Pariksha

SOFTWARE LAB PROJECT
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

TEAM: SPARTAN BOTS

Samarth Joshi   203059008
Sanchit Gupta   203050089
 Roshan Sahu   203050048

18th November, 2020

1 Introduction

A basic quiz application allows you to conduct a quiz, but does not give the professor a clear picture about what topics students have understood thoroughly and topics in which students are lagging. And also afterquiz cribs is a quite cumbersome process. Hence our system Pariksha would provide a quiz hosting as well as afterquiz analysis for the professor as well as the students. Thus helping the professor analyse how much the students have grasped particular topics and which topics they are lagging. And also providing afterquiz cribs for faster resolution of doubts.

2 Motivation

Due to global pandemic online education has seen an exponential increase, we need to leverage technology to test students. Any test taking application should allow the professor to host a quiz which the students can take.
And after the quiz, analysis should be provided to both student and professor which contains details about various attributes related to the quiz. It is visualized using graphs and charts from which professors can make judgement about the students' understanding about the topic.
For professors, the dashboard would allow him to analyse students understanding about the topics using per quiz charts/graphs, which will give detailed description of the performance of enrolled students. Also the professor will be able to address the cribs for the quiz.
For students, they could take quiz hosted by professors. The student can raise doubts in afterquiz cribs session. Students can view their performance in all the quiz.

3 Dependencies

- Python3
- Django

4 Using Pariksha

Pariksha is very easy to use platform. You need to have installed the dependencies.
To install python3 on ubuntu, run the following command in your terminal.

- ***sudo apt-get install python3.8***

To install Django on ubuntu, run the following command in your terminal.

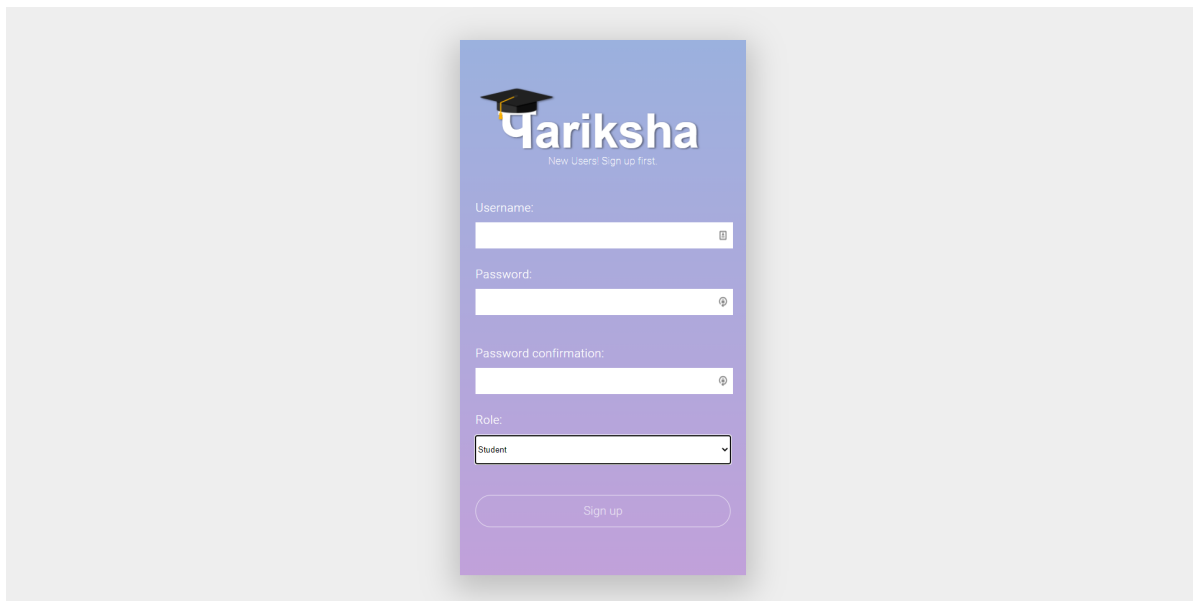- ***sudo apt-get install python3-django***

One needs to git clone the project. After that open terminal in main directory and run following commands.

