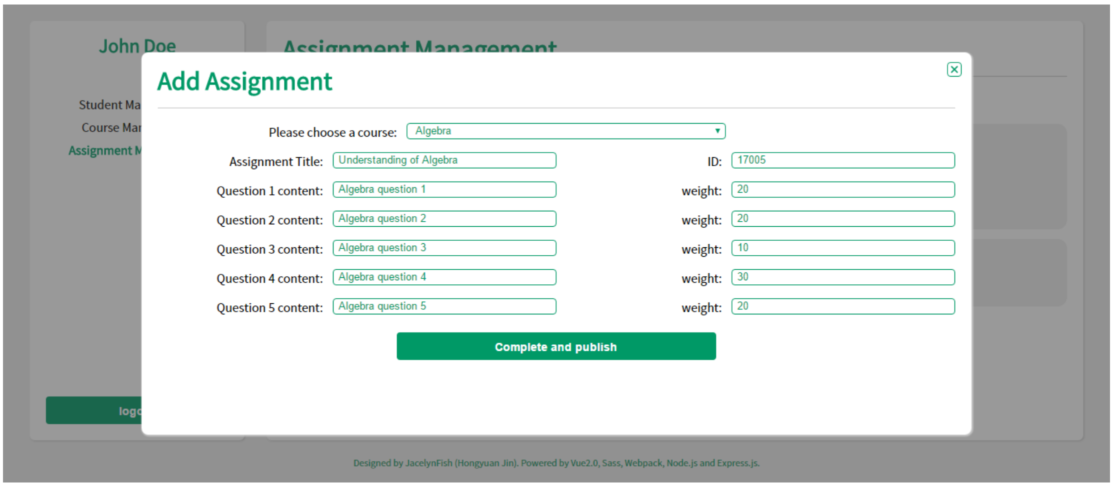


Course Management

course id: 1700501
course name: Fundamental of Algebra
select students:
15 -- Robert Downey
33 -- Mary Morstan
1701 -- Justin Timberlake
1702 -- Amanda Seyfried

Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.

In Assignment management, if you want to add an assignment, click the "add assignment" button and an editing panel will show. Choose the course you want to add the assignment into and edit the basic info of the assignment.


John Doe
Assignment Management
Student Ma
Course Mar
Assignment M
logo
Complete and publish
Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.

Add Assignment

Please choose a course: Algebra

Assignment Title: Understanding of Algebra ID: 17005

Question 1 content: Algebra question 1 weight: 20

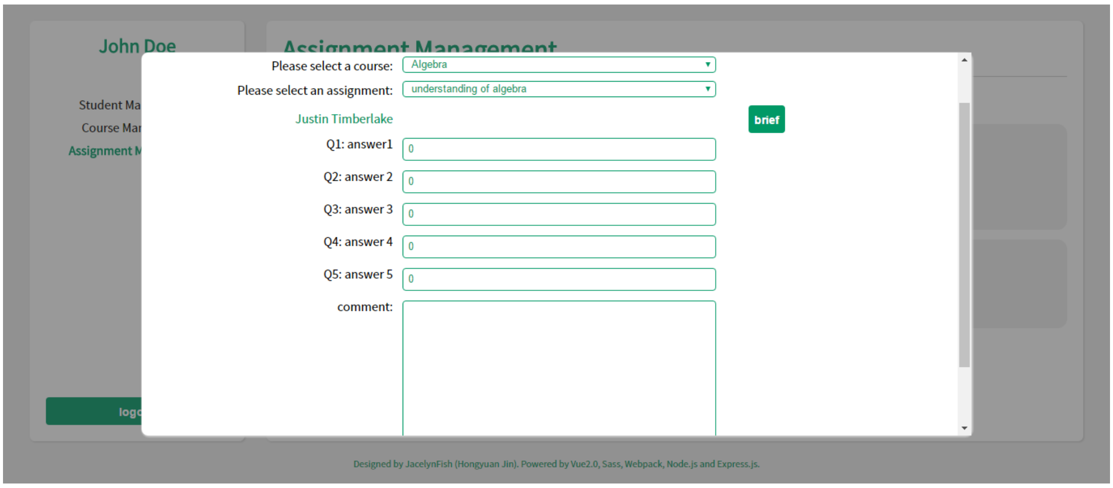
Question 2 content: Algebra question 2 weight: 20

Question 3 content: Algebra question 3 weight: 10

Question 4 content: Algebra question 4 weight: 30

Question 5 content: Algebra question 5 weight: 20

If you want to mark the assignments submitted, simple click the "mark" assignment button and choose the preferred assignments. A list of submitted assignments will show and there is a "detail" button which can be clicked to be marked or check marking details. If an assignment has already been graded, a "marked" tag will show.


John Doe
Assignment Management
Student Ma
Course Mar
Assignment M
logo
brief
Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.

