Final project

by Sri Krishna Kumaran

CHAPTER 1

INTRODUCTION

1.1 Overview of the Project

The university management system is developed to abolish the physical work of data entry which is time-consuming and it is also elevated to countermand the problem that exists in going through the existing manual system. Moreover, this system is intended for the need to improve the quality and quantity of the education provided by the university or other institutions.
This application is designed in such a way that it is user-friendly and at the same time it tends to lessen or avoid the error to the maximum while using the data. It is designed to help the institution in various kinds of tactical planning and at the same time different kinds of assessment and analytics are performed in order to evaluate the performance of the teacher, as well as student in their academics. It is used for storing the information regarding the student records, teacher's records etc. Starting from registration of new student it directs the details of the presence and rank calculation of marks of each student. The project takes charge of the analytics and performance assessment of the student as well as the teacher and the institution as a whole.

1.2 Aim of the Project

The main objective of this project is to cultivate and magnify the application with various types of analytics, assessment which will make this management system even more serviceable and utilitarian. The University Management System encompasses the self-assessment, which is pertinent to the teacher and the student, the subject wise performance of the student, feedback/query handling, the class performance analytics, the open student growth analytics, the business analytics, the digital notice board, advice and discussion forum, Identity Access management panel etc. The necessity of the student is:

- Login to the application using the register number and password.

- They can view their self-profile in the navigator menu.
- They can define their career as shown by the career helper option in their dashboard.
- They can give feedback and enter complaints and other queries in the query handling option.
- They can view the newsletter of their campus using the digital notice board.
- They can view their subject wise performances of them in each subject, in the exams conducted throughout the semester.
- They can assess their performance in each subject by doing rank calculation of the marks scored by the student in respective exams
- They can also view their attendance.

The prerequisite of the teacher is:

- Login to the application using their id and password.
- They can view the list of all the students who have done or yet to do the payment of fees like tuition fees, exam fees etc in the payment management option.
- They can view the subject-wise performance of each student in each subject.
- They can view their self-assessment report and their teaching performance analytics of them, in the subject they teach.
- They can add new events in the digital notice board in case of occurrence of a new event.
- They can do the class performance analytics to improve the quality and quantity of the education.
- They can do the open student growth analytics to help the students in further improving their performance.
- They can solve student queries and can also give feedback and other queries using the feedback handling option.

The requisite of the admin is:

- Login to the application using their id and password.
- They can access all the details of their employees/teachers and students
- They can do registration of the new employees/teachers and students
- They can use the analytics data for various purpose of their business.

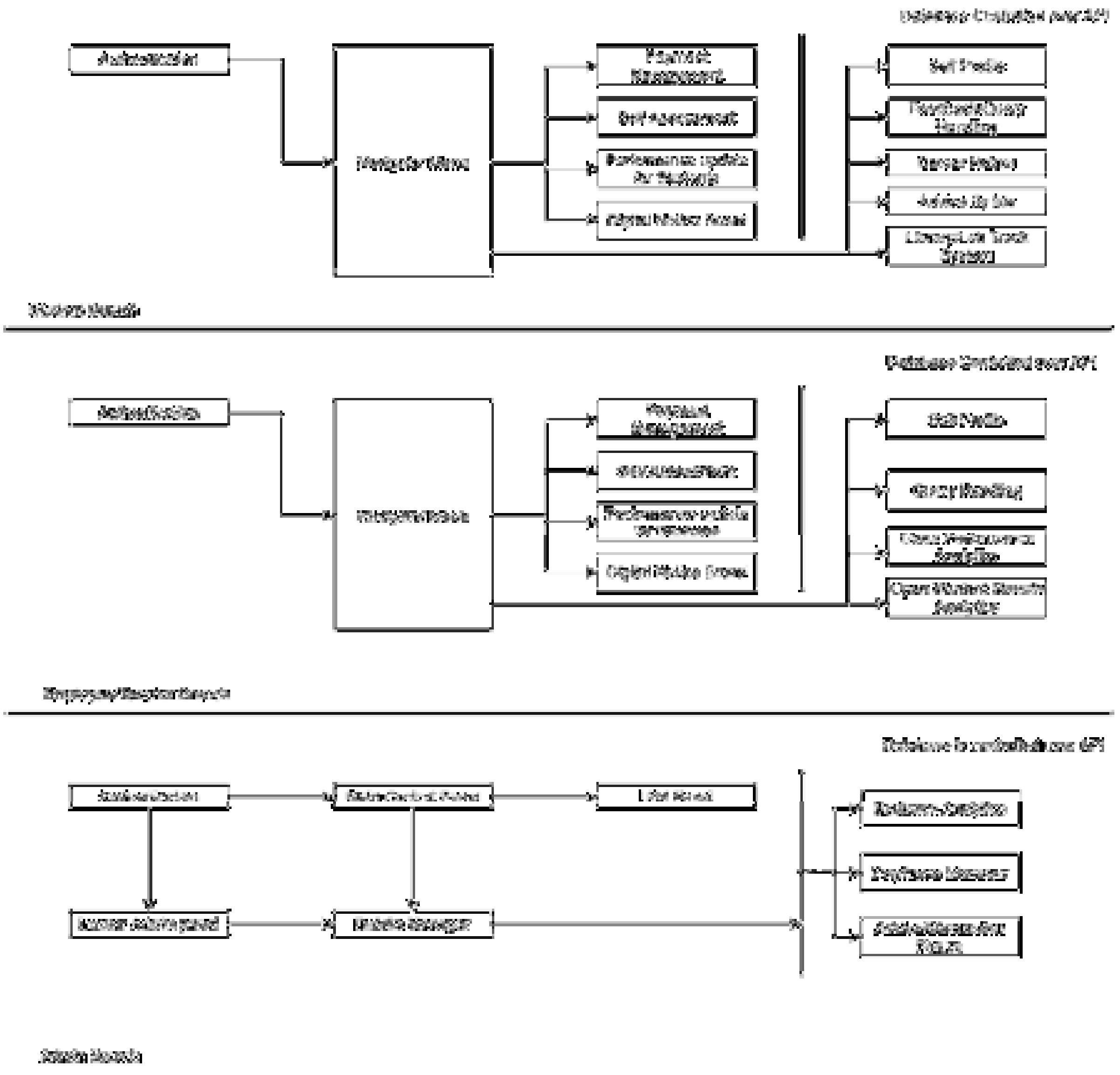


Fig1.1 :- The Architecture diagram

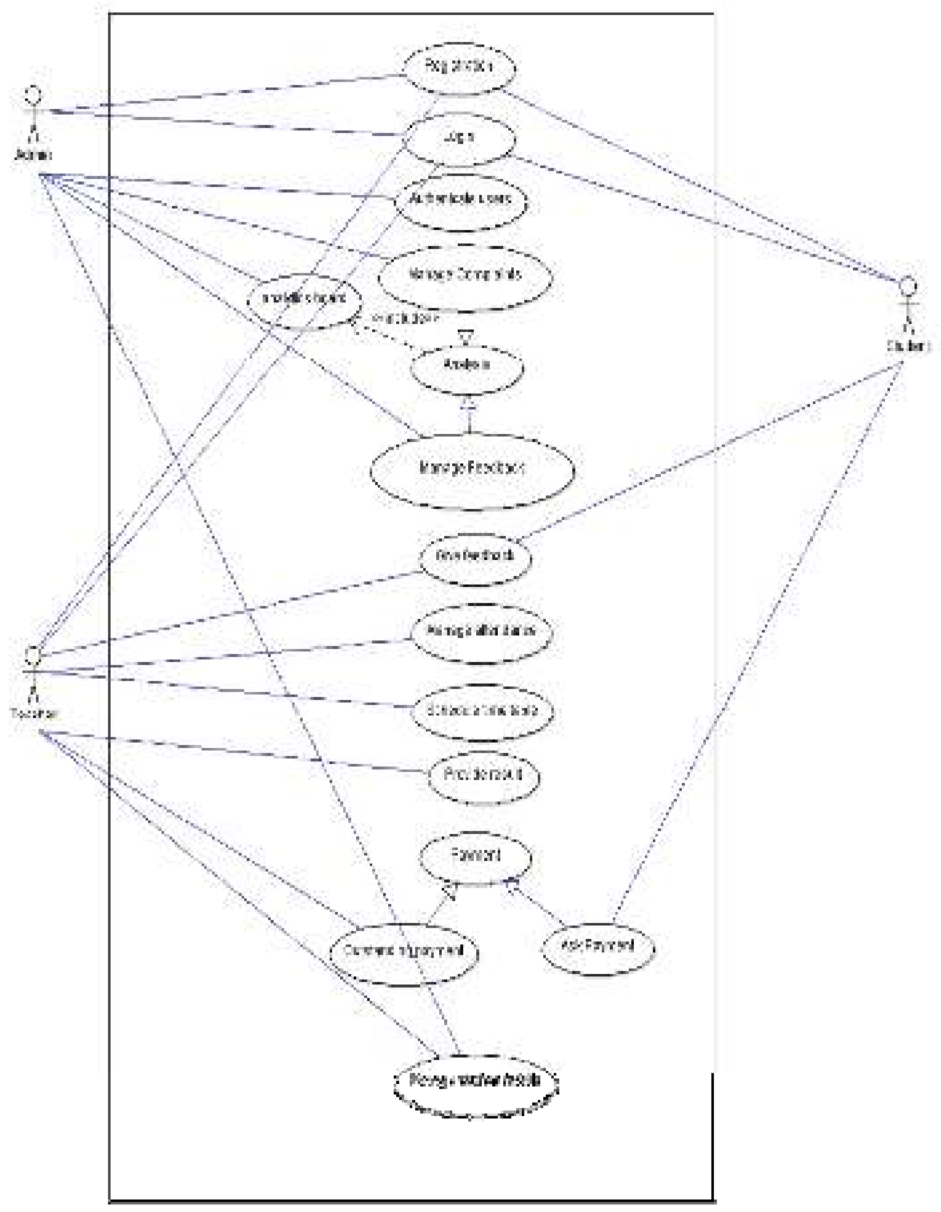


Fig 1.2 :- The Use Case Diagram

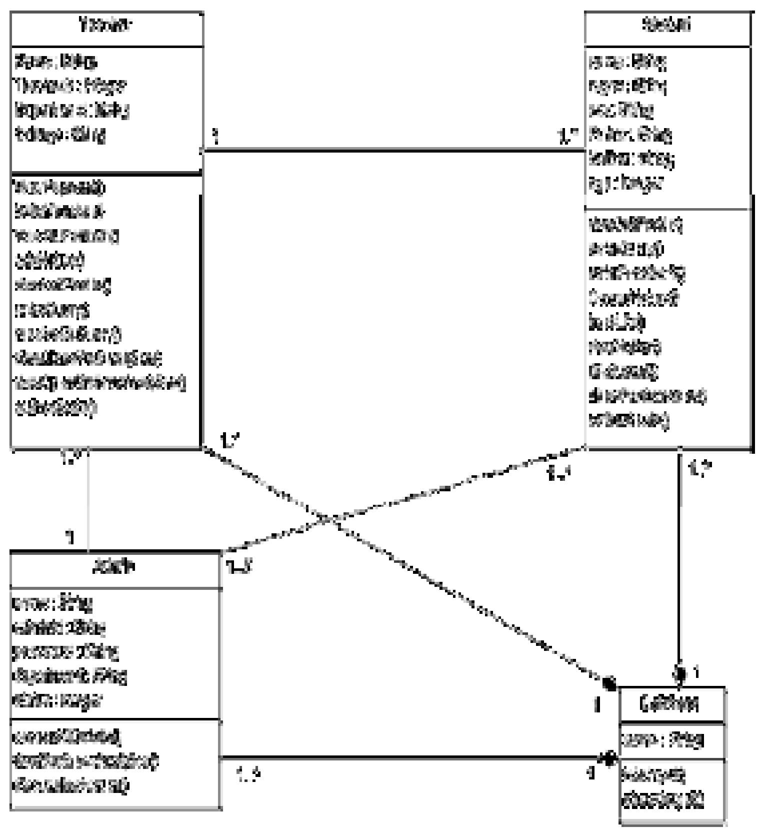


Fig 1.3 :- The Class Diagram

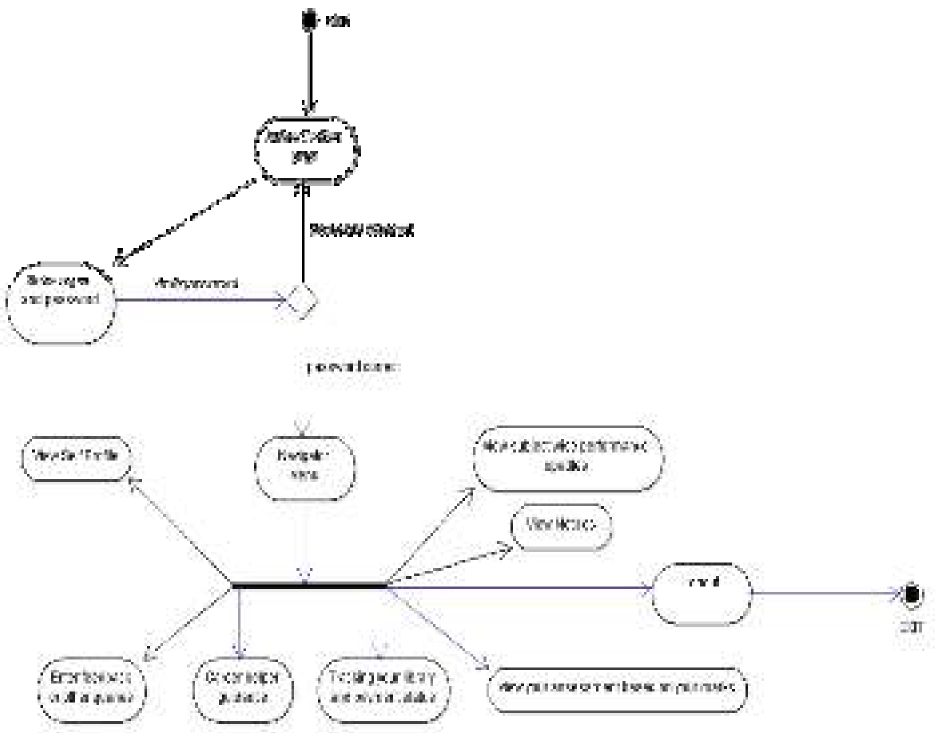


Fig 1.4.1 :- The Activity Diagram(Student)

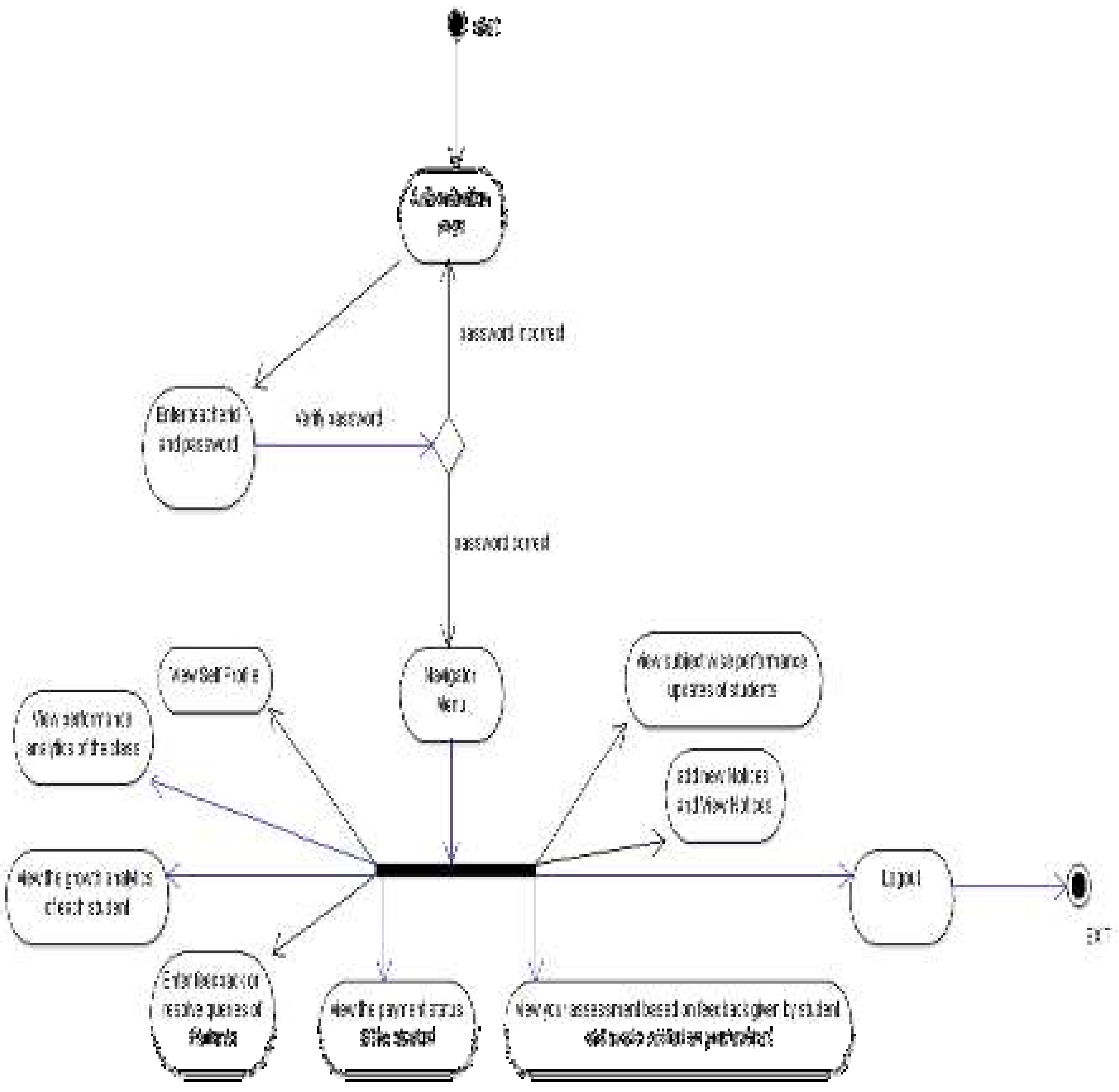

start
Enter teacher id and password
Verify password
password incorrect
password correct
Navigation Menu
View Self Profile
View performance analytics of the class
View the growth analytics of each student
Enter feedback or resolve queries of students
View the payment status
View your assessment based on feedback given by student
View subject wise performance updates of students
add new Notices and View Notices
Logout
EXIT

Fig 1.4.2 :- The Activity Diagram(Teacher/Employee)

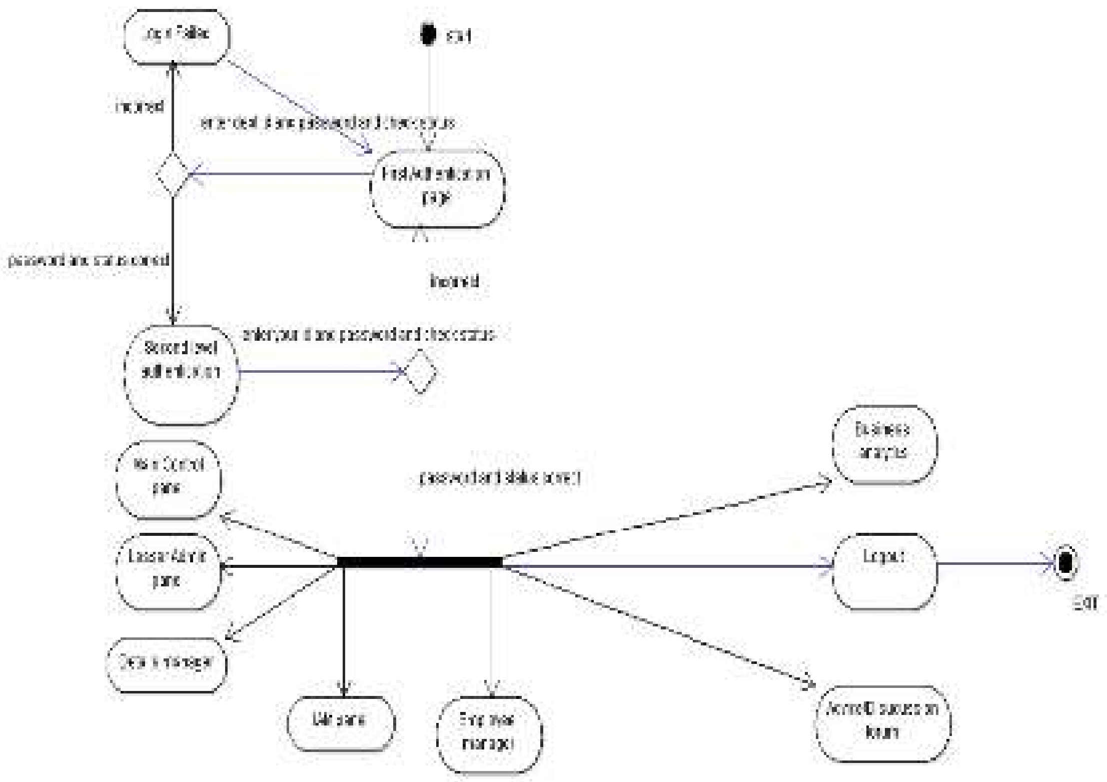

Login Failed
start
incorrect
enter userid and password and check status
First Authentication page
password and status correct
incorrect
Second level authentication
enter your id and password and check status
Business analysis
Main Control panel
password and status correct
Lesser Admin panel
Logout
Exit
Deal manager
Mail panel
Employee manager
Account discussion forum

Fig 1.4.3 :- The Activity Diagram (Admin)

The Fig 1.1 tells you about the use case diagram of the university management system. Here there are three actors in this usecase diagram. The admin and the teacher are the primary actors because they are located in the left side of the use case diagram. Primary actors are the actors who use the system to achieve the goal.The use case archives the interactions between the system and actors to achieve the goal of primary actor. Secondary actors are actors that the system needs assistance from to attain the primary actor's goal. Primary actor is a user goal and is fulfilled by the system. It is the stakeholder that calls on the system to deliver one of its services. It has a goal with respect to the system – one that can be satisfied by its operation. It is not often,but not always, the actor who triggers the use case. The secondary actor in a use case in an external actor that provides a service to the system under design. It might be a high-speed printer, a web service, or humans that have to do some research and get back to us. Both the primary as well as secondary actor are external objects of the system that produce or consume the data. They serve as sources and destinations for data. In this particular diagram, the system is shown by large rectangular boundary. Inside the boundary the various usecases are present and outside the boundary present are the actors as mentioned above. Each actor has respective relationship with the usecases present in the system. Here the each usecase represents the functions of the actors.The relationship is represented as a straight line between actor and use case. The actor "Teacher" has the following functionalities in the system like a new teacher's registration, teacher has to enter the identification number and password in order to obtain information, a teacher can give various feedbacks, a teacher can manage the attendance of students, a teacher can schedule a time table as well as they can provide the result of the students, a teacher can view the outstanding payment of the students as well as they can manage student details.

The actor "Admin" has the following functionalities in the system like a new admin employee's registration , admin has to login into the system to access the information, admin can authenticate other users, admin has to manage the feedbacks or compliants of their employees, admin can view the various types of analytics and use it for various purposes, an admin can manage the feedbacks of all the employees, an admin can also manage student details.

The actor "Student" has the following functionalities in the system like the registration of new student, the student should login into the system in order to access the information, students can do the payment of any fees pending, the student can give feedback for the teacher and other queries.

The various usecases are :-

- Registration – It is the registration of the new user into the system.
- Login – It is the usecase in which the user logs into the system using their id and password. It is applicable to all the actors.
- Authenticate users – It is the usecase in which userid and password are provided for authenticating the user.
- Manage compliants – It is the usecase which is applicable only to the admin where it manages all the compliants of their employees.It inherits the analysis usecase.
- Analysis – It is the usecase in which various types of analytics can be viewed by the admin so that it can be used for various business purposes.
- Manage Feedbacks – It is the usecase which inherits the analysis usecase.

- Give feedback – It is the usecase in which it collects the various feedbacks from the students and teachers.
- Manage Attendance – It is the usecase in which the teacher manages the attendance of the students.
- Schedule Timetable – It is the usecase in which the teacher schedules the timetable for subject hours.
- Payment – It is the usecase in which the other two usecases outstanding payment, ask payment inherit the properties.
- Manage student details – It is the usecase in which the teacher manages the student details as a whole.

The Fig 1.2 tells you the class diagram of the university management system. In this diagram there are four classes which are called as teacher, student, admin, database. In a class diagram the class is represented as rectangle with three parts in it. The first part is the top part of the rectangle in which it represents the name of the class. The second part is the middle part of the rectangle in which it contains the attributes of the class like name,id etc. The third part of the rectangle is the lowest part of the rectangle in which it contains all the neceessary functions of the class.
The first class of the class diagram is called as the “teacher”. The various attributes of the class “teacher” are name which tells you the name of the teacher, teacherid which tells you the identification part of the teacher, department tells you the department in which the teacher belongs, college tells you about the name of the college in which the teacher works.The various functions of the class “teacher” are:-

- viewPayment() – It tells you that the teacher can look into or view the payment updates of the student.
- SAforTeacher() – It tells you that teacher can do their own assessment based on the feedback given by students and other factors.
- ViewStuPerform() – It tells you that the teacher can view the subject-wise performance of each student obtained in the respective exams.
- addNotice() – It tells you that the teacher can add new notice or event in the digital notice board if required.
- viewSelfProfile() – It tells you that the student can view their complete profile details after logging into the system.

- enterQuery() – It tells you that the student can enter the query or complaint which will be resolved by the teacher.
- enterFeedback() – It tells you that the student can give feedback for the subject teacher in order to evaluate the teaching performance.
- viewClassPerfAnalytics() – It tells you that the teacher can view the performance analytics of the class as a whole.
- viewOpenStuGrowAnalytics() – It tells you that the teacher can view the growth performance analytics of each student.
- authenticate() – It tells you that the teacher has to substantiate themselves into the system using their id and password in order to access the system.

The second class of the class diagram is named as " student" . The various attributes of the class "student" are - name which tells you the name of the student, regno tells you the unique registration number of the student, sex tells you whether the student is male or female, Father tells you the name of the father of the student, Mother tells you the name of the mother of the student, age tells you the age of the student. The various functions of the class "student" are:-

- CareerHelper() – It tells the student the various career possibilities based on the chosen elective subject.
- SAstudent() – It tells you that student can do their own assessment based on doing the rank calculation of the marks obtained by the student in the exams.
- showPerformance() – It tells you that the student can view the subject-wise performance of each student obtained in the respective exams.
- viewNotice() – It tells you that the teacher can view the notices or events in the digital notice board.
- viewSelfProfile() – It tells you that the teacher can view their profile details after logging into the system.
- enterQuery() – It tells you that the teacher can enter the query or complaint which will be resolved by the admin.
- enterFeedback() – It tells you that the sudent can give the feedback for teachers in order to evaluate the teaching performance.
- trackLib() – It tells you that the student can track the status of the library book issued from the university library.
- authenticate() – It tells you that the student has to corroborate themselves to the

system in order to view their information.

The third class of the class diagram is called as “admin”. The various attributes of the class “ admin” are – name which tells you the name of the admin personnel, adminid which tells you the identification details of the admin personnel, password which tells you the password required to log into the system, department which tells you the department in which the admin personnel is working, status which is required for performing the two-level authentication. The various functions of the class “admin” are:-

- accessAllDetails() – It tells you that the admin can access details of all the employees working in the organization.
- viewBusinessAnalytics() – It tells you the various types of analytics which are required by the institution for various business purposes.
- discussionForum() – It tells you the complaints and other queries of their employees and it is used for resolving the complaints of these employees.

The fourth class of the class diagram is called as “database”. The attributes of the class “database” are – name which tells you the name of the database. The various functions of the class ”database” are:-

- takeInput() – It tells you that the database can accept various inputs in order to store them and retrieve it for future use.
- showOutput() – It tells you that the database can show the desired output as required by the user at that moment.

In the Fig 1.3 there are two types of relationship shown which represents the link between the various classes. The first relationship used in the diagram is called as a dependency which is represented by a dotted line with open arrow at the end. This relationship is shown by the class “teacher” and “admin”. Dependency is a directed relationship which shows that some set of elements depend on other model of elements for implementation. It is a relationship between named elements. The second relationship used in the diagram is called as a composite relationship which is represented by a solid line with a filled diamondat the association connected to the class of composite. It denotes a strong ownership between two classes when one class is part of another class.In the class diagram the three classes “admin”,”teacher”,”student” have composite relationship with class “database”. In the class diagram multiplicity is also shown for all the relationships. Multiplicity is a factor associated with an attribute.It specifies how many instances of attributes are created when a class is initialized. If a mulitplicity is not specified, then by default one is considered as a default multiplicity.

In the Fig 1.4.1 it tells you the activity diagram of one of the user i.e student. In an activity diagram an initial node is represented by a shaded circle which portrays the beginning of the set of actions. The final node is represented as an incircle in which the inner circle is shaded, it stops all control flows and object flows in an action. The action is represented in the form of curved rectangle which is curved on the corners of rectangle, it tells you the task that is to be performed. The control flow is represented as a straight line with an open arrow, it shows the sequence of the execution of the actions. The decision node is represented as an unshaded kite, which represents a test condition to ensure that the control flow only goes down the path. The fork node is represented as a thick horizontal line, it splits the behaviour into a set of parallel or concurrent flows of activities. In the figure 1.4.1 when the activity starts it shows you the authentication page where the user has to type his user identification and password in order to authenticate himself/herself as a member of the university. While authentication the system verifies the password entered with its actual value, if the password is correct then the process proceeds further in order to show the navigator menu. If the password is incorrect it will redirect itself to the authentication page. In the navigator menu it has various options or services that the user can access. It consists of various actions like:-

- ViewSelfProfile – It displays the important profile details of the user
- Enter feedback or other queries – It tells you that the user can give feedback of the particular subject teacher, and can also enter other complaints.
- Career Helper Guidance – It suggests the possibility of various career paths such that the user can pursue anyone of them based on the elective subjects chosen by the user.
- Tracking Library and other payment status – It tells you that the student can not only track his/her library activities but also their other outstanding payments.
- View your assessment based on your marks – It helps the student for doing their own self-assessment by doing rank calculation based marks obtained by the student in each subject, and in the respective cycle tests.
- View Notices – It helps the students to know what are all the events occuring in the institution.
- View subject-wise performance updates – It displays the subject-wise

performance analytics of the student in the respective exams like cycle tests etc.

- Logout – It tells that the student can come out of the application any time.

These actions executed sequentially are represented using the control flow notation of the activity diagram.

In the Fig 1.4.2 it tells you the activity diagram of another user i.e teacher. In an activity diagram an initial node is represented by a shaded circle which portrays the beginning of the set of actions. The final node is represented as an incircle in which the inner circle is shaded, it stops all control flows and object flows in an action. The action is represented in the form of curved rectangle which is curved on the corners of rectangle, it tells you the task that is to be performed. The control flow is represented as a straight line with an open arrow, it shows the sequence of the execution of the actions. The decision node is represented as an unshaded kite, which represents a test condition to ensure that the control flow only goes down the path. The fork node is represented as a thick horizontal line, it splits the behaviour into a set of parallel or concurrent flows of activities. In the figure 1.4.2 when the activity starts it follows the authentication process just like in the figure1.4.1 i.e, it shows you the authentication page where the user has to type his user identification and password in order to authenticate himself/herself as a member of the university. While authentication the system verifies the password entered with its actual value, if the password is correct then the process proceeds further in order to show the navigator menu. If the password is incorrect it will redirect itself to the authentication page. In the navigator menu it has various options or services that the user can access. It consists of various actions like:-

- ViewSelfProfile – It displays the important profile details of the user
- Enter feedback or resolve other queries of students – It lets the user to give feedbacks or other complaints, as well as they can resolve the queries of the students
- View payment status of the student – It lets the teacher know the details of all the students and their payment status regarding their various outstanding payments.
- View performance analytics of the class – It helps the teacher to know the performance analytics of the class in each subject in the respective exams.
- View the growth analytics of each student – It helps the teacher in understanding the growth performance analytics of each student in the academic year.
- View your assessment based on the feedback given by students and marks obtained – It helps the teacher in doing their self assessment by collecting

feedbacks based on various parameters and viewing analytics based on the marks scored by students in the subject taught by the teacher.

- Add new Notices and View Notices – It helps the teacher to know what are all the events occuring in the institution, as well as the teacher can add new events to the digital notice board so that all the users are able to access the information.
- View subject-wise performance updates of students – It displays the subject-wise performance analytics of the each student in each of their respective exams like cycle tests etc.
- Logout – It tells that the teacher/employee can come out of the application any time.

These actions executed sequentially are represented using the control flow notation of the activity diagram.

In the Fig 1.4.3 it tells you the activity diagram of the other user i.e admin. In an activity diagram an initial node is represented by a shaded circle which portrays the beginning of the set of actions. The final node is represented as an incircle in which the inner circle is shaded, it stops all control flows and object flows in an action. The action is represented in the form of curved rectangle which is curved on the corners of rectangle, it tells you the task that is to be performed. The control flow is represented as a straight line with an open arrow, it shows the sequence of the execution of the actions. The decision node is represented as an unshaded kite, which represents a test condition to ensure that the control flow only goes down the path. The fork node is represented as a thick horizontal line, it splits the behaviour into a set of parallel or concurrent flows of activities. In figure 1.4.3 when the activity starts the authentication process of the admin is different from that of authentication process of either teacher or student i.e it shows you the first authentication page in which the user had to enter the department id and its password and after that it verifies the password. If the password is correct then it proceeds to the next step for the second level authentication page otherwise it redirects itself to the first authentication page. When the user arrives at the second level authentication page then the user has to enter the unique id and password and after that it verifies the password. If this password is correct it directly takes you to the respective department page otherwise it redirects you to the first authentication page. Here there is no concept of navigator menu as it is in the other activity diagrams.

It consists of various actions like:-

- Main Control Panel– It is the department which consists of top members of the institution like Founder, CEO, Chairman etc.

