TABLE OF CONTENTS

Topic	Page
Introduction & Motivations	3
Challenges	3-4
Component Diagram	4
Requirements	5 - 6
Non-functional Requirements	5
Functional Requirements	5-6
Use Case Diagram…….	6-10
Class Diargram……	10-11

Introduction

University and Result Management System is a web based online platform that allows only the members of a particular university. I have decided to develop this application mainly for the student to check out their result easily through online. It is also very useful for the teacher to handle student's attendance, their marks and other relevant issues. By using this application students can easily check out their result, their current standings, and can make applications for a leave. Teachers can update the data of their students, can view their profile and rank and also make applications for a leave. In the controller part there would be an admin who will manage both teachers and students log in information in case they forgot. Also the admin will add new teachers and students and delete information if needed.

Motivations

- The main goal of this applications is to make an efficient way for the students to check out their results.
- Using this application both teachers and students will be benefited. As they both can check out their current status and make any application anytime from anywhere.
- Especially when a teacher will not be able to attend the class he/she can notify the students and a notification will be popped out on those course's students.
- There would be a rating system for the teachers for a particular subject. Before taking that subject student can check out the ratings of that teacher.
- Main role would be played by the admin who would grant the applications of both teachers and students. Admin will take the marks of the students from the teacher and publish the result on a fixed date.

Challenges

- Since I am the only person of my group, it will be the hardest thing to complement the whole project as well as every project relevant issues.
- Implementing features such as dynamically update the absence of teachers and making an attractive UI for the website.
- Keepings the website simple and user friendly.
- Ensuring strict security for the teacher's account as teachers contains the marks of the students.
- Using the new language and framework.

Components Design

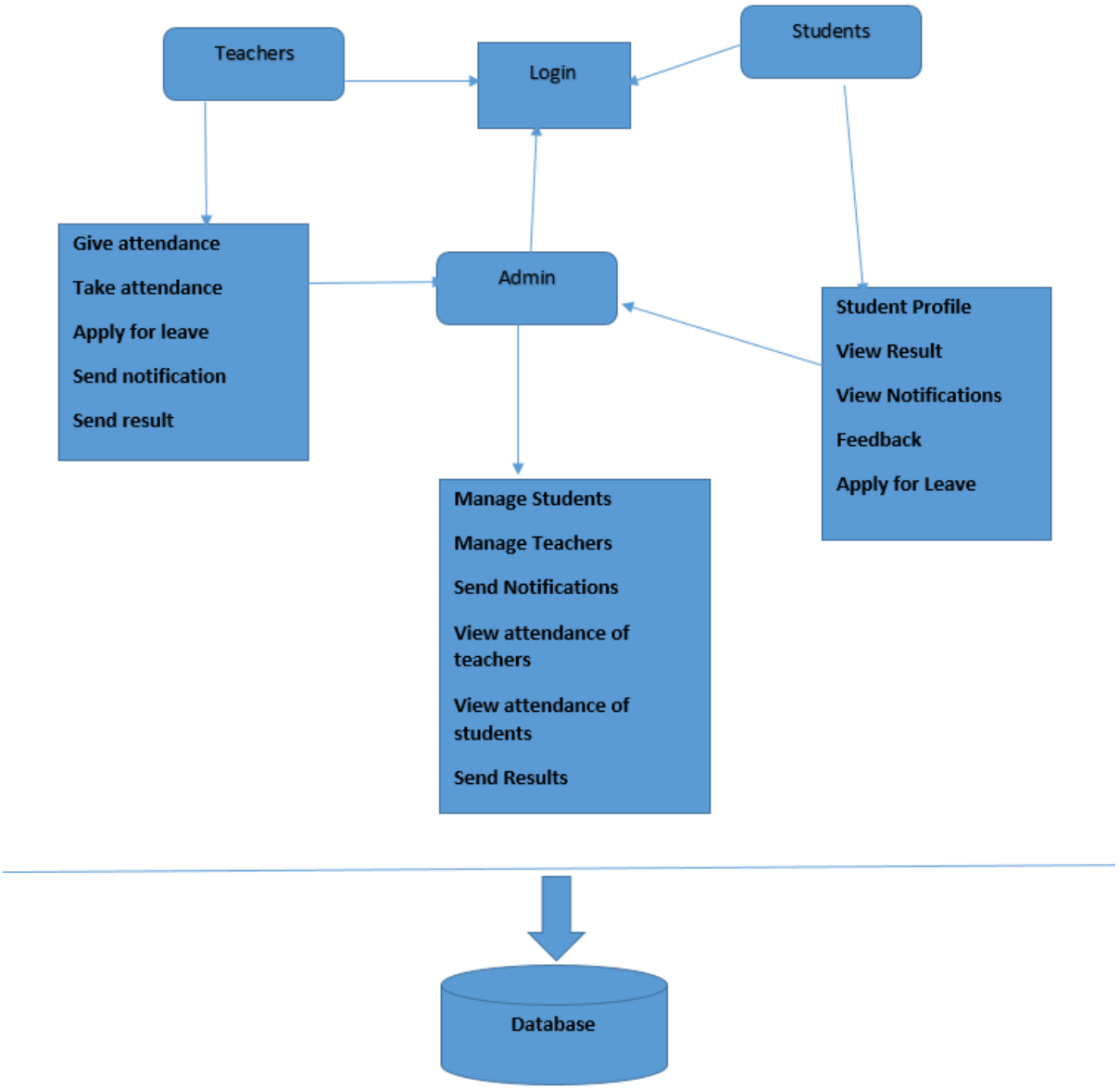


Fig: View and Model

Requirements

Non-Functional Requirements

- I have to strictly maintain teacher's information, their login details because those information are the most important thing as student's number is stored there. High priority would be given here.

- Ensuring excellent user experience (UX) by implementing a clean, intuitive, responsive UI.
- Application must be platform independent. Anyone can use it from and platform.
- Implementing features like social login and Google captcha, feedback reply.

Functional Requirements

- **User Verification:** There would be three types of users. Admin, Teachers and students. Admin will play the administration part. Teachers will be the most significant and important part of our website and strict security wild be maintain here. Teachers would be authorised as Editor. Students will be authorised as Reader.

  Scenario: When a user tries to access their account or home/profile page while they are not logged in, they would be redirected to the login page. Once the user enters their username and password and authentication is complete, they will be redirected to their desired page.
- **User Profile:** Every user would have their own profile contains their information, profile picture. Teachers can change their avatar anytime but students can't change their information. If they want to they have to contact with the admin and admin will update his data after validation.
- **Make Applications:** Teachers and students can make application for a day leave. Which would be granted by the admin.
- **Send Notification:** Both teachers and admin can send notifications to the students.
- **View Notifications:** Students can view all notifications and get updated news.

- **Check Result:** Students can login and check their results which would be updated by the admin.
- **Add or Delete:** Admin will add and delete new teachers or students when new teachers will be registered.

Use Case

Use Case 1	Registration	
Goal in Context		
Success End Condition	User can have access to all the features	
Failed End Condition	User will fail to have access to application features	
Primary Actor	User (i.e Admin,Teachers,Students)	
Trigger	Upon clicking the registration button, registration will Be completed	
Description	Step	Actions
	1	User = input their email id , password .
	2	User = re-input the password for confirmation.
	3	User information is checked for validity
	4	Upon valid Information, user is now registered and ready to have access all the features.

Use Case 2	Log in
Goal in Context	User will have the access to offered features.
Preconditions	User must have an account.
Success End Condition	User can log in their account and have access to all the features

