**INTRODUCTION**

The “Student Management System” has been developed to override the problems prevailing in the practicing manual system. The software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly Student Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will helporganization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Fees, Student, Profiles, Exams, Courses. Every Student Management System has different Student needs. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. This system will ultimately allow you to better manage resources.

**Statement of the Problem**

E-Learning Fundamentals provides the basic knowledge of necessary to tackle everything from early concepts of

e-learning down to its execution. Throughout, you'll find vignettes that bring concepts to life as well as checklists and

practical tools for designing and developing your first e-learning course.

E-Learning Fundamentals provides the basic knowledge of necessary to tackle everything from early concepts of

e-learning down to its execution. Throughout, you'll find vignettes that bring concepts to life as well as checklists and

practical tools for designing and developing your first e-learning course.

E-Learning Fundamentals provides the basic knowledge of necessary to tackle everything from early concepts of

e-learning down to its execution. Throughout, you'll find vignettes that bring concepts to life as well as checklists and

practical tools for designing and developing your first e-learning course

E-Learning Fundamentals provides the basic knowledge of necessary to tackle everything from early concepts of

e-learning down to its execution. Throughout, you'll find vignettes that bring concepts to life as well as checklists and

practical tools for designing and developing your first e-learning course

Using manual system to manage the students which are records all information and, in the book, or paper was causing the job of the teachers becomes more and troublesome. The record in the have a possibiseereey missing or destroy when happen any accident. While now already have the school use the computer to manage the student information, but both of the systems they use are already out-dated. The system they use all are standalone and separately, one system only has one function. This was cause teacher harder to use all the system on the same time. The separately system without connection with each other also cause the same data and information the needed key in every system. Besides that, the system that using also does not have the communicate Platform for the parents to communicate with the parents.

Student Management System is a web based application in which capturing and storing student’s record details locally, where hard copies of files for every student is kept in office shelves, this seemto be tiresome and time consuming in case the registrar is looking of a particular student document.The problems facing the current manual system are data redundancy, difficult to update and maintain,inconsistent data, insecurity, difficult to impose constraints on various data file and difficult tobackup. Therefore, because of these drawbacks that Student Record Management System has beendeveloped to address the problems catalogued above.

**OBJECTIVE**

The main objective of this project is to build a student database system that will store records ofstudents. It is purposed to reduce time spent on administrative tasks. The system is intended to acceptprocess, generate students reports and grades and transcripts accurately. The system is also intendedto provide better services to users, provide meaningful, consistent, and timely data and informationand finally promotes efficiency by converting paper processes to electronic form.

**SCOPE**

The project provides comprehensive Student Database System. The system is able to capture, validate, sort, classify, calculate, summarize, store and retrieve data. Student Database System store semester details, course details, department details and all the detailsof students including their background information, educational qualifications and personal details etc.

* Any college can use this system as it is not client centric.
* Deliver Electronic Workplace.
* Application Support & Maintenance after deployment to production.
* The Admin Module can be reused for projects as well which have many users with different rights.

**Software Requirements**

|  |  |
| --- | --- |
| Name of component | Specification |
| Operating System | Windows, Linux |
| Language | Java SE / EE |
| Database | ORACLE Server |
| Browser | Any of Mozilla, Opera, Chrome etc |
| Web Server | Tomcat 9 |
| Software Development Kit | Java JDK 9.0 or Above |
| Scripting Language Enable | JSP & Java Script |
| Database JDBC Driver | Type- 4 / Thin driver |
| Architecture | MVC (Model View Controller) |
| IDE | Eclipse Photon |
| Editor | Sublime |

**Hardware Requirements**

|  |  |
| --- | --- |
| Name of component | Specification |
| Processor | Pentium Dual-Core CPU or Above |
| RAM | 2 GB Or more |
| Hard disc | 20 GB or more |
| Monitor | 15” color monitor or advance |
| Keyboard | Any Keyboard |
| Mouse | Any mouse |
| Printer | In case of printing reports |

**Module description**

*Student have the following faciseereeies*

* Students can login themselves using their register no. along with their password.
* After logging in they can view records about them like, Academic records, Attendance records, Conduct records, Achievement records.
* The student can only view their records, they are not allowed to change the records.
* An invalid password is entered. The student can re-enter a password or terminate the use case.
* The students can also download the reports as a .pdf file.

*Teacher have the following faciseereeies*

* Teacher can perform several operations on the system like registration, login. He or she can also edit his or her profile.
* The existing teachers are giving his/her user id & password to access their accounts.
* An invalid password is entered. The teacher can re-enter a password or terminate the use case.
* The system informs the teacher that the username is invalid. The teacher can re-enter the username or terminate the use case.
* After logging in they can view all the records of any students they wish.
* The teachers can view records and as well as can changes to the records.
* The teacher can also download the reports of a students the wish as a .pdf file.

*Admin have the following operational powers*

* It provides the capabiseereey for the admin to verify different procedures. Admin can perform various types of operations like edit, update, delete, sending the mail etc.
* Admin perform the main activity like store the information of the students , sending the mail to students etc.
* The admin maintains the information of both students and teachers in the database whenever he/she fill up the form.
* Administrator can add, and delete the student information.
* Administrator enables to display notices and other messages on the website

**Advantages**

* Eco-Friendly: paperwork can be avoided.
* It cantain better storage capacity.
* Accuracy in work.
* Easy & fast retrieval of information.
* Decrease the load of the person involve in existing manual system.
* Access of any information individually.
* Work become very speedy.
* Easy to update information.
* Smart management of student’s data
* User accounts to control the access and maintain security.
* Cost-efficient and User-friendly.
* You will obtain the details of the students by entering his/her name or the roll number just in one click.
* The data that will be disclosed will be more secure since there will be no access to the unknown users.

**Disadvantages**

* Absence of proper internet-network makes it difficult for a user to access information.
* Applications face minor technical gseereeches
* Only, people who are accustomed to regular use of smartphones or computers can operate this software.
* Extensive modules and features make it difficult for a user to utilise the application.
* With huge flow in traffic the application is prone to performance issues.

**Introduction to the Oracle Database**

This chapter provides an overview of the Oracle database server. An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost-effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures. The following are tables used in On-Line matrimony projects.

**Admin**

|  |  |
| --- | --- |
| emailid | varchar2(50) |
| password | varchar2(50), |
| otp | varchar2(10) |

**Student**

|  |  |
| --- | --- |
| emailid | varchar2(50) primary key, |
| password | varchar2(50), |
| otp | varchar2(10), |
| status | varchar2(10), |
| name | varchar2(50), |
| regdno | varchar2(10), |
| branch | varchar2(50), |
| dob | varchar2(15), |
| gender | varchar2(10), |
| bloodgroup | varchar2(5), |
| mobileno | varchar2(12), |
| adhaarno | varchar2(20), |
| ssession | varchar2(15), |
| semester | varchar2(10), |
| permanent\_landmark | varchar2(50), |
| permanent\_city | varchar2(50), |
| permanent\_district | varchar2(50), |
| permanent\_state | varchar2(50), |
| permanent\_pincode | varchar2(10), |
| permanent\_country | varchar2(50), |
| present\_country | varchar2(50), |
| present\_landmark | varchar2(50), |
| present\_city | varchar2(50), |
| present\_district | varchar (50), |
| present\_state | varchar (50), |
| present\_pincode | varchar2(10), |
| fname | varchar2(50), |
| mname | varchar2(50), |
| photo | varchar2(50) |

