Programming Contest Management System

Team CrayHQ: Courtney Bolivar, Ryan Stecher, Andrew Wheeler, Yash Bhutwala

Professor Lea Wittie

Department of Computer Science, Bucknell University, Lewisburg, PA

Abstract

In the past, the annual Spring Programming Contest hosted by Bucknell had a costly amount of manual work involved. As a result, the judges were forced to limit the amount of teams that could participate. Our team addressed these issues by creating a new website, designed using a MERN stack (MongoDB, ExpressJS, React and NodeJS), which provides a simple and fresh interface for judges and participants alike. Judges can create contests, upload problems, view submissions of participants, view a scoreboard as well as broadcast and send private feedback to the participants. Participants can join contests, view problems, code, test, and submit solutions as well as communicate clarification questions to the judges.

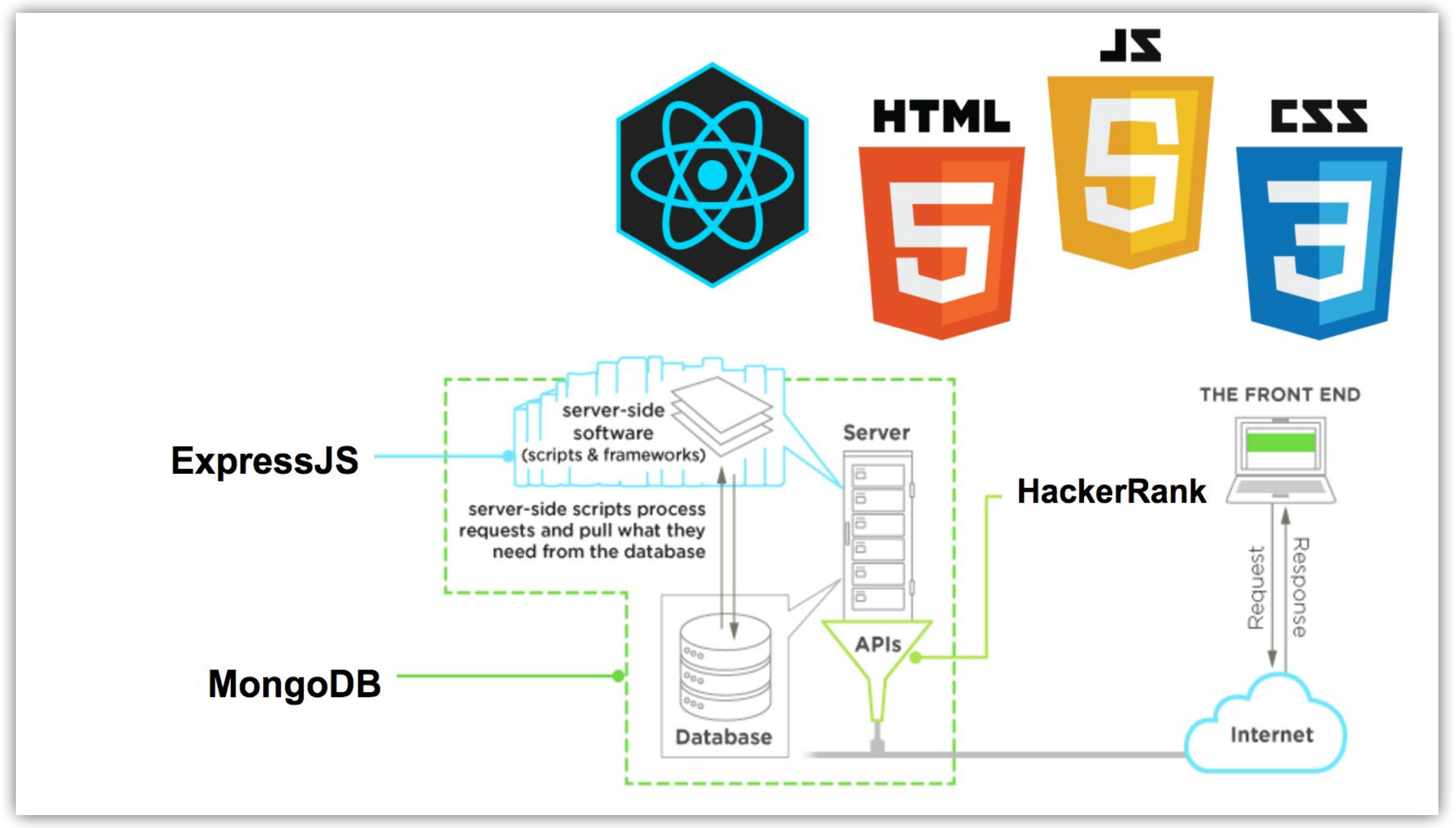
Background/Motivation

The annual Spring Programming Contest hosted by Bucknell has a costly amount of manual work involved for judges and participants alike. Because the judges have to go through the timely process of downloading, compiling and testing every submission, Bucknell is forced to limit the amount of teams that can participate. In addition, the scores are computed by hand and displayed on a whiteboard in the classroom. Our goal with this project was to give Bucknell a convenient and well-designed platform that automates these tasks and allows the organizer to communicate with the participants. As a result, judges will have an easier time running the contest and more teams will be able to participate.

