

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Proposal**

**On**

**IOE Entrance Preparation And Data Management Kit**

**Submitted By:**

Anish Timsina(THA077BEI007)

Bishal Khadka(THA077BEI015)

Nixon Raj Dhakal(THA077BEI028)

Sabin Acharya(THA077BEI035)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

August 2021

# ABSTRACT

Every year, thousands of students from all over Nepal contemplate to Crack IOE entrance examination to get qualified to study Engineering in Nepal’s one of the oldest and largest universities i.e., Tribhuvan University (T.U.), Institute Of Engineering (IOE). As the number of IOE aspirants is increasing annually, the level of competence has also increased drastically in the last decades. So, students need to be highly competitive and need a lot of practice to crack the entrance. Few institutions are providing classes regarding entrance preparation where a lot of students enroll each year. As the number of students in a single institution is very high, there is a lack of proper management in the administration and examination. They still make use of file systems to take a record of the enrolled students, which is a quite traditional and inefficient method for this purpose which takes a lot of time to find the data of the students in case if required. They still employ the system of paper-based exams for the MCQ test which is very time-consuming to check and publish the results.

This project aspires to fabricate a software that will be a complete IOE entrance preparation kit that will assist any institutions providing the IOE entrance preparation classes in student data and examination management. It also helps the students to boost up their preparation speed and quality providing them access to a lot of practice exams with instant results. In addition to this, this software would be able to assist all the other institutions providing the MCQs based entrance preparation classes with some minor modifications in the software. This software can also be linked with the internet to provide online exam question papers to the students so that they can practice and test themselves instantly at home which is highly effective in the current situation of Lockdown.

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards the Institute of Engineering, Tribhuvan University for the inclusion of the project in the course of computer programming in Bachelors in Electronics, Communication and Information Engineering. We are also thankful to our subject teacher Er. Saroj Shakya sir and the Department of Electronics and Computer Engineering, Thapathali Campus for providing us support which is needed for this project.

Anish Timsina [THA077BEI007]

Bishal Khadka [THA077BEI015]

Nixon Raj Dhakal [THA077BEI028]

Sabin Acharya [THA077BEI035]

# Table of Contents

[ABSTRACT i](#_Toc79309265)

[ACKNOWLEDGEMENT ii](#_Toc79309266)

[Table of Contents iii](#_Toc79309267)

[List of Figures v](#_Toc79309268)

[List of Abbreviations v](#_Toc79309269)

[1. INTRODUCTION 1](#_Toc79309270)

[1.1 Background Introduction 1](#_Toc79309271)

[1.2 Motivation 1](#_Toc79309272)

[1.3 Problem Definition 2](#_Toc79309273)

[1.4 Objectives 2](#_Toc79309274)

[2. LITERATURE REVIEW 4](#_Toc79309275)

[2.1 Work Of Existing/Researched Related Topic 4](#_Toc79309276)

[2.2 How Is It Done? 4](#_Toc79309277)

[2.3 Importance/Applications Of This Project 4](#_Toc79309278)

[2.3.1 Importance: 4](#_Toc79309279)

[2.3.2 Applications: 5](#_Toc79309280)

[2.4 Drawbacks Of Existing System 5](#_Toc79309281)

[3. PROPOSED SYSTEM ARCHITECTURE 7](#_Toc79309282)

[3.1 Block Diagram of System Architecture 7](#_Toc79309283)

[3.1.1 “Enroll Yourself” Block 7](#_Toc79309284)

[3.1.2 “Take Practice Exam” Block 7](#_Toc79309285)

[3.1.3 “Take Full Mock Test” Block 7](#_Toc79309286)

[3.1.4 “View Overall Score” Block 8](#_Toc79309287)

[3.1.5 “View Leaderboard” Block 8](#_Toc79309288)

[3.1.6 “Amend Question Set” Block 8](#_Toc79309289)

[3.1.7 “View All Enrolled Students” Block 8](#_Toc79309290)

[3.1.8 “Exit” block 8](#_Toc79309291)

[3.2 Data Flow Diagram 9](#_Toc79309292)