- Lesser Admin Panel – It is the department which consists of the following like:- Controller of Examinations, Public Relations, Placement Cell etc.
- Details Manager– It is the department only applicable for lesser admin panel which is basically used for making changes in the information.
- Employee Manager – It is the department applicable for lesser admin panel and is basically used for managing the details of students and teachers.
- IAM Panel – It has the top authority because it can access the details of all the employees/teachers, students, admin personnel as well as manage them.It is applicable for main control panel.
- View Business Analytics – It lets the admin personnel view various kinds of analytics for using it for various business purposes like marketing your institution etc.
- Advice/Discussion Forum – It is used by the admin personnel to resolve the complaints and other queries of their employees.
- Logout – It tells that the admin personnel can come out of the application any time.

These actions executed sequentially are represented using the control flow notation of the activity diagram.

CHAPTER 2

LITERATURE SURVEY

2.1 College Department Management System

The project provides the counselling to the administration personnel to keep spoor of the student. The management authority has acquired the access to the database of the system. In a scholastic foundation governance, it is climacteric thing. This system contains various functionalities like plebiscite occurence details, estimations, article line etc. It dispenses the supplementary characteristic news line in order to help the student to obtain various department news lines and other outlines like achievements etc. It furnishes us with the referendum property so that physical drudgery is diminished. It is a wrapper-less system which furnishes serviceability, for the student to application where in the top charge of the organization can govern, client can acquire the uploaded inscription, curriculum information. Users will get the incident details via SMS. The main objective of the system is to streamline the effort in order to develop a frame and identify obligatory trait that a taut has every confidence in online division working system must gratify to reduce breakthrough redundancy. Such a skeleton will allow us to gauge as well as compare and contrast the excellence of subsisting and hereafter candidate in the sector. The structure supports more than one user domain and it is fully automated. The set-up is competent to keep stalk of all the circumstantial elucidation of the user and the complete attribute of services proffered by the client.

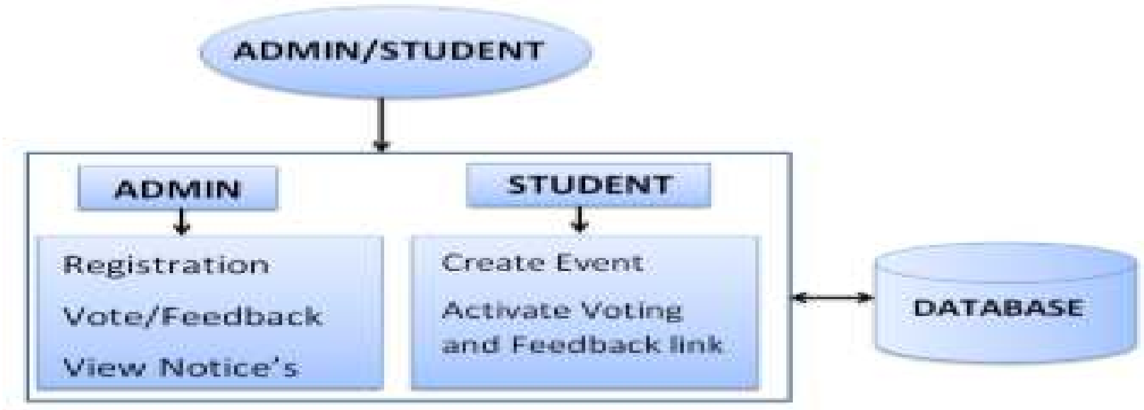


Fig 2.1 :- The College Department Management System Architecture

2.2 Automated Online College Admission Management System

The System is software advanced to work on web-rostrum to superintend the complete concession stratagem of various sections of an institution like, Capital Division Operation, tutee portion and many more segments. Over the years that the activity of revelation, notification panels, salient proclamations about pedagogical and other controls, are being transmitted out blue-collarly through wrapper and pen which is very time-engrossing and a big headache to nurture all that documentation at some place. This proceeding is not only time-gripping but also inefficacious and it is strenuous to uphold the paperwork and other data. Through this software they are trying to vanquish the complication of defending the paper-based document by laying emphasis on the digital library. In inclusion to information-based technology, we would be sending pushin apprising given by the academies and the very same egress would be again used to advise the students and even the parents about the annunciations digitally. Keeping all these points in the juncture, we have burgeoned a web instrument which is carried-out using web-services that would bridge with the database initiated on an isolated server. The Unique PRN would provide idiosyncratic recognition to all the scholars who would be utilizing this system. PRN Number would not just help the top authority to keep the trace of students, but would make it uncomplicated for the students as the student doesn't need to go through the pain of tendering diverse hard copies of the chronicles and corroborations each time the institute requires it. Automated Online College Admission Management System is a simple yet effortful tool that would result in lessening the paper-work undemanding for the organizations as well as the user who would use it.

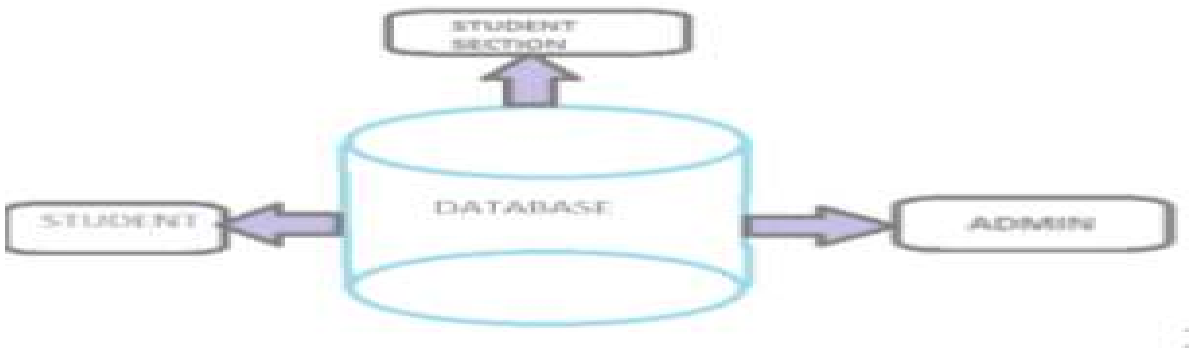


Fig 2.2.1 :- The Architecture

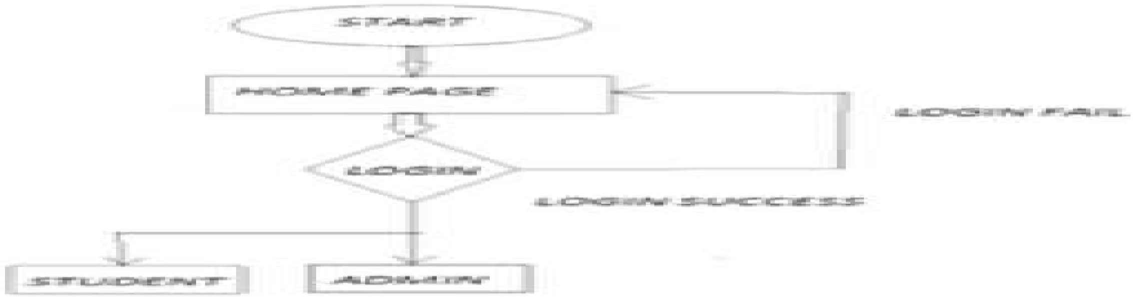


Fig 2.2.2 :- The Data Flow Graph

2.3 An Android App for University Management System

The Android University Management System is an android app which is useful for students as well as the university faculty member and other academic staff. A paper-based system all the works are done manually. Paper based work is so time consuming. The main objective behind the use of University Management System is easy supervision of the Institute. Through our application students can search their results using Android mobile. All information is stored in the university server. The faculty member is authorized to log into University Management System database backend through administrator user id and password and update the student result. In this application, students have no administrative user id or password and no scope to update their results. The student can only search their result using their student's id and password provided by the admin. Through this application student can also calculate their CGPA through CGPA calculator. Through this application notice board is all time updated by respective teachers and admin personnel. Student's attendance is also collected by this application. This application maintains a simple structural interface for student and employee records. It is often utilized by private and public universities to keep up the records of scholars simply. This application provides university a wide range of facilities like student result analysis, CGPA calculation, notice board that provide a regular update of any time of any event happens in university premises, attendance, account maintenance, result viewing reducing use of paperwork and automating the all university purposes.

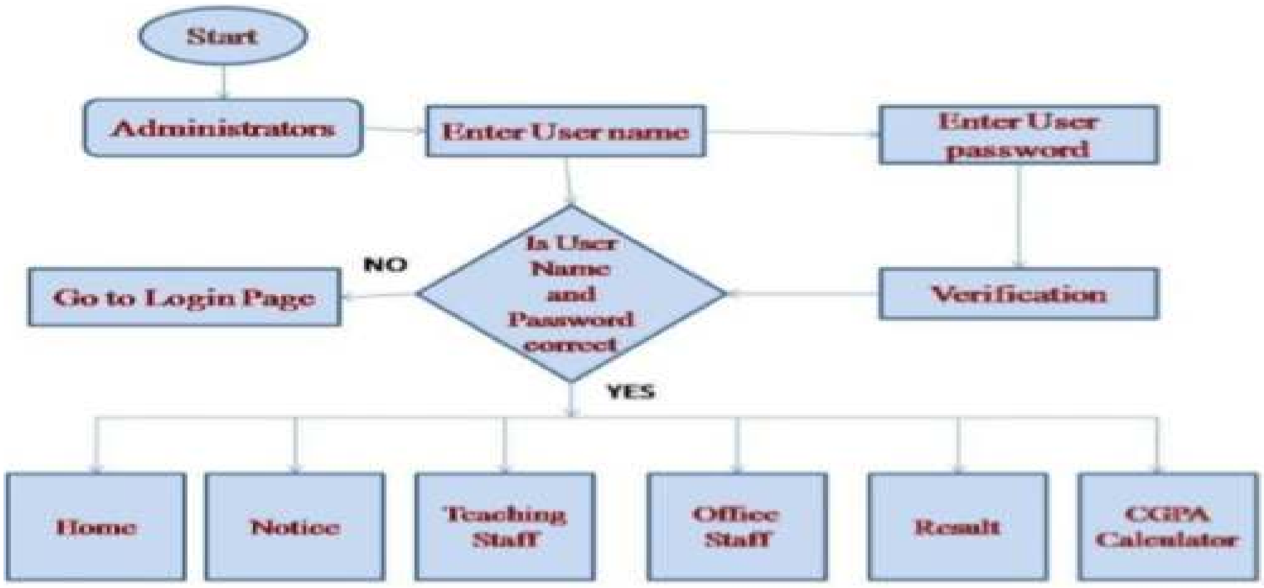


Fig 2.3.1 :- The Android App of University Management System Data Flow Graph

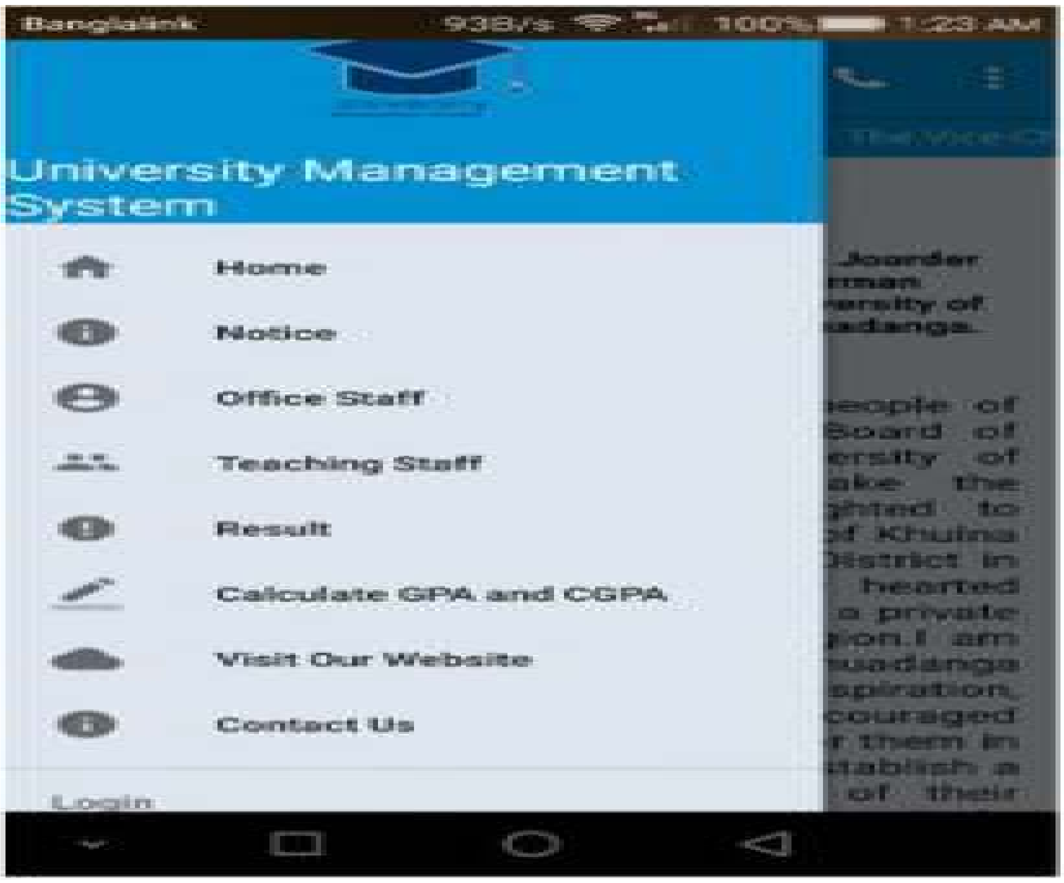


Fig 2.3.2 :- The Android App of University Management System

2.4 University Management System using MVC

University management system deals with the perpetuation of University data, documentation, other decrees, and other news within the varsity. It is a system, which is used as depository for the particulars, students counsel, and the guidance of courses. Beginning from the enlisting of a new student in the institute, it prolongs all the decrees concerning the appearance and marks of the student. The enterprise deals with the repossessing of the documentation through an INTRANET based campus extensive threshold. It accumulates coupled statistics from all the sections and subsections of an institution and advocates folders, which are used to engender transcripts in various configurations to appraise discrete and all-embracing capability of the pupils. The evolving exercise of the system begins with System anatomization. This inspection involves creating a ceremonial representation of the complication to be resolved by apprehending preconditions. The system is expanded to overthrow the issue prevailing in the enactment of the non-automatic system, this software is reinforced to annihilate and in some cases foreshorten the destitution faced by this prevailing system.

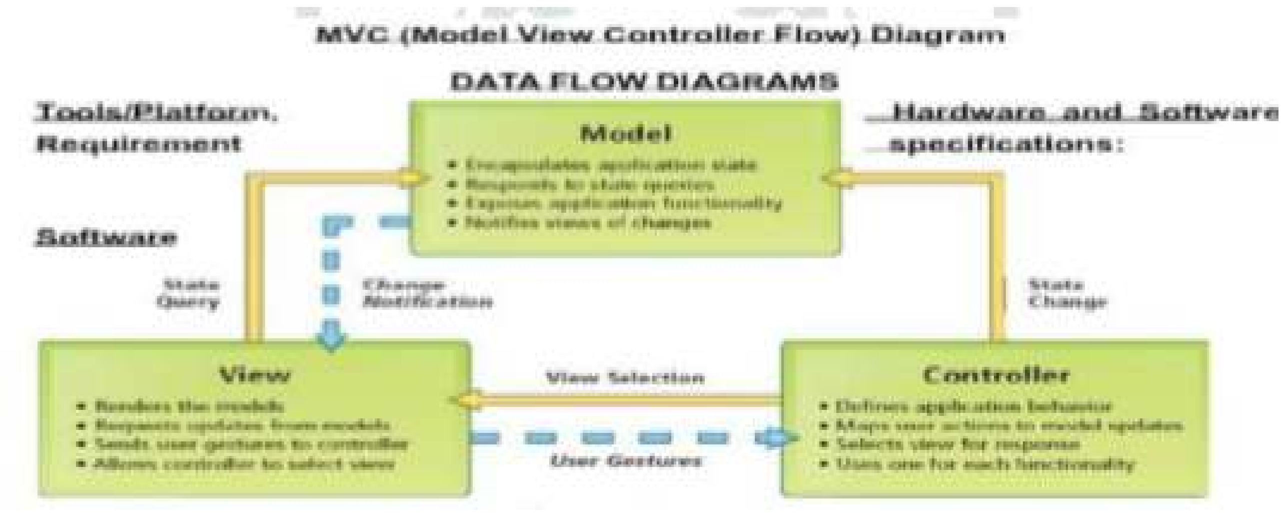

MVC (Model View Controller Flow) Diagram
DATA FLOW DIAGRAMS
Tools/Platform, Requirement
Software
Hardware and Software specifications:
Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• Uses one for each functionality

Fig 2.4.1 :- The Model View Controller diagram

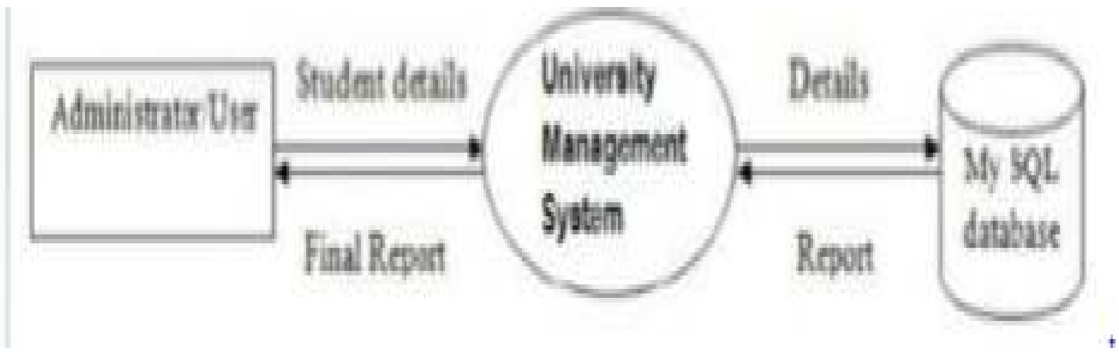


Fig 2.4.2 :- The University Management System Data Flow diagram

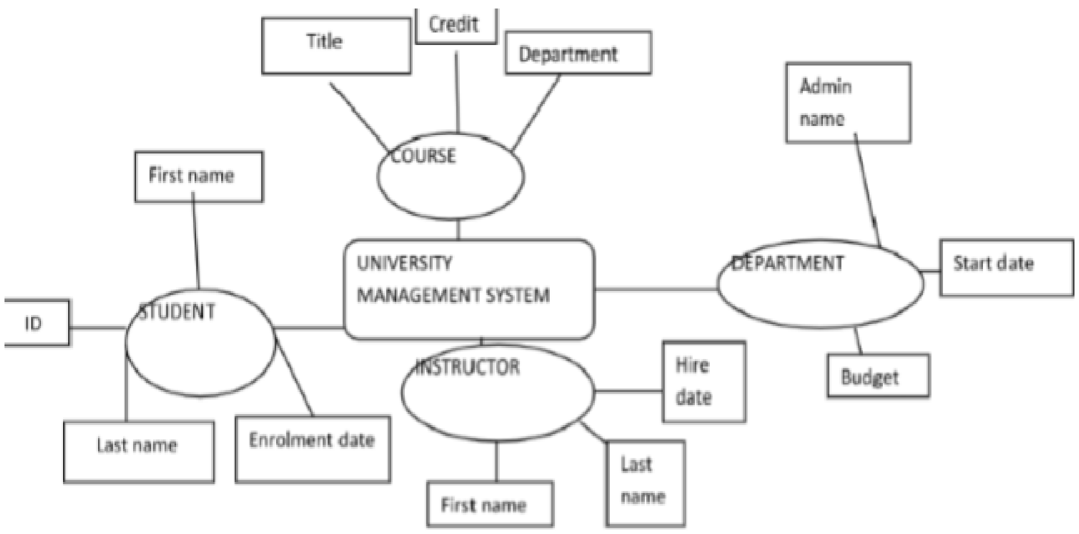


Fig 2.4.3 :- The University Management System Entity Relationship diagram

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

As mentioned, that the existing system used before were used to store only the important details like your self-profile and other personal details. As information related to the various other important postulates were not available in digitalized format, they had to be maintained manually. There was no analytics and assessment used for either the teacher or the student for improving the quality of education. In the existing system the exams are done only physically. It isn’t user friendly as a result the retrieval of information is incredibly slow and data isn’t maintained expeditiously. These systems have to be compelled to handle by specialist for maintaining and update the system which might be terribly pricey. Its need more paper work calculation to prepare the final exam result at end of an academic year. The existing system victimized the deprecated strategies as new JSON and GCM is more flexible then previous algorithmic program. All calculation to get report is completed manually therefore there’s larger probability of errors. Here the University has got to suffer plenty through the calculation and if there is a loss of some report then it’s going to cause plenty. Even then there is some fault that is incredibly frustrating for the University. These calculations conjointly affects average result of the students.

3.1.1 Drawbacks of Existing System

- Lack of security feature and no data integrity.
- Decreasing the manpower increased the various other kinds of maintenance costs.
- Customization is done on the go.
- Scalability is very less.
- There was no role of direct officials in the system.
- Analytics drawn was not in favor of either the student or teacher, but just on the favour of university.
- It devastates large compendium of retrenched work.

3.2 Proposed System

In this project, we have introduced an efficient solution for representing the system using MVC and data analytics. Here it is used to assess and process the feedback given by teacher and student. At the same time the performance of the university as a whole, can be viewed. This system represents the information of the performance of the student, teacher, departments and university in a better and efficient way than anything else. This software can be accessed by the admin, teacher and student. It improves the efficiency of university information storage system.

It will reduce the amount of time to process and view notice. Student can calculate their CGPA through CGPA calculator. Student and can easily access the result and that will be subject-wise.

3.2.1 Advantages of Proposed System

- Ensure data uprightness.
- Proper control and access available for the higher officials.
- Greater coherence and proficient.
- User amiability, affability and fraternizable.
- Analytics and other kinds of performance assessments and analytics can be shown in a complete and meaningful manner.
- Query/feedback handling is done for student, teacher, university as a whole.

3.3 System Architecture Explanation

In Fig 1.1 it tells you the system architecture diagram. In this architecture there are three different kinds of users namely – admin, teacher/employee, student. As there are three different kinds of users found in this architecture, hence there are three different software. Each software is specifically designed for specific user. All these three software is combined to form a university management system. In this architecture authentication of the user is common. So, irrespective of whoever he/she is the user has to log into the system in order to access the system. The authentication part of student, teacher are called as one-level authentication, where the user enters the userid and password in order to access the application. The authentication part of admin is called as two-level authentication, where the user first enters the department id and password in order to clear the first-level authentication, and then the user enters his unique id and password in order to clear the second-level authentication. The concept of navigator menu is only found in the teacher's part and student's part. But in the admin part after the two-level authentication is finished it will redirect the admin personnel to his/her respective department page.

3.4 Comparison of Existing and Proposed System

In the existing system it was stored used only for storing some important details like Personal Profile etc. Whereas in the proposed system it does more than just storing details i.e it represents those details in a meaningful and user-friendly manner. In the current existing system there is no analytics and other kinds of performance assessments done for student, teacher, university in a large scale. Whereas in the proposed system the analytics and other performance assessments of student, teacher, university is represented in a more meaningful and complete manner. In the existing system there is no security feature put up and there is no data integrity. Whereas in the proposed system there is data integrity and security feature put up in order to protect the data from getting misused by any third party. In the existing system there is no role for higher officials for accessing the information in the system. Whereas in the proposed system higher officials can access all the information of all their employees/teachers, students, university as a whole. In the existing system there is less scalability and it cannot be used or produced in a range of capabilities. Whereas in the proposed system the scalability is more as compared to the existing system and you can use this system remotely at any place and at any time. In the existing system the analytics is done not on the favor of student but just on university's favor. Whereas in the proposed system the analytics is done on the favor of student as well as teacher and the university.

3.5 Algorithms

- Career Clustering Algorithm
  1. It first fetches the list of electives chosen by that particular student.
  2. Then based on the electives chosen it given suggestion to the student on their various possible career paths.

3.6 Modules

3.6.1 Authentication Module

It is the module in which it is used to authenticate the user by entering their user id and password. There are two types of authentication used in this system – one-level authentication and two-level authentication. The one level authentication means that when the user enters the id and password

After that you enter into navigator menu directly. The two-level authentication means the user has to enter the user id and password two different times in order to access the system information.

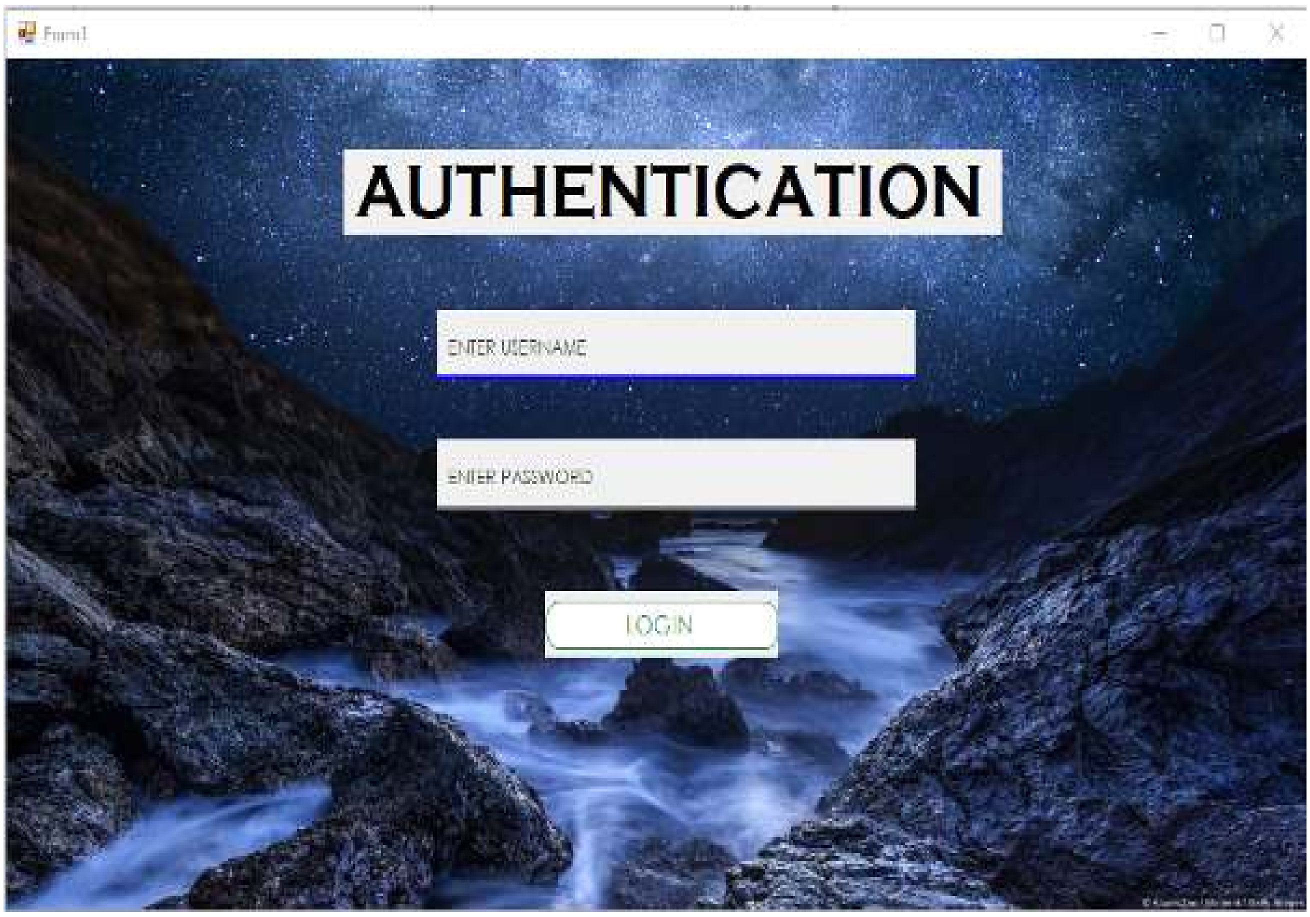


Fig 3.6.1.1 :- The Authentication Module of the user(Student)

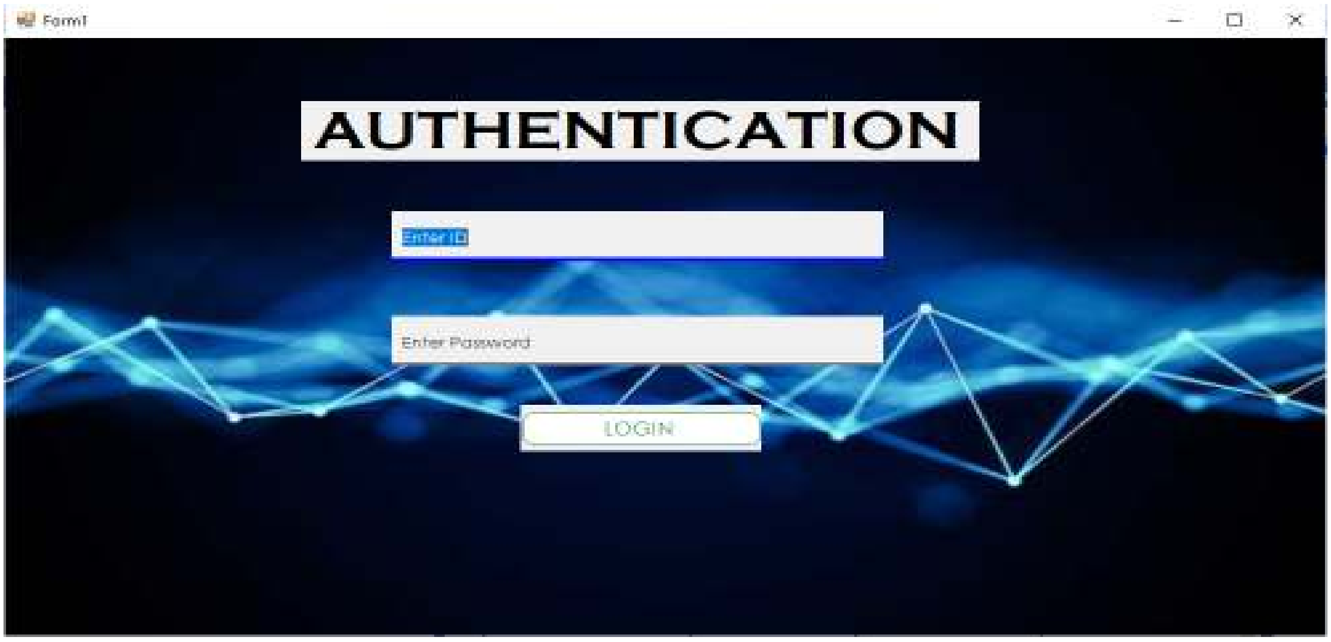


Fig 3.6.1.2 :- The Authentication Module of the user(Teacher)

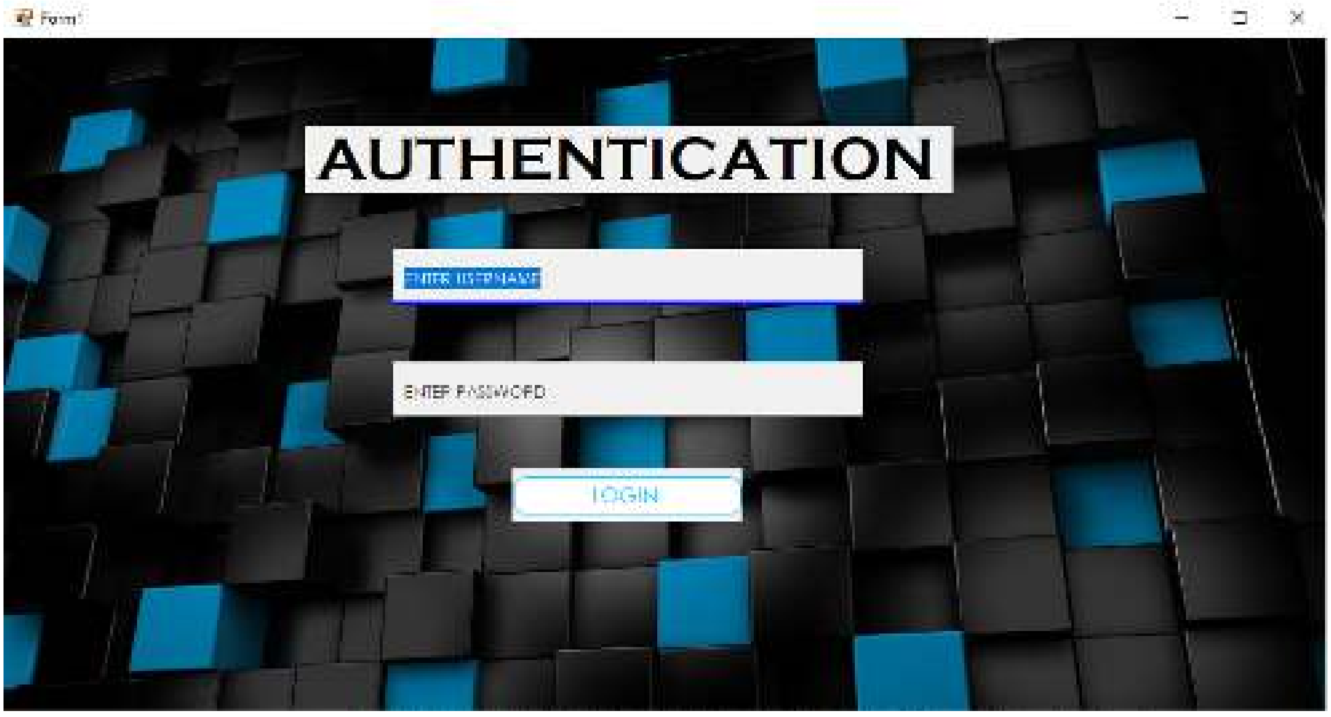


Fig 3.6.1.3 :- The First-level Authentication Page of the user(Admin)