System Design

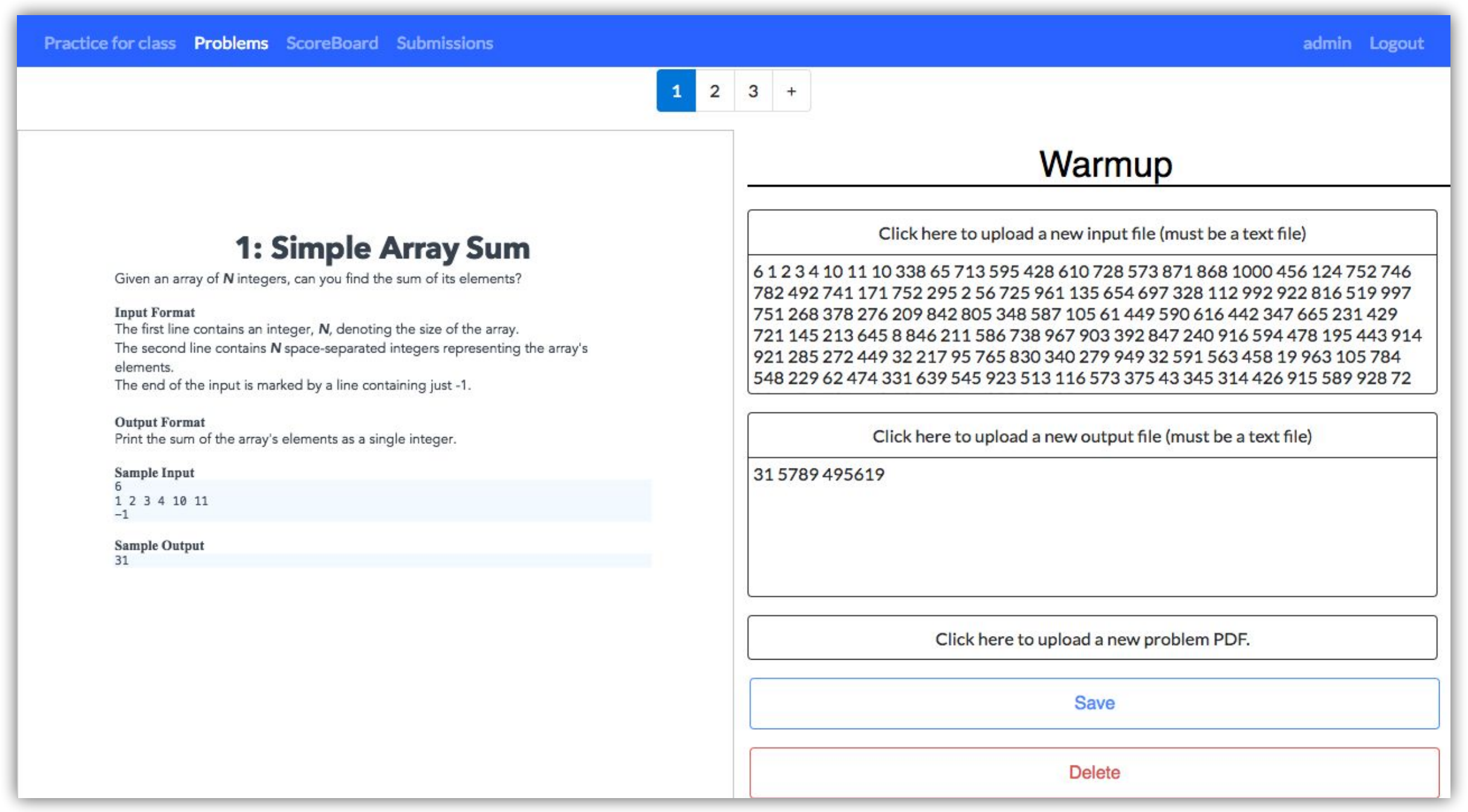
MERN Stack

- **MongoDB**: Database for maintaining users and contests
- **ExpressJS**: Backend framework
- **React**: Frontend framework built by Facebook
- **NodeJS**: Event-driven server-side JavaScript environment


HTML
JS
CSS
ExpressJS
server-side software (scripts & frameworks)
server-side scripts process requests and pull what they need from the database
Server
THE FRONT END
HackerRank
Request
Response
MongoDB
Database
APIs
Internet

https://content-static.upwork.com/blog/uploads/sites/3/2015/05/05110024/Back-end-dev-logo.png

Results



Warmup

1: Simple Array Sum

Given an array of *N* integers, can you find the sum of its elements?

Input Format
The first line contains an integer, *N*, denoting the size of the array.