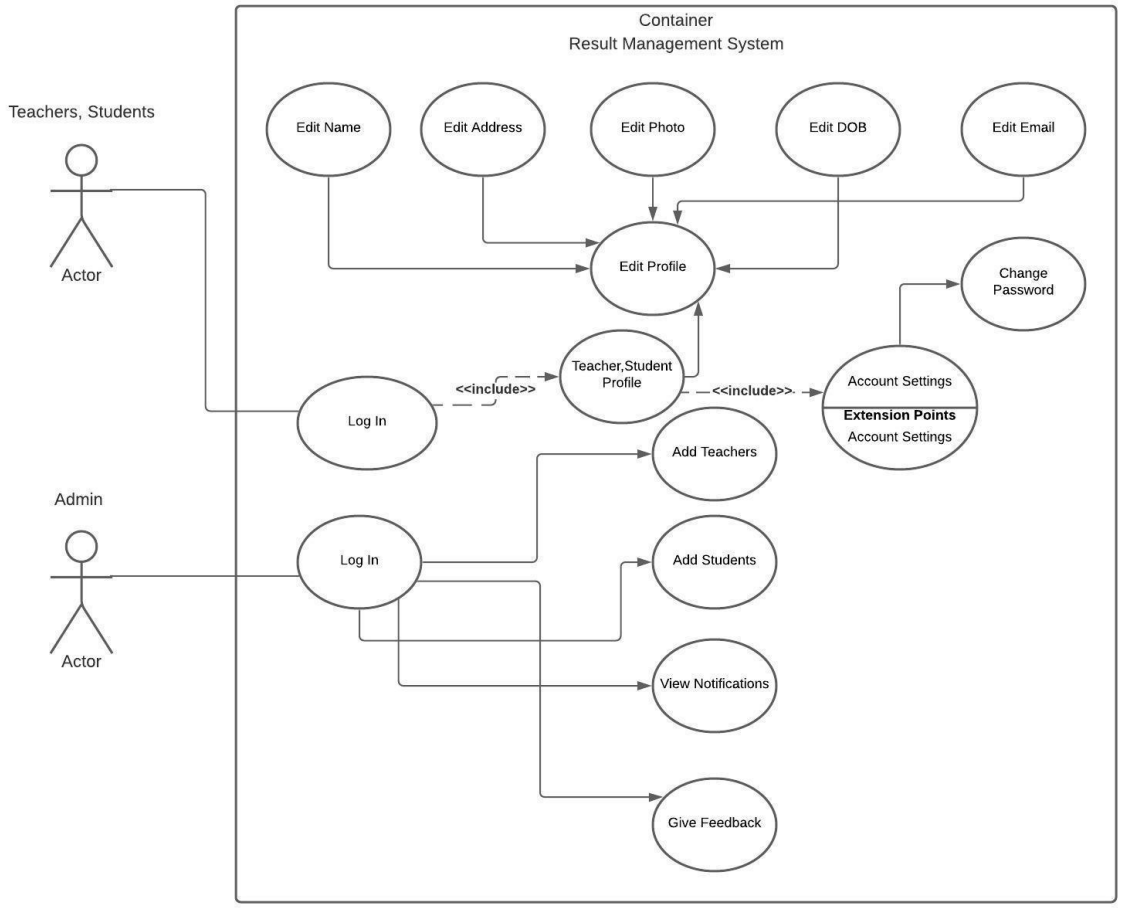
Failed End Condition	User id not logged in, can not access any features.	
Primary Actor	User (i.e Admin,Teachers,Students)	
Trigger	Upon clicking the log in button, user is now logged in	
Description	Step	Actions
	1	User inputs user email id and password
	2	Programme checks if user information is valid and present in database.
	3	User is allowed access to application and can browse through features.

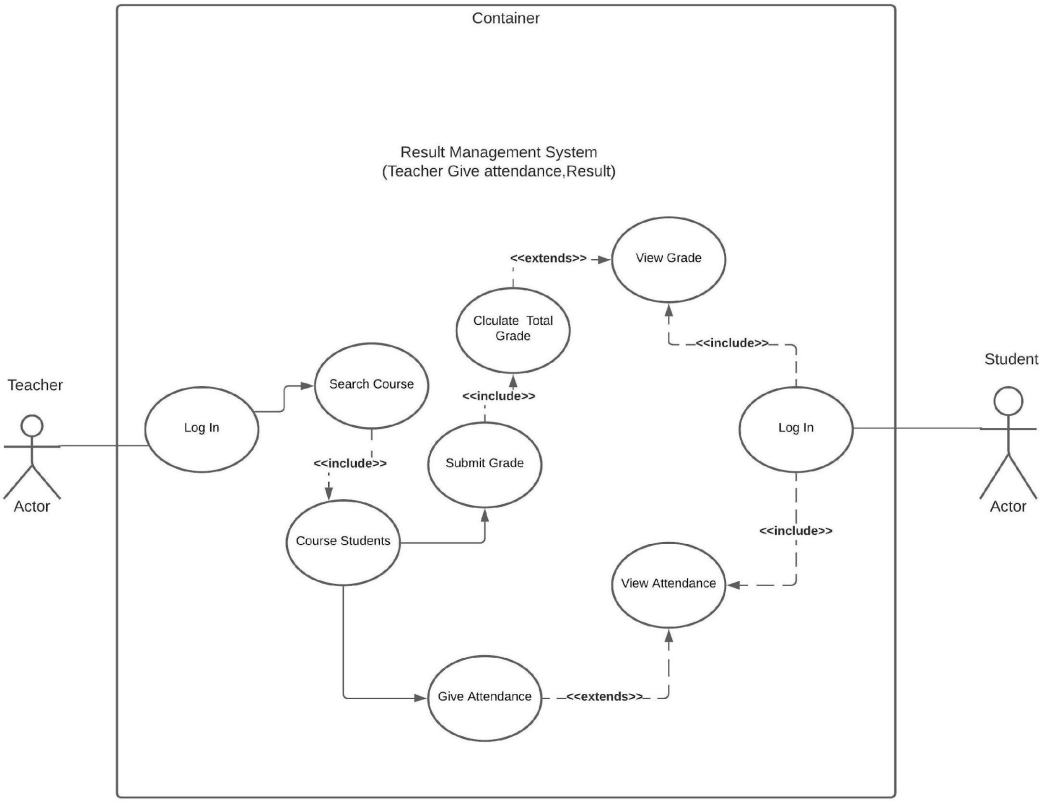
Use Case 3	Teacher give attendance	
Goal in Context	Teachers of every course will give attendance of their own course's students.	
Preconditions	Teacher must be logged in in his account.	
Success End Condition	Give attendance or any feedback.	
Failed End Condition	Attendance won't be count.	
Primary Actor	Teachers.	
Secondary Actors	Students.	
Trigger	Attendance sheet of assigned students list will appear.	
Description	Step	Actions
	1	Teachers browses assigned course.
	2	Find desired course.
	3	Give attendance.

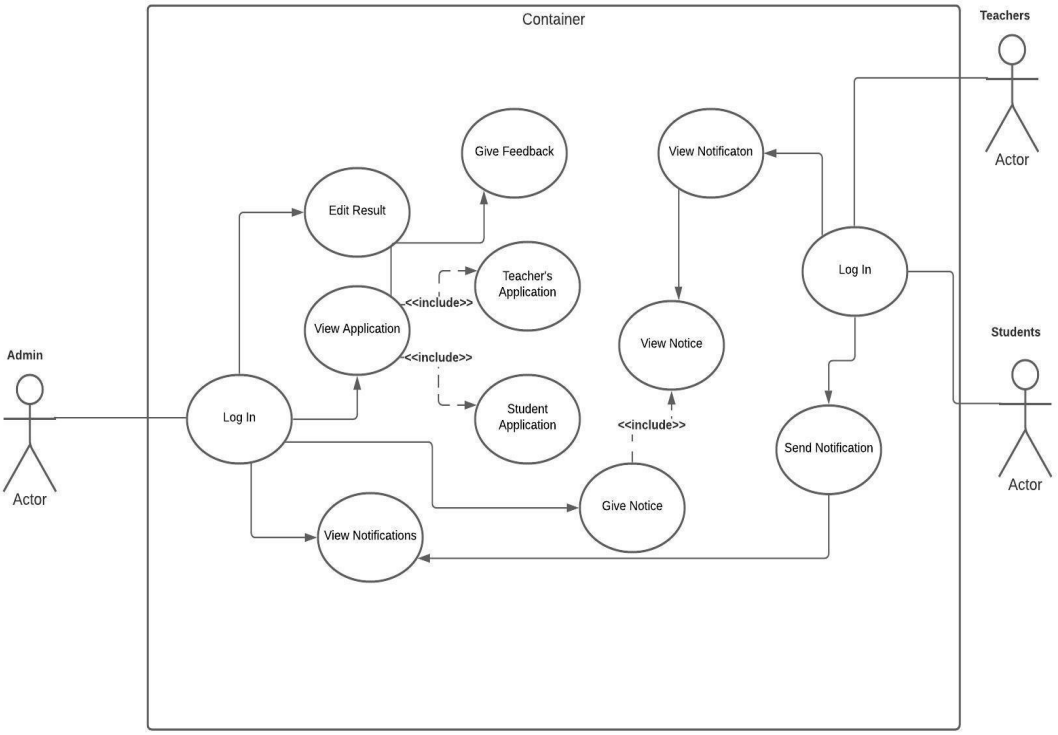
Use Case 4	Teacher give result
Goal in Context	Teachers of every course will give result of their own course's students.
Preconditions	Teacher must be logged in in his account.

Success End Condition	Submit result or any feedback.	
Failed End Condition		
Primary Actor	Teachers.	
Secondary Actors	Students.	
Trigger	Assigned student list will appear to submit their grade.	
Description	Step	Actions
	1	Teacher go to his assigned courses.
	2	Find desired course.
	3	Submit result.