**Teacher**

|  |  |
| --- | --- |
| emailid | varchar2(50) primary key, |
| password | varchar2(50), |
| otp | varchar2(10), |
| status | varchar2(10), |
| name | varchar2(50), |
| fname | varchar2(50) |
| mname | varchar2(50), |
| regdno | varchar2(10), |
| joindate | varchar2(20), |
| dob | varchar2(15), |
| gender | varchar2(10), |
| bloodgroup | varchar2(5), |
| mobileno | varchar2(12), |
| adhaarno | varchar2(20), |
| department | varchar2(50), |
| qualification | varchar2(30), |
| permanent\_landmark | varchar2(50), |
| permanent\_city | varchar2(50), |
| permanent\_district | varchar2(50), |
| permanent\_state | varchar2(50), |
| permanent\_pincode | varchar2(10), |
| permanent\_country | varchar2(50), |
| present\_country | varchar2(50), |
| present\_landmark | varchar2(50), |
| present\_city | varchar2(50), |
| present\_district | varchar (50), |
| present\_state | varchar (50), |
| present\_pincode | varchar2(10), |
| photo | varchar2(50) |

Studentfeedback

|  |  |
| --- | --- |
| regdno | varchar2(20), |
| emailid | varchar2(50), |
| message | varchar2(150) |

Teacherfeedback

|  |  |
| --- | --- |
| regdno | varchar2(20), |
| emailid | varchar2(50), |
| message | varchar2(150) |

contact

|  |  |
| --- | --- |
| name | varchar2(50), |
| emailid | varchar2(50), |
| message | varchar2(200) |

Student leave

|  |  |
| --- | --- |
| regdno | varchar2(50), |
| t | varchar2(10), |
| f | varchar2(10), |
| status | varchar2(2) |
| emailid | varchar2(50), |
| message | varchar2(200) |

Teacher leave

|  |  |
| --- | --- |
| regdno | varchar2(50), |
| t | varchar2(10), |
| f | varchar2(10), |
| status | varchar2(2) |
| emailid | varchar2(50), |
| message | varchar2(200) |

Student attendance

|  |  |
| --- | --- |
| attendance date | varchar2(50), |
| branch | varchar2(50), |
| semester | varchar2(50), |
| subject | varchar2(50), |
| emailid | varchar2(50), |
| regdno | varchar2(50), |
| status | varchar2(4), |
| name | varchar2(50) |

Teacher attendance

|  |  |
| --- | --- |
| attendance date | varchar2(50), |
| branch | varchar2(50), |
| semester | varchar2(50), |
| subject | varchar2(50), |
| emailid | varchar2(50), |
| regdno | varchar2(50), |
| status | varchar2(4), |
| name | varchar2(50) |

Chat Messages

|  |  |
| --- | --- |
| sender | varchar2(50), |
| receiver | varchar2(50), |
| message | varchar2(500), |
| time | timestamp, |
| status | varchar2(1) |

Chat Student

|  |  |
| --- | --- |
| name | varchar2(50), |
| emailid | varchar2(50), |
| password | varchar2(10), |
| status | varchar2(1) |

Subject

|  |  |
| --- | --- |
| subjectid | varchar2(50), |
| subjectname | varchar2(50), |
| teacherregdno | varchar2(50), |
| semester | varchar2(50), |
| branch | varchar2(50) |

**Data Flow Diagram (DFD)**

A picture is worth a thousand words. A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system. Here are the symbols you’ll need to use to create a DFD:

Processes are a circle or a square with a horizontal line across the top. A process is a business activity where the manipulation and transformation of data occurs. Something happens to the data during a process.

Arrows represent how the data flows. Use the type of data that is moving through the system as the name for the arrow.



External Entity is shown as a square. An external entity can be a person, system, or application. It’s where data starts or ends.

Data Stores are rectangles (sometimes they have a vertical line in the symbol), and they show where required or produced data related to the process is stored.

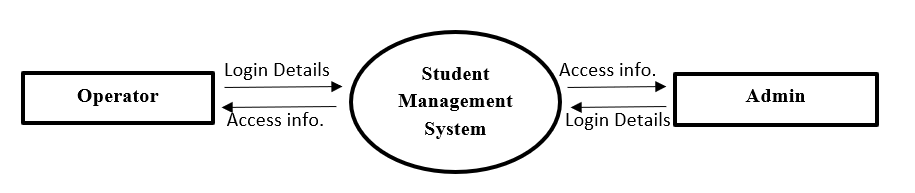
**Levels of DFD**

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

**0-Level DFD**

DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

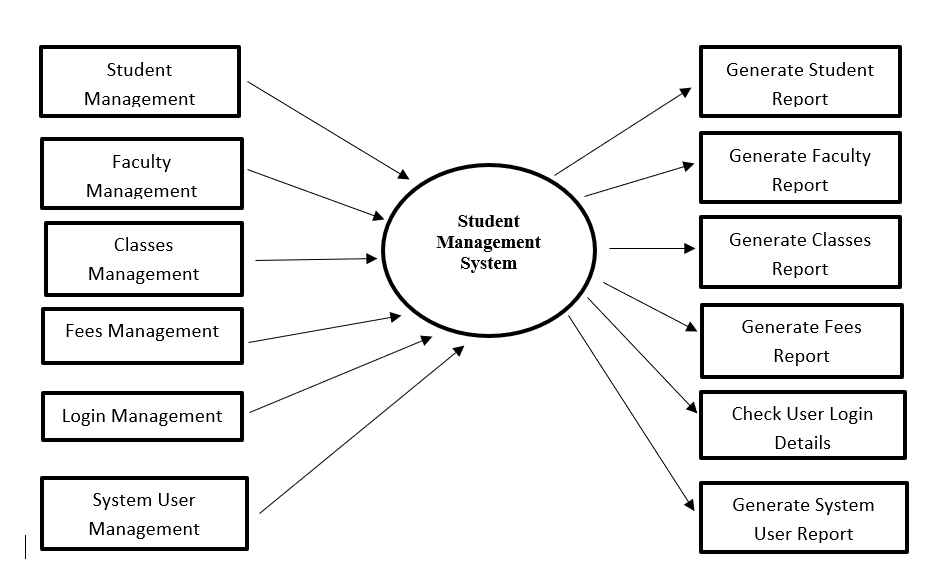
It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs.



**1-Level DFD**

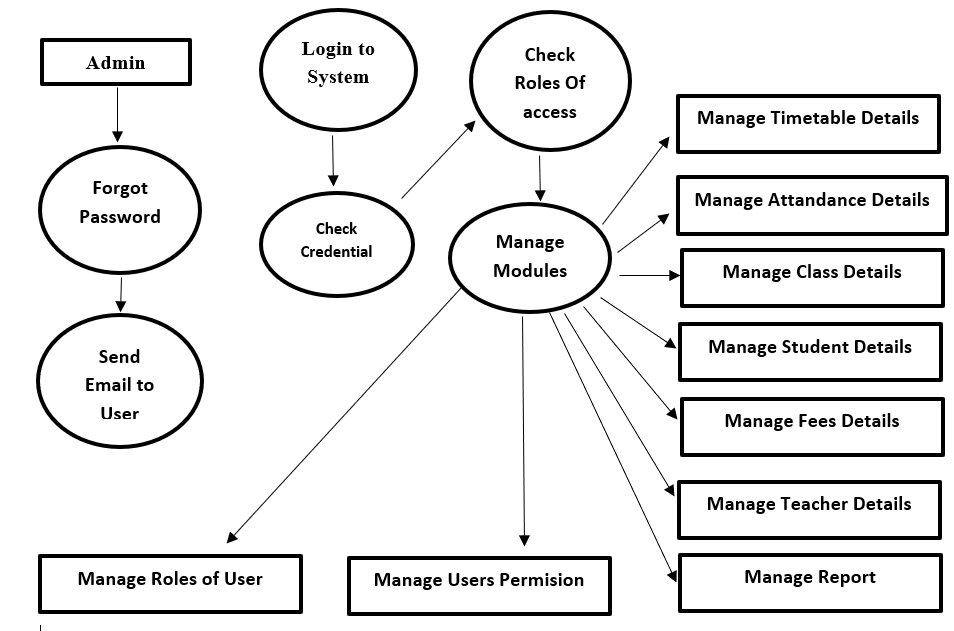
DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.

