

MINOR PROJECT ON:
STUDENT INFORMATION MANAGEMENT SYSTEM

*Minor Project Submitted in Partial Fulfillment of the Requirements
for the degree of*

Master of Computer Application By

Shankhadeep Das

30701017017

Nisha Kumari

30701017028

Monika Kumari

30701017030

Anisha Mitra

30701017045

Under the guidance of
Prof. Pradosh Bandyopadhyay



Techno India Salt Lake
EM 4/1 Salt Lake City, Sector V,
Kolkata – 700091

<2019>

(Maulana Abul Kalam Azad University of Technology)

Techno India, Salt Lake
FACULTY OF MCA DEPARTMENT

Certificate of Recommendation

This is to certify that Shankhadeep Das, Nisha Kumari, Monika Kumari, Anisha Mitra have completed their project work titled “Minor project on: Student Information Management System”, under the direct supervision and guidance of Prof. Pradosh Bandyopadhyay. We are satisfied with their work, which is being presented for the partial fulfillment of the degree of Master of Computer Application (MCA), Maulana Abul Kalam Azad University of Technology, Kolkata– 700064.

Prof. Pradosh Bandyopadhyay

Date:

Prof. (Dr.) Monalisa Banerjee
HOD MCA Department
(Techno India, Salt Lake)

Date:

(Maulana Abul Kalam Azad University of Technology)

Techno India, Salt Lake
FACULTY OF MCA DEPARTMENT

Certificate of Approval

The foregoing Minor project is hereby approved as a creditable study of Master of Computer Application (MCA) and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or any statement made, opinion expressed or conclusion therein but approve this Minor project only for the purpose for which it is submitted.

Signature of the examiners

Final Examination for
Evaluation of the Project

☐ only in case the Minor project is approved

Contents

	Preface	
01.	Scope of the Project	
02.	Concepts and Problem Analysis Cost Analysis [use COCOMO model/ Function Point] Time Analysis Team Structure Software Configuration Management Quality Assurance Plan Risk Management	
03.	Web Application Survey	
04.	Theoretical background	
05.	Software Requirement Specifications	
06.	Design/ Solution/ Methodology Data Design Architectural Design Interface Design Procedural Design	
07.	Coding Standard Followed and Assumptions	

08.	Result Set Analysis	
09.	Testing	
10.	Future Scope of the Project	
11.	Conclusion	
12.	References and Bibliography	
13.	Appendix	

Preface

The project is about to handle all the information of the student regarding admission and examination. Also it manages resources which were managed and handled by manpower previously. The main purpose of the project is to integrate distinct sections of the organization into consistent manner so that complex functions can be handled smoothly by any technical or non-technical persons.

Student Information Management System deals with all kind of student details, academic related reports, institution details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semester years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result etc.

Our design can facilitate us to explore all the activities happening in an institute, even we can get to know which faculty is assigned to which course, the current status of a student, attendance percentage of a student and upcoming requirements of a student. The student data management system is an automated version of manual Student Data Management System. It can handle all details about student. The details include college details, subject details, student personnel details, academic details, exam details etc.

In case of manual system they need a lot of time, manpower etc. here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. Our system has two types of accessing modes, administrator and student. Student data management system is managed by an administrator. It is the job of the administrator to insert update and monitor whole process. When a student login to the system. He/she will view details of the student and they can perform some changes like updation of their contact details.

Organization and Content

This project is made to reduce manpower and manual paperwork for maintaining the records and to provide such a mechanism to make the manpower fast to maintain the information about teacher and student of an institute.

Table 1 is the project characteristics table from where we take some values for the calculation of EAF (effort adjustment factor).

Table 2 is the coefficients for intermediate COCOMO which we used in effort calculation of the project.

Table 3 is used to show the time analysis.

Table 4 is the list of test cases.

Diagram 1 describes team structure of this project.

Image 1 is entity relationship diagram (ERD), **Image 2** is controller class diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects, **Image 3** is model class diagram which describe a system at various levels of abstraction and **Image 4** is pojo class diagram which is used to describe a class that doesn't need to be a subclass of anything, or implement specific interface.

Acknowledgements

We would like to express our special thanks of gratitude to our principal xyz....., our guide Prof. Pradosh Bandyopadhyay as well as our HOD Prof. (Dr.) Monalisa Banerjee who gave us the golden opportunity to do this wonderful project on the topic Student Information Management System which also helped us in doing a lot of research and we came to know about so many new things. We are really thankful to them. Secondly we would also like to thank our parents and friends who helped us a lot in finishing this project within the limited time. We are making this project not only for marks but to also increase our knowledge.

THANKS AGAIN TO ALL WHO HELPED US

SHANKHADEEP DAS
NISHA KUMARI
MONIKA KUMARI
ANISHA MITRA

(MCA 5th Sem.)

STUDENT INFORMATION MANAGEMENT SYSTEM

Scope of the Project

The name of our project is Student Information Management System where the maintenance of student information has been done as the minor project of the college. There are many departments of administration for the maintenance of college information and student databases in any institution. This is one the most essential requirement of any college. Our project provides different and various record regarding students which will be very user friendly and easy to access for the students and as well as for the teachers it will be easier to maintain all the records of the students.

In our following project, firstly, student can register themselves in their following college to get their respective student id which will be unique and also easier to recognize the students with their respective student id.

The student request for generating the id will be approved by the administrator. The approval has been done by the administrator after verifying the valid details of the student. The students can view their particular fields that they have given while filling the registration form which will be required if any student by mistake provided a wrong field like phone number or address, it will be changed and updated by them.

Administrator can also add new user like coordinator, faculty, and head of the department.

All the student request can be approved by the head of the department as well.

Administrator can allocate teachers for each semester with respect to different subjects. Now the teachers allocated for the particular subject for a particular semester can upload the marks of their respective students and also can view the marks of the particular subject. Also teachers can search record of a particular student as well as the old students.

The head of the department of a particular department can only view the marks year wise of all the students of all semesters. He can also provide the notice which will be visible to all the students of that particular department.

Student can only view the marks, uploaded by the teachers. Student can search a particular record which they need. Students can also see their semester results which will be uploaded by the teachers. Also, after semester the 1st, 2nd, 3rd position will be determined.

This project will be very helpful for the organization in building a better communication and management between the HOD, teachers and students.

Cost Analysis

COCOMO

The basic COCOMO model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software system. However, in reality, no system's effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, Capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation.

(i) Product attributes –

- Required software reliability extent
- Size of the application database
- The complexity of the product

(ii) Hardware attributes –

- Run-time performance constraints
- Memory constraints
- The volatility of the virtual machine environment
- Required turnabout time

(iii) Personnel attributes –

- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience

(iv) Project attributes –

- Use of software tools
- Application of software engineering methods
- Required development schedule

The project manager is to rate these 15 different parameters for a particular project on a scale of one to three. Then, depending on these ratings, appropriate cost driver values are taken from the above table. These 15 values are then multiplied to calculate the EAF (Effort Adjustment Factor). The Intermediate COCOMO formula now takes the form:

$$E = (a (KLOC)^b) * EAF$$

Our project is of organic type and we are taking the KLOC as 2.5 according to our project. Now we need to calculate the value of the EAF taking all high values.

We take those values from this following chart

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personal attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Table 1

Values which are taken from table 1 for EAF calculation are:

Software reliability : 1.15
 Product complexity : 1.15
 Required turnabout time : 1.07
 Analyst capability : 0.86
 Applications experience : 0.91
 Software engineer capability : 0.86
 Programming language experience : 0.95
 Use of software tools : 0.91

$$\begin{aligned} \text{EAF} &= 1.15 * 1.15 * 1.07 * 0.86 * 0.91 * 0.86 * 0.95 * 0.91 \\ &= 0.823 \end{aligned}$$

The values of a and b in case of the intermediate model are as follows

	a	B	C	d
ORGANIC	2.4	1.05	2.5	0.38
SEMI- DETACHED	3.0	1.12	2.5	0.35
EMBEDDED	3.6	1.20	2.5	0.32

Table 2

As our software is organic in nature,
 a=2.4, b=1.05, c=2.5, d=0.38

$$\begin{aligned} E_i &= 2.4 * (2.5)^{1.05} \\ &= 6.28128141 \\ &= 6.281(\text{approx.}) \end{aligned}$$

$$\begin{aligned} \text{Effort} &= E_i * \text{EAF} \\ &= 6.281 * 0.823 \\ &= 5.169 \text{ Person-Months} \end{aligned}$$

$$\begin{aligned}
 \text{Development time} &= c * (\text{Effort})^d \\
 &= 2.5 * (5.169)^{0.38} \\
 &= 4.669 \text{ Months} \\
 &= 142 \text{ days}
 \end{aligned}$$

Time Analysis

ANALYSIS	20%	28 DAYS
DESIGN	20%	28 DAYS
CODING	35%	50 DAYS
TESTING	25%	36 DAYS

Table 3

Team Structure:

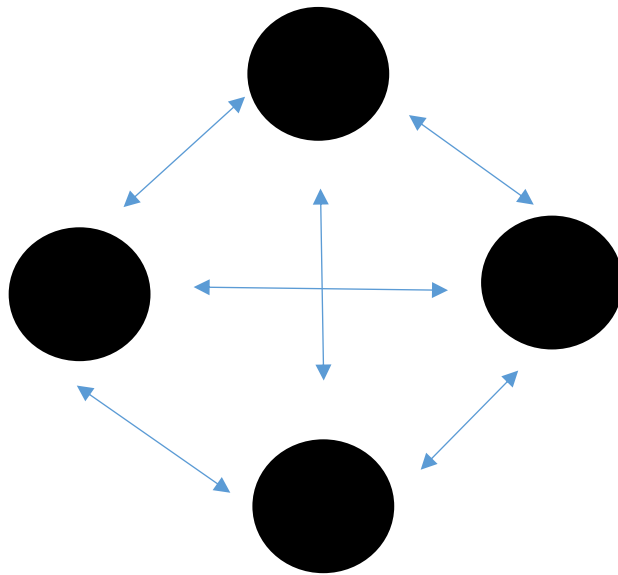
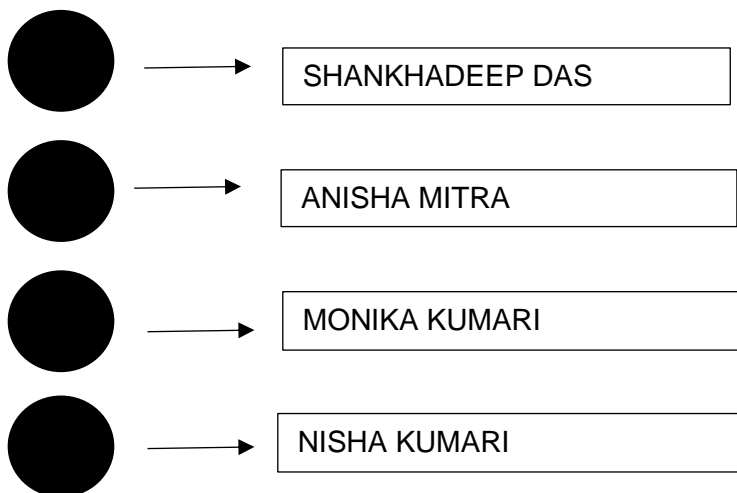


Diagram 1



Software Configuration Management

The software configurations are as follows:

- **Usability**: The links are provided for each form. The user is facilitated to assign teachers, un-assign teachers, update, view marks of students. Also, students can see their details which includes their roll number, year, subject and that respective subject's number. Validations are provided in each field to avoid inconsistent or invalid entry in the databases. The forms contain text boxes and drop-down lists, so that reports can be produced.
- **Security**: Application will allow only valid users to access the system. There is only one type of user. Security is based upon the individual user ID and Password.
- **Maintainability**: This system will be maintained by the database administrator.
- **Availability**: System will be available around the clock except for the time required for the backup of data. The system will always be available to the students to check their marks as well as to view notices.
- **Portability**: The application is developed in HTML and CSS platform. It would be portable to other operating system provided android is available for the OS. As the database is made in SQL Server 2008 R2, porting the database to another database server would require some development effort

Quality Assurance Plan

Software Quality Assurance (SQA) is simply a way to assure quality in the software. It is the set of activities which ensure processes, procedures as well as standards suitable for the project and implemented correctly.