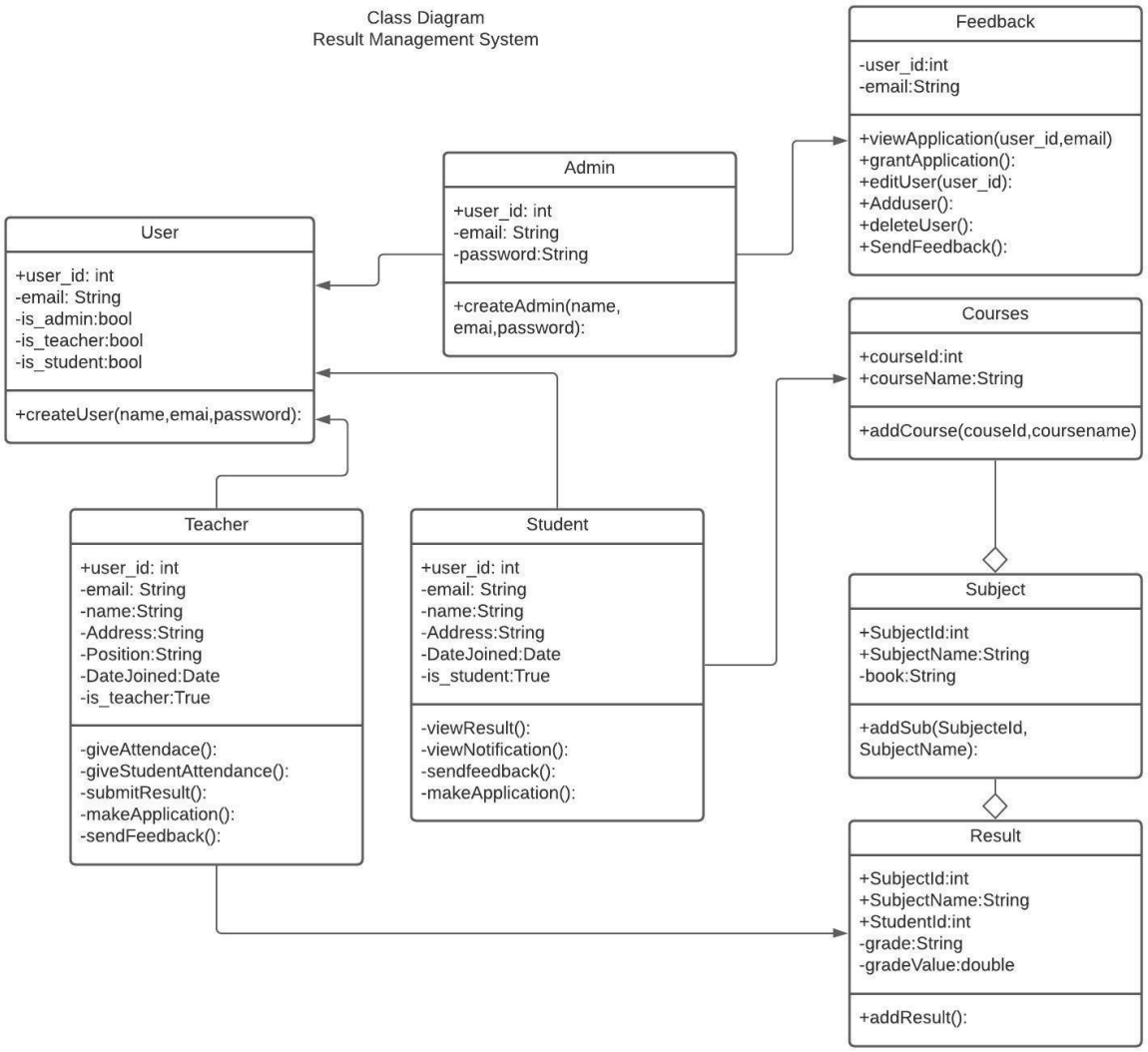
UML USE CASE



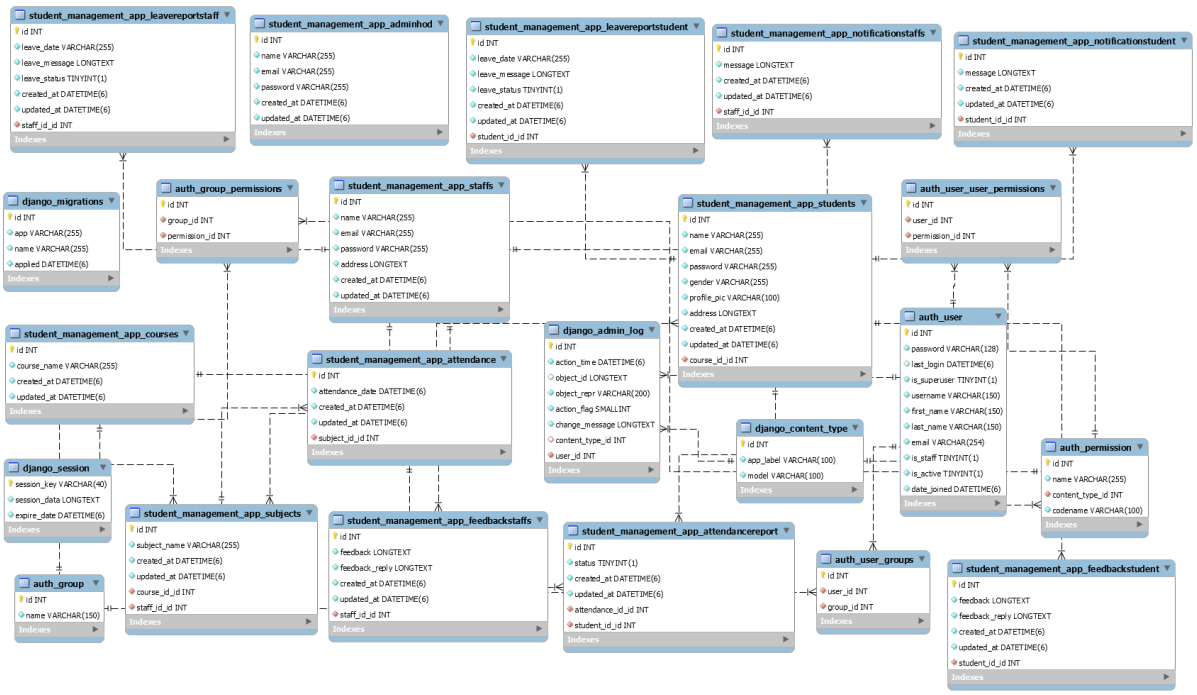




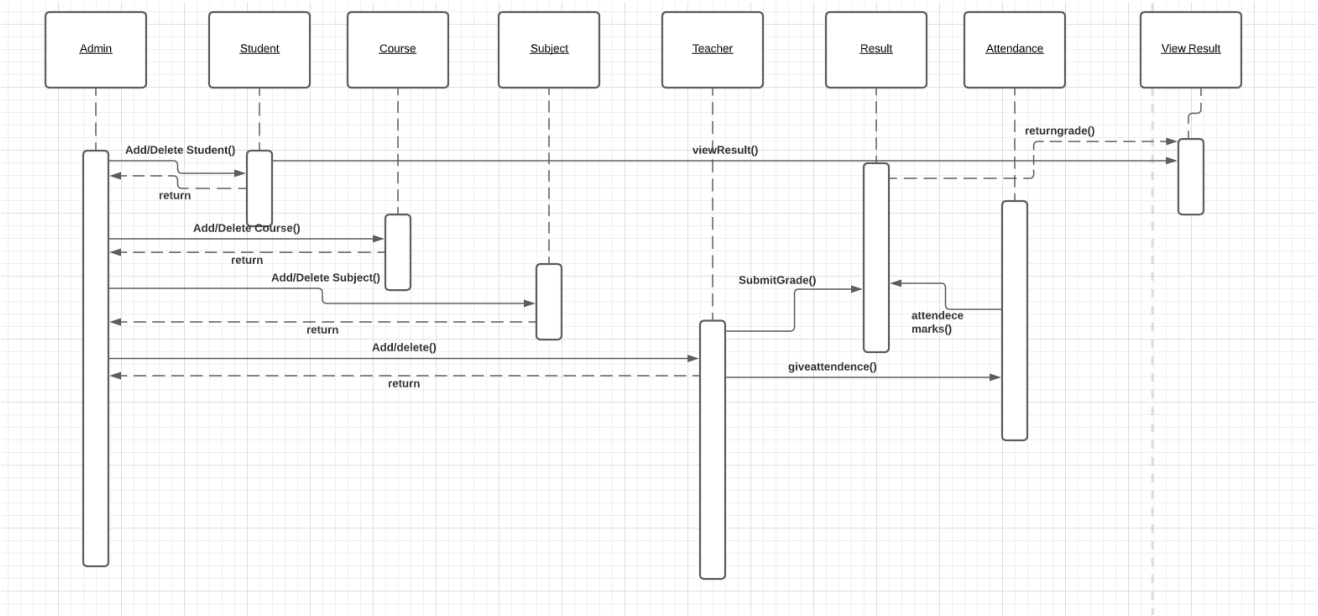
Class Diagram


Class Diagram
Result Management System
Feedback
-user_id:int
-email:String
+viewApplication(user_id,email)
+grantApplication():
+editUser(user_id):
+Adduser():
+deleteUser():
+SendFeedback():
Admin
+user_id: int
-email: String
-password:String
+createAdmin(name, emai,password):
User
+user_id: int
-email: String
-is_admin:bool
-is_teacher:bool
-is_student:bool
+createUser(name,emai,password):
Courses
+courseId:int
+courseName:String
+addCourse(couseId,coursename)
Teacher
+user_id: int
-email: String
-name:String
-Address:String
-Position:String
-DateJoined:Date
-is_teacher:True
-giveAttendace():
-giveStudentAttendance():
-submitResult():
-makeApplication():
-sendFeedback():
Student
+user_id: int
-email: String
-name:String
-Address:String
-DateJoined:Date
-is_student:True
-viewResult():
-viewNotification():
-sendfeedback():
-makeApplication():
Subject
+SubjectId:int
+SubjectName:String
-book:String
+addSub(SubjecteId, SubjectName):
Result
+SubjectId:int
+SubjectName:String
+StudentId:int
-grade:String
-gradeValue:double
+addResult():