The second line contains *N* space-separated integers representing the array's elements.
The end of the input is marked by a line containing just -1.

Output Format
Print the sum of the array's elements as a single integer.

Sample Input
```
6
1 2 3 4 10 11
-1
```

Sample Output
```
31
```

Click here to upload a new input file (must be a text file)

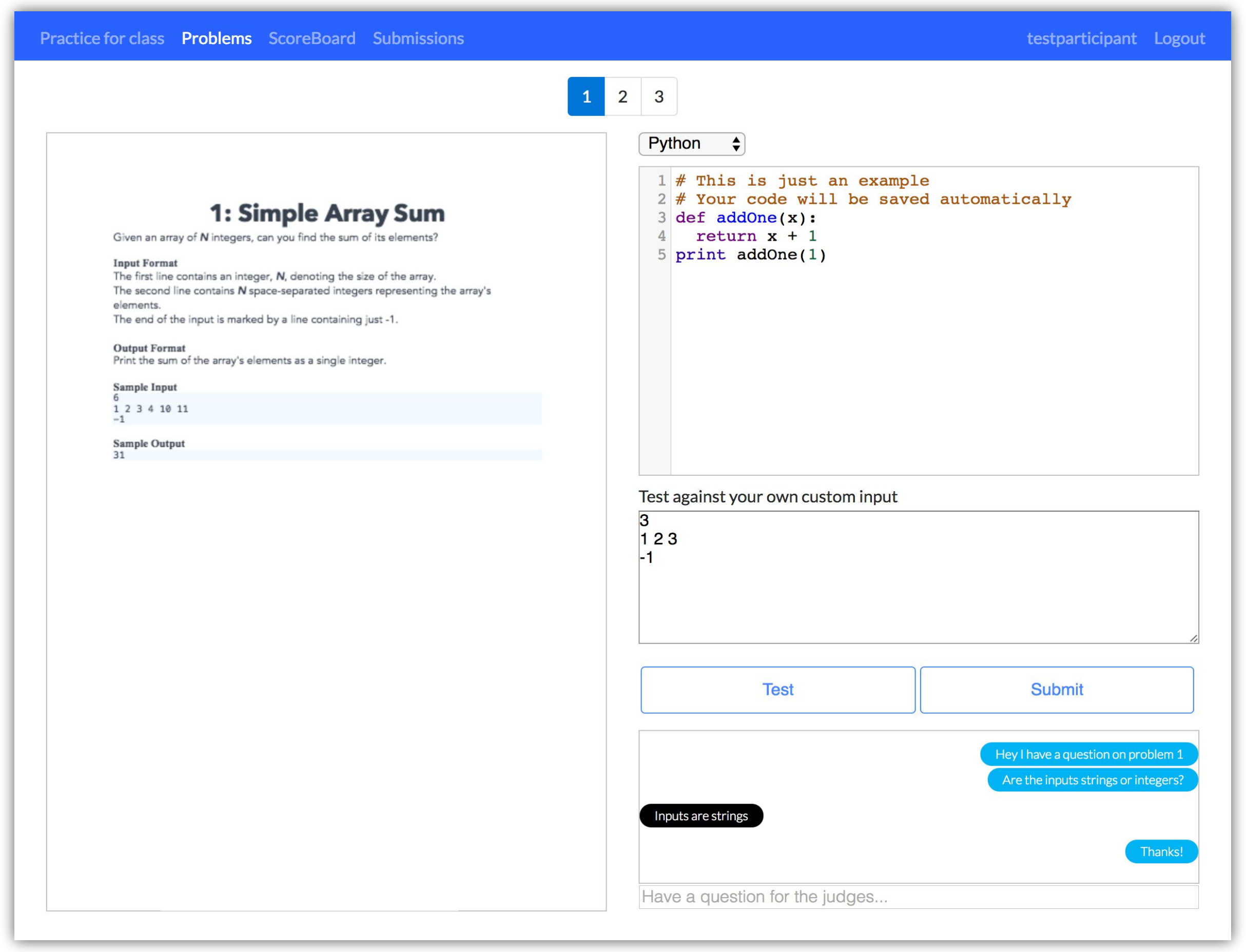
```
6 1 2 3 4 10 11 10 338 65 713 595 428 610 728 573 871 868 1000 456 124 752 746
782 492 741 171 752 295 2 56 725 961 135 654 697 328 112 992 922 816 519 997
751 268 378 276 209 842 805 348 587 105 61 449 590 616 442 347 665 231 429
721 145 213 645 8 846 211 586 738 967 903 392 847 240 916 594 478 195 443 914
921 285 272 449 32 217 95 765 830 340 279 949 32 591 563 458 19 963 105 784
548 229 62 474 331 639 545 923 513 116 573 375 43 345 314 426 915 589 928 72
```