Software Quality Assurance is a process which works parallel to development of a software. It focuses on improving the process of development of software so that problems can be prevented before they become a major issue. Software Quality Assurance is a kind of an Umbrella activity that is applied throughout the software process.

Here we will be doing few steps to do the quality assurance;

SQA Management Plan:

Make a plan how you will carry out the SQA throughout the project. Think which set of software engineering activities are the best for project.

Set the Check Points:

SQA team should set checkpoints. Evaluate the performance of the project on the basis of collected data on different check points.

Multi testing Strategy:

Do not depend on single testing approach. When you have lot of testing approaches available use them.

Measure Change Impact:

The changes for making the correction of an error sometimes re introduces more errors keep the measure of impact of change on project. Reset the new change to change check the compatibility of this fix with whole project.

Manage Good Relations:

In the working environment managing the good relation with other teams involved in the project development is mandatory. Bad relation of SQA team with programmers team will impact directly and badly on project. Don't play politics.

Benefits of Software Quality Assurance (SQA):

SQA produce high quality software.
High quality application saves time and cost.
SQA is beneficial for better reliability.
SQA is beneficial in the condition of no maintenance for long time.
High quality commercial software increase market share of company.
Improving the process of creating software.
Improves the quality of the software.

Disadvantage of SQA:

There are a number of disadvantages of quality assurance. Some of them include adding more resources, employing more workers to help maintain quality and so much more.

Risk Management

Every project has uncertainties and they may affect the project success. These uncertainties are known as risk. There is a need to determine the risks involved in the project to reduce the chance of its occurrence as well as develop a plan to either mitigate or avoid the risks that have been identified. There are many risks like crashing your software due to many occurrences of risks during the ongoing of project. Technology is such a big part of academic processes today and it's hard to imagine to live without them. But it has a downside. Universities have been a favourite target for identity theft and frauds. If proper precautions are not taken, records in educational institutions are vulnerable to security breaches. Colleges and universities have a reputation of weaker data security, making them susceptible to data loss. The huge volume of student and staff data built into the system, along with high turnover and a collaborative multi-platform environment results in rampant data loss at institutions. Here are five top data risks and tips on how to protect education institutes against them.

1. Time

Time, one of the triple constraints (time, cost, quality) is likely one of the most important and well-understood resources that everyone acknowledges. Everyone, regardless of whether they are educated in project management or not, understands that regardless of whether time is used it is consumed and therefore priceless.

You will need to identify early the impact that time will have on cost—for example, if you have an opportunity to complete some tasks early with additional staff working on the event you will increase your costs.

2. Knowledge

Knowledge risk is a term that is seldom defined in the literature. One definition found was proposed by Perrott (2007) who describes a knowledge risk as a likelihood of any loss resulting from the identification, storage or protection of knowledge that may decrease the operational or strategic benefit of a company. In the opinion of the authors of the present paper, this definition needs further elaboration. First, the term “risk” should be defined. According to Haimes (2009), risk is “a measure of the probability and severity of adverse effects (i.e., consequences)” (p. 1648). When analyzing risks, one should determine: “What can go wrong?”, “What is the likelihood?” and “What are the consequences?” (Kaplan & Garrick, 1981). Transferring this general approach towards risks in the field of KM, the following definition is proposed: knowledge risk is a measure of the probability and severity of adverse effects of any activities engaging or related somehow to knowledge that can affect the functioning of an organization on any level.

Depending on its origin, the authors of this paper argue that knowledge risks can be classified into the following categories: human, technological and operational. Human knowledge risks are connected with an individual's personal, social, cultural and

psychological factors and thus human resources management. For example, the risk of knowledge hiding is related to the human dimension of knowledge risks. Technological knowledge risks result from the usage of various technologies, including information and communication technologies (ICT), but are not limited to those. Risks from this category may be the outcome of or initiated by, for example, the use of old technologies or hacker attacks. Finally, the operational category of knowledge risks embraces all the risks resulting from everyday operations and functioning of organizations, e.g. making alliances or mergers, outsourcing, applying wrong or obsolete knowledge in operations. Like any risk, knowledge risks should be managed, acknowledging that they cannot be eliminated.

In the following sections, the identified knowledge risks will be discussed.

- **Human knowledge risks**

In the following, knowledge hiding, knowledge hoarding, unlearning, forgetting, missing/inadequate competencies of organizational members will be discussed.

3. Resource

Resource risks represent less than one-third of the records in the PERIL database. There are three categories of resource risk: people, outsourcing, and money. People risks arise within the project team. Outsourcing risks result from the use of people and services outside the project team to perform critical project work. The third category, money, is something of an anomaly in the data, as very few of the problems reported were primarily about funding. Money is, however, a key factor in many of the people and outsourcing problems, and the effect of insufficient funding on projects has substantial impact on a project in many other ways. Resource planning is a useful tool for anticipating many of the people, outsourcing, and money risks. Inputs to the resource planning process include the project work breakdown structure (WBS), scope definition, activity descriptions, preliminary duration estimates, and the project schedule. Resource requirements planning can be done in a number of ways, using manual methods, histogram analysis, or computer tools.

On the basis of the preliminary schedule and assumptions about each project activity, you will need to determine the skills and staffing required for each activity. It is increasingly common, even for relatively small projects, to load this data into a computer scheduling tool. As early as possible, identify staffing for each activity using the names of each individual. While preliminary resource planning can be done with functions or roles, resource analysis based on names makes it more likely that estimates will be accurate and that staffing will be committed to the project as planned. Identify as a risk any required project staff members who cannot be named during project planning. For the project as a whole, also identify all holidays, scheduled time off, significant non project meetings, and other time that will not be available to the project. Do this for each person, as well, and identify any scheduling differences for different regions, countries,

and companies involved in the project. A computer scheduling tool is a good place to store calendar information, such as holidays, vacations, and any other important dates. If you do use a computer tool, enter all the calendar data into the database before you begin resource analysis.

Web Application Survey

The project of ours, STUDENT INFORMATION MANAGEMENT SYSTEM has come up with many functionalities for educational institutions to track the student progress, assignment, unassignment of teachers, uploading and viewing marks etc. It helps both student and guardian to keep track of student progress without visiting to the college. It also notifies student and guardian during the time of important events which are happening in institution. One more feature is guardian get alerted whenever student get failed in the exam or student not able to meet the expected attendance average. Student Information Management System is application software and which has intention to begin a conductive and direct interchanging the statistics in a secure platform to coalesce with students, faculties, parents and the college/school administration. The student information has the particulars (like register number sem, date-of-birth, sex, parent phone number, address, parent name, etc.) invade to the system by the faculties. All these particulars is stored in the database. SIRS application is trouble free to use in schools, colleges, universities, and any other educational institutions. It can be customized as per the need. It can be used in private and government educational institutions also. SIRS application is an internet based application we can login to the system from anywhere irrespective of geographical area it will give seamless navigation. The paper provides the particulars to carry out the performance, management and decision-making functions of enterprises or organizations. Enormous grow of students is caused to expand the functionality in the respective educational institutions. As student added to the educational system it is difficult to manage and track student details. To overcome difficulties we come up with this new approach student information management system with additional features. This new approach will provide fast processing, efficient student tracking, and produces desired result. This approach will allow students to save their personal details. It is more secure, reliable and easy to use. Attendance is part of any system to keep track of the particular person. It is mandatory process in educational system which directly reflects the student progress. In educational institutes attendance management is normally a manual process. There is enormous grow in the software industry which has privileged colleges to maintain the attendance system by using gadgets which is the best way. As we are using the smart phones we not require maintaining attendance register. It can be easily done in mobile application. Faculty will be going to take the attendance when class gets started. They will initially login to the system through mobile application. Once attendance has been taken successfully for the class it will sent to sever through GPRS. The faculties can also do the necessary functions like registering new students, deleting the information about a particular student, modifying the information regarding the student etc. The main intention of this process is to reduce the risk of manual efforts. It also reduces the time consumption. Also we want to give importance to reduce the paper wastage that daily happens. The model utilizes computer aided system. The model plays main role in an institution or in the college management. Initially, the system has developed with four layers based on the hierarchy such as Web display layer where application is deployed and displayed for end users. Business logic layer responsible for handling the functionality of the product. Data access layer is

responsible for viewing the data. Database layer responsible for storing the student data. In Database layer ER diagram has been designed to provide data normalization. The process provides complete information about student, faculties and educational institution. Third thing in this project is to allowing user based on their categories. The paper provides end user to seamless navigation to the application and ease of access. The model provides information management storing of student academic reports. This model consists of various functionalities like information about the courses available in the college starting from first grade to graduation. It also enables students to enroll to particular course through online, online fees payment, examination results, and also get notified when important events occur. All data stored and retrieved through the application is secure. So to achieve this we have developed a powerful web based secured interface application which supports all type of request which are coming from the students also which gathers and corrects all student information. To achieve this we have used similarity (Euclidean distance) algorithm. The results showed that the new information gathered by the SDS has the ability to fill the requirement and done the error correction in the traditional model. The papers will explains how it is playing an important role in the education domain. This system is provides seamless access through the web based application to access and manage different department or all over the organization. This system is used to mainly monitor the attendance for the university. Students are provided access to login to application and view the progress report and attendance report. This system is developed for an engineering college it will provide end users to maintain their data with minimum effort. Initially faculties/students get registered with the system once they finish registration process they can access the system as well as they are able to do the changes in the data. As per the requirement users has been granted with certain level permission to manage and track the student information. Either student or faculty can upload and copy the statistics from the database. Since it is a web based application which is accessible from any part of the world it has certain features like accessibility, easy to use. IOT, easy to manage etc. It is developed to suite the current environment which is rapidly growing in the student domain.