- ***python3 manage.py makemigrations***
- ***python3 manage.py migrate***
- ***python3 manage.py runserver***

All the functionalities and how to use them is depicted ahead.

4.1 Student

The student first needs to sign up.

Figure 1: Sign-up Page

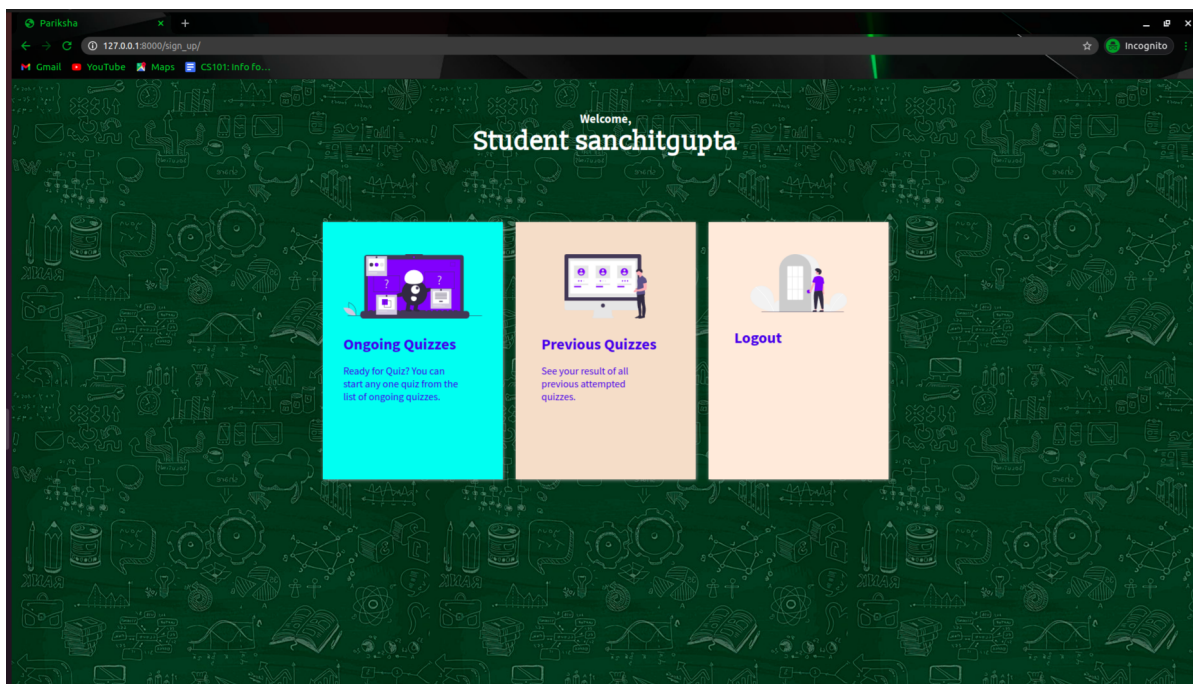After successfully signing up student redirected to Student homepage.

Figure 2: Student Dashboard

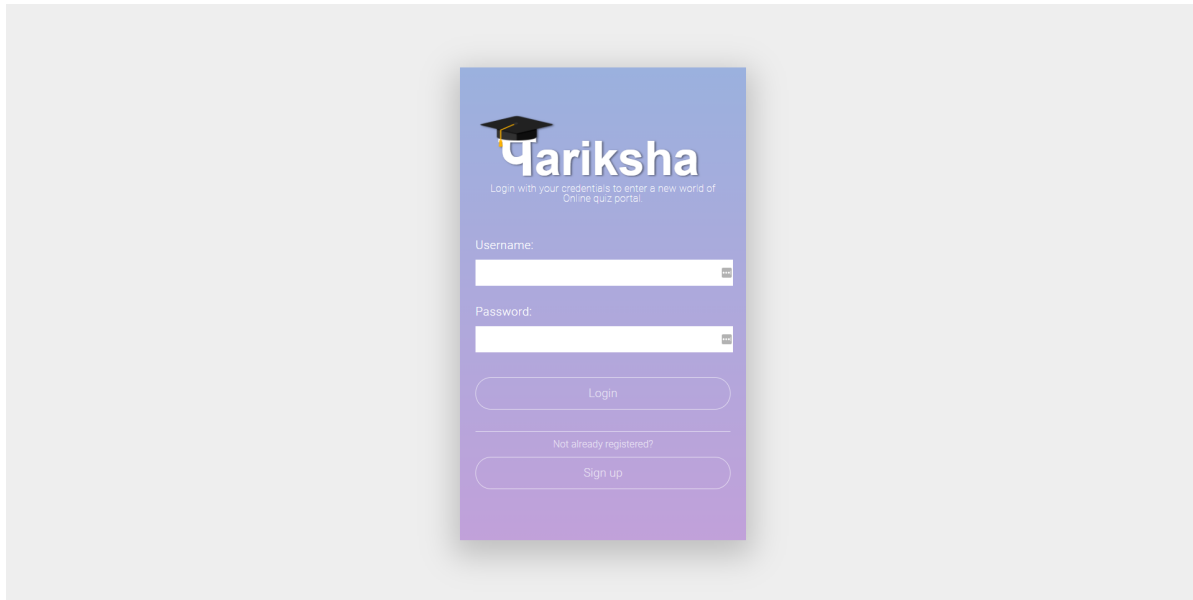Next time a student only needs to login with previously using credentials.