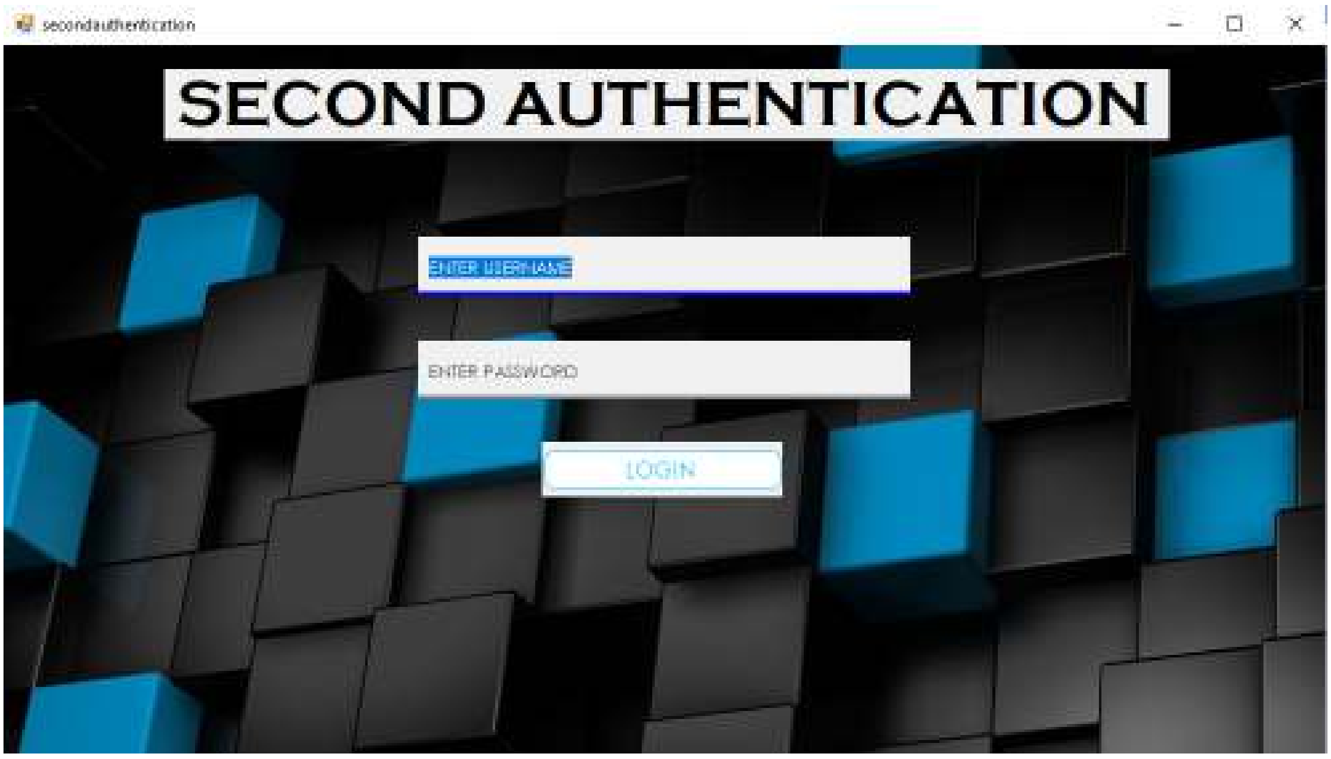
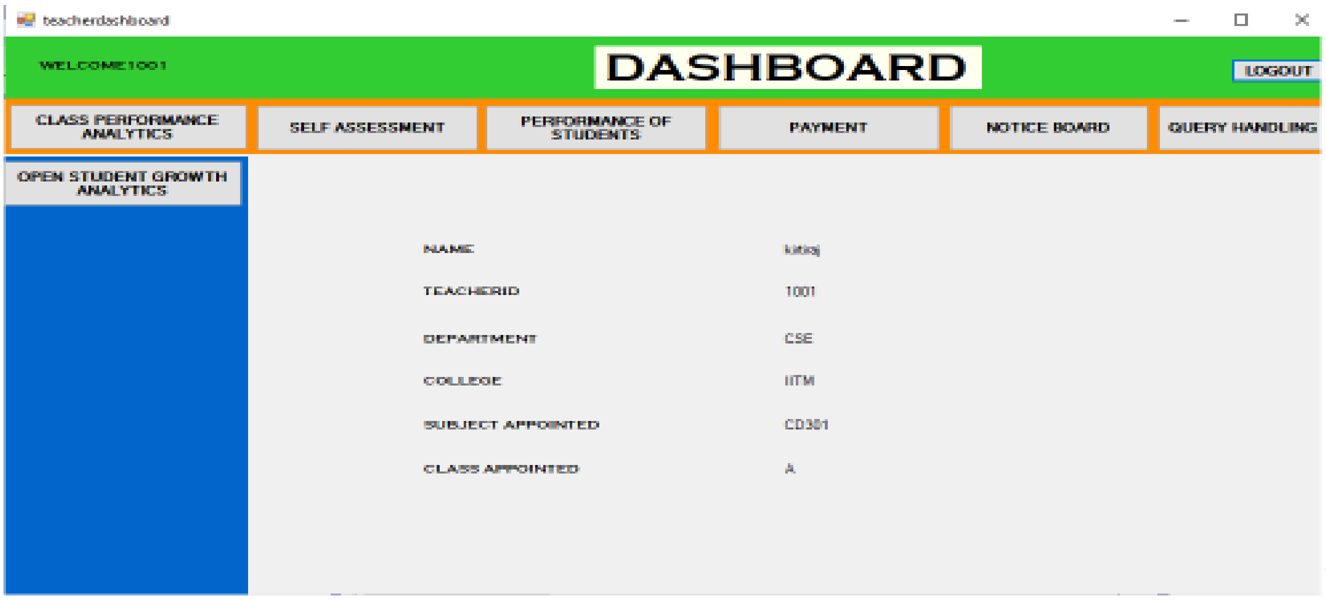


Fig 3.6.1.4 :- The Second-level Authentication Page of the user(Admin)

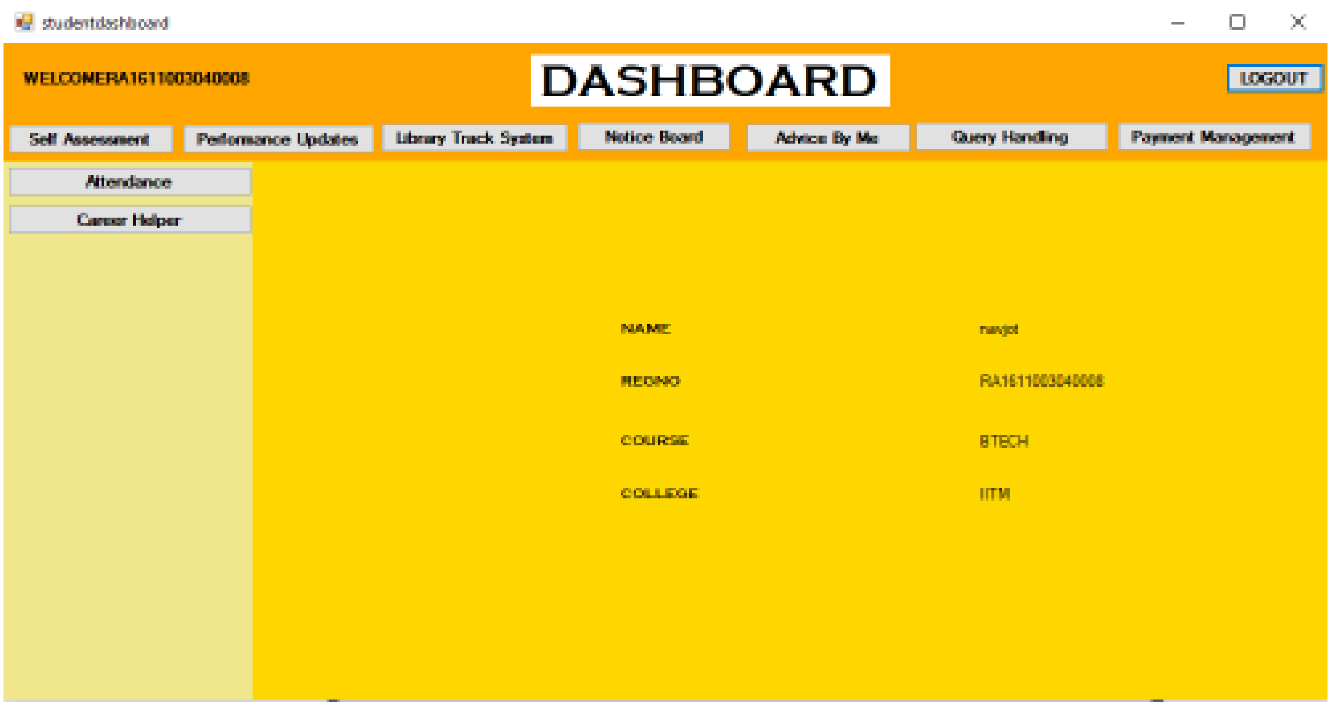
3.6.2 Navigator Menu Module

It is the module in which the user (teacher and student) after doing their one-level authentication get to access the navigator menu. It basically encapsulates all the other modules in order to display it as a single unit. It is the module where the user can access any information regarding himself/herself. It works just like user dashboard.



NAME	kitiraj
TEACHERID	1001
DEPARTMENT	CSE
COLLEGE	IITM
SUBJECT APPOINTED	CD301
CLASS APPOINTED	A

Fig 3.6.2.1 :- The Navigator Menu Module of the user(Teacher)



NAME	ranjot
REGNO	RA1611003040008
COURSE	BTECH
COLLEGE	IITM

Fig 3.6.2.2 :- The Navigator Menu Module of the user(Student)

3.6.3 Main Control Panel Module

It is the module which is found only in the admin part of the architectural diagram. It is the module which can be accessed by the top board members of the institution like – Founder, Managing Director, CEO, Chairman etc. This module basically proves that there is role for higher officials while using this system.

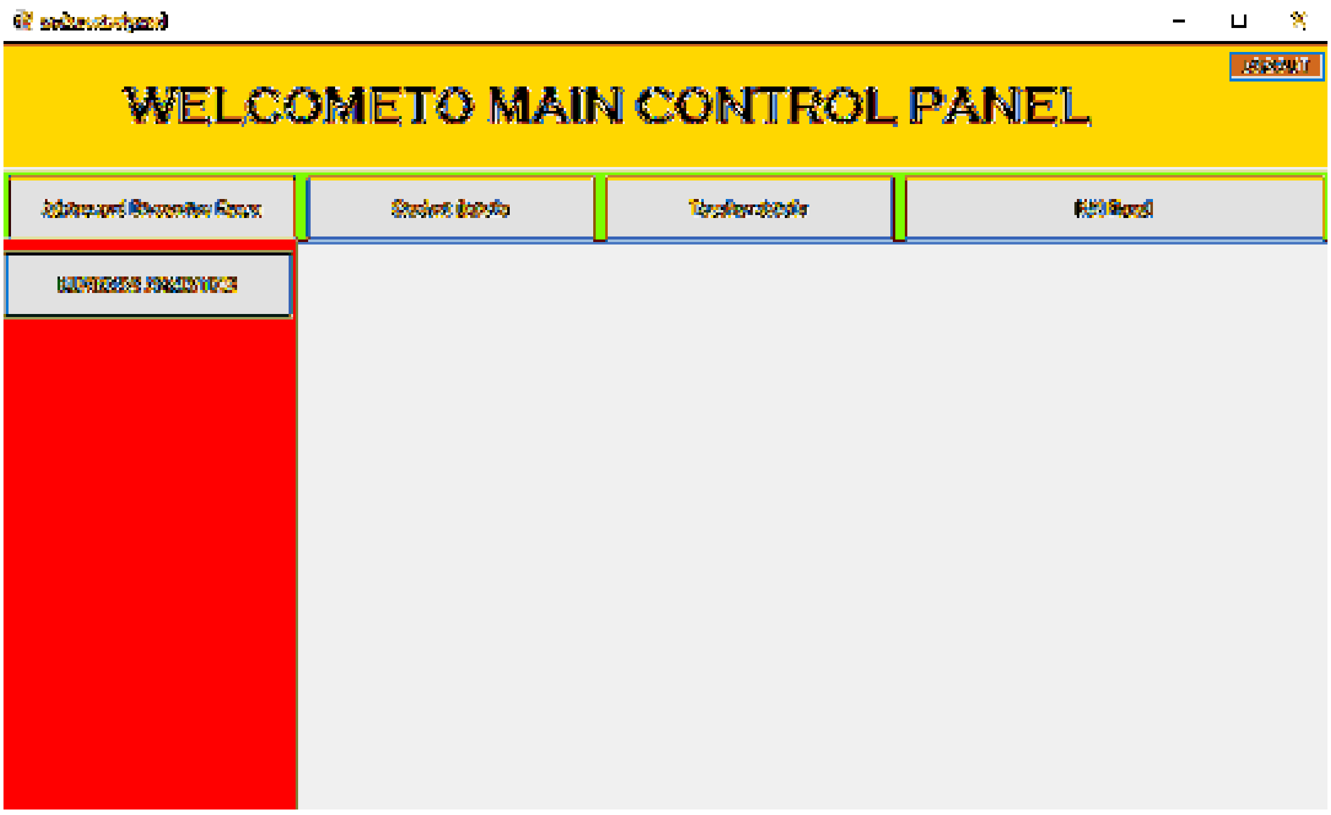


Fig 3.6.3.1 :- The Main Control Panel Module of the user(admin)

3.6.4 Lesser Admin Panel Module

It is the module which can be accessed by the rest of the admin personnel other than the main control panel persons. It consists of various departments like Controller of Examinations, Placement Cell, Public Relations etc. All the users working in particular department has to pass the two-level authentication such that it directs them to their department page.

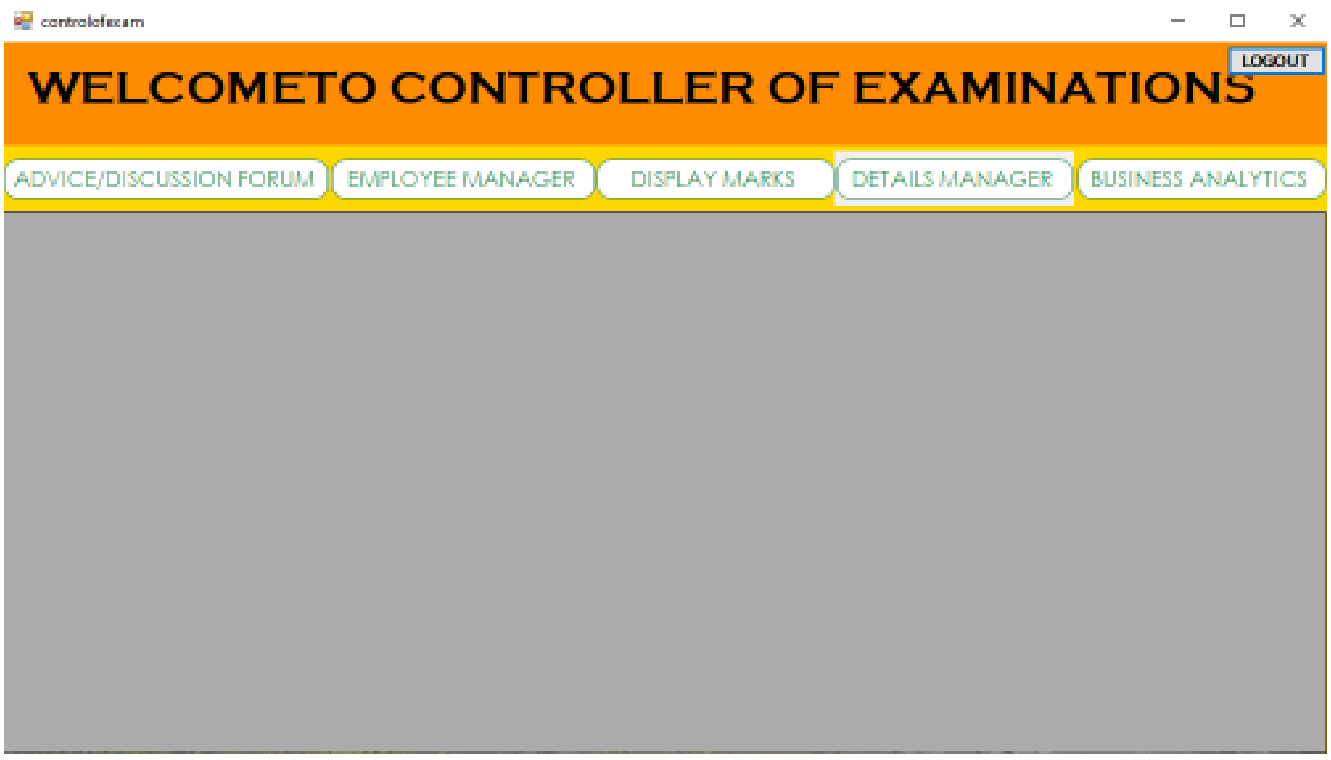


Fig 3.6.4.1 :- The Lesser Admin Panel Module of the user(admin) – Controller of Examination

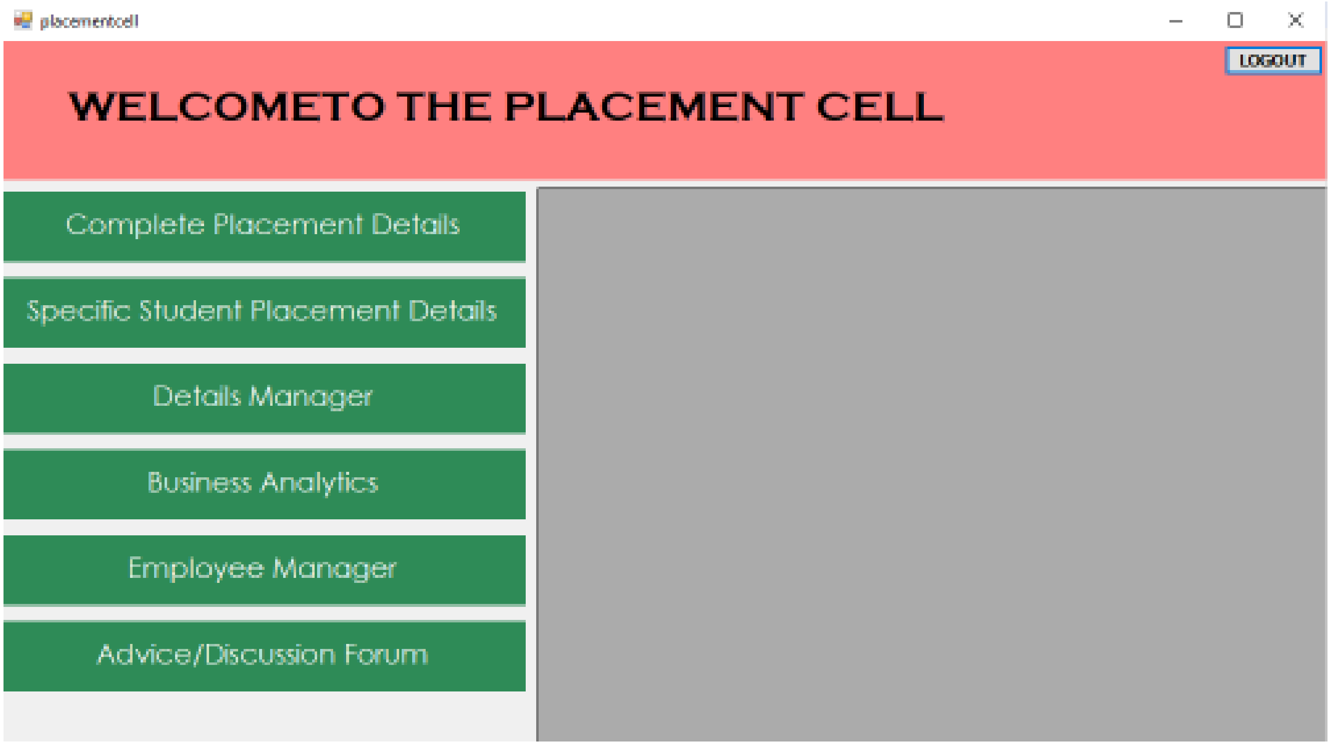


Fig 3.6.4.2 :- The Lesser Admin Panel Module of the user(admin) – Placement Cell

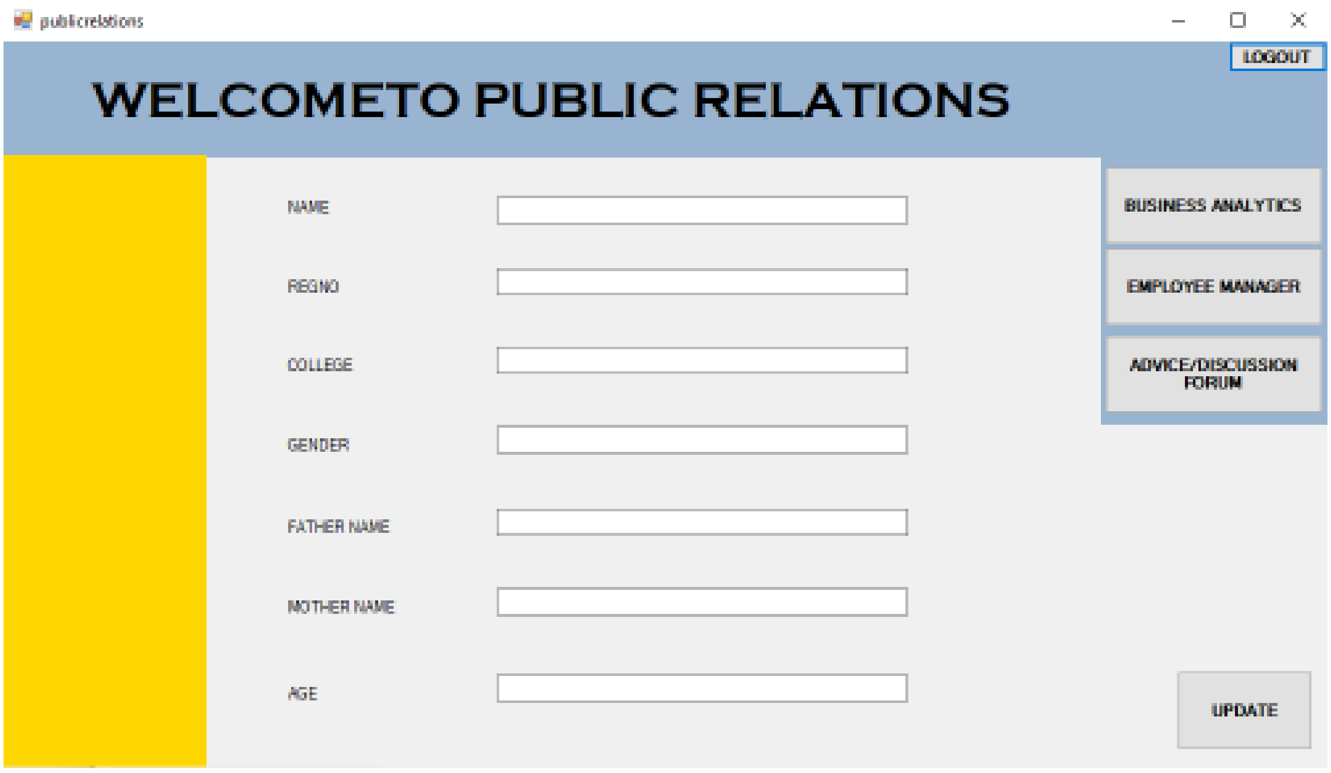


Fig 3.6.4.3 :- The Lesser Admin Panel Module of the user(admin) – Public Relations

3.6.5 IAM Panel Module

IAM stands for Identity Access Management. In this module it is only answerable to the main control panel. It has the right to access the information of all the employees/teachers, students, and the admin personnel. It is used to manage all the details of all the users including the sub admin personnel. It is a part of admin as shown in the system architecture.

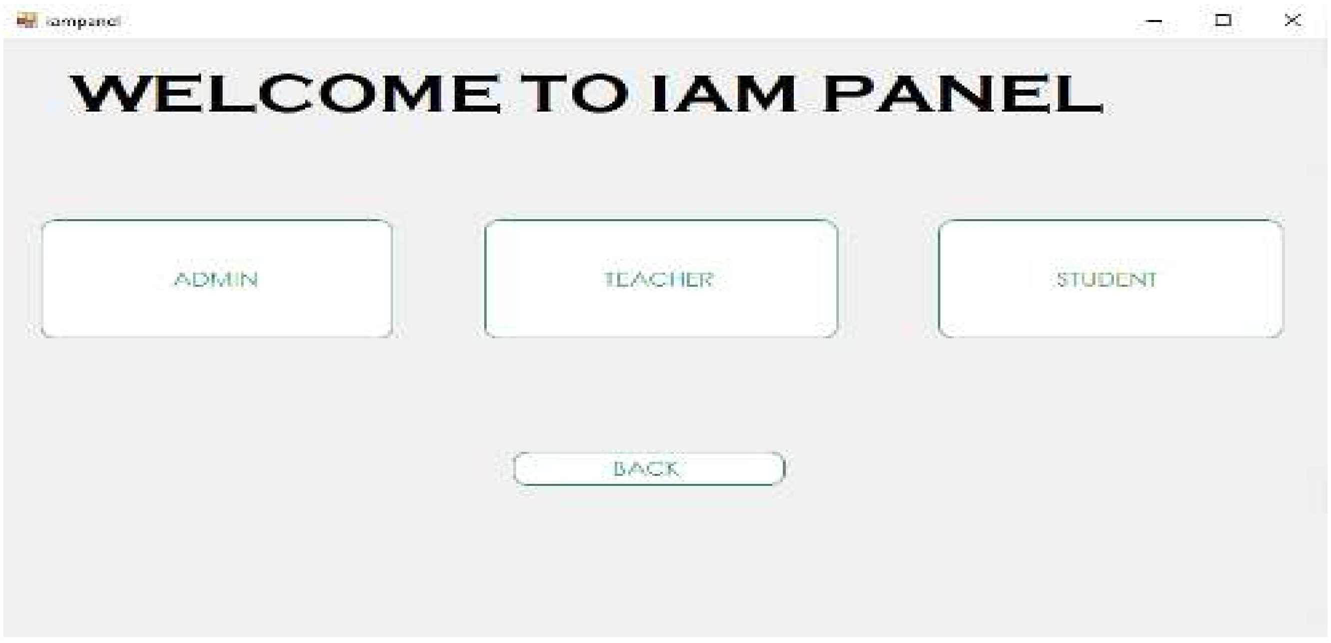


Fig 3.6.5.1 :- The IAM Panel (Home Page)

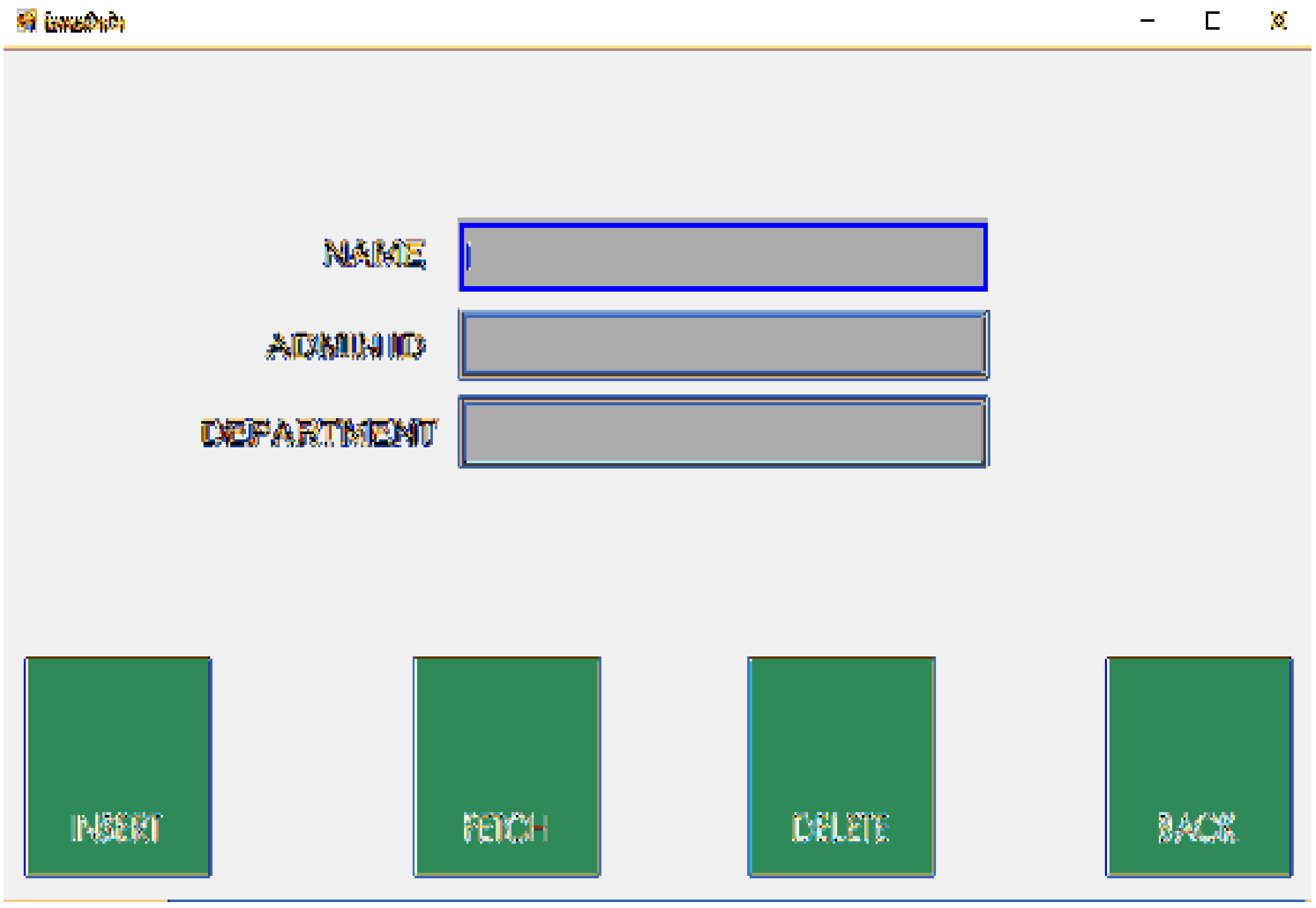


Fig 3.6.5.2 :- The IAM Panel (Admin)

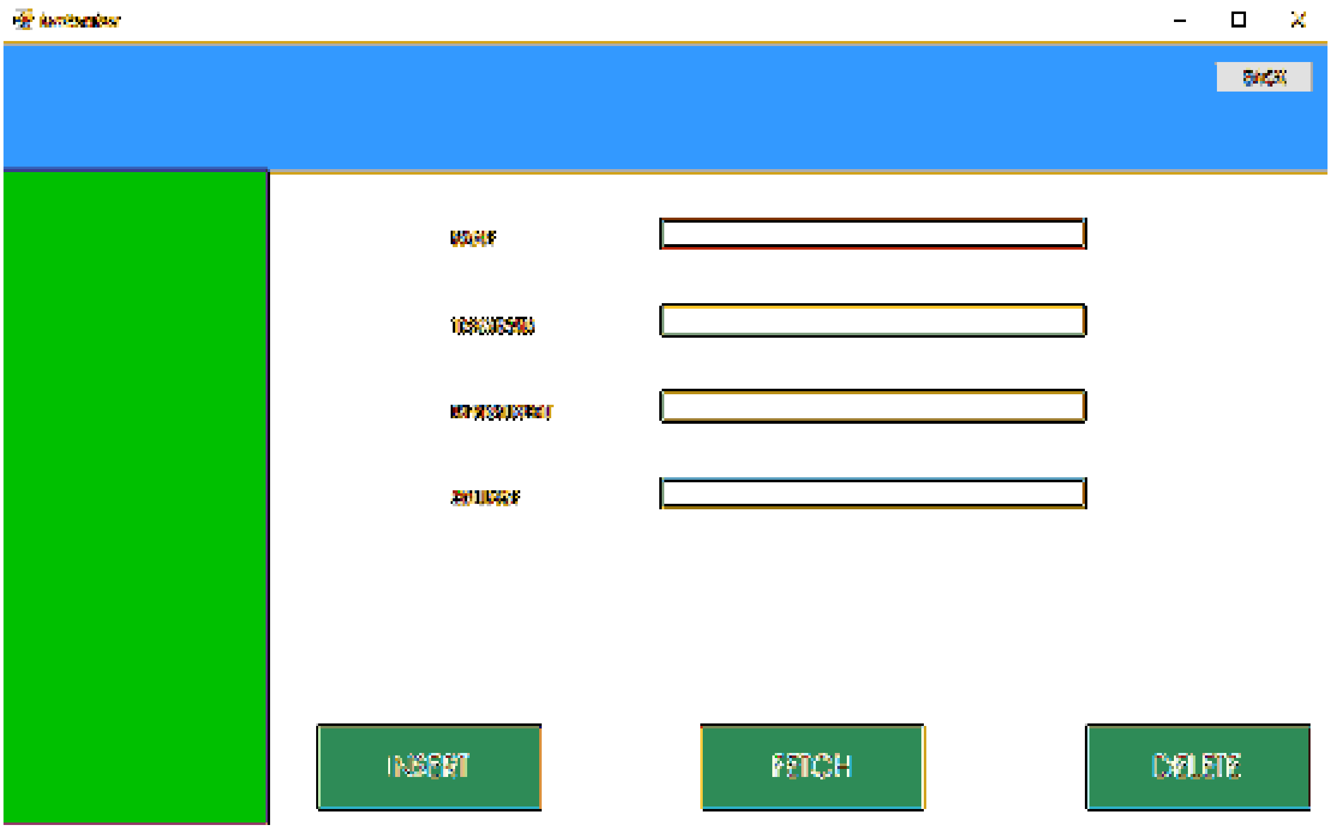


Fig 3.6.5.3 :- The IAM Panel (Teacher)