[3.3 Tools And Environment 9](#_Toc79309293)

[4. METHODOLOGY 10](#_Toc79309294)

[4.1 Introductory Block 10](#_Toc79309295)

[4.1.1 Algorithm Of Introductory Block 10](#_Toc79309296)

[4.1.2 Flowchart Of Introductory Block 11](#_Toc79309297)

[4.2 Option Selector Block 12](#_Toc79309298)

[4.2.1 Algorithm Of Option Selector Block 12](#_Toc79309299)

[4.2.2Flowchart Of Option Selector Block 13](#_Toc79309300)

[4.3 Enroll Yourself Block 14](#_Toc79309301)

[4.3.1 Algorithm Of Enroll Yourself Block 14](#_Toc79309302)

[4.3.2 Flowchart Of Enroll Yourself Block 15](#_Toc79309303)

[4.3.3 Flowchart of Option\_or\_exit() function. 16](#_Toc79309304)

[4.4 Take Practice Exam Block 17](#_Toc79309305)

[4.4.1 Algorithm Of Take Practice Exam Block 17](#_Toc79309306)

[4.4.2 Flowchart Of Take Practice Exam Block 18](#_Toc79309307)

[4.5 Take Full Mock Test Block 20](#_Toc79309308)

[4.5.1 Algorithm Of Take Full Mock Test Block 20](#_Toc79309309)

[4.5.2 Flowchart Of Take Full Mock Test Block 21](#_Toc79309310)

[4.6 View Overall Score Block 23](#_Toc79309311)

[4.6.1 Algorithm Of View Overall Score Block. 23](#_Toc79309312)

[4.6.2 Flowchart Of View Overall Score Block. 24](#_Toc79309313)

[4.7 View Leaderboard Block 25](#_Toc79309314)

[4.7.1 Algorithm Of View Leaderboard Block 25](#_Toc79309315)

[4.7.2 Flowchart Of View Leaderboard Block 26](#_Toc79309316)

[4.8 Amend Question Set Block 27](#_Toc79309317)

[4.8.1 Algorithm Of Amend Question Set Block 27](#_Toc79309318)

[4.8.2 Flowchart Of Amend Question Set Block 28](#_Toc79309319)

[4.9 View All Enrolled Students Block 29](#_Toc79309320)

[4.9.1 Algorithm Of View All Enrolled Students Block. 29](#_Toc79309321)

[4.9.2 Flowchart Of View All Enrolled Students Block. 30](#_Toc79309322)

[4.10 Exit Block 31](#_Toc79309323)

[4.10.1 Algorithm Of Exit Block 31](#_Toc79309324)

[4.10.2 Flowchart Of Exit Block 31](#_Toc79309325)

[5. SCOPE AND APPLICATIONS 32](#_Toc79309326)

[6. TIME ESTIMATION 33](#_Toc79309327)

[7. FEASIBILITY ANALYSIS 34](#_Toc79309328)

[7.1 Practical Uses 34](#_Toc79309329)

[7.2 Cost 34](#_Toc79309330)

[7.3 Complexity Of The Software 35](#_Toc79309331)

# List of Figures

[Figure 3- 1: Data flow Diagram Of the Project. 9](#_Toc79309580)

[Figure 4- 1: Flow-Chart of Introductoy Block. 11](#_Toc79309570)

[Figure 4- 2: Flow-Chart of Option Selector Block 13](#_Toc79309571)

[Figure 4- 3: Flow-Chat of Enroll Yourself Block 15](#_Toc79309572)

[Figure 4- 4: Flow-Chart of Option\_or\_exit() function. 16](#_Toc79309573)

[Figure 4- 5: Flow-Chart of Take Practice Exam Block. 19](#_Toc79309574)

[Figure 4- 6: Flow-Chart of Take Full Mock Test Block 22](#_Toc79309575)

[Figure 4- 7: Flow-Chart of View Overall Score Block. 24](#_Toc79309576)

[Figure 4- 8: Flow-Chart of View Leaderboard Block. 26](#_Toc79309577)

[Figure 4- 9: Flow-Chart of Ammend Question Set Function. 28](#_Toc79309578)

[Figure 4- 10: Flow-Chart of View All Enrolled Students Block. 30](#_Toc79309579)

# List of Abbreviations

# INTRODUCTION

C was initially used for system development work, in particular the programs that make up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as code written in assembly language. Thus, we used this versatile and fast language to program software for assisting students and teaching institutions regarding the IOE Entrance preparation.