Figure 3: Login Page

A student can select from the list of ongoing quizzes the quiz of his/her choice.

ALL Quizzes

Show 10 entries   Search:

Code	Quiz Name	Organized By	Date	Time	Duration	Status	Attempt quiz
AX5L05	CS744.MIDSEM	roshan	Nov. 15, 2020	9:41 a.m.	60.0 minutes	Ended	-
HIKWQ4	CS699midsem	roshan	Nov. 7, 2020	3:46 p.m.	20.0 minutes	Ended	-
6KLBYJ	CS699.SE	roshan	Nov. 7, 2020	8:09 a.m.	20000.0 minutes	Ongoing	Start Quiz

Showing 1 to 3 of 3 entries   Previous 1 Next

Figure 4: Ongoing Quizzes

The quiz starts at the scheduled time and a timer is started and students can attempt the quiz till the timer ends which will auto submit the quiz.
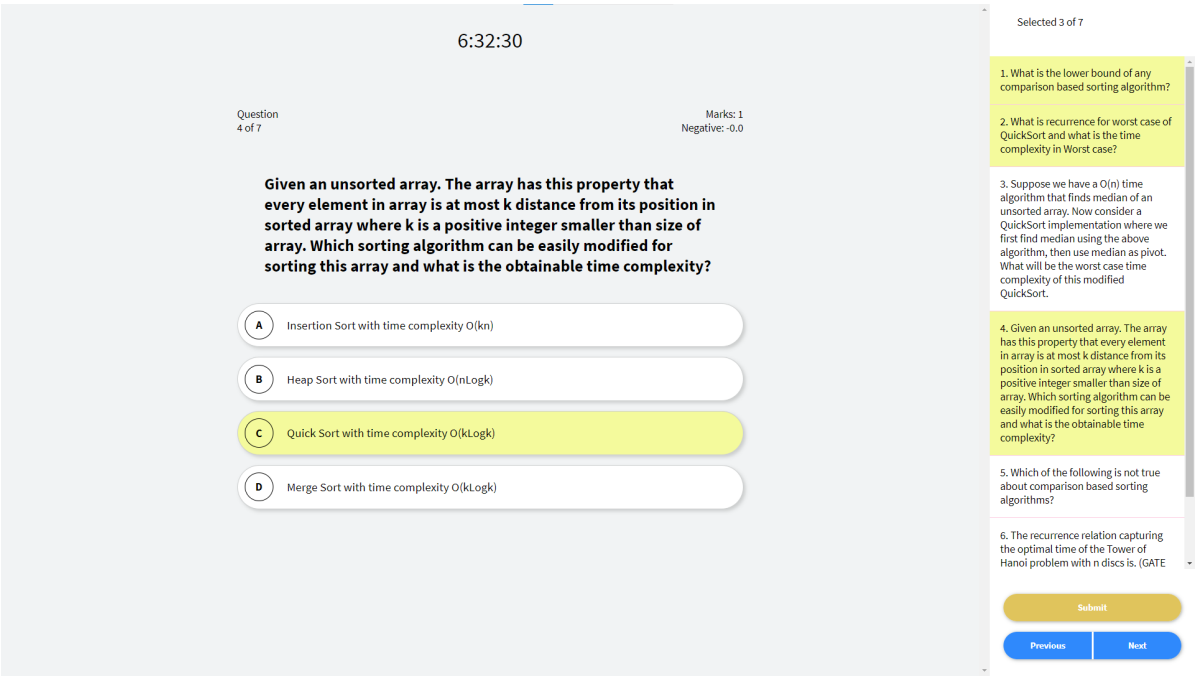