**1-Level DFD**



**2-Level DFD**

DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system’s functioning.



**Entity Relationship Diagrams (ER-Diagram)**

ENTITY RELATIONAL (ER) MODEL is a high-level conceptual data model diagram. ER modeling helps you to analyze data requirements systematically to produce a well-designed database. The Entity-Relation model represents real-world entities and the relationship between them. It is considered a best practice to complete ER modeling before implementing your database. ER modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

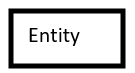
ENTITY-RELATIONSHIP DIAGRAM (ERD) displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

**Facts about ER Diagram Model:**

* ER model allows you to draw Database Design
* It is an easy to use graphical tool for modeling data
* Widely used in Database Design
* It is a GUI representation of the logical structure of a Database
* It helps you to identifies the entities which exist in a system and the relationships between those entities

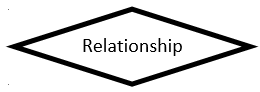
**Common Entity Relationship Diagram Symbols**

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

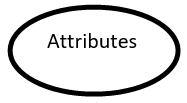
Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information.

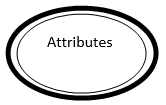


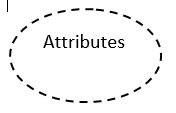
A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

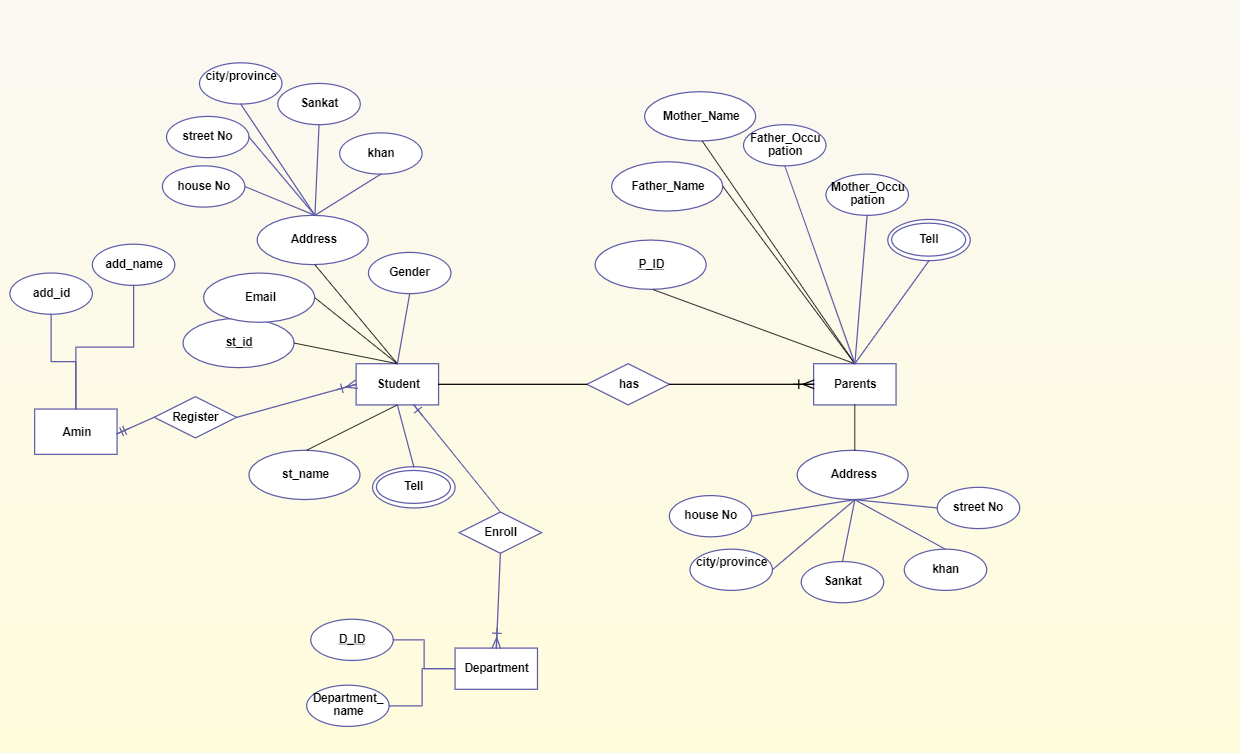


Actions, which are represented by diamond shapes, show how two entities share information in the database.

Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

 A derived attribute is an attribute whose value is calculated (derived) from other attributes.



**Unified Modeling Language (UML) Diagram**

Unified Modeling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

* Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.
* Behavior Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