## Background Introduction

The sole purpose of the IOE Entrance preparation kit is to automate the existing manual system of practicing for IOE entrance with the help of computerized equipment’s and full-fledged computer software, fulfilling their requirements listing lots of sets of questions for practice and mock test with the marks scored in their roll number. This project gives real experience for the student as if they are sitting in the IOE entrance examination. The required software is easily available and easy to work with. The entrance practice System, as described above, can lead to error-free, secure, reliable, and fast practice sessions. It can assist the user to concentrate on their other things rather than feeling nervous by thinking the way of entrance management. Thus, it will help the user in better utilization of resources and time. The aim is to help the entrance aspirants for practicing for the entrance examination with the help of computerized equipment and full-fledged computer software, fulfilling their requirements. The project gives a real experience of entrance examination.

This software revamps the system of data management and the current approach of examination (i.e., paper-based). It facilitates the institute for quick data record and access of a large number of students. This software can take the whole burden of systemized, error-free, and eased examination system for the institution taking account of the fastest result production, leadership board, and many more.

## Motivation

The idea of this IOE Entrance Preparation kit emerged from our recent experience that we acquired while we were preparing for the IOE Entrance. To get shortlisted in IOE, we need to practice as many questions as we can. The more we get familiar with the questions i.e., by practicing them, the more we gain an idea on solving the questions that we get queried in the Entrance. But on knowing these facts too, we couldn’t practice enough due to the smaller extent of question sources and the very rare number of examinations that we could attend. Also, we had a sort of fear regarding the computer-based examination, which is the format of IOE entrance as we were used to only paper-based examination systems which were obstructing our path to gain confidence.

Hence, to overcome these problems we got motivated to develop a software that could take a test based on MCQs and provide instant results of the tests. This aids a lot for the student to evaluate him/herself and build confidence in giving the actual computer-based entrance exam.

Furthermore, we thought to make this software a little bit commercialized. We thought of designing the software in such a way that it also aids the institutions that provide classes on IOE Entrance Preparation. This software would help them to keep a record of the students, examination, students’ performance, and many more.

## Problem Definition

This project aims to flourish a software that will assist any institution providing classes for IOE entrance preparation regarding enrolling students to the institution, student data management, computer-based and error-free examination system, generating the result of the examinations, analysis of the student progress, access to all student data and add practice question by the administration.

## Objectives

The main objectives of our project are listed below:

* To help IOE entrance aspirants practice effectively for the entrance examination.
* To develop a commercial type of software for institutions conducting entrance preparation classes.
* To promote paperless examination to promote Environmental Conservation.

# LITERATURE REVIEW

## 2.1 Work Of Existing/Researched Related Topic

The work of our researched topic is to provide a system for the management of the student data, an examination system for those institutes which provide the IOE entrance preparation classes, and also provide best practices for the students preparing for the IOE entrance examination.

It focuses on providing the students easy, fast and flawless methods of giving the practice exams and data of their progress so that they can evaluate their level and prepare the best strategy that suits them to crack the IOE entrance.

## How Is It Done?

This project can be accomplished by developing software that could perform all the above-mentioned tasks. To develop this software, we are using the fast and versatile computer programming language i.e., ‘C Programming language’.

This software makes maximum use of the data input and output from the files. It takes data from the students while enrolling them in the institution and then stores it in a file from where the data can be accessed for future reference. Similarly, while giving the examination, the questions and the respective answers come from a separate text file which the software analyzes and provides the result. The students’ scores and performances in the test also will be recorded in a separate file so that he/she can view it and analyze it.

This software also embeds a system that checks whether the student has already enrolled in the institution or not by comparing their details with the existing detail to make a bit more managed system.