6:32:30

Question
4 of 7

Marks: 1
Negative: -0.0

Given an unsorted array. The array has this property that every element in array is at most k distance from its position in sorted array where k is a positive integer smaller than size of array. Which sorting algorithm can be easily modified for sorting this array and what is the obtainable time complexity?

A Insertion Sort with time complexity O(kn)

B Heap Sort with time complexity O(nLogk)

C Quick Sort with time complexity O(kLogk)

D Merge Sort with time complexity O(kLogk)

1. What is the lower bound of any comparison based sorting algorithm?

2. What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?

3. Suppose we have a O(n) time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort.

4. Given an unsorted array. The array has this property that every element in array is at most k distance from its position in sorted array where k is a positive integer smaller than size of array. Which sorting algorithm can be easily modified for sorting this array and what is the obtainable time complexity?

5. Which of the following is not true about comparison based sorting algorithms?

6. The recurrence relation capturing the optimal time of the Tower of Hanoi problem with n discs is. (GATE

Submit

Previous   Next

Figure 5: Ongoing Quizzes

A student can also view details of marks obtained in the previous quizzes from homepage.

100/100

1. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.pop(1)?

A [3, 4, 5, 20, 5, 25, 1, 3]

B [1, 3, 3, 4, 5, 5, 20, 25]

C [3, 4, 5, 20, 5, 25, 3]

D [3, 5, 20, 5, 25, 1, 3]