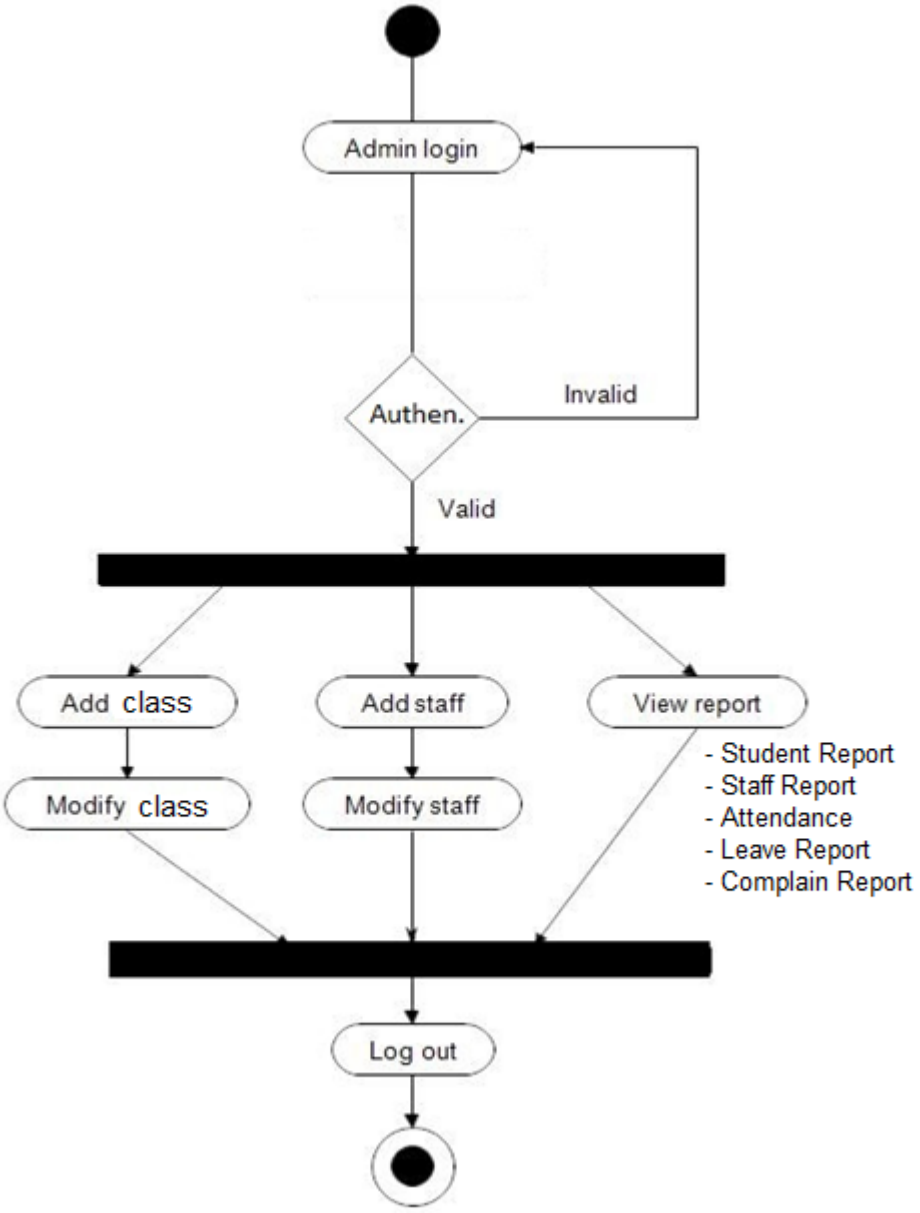
ER Diagram



Sequence Diagram

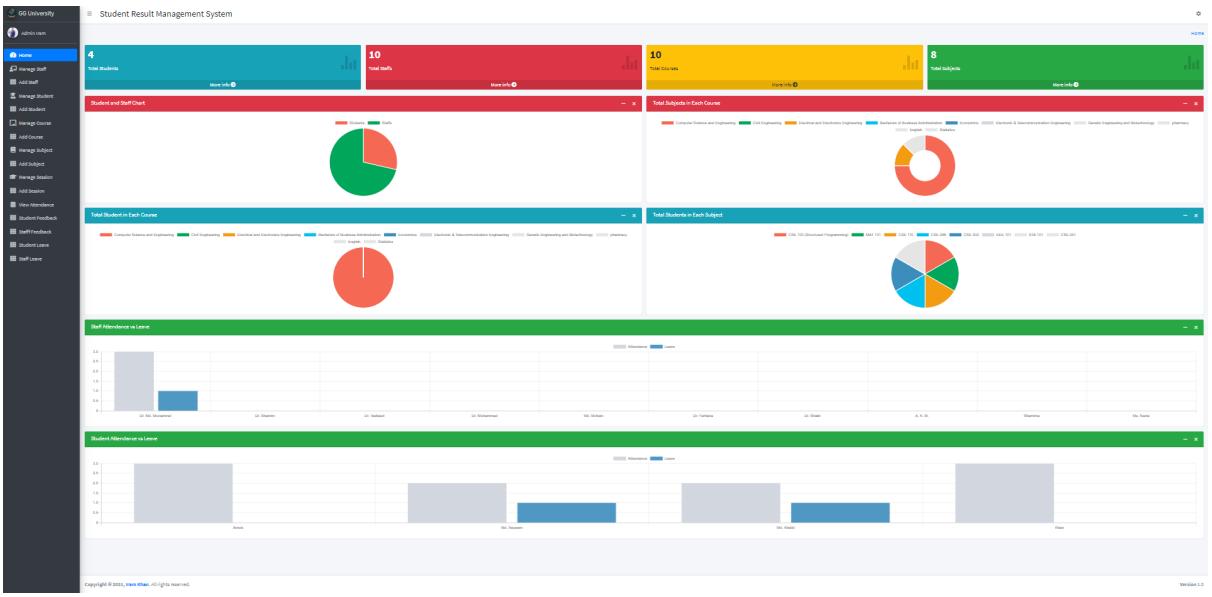


Activity Diagram

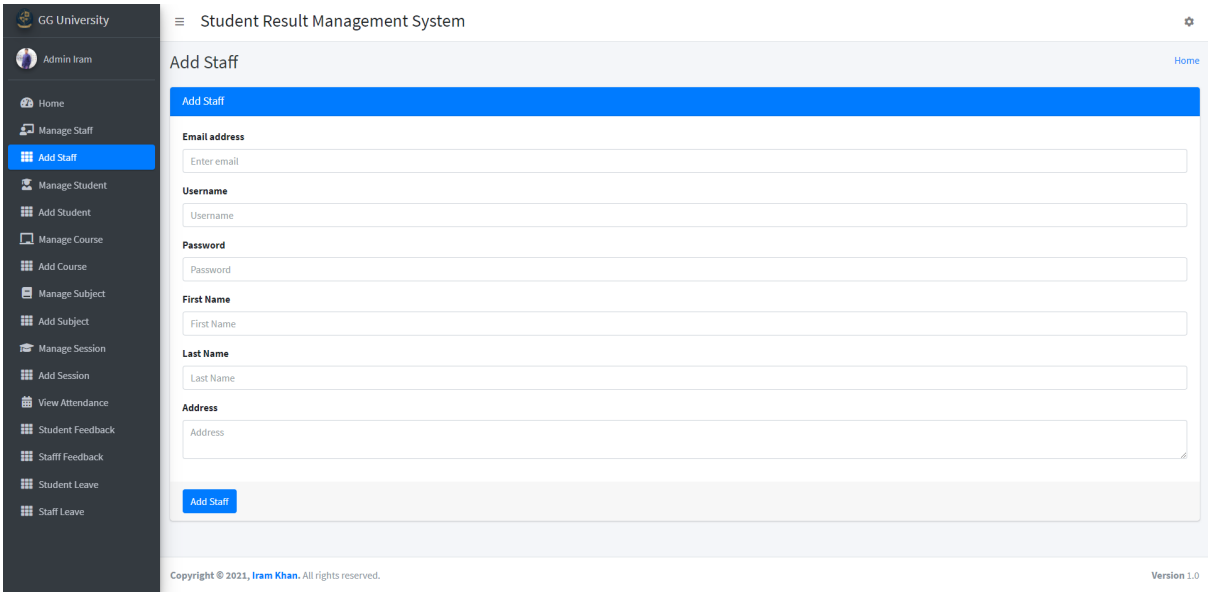


Graphical User Interface

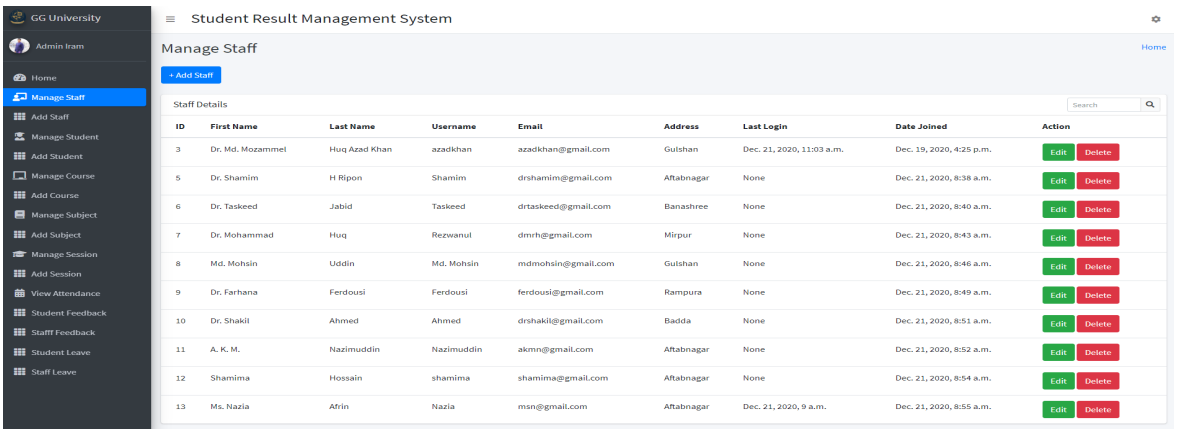
#Admin Home Interface



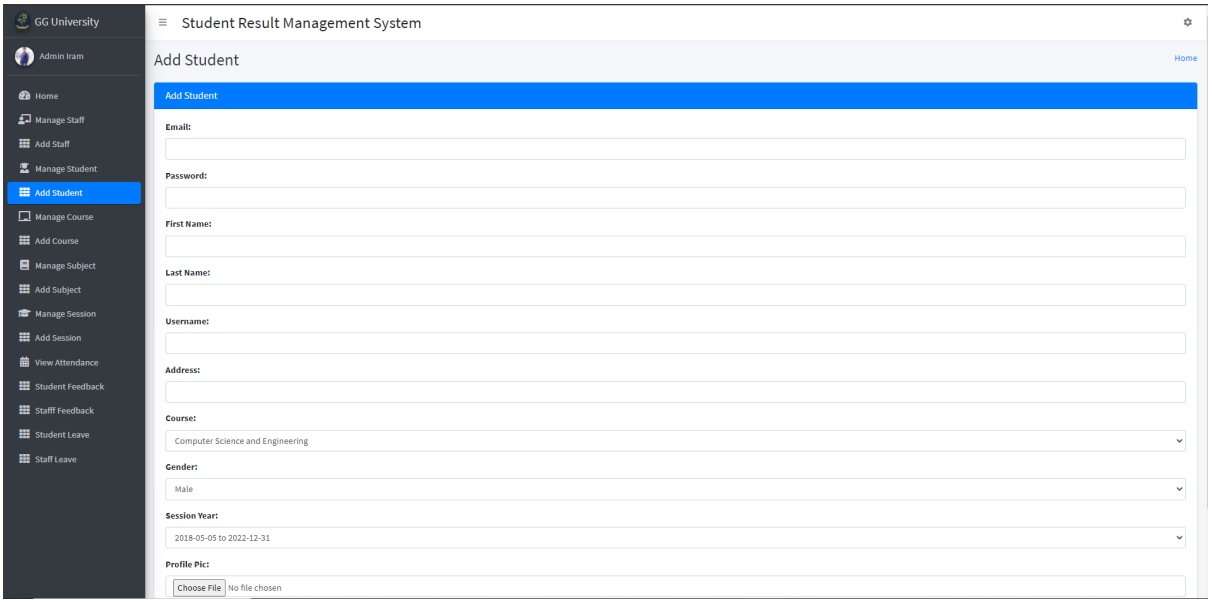
#Add Staff