## Importance/Applications Of This Project

### 2.3.1 Importance:

* It aids any of the institutes in keeping managed student data.
* It also assists the institutes to take a systematic examination and providing instant results to the students.
* Students can also give instant exams and get instant results.
* Students will gain the privilege to check their scores and see how they are performing day by day and make an effective strategy to perform better.
* This will also help the student to gain confidence in the computer-based examination which is the format of IOE Entrance.

### 2.3.2 Applications:

This software can be used by any institutes that provide classes on any type of examinations with MCQs format such as:

* IOE Entrance Preparation.
* Medical Entrance Examination
* Lok-Sewa Examination
* Schools/Colleges/Universities.

## Drawbacks Of Existing System

Most of the institutions in today’s date, use paperwork for data management, student examination, and many more. There are many drawbacks of this primitive system for these works. Some of them are:

* It takes a lot of time to navigate to the student details on searching through several files and is tedious work to do.
* The examination process by paper examination method is extremely slow as it takes a lot of time to check all the answers of a large number of students and it requires a lot of manpower to do so.
* The students cannot give the examinations whenever they want.
* This paper system also adversely affects nature because the use of such a large amount of paper for all these works will result in cutting down of a large number of trees for the manufacture of paper.

# PROPOSED SYSTEM ARCHITECTURE

## 3.1 Block Diagram of System Architecture

The software consists of an **introductory block** that displays the project details and the details of the developers of the software when the software is initiated. Then it navigates the user to the **option selector block** where the user can select one upon various options listed there. From this option selector block, the software navigates the user to 8 different blocks with their own defined task which is described as follows:

### 3.1.1 “Enroll Yourself” Block

This block takes the personal details from the students and enrolls the students in the institutions by providing them a unique automatically generated identification number. The personal details entered will be recorded for future reference. It also checks if the person has already enrolled or not by comparing their unique identity details like citizenship number.

### 3.1.2 “Take Practice Exam” Block

This block takes a small test of the students where the students can choose the number of questions for the exam. Here, 1.5 minutes will be given to the students to answer one question. The score of the students will be recorded for future reference.

### 3.1.3 “Take Full Mock Test” Block

This block takes the full 140 marks exam of the student where the student will not be able to select the number of questions but can select among the question sets. The students will be given a maximum time of 2 hours for the exam after which the examination will end automatically. The score of the students will be recorded for future reference.

### 3.1.4 “View Overall Score” Block

This block shows scores of all the examinations the student has given based on the data recorded from the “Take Practice Exam” and “Take Full Mock Test” sections. The data will be displayed according to the identification number entered by the user.

### 3.1.5 “View Leaderboard” Block

This block shows the data of all the students and organizes them in descending order based on the overall percentage of the students.

### 3.1.6 “Amend Question Set” Block

This block allows the admin to add the question set or add questions to the existing question set. The admin will be recognized through a certain passcode.

### 3.1.7 “View All Enrolled Students” Block

This block allows the admin to see all the students that are enrolled in the institution. The admin will be recognized through a certain passcode.

### 3.1.8 “Exit” block

This block on selection exits the program.