You selected: C
Correct answer: C

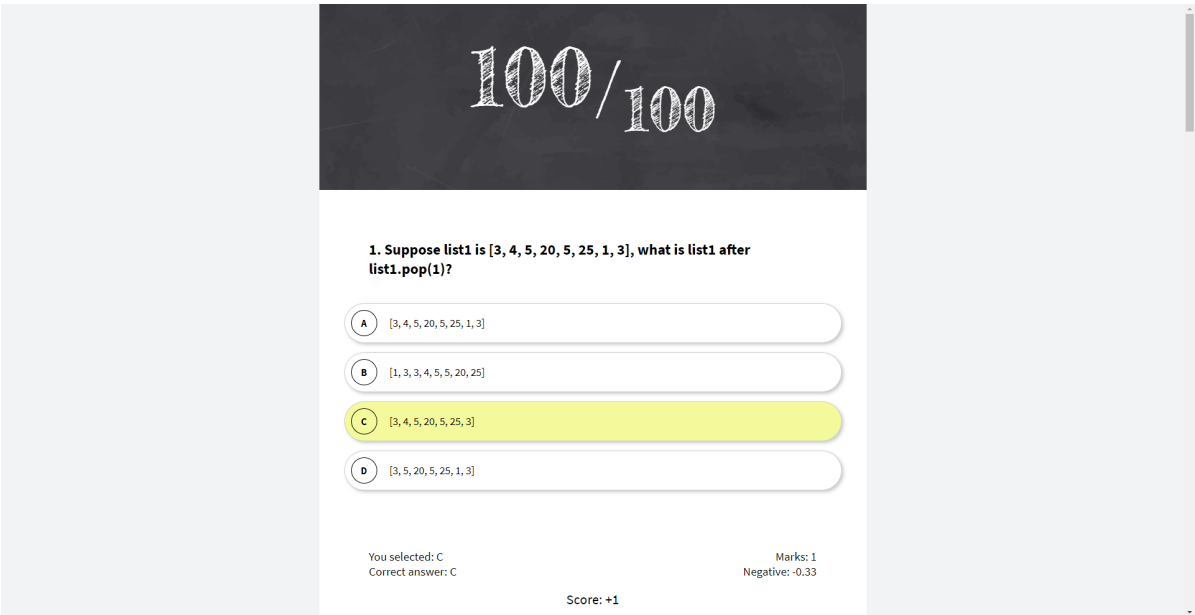Marks: 1
Negative: -0.33

Score: +1

Figure 6: Previous Submissions

4.2 Professor

A professor needs to sign up first as a professor. After successfully signing-up the professor redirect to professor homepage.
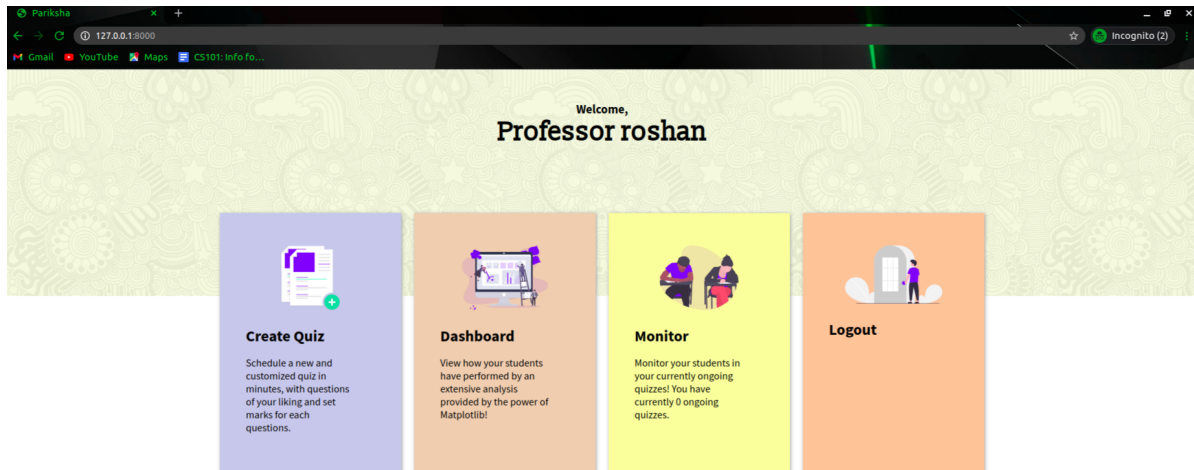

Figure 7: Professor Dashboard

Professor can create a new quiz by clicking on Create quiz in homepage. A professor can add questions in the quiz in two ways.
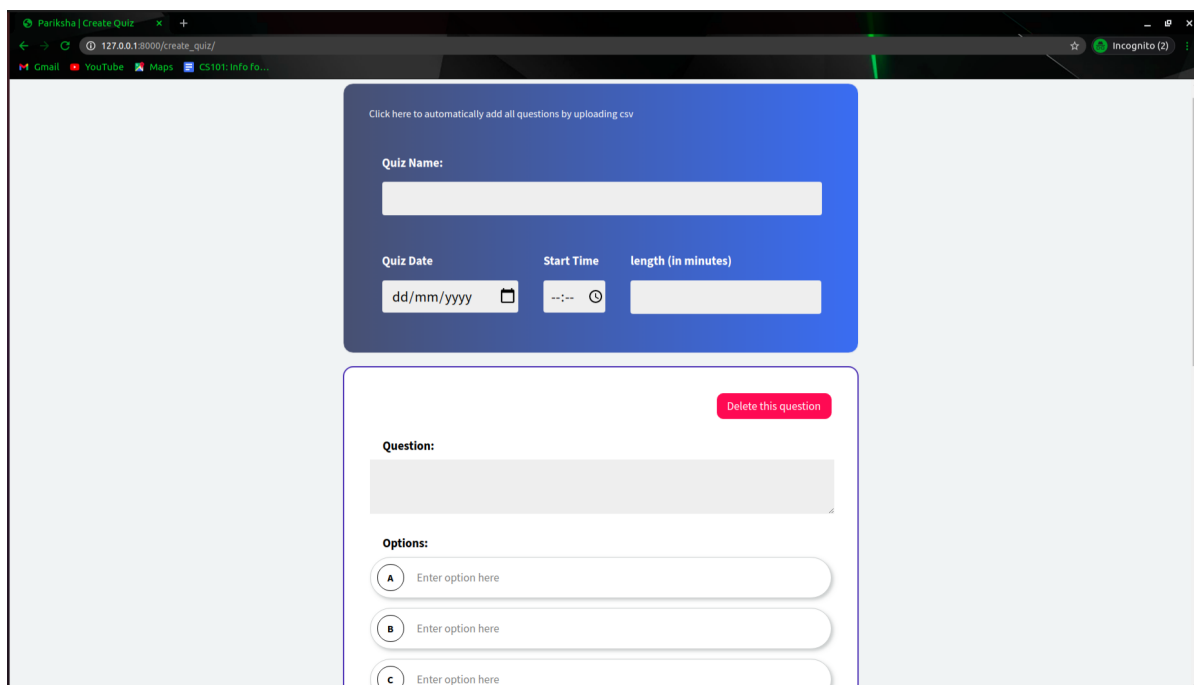1- Manually