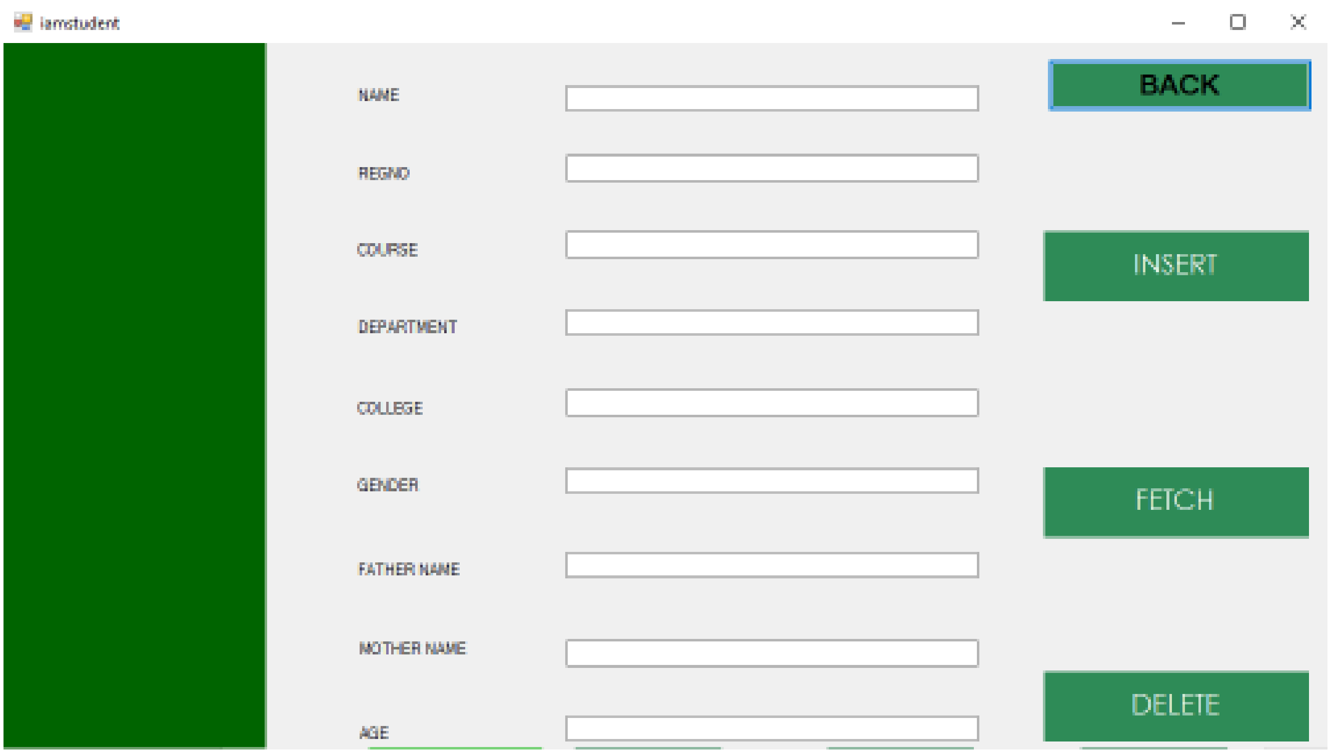
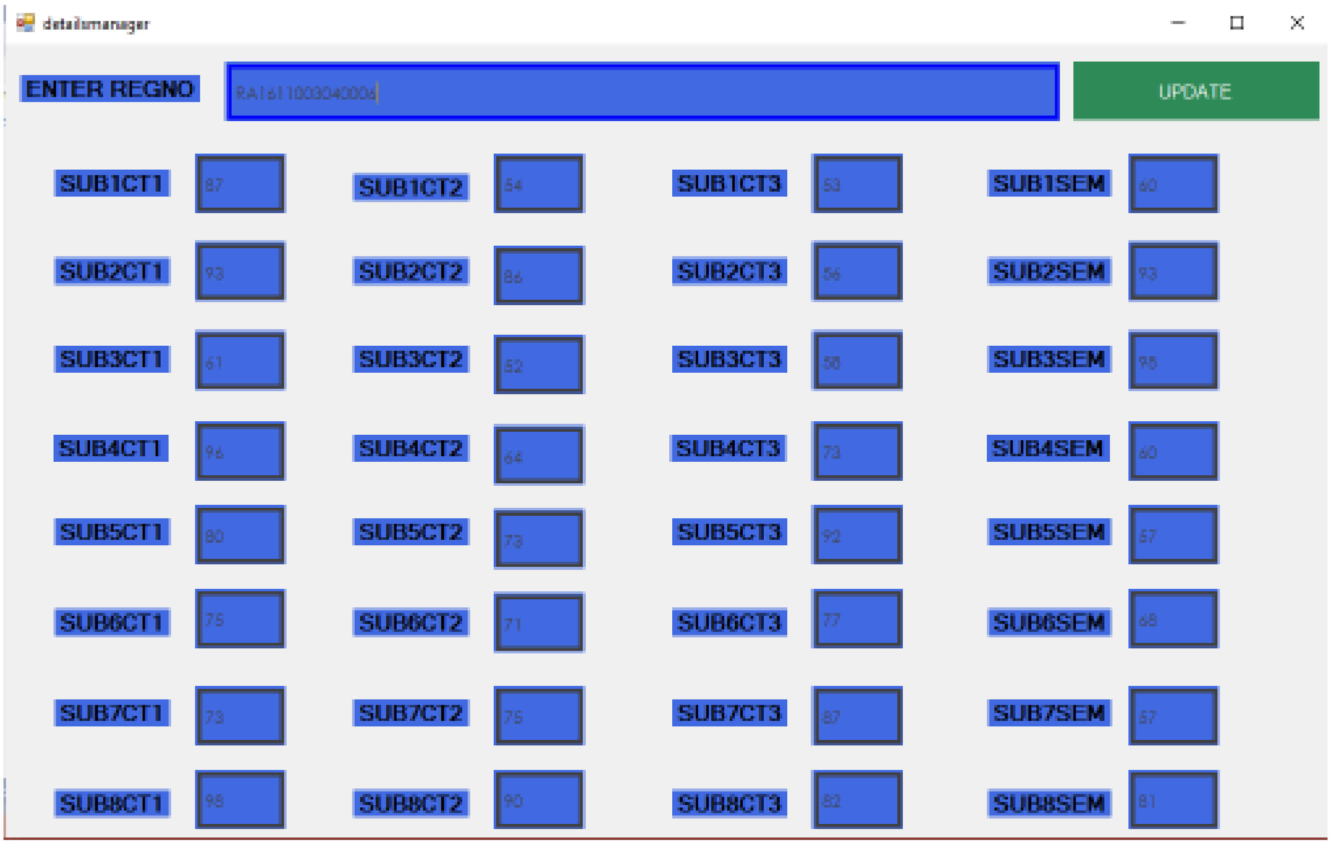


Fig 3.6.5.4 :- The IAM Panel (Student)

3.6.6 Details Manager Module

It is the module which can be accessed only by the sub admin personnel. It is used to manage the details of the employees/teachers and students only i.e if at all there is any information that needs to be updated like the marks, department etc.

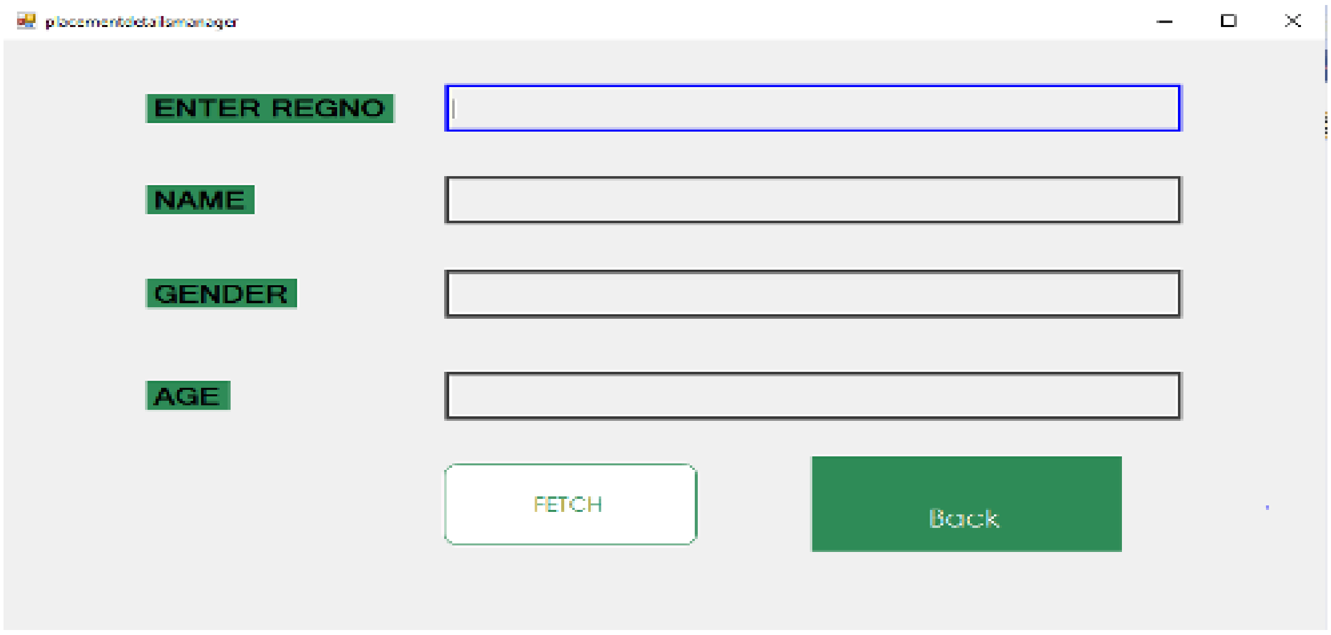

detailsmanager

ENTER REGNO: RA1611003040006

UPDATE

SUB1CT1	87	SUB1CT2	54	SUB1CT3	53	SUB1SEM	60
SUB2CT1	93	SUB2CT2	86	SUB2CT3	56	SUB2SEM	93
SUB3CT1	61	SUB3CT2	52	SUB3CT3	58	SUB3SEM	95
SUB4CT1	96	SUB4CT2	64	SUB4CT3	73	SUB4SEM	60
SUB5CT1	80	SUB5CT2	73	SUB5CT3	92	SUB5SEM	57
SUB6CT1	75	SUB6CT2	71	SUB6CT3	77	SUB6SEM	68
SUB7CT1	73	SUB7CT2	75	SUB7CT3	87	SUB7SEM	57
SUB8CT1	98	SUB8CT2	90	SUB8CT3	82	SUB8SEM	81

Fig 3.6.6.1 :- The Details Manager for Controller of Examination – lesser admin panel


placementdetailsmanager

ENTER REGNO	
NAME	
GENDER	
AGE	

Fig 3.6.6.2 :- The Details Manager for Placement Cell – lesser admin panel

3.6.7 Employee Manager Module

It is the module which is applicable for the sub-admin personnel in the lesser admin panel. It is used to manage the personal information of the employees/teachers and students. Hence the employee manager module and the details manager module together manage the details of all the employees/teachers and students.

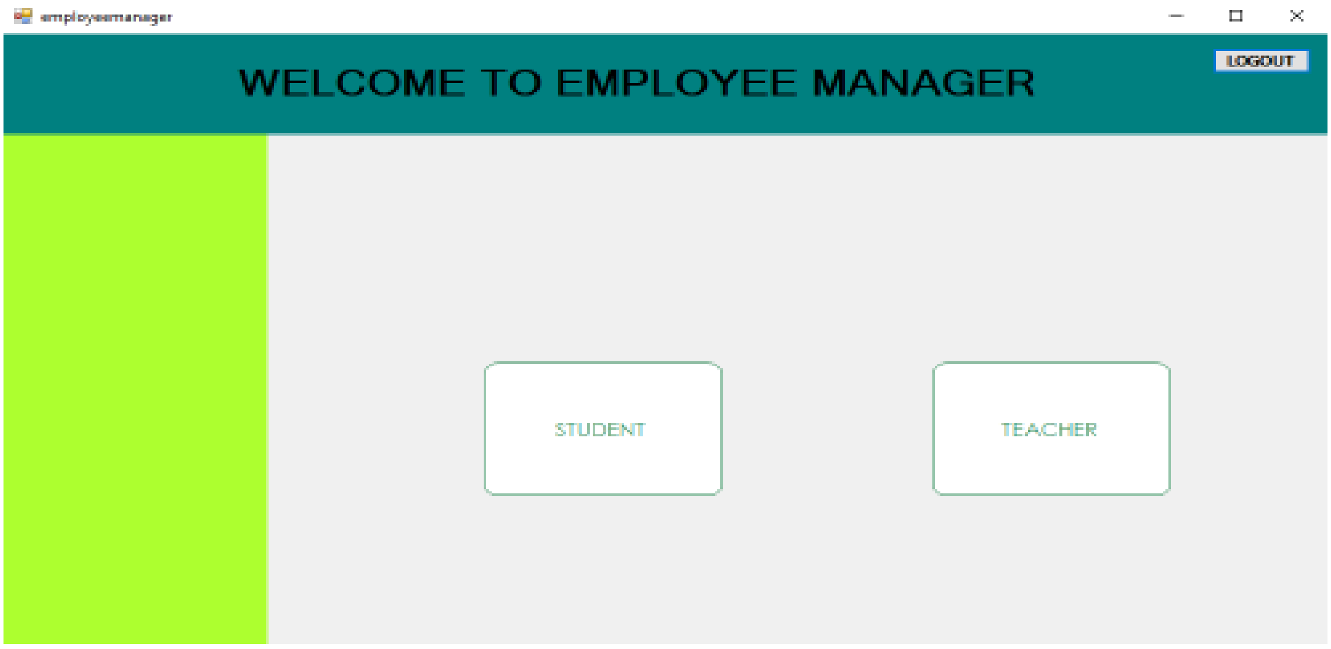


Fig 3.6.7.1 :- The Employee Manager for sub admin personnel – lesser admin panel

3.6.8 Business Analytics Module

It is the module in which the all the admin personnel use the various types of analytics shown in the system in order to use it for various business purposes like marketing the institution etc. Here various types of analytics like perspective of marks, teachers, placement of students, admitted students etc, and then combine it in such a way that it can be useful information.

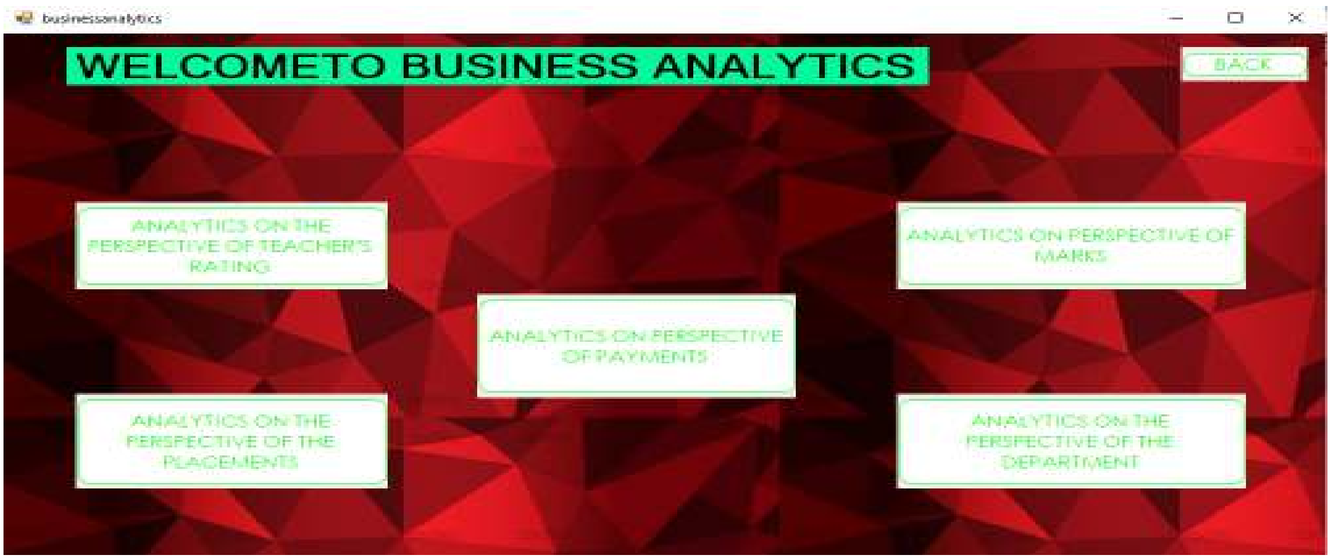
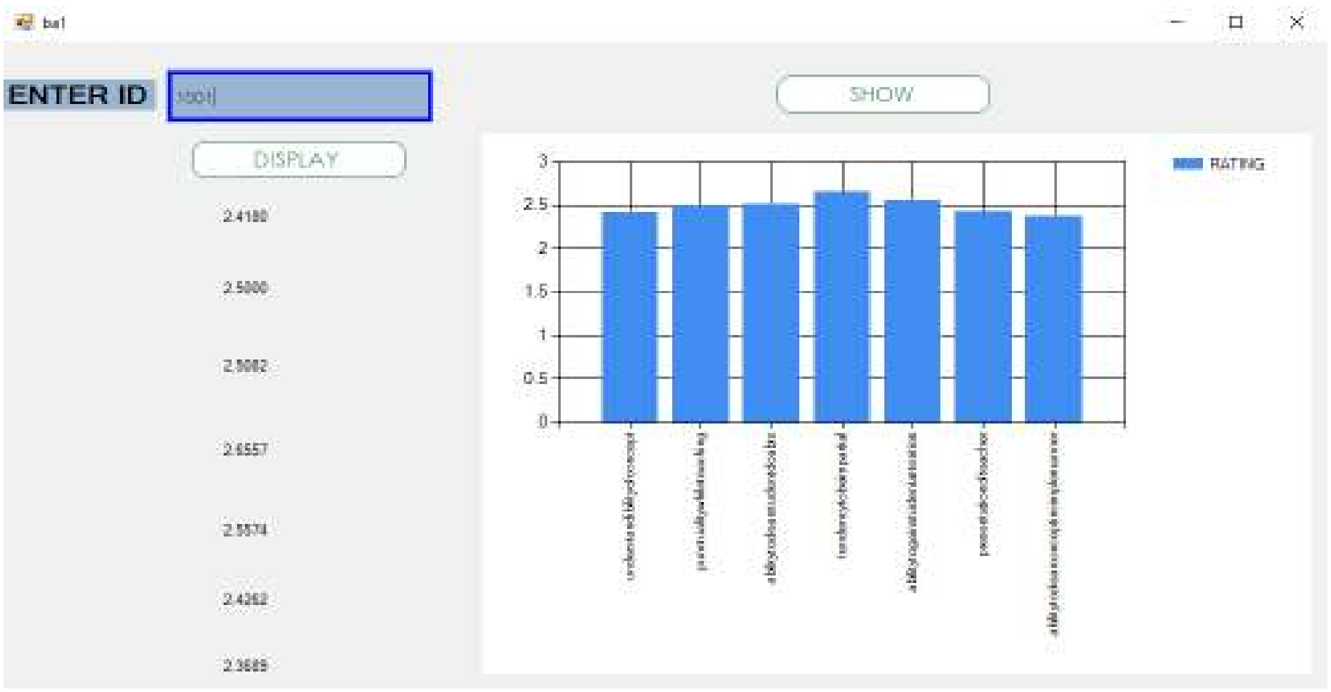
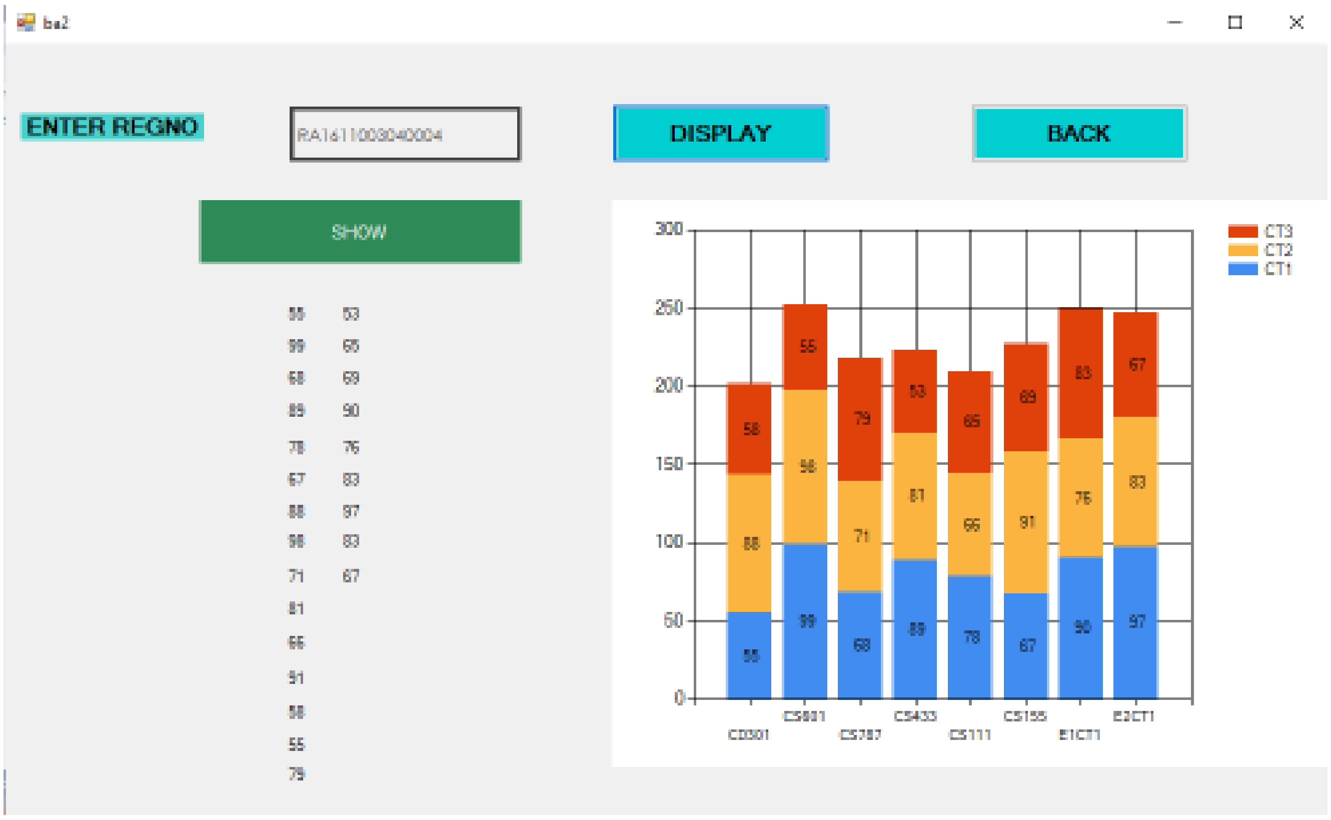

businessanalytics
WELCOMETO BUSINESS ANALYTICS
BACK
ANALYTICS ON THE PERSPECTIVE OF TEACHER'S RATING
ANALYTICS ON PERSPECTIVE OF MARKS
ANALYTICS ON PERSPECTIVE OF PAYMENTS
ANALYTICS ON THE PERSPECTIVE OF THE PLACEMENTS
ANALYTICS ON THE PERSPECTIVE OF THE DEPARTMENT

Fig 3.6.8.1 :- The Business Analytics Module (Home Page)



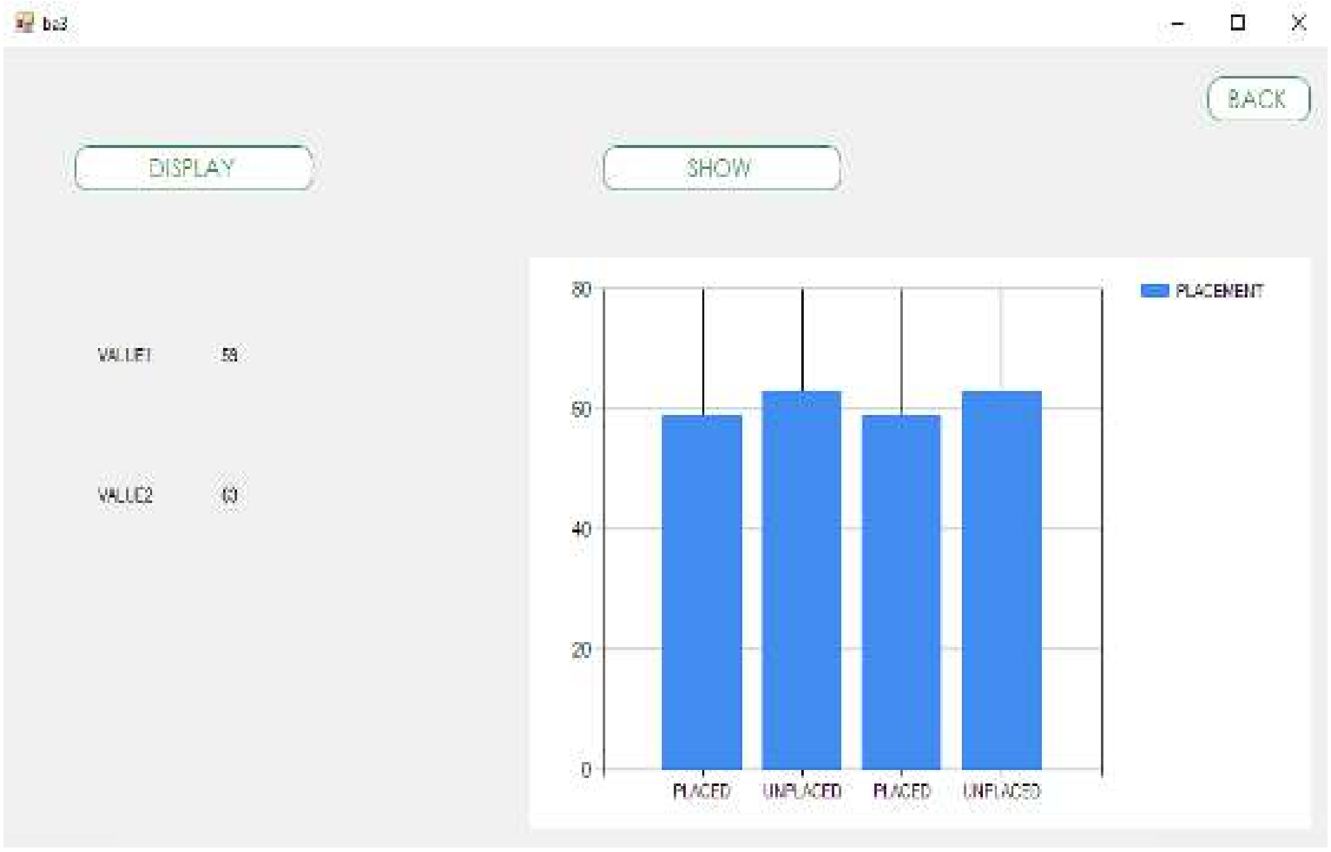
```
2.4180

2.5000

2.5082

2.6557

2.5574

2.4262

2.3689
```

Fig 3.6.8.2 :- The Business Analytics Module (Perspective of Teacher rating)



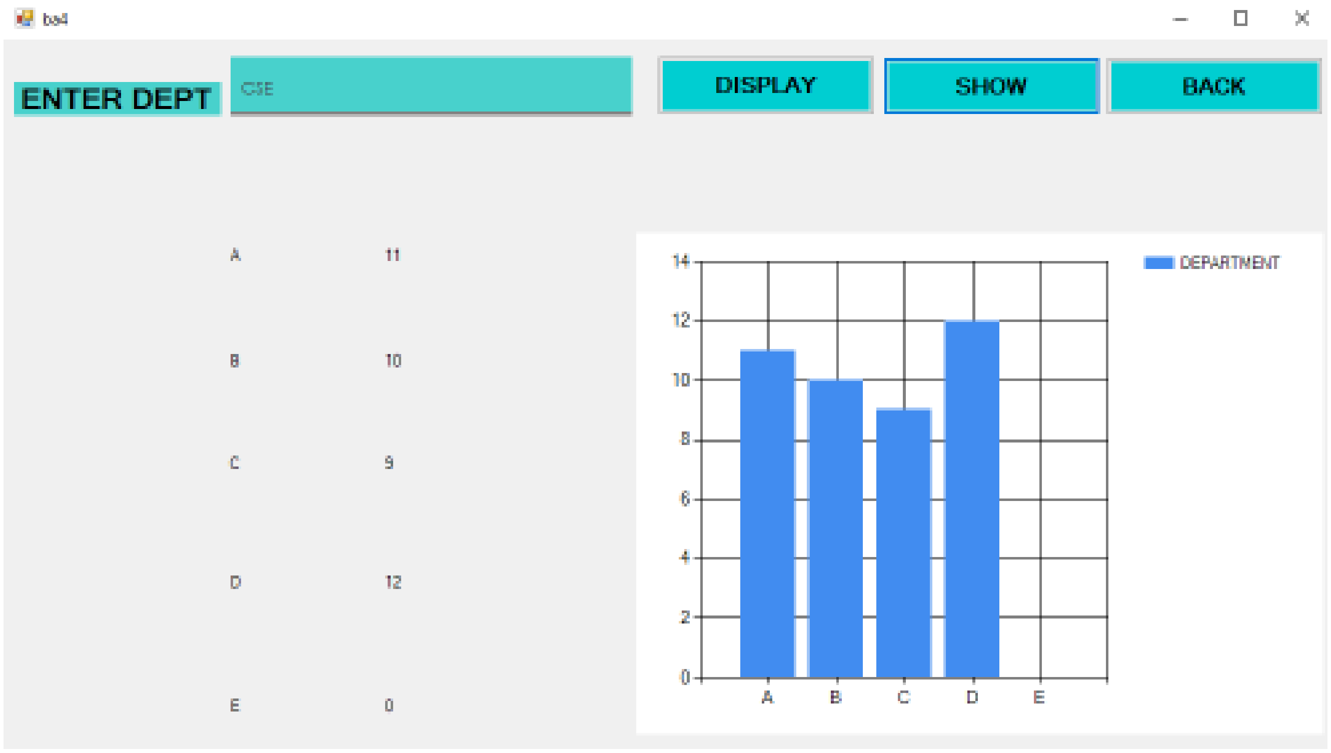
```
55    53
99    65
68    69
89    90
78    76
67    83
88    97
98    83
71    67
81
66
91
58
55
79
```

Fig 3.6.8.3 :- The Business Analytics Module (Perspective of Marks)



VALUE1	59
VALUE2	63

Fig 3.6.8.4 :- The Business Analytics Module (Perspective of Placement of Students)



A	11
B	10
C	9
D	12
E	0

Fig 3.6.8.5 :- The Business Analytics Module (Perspective of Department)

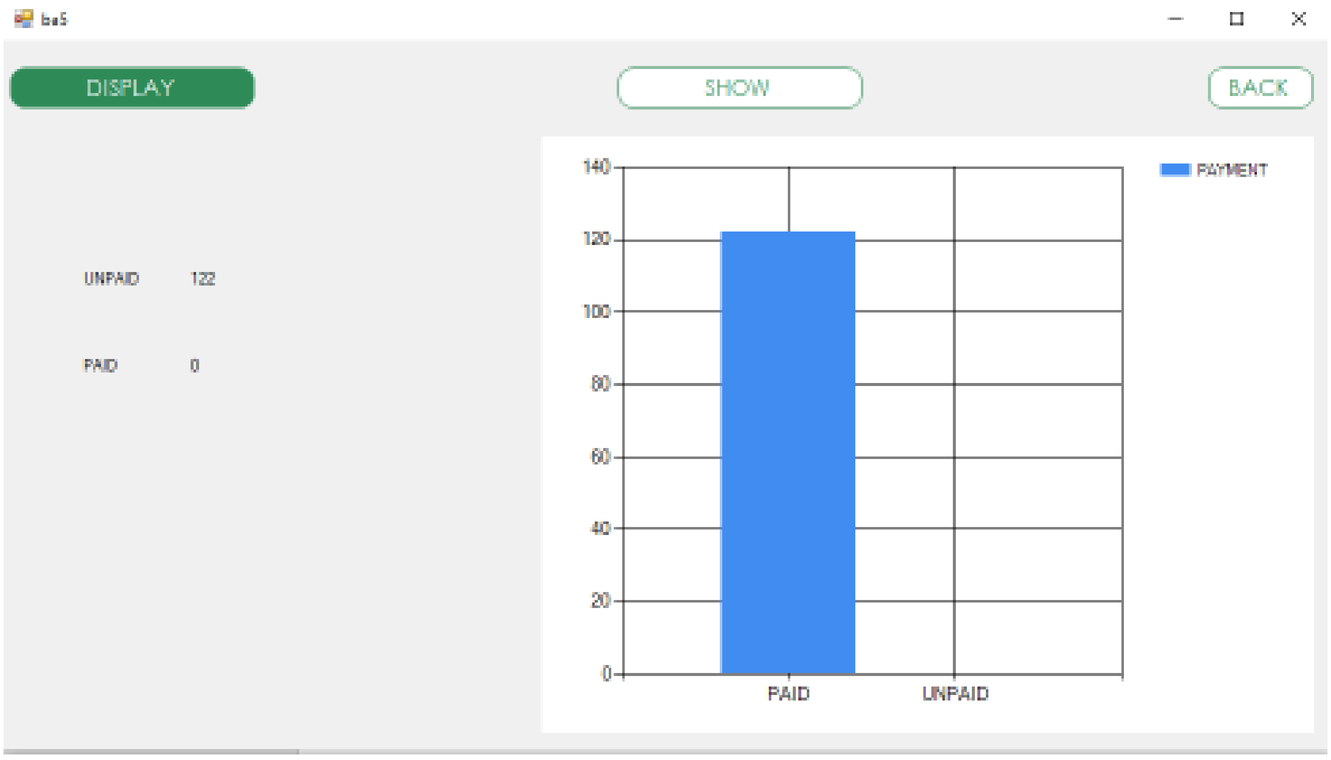


Fig 3.6.8.6 :- The Business Analytics Module (Perspective of Payments of students)

3.6.9 Advice/Discussion Forum Module

It is the module which handles and manages all the queries/feedbacks and other complaints of the teacher and student. Here there are two cases – if the queries/feedbacks and other complaints of the student is resolved then the status is updated as 1. Otherwise if the queries/feedbacks and other complaints of the teacher is resolved then the status is updated as 2.