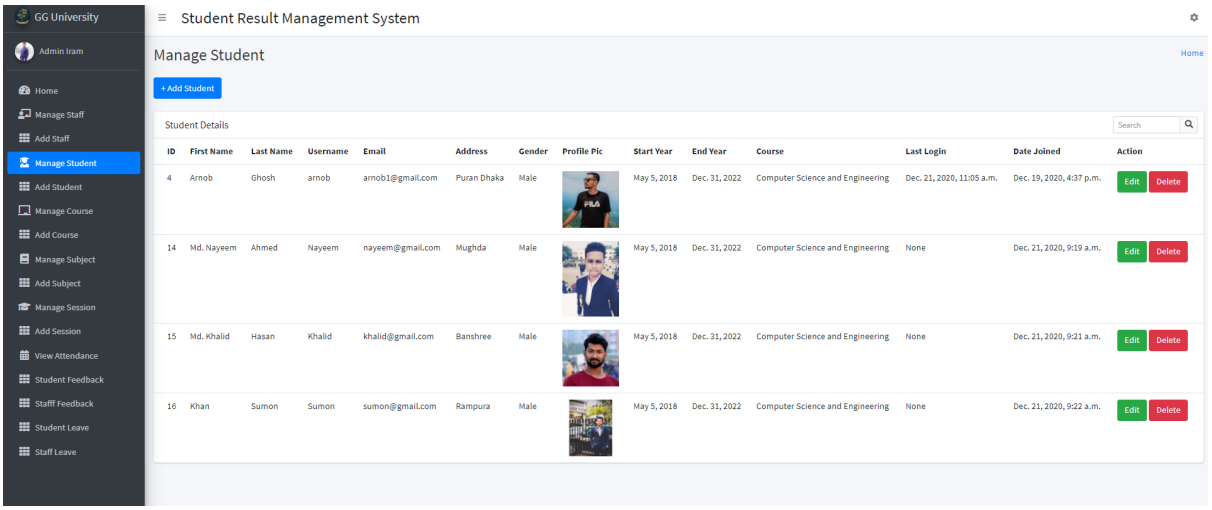
#Manage Staff



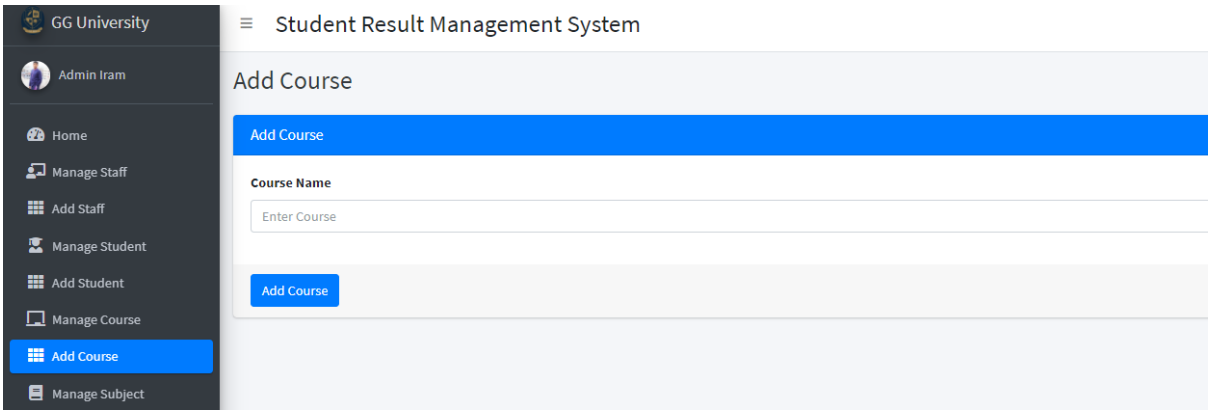
#Add Student



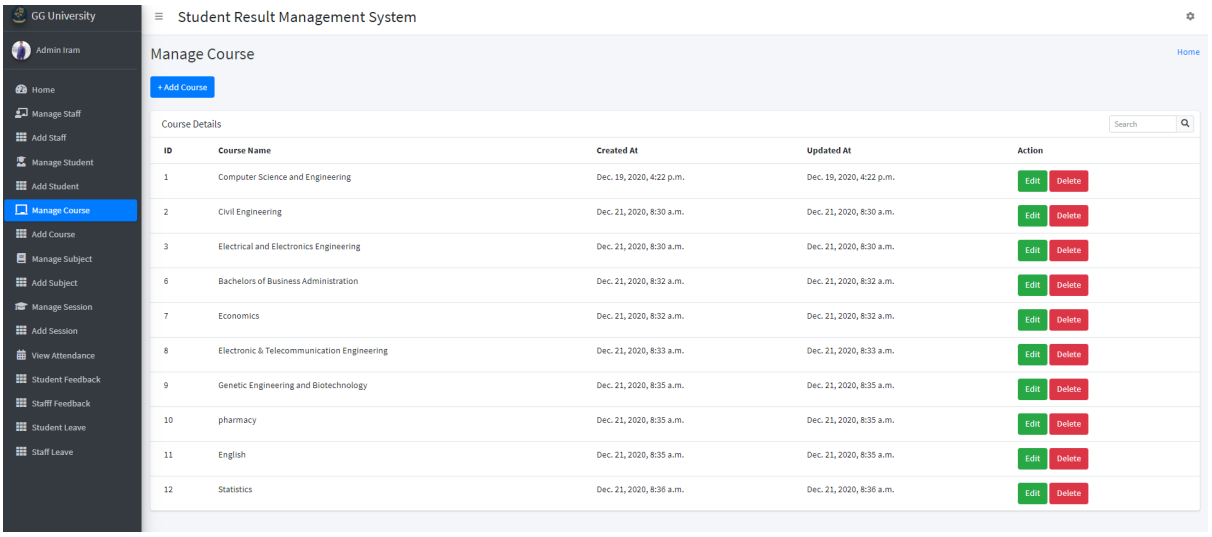
#Manage Student



#Add Course

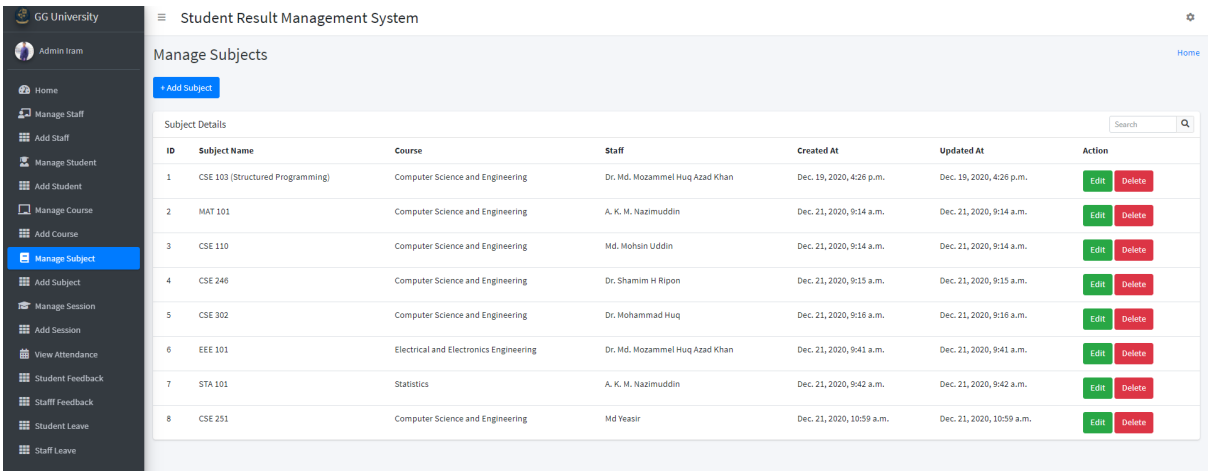


#Manage Course

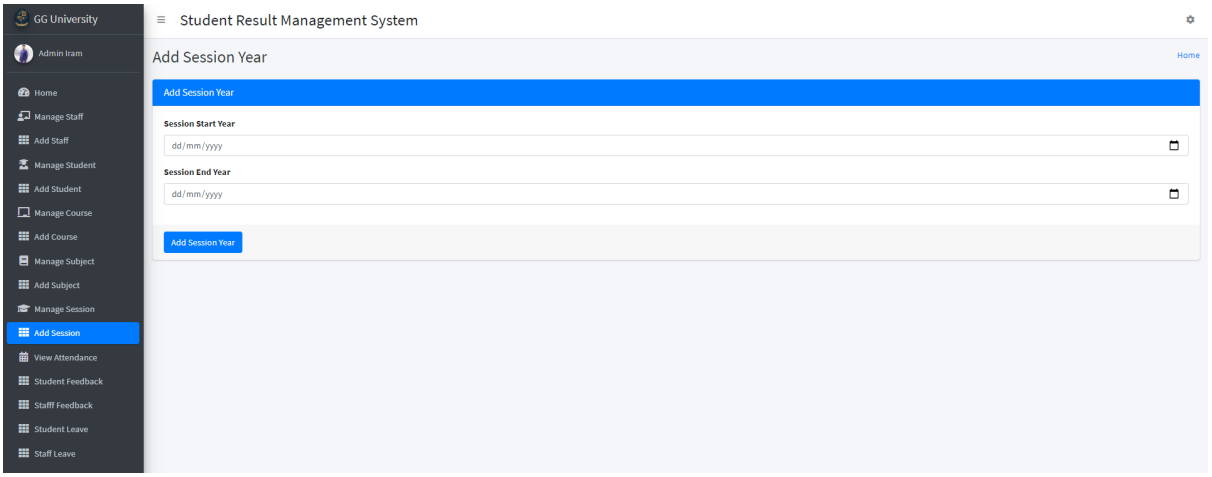


#Add Subject

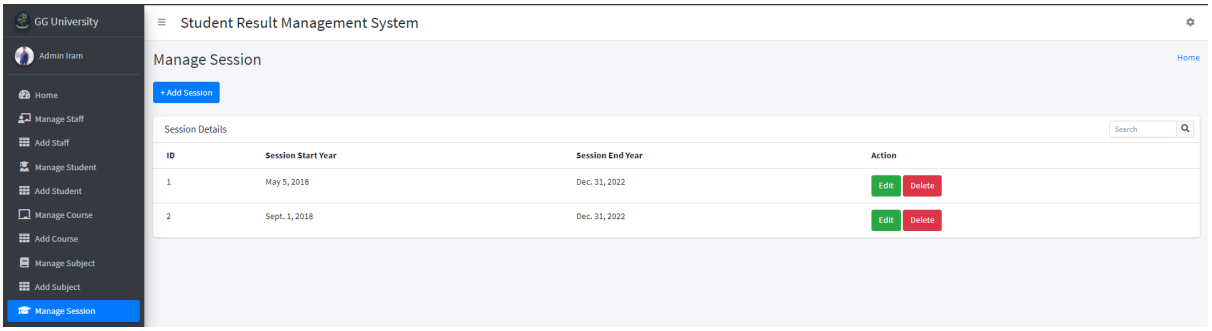