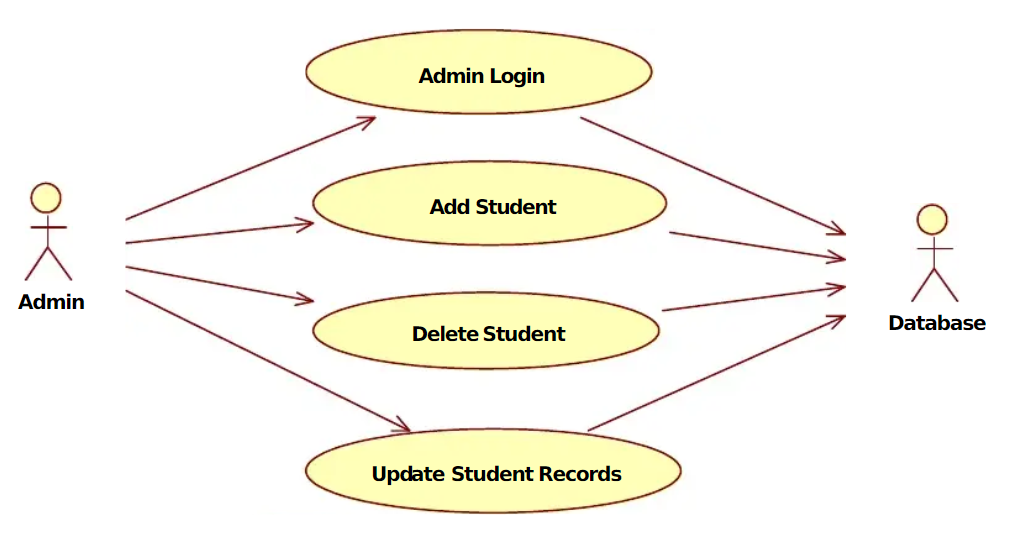
**Structural UML Diagrams**

* ***Class Diagram* –** The most widely use UML diagram is the class diagram. It is the building block of all object-oriented software systems. We use class diagrams to depict the static structure of a system by showing system’s classes, their methods and attributes. Class diagrams also help us identify relationship between different classes or objects.
* ***Composite Structure Diagram –*** We use composite structure diagrams to represent the internal structure of a class and its interaction points with other parts of the system. A composite structure diagram represents relationship between parts and their configuration which determine how the classifier (class, a component, or a deployment node) behaves. They represent internal structure of a structured classifier making the use of parts, ports, and connectors. We can also model collaborations using composite structure diagrams. They are similar to class diagrams except they represent individual parts in detail as compared to the entire class.
* ***Object Diagram –*** An Object Diagram can be referred to as a screenshot of the instances in a system and the relationship that exists between them. Since object diagrams depict behaviour when objects have been instantiated, we are able to study the behaviour of the system at a particular instant. An object diagram is similar to a class diagram except it shows the instances of classes in the system. We depict actual classifiers and their relationships making the use of class diagrams. On the other hand, an Object Diagram represents specific instances of classes and relationships between them at a point of time.
* ***Component Diagram –*** Component diagrams are used to represent the how the physical components in a system have been organized. We use them for modelling implementation details. Component Diagrams depict the structural relationship between software system elements and help us in understanding if functional requirements have been covered by planned development. Component Diagrams become essential to use when we design and build complex systems. Interfaces are used by components of the system to communicate with each other.
* ***Deployment Diagram –*** Deployment Diagrams are used to represent system hardware and its software. It tells us what hardware components exist and what software components run on them. We illustrate system architecture as distribution of software artifacts over distributed targets. An artifact is the information that is generated by system software. They are primarily used when a software is being used, distributed or deployed over multiple machines with different configurations.
* ***Package Diagram –*** We use Package Diagrams to depict how packages and their elements have been organized. A package diagram simply shows us the dependencies between different packages and internal composition of packages. Packages help us to organise UML diagrams into meaningful groups and make the diagram easy to understand. They are primarily used to organise class and use case diagrams.

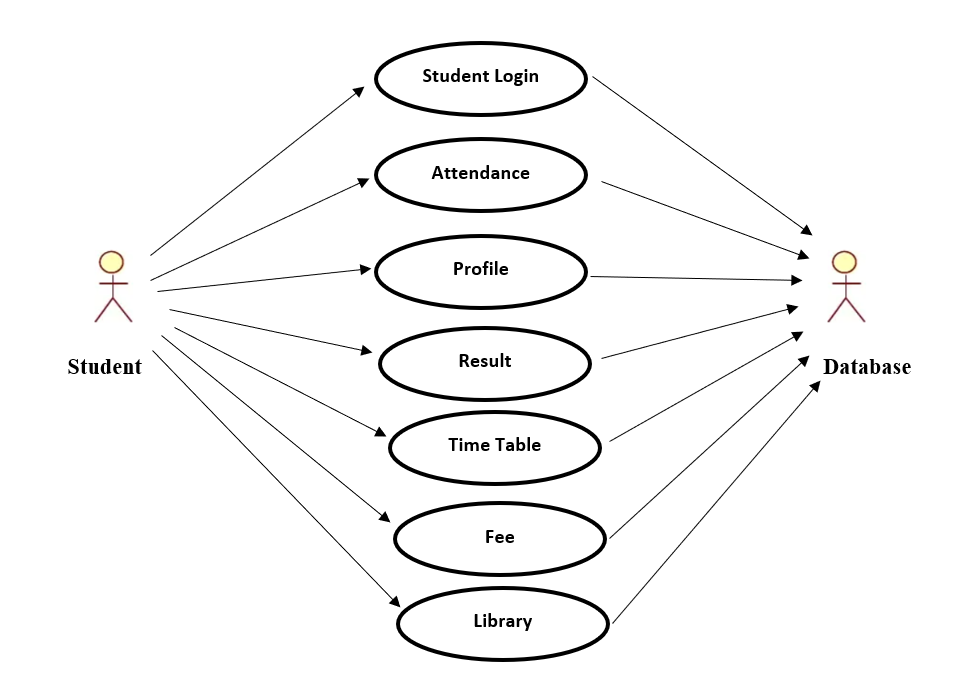
**Behavior Diagrams –**

* ***State Machine Diagrams –*** A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli.
* ***Activity Diagrams –*** We use Activity Diagrams to illustrate the flow of control in a system. We can also use an activity diagram to refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.
* ***Use Case Diagrams –*** Use Case Diagrams are used to depict the functionaseereey of a system or a part of a system. They are widely used to illustrate the functional requirements of the system and its interaction with external agents(actors). A use case is basically a diagram representing different scenarios where the system can be used. A use case diagram gives us a high-level view of what the system or a part of the system does without going into implementation details.
* ***Sequence Diagram –*** A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.
* ***Communication Diagram –*** A Communication Diagram (known as Collaboration Diagram in UML 1.x) is used to show sequenced messages exchanged between objects. A communication diagram focuses primarily on objects and their relationships. We can represent similar information using Sequence diagrams; however, communication diagrams represent objects and links in a free form.
* ***Timing Diagram –*** Timing Diagram are a special form of Sequence diagrams which are used to depict the behavior of objects over a time frame. We use them to show time and duration constraints which govern changes in states and behavior of objects.
* ***Interaction Overview Diagram –*** An Interaction Overview Diagram models a sequence of actions and helps us simplify complex interactions into simpler occurrences. It is a mixture of activity and sequence diagrams.