## Data Flow Diagram

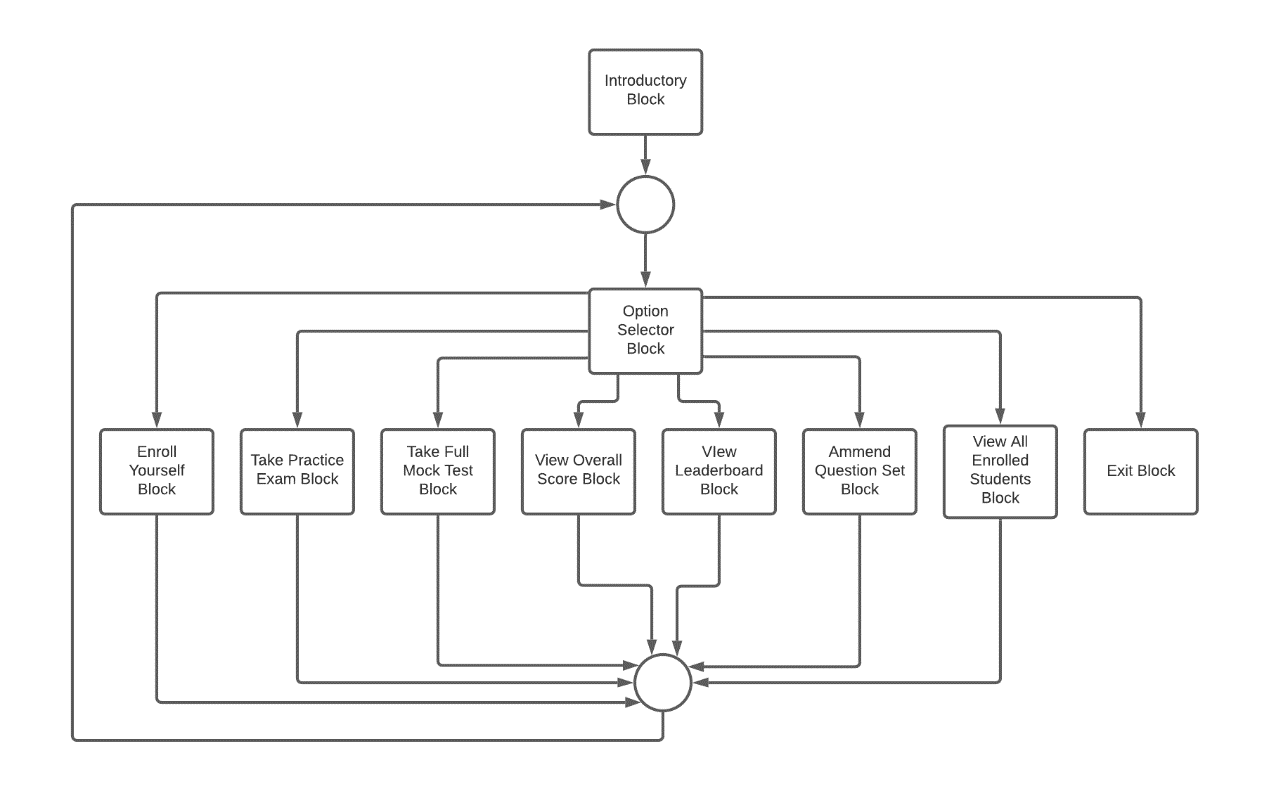


Figure 3- : Data flow Diagram Of the Project.

## Tools And Environment

To develop this software, we used IDE “Visual Studio Code” The software is compiled and run through this IDE.

# METHODOLOGY

## 4.1 Introductory Block

This block prints the name of the software, student details of the developers of the software i.e. team members. The student details include the Name and Roll Number of the team member. It also displays the faculty, college name, semester, and Enrolled year of the project team member. At last, it asks the user to press “Enter” to continue and directs to the option selector block as the user presses Enter.

### 4.1.1 Algorithm Of Introductory Block

1. Start
2. Print the software names and student details.
3. Take character input from the user.
4. If the user inputs Enter, call upon the Option\_Selector() function to navigate to Option Selector Block. Else ask the user to press Enter.
5. Stop

### 4.1.2 Flowchart Of Introductory Block

Start

Print software name and student details

Char input

Input == \n

Option\_Selector()

Stop

True

False

Figure 4- : Flow-Chart of Introductoy Block.

## Option Selector Block

This Block consists of a Option\_Selector() function that lists options from 1 to 8 that links to the 8 blocks of the program described in section 3. First of all the list of the options is displayed and then the user is asked to select among the options using the labeled number of the option. Then an integer input is taken from the user, according to which the program navigates to that part calling the respective function to display its content.