Figure 8: Adding quiz questions manually

2- By uploading CSV file

Figure 9: Adding quiz questions using CSV

Professor can view how its students have performed in previous quizzes by an extensive analysis using various graphs. Professor can also see corresponding cribs of students in that quiz.
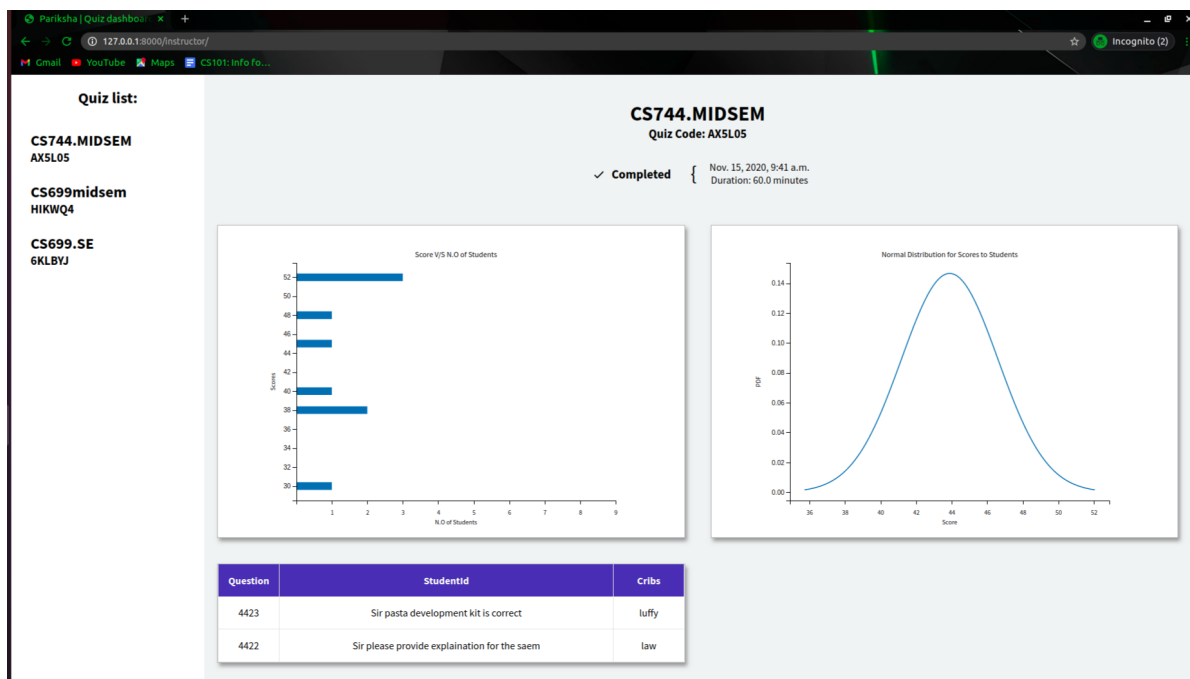

Figure 10: Analysis Graphs and Cribs Table

Professor can monitor its students in his/her currently ongoing quizzes.

Figure 11: Monitor Page

5   Future Scope

- The questions could be provided with LaTeX editor.

- Functionality to support numeric answer type.

- Video proctoring of the student.