Click here to upload a new output file (must be a text file)

```
31 5789 495619
```

Click here to upload a new problem PDF.

On this page, the admin is able to view, edit, and add problems for the contest. This contest was created with three problems, but by clicking the plus button, the admin is able to upload a PDF of a problem prompt, input for the problem, and its expected output. The expected output is used to determine correctness of submissions of participants.



1: Simple Array Sum

Given an array of *N* integers, can you find the sum of its elements?

Input Format
The first line contains an integer, *N*, denoting the size of the array.
The second line contains *N* space-separated integers representing the array's elements.
The end of the input is marked by a line containing just -1.

Output Format
Print the sum of the array's elements as a single integer.

Sample Input
```
6
1 2 3 4 10 11
-1
```

Sample Output
```
31
```

```python
# This is just an example
# Your code will be saved automatically
def addOne(x):
  return x + 1
print addOne(1)
```

Test against your own custom input

```
3
1 2 3
-1
```

Hey I have a question on problem 1

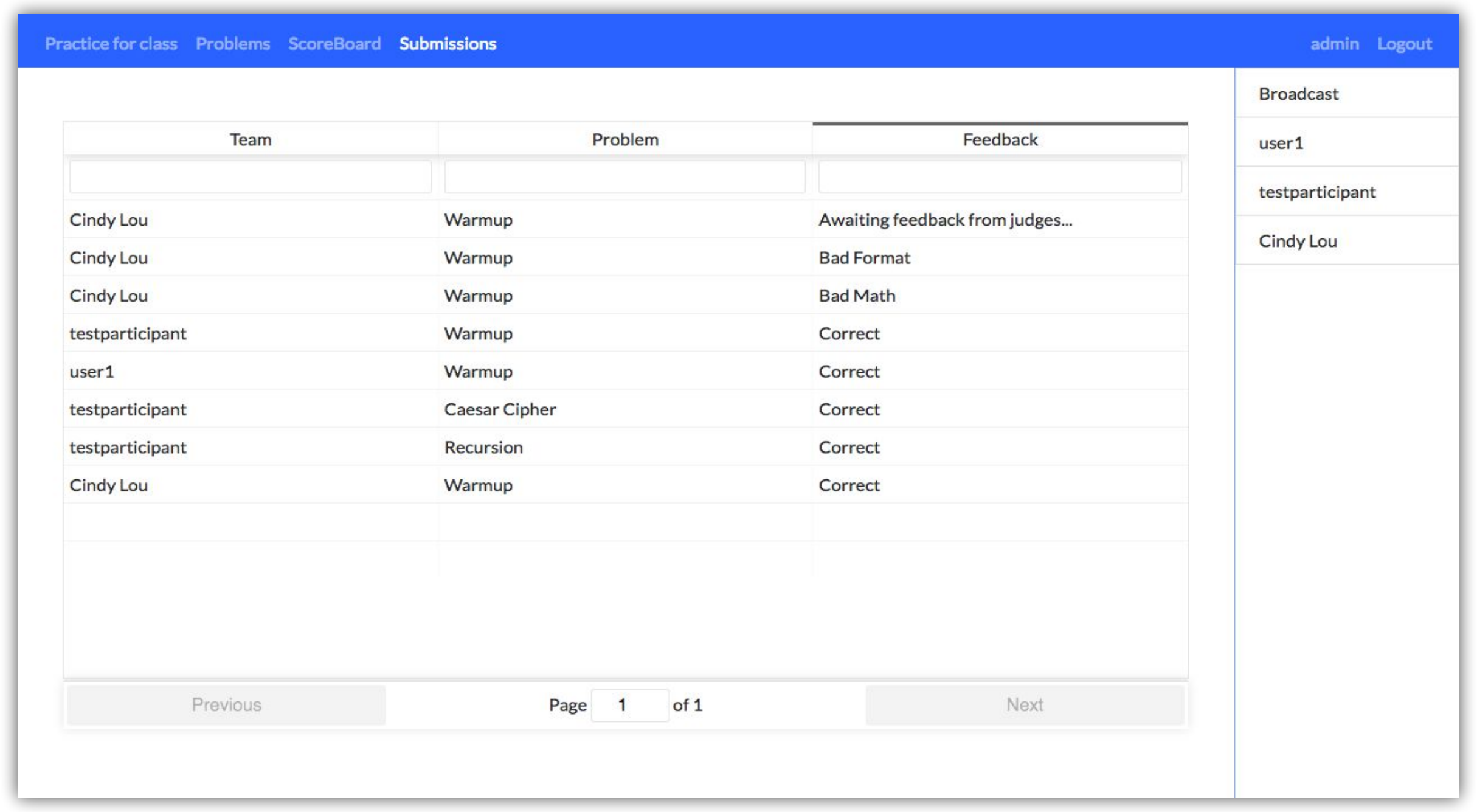
Are the inputs strings or integers?

Inputs are strings

Thanks!

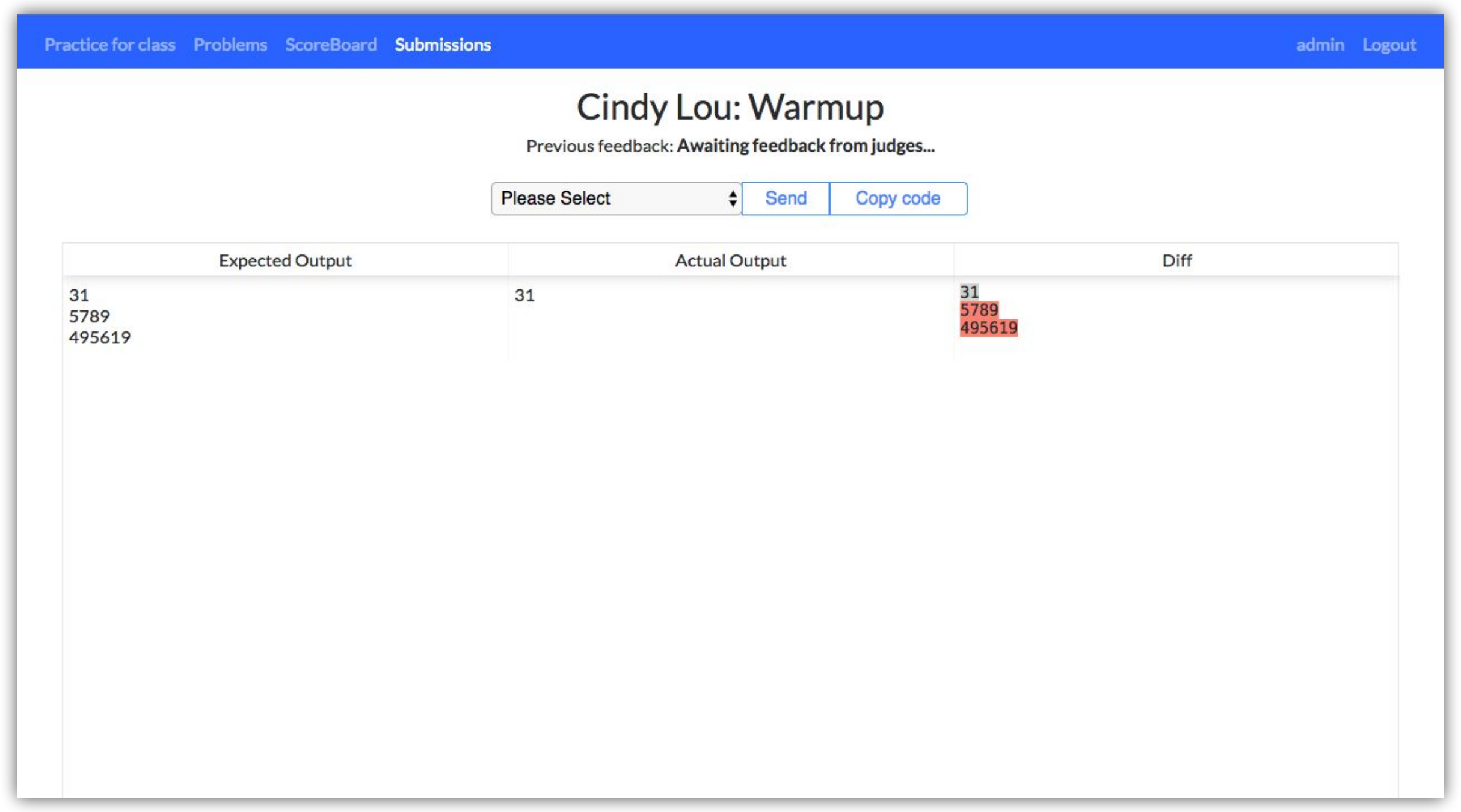
Have a question for the judges...