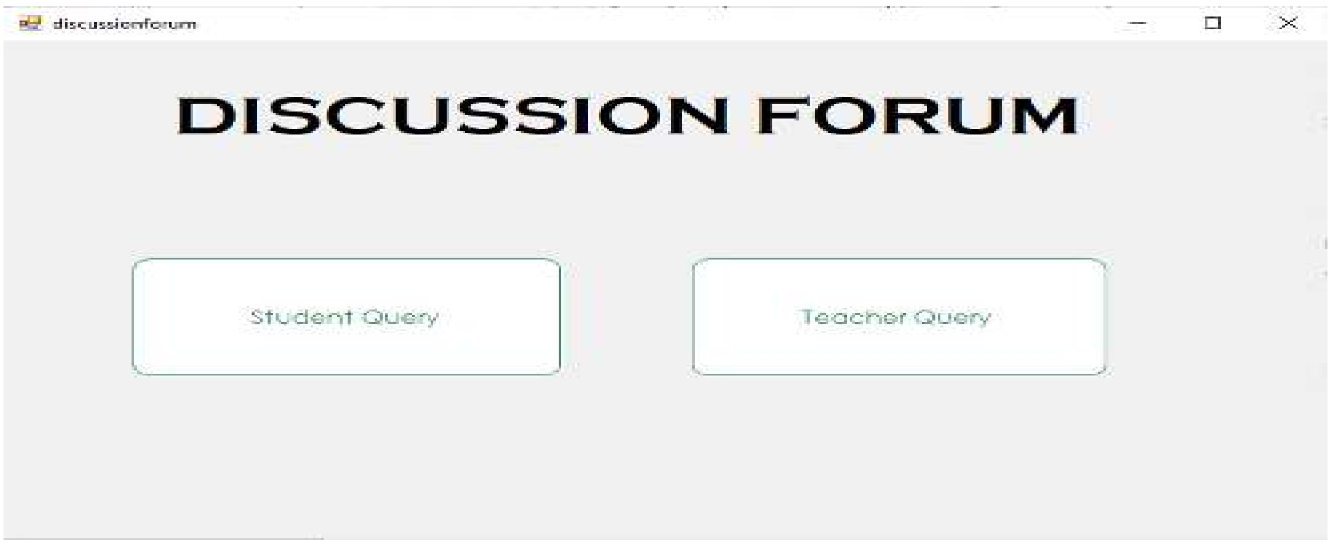


Fig 3.6.9.1 :- The Advice/Discussion Module (Homepage)

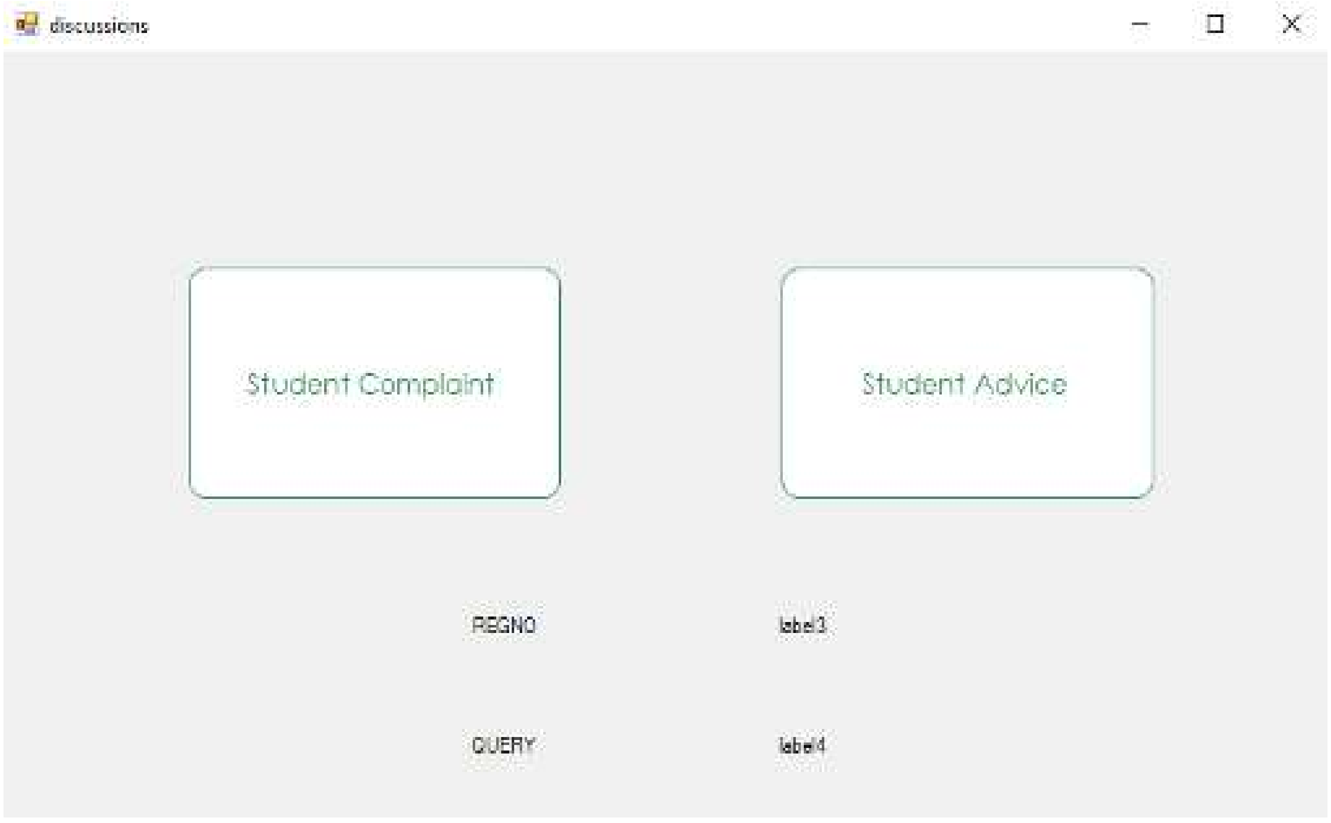


Fig 3.6.9.2 :- The Advice/Discussion Module (of student)

3.6.10 Performance updates of students Module

It is the module in which the student as well as the teacher can view the subject-wise performance of each subject by the student in the various respective examinations like cycle tests. It is found in the student part as well as the teacher's part of the system architecture diagram. Here the performance analytics is shown in a more meaningful manner.

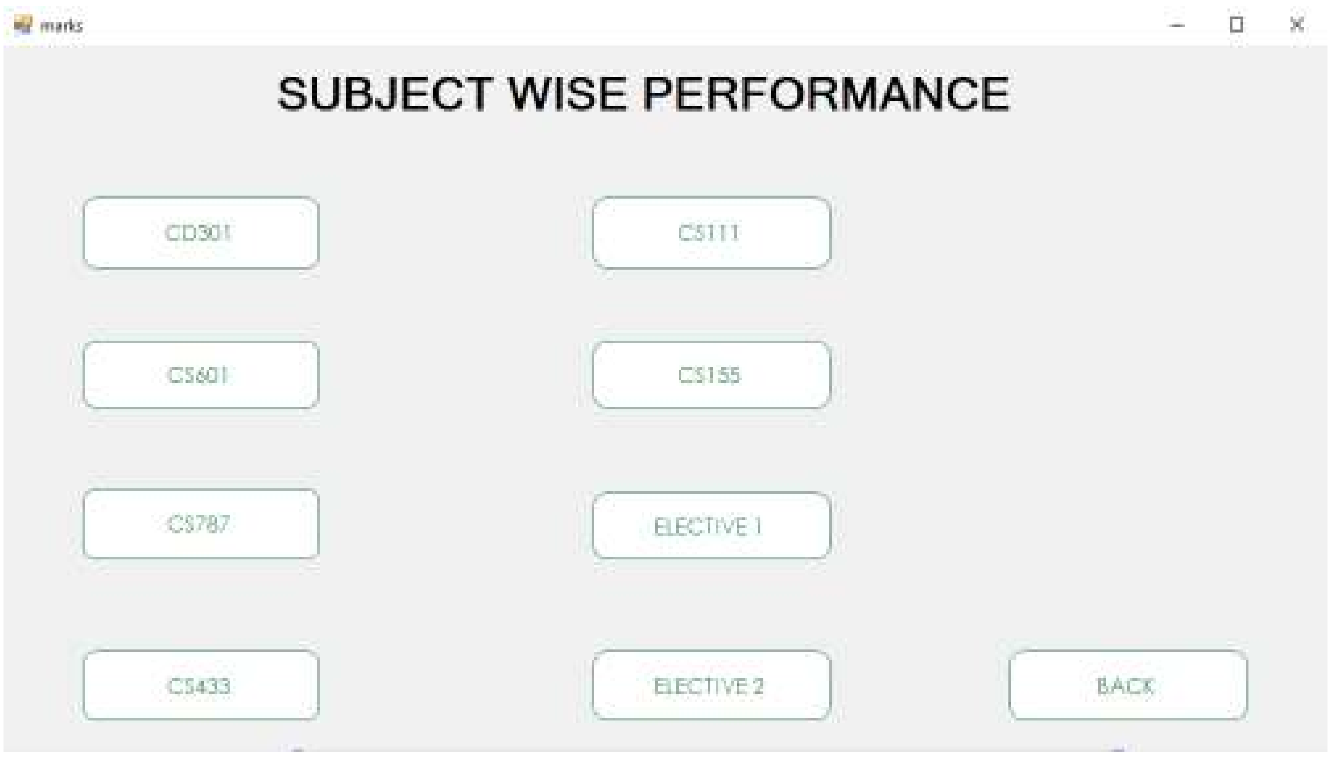
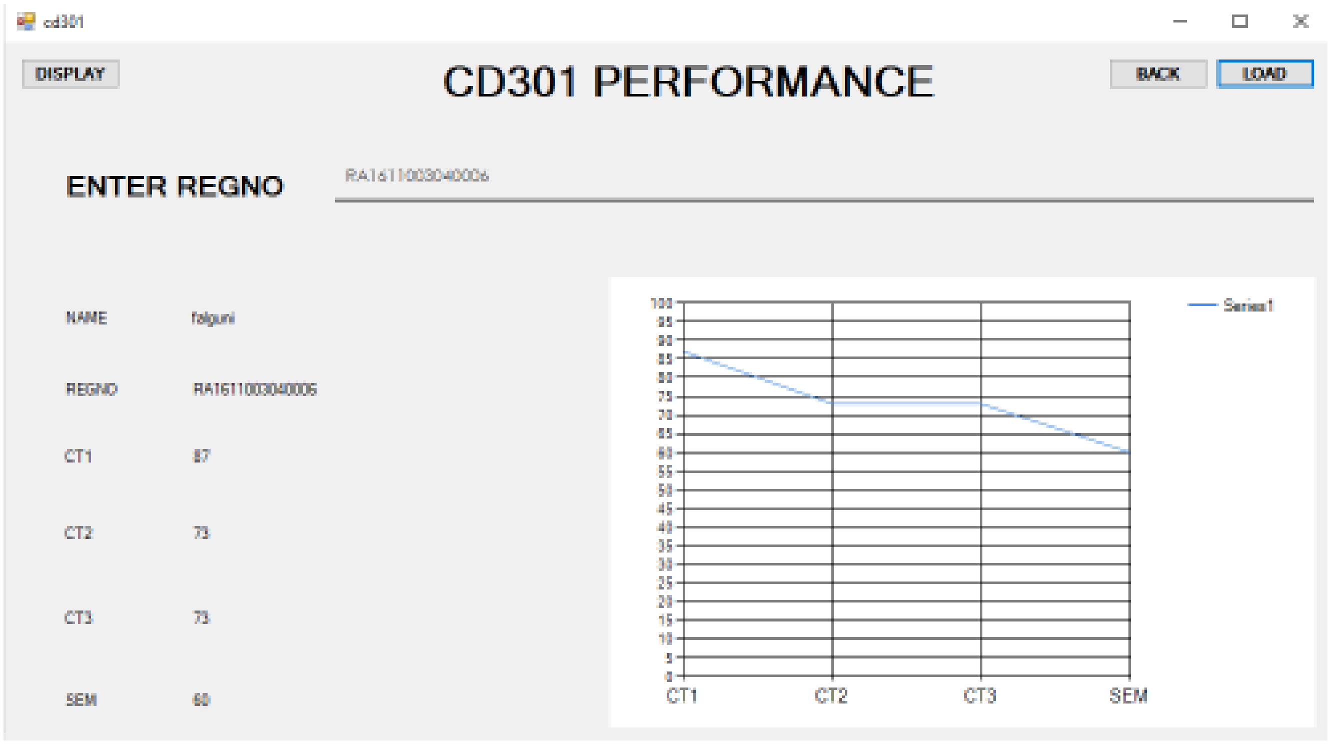


Fig 3.6.10.1 :- The Performance updates of Students Module (of student)



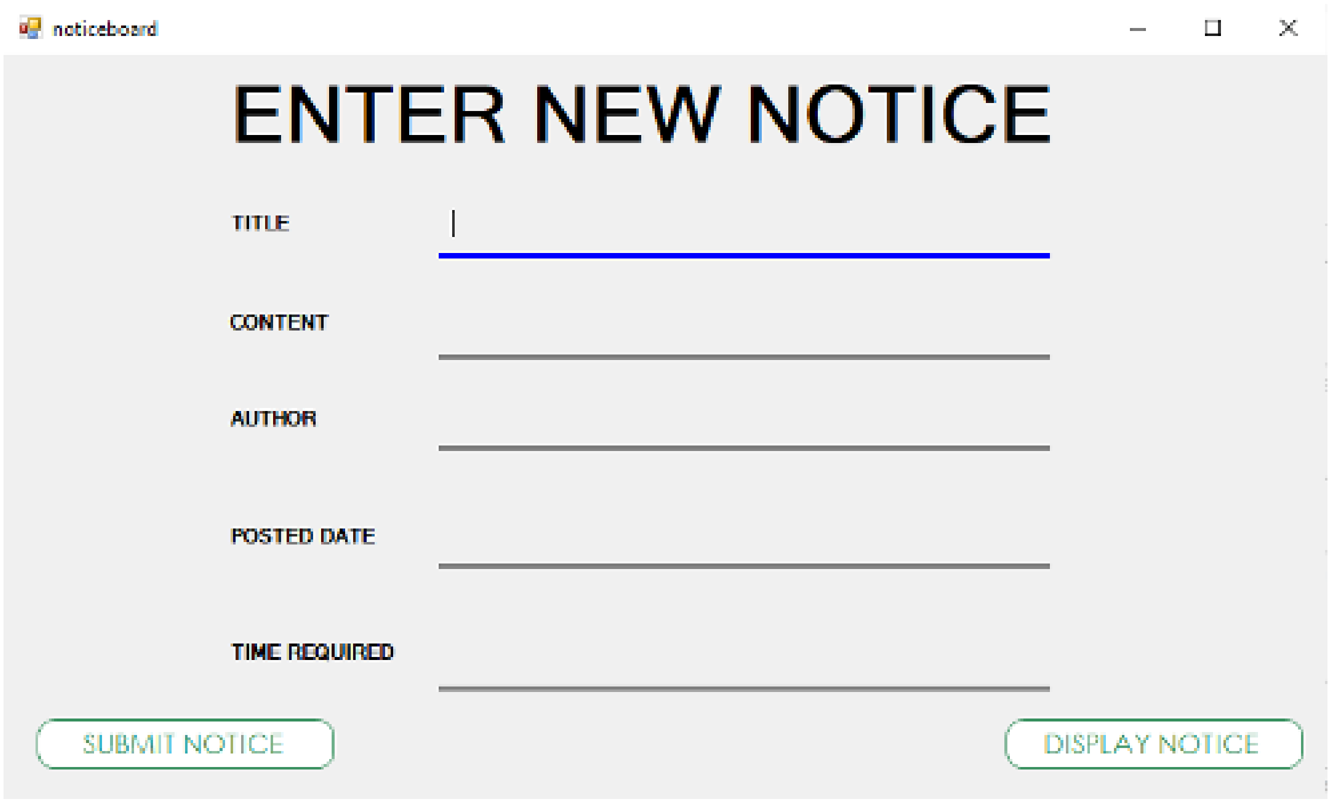
ENTER REGNO RA1611003040006

NAME	falguni
REGNO	RA1611003040006
CT1	87
CT2	73
CT3	73
SEM	60

Fig 3.6.10.2 :- The Performance updates of a subject of a Student Module

3.6.11 Digital Notice Board

It is the module in which it shows the various notices or events in the digital format in such a way that it can be viewed by the whole university at that moment. It is found in the student's part as well as the teacher's part in the system architecture. Here the student can only view notices whereas the teacher can add new notices if required.



noticeboard

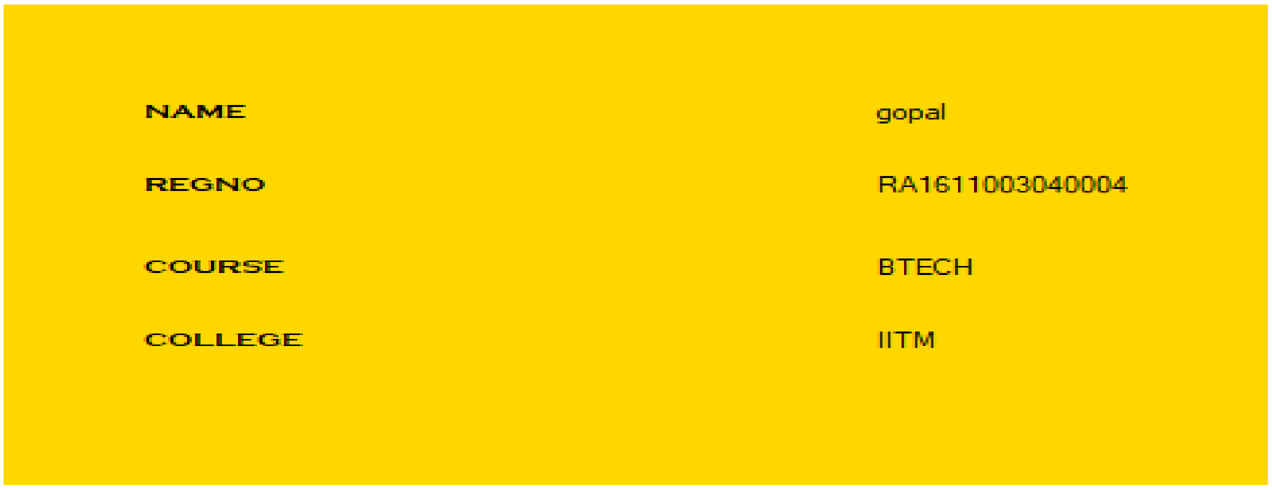
ENTER NEW NOTICE

TITLE	
CONTENT	
AUTHOR	
POSTED DATE	
TIME REQUIRED	

Fig 3.6.11.1 :- The Digital Notice Board

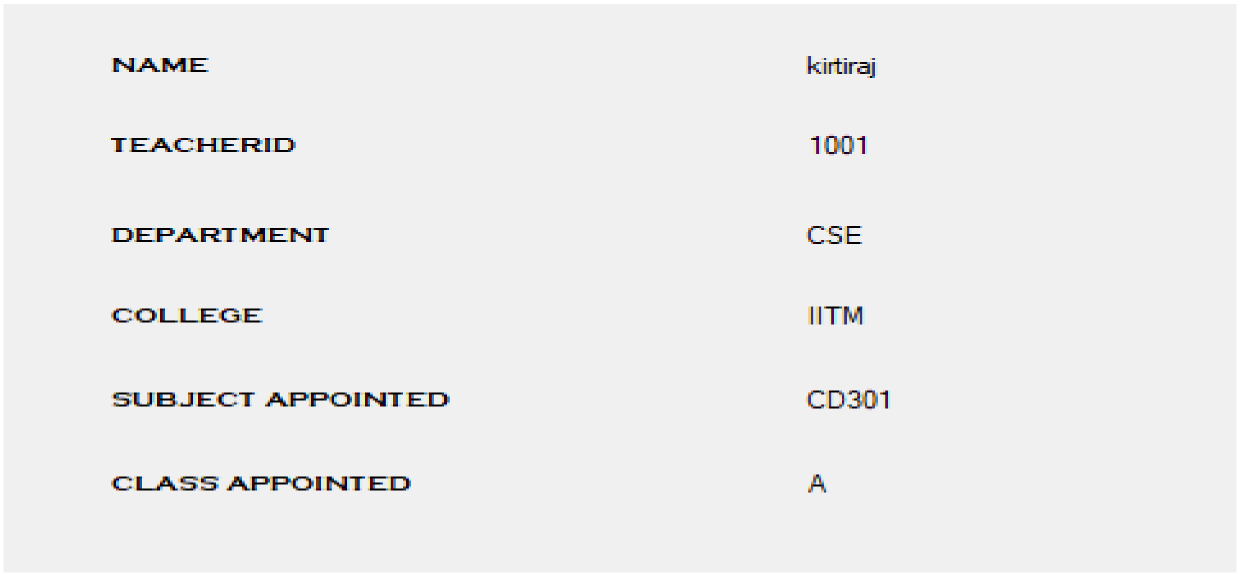
3.6.12 Self-Profile Module

It is the module in which it shows you the basic profile details like name, register number, department, college etc. This module is found in the teacher's part as well as in the student's part as seen in the system architecture diagram. It is not found in the admin's part directly. It is visible directly in the navigator menu as soon as the authentication process is finished.



NAME	gopal
REGNO	RA1611003040004
COURSE	BTECH
COLLEGE	IITM

Fig 3.6.12.1 :- The Self-Profile of student



NAME	kirtiraj
TEACHERID	1001
DEPARTMENT	CSE
COLLEGE	IITM
SUBJECT APPOINTED	CD301
CLASS APPOINTED	A

Fig 3.6.12.2 :- The Self-Profile of Teacher

3.6.13 Feedback/Query handling Module

It is a module in which both the student and teacher can enter the complaints or other queries and other suggestions, which are required to be resolved by either the teacher or the admin personnel. It is found in the student's part as well as in the teacher's part as shown in the system architecture diagram. The feedback/query handling module of both the teacher as well as the student are combined and then displayed in the advice/discussion forum module in order to resolve the complaints or other queries given by the employees/teacher and student.

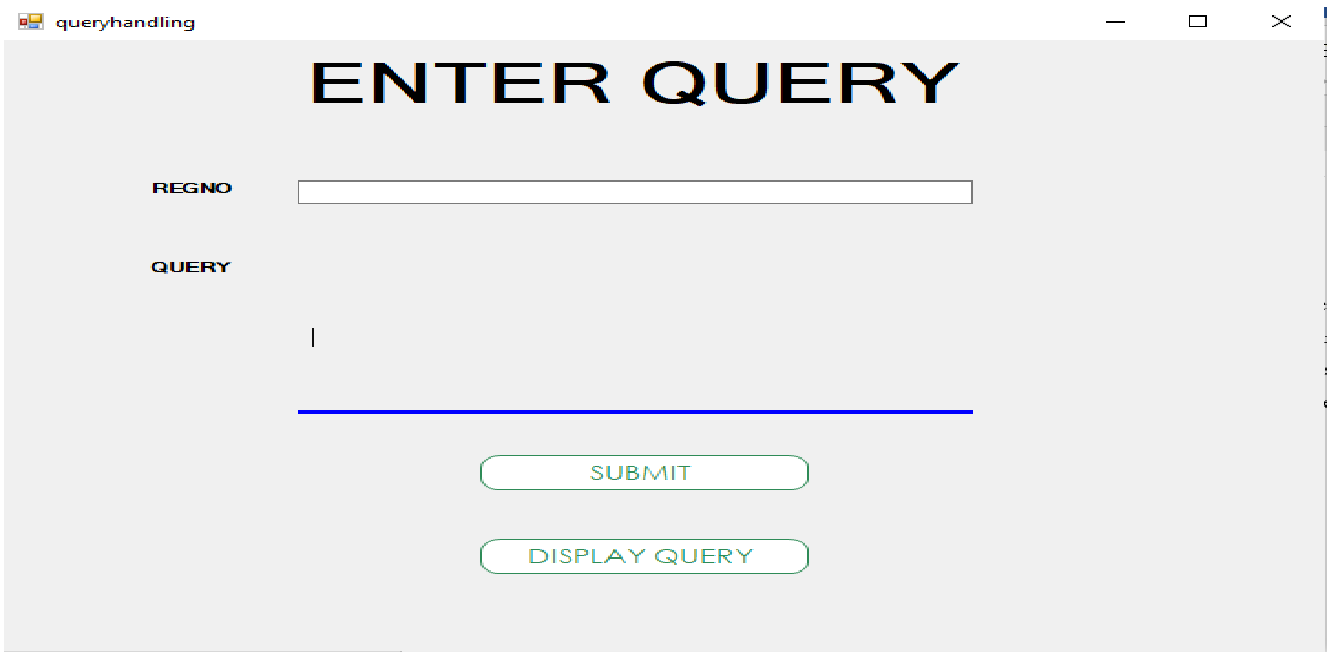


Fig 3.6.13.1 :- The Feedback/Query Handling of Teacher

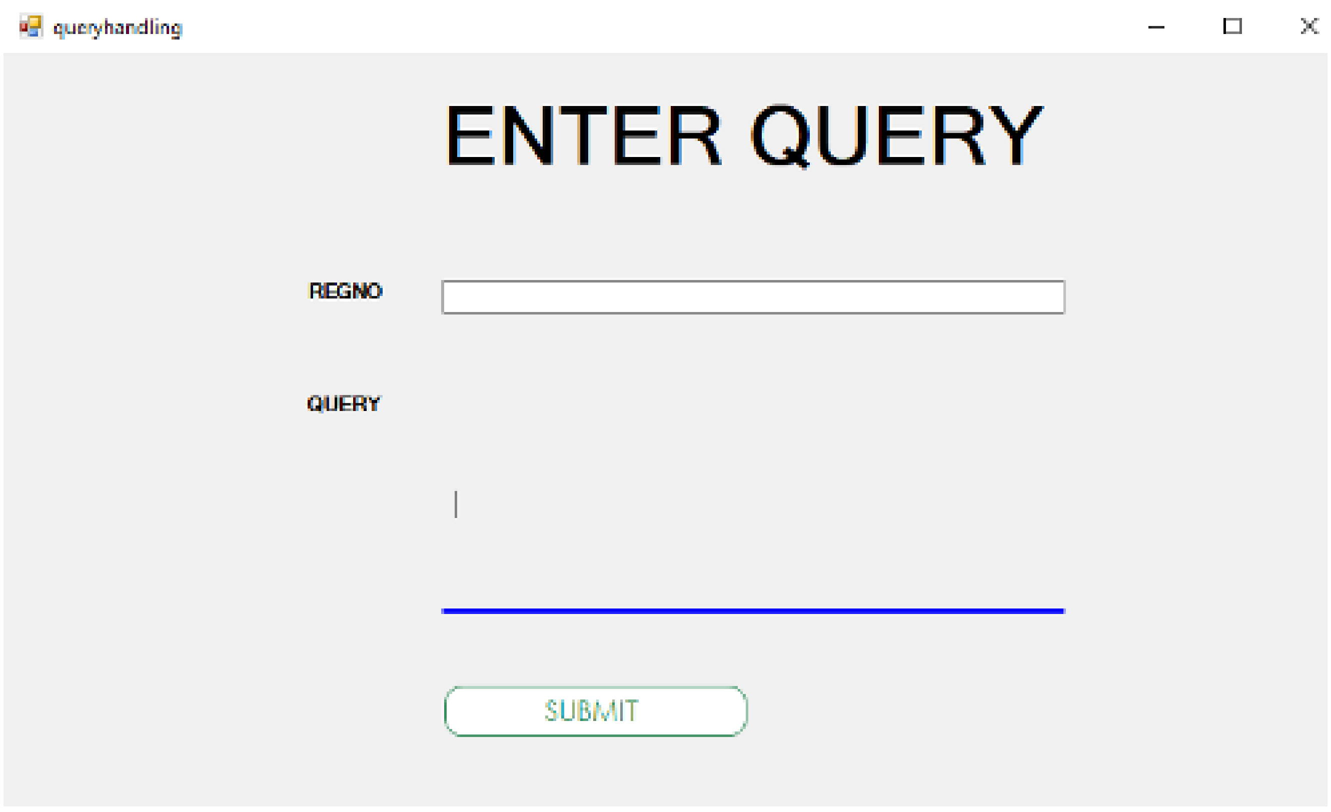
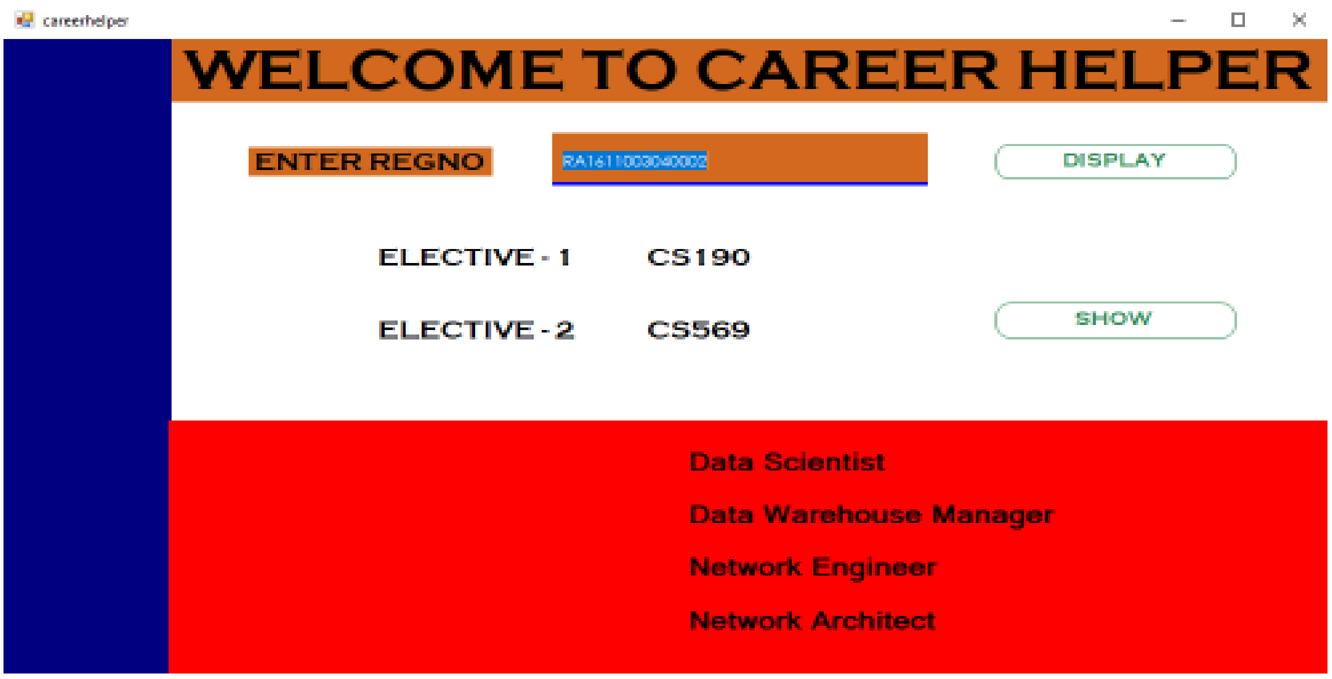


Fig 3.6.13.2 :- The Feedback/Query Handling of Student

3.6.14 Career Helper Module

It is the module in which it first fetches the elective subjects chosen by the student based on the register number entered. After fetching the subjects list it shows the various possibilities of career paths that the student can chose to pursue. Here the career clustering algorithm is used in order to map the elective subjects chosen and then showing the student the various career paths.