### 4.2.1 Algorithm Of Option Selector Block

1. Start
2. Print the options with their labeling number.
3. Take integer input from the user.
4. Call the respective function using the switch case statement.
5. Stop

### 4.2.2Flowchart Of Option Selector Block

Start

Print options with label number

Integer input a

a == 1

Enroll\_Yourself()

Stop

True

False

a == 2

Take\_Practice\_Set()

a == 3

Take\_Full\_Mock\_Test()

a == 4

View\_Overall\_Score()

a == 5

View\_Leaderboard()

a == 6

Amend\_Question\_Set()

a == 7

View\_Enrolled\_Students()

a == 8

Exit()

False

False

False

False

False

False

False

True

True

True

True

True

True

True

Figure 4- : Flow-Chart of Option Selector Block

## Enroll Yourself Block

The sole function of this block is to ask the students who want to enroll in the institute to ask for their details, record them and provide a unique identification number to the student specifying that he/she has enrolled in the institute. The details of the students include their name, gender, date of birth, and Citizenship Number. There are functions to calculate the age from the date of birth and store it with other details in a file. There is also a function that checks if the citizenship number entered by the user is pre-registered or not and asks the user to input another citizenship number in case of registration. There is also a separate function to generate the identification number for the students.

### Algorithm Of Enroll Yourself Block

1. Start
2. Input name(string), Gender(char), DOB(string) and Citizenship\_number(string)
3. Calculate the age using Age\_Finder() function
4. Check the Citizenship\_Number using Citizenship\_Checker() function.
5. Generate the identification number for the student using Identification\_Number\_generator() function.
6. Save all the details to a file named Student\_details.txt.
7. Save Citizenship\_Number to a file named Citizenship\_number.txt.
8. Display the identification number to the user at the console screen.
9. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
10. Stop.

### Flowchart Of Enroll Yourself Block

Start

Input Name, Gender, DOB,

Check

Identification\_Number = Identification\_Number\_Generator()

Stop

Int age = Age\_Finder(DOB)

Bool check = Citizenship\_Chekcker(Citizenship\_Number)

Input Citizenship\_Number

Print all details to Student\_details.txt

Print Citizenship\_Number details to Citizenship\_number.txt

False

True

Display the identification Number

Option\_or\_exit()

Figure 4- : Flow-Chat of Enroll Yourself Block

### Flowchart of Option\_or\_exit() function.

Stop

True

False

Input char

Input == ‘x’ or ‘X’ or \n

Input == \n

Option\_selector()

Exit()

False

True

Start

Figure 4- 4: Flow-Chart of Option\_or\_exit() function.

## Take Practice Exam Block

This block provides a short test for the user. The user can select the question set and the number of questions from a minimum of 5 to a maximum of 20 questions. Each question will be given 1.5 minutes to answer. For the calculation of the time, we need to use time.h library to access the time and functions to manipulate time. We need to use the conio.h header file to use the kbhit() function for this program.

### Algorithm Of Take Practice Exam Block

1. Start
2. Take the input for the number of questions from the user and the question set among the question sets displayed.
3. Record the time when the test was initiated using time() function.
4. Make a loop that runs for the number of times equal to the number of questions entered by the user to get questions and answers from a different file.
5. Take question input from the selected file.
6. Display the Question to the user
7. Make an integer variable (say end\_time) and initialize it with a value zero.
8. Make a while loop that runs until there is any keyboard input from the keyboard using kbhit() function.
9. In the loop, on the delay of 1 second increase the value of end\_time by 1.
10. If the value of end\_time becomes equal to 90, display “times up” and display another question to the user.
11. When the user inputs the answer, check it with the answer and provide the current score of the user and the loop continues.
12. At last when all questions get finished, then display the result and store the result with the time in the separate files as needed for future reference.
13. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
14. Stop.

### Flowchart Of Take Practice Exam Block

Start

Input number\_of\_questions, Question\_set\_no

Kbhit() = 0

Stop

time = time(), int a = 0

Input Question from Question file

False

True

Display Question, end\_time = 0

end\_time == 90

Delay 1s, end\_time++;

Input answer

Answer == file answer

Score ++, Display score

a<number\_of\_questions

Print total score

Save score and time to file

a++

False

True

False

True

True

False

Option\_or\_exit()

Figure 4- 5: Flow-Chart of Take Practice Exam Block.