Theoretical Background

A project on Student Database Management system which was carried out by Anisha Mitra, Nisha Kumari, Monika Kumari, Shankhadeep Das, MCA students of Techno Main Salt Lake, Kolkata. Provides a simple interface for maintenance of student information. It can be used by any educational institute or colleges to maintain records of students easily. Achieving these objectives is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information can be very time consuming. All these problems are solved using this project. Throughout the project the focus has been on presenting information in an easy and intelligible manner. The project is very useful for those who want to know about Student Information Management Systems and want to develop software/websites based on the same concept. The project provides facilities like on line registration and profile creation of student thus reducing paper work and automating the records generation process in an educational institution. The goal of this chapter is to explain the theory of database management system development which will be applied in the development of student database management system for any University. The following issues will be presented respectively.

This will give an insight into the project area and help to get information that will enhance the development of the student database management system.

a) Integrated Information System

Integrated Information Systems offer users with a unified view of heterogeneous data sources. To provide a single consistent result for every object represented in these data sources, data fusion is concerned with resolving data inconsistencies present in the sources. Querying the heterogeneous data sources, combining the results, and presenting them to the user is performed by the integration system.

When multiple sources are to be integrated into a single and consistent view, at least the following three steps need to be performed:

Uncertainty is a conflict between a non-null value and one or more null values that are used to describe the same property of an object. Usually it occurs as a result of missing information. Null values present in tables can have different meanings. Three interpretations of null values can be distinguished as follows:

The other type of conflicts is contradiction. It is the conflict between two or more different non-null values that are used to describe the same property of an object. An example is the case if two or more data sources provide two or more different values for the same attribute on the same object.

One good reason for regarding uncertainty as a special case of conflict is the fact that it is easier to deal with uncertainty than contradiction.

Several forms and reports are used in day to day processing of documents. A database can integrate these several components hence resulting in improved and more efficient operations (Greenwald *et al.*, 2005; Connolly and Begg, 2004).

b) Database and Database Management System

A database is an integrated collection of data, usually so large that it has to be stored

on secondary storage devices such as disks or tapes. This data can be maintained as a collection of operating system files, or stored in a DBMS (database management system).

A Database Management System (DBMS) is computer software designed for the purpose of managing databases based on a variety of data models. A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. DBMS are categorized according to their data structures or type, sometime DBMS is also known as Database Manager. It is a set of prewritten programs that are used to store, update and retrieve a Database.

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system. Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis.

Database servers are specially designed computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with RAID disk arrays used for stable storage. Connected to one or more servers via a high speed channel, hardware database accelerators are also used in large volume transaction processing environments.

Advantages of DBMS

- Improved strategic use of corporate data
- Reduced complexity of the organization's information systems environment
- Reduced data redundancy and inconsistency
- Enhanced data integrity
- Application-data independence
- Improved security
- Reduced application development and maintenance costs
- Improved flexibility of information systems
- Increased access and availability of data and information
- Logical & Physical data independence
- Concurrent access anomalies.
- Facilitate atomicity problem.
- Provides central control on the system through DBA.

c) Student Records/Data Management

According to student records manual prepared by University of South Florida (office of the registrar) the creation and maintenance of records relating to the students of an institution are essential to:

- managing the relationship between the institution and the student;
- providing support and other services and facilities to the student;
- controlling the student's academic progress and measuring their achievement, both at the institution and subsequently;
- providing support to the student after they leave the institution.

- In addition, student records contain data which the institution can aggregate and analyze to inform future strategy, planning and service provision.
- A student is an individual who is registered for a university credit course or program.

A student record/data contains information directly related to a student, which means that the record is personally identifiable. Personal identifiers that relate a record to a student include student name, student ID, student address, parent/family member names, and a list of personal characteristics. Student records could be maintained in multiple media including handwriting, print, microfilm/fiche, computer's main memory, magnetic tape, cassette, disk or diskette.

Student records/data may be presented by the student, submitted on behalf of the student, or created by the University. These records are used to assist offices in their support of basic institutional objectives and to document student progress and achievement in the educational process of the University.

Educational institutions and agencies are required to conform to fair information practices. This means that persons who are subjects of data systems (i.e., students at an institution) must:

- be informed of the existence of such systems
- have identified for them what data about them are on record,
- be given assurances that such data are used only for intended purposes
- be given the opportunity to request an amendment or correction to their record and
- be certain that those responsible for data systems take reasonable precautions to prevent misuse of the data.

d) Web Technology

The web is a hypermedia-based system that provides a means of browsing information on the Internet in a non-sequential way using hyperlinks. The web is a compelling platform for the delivery and dissemination of data-centric, interactive applications. The web's ubiquity provides global application availability to both users and organizations. Because the architecture of the web has been designed to be platform-independent, it has the potential to significantly lower deployment and training cost. Organizations are now rapidly building new database applications or reengineering existing ones to take full advantage of the web as a strategic platform for implementing innovative business solutions, in effect becoming web-centric organization.

The World Wide Web (web for short) provides a simple point and click means of exploring the immense volume of pages of information residing on the Internet (Berners-Lee 1992; Berners-Lee et al.. 1994). Information on the web is presented on the web pages which appear as a collection of text, graphic, pictures, sound and video.

e) Conclusion

Finally, since the data generated in day to day transactions by School of Public Health increase geometrically according to the registrar, it is worthwhile and holistic to develop robust student database management system for the school to hold the large amount of data that is generated. The proposed system, SDMS, should be able

to stand the test of time because student records should be kept as long as is necessary to:

- 1) Fulfill and discharge the contractual obligations established between the institution and the student, including the completion of any non-academic disciplinary action;
- 2) Provides information on the academic career and achievements of the student to employers, licensing/regulatory bodies and other organizations, as well as to the student as part of their lifelong learning record.

The information gathered whilst reviewing existing literature on student information management system has been very useful in the development of the proposed system. The project when completed will provide an efficient way to store and organize data than spreadsheet. It will also serve as a centralized facility that can easily be modified and quickly shared among multiple users. The proposed system will have a well-designed interface that allows users to interact with the system via internet connection and a web browser.

Software Requirement Specifications

SOFTWARE REQUIREMENTS:-

NAME OF COMPONENT	SPECIFICATION
Operating System	Windows 10
Language	Java
Database	MySQL, MySQL workbench
Browser	Google Chrome
Web Server	Tomcat 9.0
Software Development Kit	Java Development kit
Scripting language	JavaScript
Web Development Platform	HTML,CSS,jQuery

HARDWARE REQUIREMENTS:-

NAME OF COMPONENT	SPECIFICATION
Processor	Intel i5 8th Generation
RAM	8 GB
Hard Disk	512 GB SSD

Design

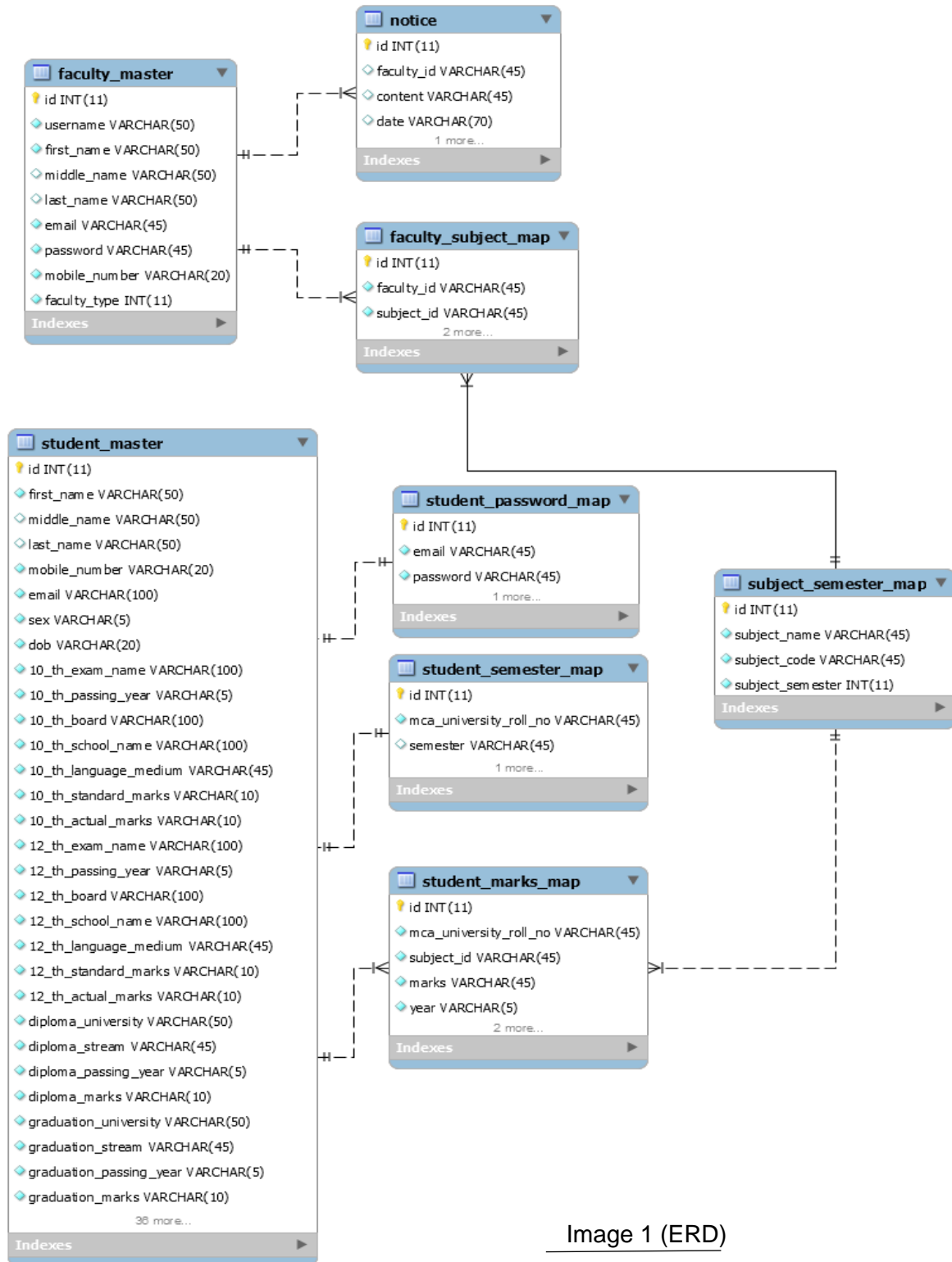


Image 1 (ERD)



Image 2 (controller class diagram)



Image 3 (model class diagram)

Coding Standard Followed and Assumptions

Some coding standards are followed in this project like,

- The name of all the Text Box starts with text,
- The name of all the Button starts with btn
- The name of all the Label starts with lbl,

For example, in the Login Page the coding standard is as follows:

Frontend:index.html

```
<html>
<head>
  <title>LOGIN</title>

  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="loginbox">
    
    <h1>Login Here</h1>

    <p> Username/email</p>
    <input type="text" name="username_email" id="username_email"
      placeholder="Enter Username/email"/>
    <p>Password</p>
    <input type="password" name="pass" id="pass"
      placeholder="Enter Password"/>
    <input id="btn1" type="submit" value="Login">
  </div>
  <script
    src="https://code.jquery.com/jquery-3.4.1.js"></script>
  <script type="text/javascript">

    $(function () {
      $("#btn1").on('click', function () {
        let admin = $("#username_email").val();
        let pass = $("#pass").val();

        $.ajax({
          type: "POST",
```

```

        url: "http://localhost:8080/Student_Management_System/api
            /auth" ,
        data: {username_email:admin,pass:pass} ,
        //dataType:JSON,
        contentType: "application/x-www-form-urlencoded" ,

        success: function (response) {
            console.log(response);
            if (response.response.status == "success") {
                //alert("Success");
                if (response.response.user_type == "admin")
                    window.location.href = "admin.html";
                if (response.response.user_type == "student")
                    window.location.href = "student.html";
            }else{
                alert("failed");
            }
        },
        failure: function (response) {
            alert("Ajax Failed");
        }
    });
});
});
</script>

</body>
</html>

```

Style.css

```
body{
  margin:0;
  padding:0;
  background: url('background4.jpg');
  background-size: cover;
  background-position: center;
  font-family: sans-serif;
}
.loginbox{
  width: 320px;
  height:420px;
  background:#000;
  color:#fff;
  top:50%;
  left:50%;
  position:absolute;
  transform: translate(-50%,-50%);
  box-sizing:border-box;
  padding: 70px 30px;
}
.avatar{
  width:100px;
  height:100px;
  border-radius:50%;
  position: absolute;
  top:-50px;
  left:calc(50% - 50px);
}
h1{
  margin: 0;
  padding: 0 0 20px;
  text-align: center;
  font-size: 22px;
}
.loginbox p{
  margin: 0;
  padding: 0;
  font-weight: bold;
}
.loginbox input{
  width: 100%;
  margin-bottom: 20px;
}
```

```
.loginbox .input[type="text"],input[type="password"]
{
    border: none;
    border-bottom: 1px solid #fff;
    background:transparent;
    outline: none;
    height: 40px;
    color: #fff;
    font-size: 16px;
}
.loginbox input[type="submit"]
{
    border:none;
    outline: none;
    height:none;
    background:#fb2525;
    color:#fff;
    font-size: 18px;
    border-radius: 20px;
}
.loginbox input[type="submit"]:hover{
    cursor: pointer;
    background:#ffc107;
    color:#000;
}
```

Result Set Analysis

This section lists the results that were produced by running the test cases. Table 4 lists the test cases that were used while testing the interface along with the expected result and the actual results for each test case.

List of test case result

Test case id	Expected Result	Actual result
TC_VC_SIGNIN_001	PASS	PASS
TC_VC_SIGNIN_002	FAIL	FAIL
TC_VC_SIGNIN_003	FAIL	FAIL
TC_VC_SIGNIN_004	FAIL	FAIL
TC_VC_SIGNIN_005	PASS	PASS
TC_VC_SIGNIN_006	FAIL	FAIL
TC_VC_SIGNIN_007	FAIL	FAIL
TC_VC_SIGNIN_008	FAIL	FAIL

Testing

SYSTEM TESTING

Testing is the process of executing a program with the intent of finding any errors. Testing is vital to the success of the system. Without proper testing hidden errors will surface after sometime of use and perhaps irreversible damage has been done to valuable data. A series of tests like responsiveness, its value, stress and security are performed before the system is ready for user acceptance testing. System testing follows the logical conclusion that is all part of the system are tested and found to be working properly under all kinds of situations, and then the system is achieving its goal of processing the data perfectly according to user rules and requirements. The different types of testing are given below.

- **Unit testing**
- **Validation testing**
- **Output testing**
- **User acceptance testing**

UNIT TESTING

It involves the basic testing of a piece of code, the size of which is often undefined in practice. During the unit testing it is tested to know whether that particular unit in the proper manner as expecting, if not appropriate modifications are applied to get proper outputs.

In our project, we need to run the whole code under localhost:8080/SMS/ if it shows connected to sql server then we can assure that the proper block of code is running properly.

VALIDATION TESTING

At the conclusion of the black box testing, s/w is completely assembled as a package. Interfacing errors have been uncovered and correct and final series of test, i.e. validation begins. Validation test can be defined with a simple definition that validation succeeds when the software functions in a manner that can be reasonably accepted by the customer. Thus we have successfully done validation testing.

OUTPUT TESTING

After performing the validation testing, next is the output testing of the proposed system. The system cannot be useful if it not produce required output. The output displayed by the system under consideration will be compared with the user needs. Here the output format is considered in 2 ways, screen format and printed format. The o/p format on the screen is found to be correct as the format was defined in the design phase according to user needs. As for the hardcopy the o/p comes according to the specification requested by the user. Here the o/p testing does not result in any correction in the system. Thus we have successfully done output testing.

Here in our project it means that the output received from the particular project can be viewed properly by the user and the output we receive is correct.

USER ACCEPTANCE TESTING

It is the key for success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with perspective system at the time of development and making changes whenever required. This is done with regarding to the input screen design and output screen design. Thus we have successfully done user acceptance testing.

SYSTEM IMPLEMENTATION

System implementation is the final phase that is putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively. The system is implemented only after thorough checking is done and it is found working according to the specifications.

System implementation is the final phase. i.e., putting the utility into action.

Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively.

Implementation is the state in the project where theoretical design turned into working system. The implementation stage is a system project in its own right. It involves careful planning, design, investigation of the current system and constraints on implementation, design of methods to achieve change over and evolution method. Once the planning has been completed the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training user staff.

The major implementation procedures are:-

- Test plans
- Training
- Equipment installation
- Conversion

4.2.1:Test plans

The implementation of a computer based system requires that the test data can be prepared and the system and its elements be tested in a structured manner.

4.2.2:Training

The purpose of training is to ensure that all the personnel who are to be associated with the computer based system possesses necessary knowledge skills.

4.2.3:Equipment installation

Equipment vendors can provide specifications for equipment installation. They usually work with projects equipment installation team is planning for adequate space, power and light, and a suitable environment. After a suitable site has been completed, the computer equipment can be installed.

4.2.4:Conversion

It is the processes of performing all of the operations that result directly in turnover of the new system to the user. Conversion has two parts:-

1. The creation of a conversion plan at the start of the development phase and the implementation of the plan throughout the development phase.
2. The creation of a system change over plan at the end of the development phase and the implementation of the plan at the beginning of operation phase.

PROJECT NAME	STUDENT INFORMATION MANAGEMENT SYSTEM
CREATED BY	ANISHA,NISHA,MONIKA, SANKHADEEP
CREATION DATE	
REVIEWED BY	PRADOSH BANDYOPADHYAY
REVIEW DATE	

TEST SCENARIO ID	TEST SCENARIO DESCRIPTION	TEST CASE ID	TEST CASE DESCRIPTION	TEST STEPS	TEST DATA	POST CONDITIONS	EXPECTED RESULT	ACTUAL RESULT	STATUS	EXECUTED BY	EXECUTED DATE	COMMENTS (IF ANY)
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_001	1.Enter valid student email 2.Enter valid student username/email id and valid password 3.Click on login button	1.Id/username 2.password 3.password Pass@123	username: xxxxxxxxxxxx	Student will be redirected to the student panel to perform tasks	successful login	pass		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_002	1.Enter invalid email 2.Enter an invalid student email 3.Click on login button	1.Id/username 2.Id/username and valid password 3.password	username: xxxxxx password Pass@124	FAILED	a pop up message will show in the box, FAILED	fail		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_003	1.Enter valid email 2.Enter an invalid password 3.Click on login button	1.Id/username 2.Id/username and invalid password 3.password	username: xxxxxxxxxxxx password: Pas125	FAILED	a pop up message will show in the box, FAILED	fail		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_004	1.Enter invalid email 2.Enter an invalid password 3.Click on login button	1.Id/username 2.Id/username and invalid password 3.password	username: xxxxxxxx password: Pass26	FAILED	a pop up message will show in the box, FAILED	fail		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_005	1.Enter valid Admin Id 2.Enter valid admin password 3.Click on login button	1.Id/username 2.password 3.password:admin	username: admin password:admin	Admin can directly go to the page where he can do his tasks	successful login	pass		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_006	1.Enter valid Admin Id 2.Enter invalid admin password 3.Click on login button	1.Id/username 2.password 3.password:admin	username: admin password: Pass@128	FAILED	a pop up message will show in the box, FAILED	fail		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_007	1.Enter invalid Admin Id 2.Enter valid admin password 3.Click on login button	1.Id/username 2.password 3.password: admin	username: xxxxxxxxxxxx password: admin	FAILED	a pop up message will show in the box, FAILED	fail		ANISHA,MONIKA,NISHA,SANKHA		
TS_VC_001	Verify whether a valid user or not	TC_VC_SIGNIN_008	1.Enter invalid Admin Id 2.Enter invalid admin password 3.Click on login button	1.Id/username 2.password 3.password	username: xxxxx password Pass@130	FAILED	a pop up message will show in the box, FAILED	fail		ANISHA,MONIKA,NISHA,SANKHA		

Future Scope of the Project

Such a web based platform has immense potential for future growth not only for students but also for faculty as well as department concerned in improving upon their efficiency of work with no or less burden. Some of the areas where future scope lies and if implemented would help a great deal to the students at large.

- In the Future, Student can also download and upload notes.
- Also, it would be possible for students to watch lectures in the form of video content which would facilitate easy access to study material.
- Further enhancement would help in Statistics tracking and analytics of student's performance according to his/her result.
- In future, the head of the department can also do their contribution like seeing all student as well as teachers performance also can main record and student data.
- They can also search students by their records.

Conclusion

This application is automating the existing manual system. This is a paperless work and it reduces the man power. It will always provide accurate information. All years together gathered information can be saved and accessed using application at any time. This system is essential in institute. The entire administrator, HOD, faculty, student can get the required information without delay. As this application is only made for the general purpose it can be generalize to big scale such as use in colleges, university and even in distance education can be benefitted from this application as this app can provide all information according to the ward without direct contacting the staff. This student information management system can be implemented into N number of schools and colleges without any problem. In future we can also add the features like paying the fees of the ward.

This system works on very minimum requirements and in the testing, it is found that this student information management system does not have any errors till now and works perfectly.

Module Specification

- **Admin login**

Admin will create the login ID for each faculty of different departments, assign or unassign teacher, update details and posting notices etc.

- **Student login**

In this module student can view their details, update their information like contact number, address and they can also view notices issued by institute.

All the personal details of the students can be obtained in just one mouse click.

References

- 1) www.google.com
- 2) M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", IEEE Transactions on Signal Processing, Vol. 50, No. 2, pp:174-188, Feb' 2002.
- 3) R. N. Banavar, J. L. Speyer, "Properties of Risk-Sensitive Filters/Estimators", IEE Proceedings of Control Theory Application, Vol.145, No. 1, January 1998.
- 4) R. G. Brown, and P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering with Mat lab Exercises and Solutions, 3rd Edition, John Wiley & Sons, Inc., 1997.
- 5) Universal Description, Discovery and Integration, UDDI;<http://www.uddi.org>; October 5, 2007.