Please select a course: Algebra

Please select an assignment: understanding of algebra

Justin Timberlake

Q1: answer1 0

Q2: answer 2 0

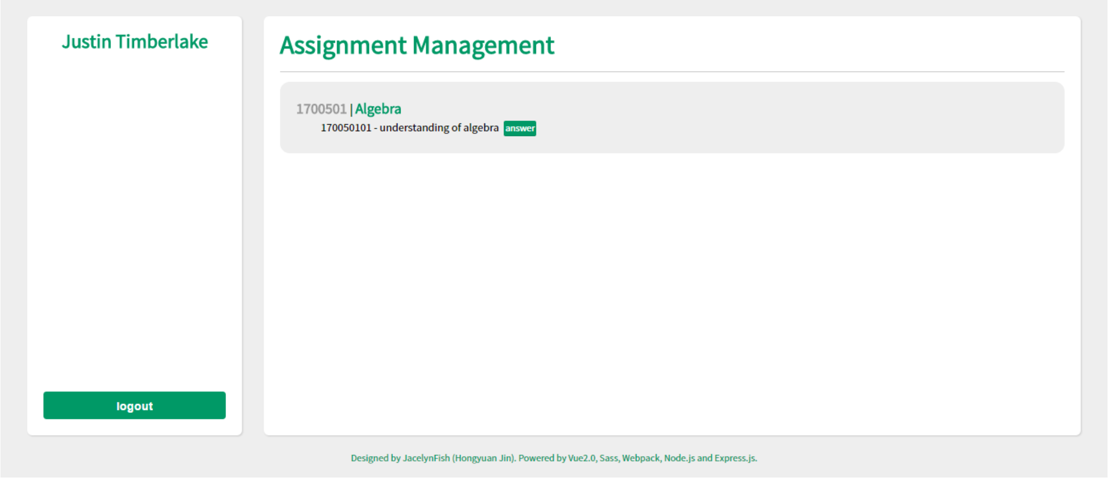
Q3: answer 3 0

Q4: answer 4 0

Q5: answer 5 0

comment:

e. For students: after logging in, the dashboard will present a list of assignments that are available. Those marked will have a "marked" tag followed and you can click the "details" button to check your marks for each question as well as the instructor's comment. Those that are to be submitted will also have an "answer" button that are used to edit the answers for this assignment and submit. After submission, a "submitted" tag will show.

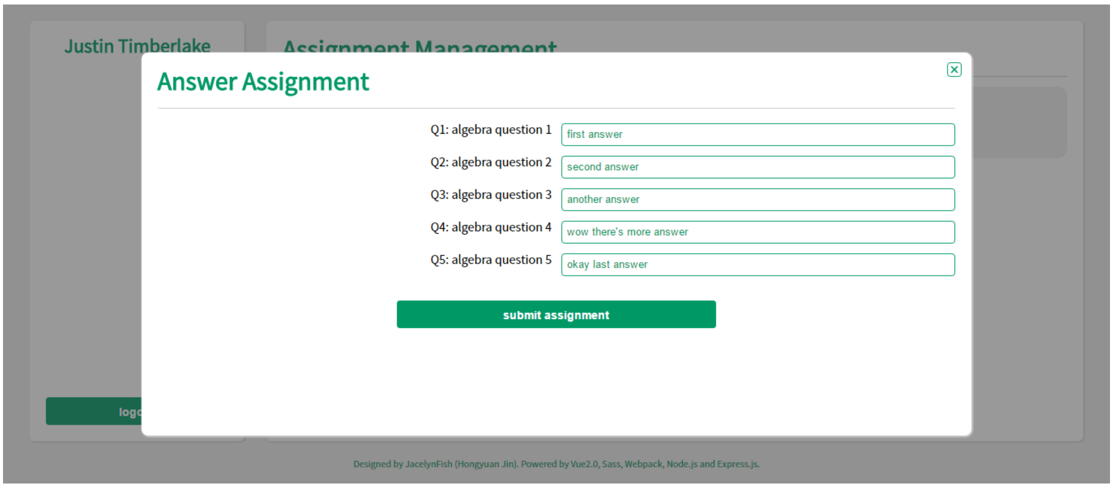

Justin Timberlake

Assignment Management

1700501 | Algebra

170050101 - understanding of algebra

Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.


Answer Assignment

Q1: algebra question 1 first answer

Q2: algebra question 2 second answer

Q3: algebra question 3 another answer

Q4: algebra question 4 wow there's more answer

Q5: algebra question 5 okay last answer

Designed by JacelynFish (Hongyuan Jin). Powered by Vue2.0, Sass, Webpack, Node.js and Express.js.

5. Query description

The query used in the system are straight-forward and easy for understanding. For the creation of the database and its tables, please check out the file “table_creation_query.sql”. For data query and update, examples are as the following:

/getStuAsmInfo: I joined the CourseInfo, Assignment, StuAssignment tables together to get a course that a student takes and its assignment. The information includes assignment names, five questions, answers, weights and scores respectively, submitting and marking Boolean info as well as the instructor’s comment.

```
SELECT *
FROM CourseInfo AS CI
JOIN Assignment AS A
ON CI.cid = A.cid
JOIN StuAssignment AS SA
ON A.id = SA.aid AND SA.sid = ${sid}
WHERE CI.cid = ${cid} AND CI.sid = ${sid}
```

/updateScore: for update the score information of a StuAssignment, I find the assignment ID as well as the student ID matched and update the related information.

```
UPDATE StuAssignment
SET comment = '${data.comment}', marked = ${data.marked}, Q1_score =
${data.Q1_score},Q2_score = ${data.Q2_score},
Q3_score = ${data.Q3_score},Q4_score = ${data.Q4_score},Q5_score =
${data.Q5_score}, final = ${data.final}
WHERE aid = ${data.aid} AND sid=${data.sid}
```

Other queries are just simple INSERT, UPDATE and SELECT operations.

6. Conclusion

Mini Moodle is a web-based online grading system powered as a Single Page Application. With a user-friendly and carefully styled interface, it can efficiently help both teachers and students to manage their courses and assignment. In the future optimization, I will continue to better its function in identity authentication as well as data validation, and add more practical utilities such as file upload and students' comment on the course.

Github repository: https://github.com/jacelynfish/mini_moodle