## Take Full Mock Test Block

This block takes the full mock test of 100 questions where the time of 2 hours will be provided to the student to give the exam. The score of the student goes on increasing if the answer is correct and is displayed after each question. After all the questions are answered, the total score of the student is displayed and it is recorded with time in a file for future reference. The student giving the examination is identified with the help of their identification number which he/she has to enter before starting the exam. After 2 hours, the exam automatically ends. Here also, we need the time.h and conio.h library to use their functions.

### Algorithm Of Take Full Mock Test Block

1. Start
2. Take input to select the question set.
3. Record the time when the test was initiated using time() function.
4. Make a loop to run 100 times.
5. Make two integer variables (say end\_time and timer) and initialize them with a value zero.
6. If the value of end\_time becomes equal to 7200, display “times up” and end the exam.
7. Else, take questions from the file and display them to the user.
8. Make a while loop that runs until there is any keyboard input from the keyboard using kbhit() function.
9. In the loop, on the delay of 1 second increase the value of timer by 1.
10. After the user inputs the answer and the loop ends, sum the value of timer to the end\_time.
11. Check answer and increase the score value by one
12. Display another question
13. After all, the questions get finished or the time gets up, print the score and record it with examination time in separate files as needed for future reference.
14. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
15. Stop.

### Flowchart Of Take Full Mock Test Block

Start

Input Question\_set\_no

Kbhit() = 0

Stop

time = time(), int a = 0, end\_time = 0, timer = 0

Input Question from Question file

False

True

Display Question, timer = 0

Delay 1s, timer++;

Input answer

Answer == file answer

Score ++, Display score

a<100 & end\_time<7200

Print total score

Save score and time to files

a++

False

True

False

True

True

False

end\_time += timer;

Option\_or\_exit()

Figure 4- 6: Flow-Chart of Take Full Mock Test Block

## View Overall Score Block

The sole function of this block is to display the overall score of the exams given by the user. This block identifies the person with the help of the identification number and prints the overall score of the person in the practice exam and the mock test, taking data from the file where it was stored previously.

### Algorithm Of View Overall Score Block.

1. Start
2. Take identification number input from the user.
3. If the identification number doesn’t match the record, ask the user to input a valid identification number.
4. Open the file that matches with the identification number and print the scores in the file to the console screen.
5. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
6. Stop.

### Flowchart Of View Overall Score Block.

Stop

Take score from the respective files

Start

Input identification\_number

Identification\_number == match with any one from file

Display scores

Option\_or\_exit()

Figure 4- 7: Flow-Chart of View Overall Score Block.

## View Leaderboard Block

This block takes the input of the student percentage in each exam from the file where it is stored in the above exam sections. Rearrange it based on the marks to create a leaderboard. The leader board is based on the average overall performance of the students in all examinations.

### Algorithm Of View Leaderboard Block

1. Start
2. Take input of average percentage of students from the file.
3. Arrange the students' names based on the percentage.
4. Print the arranged data to the console screen.
5. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
6. Stop.

### Flowchart Of View Leaderboard Block

Stop

Take average percentage data from file

Start

Arrange data according to percentage.

Display the arranged data

Option\_or\_exit()

Figure 4- 8: Flow-Chart of View Leaderboard Block.

## Amend Question Set Block

This block allows the admin to make a new question set and input the questions and answers and also to add questions to the existing question set. To access this feature, the passcode is required which is already defined in the program and is provided to the administration of the institution only. When the admin enters this block, first of all, it asks passcode to access the block. If the passcode is wrong, it asks to enter the correct passcode, and if the passcode matches it shows two options, either to make a new question set or to amend the existing question set. Based on the selection made by the user, the functions are called and the respective task is accomplished.

### Algorithm Of Amend Question Set Block

1. Start
2. Input passcode from the user.
3. If the passcode doesn’t match, ask the user to input the correct passcode.
4. If the passcode matches, then show the option to amend the question set or to make a new question set.
5. Take input from the user to choose the options.
6. Use switch case statement to call the respective function to perform the respective task.
7. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
8. Stop.