careerhelper

WELCOME TO CAREER HELPER

ENTER REGNO RA1611003040002

ELECTIVE - 1 CS190

ELECTIVE - 2 CS569

Data Scientist

Data Warehouse Manager

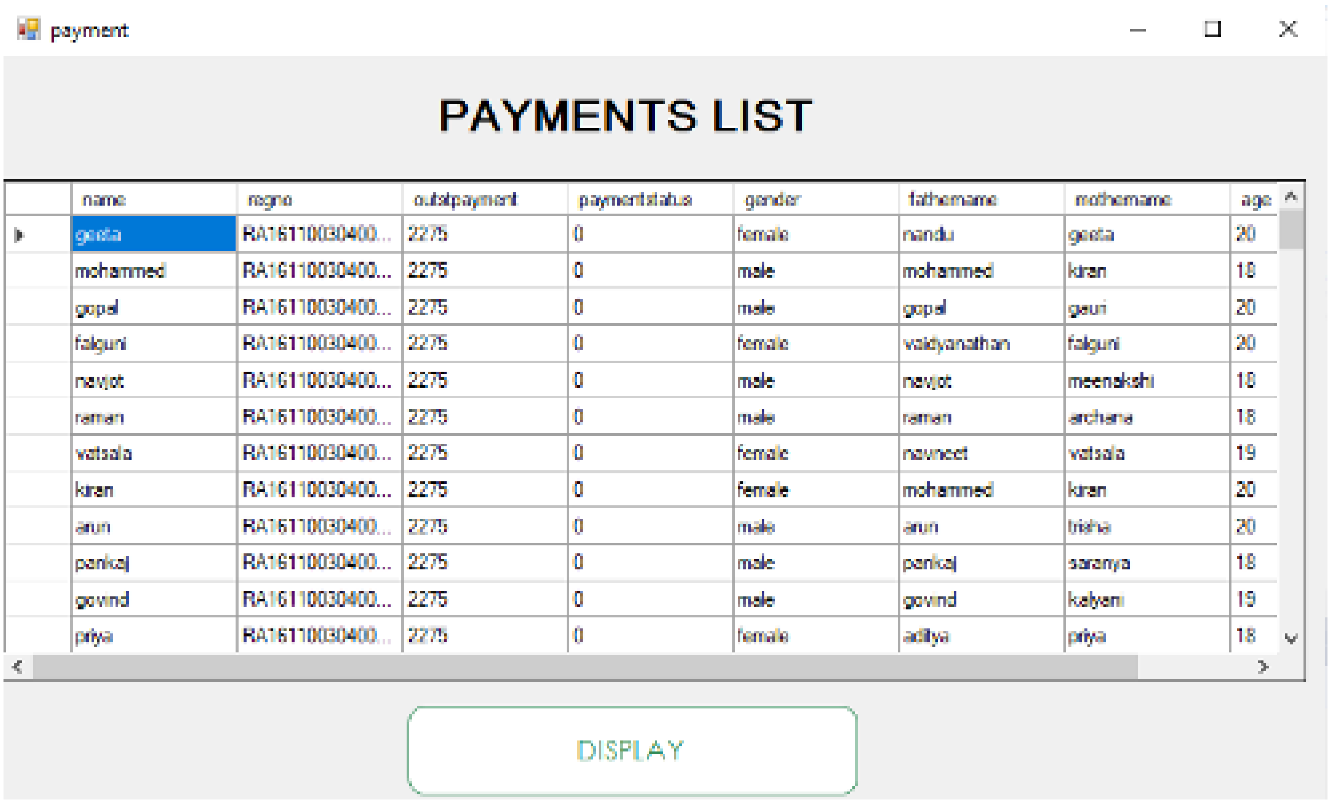
Network Engineer

Network Architect

Fig 3.6.14.1 :- The Career Helper Module

3.6.15 Payment Management Module

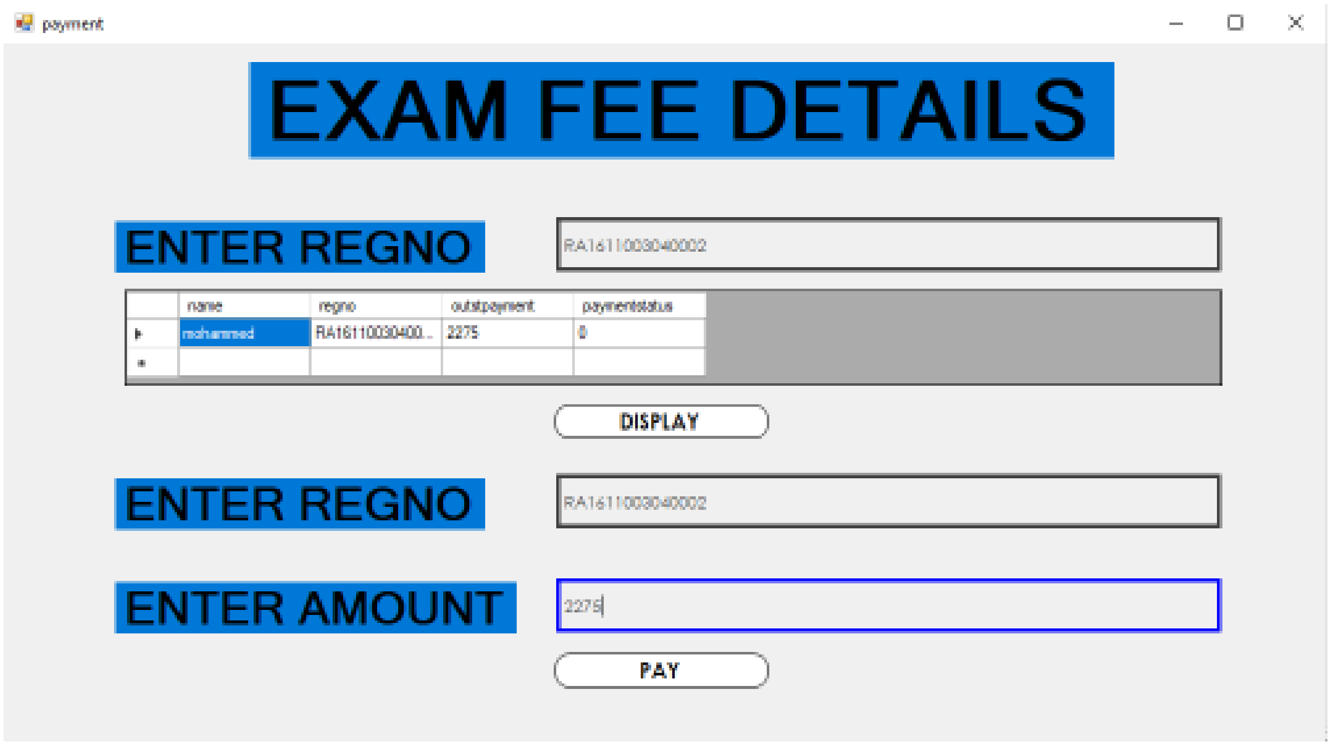
It is the module which is found in both the student’s part as well as in the teacher’s part as shown in the system architecture diagram. In the student’s part, the student can make the payment when his outstanding payment pending is fetched from the database and then accordingly the student can complete the payment. In the teacher’s part, the teacher can view the list of all the students who have made their payment as well as who have not made their payment.


payment

PAYMENTS LIST

	name	regno	outstpayment	paymentstatus	gender	fathername	mothername	age
▸	geeta	RA16110030400...	2275	0	female	nandu	geeta	20
	mohammed	RA16110030400...	2275	0	male	mohammed	kiran	18
	gopal	RA16110030400...	2275	0	male	gopal	gauri	20
	falguni	RA16110030400...	2275	0	female	vaidyanathan	falguni	20
	navjot	RA16110030400...	2275	0	male	navjot	meenakshi	18
	raman	RA16110030400...	2275	0	male	raman	archana	18
	vatsala	RA16110030400...	2275	0	female	navneet	vatsala	19
	kiran	RA16110030400...	2275	0	female	mohammed	kiran	20
	arun	RA16110030400...	2275	0	male	arun	trisha	20
	pankaj	RA16110030400...	2275	0	male	pankaj	saranya	18
	govind	RA16110030400...	2275	0	male	govind	kalyani	19
	priya	RA16110030400...	2275	0	female	aditya	priya	18

Fig 3.6.15.1 :- The Payment Management Module – teacher's part


payment

EXAM FEE DETAILS

ENTER REGNO: RA1611003040002

	name	regno	outstpayment	paymentstatus
▸	mohammed	RA16110030400...	2275	0
*				

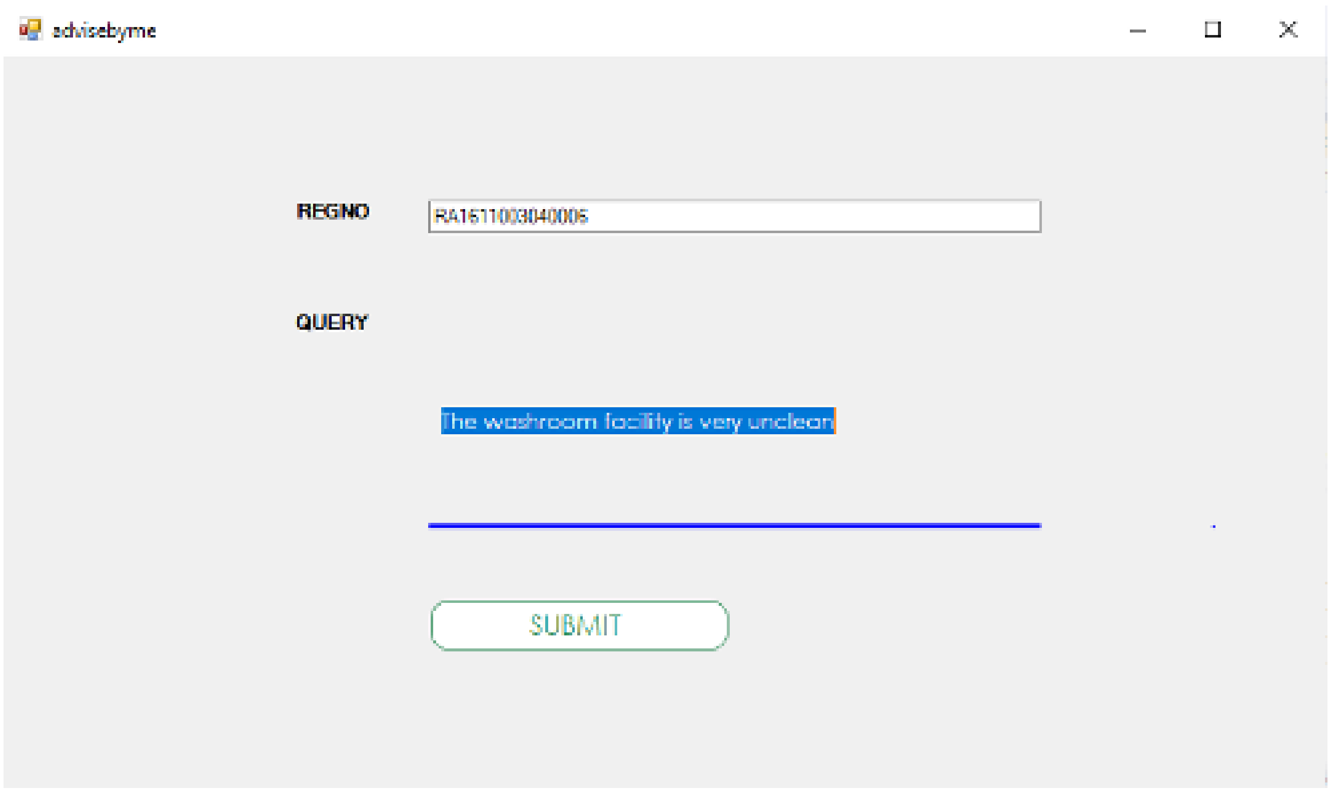
ENTER REGNO: RA1611003040002

ENTER AMOUNT: 2275

Fig 3.6.15.2 :- The Payment Management Module – student's part

3.6.16 Advice By Me Module

This module is the integral part of the feedback/query handling module in the student's part. It is the module in which the student can give some suggestions and advice if possible. Its aim is to record the suggestions given by student.



REGNO RA1611003040006

QUERY The washroom facility is very unclean

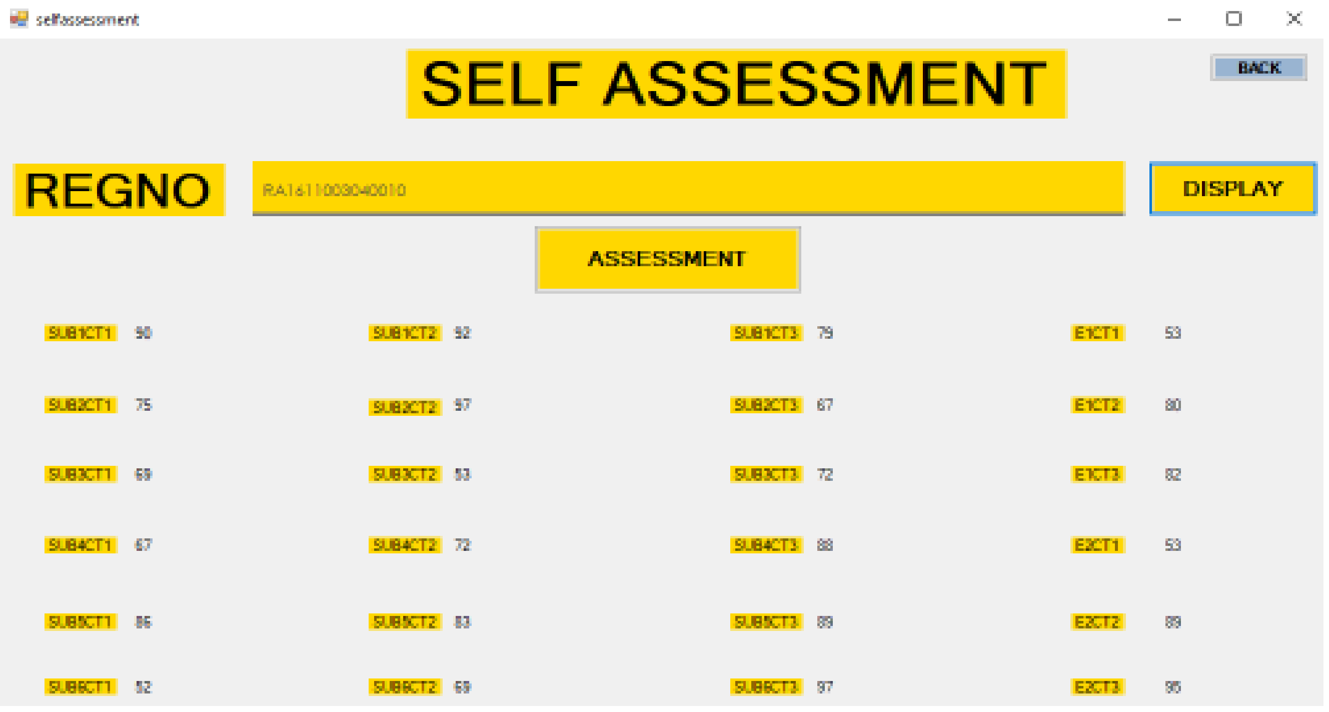
Fig 3.6.16.1 :- The Advice By Me Module – student's part

3.6.17 Self-Assessment Module

It is the module which is found in the student's part as well as in the teacher's part as shown in the system architecture diagram. In the case of self-assessment of the student the subjects are categorized in terms of the marks obtained by the student. In the case of self-assessment of the teacher there are various type of rating are given in order to evaluate the teaching performance of that teacher teaching that particular subject. The teaching performance is evaluated by considering various parameters like

Understandability of the concept, special teaching method used, ability to gain student attention, tendency to be impartial, ability to clear concept in simple manner, presentation of teacher etc and then two other kinds of rating are given i.e lower expectancy rating, higher expectancy rating etc. Lower expectancy rating tells you the rating based on the count of the students who have scored less marks.

Higher expectancy rating tells you the rating based on the count of the students who have scored more marks.



RA1611003040010

SUB1CT1	90	SUB1CT2	92	SUB1CT3	79	E1CT1	53
SUB2CT1	75	SUB2CT2	97	SUB2CT3	67	E1CT2	80
SUB3CT1	69	SUB3CT2	53	SUB3CT3	72	E1CT3	82
SUB4CT1	67	SUB4CT2	72	SUB4CT3	88	E2CT1	53
SUB5CT1	86	SUB5CT2	83	SUB5CT3	89	E2CT2	89
SUB6CT1	52	SUB6CT2	69	SUB6CT3	97	E2CT3	95

Fig 3.6.17.1 :- The Self Assessment (Homepage) – student's part

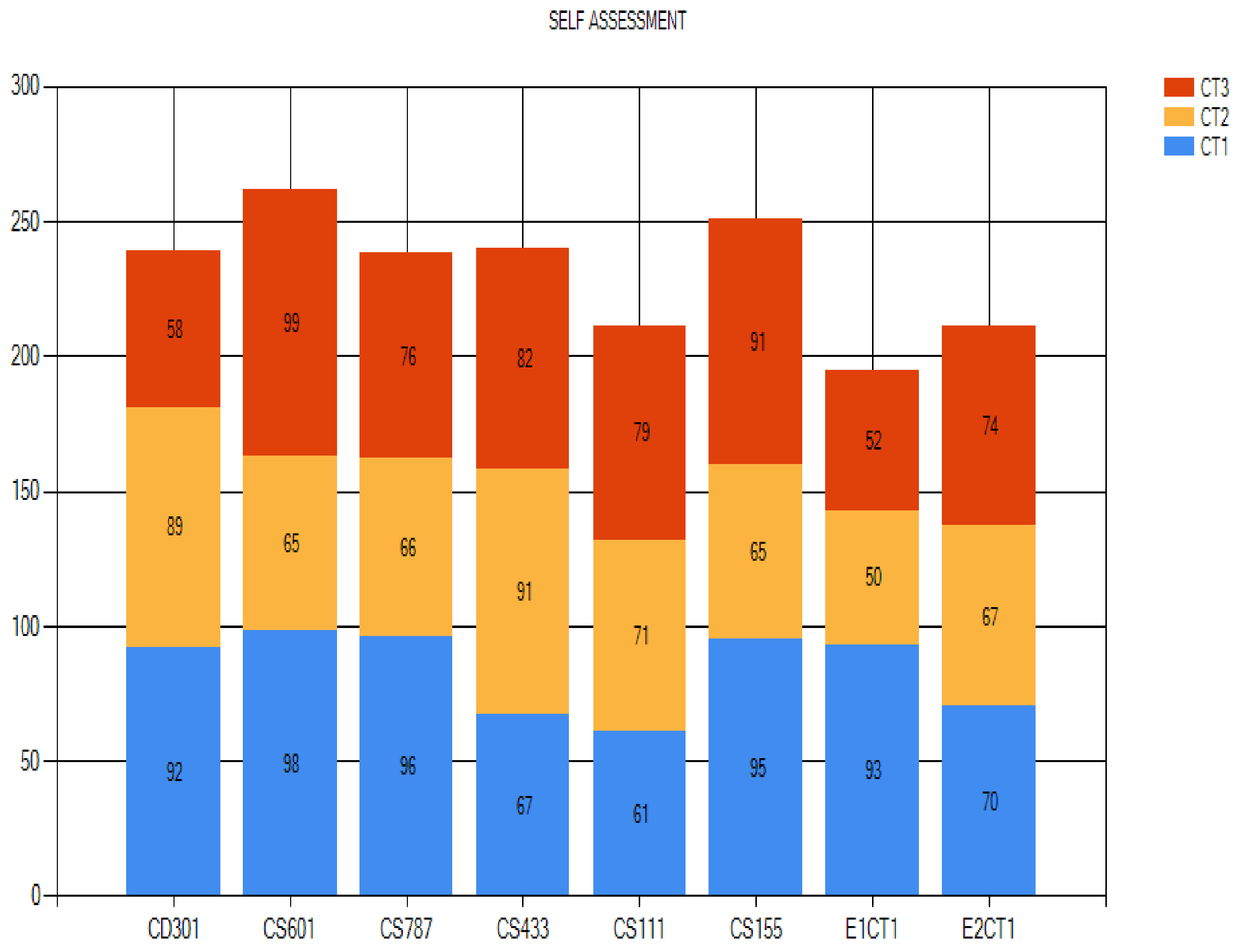


Fig 3.6.17.2 :- The Self Assessment – student's part

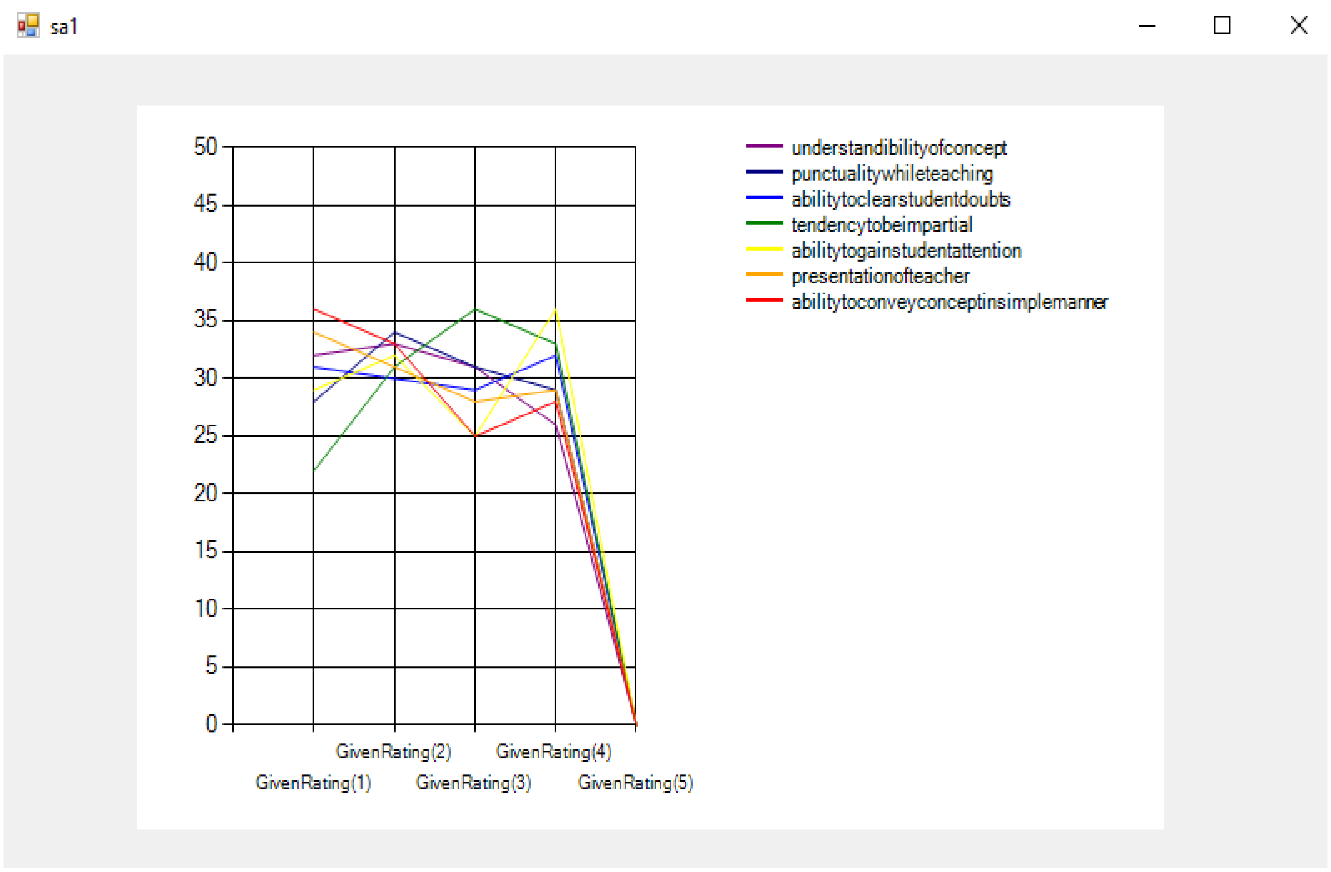
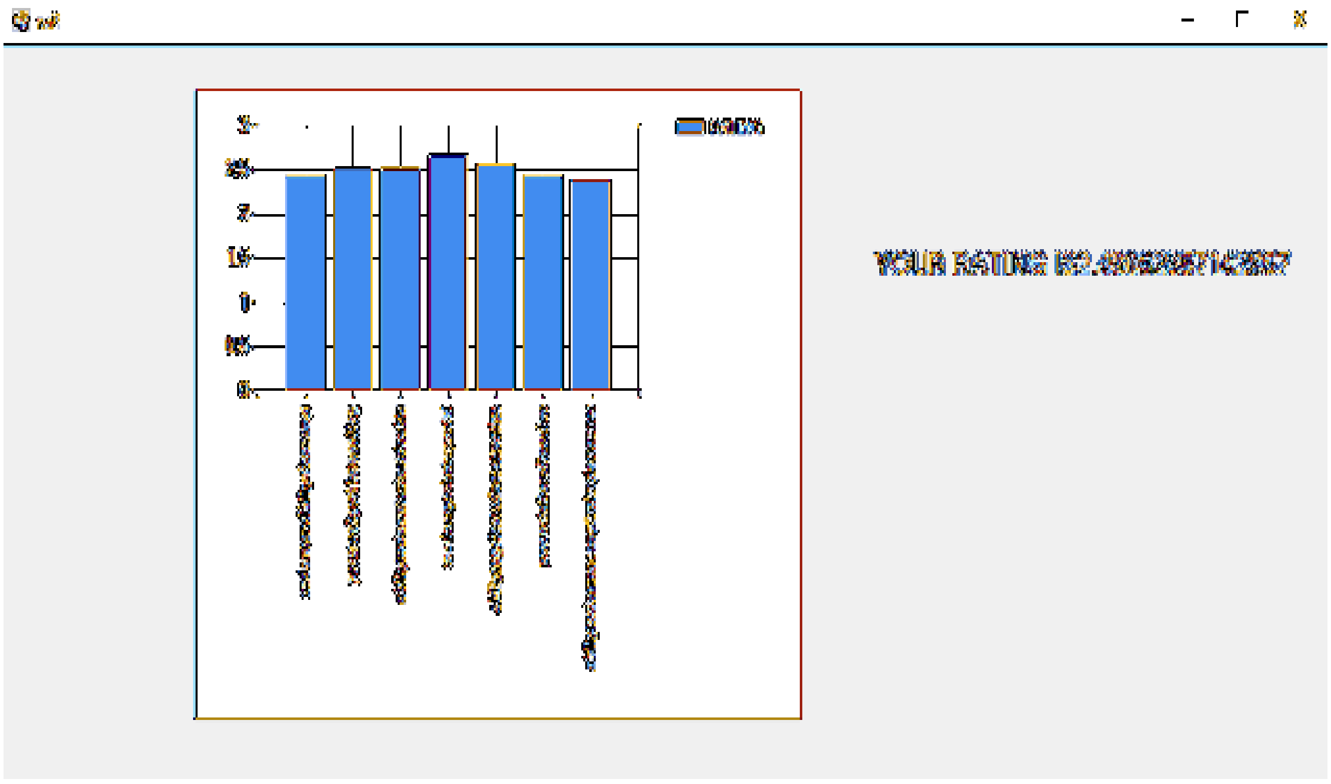
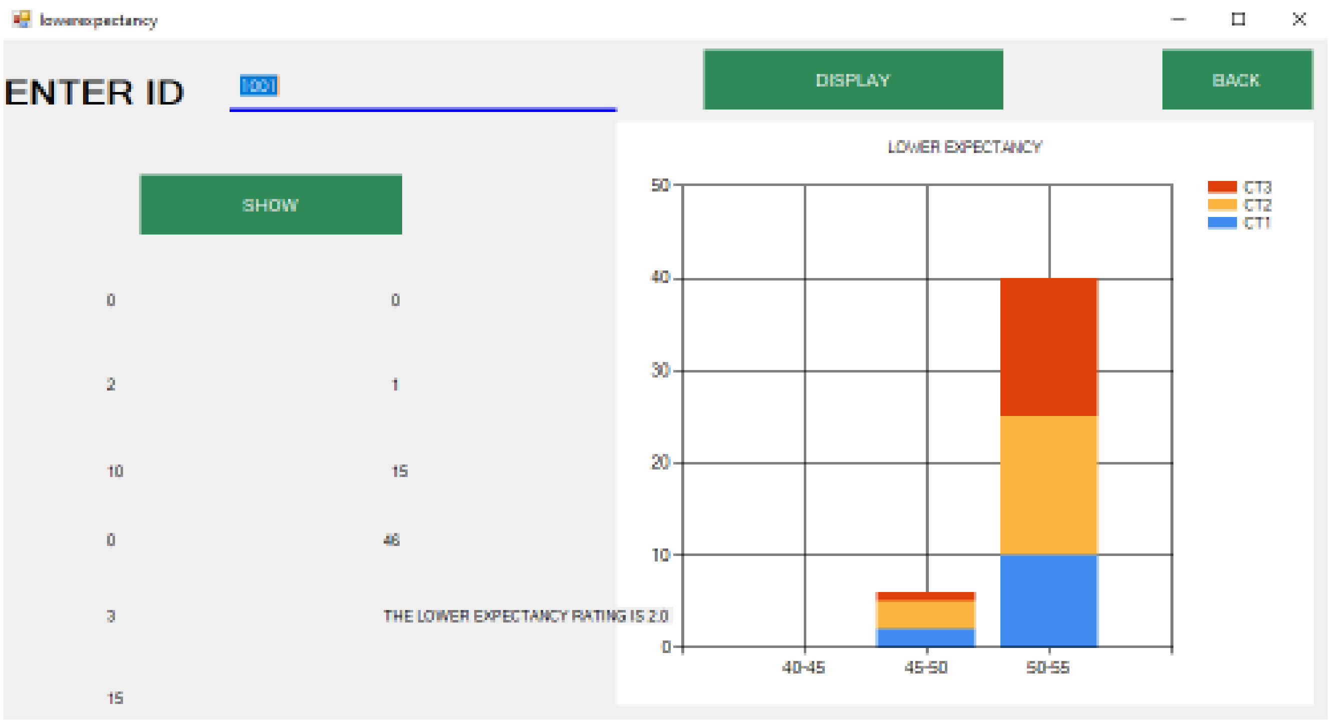


Fig 3.6.17.3 :- The Self Assessment – teacher's part



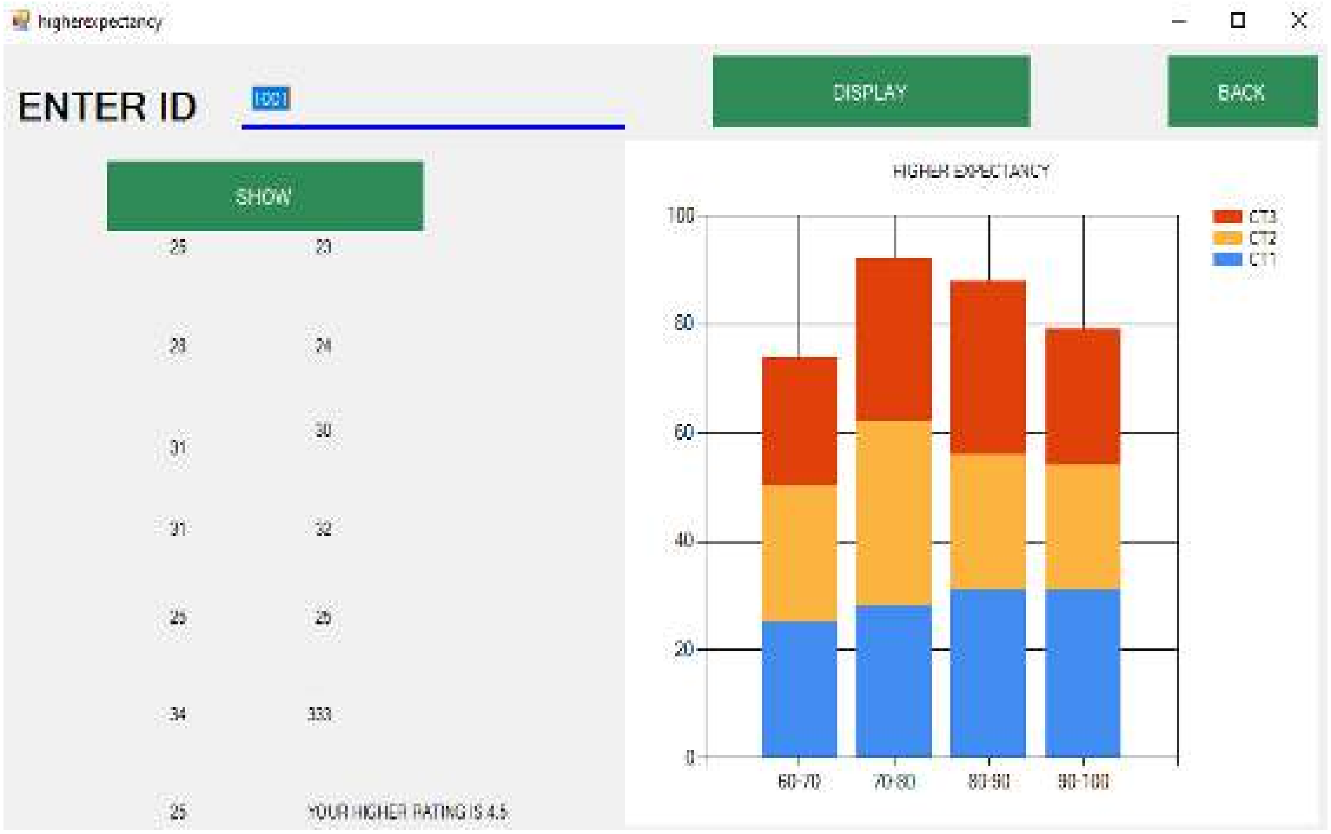
YOUR RATING IS 82.[illegible]

Fig 3.6.17.4 :- The Self Assessment – teacher's part



```
0          0
2          1
10         15
0          46
3          THE LOWER EXPECTANCY RATING IS 2.0
15
```

Fig 3.6.17.5 :- The Self Assessment (Lower Expectancy Rating) – teacher's part

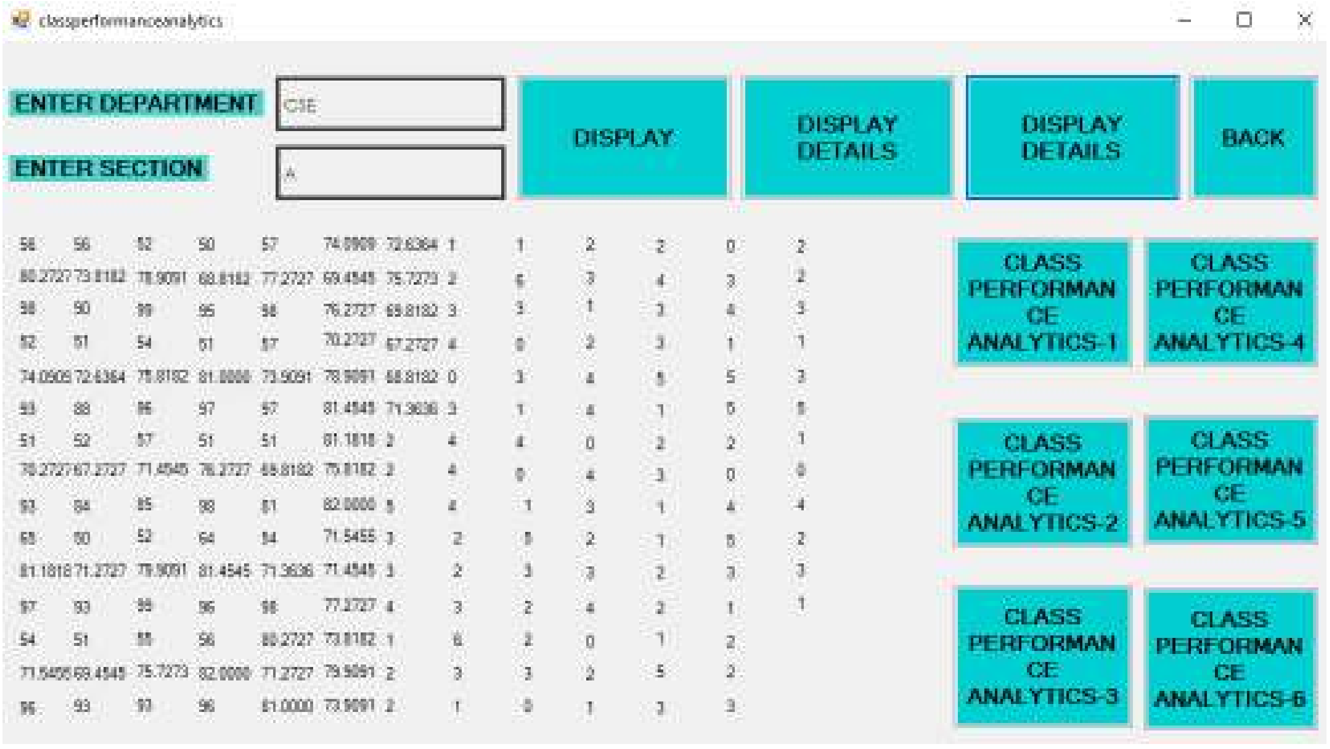


```
25         23
28         24
31         30
31         32
25         25
34         333
25         YOUR HIGHER RATING IS 4.5
```

