

# Sugarizer: Expanding Exerciser activity

## 1. Basic Details

**Full name** - Temirzhan Yussupov

**Email** - [temirzhan.yussupov@nu.edu.kz](mailto:temirzhan.yussupov@nu.edu.kz)

**Github** - <https://github.com/ironsoul0>

**Personal website** - <https://ironsoul.me>

**First language** - Russian

**Location** - Nur-Sultan, Kazakhstan

**Timezone** - GMT+6

## 2. Contributions to SugarLabs

I have studied the code of [ExerciserReact repository](#) and was able to find some bugs and open three pull requests.

### [PR 1](#)

Since the main logic of any React/Redux app is written in reducers, the source code of reducers was the first thing I started to read. I found one, not so serious bug that in the future has the potential to become a serious problem. Cloning objects in JavaScript is a tricky thing and it should be done very accurately since misunderstanding the concept of cloning may lead to serious problems which are very difficult to tackle. To be more

precise, I found a line of code in which the previous developer did not create a shallow copy of an object but just assigned it not a new variable. The following is an exact GitHub commit I made:

```
2 src/store/reducers/exercises.js
@@ -17,7 +17,7 @@ const exercises = (state = DEFAULT_STATE, actions) => {
17 17     case ADD_SCORE_TIME:
18 18     return state.map((exercise, i)=>{
19 19         if(exercise.id=== actions.id){
20 -         let temp= exercise;
20 +         const temp = {...exercise};
21 21         temp.scores.push(actions.score);
22 22         temp.times.push(actions.time);
23 23         return temp;
}
```

## [PR 2](#)

After running ExerciserReact app on my machine, I briefly tested all the functionality and found out that buttons in one of the UIs were not displayed at all. They for some reason are displayed properly in the deployed version of the app. To be exact, there is an image of what I got before committing changes:

### MCQ

Title of Exercise

1. Question:

1 Correct Option

2 Wrong Option

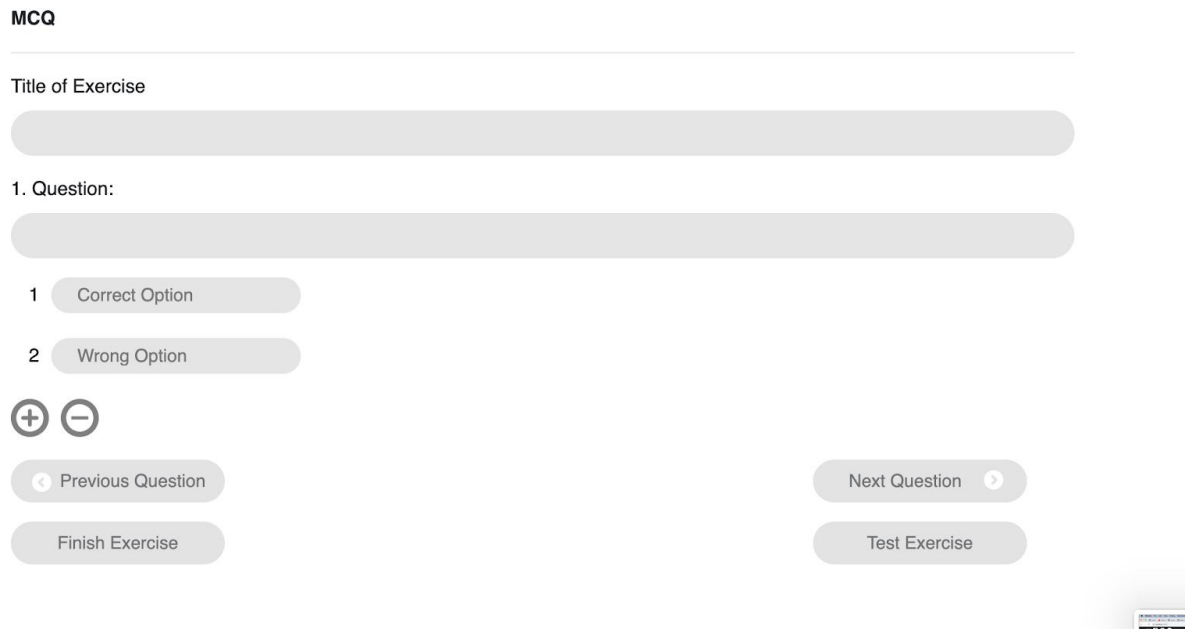
Previous Question

Next Question

Finish Exercise

Test Exercise

Actually, there should be + and - buttons for adding new options. You will see them in the following picture, a version with my committed changes. All I did is removed unnecessary bootstrap class and buttons started to be displayed as expected. Here is a result:



### [PR 3](#)

This pull request is a solution for [issue #37](#). I added additional logic that does not let a user create a situation in which the bug is being produced. The bug appeared because in the Cloze Text exercise builder there was no check for an equal amount of options and clozes. I added this logic and now it works fine. The following is a code I committed to GitHub.

```

12 src/containers/Builders/CLOZEForm.js Copy path View file
@@ -174,6 +174,18 @@ class CLOZEForm extends Component {
    174         }
    175     });
    176
    177     +     const next_blank = this.findNextBlank(this.state.clozeText);
    178     +
    179     +     for (let i = 1; i < next_blank; i++) {
    180     +         if (this.state.answers.length <= i - 1 || this.state.answers[i - 1].length === 0) {
    181     +             isValid = false;
    182     +         }
    183     +     }
    184     +
    185     +     if (next_blank - 1 !== this.state.answers.length) {
    186     +         isValid = false;
    187     +     }
    188     +
    177     189     this.setState({
    178     190         ...this.state,
    179     191         isValid: isValid
  
```

### 3. Abstract

Although powerful, Sugarizer Exerciser activity can be improved by adding new intuitive features that will make the experience of users even more enjoyable. I plan to achieve this by working on the existing code base and following the prepared timeline. It is also essential to note that integrating new features into Sugarizer should be accompanied with an adequate understanding that the product will be used by children and functionality should remain simple and intuitive as it was before.

### 4. Benefits to community

I made some research and found out that Sugarizer is used by a lot of schools in order to help their students develop their analytical skills. It is a big honour and responsibility for me to contribute to the project that is used for educating purposes. The idea description highlights the importance of Sugarizer app for Saint-Ouen schools which

use the app on a daily basis. I am now pretty sure that my work will be useful for a community since Sugarizer is a well-known and leading platform for learning.

## **5. Timeline and Project Details**

After the accepted students are announced at May 7th the “Community Bonding Period” starts. This time I will learn more about SugarLabs development structure and its tools: ReactJS, NodeJS, MongoDB, any other third-party Javascript libraries. I will try to define the details of the project and identify parts that are more difficult and will start to prepare for them.

Coding than officially begins at May 27 and ends at August 19. In this period, on June 25-28 and July 23-26, we will submit our evaluations of one another. All other time I will work on the project presented in this document. I tried to describe several phases each of which highlights some important time period of upcoming GSoC.

### **Phase 1: Feature allowing to integrate multimedia elements**

Currently, there is no possibility to ask a question via image, gif or video because everything is answered/asked through a plain text. In order to make the experience of a user more diverse and interesting, it is essential to integrate new types of multimedia elements. Also, it is noted that we should be able to import these elements from our journal.

**27 May - 2 June** - Understanding how to integrate journal content into existing React application.

**3 June - 9 June** - Designing and coding UI for drag & drop, upload image and other necessary functionality.

**10 June - 16 June** - Integrating multimedia into question builders. Adding functionality to add images, gifs while building some type of exercise.

## **Phase 2: Adding new templates**

There are now only three types of exercises users can build and solve: reorder list, cloze text, MCQ. On phase 2, I will brainstorm what kind of new templates will be suitable for Exerciser activity and will implement them.

**17 June - 23 June** - Brainstorming new possible templates and discussing them with a community.

**24 June - 30 June** - Implementing ideas that we came up together on the previous step.

**1 July - 7 July** - Testing new templates and making sure they work properly.

## **Phase 3: Improving the result screen**

What I can do here is to bring more detailed statistics for the shared type of game. Now I think of integrating pie charts, histograms and also define a function which takes into

account the time of finishing the exercise, the number of correct answers. Using this function, I will be able to more accurately range users based on their performance in the shared game.

**8 July - 14 July** - Understanding what kind of additional libraries will be needed for drawing pie charts, histograms. Learning how to use these libraries.

**15 July - 21 July** - Coding UI and main functionality for displaying the results.

**22 July - 28 July** - Testing coded features and make sure they work properly in shared mode.

#### **Phase 4: Adding access to a remote library of exercises**

Since it is quite messy to store all exercises right in the code base of the app, it is much better to store them somewhere on a server and fetch when they become needed. Here I am thinking to implement REST routes using Express.JS framework in order to GET or POST exercises.

**29 July - 4 August** - Defining and documenting new routes. Possibly creating a MongoDB instance for storing exercises.

**5 August - 11 August** - Coding routes and migrate existing exercises to the newly created database.

**12 August - 19 August** - Integrating routes into Exerciser activity.

## **Phase 4: Final evaluation and code submitting**

Tidy code, writing final tests, submitting code sample and final mentor evaluation.

Dates: August 20 - 26, 2019

## **6. About me**

I am an inspired young developer who wants to bring something useful for the community. I participated in many programming camps ([LKSH](#), [N17R](#)) and had an experience of developing mobile apps, front-end and back-end parts of the website.

I believe I would fit the SugarLabs project really well. I have a solid background in Algorithms, Data structures and Javascript web development. I also worked with different technologies such as HTML, CSS, SASS, MongoDB, Bootstrap, etc.

I am really excited to participate in GSoC with SugarLabs this summer as a chance to make a positive change for thousands of users as well as to learn from great mentors.