User Documentation
Sprint Release 6

Student View

Loading the Canvas Assignment

Once the application is running on your device and the development setup has been done, you can navigate to the Canvas assignments page, select **CodingExam Noah Assignment** which will load the **CodingExam** application and pull the exam questions already existing in the database. If the user is a student, they should be presented with four questions and a submit button. The four question types are a multiple-choice question with 4 answer options, a true-false question with a true or false answer option, a text-box entry question, and a code editor question. The screen should look like the screenshot below.

What's the best programming language?

○ C#

○ Java

○ TypeScript

○ Fortran

Computer Science is dope.

○ False

○ True

How do you feel today?

Print "Hello World" in the language

```
1    undefined
```

Multiple Choice Component

The user will be presented with a question and four different options for them to select. Once selected, the component should look like the screenshot below.

True False Choice Component

The user will be presented with a question and then true or false options for them to select. Once selected, the component should look like the screenshot below.

Short Answer Component

The user will be presented with a question prompt and a blank text entry box to answer the question with. Once text is entered, the component should look like the screenshot below.

Code Editor Component

The second to last question visible should be a code editor question. This will allow for the user to write only Java code and have syntax highlighting for it appear. The user is not able to compile the code with the component. Once code has been entered the syntax highlighting should look like the screenshot below.

Print "Hello World" in the language

```
1   System.out.println("Hello World")
```

Parson's Problems Component

The last question visible should be the parson's problem, the user should see a blue **Submit** button that will save and submit their answers to the database. The parson's problems code snippets should look like the screenshot below.

Print "Hello World" 3 times

Drag from here   Construct your solution here

```
for(int i; i< 3; i++){

print("Hello");

print("World");

}
```

Submit Button Component

At the end of the CodingExam Application, the user should see a blue **Submit** button that will save and submit their answers to the database. The button should look like the screenshot below.

Submit

The user can select an answer and hit the **Submit** button to submit their answer to the server. The application will then display a message at the top of the screen showing the success of the submission to the database. It should appear like the screenshot below.

Valid submission

Reloading the CodingExam Application

Once the user has submitted their response by clicking the **Submit** button, the data should be saved and stored in the database. Upon reloading the webpage, the page should repopulate the questions with their responses saved in the database. Each question should be disabled, and feedback boxes should appear

Instructor View

Loading the Application

When the user is an instructor and opens the application on a particular assignment, they will be greeted with a screen showing a list of the students who have submitted the assignment. The assignment from the Student section will be used for this section. The screen should look like the screenshot below.

Clicking a Student from the List

When a student's name is clicked on by the user, the application will load the student's exam submission and display it for the instructor. The following screenshots will be based on the **Test User** student. For the exam from the student section of this document, the screen loaded by the application should look like the screenshot below.

What's the best programming language?

- ○ C#
- ○ Java
- ● TypeScript
- ○ Fortran

Computer Science is dope.

- ● False
- ○ True

Clicking the Back Button

The back button at the top of the page can be clicked to return to the list of students for the current exam. Note that any feedback entered into the boxes on the page will not be saved when the back button is clicked. The button should look like the screenshot below.

Back

Leaving Feedback

Each question loaded in the instructor view should also have a text box loaded below it. This text box is for leaving feedback. The instructor can write feedback for each of the questions the student answered in the appropriate box below the question. If feedback has already been left, the boxes should contain the feedback that was left, and if feedback has not been left yet, the boxes should appear blank. A question and feedback box should look like the screenshots below if filled in and not filled in respectively.

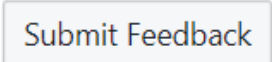What's the best programming language?

- ○ C#
- ○ Java
- ● TypeScript
- ○ Fortran

feedback for question 1

What's the best programming language?

- ○ C#
- ○ Java
- ● TypeScript
- ○ Fortran

Submit Feedback Button

Once the instructor has left the feedback they wish to leave, the submit feedback button can be clicked to submit the feedback to the database. After clicking the button, the application's screen should return to the initial student list view. The button should look like the screenshot below
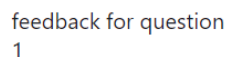

Viewing Given Feedback

After the instructor has left feedback for a particular student's exam, the same student can be clicked on again to view the feedback that was left. Note that currently, only instructors can view feedback, not students, but this will change in future iterations. The loaded feedback should look like the screenshot below.

Creating an Exam

The instructor can now create an exam by clicking the "Create Exam" button in the base view. This will present them with the following view, where they can create each question and see a preview of the exam.

From here, clicking the "+" will present the user with a list of question options to create, as follows.

Creating a Multiple-Choice Question

After clicking the "MultipleChoice" option, the user will be prompted with the following. Here, they can enter the question text and add answer choices, as well as specify the correct answer index (zero-based). Once they click the "Add" button, a preview of the exam will show.

Back

Question Text

Test

Answer Choices

TestA

TestB

TestC

TestD

+   —

Index of Correct Answer

3

Add

Question 2   —

Test

○ TestA

○ TestB

○ TestC

○ TestD

Creating a True False Question

After clicking the "TrueFalse" option, the user will be prompted with the following. Here, they can enter the question text and specify "0" for False and "1" for true. Once they click the "Add" button, a preview of the exam will show.

Back

Question Text

Test

Enter 1 for True or 0 for False

1

Add

Question 2   —

Test

○ False

○ True

Creating a Short Answer Question

After clicking the "ShortAnswer" option, the user will be prompted with the following. Here, they can enter the question text. Once they click the "Add" button, a preview of the exam will show.

Back

Question Text

Test

Add

Question 1   —

Test

Creating a Coding Answer Question

After clicking the "CodingAnswer" option, the user will be prompted with the following. Here, they can enter the question text. They can also select the coding language they want the answer to be written in. Once they click the "Add" button, a preview of the exam will show.

Question Text

Test

Select language...

Add

Cancel

Question 3

—

Test

1 |

Creating a Parson's Problems Question

After clicking the "Parsons Problems" option, the user will be prompted with the following. Here, they can enter the question text. They can also enter the answer choices, as well as the correct order of elements, in numerical form, as they should be graded. Once they click the "Add" button, a preview of the exam will show.

Question Text

Answer Choices

+   −

Correct Order of Elements (enter in the format ###)

Add

Cancel

Question 1   −

Test

Drag from here   Construct your solution here

```
print(Hello)
```
```
print(World)
```

Cancelling a Question

At any time, the user can click the "Cancel" button at the bottom-middle of a question, which will delete the question before it is added to the exam.

Cancel

Deleting a Question

At any time, the user can click the red "-" to the upper-right of a question, which will delete the question from the exam.

−