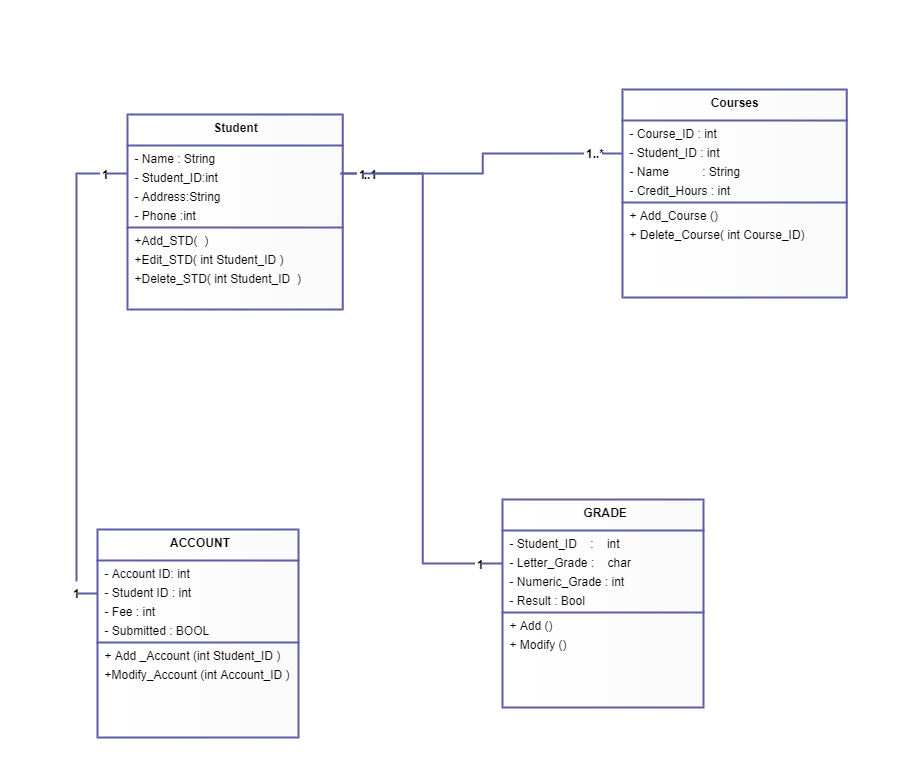
**Use case diagram of Admin**



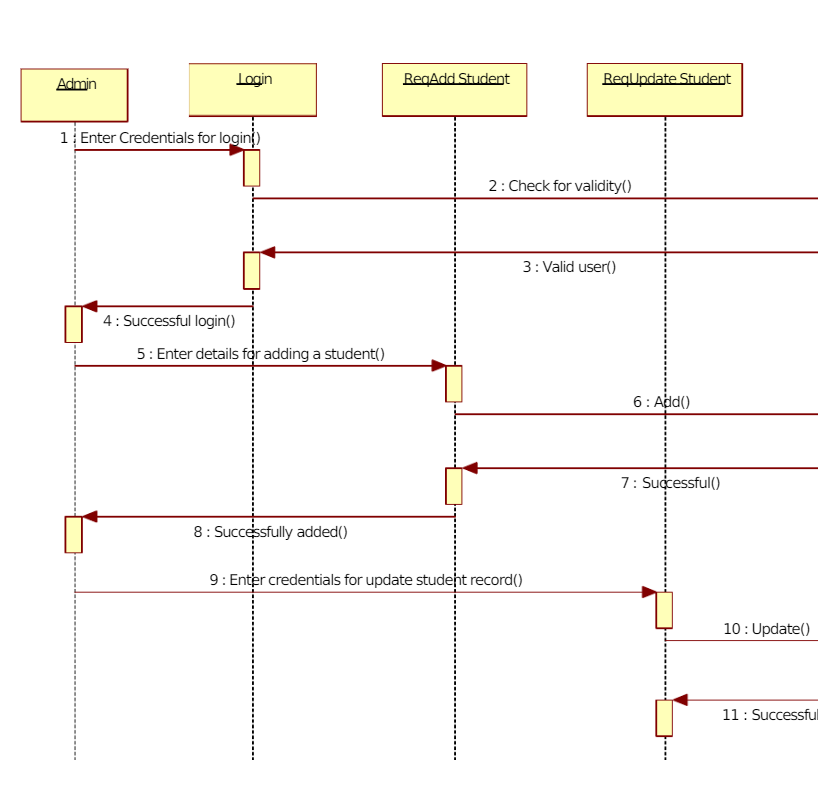
**Use case diagram of Student**



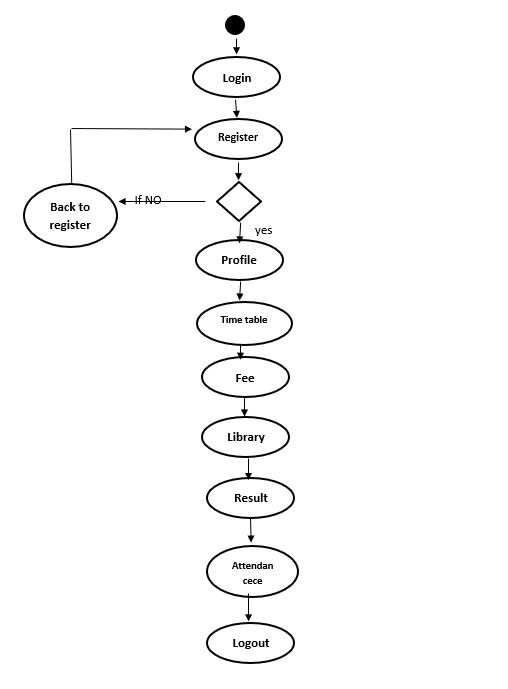
**Class Diagram of Student Management System**



**Sequence Diagram of Student Management System**



**Activity Diagram of E-Learning**



**State-chat diagram of E-Learning**

**Coding**

Coding is the process of using a programming language to get a computer to behave how you want it to. Every line of code tells the computer to do something,

**index.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>ITER</title>

</head>

<body>

<jsp:include page=*"mainHeader.jsp"*></jsp:include>

<jsp:include page=*"mainSlider.jsp"*></jsp:include>

<jsp:include page=*"mainBody.jsp"*></jsp:include>

<jsp:include page=*"mainFooter.jsp"*></jsp:include>

</body>

</html>

**addStudent.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Add Student</title>

</head>

<body>

<jsp:include page=*"adminWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"addStudentBody.jsp"*></jsp:include>

<jsp:include page=*"adminWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**addSubject.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin</title>

</head>

<body>

<jsp:include page=*"adminWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"addSubjectBody.jsp"*></jsp:include>

<jsp:include page=*"adminWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**addTeacher.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin</title>

</head>

<body>

<jsp:include page=*"adminWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"addSubjectBody.jsp"*></jsp:include>

<jsp:include page=*"adminWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**adminAttendance.jsp**

<!DOCTYPE html>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>Student Attendance</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<form action=*"sms.controller.AdminStudentAttendance"* method=*"post"*>

<div class=*"row header"*>

<div class=*"cell"*>

Date: <input type=*"date"* name=*"attendancedate"*>

</div>

<div class=*"cell"*>

Branch: <select name=*"branch"*>

<option value=*"CSE"*>CSE</option>

<option value=*"Civil"*>Civil</option>

<option value=*"MEC"*>ME</option>

<option value=*"EEE"*>EEE</option>

<option value=*"EE"*>EE</option>

</select>

</div>

<div class=*"cell"*>

Semester: <select name=*"semester"*>

<option value=*"1st Semester"*>1st</option>

<option value=*"2nd Semester"*>2nd</option>

<option value=*"3rd Semester"*>3rd</option>

<option value=*"4th Semester"*>4th</option>

<option value=*"5th Semester"*>5th</option>

<option value=*"6th Semester"*>6th</option>

<option value=*"7th Semester"*>7th</option>

<option value=*"8th Semester"*>8th</option>

</select>

</div>

<div class=*"cell"*>

Subject: <select name=*"subject"*>

<option value=*"Java"*>Java</option>

</select>

</div>

<div class=*"cell"*>

<input type=*"submit"* value=*"Submit"*>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**adminChangePassword.jsp**

<!DOCTYPE html>

<html lang=*"en"*>

<head>

<title>Admin Change Password</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/bootstrap/css/bootstrap.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/iconic/css/material-design-iconic-font.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/css-hamburgers/hamburgers.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/animsition/css/animsition.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/daterangepicker/daterangepicker.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/util.css"*>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/main.css"*>

<!--===============================================================================================-->

</head>

<body>

<div class=*"limiter"*>

<div class=*"container-login100"*>

<div class=*"wrap-login100"*>

<form action=*"sms.controller.AdminChangePassword"*

class=*"login100-form validate-form"* method=*"post"*>

<span class=*"login100-form-title p-b-26"*> Change Password </span> </span>

<div class=*"wrap-input100 validate-input"*

data-validate=*"Enter password"*>

<span class=*"btn-show-pass"*> <i class=*"zmdi zmdi-eye"*></i>

</span> <input class=*"input100"* type=*"password"* name=*"oldpassword"*>

<span class=*"focus-input100"* data-placeholder=*"Old Password"*></span>

</div>

<div class=*"wrap-input100 validate-input"*

data-validate=*"Enter password"*>

<span class=*"btn-show-pass"*> <i class=*"zmdi zmdi-eye"*></i>

</span> <input class=*"input100"* type=*"password"* name=*"newpassword"*>

<span class=*"focus-input100"* data-placeholder=*"New Password"*></span>

</div>

<div class=*"wrap-input100 validate-input"*

data-validate=*"Enter password"*>

<span class=*"btn-show-pass"*> <i class=*"zmdi zmdi-eye"*></i>

</span> <input class=*"input100"* type=*"password"* name=*"confirmpassword"*>

<span class=*"focus-input100"* data-placeholder=*"Confirm Password"*></span>

</div>

<div class=*"container-login100-form-btn"*>

<div class=*"wrap-login100-form-btn"*>

<div class=*"login100-form-bgbtn"*></div>

<button class=*"login100-form-btn"*>Submit</button>

</div>

</div>

</form>

</div>

</div>

</div>

<div id=*"dropDownSelect1"*></div>

<!--===============================================================================================-->