#Manage Subject



#Add Session

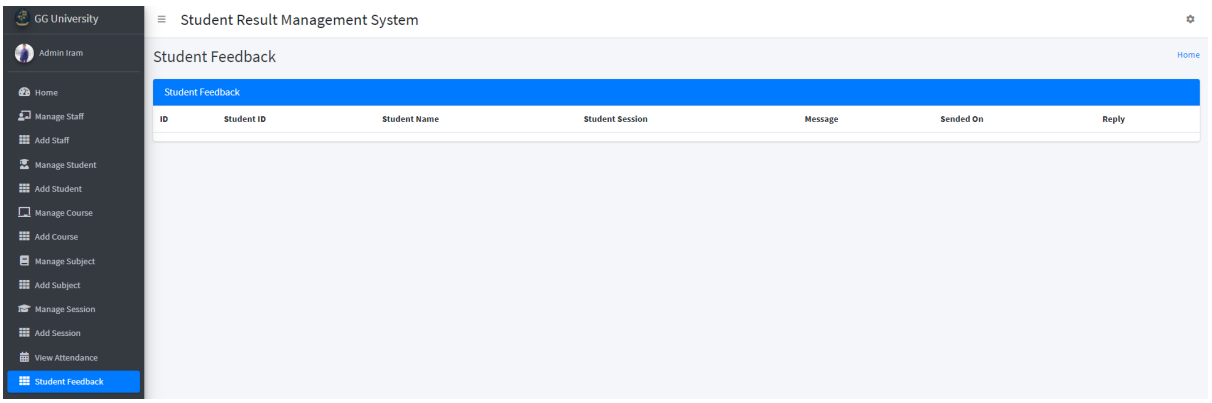


#Manage Session

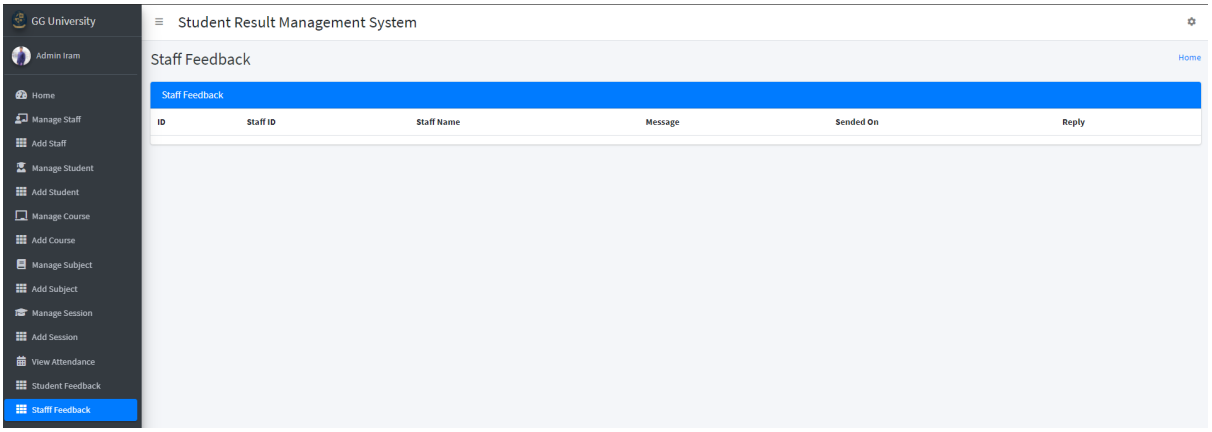


#View Attendance

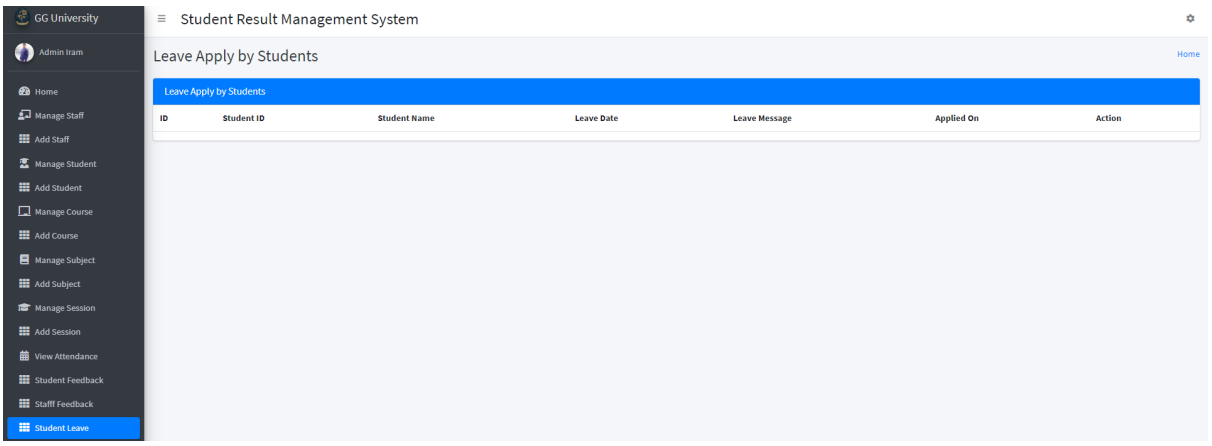
#Student Feedback



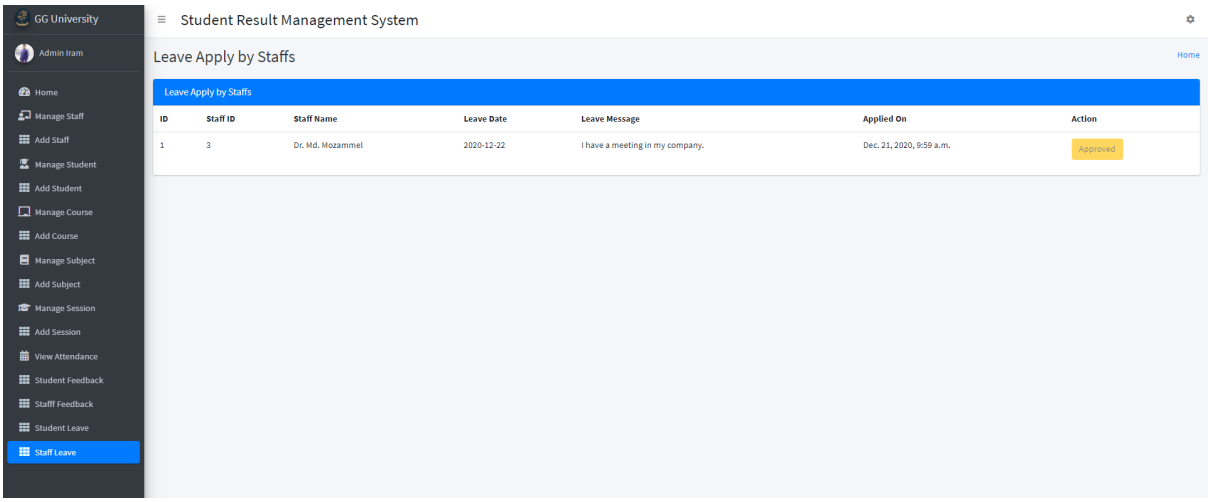
#Staff Feedback



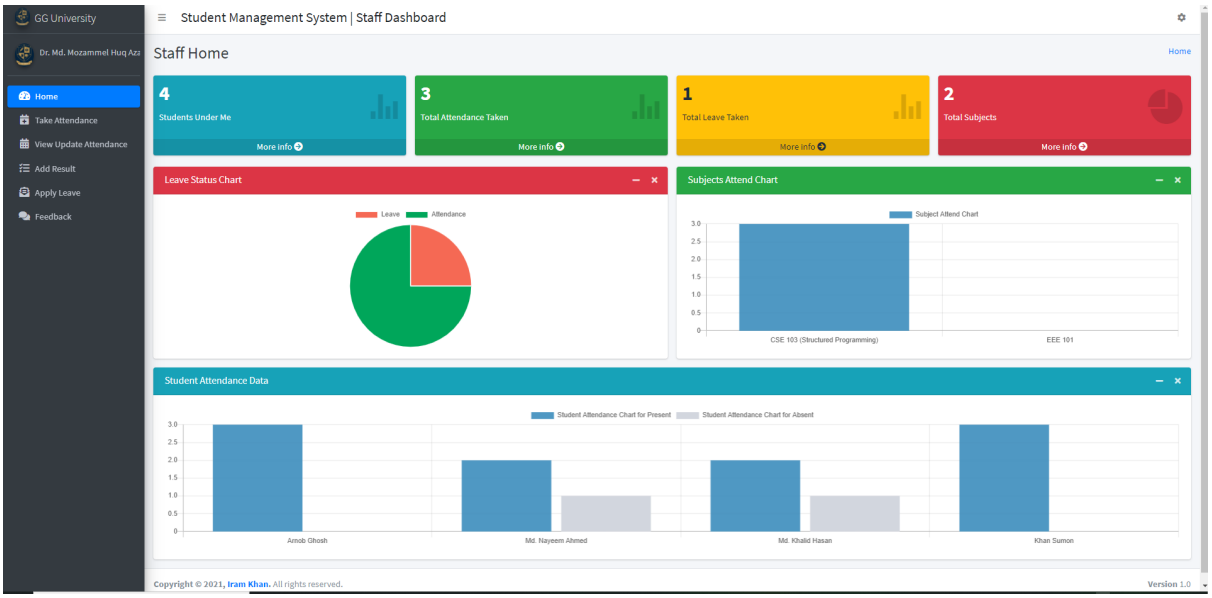
#Student Leave



#Staff Leave