This is the problem page for participants where they can view a PDF of the problem, test against their own input, and submit their code. Users can switch between problems using the centered pagination bar. And finally there is a chat box in the bottom right that shows feedback for the submissions and allows them to communicate with the judges.



Team	Problem	Feedback
Cindy Lou	Warmup	Awaiting feedback from judges...
Cindy Lou	Warmup	Bad Format
Cindy Lou	Warmup	Bad Math
testparticipant	Warmup	Correct
user1	Warmup	Correct
testparticipant	Caesar Cipher	Correct
testparticipant	Recursion	Correct
Cindy Lou	Warmup	Correct

The submissions table is where the judges are able to see all of submissions in the contest and chat with the participants. The judges are able to sort and filter the submissions by user, problem, or feedback. This allows judges to only see submissions for one problem, or only see submissions with feedback 'Awaiting feedback from Judges'. By clicking on a row in the table, the judge is taken to individual submissions page.



Cindy Lou: Warmup

Previous feedback: **Awaiting feedback from judges...**

Expected Output	Actual Output	Diff
31 5789 495619	31	31 5789 495619

Judges can view individual submissions where they are shown the expected output, the actual output, and a diff between the two. The judge is also able to copy the user's code to their clipboard and use the drop-down menu to send feedback for the submission to the user. The drop-down options include: correct, bad math, bad format, compiler error and even delete submission.

Discussion

Although this website is designed for the Bucknell Programming Competition, it has the potential to be used in other places as well such as in the classroom. Professors can upload practice problems or homework, give feedback for the submissions, and help students in real time via the chat client. Our project is open source and shared on GitHub. Also, students interested in working with the MERN stack can look at our project for examples and anyone can contribute to our repository.

Conclusion

As a whole, future programming contests will benefit from our system. The judges will have an easier time running contests, and the participants will be able to focus more on the competition. Additionally, more students will have the opportunity to participate in the programming competition.

References

- Mern Starter: MERN stack boilerplate
- HackerRank: Handles secure compilation of code from 54 languages, and sends results from standard output.
- CodeMirror: Text editor interface for the web which handles syntax highlighting
- bcrypt: Library to hash passwords and facilitate user authentication
- Chart-js: Displays scoreboard information
- simple-react-pdf: Displays PDFs on the web
- react-table: Library for displaying tables with filters and paging

Acknowledgements

Thanks to Professor Lea Wittie for presenting us with this project and for being a great client. Also thank you to Professor Brian King for helping us successfully bring this website to completion. Lastly, thank you to our senior design class for helping test our system with real participants!