<script src=*"vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/animsition/js/animsition.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/daterangepicker/moment.min.js"*></script>

<script src=*"vendor/daterangepicker/daterangepicker.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/countdowntime/countdowntime.js"*></script>

<!--===============================================================================================-->

<script src=*"js/main.js"*></script>

</body>

</html>

**adminVerify.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%

String emailid = request.getParameter("emailid");

%>

<h1>Verify OTP</h1>

<form action=*"sms.controller.VerifyAdminOTP"* method=*"post"*>

<input type=*"hidden"* name=*"emailid"* value=*"*<%=emailid%>*"*> OTP

: <input type=*"text"* name=*"otp"*><br> <input type=*"submit"*

value=*"Verify"*>

</form>

</body>

</html>

**adminViewStudentLeave.jsp**

<!DOCTYPE html>

<%@page import=*"sms.db.StudentLeave"*%>

<%@page import=*"sms.model.StudentLeaveDao"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>View Student Leave</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>To</div>

<div class=*"cell"*>From</div>

<div class=*"cell"* style="width: *300px*;">Message</div>

<div class=*"cell"* style="width: *300px*;">Status</div>

</div>

<%

ArrayList<StudentLeave> all = StudentLeaveDao.fetchStudentLeave();

Iterator it = all.iterator();

**while** (it.hasNext()) {

StudentLeave s1 = (StudentLeave) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Regd NO"*>

<%=s1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Emailid"*>

<%=s1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"To"*>

<%=s1.getTo()%>

</div>

<div class=*"cell"* data-title=*"From"*>

<%=s1.getFrom()%>

</div>

<div class=*"cell"* data-title=*"Message"*>

<%=s1.getMessage()%>

</div>

<div class=*"cell"* data-title=*"Status"*>

<%

String status = s1.getStatus();

**if** (status != **null**) {

**if** (status.equals("1")) {

%>

Approved

<%

}

**if** (status.equals("0")) {

%>

Rejected

<%

}

} **else** {

%>

<a

href=*"sms.controller.ApproveStudent?emailid=*<%=s1.getEmailid()%>*&status=1&to=*<%=s1.getTo()%>*"*>Approve</a>

&nbsp;&nbsp; <a

href=*"sms.controller.ApproveStudent?emailid=*<%=s1.getEmailid()%>*&status=0&to=*<%=s1.getTo()%>*"*>Reject</a>

<%

}

%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**adminViewTeacherLeave.jsp**

<!DOCTYPE html>

<%@page import=*"sms.model.TeacherLeaveDao"*%>

<%@page import=*"sms.db.TeacherLeave"*%>

<%@page import=*"sms.db.Teacher"*%>

<%@page import=*"sms.model.TeacherDao"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>View Teacher Leave</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>To</div>

<div class=*"cell"*>From</div>

<div class=*"cell"* style="width: *300px*;">Message</div>

<div class=*"cell"* style="width: *300px*;">Status</div>

</div>

<%

ArrayList<TeacherLeave> all = TeacherLeaveDao.fetchTeacherLeave();

Iterator it = all.iterator();

**while** (it.hasNext()) {

TeacherLeave t1 = (TeacherLeave) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Regd No"*>

<%=t1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Emailid"*>

<%=t1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"To"*>

<%=t1.getTo()%>

</div>

<div class=*"cell"* data-title=*"From"*>

<%=t1.getFrom()%>

</div>

<div class=*"cell"* data-title=*"Message"*>

<%=t1.getMessage()%>

</div>

<div class=*"cell"* data-title=*"Status"*>

<%

String status = t1.getStatus();

**if** (status != **null**) {

**if** (status.equals("1")) {

%>

Approved

<%

}

**if** (status.equals("0")) {

%>

Rejected

<%

}

} **else** {

%>

<a

href=*"sms.controller.ApproveTeacher?emailid=*<%=t1.getEmailid()%>*&status=1&to=*<%=t1.getTo()%>*"*>Approve</a>

&nbsp;&nbsp; <a

href=*"sms.controller.ApproveTeacher?emailid=*<%=t1.getEmailid()%>*&status=0&to=*<%=t1.getTo()%>*"*>Reject</a>

<%

}

%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**adminWelcome.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin</title>

</head>

<body>

<jsp:include page=*"adminWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"adminWelcomeBody.jsp"*></jsp:include>

<jsp:include page=*"adminWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**allStudentBody.jsp**

<!DOCTYPE html>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>All Student</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Full Name</div>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>Photo</div>

</div>

<%

ArrayList<Student> all = StudentDao.fetchStudent();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Student s1 = (Student) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=s1.getName()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=s1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"Location"*>Photo</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**allStudentFeedback.jsp**

<!DOCTYPE html>

<%@page import=*"sms.model.StudentFeedbackDao"*%>

<%@page import=*"sms.db.FeedbackStudent"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>Student Feedback</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>Message</div>

</div>

<%

ArrayList<FeedbackStudent> all = StudentFeedbackDao.fetchStudentFeedback();

Iterator it = all.iterator();

**while** (it.hasNext()) {

FeedbackStudent fs = (FeedbackStudent) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Age"*>

<%=fs.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=fs.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"Location"*>

<%=fs.getMessage()%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**allSubjectBody.jsp**

<!DOCTYPE html>

<%@page import=*"sms.db.Subject"*%>

<%@page import=*"sms.model.AdminDao"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>All Subject</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Subject Id</div>

<div class=*"cell"*>Subject Name</div>

<div class=*"cell"*>Teacher Regd No</div>

<div class=*"cell"*>Semester</div>

<div class=*"cell"* style="width: *110px*;">Branch</div>

</div>

<%

ArrayList<Subject> all = AdminDao.fetchSubject();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Subject s1 = (Subject) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=s1.getSubjectid()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=s1.getSubjectname()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getTeacherregdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getSemester()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getBranch()%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**chat.jsp**

<%@page import=*"sms.db.Teacher"*%>

<%@page import=*"sms.db.ChatStudent"*%>

<%@page import=*"sms.model.ChatStudentDao"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"java.util.Iterator"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Chatting room</title>

<style type=*"text/css"*>

*.mainDiv* {

}

*.sideDiv* {

width: *100%*;

min-height: *60px*;

background: *linear-gradient(to right, rgba(208, 240, 162, 1) 2%,*

*rgba(69, 117, 1, 1) 47%)*;

color: *red*;

}

*.leftDiv* {

border-top-left-radius: *15px*;

overflow-y: *scroll*;

overflow-x: *hidden*;

position: *absolute*;

height: *584px*;

width: *20%*;

margin-top: *5px*;

/\* background: -webkit-linear-gradient(45deg, rgba(154,30,80,1) 0%,rgba(232,126,179,0) 100%); \*/