Fig 3.6.17.6 :- The Self Assessment (Higher Expectancy Rating) – teacher's part

3.6.18 Class Performance Analytics Module

In this module it shows the performance analytics of the class in each subject in each of the respective examinations like cycle tests etc. It is slightly different from the self-assessment module of the student’s part such that in class performance analytics, performance of the class in each subject is considered whereas in the self-assessment module of the student’s part the performance of that particular student in each subject is considered.



classperformanceanalytics
ENTER DEPARTMENT
CSE
ENTER SECTION
A
DISPLAY
DISPLAY DETAILS
DISPLAY DETAILS
BACK
CLASS PERFORMANCE ANALYTICS-1
CLASS PERFORMANCE ANALYTICS-4
CLASS PERFORMANCE ANALYTICS-2
CLASS PERFORMANCE ANALYTICS-5
CLASS PERFORMANCE ANALYTICS-3
CLASS PERFORMANCE ANALYTICS-6

```
56      56      52      50      57      74.0909 72.6364 1       1       2       2       0       2
80.2727 73.8182 78.9091 68.8182 77.2727 69.4545 75.7273 2       6       3       4       3       2
98      90      99      95      98      76.2727 69.8182 3       3       1       3       4       3
82      81      54      61      87      70.2727 67.2727 4       0       2       3       1       1
74.0909 72.6364 78.8182 81.0000 73.9091 78.9091 68.8182 0       3       4       5       5       3
93      88      96      97      97      81.4545 71.3636 3       1       4       1       0       5
51      52      57      51      51      81.1818 2       4       4       0       2       2       1
70.2727 67.2727 71.4545 76.2727 69.8182 75.8182 2       4       0       4       3       0       0
93      94      85      98      81      82.0000 5       4       1       3       1       4       4
68      60      52      64      84      71.5455 3       2       5       2       1       5       2
81.1818 71.2727 78.9091 81.4545 71.3636 71.4545 3       2       3       3       2       3       3
97      93      99      96      98      77.2727 4       3       2       4       2       1       1
54      51      55      56      80.2727 73.8182 1       6       2       0       1       2
71.5455 69.4545 75.7273 82.0000 71.2727 79.9091 2       3       3       2       5       2
96      93      93      96      81.0000 73.9091 2       1       0       1       3       3
```

Fig 3.6.18.1 :- The Class Performance Analytics Module(Homepage) – teacher’s part

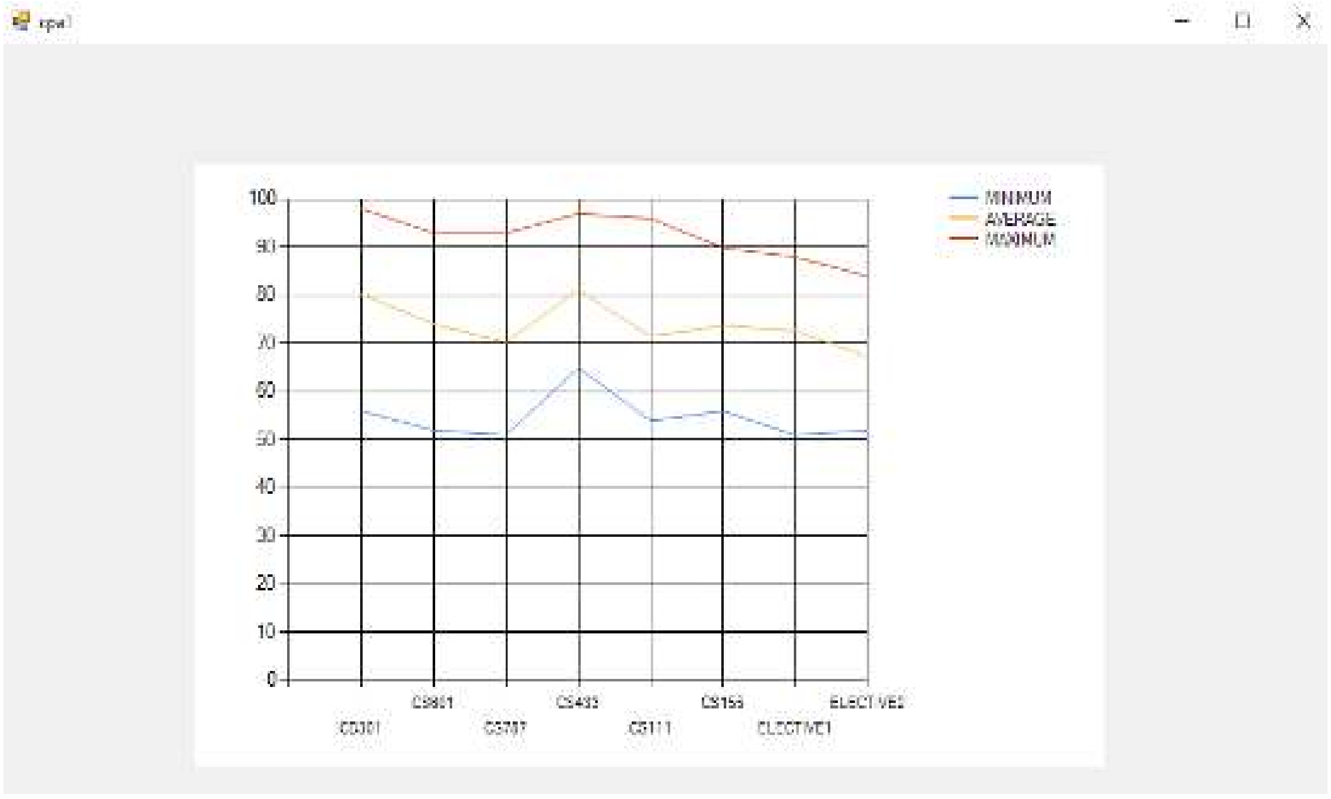


Fig 3.6.18.2 :- The Class Performance Analytics -1 – teacher's part

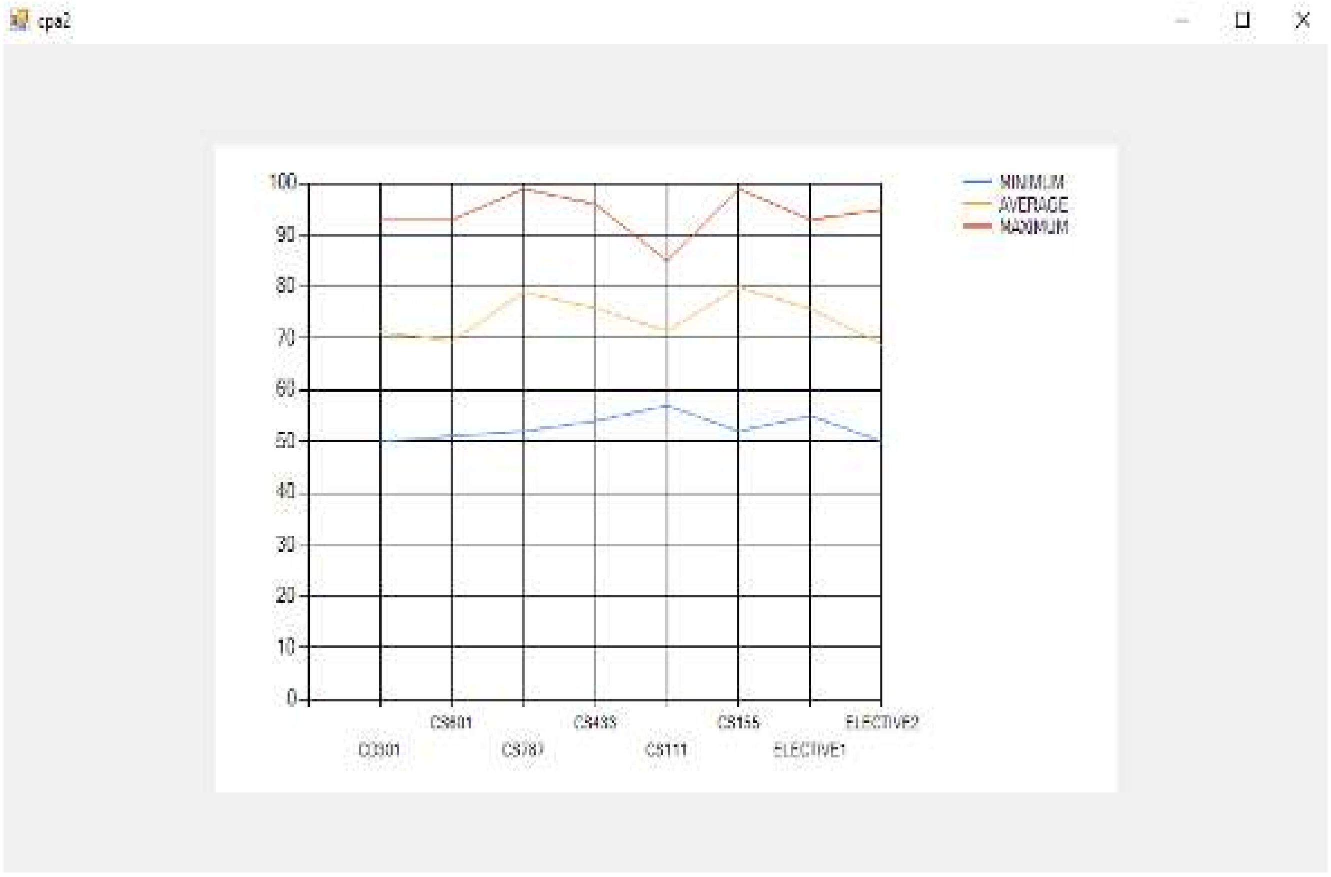


Fig 3.6.18.3 :- The Class Performance Analytics -2 – teacher's part

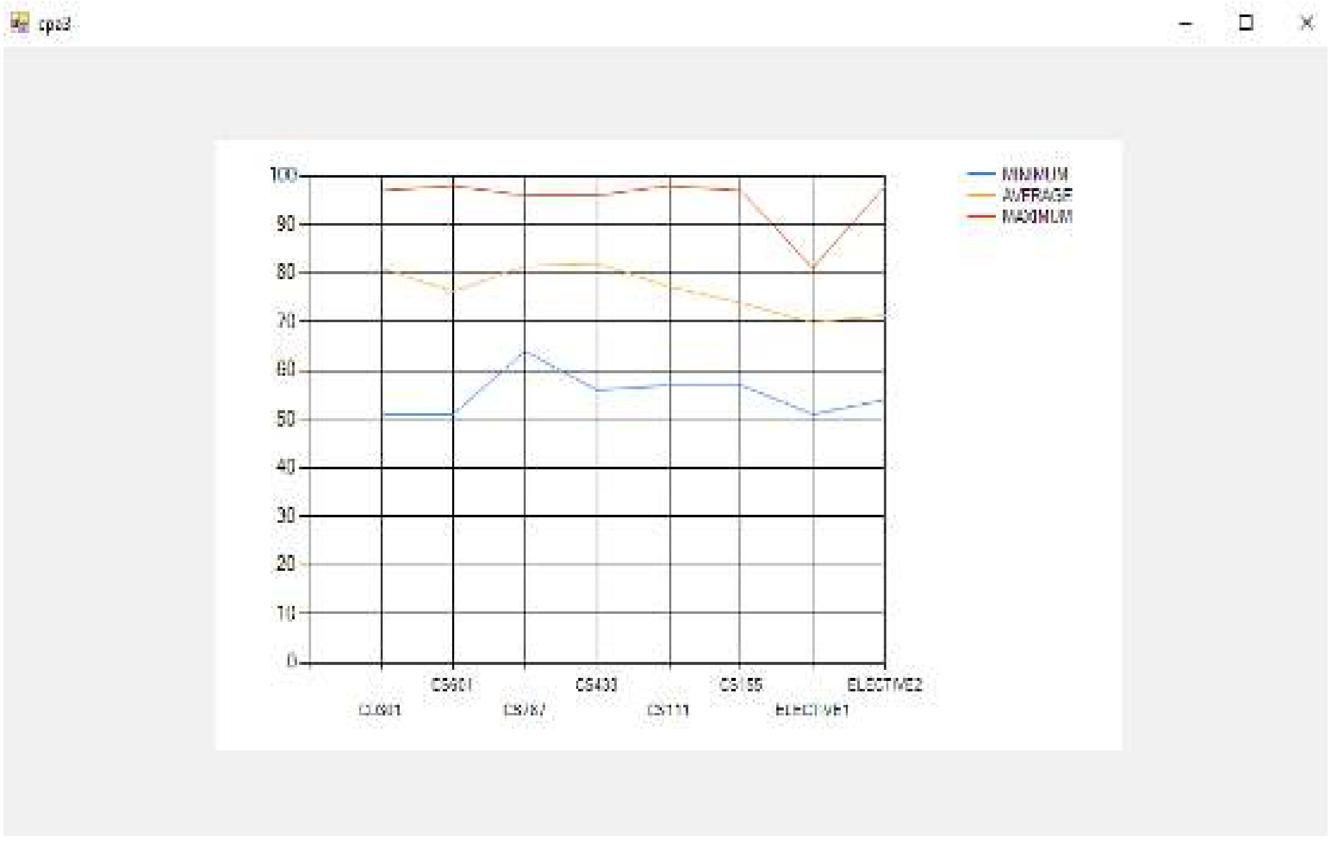


Fig 3.6.18.4 :- The Class Performance Analytics -3 – teacher's part

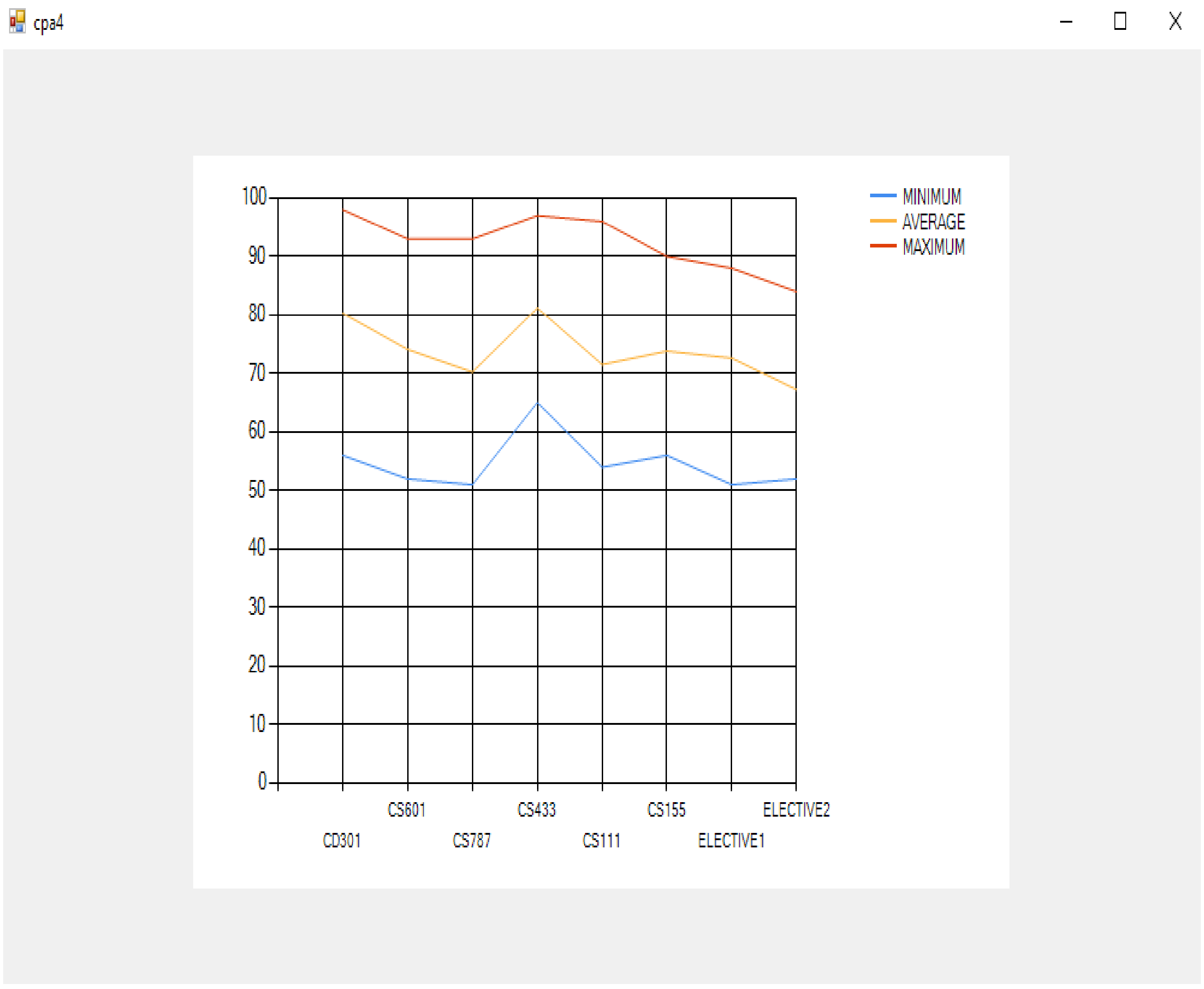


Fig 3.6.18.5 :- The Class Performance Analytics -4 – teacher's part

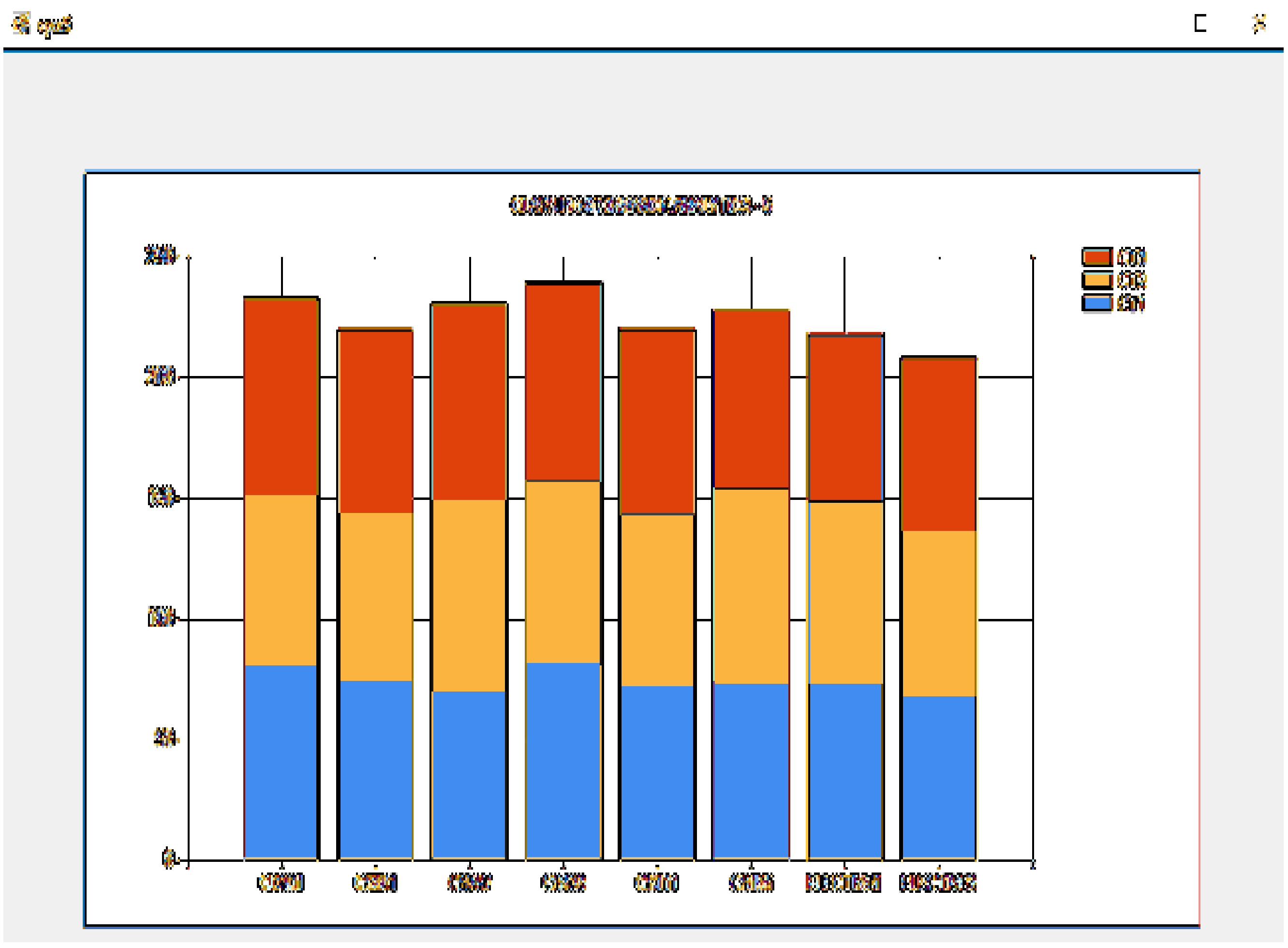


Fig 3.6.18.6 :- The Class Performance Analytics -5 – teacher’s part

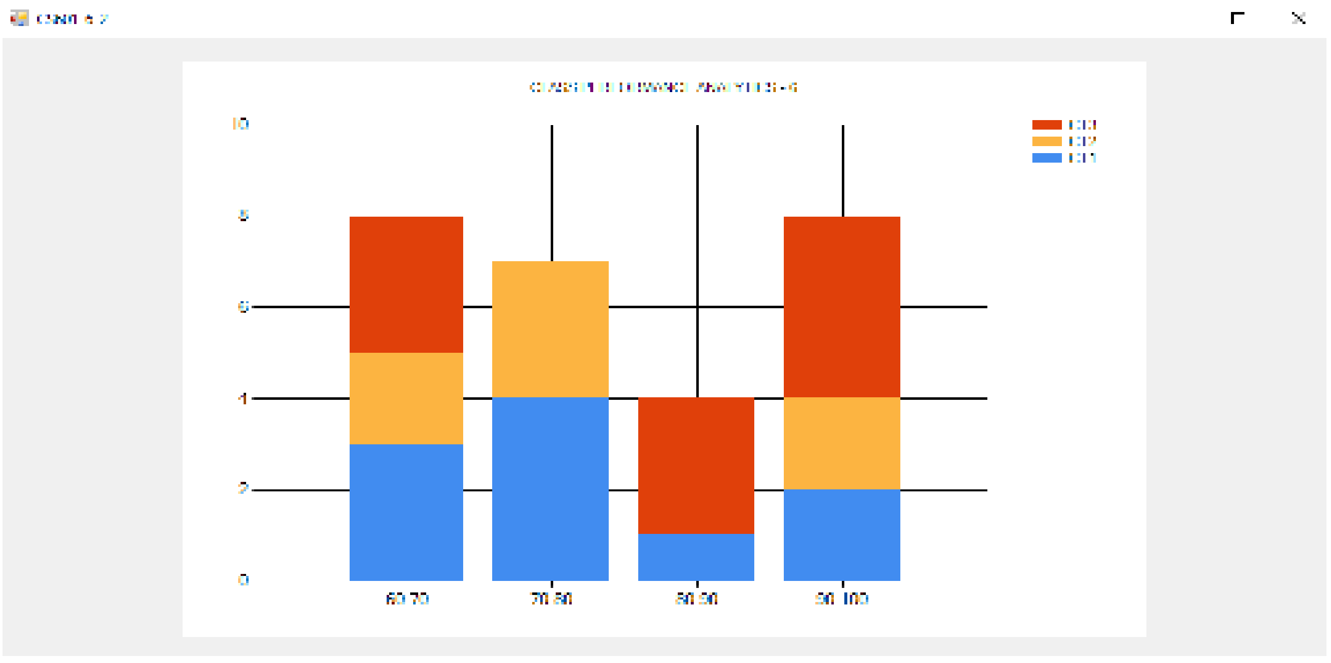


Fig 3.6.18.7 :- The Class Performance Analytics -6 – teacher’s part

3.6.19 Open Student Growth Analytics Module

In this module it tells you the complete performance growth analytics of each student in each subject in the respective cycle tests and other tests etc. The performance updates of students module is the integral part of this module.

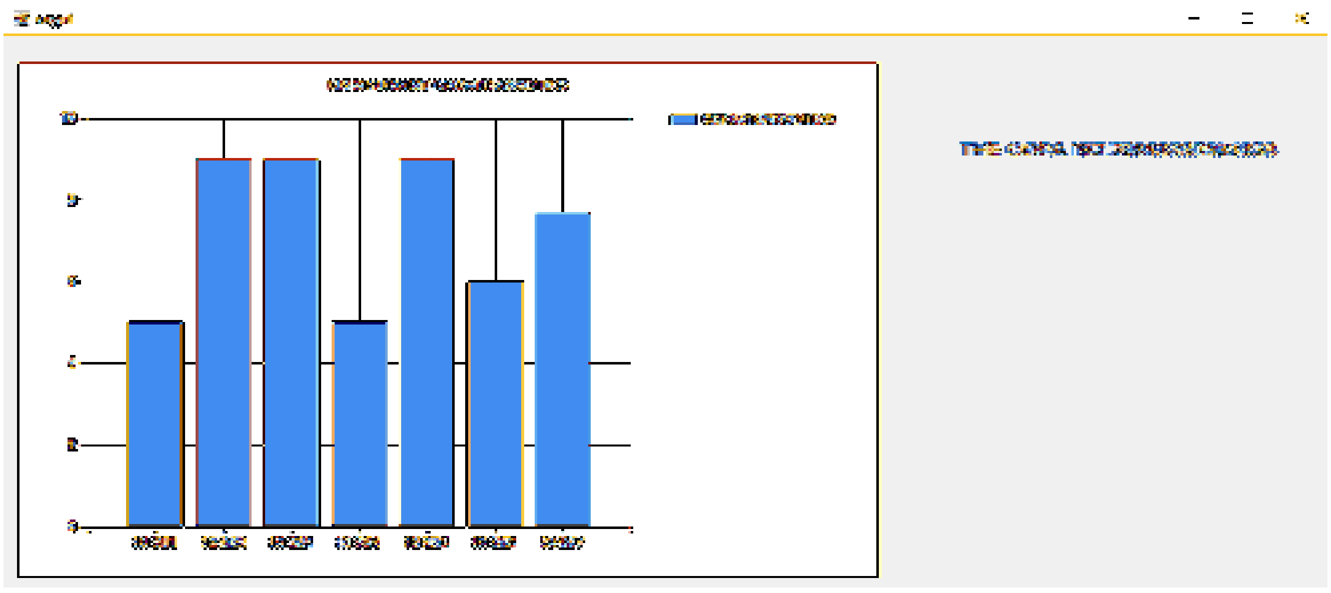
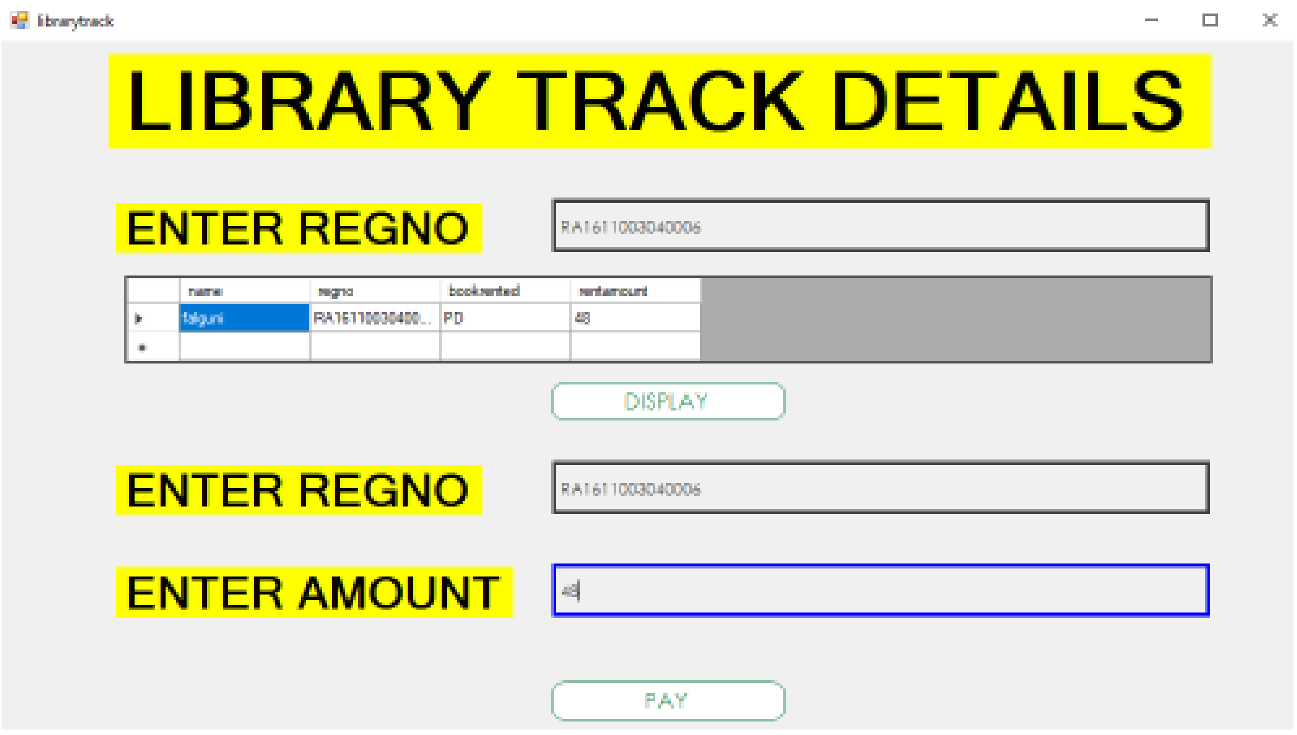


Fig 3.6.19.1 :- The Open Student Growth Analytics – teacher's part

3.6.20 Library Track System Module

It is the module in which it tells you the details of library book rented or issued from the university library. In this module it displays the book rented by the student and the library fee which is to be paid by the student for the book rented.


librarytrack

LIBRARY TRACK DETAILS

ENTER REGNO: RA1611003040006

	name	regno	bookrented	rentamount
▸	falguni	RA16110030400...	PD	48
*				

DISPLAY

ENTER REGNO: RA1611003040006

ENTER AMOUNT: 48

PAY

Fig 3.6.20.1 :- The Library Track System Module

3.7 Proposed Methodology

The teaching performance is evaluated by considering some of the parameters like understandability of the concept, punctuality while teaching, ability to clear student doubts, tendency to be impartial, ability to gain student attention, presentation of teacher, ability to clear concept in simple manner etc and some other parameters like overall subject performance, lower expectancy, higher expectancy. Furthermore, the algorithm calculates the rating of the teacher by doing the summation of all the parameters and multiply it with 5 for each parameter in order to show how many students have rated the teacher accordingly. And then for each parameter the average value is calculated by doing the summation of the values obtained for that particular parameter and then repeat this process for all the seven parameters and then add all the summation values obtained and then divide it by 7 and this is the value which tells the teacher’s rating as a whole. The lower expectancy is calculated by first, obtaining the number of the students who have scored less in their exams based on mark categories. And then do the summation of the number of students based on the number of categories. Higher expectancy is calculated by doing the summation of the number of students based on the number of categories. And then the lower expectancy value is compared with higher expectancy value and overall subject performance and accordingly the rating of the teaching performance is given. The following parameters are :-

UoC- Understandability of Concept.

PwT- Punctuality while Teaching

AtCSD- Ability to Clear Student Doubts

TtbI- Tendency to be Impartial

AtGSA- Ability to Gain Student Attention

PoT- Presentation of Teacher

AtCCiSM- Ability to Clear Concept in Simple Manner

CountCT1[40-45]- number of students who got marks between 40 and 45 in cycle test 1

CountCT2[40-45]- number of students who got marks between 40 and 45 in cycle test 2

CountCT3[40-45]- number of students who got marks between 40 and 45 in cycle test 3

CountCT1[45-50]- number of students who got marks between 45 and 50 in cycle test 1

CountCT2[45-50]- number of students who got marks between 45 and 50 in cycle test2

CountCT3[45-50]- number of students who got marks between 45 and 50 in cycle test3

CountCT1[50-55]- number of students who got marks between 50 and 55 in cycle test 1

CountCT2[50-55]- number of students who got marks between 50 and 55 in cycle test 2

CountCT3[50-55]- number of students who got marks between 50 and 55 in cycle test3

CountCT1[60-70]- number of students who got marks between 60 and 70 in cycle test 1

CountCT2[60-70]- number of students who got marks between 60 and 70 in cycle test 2

CountCT3[60-70]- number of students who got marks between 60 and 70 in cycle test3

CountCT1[70-80]- number of students who got marks between 70 and 80 in cycle test 1

CountCT2[70-80]- number of students who got marks between 70 and 80 in cycle test 2

CountCT3[70-80]- number of students who got marks between 70 and 80 in cycle test 3

CountCT1[80-90]- number of students who got marks between 80 and 90 in cycle test 1

CountCT2[80-90]- number of students who got marks between 80 and 90 in cycle test2

CountCT3[80-90]- number of students who got marks between 80 and 90 in cycle test3

CountCT1[90-100]- number of students who got marks between 90 and 100 in cycle test 1

CountCT2[90-100]- number of students who got marks between 90 and 100 in cycle test 2

CountCT3[90-100]- number of students who got marks between 90 and 100 in cycle test 3

Minimum[subjectCT1]- Minimum score obtained by the student in that subject in cycle test 1.

Average[subjectCT1]- score obtained by the student in that subject in cycle test 1 on an average.

Maximum[subjectCT1]- Maximum score obtained by the student in that subject in cycle test 1.