### Flowchart Of Amend Question Set Block

Stop

Input Passcodde

Passcode = match

Start

Add question set()

Display options: 1. Ammend question 2. Add question set

Input ==1

Ammend\_set()

Input to select option

Input ==2

Add Question to a new file

Append Questions to selected file

True

False

False

True

True

False

Option\_or\_exit()

Figure 4- 9: Flow-Chart of Ammend Question Set Function.

## View All Enrolled Students Block

This block lists all the details of the enrolled students to the administration. To access this block also, we need the admin passcode. In addition to this, this block also contains a function to reset the software which clears all the data stored in all the files and makes the software ready to be used for a new session. For this section, we need to use dirent.h header file to delete the files from their respective locations.

### Algorithm Of View All Enrolled Students Block.

1. Start
2. Input passcode from the user.
3. If the passcode doesn’t match, ask the user to input the correct passcode.
4. If the passcode matches, then show the option to view student details or to reset the software.
5. Take input from the user to choose the options.
6. Use switch case statement to call the respective function to perform the respective task.
7. Call Option\_or\_exit() function that asks to press ‘X’ to exit and “Enter” to go back to the option selector block and call the respective functions as per the input by the user.
8. Stop.

### Flowchart Of View All Enrolled Students Block.

Stop

Input Passcodde

Passcode = match

Start

Reset function()

Display options: 1. View students 2. Reset software

Input ==1

View students()

Input to select option

Input ==2

Take input form the file and display the student data

True

False

False

True

True

False

Delete files and data

Option\_or\_exit()

Figure 4- 10: Flow-Chart of View All Enrolled Students Block.

## Exit Block

The only function of this block is to exit the program upon its function call. It consists of a function exit which is included in stdlib.h library.

### Algorithm Of Exit Block

1. Start
2. Call exit(1) function.
3. Stop.

### Flowchart Of Exit Block

Start

exit(1)

Stop

# SCOPE AND APPLICATIONS

The main scope of this program is in the field of institutes that run IOE entrance classes. All the functions are developed to assist them in the proper management of the preparation classes, data, and examination. But with some minor modifications in the source code, the project can be used by different institutes that run classes for entrance examinations having MCQs type pattern.

Some of the fields where this software can be used are:

* IOE Entrance Preparation.
* Medical Entrance Examination
* Lok-Sewa Examination
* Schools/Colleges/Universities and many more.

# TIME ESTIMATION

Table 1: Gantt Chart For Time Estimation.

# FEASIBILITY ANALYSIS

## 7.1 Practical Uses

This software can be used by any of the institutes providing classes regarding IOE Entrance Preparation. They will be able to make a record of all the students that enroll in the institute in a very organized way such that they can access the data anytime in the future. They can take examinations and produce results instantly. They can publish the leaderboard to show which of the student is performing well and who is lagging in performance. They can amend the question sets at any time they want using the admin passcode. They can instantly access all the student data at once using their admin passcode. They can also reset the software and use it for the other academic sessions.

Not only this, but a student can also use this software at home to take the exam and check their status regarding preparation. They can access all the sets of question sets that are added by their institution and take short practice tests or a full mock test exam. In this context, the student will have access to all the question sources provided by the institute without needing any question banks and can give exams any time they want which will boost up their confidence to give the actual computer-based exam of IOE entrance.

To conclude, the software is very feasible in the context of efficient IOE entrance preparation, both for the institute and the students.

## Cost

Use of this software by any institute can instantly **reduce the human effort** and manpower regarding data management and examination which is added benefit for the institute. The student performance can also be drastically improved so a maximum number of students can crack the entrance examination which will add **goodwill** to the institute alluring more students to the institute in coming years. The software is also very easy to use and can be learned very easily.

Considering the mentioned benefits for the institute and the features of the software, we have fixed the cost of software to be NRs 20,000 only. This cost is very nominal concerning the features of the software.

## Complexity Of The Software

The software is very user-friendly in design. There are instructions in each part of the software that allows the user to easily operate the software. A person who knows the English language can easily use this software without any training. The options in the software are also self-descriptive making the software operation as simple as eating a pie.