/\* background-image: url(images/webchat.jpg); \*/

background-color: *white*;

}

*.rightDiv* {

height: *584px*;

width: *79%*;

margin-top: *5px*;

margin-left: *30%*;

float: *right*;

/\* background: linear-gradient(135deg, #f3c5bd 0%,#ff6600 50%); \*/

background-color: *white*;

}

*#ownName* {

max-height: *60px*;

max-width: *28.66%*;

position: *absolute*;

border-top-left-radius: *15px*;

font-size: *35px*;

padding: *10px*;

}

*#friends* {

height: *540px*;

min-width: *28.66%*;

/\* background: linear-gradient(to bottom, rgba(154,121,30,1) 0%,rgba(166,132,44,1) 15%,rgba(207,170,95,0.7) 68%,rgba(232,193,126,0) 100%); \*/

background-repeat: *no-repeat*;

}

**body** {

/\* background: linear-gradient(45deg, rgba(102,207,252,1) 0%,rgba(3,149,207,1) 62%); \*/

}

</style>

</head>

<body>

<%

String emailid = (String) session.getAttribute("emailid");

System.out.println(emailid);

**if** (emailid == **null**)

response.sendRedirect("index.jsp");

%>

<form action=*"studentWelcome.jsp"*>

<input type=*"submit"* value=*"back"*

style="float: *right*; margin-right: *30px*; margin-top: *20px*; background-color: *gray*; color: *white*;">

</form>

<div class=*"mainDiv"*>

<div class=*""*>

<div id=*""*>

<%

String name = request.getParameter("name");

System.out.println(name);

//out.print(name);

%>

</div>

<!-- <a href="mm.controller.Logout"><img alt="LogOut" src="images/exit.png" height="40px" width="40px" style="box-shadow: -42px 12px 73px -27px rgba(0,0,0,0.75);float: right; padding-top: 10px;padding-right: 20px;"></a> -->

</div>

<div class=*"rightDiv"*></div>

<div class=*"leftDiv"*>

<div id=*"friends"*>

<table>

<%

ArrayList<Teacher> all = ChatStudentDao.fetchTeacher();

Iterator<Teacher> it = all.iterator();

**while** (it.hasNext()) {

Teacher tr = it.next();

%>

<tr>

<td

style="font-size: *25px*; padding-top: *10px*; text-decoration: *none*; color: *black*; padding-left: *20px*;"><a

href=*"sms.controller.ChatPage?name=*<%=tr.getName()%>*&emailid=*<%=tr.getEmailid()%>*"*

style="text-decoration: *none*; color: *black*;"><%=tr.getName()%></a></td>

<%

**if** (tr.getStatus() != **null**) {

%>

<td><img src=*"images/online.png"*

style="float: *right*; height: *20px*; width: *20px*; position: *absolute*; right: *20px*;"></td>

<%

} **else** {

%>

<td><img src=*"images/ofline.png"*

style="float: *right*; height: *20px*; width: *20px*; position: *absolute*; right: *20px*;"></td>

<%

}

%>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

</body>

</html>

**deleteStudent.jsp**

<!DOCTYPE html>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>Delete Student</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Full Name</div>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>Photo</div>

<div class=*"cell"* style="width: *130px*;">Delete</div>

</div>

<%

ArrayList<Student> all = StudentDao.fetchStudent();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Student s1 = (Student) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=s1.getName()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=s1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"Location"*>Photo</div>

<div class=*"cell"* data-title=*"Location"*>

<a

href=*"sms.controller.DeleteStudent?emailid=*<%=s1.getEmailid()%>*"*>Delete</a>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**deleteSubject.jsp**

<!DOCTYPE html>

<%@page import=*"sms.db.Subject"*%>

<%@page import=*"sms.model.AdminDao"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>Delete Subject</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Subject Id</div>

<div class=*"cell"*>Subject Name</div>

<div class=*"cell"*>Teacher Regd No</div>

<div class=*"cell"*>Semester</div>

<div class=*"cell"* style="width: *130px*;">Branch</div>

<div class=*"cell"* style="width: *130px*;">Delete</div>

</div>

<%

ArrayList<Subject> all = AdminDao.fetchSubject();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Subject s1 = (Subject) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=s1.getSubjectid()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=s1.getSubjectname()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getTeacherregdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getSemester()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getBranch()%>

</div>

<div class=*"cell"* data-title=*"Location"*>

<a

href=*"sms.controller.DeleteSubject?subjectid=*<%=s1.getSubjectid()%>*"*>Delete</a>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**deleteTeacher.jsp**

<!DOCTYPE html>

<%@page import=*"sms.db.Teacher"*%>

<%@page import=*"sms.model.TeacherDao"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>Delete Teacher</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Full Name</div>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>Photo</div>

<div class=*"cell"* style="width: *130px*;">Delete</div>

</div>

<%

ArrayList<Teacher> all = TeacherDao.fetchTeacher();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Teacher t1 = (Teacher) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=t1.getName()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=t1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=t1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"Location"*>Photo</div>

<div class=*"cell"* data-title=*"Location"*>

<a

href=*"sms.controller.DeleteTeacher?emailid=*<%=t1.getEmailid()%>*"*>Delete</a>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**editStudentProfile.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Edit Profile</title>

</head>

<body>

<jsp:include page=*"studentWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"editStudentProfileBody.jsp"*></jsp:include>

<jsp:include page=*"studentWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**editTeacherProfile.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin</title>

</head>

<body>

<jsp:include page=*"teacherWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"editTeacherProfileBody.jsp"*></jsp:include>

<jsp:include page=*"teacherWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**studentChangePassword.jsp**

<!DOCTYPE html>

<html lang=*"en"*>

<head>

<title>Admin Change Password</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/bootstrap/css/bootstrap.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/iconic/css/material-design-iconic-font.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/css-hamburgers/hamburgers.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/animsition/css/animsition.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor/daterangepicker/daterangepicker.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/util.css"*>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/main.css"*>

<!--===============================================================================================-->

</head>

<body>

<div class=*"limiter"*>

<div class=*"container-login100"*>

<div class=*"wrap-login100"*>

<form action=*"sms.controller.StudentChangePassword"*

class=*"login100-form validate-form"* method=*"post"*>

<span class=*"login100-form-title p-b-26"*> Change Password </span> </span>

<div class=*"wrap-input100 validate-input"*

data-validate=*"Enter password"*>

<span class=*"btn-show-pass"*> <i class=*"zmdi zmdi-eye"*></i>

</span> <input class=*"input100"* type=*"password"* name=*"oldpassword"*>

<span class=*"focus-input100"* data-placeholder=*"Old Password"*></span>

</div>

<div class=*"wrap-input100 validate-input"*

data-validate=*"Enter password"*>

<span class=*"btn-show-pass"*> <i class=*"zmdi zmdi-eye"*></i>

</span> <input class=*"input100"* type=*"password"* name=*"newpassword"*>

<span class=*"focus-input100"* data-placeholder=*"New Password"*></span>

</div>

<div class=*"wrap-input100 validate-input"*

data-validate=*"Enter password"*>

<span class=*"btn-show-pass"*> <i class=*"zmdi zmdi-eye"*></i>

</span> <input class=*"input100"* type=*"password"* name=*"confirmpassword"*>

<span class=*"focus-input100"* data-placeholder=*"Confirm Password"*></span>

</div>

<div class=*"container-login100-form-btn"*>

<div class=*"wrap-login100-form-btn"*>

<div class=*"login100-form-bgbtn"*></div>

<button class=*"login100-form-btn"*>Submit</button>

</div>

</div>

</form>

</div>

</div>

</div>

<div id=*"dropDownSelect1"*></div>

<!--===============================================================================================-->

