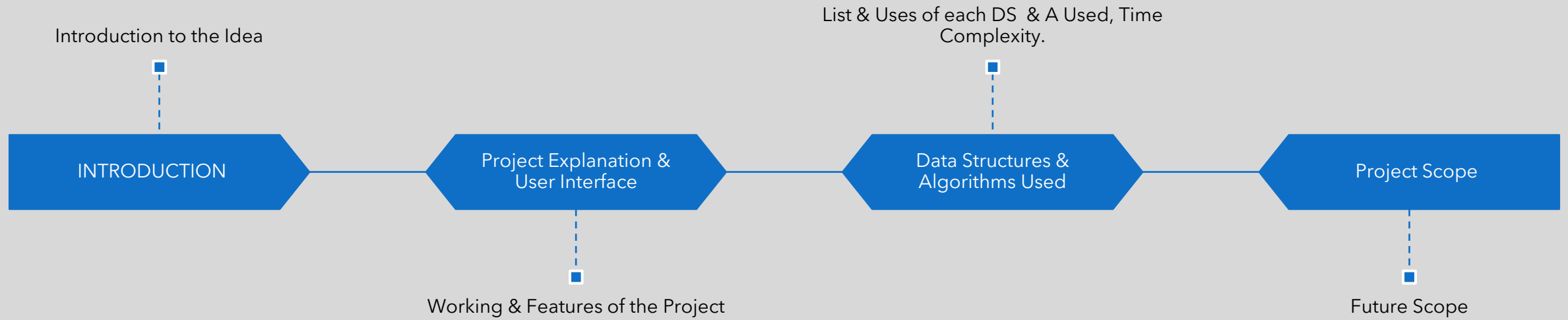




INTEGRATED QUIZ PLATFORM

DSA II PROJECT
112003133

CONTENTS



INTRODUCTION

- The idea of the project is to provide a platform for **integrated development** in the subjects of Computer Science to the users through the command line and help them obtain knowledge on a range of topics through an interactive quiz.
- The main portion of the project is dedicated to the quiz and its interactive and engaging nature. Including that the program will also include Sign IN/Log IN option and it will save the progress for a unique user, letting them continue where they left off.
- The quiz will be designed specifically for the user in such a way that it will prompt him for the answer and the quiz will become harder if you get the question right, and it will become easier if you get it wrong. Solution will be provided instantly after you get the question wrong.
- An algorithm will also be designed to further calculate the preparedness level of the user and based on that level the next questions will be divided into further difficulty tiers. The user will be prompted further from these tiers.
- The storage of questions and the creation of the quiz will be done through Hash Tables, Priority Queue(PQ) respectively.

USER INTERFACE & FEATURES

- The User Interface or the Surface Level of the Project will consist of a command line interface, which will prompt you for a Sign IN/Log IN option and after you have done that, you will enter the Quiz Platform.
- The Quiz Platform will consist of a Progress Bar, Quiz Topics, and Solutions Review, Reset Function.
- **Progress Bar** : The Progress Bar will tell you which subject quizzes you have completed, and which you have left, and how many questions of the question bank have you solved correctly. (To Complete the Quiz, you have to solve all 10 questions (Correctly/Incorrectly).)
- **Quiz Topics** : The topics which are available will be displayed in this section, and you will choose one topic for the quiz.
- **Solutions Review** : The Solutions Review will give you the answer to the questions you have solved correctly, for further revision. The solutions will be stored in another Hash Table similar to the questions.
- **Reset** : You can also reset all your progress as a user.

PROJECT WORKING 1

- **Questions** : The questions will be stored in Hash Table, each questions unique Hash Key will be designed based on the Subject/Topic, Difficulty(1-19) and the nth question of that Subject & Difficulty.
- **Quiz** : The quiz will be created on the spot, by taking random questions from each difficulty from the hash table and making priority queue of questions where the priority will be based on difficulty. Priority of the question will be based on the difference between the difficulties of the initial question(10) and the question in consideration.
- The questions of the quiz will start at the absolute value of least priority ($P = 0$) and will deque the priority queue return the question and based on if the question is correct or wrong it will perform the following algorithm.
- **Correct** - It will take the absolute value of priority and only return the least priority among the questions which have a positive priority.

PROJECT WORKING 1

- **Wrong** - It will take the absolute value of priority and only return the least priority among the questions which have a negative priority.
- The **PQ** will be designed in such a way that you will get a harder question if you get the previous one right, and an easier one if you get the question wrong.
- A separate algorithm will also be designed to restructure the **PQ** based on the preparedness level of the user, which will be calculated after a few questions. So the **PQ** will be restructured to suit the user, which will make the quiz more engaging, which is what we want in the end.
- Restructuring of the Priority Queue :-
 - 5 correct - Redesign PQ from (Difficulty : 10-19) Mid(15)
 - 5 wrong - Redesign PQ from (Difficulty : 1-10) Mid(5)
- After the user finishes the quiz properly, the progress bar will be updated and the solutions will be on display in the solutions review.

DATA STRUCTURES & ALGORITHMS USED

- **Hash Table** : Storage of questions & solutions.
- **Priority Queue**: Generate and traverse the Quiz. Efficiency can be obtained through implementation of **PQ** based on the difficulty level and number of questions available.
- **Progress Report** : A user defined structure which will keep track of the questions solved, quizzes taken, solutions, overall progress and other meaningful data for each specific user.
- **PQ Restructuring Algorithm** : Quiz restructuring algorithm designed to keep the quiz engaging.
- **Time Complexity** : Hash Tables have $O(n)$ time complexity for insertion, $O(1)$ time complexity for searching of the questions. ($n \rightarrow$ number of questions)
- Priority Queue will have a time complexity of $O(n)$ of insertion because functions will be created to balance the trees and $O(n)$ traversal time complexity for the quiz. ($n \rightarrow$ number of questions in the quiz).

PROJECT SCOPE

- The project scope is vast for example the quizzes can be expanded for a range of topics and the question bank can also be expanded to include questions from a wide range of difficulties.
- The users can also be prompted to insert questions of their choice (With Difficulty/Options) and these will be incorporated in the question bank and quiz.
- Also the application can be elevated from the command line and be given a graphical interface so the functionality and design of the app will become better.
- The Algorithms overseeing the creation, traversing, and restructuring of the Priority Queue can also be improved and fine tuned to ensure a more engaging quiz experience.
- Also the efficiency and time complexity of these algorithms can also be improved to increase the speed of the application.

REFERENCES

- <https://www.geeksforgeeks.org/quiz-game-in-c/>
- https://www.codewithc.com/quiz-game-mini-project-in-c/#google_vignette
- <https://www.geeksforgeeks.org/priority-queue-set-1-introduction/>