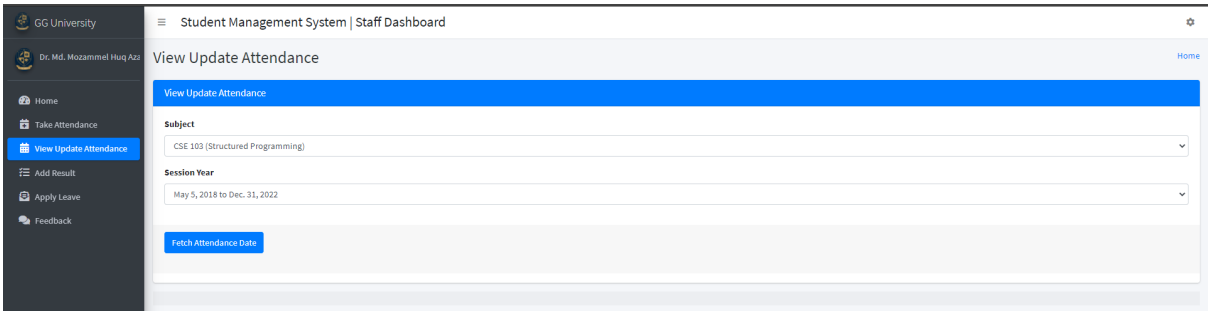


#Staff Home



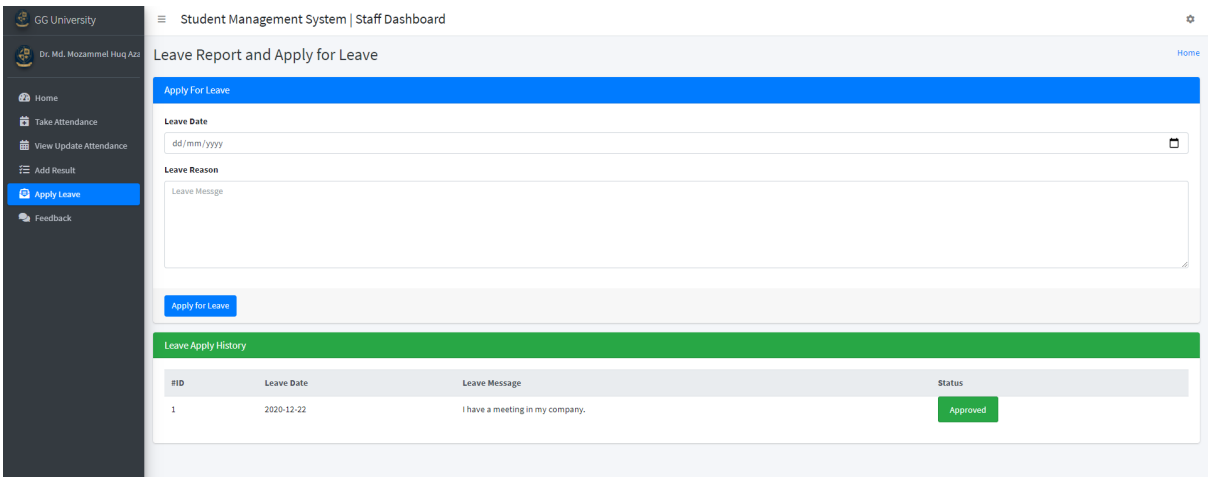
#Take Attendance

#view/Update Attendance

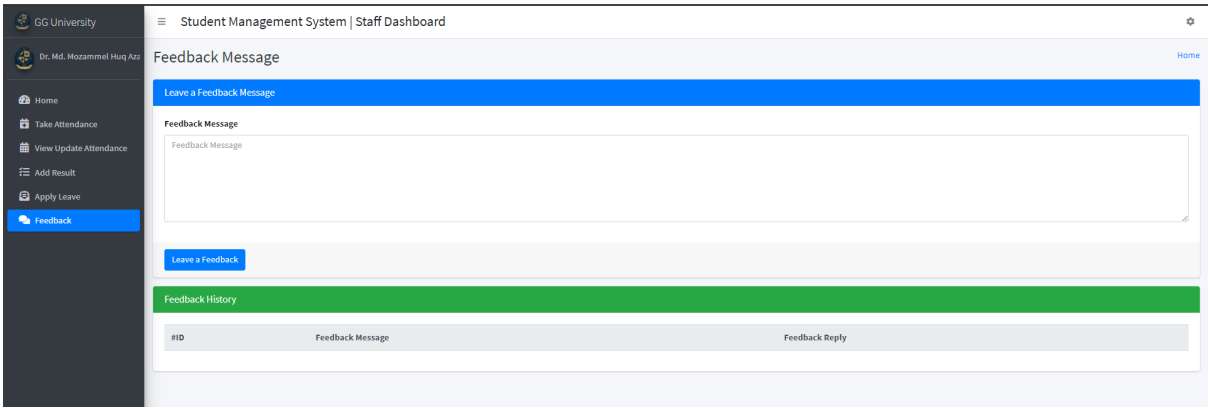


#Add Result

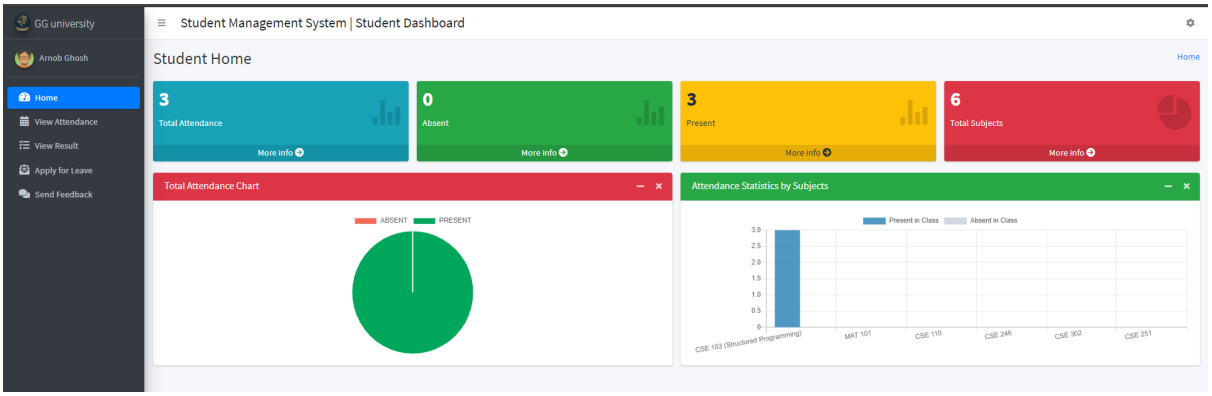
#Apply For Leave



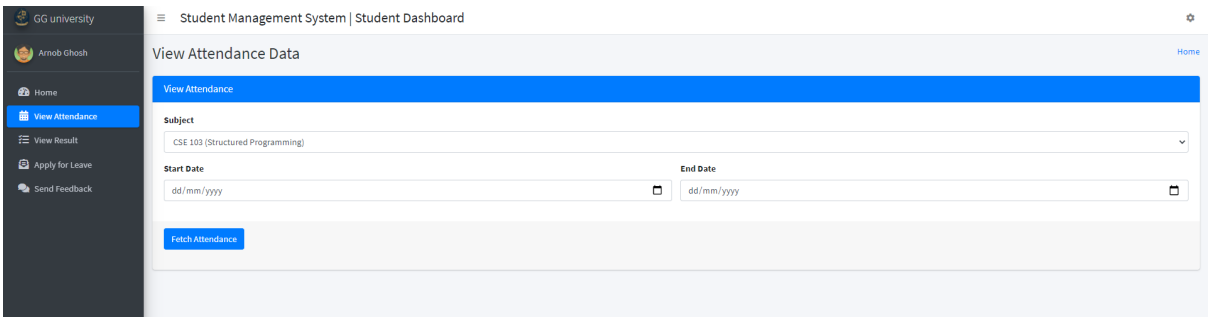
#Feedback



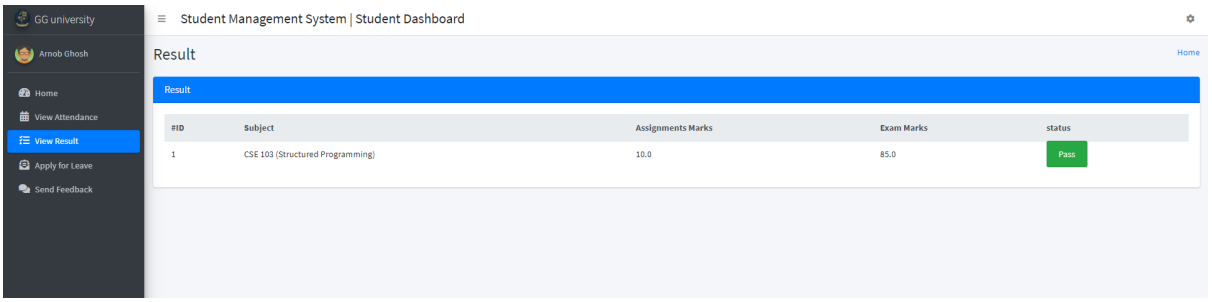