Appendix

Backend code of login page:-

Api.java

```
package base;

import static base.Controller.request;
import com.sha.ExcelHandler;
import com.tmsl.capability.AdminCapability;
import com.tmsl.capability.CoordinatorCapability;
import com.tmsl.capability.HodCapability;
import com.tmsl.capability.StudentCapability;
import com.tmsl.capability.TeacherCapability;
import com.tmsl.model.AuthModel;
import com.tmsl.pojo.Faculty;
import com.tmsl.pojo.Student;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.SQLException;
import java.util.AbstractMap;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 *
 * <pre>
 * 1. Path: /api/auth
 *    Post: a. (username or email) and b. pass
 * {
 *   "response": {
 *     "user": {
 *       "username": "admin",
```

```

*      "first_name": "Admin",
*      "middle_name": "",
*      "last_name": "",
*      "email": "admin@admin.com",
*      "password": "***",
*      "mobile_number": "-1",
*      "faculty_type": "1"
*    },
*    "status": "success"
*  },
*  "apipath": "/auth",
*  "responsetime": "Thu Sep 26 21:33:22 IST 2019"
* }

```

```

* </pre>

```

```

*/
@WebServlet(name = "Api", urlPatterns = {"/api/*"})

```

```

public class Api extends Controller {

    @Override
    protected void processRequest(HttpServletRequest _request, HttpServletResponse
response) throws ServletException, IOException {
        Controller.request = _request;
        boolean route = true;
        response.setContentType ("text/json;charset=UTF-8");
        response.setHeader("Access-Control-Allow-Origin", "");

        Object output = null;
        String requestApi = getRequestApi();
        String userLevel = loggedInFacultyType();
        /**
         * user level : 1 -> admin 2 -> hod 3 -> coordinator 4 -> teacher 5 ->
         * student
         */

        if (!requestApi.equals("/auth")) {
            if (isLoggedIn()) {
                switch (userLevel) {
                    case "1": // Admin
                        output = admin(requestApi);
                        break;
                    case "2": // Hod

```

```

        output = hod(requestApi);
        break;
    case "3": // Coordinator
        output = coordinator(requestApi);
        break;
    case "4": // Teacher
        output = teacher(requestApi);
        break;
    case "5": // Student
        output = student(requestApi);
        break;
    default:
        output = new AbstractMap.SimpleEntry<>("status", "invalid user level");
    }
} else {
    Map data = new HashMap<String, String>();
    data.put("status", "failed");
    data.put("info", "login required");
    output = data;
}
} else if (requestApi.equals("/auth")) {
    try {
        output = auth();
    } catch (SQLException ex) {
    }
} else {
    Map data = new HashMap<String, String>();
    data.put("status", "failed");
    data.put("info", "invalid request");
    output = data;
}

PrintWriter out = response.getWriter();
try {
    out.print(jsonOut(output));
} finally {
    out.close();
}
//Runtime.getRuntime().runFinalization();

}

private Map<String, Object> auth() throws SQLException {
    if (isLoggedIn()) {
        Map<String, Object> data = new HashMap<String, Object>();
        data.put("status", "failed");
    }
}

```



```

        data.put("info", "already logged in");
        return data;
    }

    AuthModel authModel = new AuthModel();

    Faculty faculty = new Faculty();
    Student student = new Student();
    String username = null;
    String email = null;
    String pass = null;

    //authModel.validate(faculty);
    username = gPost("username_email");
    pass = gPost("pass");

    Map<String, Object> output = new HashMap<String, Object>();
    output.put("status", "failed");

    if (!username.equals("") && !pass.equals("")) {
        String regex = "^(.+)@(.+)$"; //Email pattern

        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(username);

        if (matcher.matches()) {
            email = username;
            username = null;
        }

        if (email != null) {
            faculty.setEmail(email);
            student.setEmail(email);
        }
        if (username != null) {
            faculty.setUsername(username);
            student.setEmail("");
        }

        faculty.setPassword(pass);
        System.out.println(pass);
        student.setPassword(pass);

        try {
            faculty = authModel.validateFaculty(faculty);

```

```

if (faculty != null) {
    output.put("status", "success");
    if (faculty.getFaculty_type().equals("1")) {
        output.put("user_type", "admin");
    } else {
        switch(faculty.getFaculty_type()){
            case "2":
                output.put("user_type", "hod");
                break;
            case "3":
                output.put("user_type", "coordinator");
                break;
            case "4":
                output.put("user_type", "teacher");
                break;
        }
    }
}

HttpSession session = request.getSession();
session.setAttribute("login_status", "true");
session.setAttribute("email", faculty.getEmail());
session.setAttribute("faculty_type", faculty.getFaculty_type());
session.setAttribute("username", faculty.getUsername());
} else {
    /* If not a faculty login */
    //student = authModel.getStudent(student);
    student = authModel.validateStudent(student);
    if (student != null) {
        output.put("status", "success");
        output.put("user_type", "student");
        HttpSession session = request.getSession();
        session.setAttribute("login_status", "true");
        session.setAttribute("roll", student.getMca_university_roll_no());
        session.setAttribute("faculty_type", "5");
    }
}

} catch (SQLException ex) {

}

}

return output;
}

```

Authmodel.java

```
package com.tmsl.model;

import base.Model;
import com.tmsl.pojo.Faculty;
import com.tmsl.pojo.Student;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 *
 *
 */
public class AuthModel extends Model {

    /**
     * Validate Student by Semi-Complete Student Object
     * @param student
     * @return
     * @throws SQLException
     */
    public Student validateStudent (Student student) throws SQLException {
        Student output_student = new Student ();
        db.joinTables (new String [] {"student_master", "student_password_map"});
        db.select (new String [] {"A.mca_university_roll_no"});
        db.where ("A.email", "B.email");

        db.where ("A.email", student.getEmail ());
        db.where ("B.password", student.getPassword());

        //System.out.println (db.getQueryString());

        //return null;
        ResultSet rs = db.access ();

        If (rs.next ()) {
            output_student.setMca_university_roll_no(rs.getString("mca_university_roll_no"));
        } else {
            return null;
        }

        return output_student;
    }
}
```

```

/**
 * Validate faculty by Semi-Complete faculty Object
 * @param faculty
 * @return
 * @throws SQLException
 */
public Faculty validateFaculty (Faculty faculty) throws SQLException {
    Faculty output_faculty = new Faculty();

    db.selectTable ("faculty_master");
    db.select (new String [] {"*"});

    if (faculty.getPassword().trim ()!= "") {
        db.where ("password", faculty.getPassword ().trim ());
        if (faculty != null && faculty.getEmail().trim() != "") {
            db.where ("email", faculty.getEmail ().trim ());
        } else if (faculty!= null && faculty.getUsername ().trim ()!= "") {
            db.where ("username", faculty.getUsername ().trim ());
        }
    } else {
        return null;
    }

    ResultSet rs = db.access();
    if (rs.next()) {
        output_faculty.setUsername (rs.getString ("username").trim());
        output_faculty.setFirst_name (rs.getString("first_name").trim());
        if (rs.getString("middle_name") != null) {
            output_faculty.setMiddle_name (rs.getString ("middle_name").trim());
        }
        if (rs.getString("last_name") != null) {
            output_faculty.setLast_name (rs.getString ("last_name").trim());
        }
        output_faculty.setEmail (rs.getString ("email").trim());
        output_faculty.setPassword(rs.getString("password").trim());
        if (rs.getString("mobile_number")!= null) {
            output_faculty.setMobile_number(rs.getString("mobile_number").trim());
        }
        output_faculty.setFaculty_type(rs.getString("faculty_type").trim());
    } else {
        return null;
    }

    return output_faculty;
}
}

```

Frontend code of login page

Index.html

```
<html>
<head>
  <title>LOGIN</title>

  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="loginbox">
    
    <h1>Login Here</h1>

    <p> Username/email</p>
    <input type="text" name="username_email" id="username_email" placeholder
="Enter Username/email"/>
    <p>Password</p>
    <input type="password" name="pass" id="pass" placeholder="Enter Passwor
d"/>

    <input id="btn1" type="submit" value="Login">
  </div>
  <script
    src="https://code.jquery.com/jquery-3.4.1.js"
    integrity="sha256-
WpOohJOqMqqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU="
    crossorigin="anonymous"></script>
  <script type="text/javascript">

    $(function () {
      $("#btn1").on('click', function () {
        let admin = $("#username_email").val();
        let pass = $("#pass").val();

        $.ajax({
          type: "POST",
          url: "http://localhost:8080/Student_Management_System/api/auth" ,
          data: {username_email:admin,pass:pass} ,
          //dataType:JSON,
          contentType: "application/x-www-form-urlencoded" ,

          success: function (response) {
            console.log(response);
            if (response.response.status == "success") {
```

```

        //alert("Success");
        if (response.response.user_type == "admin")
            window.location.href = "admin.html";
        if (response.response.user_type == "student")
            window.location.href = "student.html";
        }else{
            alert("failed");
        }
    },
    failure: function (response) {
        alert("Ajax Failed");
    }
});
});
});
</script>

</body>
</html>

```

Style.css

```

body{
    margin:0;
    padding:0;
    background: url('background4.jpg');
    background-size: cover;
    background-position: center;
    font-family: sans-serif;
}
.loginbox{
    width: 320px;
    height:420px;
    background:#000;
    color:#fff;
    top:50%;
    left:50%;
    position:absolute;
    transform: translate(-50%,-50%);
    box-sizing:border-box;
    padding: 70px 30px;
}
.avatar{
    width:100px;
    height:100px;
    border-radius:50%;
    position: absolute;