Minimum[subjectCT2]- Minimum score obtained by the student in that subject in cycle test 2.

Average[subjectCT2]- score obtained by the student in that subject in cycle test 2 on an average.
Maximum[subjectCT2]- Maximum score obtained by the student in that subject in cycle test 2.
Minimum[subjectCT3]- Minimum score obtained by the student in that subject in cycle test 3.
Average[subjectCT3]- score obtained by the student in that subject in cycle test 3 on an average.
Maximum[subjectCT3]- Maximum score obtained by the student in that subject in cycle test 3.

Overall Subject Performance(OSP)= Minimum[subjectCT1]+ Average[subjectCT1]+ Maximum[subjectCT1]+ Minimum[subjectCT2]+ Average[subjectCT2]+ Maximum[subjectCT2]+ Minimum[subjectCT3]+ Average[subjectCT3]+ Maximum[subjectCT3];

Lower Expectancy(LE)= CountCT1[40-45]+ CountCT1[45-50]+ CountCT1[50-55]+ CountCT2[40-45]+ CountCT2[45-50]+ CountCT2[50-55]+ CountCT3[40-45]+ CountCT3[45-50]+ CountCT3[50-55];

Higher Expectancy(HE)= CountCT1[60-70]+ CountCT1[70-80]+ CountCT1[80-90]+ CountCT1[90-100]+ CountCT2[60-70]+ CountCT2[70-80]+ CountCT2[80-90]+ CountCT2[90-100]+ CountCT3[60-70]+ CountCT3[70-80]+ CountCT3[80-90]+ CountCT3[90-100];
If value of LE is lesser than HE
The teaching performance rating is 4.0
If value of LE is lesser than HE but greater than the LE limit
The teaching performance rating is 3.0
If value of LE is greater than HE
The teaching performance rating is 2.0

CHAPTER 4

SYSTEM DESIGN

4.1 Hardware Requirements

- Intel Core i5 processor 7th Generation
- 1 TB Hard Disk
- 64 GB RAM

4.2 Software Requirements

- Microsoft Visual Studio
- C#
- phpMyAdmin MySQL Database

4.3 Microsoft Visual Studio

It is an unsegregated evolution domain from Microsoft. It is used to expand computer solutions, as well as websites, software, web-ministrations etc. It uses software establishment principles such as Windows API, Windows Forms, Windows Presentation, Windows-Foundation, Windows Store and Microsoft Silverlight etc. It can fabricate both autochthonous code and supervised code.

Visual Studio includes a program instructions editor supporting the code execution element and code refactoring. The desegregated debugger works both as a fount-echelon debugger and an instrument-stratum debugger. Other intrinsic instruments consists of a code hieroglyphics, artificer for assembling GUI routines, web originator, class creator, and database plan representation couturier. It receives plug-ins that magnifies the consequence at every extent—including appending support for spring command systems and adding new toolsets like editors and optical planners for domain-fixed languages or toolsets for other direction of the SDL.

4.4 C#

It is a universal-impetus, multi-archetype programming language circumscribe powerful typing, lexically purviewed, indispensable, problem solving capability, utilitarian, broad, object-oriented, and component-oriented programming field. The language, and implementations thereof, should stipulate underprop for software engineering tenets such as forceful type inspection, array obliged examination, probing of efforts to use uninitialized variables, and perfunctory garbage procurement. Software's evincing strength, permanent, and coder's productiveness are crucial. The language is planned for use in inventing software components fit for the purpose of implementation in sophisticated umbworlds. C# is intended to be applicable for writing softwares for both presenter and deep-rooted systems, ranging from the very large that use complicated operating systems, to the very small having devoted functions. Though these applications are meant to be economical with regard to the core and manipulating electricity requirements, the language was not meant to fight straight wise on assessment with assembly language.

4.5 phpMyAdmin MySQL Database

It is an unlimited and unbolted fount controlling tool for MySQL and MariaDB. As a conveyable web exercise programmed first and foremost in PHP, it has overtook one of the most in demand MySQL orchestrating instruments, especially for organizing different kinds of web services using PHP.
The various features of the software are :-

1. It can be connected to more than one computer or piece of equipment
2. It comprises of MySQL and MariaDB directory superintendence.
3. It is used for purporting particulars from CSV and SQL
4. It can disseminate data to various formats via the TCPDF bibliotheca.
5. It is used for regulating multitudinous hirelings
6. It can create compact indenture arrangement illustrations of the bibliographic formation.
7. We can create compounded scepticism using query-by-exemplar
8. You can discern comprehensively in an index or a subdivision of it
9. It can metamorphose stowed information into any configuration using a collection of pre-elucidated concomitants, like unveiling binary large object as picture or digitized connection.

10. It contains live histograms for surveiling MySQL server occupations like interrelatedness, procedures, reminiscence handling, etc.
11. It is designed for functioning with different kinds of systems program.
12. Make multiplex SQL queries facile.

CHAPTER 5

TESTING

5.1 Unit Testing

It is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing. There are some advantages of using unit testing, they are :-

- Unit testing increases the confidence in maintaining the code.
- Codes are more reusable.
- Development is faster, as you need not fire up the graphical user interface and provide all those inputs.
- The cost of fixing the defect is lesser as comparison to that of defects detected at higher levels.
- Debugging is easy.
- Codes are more reliable.

The unit testing is done for all the technical modules found in the project. A table is drawn to illustrate the unit testing done for each module :-

Module Name	Expected Output	Observation
Authentication Module	The user enters the username and password in order to verify themselves.	The password and the username is verified and the user proceeds further to the navigator menu.
Navigator Menu Module	It contains all the technical modules of that software.	You can access various services, functionalities and other facilities available in it. Note:- only in case of the student user and the teacher user.
Main Control Panel Module	It is the portal which can be accessed only by the top-level management officials. It basically does the business analytics for various purposes.	It contains the accessibility to the details of the students, teachers and their performance analytics.
Lesser Admin Panel Module	It is the portal which can be accessed by those officials who are responsible for managing the important functionalities of the institution.	It contains the academic details of each student in the institution.
Details Manager Module	It is used to manage the academic details of each student.	It first fetches the academic details of the student based on the register number. Then it updates the details if required.
Employee Manager Module	It is used to manage the personal details of all the employees/teachers and students	It first fetches the personal details of either the employee or the student based on their identification number. Then updating of the personal details is done if required.
IAM Panel Module	It is the panel which has access to all kinds of details of the institution, including the administration personnel.	It manages all the details of their employees including admin personnel and then can access the performance analytics details of all the students and teachers.
Business Analytics Module	It basically contains the information of the performance analytics of the	It contains the information based on various factors. This information is used for

	institution.	business purposes only, like marketing the institution etc.
Advice/Discussion Forum Module	It is controlled by the admin personnel for resolving the complaints and other queries of their employees.	The status is updated as 1 if a student query is resolved. Otherwise the status is updated as 2 if the employee/teacher's query is resolved. If the query is unresolved the status is 0 for both the employee and student.
Self-Assessment Module	● In case of the student it basically shows the assessment based on the rank calculation. ● In case of the teacher it basically evaluates the teaching performance of the teacher while teaching that subject.	● In case of the student, rank calculation is done by plotting the subjects in X-axis and the marks in Y-axis. ● In case of the teacher, three kinds of rating are given like overall teacher's rating, lower expectancy rating, higher expectancy rating.
Digital Notice Board Module	It shows you the details of various events taking place in the institution.	● In case of the student it just shows you the list of events happening in the institution. ● In case of the teacher, it gives them right to update the notice board based on occurrence of new event.
Payment Management Module	It shows the outstanding payment information of each student.	● In case of the student, it first fetches the payment details based on the register number of the student. Then the payment status is updated as 1 if the student has done payment, otherwise

		the payment status is 0. ● In case of the teacher, it just shows them the outstanding payment details of all the students.
Performance Updates of Students Module	It shows the subject wise performance analytics of each student.	● In case of the student, it just shows the subject wise performance of him/her in their respective exams. ● In case of the teacher, they have the right to update the marks of the students if required.
Self-Profile Module	It just shows their personal details like name, department, college etc.	It first fetches the personal details of the user based on their username entered in the authentication module.
Feedback/Query Handling Module	It lets the user to comment on the performance of the institution by giving feedback. It is managed by the Advice/Discussion Forum Module.	● It lets the user(student or employee/teacher) to enter the complaint or other queries in order for it to be resolved. ● Here as it is connected to the Advice/Discussion Forum Module the status is updated accordingly.
Career Helper Module	It helps the student by giving them an idea to pursue various career paths.	It first fetches the elective subjects chosen by the student, based on the entered register number. It then maps these subjects and then provides you various possibilities for pursuing their career.
Library Track System	It shows you the library	It first fetches the library

Module	details of that student, like book rented, rent amount etc.	details of that student based on their register number. If there is rent amount that needs to be paid, then the payment status is updated as 1 for paid and 0 for unpaid.
Class Performance Analytics Module	It tells you the performance analytics of the class on each subject in their respective exams.	It does various kinds of analytics to illustrate the performance analytics of the class in better and meaningful manner.
Open Student Growth Analytics Module	It tells you the complete performance analytics of a particular student throughout their academic career.	The Performance updates of Students Module is an integral part of this module.

Table 5.1 :- Unit Testing of each Module

5.2 Integration Testing

As we see in this project, the one place where all the technical modules are integrated into one, is the navigator menu. It is the place where it encapsulates all the technical modules of the software application (namely student and teacher). It is found in two out of the three softwares (basically software application for student and software application of teacher). In case of the student, the navigator menu contains modules like library track system, career helper, performance updates of students, self-assessment and so on, which is basically suitable for the student user to handle. In case of teacher, the navigator menu contains modules like class performance analytics, open student growth analytics, self-assessment, digital notice board and so on, which is basically suitable for the teacher user to handle. The only difference in the two navigator menus is the type of information conveyed to the user. The similarities between the two navigator menus are – the self-profile of the user is obtained based on the username entered during the authentication; the feedback/query handling of the user has the same functionality for both the users. There are some modules which are present in both the navigator menus same in name but different in functionality like payment management module, self-assessment module etc.

CHAPTER 6

Conclusion and Future Scope

By this project I am trying to enhance this system in order to heighten and implement the improvements of the performance in teaching as well as improvements in exam performance. Presently my system is going to be used to be used as a good management system in various colleges and universities. I am aiming to represent the performance analytics and other assessments in an even more meaningful manner so that users those who are using the system find it easy to use and will be able to contribute to the growth and other improvements in the field of education. The project can be further made more exclusive by including the departmental analytics in such a way that this data will be helpful in use by the university for various marketing strategies and other business purposes. I will be including access privileges part in the IAM panel where the user once logged in cannot access information of other people. This system will be even more dynamic and robust and at the same time the security protocols and procedures installed will be able to provide data integrity to the user. As it can be accessed in any remote location it will be designed to work on any kind of different conditions. It will be created in such a way that it will be used by any kind of educational institution in order to evaluate the quality and quantity of the education conveyed by the institution. Different kinds of analytics will also be added in order to evaluate yourself and let it be the teaching performance or performance of the university in the particular quarter of the financial year. More parameters will be included to evaluate the teaching performance of the teacher. There will be a provision for entering feedback for measuring the quality of each and every service provided by the institution.

APPENDIX A

PROJECTCODE

A.1 For Self-Assessment of student

A.1.1 selfassessment.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Sql;
using MySql.Data.MySqlClient;

namespace WindowsFormsApp12
{
  public partial class selfassessment : Form
  {
    public selfassessment()
    {
      InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
      studentlogin sl = new studentlogin();
      this.Hide();
      sl.Show();
    }


    private void button2_Click(object sender, EventArgs e)
    {
      MySqlConnection con17 = new
MySqlConnection("Server=db4free.net;Port=3306;Database=srikrishna;Username=srikrish;password=
987654321;old guids=true");
      MySqlCommand cmd17 = new MySqlCommand("select
CD301CT1,CS601CT1,CS787CT1,CS433CT1,CS111CT1,CS155CT1,CD301CT2,CS601CT2,CS787
CT2,CS433CT2,CS111CT2,CS155CT2,CD301CT3,CS601CT3,CS787CT3,CS433CT3,CS111CT3,C
S155CT3,E1CT1,E1CT2,E1CT3,E2CT1,E2CT2,E2CT3 from studentdetails where regno='" +
bunifuMaterialTextbox1.Text + "'", con17);
      con17.Open();
      MySqlDataReader read17 = cmd17.ExecuteReader();
      if(read17.Read())
      {
```

```
                label19.Text = read17[0].ToString();
                label20.Text = read17[1].ToString();
                label21.Text = read17[2].ToString();
                label22.Text = read17[3].ToString();
                label23.Text = read17[4].ToString();
                label24.Text = read17[5].ToString();
                label31.Text = read17[6].ToString();
                label32.Text = read17[7].ToString();
                label33.Text = read17[8].ToString();
                label34.Text = read17[9].ToString();
                label35.Text = read17[10].ToString();
                label36.Text = read17[11].ToString();
                label43.Text = read17[12].ToString();
                label44.Text = read17[13].ToString();
                label45.Text = read17[14].ToString();
                label46.Text = read17[15].ToString();
                label47.Text = read17[16].ToString();
                label48.Text = read17[17].ToString();
                label55.Text = read17[18].ToString();
                label56.Text = read17[19].ToString();
                label57.Text = read17[20].ToString();
                label58.Text = read17[21].ToString();
                label59.Text = read17[22].ToString();
                label60.Text = read17[23].ToString();

            }
            con17.Close();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            assessment ca1 = new
assessment(label19.Text,label20.Text,label21.Text,label22.Text,label23.Text,label24.Text,label31.Tex
t,label32.Text,label33.Text,label34.Text,label35.Text,label36.Text,label43.Text,label44.Text,label45.T
ext,label46.Text,label47.Text,label48.Text,label55.Text,label56.Text,label57.Text,label58.Text,label5
9.Text,label60.Text);
            this.Hide();
            ca1.Show();
        }
    }
}
```

A.1.2 assessment.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace WindowsFormsApp12
{

    public partial class assessment : Form
    {
        string a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, a16, a17, a18, a19, a20, a21,
a22, a23, a24;
        public assessment(string s1,string s2,string s3, string s4, string s5, string s6, string s7,string
s8,string s9,string s10, string s11, string s12, string s13, string s14, string s15, string s16, string s17,
string s18, string s19, string s20, string s21, string s22,string s23, string s24)
        {
            InitializeComponent();
            a1 = s1;
            a2 = s2;
            a3 = s3;
            a4 = s4;
            a5 = s5;
            a6 = s6;
            a7 = s7;
            a8 = s8;
            a9 = s9;
            a10 = s10;
            a11 = s11;
            a12 = s12;
            a13 = s13;
            a14 = s14;
            a15 = s15;
            a16 = s16;
            a17 = s17;
            a18 = s18;
            a19 = s19;
            a20 = s20;
            a21 = s21;
            a22 = s22;
            a23 = s23;
            a24 = s24;
            chart1.Titles.Add("SELF ASSESSMENT");
            chart1.Series["CT1"].Points.AddXY("CD301", a1);
            chart1.Series["CT2"].Points.AddXY("CD301", a7);
            chart1.Series["CT3"].Points.AddXY("CD301", a13);

            chart1.Series["CT1"].Points.AddXY("CS601", a2);
            chart1.Series["CT2"].Points.AddXY("CS601", a8);
            chart1.Series["CT3"].Points.AddXY("CS601", a14);

            chart1.Series["CT1"].Points.AddXY("CS787", a3);
            chart1.Series["CT2"].Points.AddXY("CS787", a9);
            chart1.Series["CT3"].Points.AddXY("CS787", a15);

            chart1.Series["CT1"].Points.AddXY("CS433", a4);
            chart1.Series["CT2"].Points.AddXY("CS433", a10);
            chart1.Series["CT3"].Points.AddXY("CS433", a16);

            chart1.Series["CT1"].Points.AddXY("CS111", a5);
            chart1.Series["CT2"].Points.AddXY("CS111", a11);
            chart1.Series["CT3"].Points.AddXY("CS111", a17);

            chart1.Series["CT1"].Points.AddXY("CS155", a6);
            chart1.Series["CT2"].Points.AddXY("CS155", a12);
            chart1.Series["CT3"].Points.AddXY("CS155", a18);

            chart1.Series["CT1"].Points.AddXY("E1CT1", a19);
            chart1.Series["CT2"].Points.AddXY("E1CT2", a20);
            chart1.Series["CT3"].Points.AddXY("E1CT3", a21);

            chart1.Series["CT1"].Points.AddXY("E2CT1", a22);
            chart1.Series["CT2"].Points.AddXY("E2CT2", a23);
            chart1.Series["CT3"].Points.AddXY("E2CT2", a24);

            chart1.Series["CT1"].IsValueShownAsLabel = true;
            chart1.Series["CT2"].IsValueShownAsLabel = true;
            chart1.Series["CT3"].IsValueShownAsLabel = true;
        }
    }
}
```

A.2 For Self-Assessment of Teacher

A.2.1 selfassessment.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Sql;
using MySql.Data.MySqlClient;

namespace WindowsFormsApp11
{
    public partial class selfassessment : Form
    {
        public selfassessment()
        {
            InitializeComponent();

        }
        private void button2_Click(object sender, EventArgs e)
        {
            teacherrating tr = new teacherrating();
            this.Hide();
            tr.Show();

        }

        private void button1_Click(object sender, EventArgs e)
        {
            teacherlogin tl = new teacherlogin();
            this.Hide();
            tl.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            subjectperformance sp = new subjectperformance();
            this.Hide();
            sp.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            lowerexpectancy le = new lowerexpectancy();
            this.Hide();
            le.Show();
        }

        private void button5_Click(object sender, EventArgs e)
        {
              higherexpectancy he = new higherexpectancy();
              this.Hide();
              he.Show();
          }
      }
  }
```

A.2.2 teacherrating.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Sql;
using MySql.Data.MySqlClient;

namespace WindowsFormsApp11
{
    public partial class teacherrating : Form
    {
        public teacherrating()
        {
            InitializeComponent();

        }

        private void bunifuThinButton21_Click_1(object sender, EventArgs e)
        {
            if (bunifuMaterialTextbox1.Text == "1001")
            {

                MySqlConnection con21 = new MySqlConnection("Server = db4free.net; Port = 3306;
Database = srikrishna; Username = srikrish; password = 987654321; old guids = true");
                MySqlCommand cmd21 = new MySqlCommand("select
count(sub1understandibilityofconcept) from studentdetails where sub1understandibilityofconcept='" +
1 + "'", con21);
                MySqlCommand cmd22 = new MySqlCommand("select
count(sub1understandibilityofconcept) from studentdetails where sub1understandibilityofconcept='" +
2 + "'", con21);
                MySqlCommand cmd23 = new MySqlCommand("select
count(sub1understandibilityofconcept) from studentdetails where sub1understandibilityofconcept='" +
3 + "'", con21);
                MySqlCommand cmd24 = new MySqlCommand("select
count(sub1understandibilityofconcept) from studentdetails where sub1understandibilityofconcept='" +
4 + "'", con21);
                MySqlCommand cmd25 = new MySqlCommand("select
count(sub1understandibilityofconcept) from studentdetails where sub1understandibilityofconcept='" +
5 + "'", con21);
                MySqlCommand cmd26 = new MySqlCommand("select
count(sub1punctualitywhileteaching) from studentdetails where sub1punctualitywhileteaching='" + 1 +
"'", con21);
                MySqlCommand cmd27 = new MySqlCommand("select
count(sub1punctualitywhileteaching) from studentdetails where sub1punctualitywhileteaching='" + 2 +
"'", con21);
                MySqlCommand cmd28 = new MySqlCommand("select
count(sub1punctualitywhileteaching) from studentdetails where sub1punctualitywhileteaching='" + 3 +
"'", con21);
            MySqlCommand cmd29 = new MySqlCommand("select
count(sub1punctualitywhileteaching) from studentdetails where sub1punctualitywhileteaching='" + 4 +
"'", con21);
            MySqlCommand cmd30 = new MySqlCommand("select
count(sub1punctualitywhileteaching) from studentdetails where sub1punctualitywhileteaching='" + 5 +
"'", con21);
            MySqlCommand cmd31 = new MySqlCommand("select
count(sub1abilitytoclearstudentdoubts) from studentdetails where sub1abilitytoclearstudentdoubts='" +
1 + "'", con21);
            MySqlCommand cmd32 = new MySqlCommand("select
count(sub1abilitytoclearstudentdoubts) from studentdetails where sub1abilitytoclearstudentdoubts='" +
2 + "'", con21);
            MySqlCommand cmd33 = new MySqlCommand("select
count(sub1abilitytoclearstudentdoubts) from studentdetails where sub1abilitytoclearstudentdoubts='" +
3 + "'", con21);
            MySqlCommand cmd34 = new MySqlCommand("select
count(sub1abilitytoclearstudentdoubts) from studentdetails where sub1abilitytoclearstudentdoubts='" +
4 + "'", con21);
            MySqlCommand cmd35 = new MySqlCommand("select
count(sub1abilitytoclearstudentdoubts) from studentdetails where sub1abilitytoclearstudentdoubts='" +
5 + "'", con21);
            MySqlCommand cmd36 = new MySqlCommand("select count(sub1tendencytobeimpartial)
from studentdetails where sub1tendencytobeimpartial='" + 1 + "'", con21);
            MySqlCommand cmd37 = new MySqlCommand("select count(sub1tendencytobeimpartial)
from studentdetails where sub1tendencytobeimpartial='" + 2 + "'", con21);
            MySqlCommand cmd38 = new MySqlCommand("select count(sub1tendencytobeimpartial)
from studentdetails where sub1tendencytobeimpartial='" + 3 + "'", con21);
            MySqlCommand cmd39 = new MySqlCommand("select count(sub1tendencytobeimpartial)
from studentdetails where sub1tendencytobeimpartial='" + 4 + "'", con21);
            MySqlCommand cmd40 = new MySqlCommand("select count(sub1tendencytobeimpartial)
from studentdetails where sub1tendencytobeimpartial='" + 5 + "'", con21);
            MySqlCommand cmd41 = new MySqlCommand("select
count(sub1abilitytogainstudentattention) from studentdetails where
sub1abilitytogainstudentattention='" + 1 + "'", con21);
            MySqlCommand cmd42 = new MySqlCommand("select
count(sub1abilitytogainstudentattention) from studentdetails where
```

```
sub1abilitytogainstudentattention='" + 2 + "'", con21);
            MySqlCommand cmd43 = new MySqlCommand("select
count(sub1abilitytogainstudentattention) from studentdetails where
sub1abilitytogainstudentattention='" + 3 + "'", con21);
            MySqlCommand cmd44 = new MySqlCommand("select
count(sub1abilitytogainstudentattention) from studentdetails where
sub1abilitytogainstudentattention='" + 4 + "'", con21);
            MySqlCommand cmd45 = new MySqlCommand("select
count(sub1abilitytogainstudentattention) from studentdetails where
sub1abilitytogainstudentattention='" + 5 + "'", con21);
            MySqlCommand cmd46 = new MySqlCommand("select count(sub1presentationofteacher)
from studentdetails where sub1presentationofteacher='" + 1 + "'", con21);
            MySqlCommand cmd47 = new MySqlCommand("select count(sub1presentationofteacher)
from studentdetails where sub1presentationofteacher='" + 2 + "'", con21);
            MySqlCommand cmd48 = new MySqlCommand("select count(sub1presentationofteacher)
from studentdetails where sub1presentationofteacher='" + 3 + "'", con21);
            MySqlCommand cmd49 = new MySqlCommand("select count(sub1presentationofteacher)
from studentdetails where sub1presentationofteacher='" + 4 + "'", con21);
            MySqlCommand cmd50 = new MySqlCommand("select count(sub1presentationofteacher)
from studentdetails where sub1presentationofteacher='" + 5 + "'", con21);
            MySqlCommand cmd51 = new MySqlCommand("select
count(sub1abilitytoconveyconceptinsimplemanner) from studentdetails where
sub1abilitytoconveyconceptinsimplemanner='" + 1 + "'", con21);
            MySqlCommand cmd52 = new MySqlCommand("select
count(sub1abilitytoconveyconceptinsimplemanner) from studentdetails where
sub1abilitytoconveyconceptinsimplemanner='" + 2 + "'", con21);
            MySqlCommand cmd53 = new MySqlCommand("select
count(sub1abilitytoconveyconceptinsimplemanner) from studentdetails where
sub1abilitytoconveyconceptinsimplemanner='" + 3 + "'", con21);
            MySqlCommand cmd54 = new MySqlCommand("select
count(sub1abilitytoconveyconceptinsimplemanner) from studentdetails where
sub1abilitytoconveyconceptinsimplemanner='" + 4 + "'", con21);
            MySqlCommand cmd55 = new MySqlCommand("select
count(sub1abilitytoconveyconceptinsimplemanner) from studentdetails where
sub1abilitytoconveyconceptinsimplemanner='" + 5 + "'", con21);
            MySqlCommand cmd56 = new MySqlCommand("select
avg(sub1understandibilityofconcept) from studentdetails", con21);
```

```
            MySqlCommand cmd57 = new MySqlCommand("select avg(sub1punctualitywhileteaching)
from studentdetails", con21);
            MySqlCommand cmd58 = new MySqlCommand("select
avg(sub1abilitytoclearstudentdoubts) from studentdetails", con21);
            MySqlCommand cmd59 = new MySqlCommand("select avg(sub1tendencytobeimpartial)
from studentdetails", con21);
            MySqlCommand cmd60 = new MySqlCommand("select
avg(sub1abilitytogainstudentattention) from studentdetails", con21);
            MySqlCommand cmd61 = new MySqlCommand("select avg(sub1presentationofteacher)
from studentdetails", con21);
            MySqlCommand cmd62 = new MySqlCommand("select
avg(sub1abilitytoconveyconceptinsimplemanner) from studentdetails", con21);

            con21.Open();

             MySqlDataReader read21 = cmd21.ExecuteReader();
             if (read21.Read())
             {
                 label1.Text = read21[0].ToString();
             }
             read21.Close();

             MySqlDataReader read22 = cmd22.ExecuteReader();
             if (read22.Read())
             {
                 label2.Text = read22[0].ToString();


             }
             read22.Close();

             MySqlDataReader read23 = cmd23.ExecuteReader();
             if (read23.Read())
             {
                 label3.Text = read23[0].ToString();
}
read23.Close();

MySqlDataReader read24 = cmd24.ExecuteReader();
if (read24.Read())
{
   label4.Text = read24[0].ToString();

}
read24.Close();

MySqlDataReader read25 = cmd25.ExecuteReader();
if (read25.Read())
{
   label5.Text = read25[0].ToString();

}
read25.Close();
```

```
MySqlDataReader read26 = cmd26.ExecuteReader();
if (read26.Read())
{
    label6.Text = read26[0].ToString();

}
read26.Close();

MySqlDataReader read27 = cmd27.ExecuteReader();
if (read27.Read())
{
    label7.Text = read27[0].ToString();

}
read27.Close();

MySqlDataReader read28 = cmd28.ExecuteReader();
if (read28.Read())
```

```
{
    label8.Text = read28[0].ToString();

}
read28.Close();

MySqlDataReader read29 = cmd29.ExecuteReader();
if (read29.Read())
{
    label9.Text = read29[0].ToString();

}
read29.Close();

MySqlDataReader read30 = cmd30.ExecuteReader();
if (read30.Read())
{
    label10.Text = read30[0].ToString();

}
read30.Close();

MySqlDataReader read31 = cmd31.ExecuteReader();
if (read31.Read())
{
    label11.Text = read31[0].ToString();
}
read31.Close();

MySqlDataReader read32 = cmd32.ExecuteReader();
if (read32.Read())
{
    label12.Text = read32[0].ToString();
}
read32.Close();

MySqlDataReader read33 = cmd33.ExecuteReader();
```

```
  if (read33.Read())
  {
     label13.Text = read33[0].ToString();
  }
  read33.Close();

  MySqlDataReader read34 = cmd34.ExecuteReader();
  if (read34.Read())
  {
     label14.Text = read34[0].ToString();
  }
  read34.Close();

  MySqlDataReader read35 = cmd35.ExecuteReader();
  if (read35.Read())
{
   label15.Text = read35[0].ToString();
}
read35.Close();

MySqlDataReader read36 = cmd36.ExecuteReader();
if (read36.Read())
{
   label16.Text = read36[0].ToString();
}
read36.Close();

MySqlDataReader read37 = cmd37.ExecuteReader();
if (read37.Read())
{
   label17.Text = read37[0].ToString();
}
read37.Close();
```

```
  MySqlDataReader read38 = cmd38.ExecuteReader();
  if (read38.Read())
  {
  label18.Text = read38[0].ToString();
}
read38.Close();

MySqlDataReader read39 = cmd39.ExecuteReader();
if (read39.Read())
{
   label19.Text = read39[0].ToString();
}
read39.Close();

MySqlDataReader read40 = cmd40.ExecuteReader();
if (read40.Read())
{
   label20.Text = read40[0].ToString();
}
read40.Close();

MySqlDataReader read41 = cmd41.ExecuteReader();
```

```
if (read41.Read())
{
    label21.Text = read41[0].ToString();
}
read41.Close();

MySqlDataReader read42 = cmd42.ExecuteReader();
if (read42.Read())
{
    label22.Text = read42[0].ToString();
}
read42.Close();

MySqlDataReader read43 = cmd43.ExecuteReader();
if (read43.Read())
{
    label23.Text = read43[0].ToString();
}
```

```
read43.Close();

MySqlDataReader read44 = cmd44.ExecuteReader();
if (read44.Read())
{
    label24.Text = read44[0].ToString();
}
read44.Close();

MySqlDataReader read45 = cmd45.ExecuteReader();
if (read45.Read())
{
    label25.Text = read45[0].ToString();
}
read45.Close();

MySqlDataReader read46 = cmd46.ExecuteReader();
if (read46.Read())
{
        label26.Text = read46[0].ToString();
    }
    read46.Close();

    MySqlDataReader read47 = cmd47.ExecuteReader();
    if (read47.Read())
    {
        label27.Text = read47[0].ToString();
    }
    read47.Close();



    MySqlDataReader read48 = cmd48.ExecuteReader();
    if (read48.Read())
    {
        label28.Text = read48[0].ToString();
    }
    read48.Close();
MySqlDataReader read49 = cmd49.ExecuteReader();
if (read49.Read())
{
    label29.Text = read49[0].ToString();
}
read49.Close();


MySqlDataReader read50 = cmd50.ExecuteReader();
if (read50.Read())
{
    label30.Text = read50[0].ToString();
}
read50.Close();


MySqlDataReader read51 = cmd51.ExecuteReader();
if (read51.Read())
{
    label31.Text = read51[0].ToString();
  }
  read51.Close();


  MySqlDataReader read52 = cmd52.ExecuteReader();
  if (read52.Read())
  {
    label32.Text = read52[0].ToString();
  }
  read52.Close();


  MySqlDataReader read53 = cmd53.ExecuteReader();
  if (read53.Read())
  {
   label33.Text = read53[0].ToString();
}
read53.Close();


MySqlDataReader read54 = cmd54.ExecuteReader();
if (read54.Read())
{
   label34.Text = read54[0].ToString();
}
read54.Close();


MySqlDataReader read55 = cmd55.ExecuteReader();
if (read55.Read())
{
   label35.Text = read55[0].ToString();
}
read55.Close();
```

```
        MySqlDataReader read56 = cmd56.ExecuteReader();
        if (read56.Read())
        {
            label36.Text = read56[0].ToString();
        }
        read56.Close();

        MySqlDataReader read57 = cmd57.ExecuteReader();
        if (read57.Read())
        {
            label37.Text = read57[0].ToString();
        }
        read57.Close();

        MySqlDataReader read58 = cmd58.ExecuteReader();
        if (read58.Read())
        {
    label38.Text = read58[0].ToString();
}
read58.Close();

MySqlDataReader read59 = cmd59.ExecuteReader();
if (read59.Read())
{
    label39.Text = read59[0].ToString();
}
read59.Close();

MySqlDataReader read60 = cmd60.ExecuteReader();
if (read60.Read())
{
    label40.Text = read60[0].ToString();
}
read60.Close();

MySqlDataReader read61 = cmd61.ExecuteReader();
```

```
if (read61.Read())
{
   label41.Text = read61[0].ToString();
}
read61.Close();

MySqlDataReader read62 = cmd62.ExecuteReader();
if (read62.Read())
{
   label42.Text = read62[0].ToString();


}
read62.Close();

double result1 = double.Parse(label36.Text);
double result2 = double.Parse(label37.Text);
double result3 = double.Parse(label38.Text);
double result4 = double.Parse(label39.Text);
```

```
            double result5 = double.Parse(label40.Text);
            double result6 = double.Parse(label41.Text);
            double result7 = double.Parse(label42.Text);

            double result = result1 + result2 + result3 + result4 + result5 + result6 + result7;
            double v = result / 7;
            label43.Text = v.ToString();

            con21.Close();

          }
        }

        private void bunifuFlatButton1_Click(object sender, EventArgs e)
        {
            sa2 s2 = new
    sa2(label36.Text,label37.Text,label38.Text,label39.Text,label40.Text,label41.Text,label42.Text,label4
    3.Text);
            this.Hide();
            s2.Show();
        }

        private void bunifuThinButton22_Click(object sender, EventArgs e)
        {
            selfassessment sa = new selfassessment();
            this.Hide();
            sa.Show();
        }

        private void bunifuThinButton23_Click(object sender, EventArgs e)
        {
            sa1 s1 = new sa1(label1.Text, label2.Text, label3.Text, label4.Text, label5.Text, label6.Text,
    label7.Text, label8.Text, label9.Text, label10.Text, label11.Text, label12.Text, label13.Text,
    label14.Text, label15.Text, label16.Text, label17.Text, label18.Text, label19.Text, label20.Text,
    label21.Text, label22.Text, label23.Text, label24.Text, label25.Text, label26.Text, label27.Text,
    label28.Text, label29.Text, label30.Text, label31.Text, label32.Text, label33.Text, label34.Text,
    label35.Text);
```

```
            this.Hide();
            s1.Show();
    }
    }
    }
    }
```

Final project

ORIGINALITY REPORT

2%	1%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	1%
2	Submitted to Softwarica College of IT & E-Commerce Student Paper	1%
3	www.guru99.com Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		