#Student Home



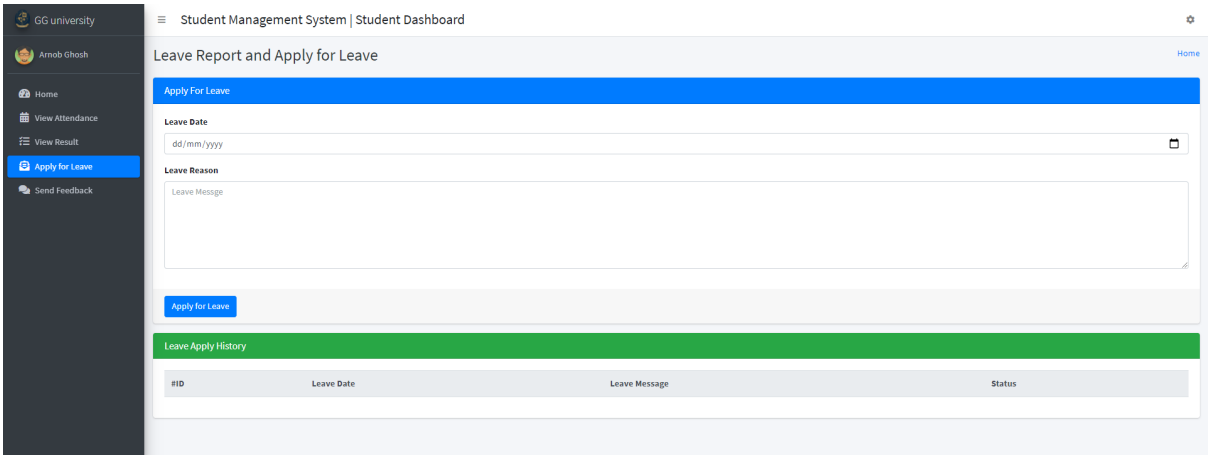
#View Attendance



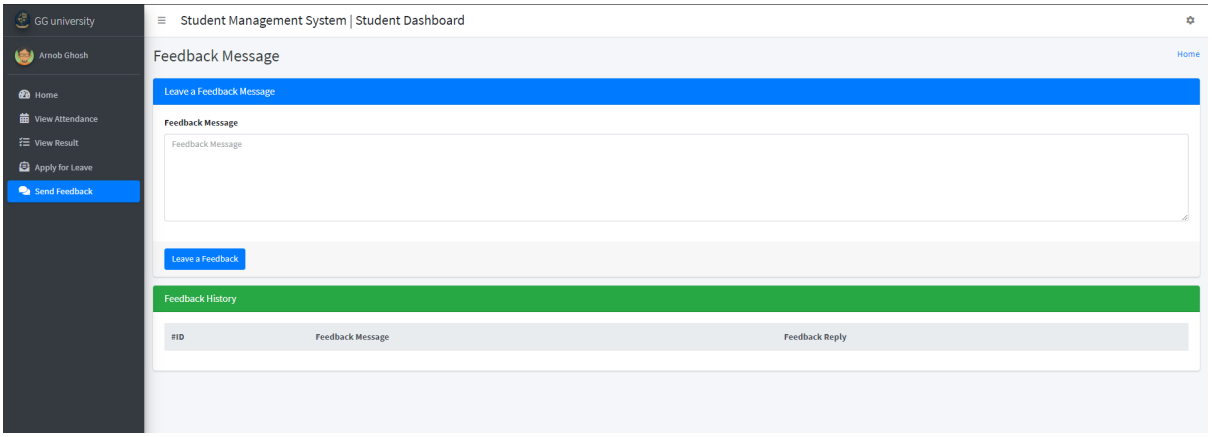
#View Result



#Apply For Leave



#Send Feedback



Use Case ID and Password

Admin: admin101@gmail.com; Pass: admin

Teacher: azadkhan@gmail.com; Pass: azadkhan

Student: arnob1@gmail.com; Pass: arnob