```

```

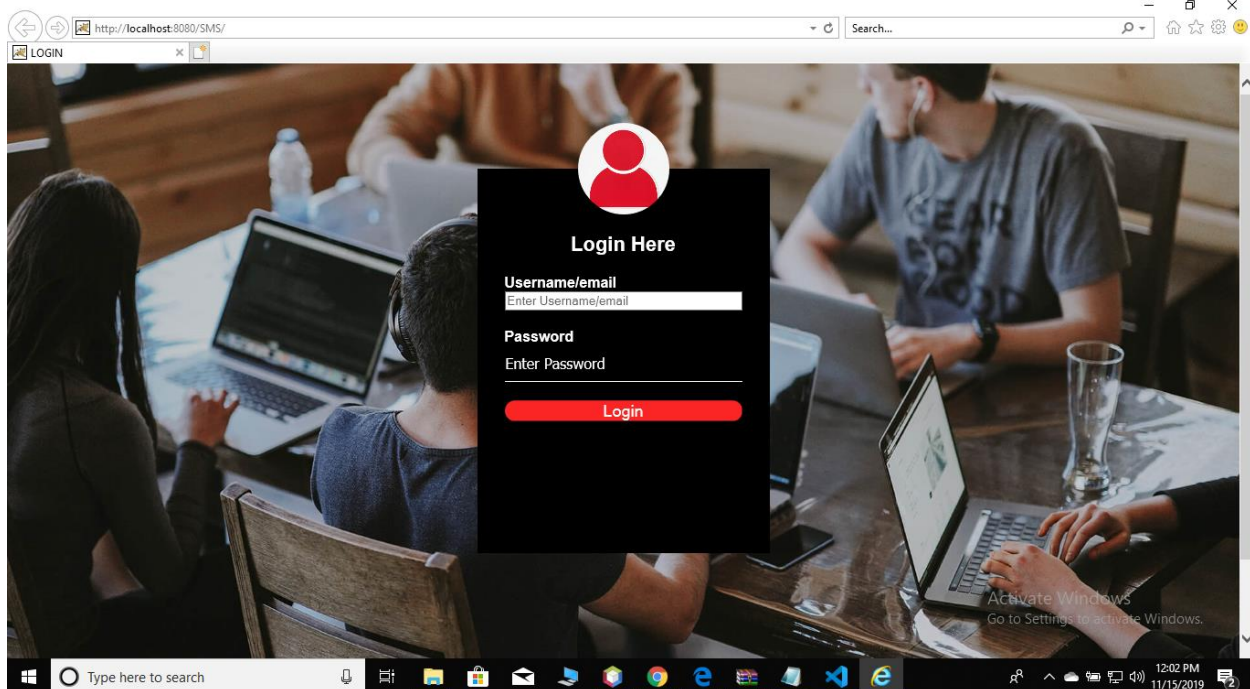
    top:-50px;
    left:calc(50% - 50px);
}
h1{
    margin: 0;
    padding: 0 0 20px;
    text-align: center;
    font-size: 22px;
}
.loginbox p{
    margin: 0;
    padding: 0;
    font-weight: bold;
}
.loginbox input{
    width: 100%;
    margin-bottom: 20px;
}
.loginbox .input[type="text"],input[type="password"]
{
    border: none;
    border-bottom: 1px solid #fff;
    background:transparent;
    outline: none;
    height: 40px;
    color: #fff;
    font-size: 16px;
}
.loginbox input[type="submit"]
{
    border:none;
    outline: none;
    height:none;
    background:#fb2525;
    color:#fff;
    font-size: 18px;
    border-radius: 20px;
}
.loginbox input[type="submit"]:hover{
    cursor: pointer;
    background:#ffc107;
    color:#000;
}

```

SCREEN LAYOUT DESIGN WITH SCREENSHOTS

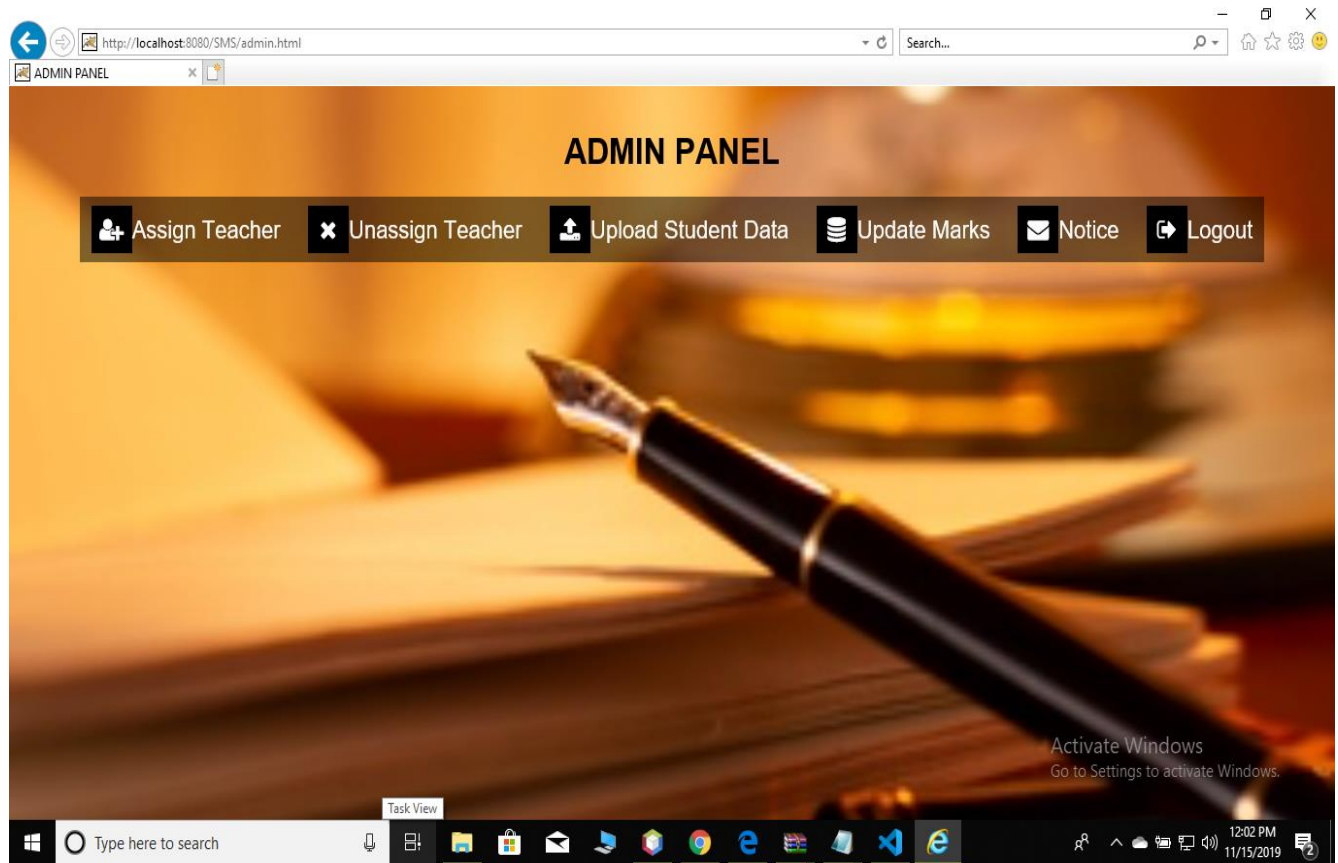
SCREEN DESIGN LAYOUT CONSTRAINS FOR LOGIN PAGE :

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Login page	Background4.jpg		sans-serif	22px	No validation	Students can login also admins can



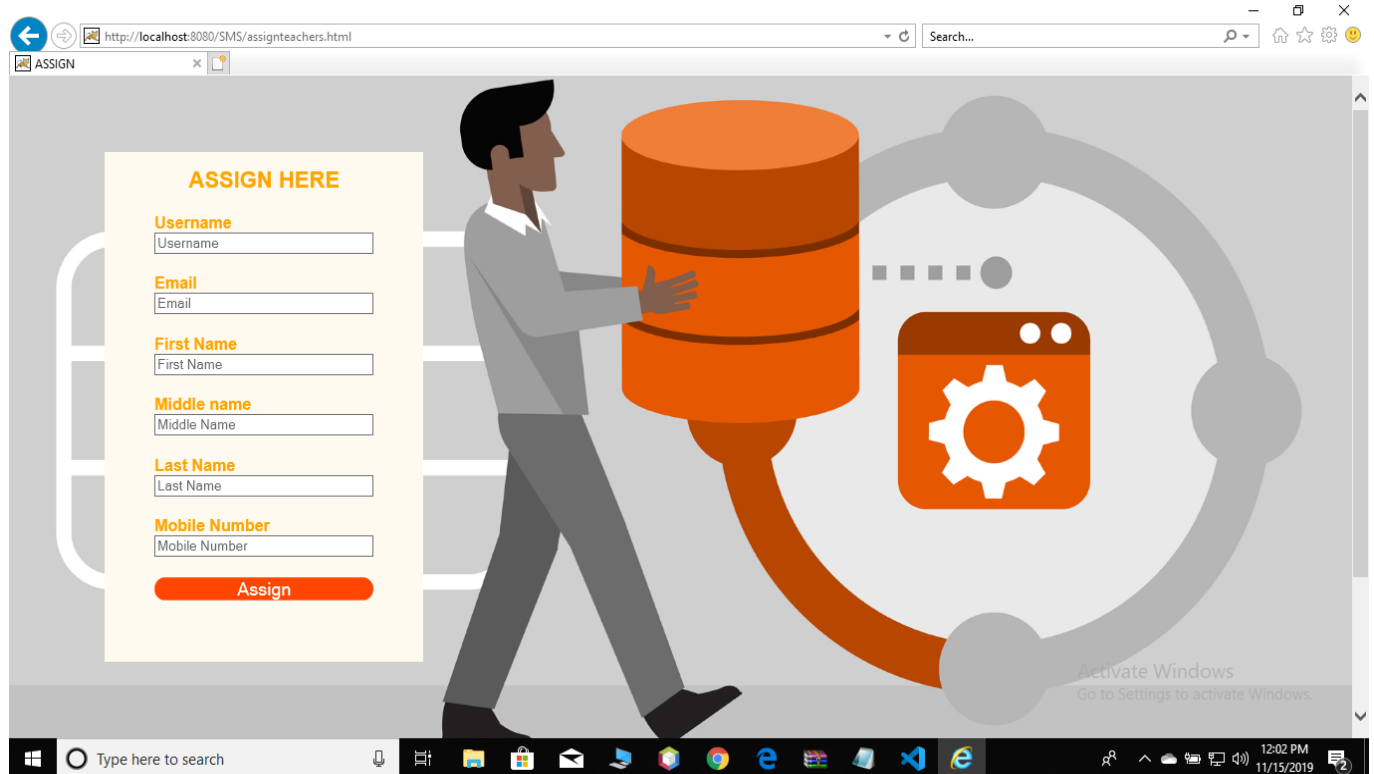
SCREEN DESIGN LAYOUT CONSTRAINS FOR ADMIN PAGE :

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Admin page	Bg2.jpg		sans-serif	22px	No validation	Admin can assign or unassign teacher , upload student data and marks,add notice.



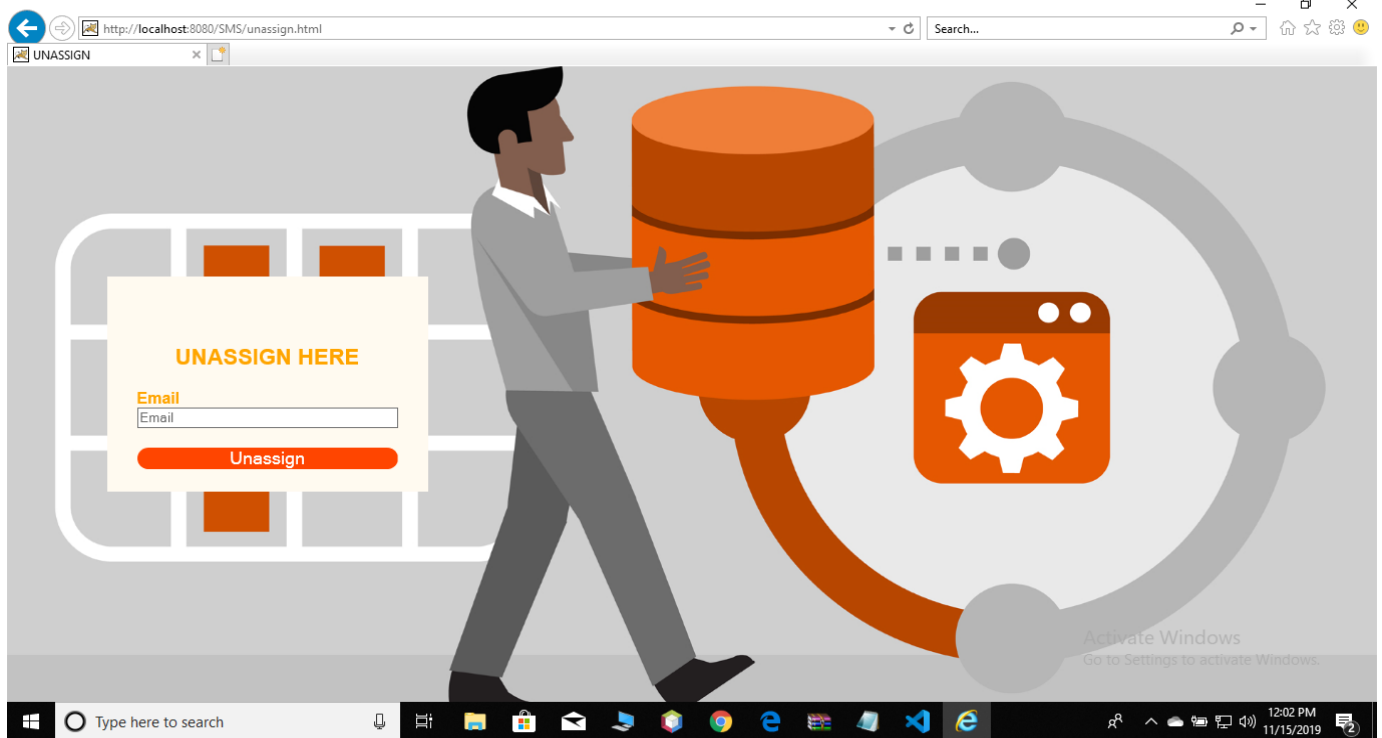
SCREEN DESIGN LAYOUT CONSTRAINS FOR ASSIGN TEACHER PAGE :

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Assign teacher page	bg4.jpg		sans-serif	22px	First name,middle name and last name should be characters only and first and last name cannot be empty. Mobile no. should be of 10 digits.	Admin can add teacher.



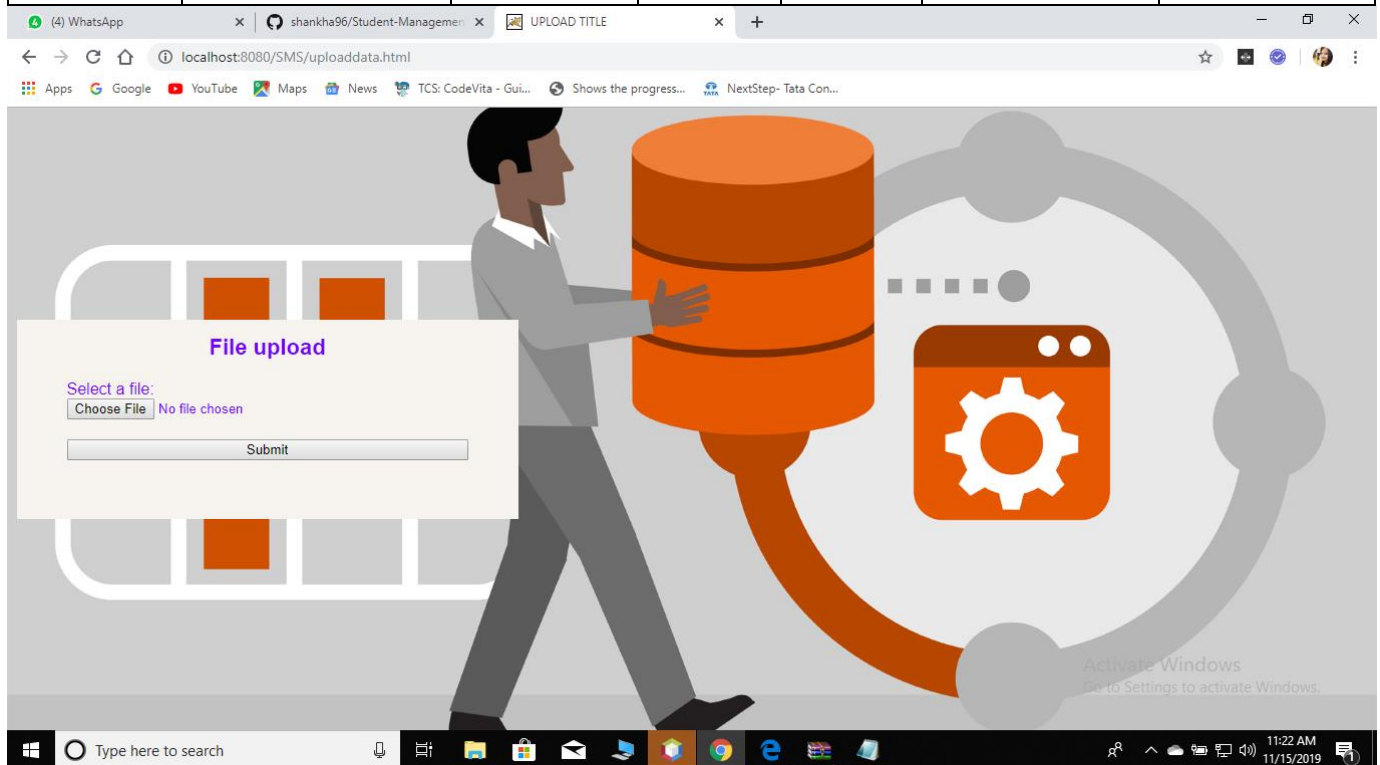
SCREEN DESIGN LAYOUT CONSTRAINS FOR UNASSIGN TEACHER PAGE :

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Unassign teacher page	bg4.jpg		sans-serif	22px	No validation	Admin can remove teacher.



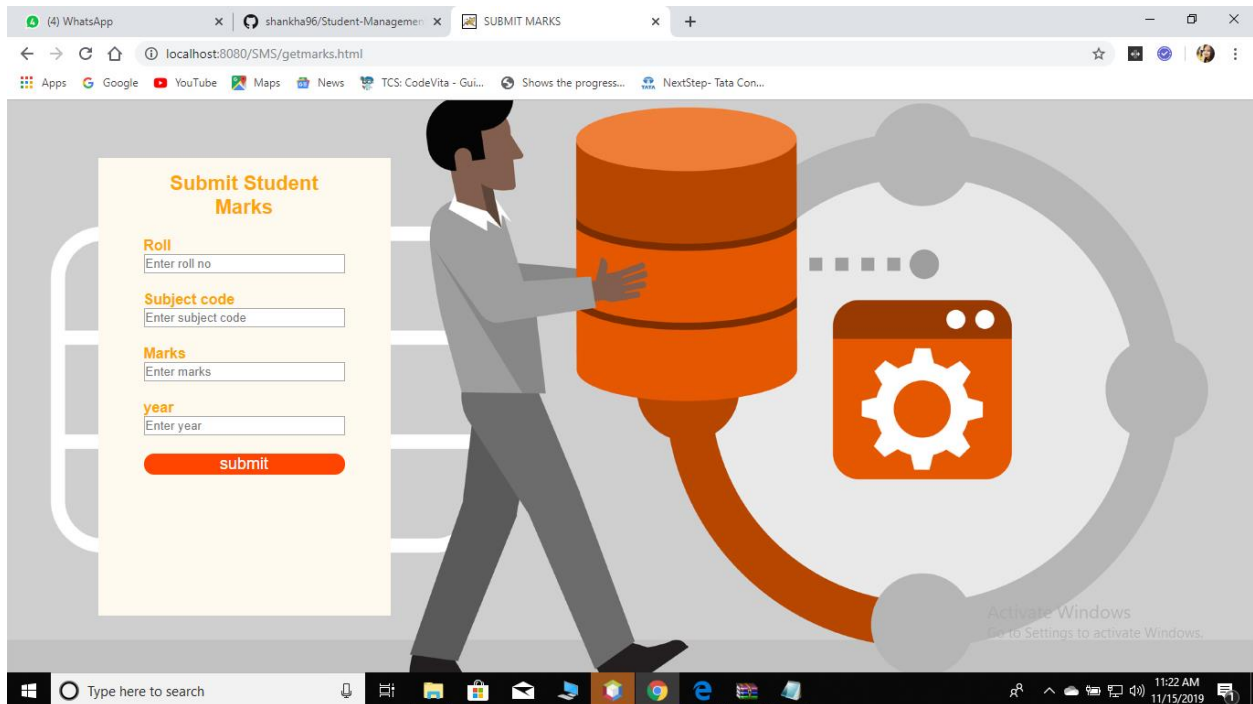
SCREEN DESIGN LAYOUT CONSTRAINS FOR UPLOAD STUDENT DATA :

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Upload student data page	bg4.jpg		sans-serif	22px	No validation	Admin can upload student data.



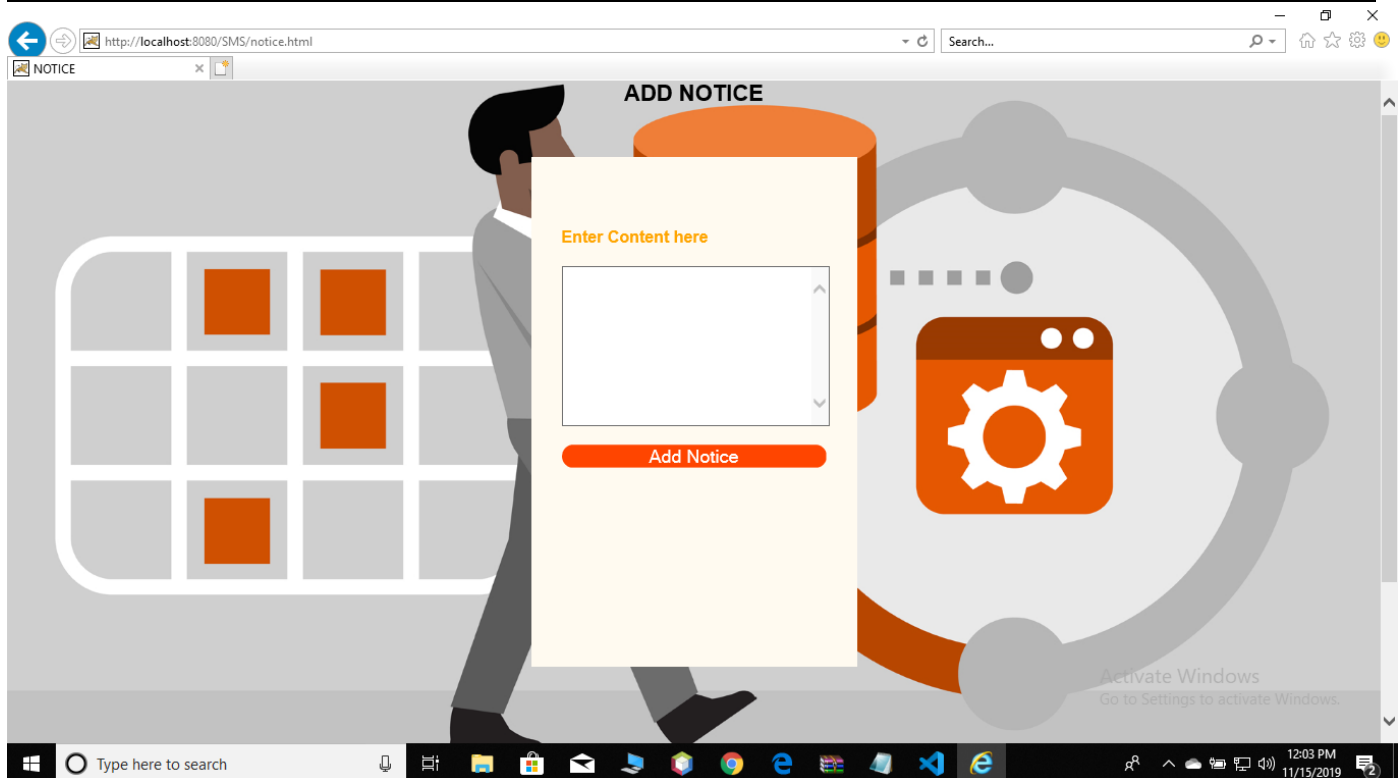
SCREEN DESIGN LAYOUT CONSTRAINS FOR UPLOAD STUDENT MARKS:

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Upload student marks	bg4.jpg		sans-serif	22px	No validation	Admin can upload marks of the students of their respective subject.



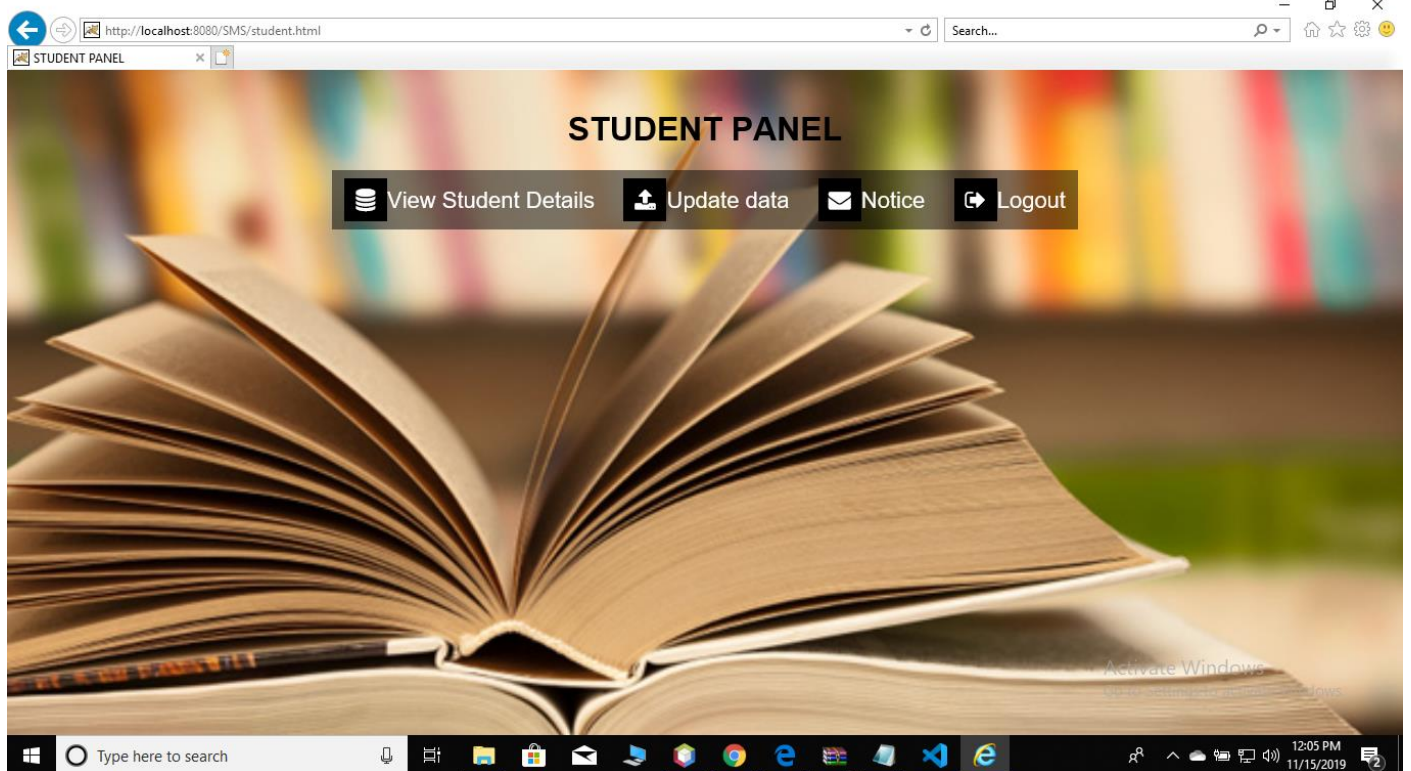
SCREEN DESIGN LAYOUT CONSTRAINS FOR UPLOAD NOTICE:

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Upload notices	bg4.jpg		sans-serif	22px	No validation	Admin can upload notices.



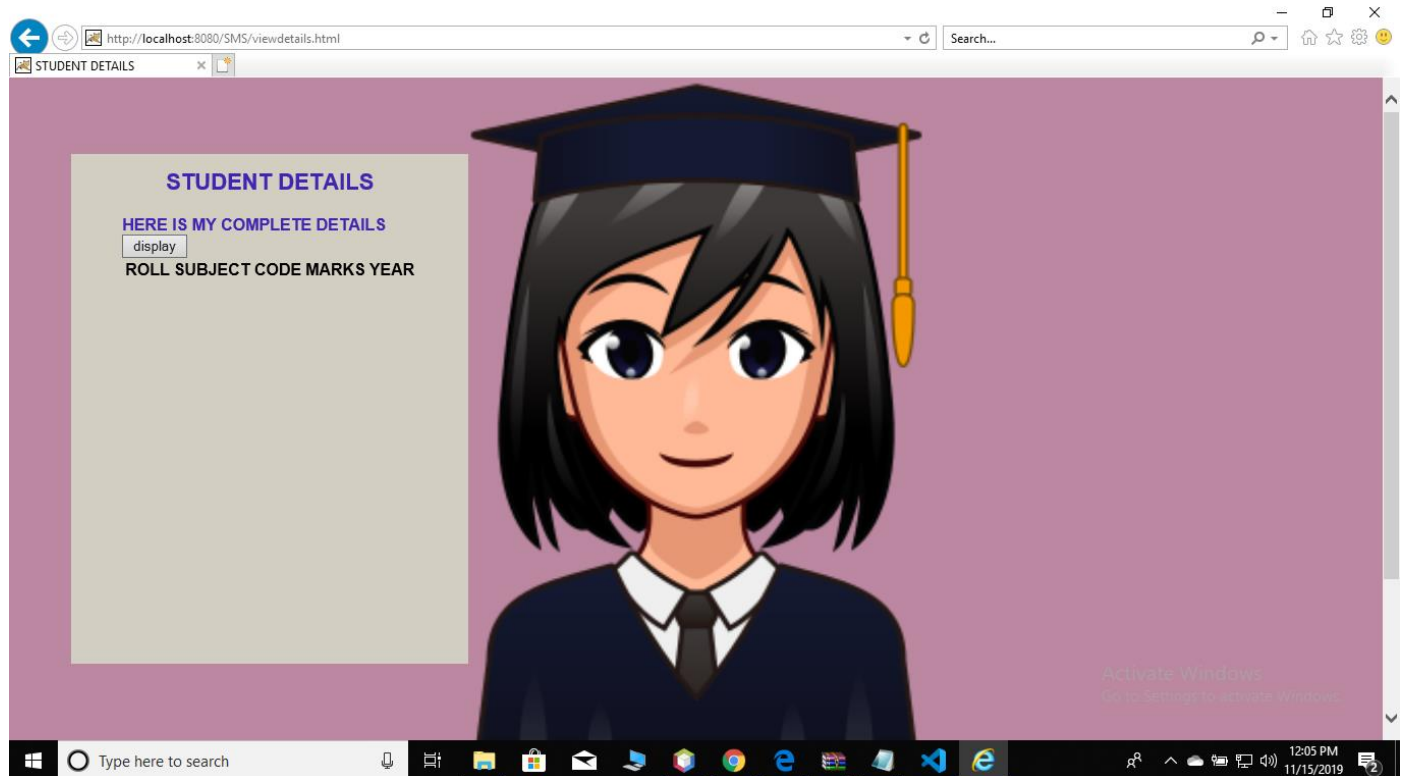
SCREEN DESIGN LAYOUT CONSTRAINS FOR VIEW STUDENT PANEL:

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Student panel	Bg3.jpg		sans-serif	22px	No validation	Students can view their details including marks, they can also update if any error occur. They can view notice too.



SCREEN DESIGN LAYOUT CONSTRAINS FOR VIEW STUDENT DETAILS:

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Upload Student details	Student.png		sans-serif	22px	No validation	Students can view their details including marks, roll number, year, subject id .



SCREEN DESIGN LAYOUT CONSTRAINS FOR UPDATE STUDENT DETAILS:

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
Update Student details	Bg5.png		sans-serif	22px	No validation	Students can update their respective fields where the errors occur.

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/SMS/update.html`. The browser tab is titled 'UPDATION FORM'. The main content area features a dark blue background with a faint image of a laptop keyboard. Overlaid on this is a large, semi-transparent white box containing the text 'Updating!' in a large, white, sans-serif font. Below the text is a horizontal progress bar that is approximately half-filled with a light blue color. In the bottom right corner of the main content area, there is a small white box with the text 'Activate Windows' and 'Go to Settings to activate Windows.'.

On the left side of the main content area, there is a white rectangular box with a yellow background, titled 'UPDATE HERE' in bold black text. This box contains several input fields with labels and placeholder text:

- date of birth:
- email:
- 10_th_exam_name:
- father's-name:
- diploma_stream:
- mobile-number:

At the bottom of this white box is a green button with the text 'UPDATE HERE' in white.

The browser's taskbar at the bottom shows the Windows logo, a search bar with the text 'Type here to search', and several application icons including File Explorer, Mail, and various web browsers. The system clock in the bottom right corner shows the time as 12:05 PM on 11/15/2019.

SCREEN DESIGN LAYOUT CONSTRAINS FOR VIEW NOTICES:

COMPONENT TYPE	BACKGROUND COLOR/IMAGE	SIZE MAXIMUM SIZE	FONT TYPE	FONT SIZE	TYPE OF VALIDATION INCLUDED	PURPOSE OF THE COMPONENTS
View Notices			sans-serif	22px	No validation	Students can view notices.