<script src=*"vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/animsition/js/animsition.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/daterangepicker/moment.min.js"*></script>

<script src=*"vendor/daterangepicker/daterangepicker.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor/countdowntime/countdowntime.js"*></script>

<!--===============================================================================================-->

<script src=*"js/main.js"*></script>

</body>

</html>

**studentFeedback.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Admin</title>

</head>

<body>

<jsp:include page=*"studentWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"studentFeedbackBody.jsp"*></jsp:include>

<jsp:include page=*"studentWelcomeFooter.jsp"*></jsp:include>

</body>

</html>

**studentVerify.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%

String emailid = request.getParameter("emailid");

%>

<h1>Verify OTP</h1>

<form action=*"sms.controller.VerifyStudentOTP"* method=*"post"*>

<input type=*"hidden"* name=*"emailid"* value=*"*<%=emailid%>*"*> OTP

: <input type=*"text"* name=*"otp"*><br> <input type=*"submit"*

value=*"Verify"*>

</form>

</body>

</html>

**studentViewProfile.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%

String email = (String) session.getAttribute("emailid");

**if** (email == **null**)

response.sendRedirect("studentLogin.jsp");

%>

<jsp:include page=*"studentWelcomeHeader.jsp"*></jsp:include>

<jsp:include page=*"studentViewProfileBody.jsp"*></jsp:include>

**teacherVerify.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%

String emailid = request.getParameter("emailid");

%>

<h1>Verify OTP</h1>

<form action=*"sms.controller.VerifyTeacherOTP"* method=*"post"*>

<input type=*"hidden"* name=*"emailid"* value=*"*<%=emailid%>*"*> OTP

: <input type=*"text"* name=*"otp"*><br> <input type=*"submit"*

value=*"Verify"*>

</form>

</body>

</html>

**viewStudentLeave.jsp**

<!DOCTYPE html>

<%@page import=*"sms.model.StudentLeaveDao"*%>

<%@page import=*"sms.db.StudentLeave"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>View Student Leave</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>To</div>

<div class=*"cell"*>From</div>

<div class=*"cell"* style="width: *300px*;">Message</div>

<div class=*"cell"* style="width: *300px*;">Status</div>

</div>

<%

ArrayList<StudentLeave> all = StudentLeaveDao.fetchStudentLeave();

Iterator it = all.iterator();

**while** (it.hasNext()) {

StudentLeave s1 = (StudentLeave) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Regd NO"*>

<%=s1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Emailid"*>

<%=s1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"To"*>

<%=s1.getTo()%>

</div>

<div class=*"cell"* data-title=*"From"*>

<%=s1.getFrom()%>

</div>

<div class=*"cell"* data-title=*"Message"*>

<%=s1.getMessage()%>

</div>

<div class=*"cell"* data-title=*"Status"*>

<%

String status = s1.getStatus();

**if** (status != **null**) {

**if** (status.equals("1")) {

%>

Approved

<%

}

**if** (status.equals("0")) {

%>

Rejected

<%

}

} **else** {

%>

Not Approve

<%

}

%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**viewTeacherLeave.jsp**

<!DOCTYPE html>

<%@page import=*"sms.db.TeacherLeave"*%>

<%@page import=*"sms.model.TeacherLeaveDao"*%>

<%@page import=*"sms.model.StudentLeaveDao"*%>

<%@page import=*"sms.db.StudentLeave"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>View Teacher Leave</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>To</div>

<div class=*"cell"*>From</div>

<div class=*"cell"* style="width: *300px*;">Message</div>

<div class=*"cell"* style="width: *300px*;">Status</div>

</div>

<%

ArrayList<TeacherLeave> all = TeacherLeaveDao.fetchTeacherLeave();

Iterator it = all.iterator();

**while** (it.hasNext()) {

TeacherLeave t1 = (TeacherLeave) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Regd NO"*>

<%=t1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Emailid"*>

<%=t1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"To"*>

<%=t1.getTo()%>

</div>

<div class=*"cell"* data-title=*"From"*>

<%=t1.getFrom()%>

</div>

<div class=*"cell"* data-title=*"Message"*>

<%=t1.getMessage()%>

</div>

<div class=*"cell"* data-title=*"Status"*>

<%

String status = t1.getStatus();

**if** (status != **null**) {

**if** (status.equals("1")) {

%>

Approved

<%

}

**if** (status.equals("0")) {

%>

Rejected

<%

}

} **else** {

%>

Not Approve

<%

}

%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**viewStudent.jsp**

<!DOCTYPE html>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>View Student</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Full Name</div>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>Photo</div>

</div>

<%

ArrayList<Student> all = StudentDao.fetchStudent();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Student s1 = (Student) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=s1.getName()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=s1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=s1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"Location"*>

<%

String photo\_name = s1.getPhoto();

**if** (photo\_name == **null**) {

%>

<img src=*"profileimg/b2.jpg"* style="height: *80px*; width: *70px*;">

<%

} **else** {

//String p1 = "profileimg\\";

String p2 = s1.getPhoto();

%>

<img src=*"*<%=p2%>*"* style="height: *80px*; width: *70px*;">

<%

}

%>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**viewTeacher.jsp**

<!DOCTYPE html>

<%@page import=*"sms.db.Teacher"*%>

<%@page import=*"sms.model.TeacherDao"*%>

<%@page import=*"java.util.Iterator"*%>

<%@page import=*"sms.model.StudentDao"*%>

<%@page import=*"sms.db.Student"*%>

<%@page import=*"java.util.ArrayList"*%>

<html lang=*"en"*>

<head>

<title>View Teacher</title>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<!--===============================================================================================-->

<link rel=*"icon"* type=*"image/png"* href=*"images/icons/favicon.ico"* />

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"fonts/fonts/font-awesome-4.7.0/css/font-awesome.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/animate/animate.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/select2/select2.min.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"*

href=*"vendor2/vendor/perfect-scrollbar/perfect-scrollbar.css"*>

<!--===============================================================================================-->

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/css/main.css"*>

<!--===============================================================================================-->

<div class=*"limiter"*>

<div class=*"container-table100"*>

<div class=*"wrap-table100"*>

<div class=*"table"*>

<div class=*"row header"*>

<div class=*"cell"*>Full Name</div>

<div class=*"cell"*>Regd No</div>

<div class=*"cell"*>Emailid</div>

<div class=*"cell"*>Photo</div>

</div>

<%

ArrayList<Teacher> all = TeacherDao.fetchTeacher();

Iterator it = all.iterator();

**while** (it.hasNext()) {

Teacher t1 = (Teacher) it.next();

%>

<div class=*"row"*>

<div class=*"cell"* data-title=*"Full Name"*>

<%=t1.getName()%>

</div>

<div class=*"cell"* data-title=*"Age"*>

<%=t1.getRegdno()%>

</div>

<div class=*"cell"* data-title=*"Job Title"*>

<%=t1.getEmailid()%>

</div>

<div class=*"cell"* data-title=*"Location"*>Photo</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/jquery/jquery-3.2.1.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/bootstrap/js/popper.js"*></script>

<script src=*"vendor2/vendor/bootstrap/js/bootstrap.min.js"*></script>

<!--===============================================================================================-->

<script src=*"vendor2/vendor/select2/select2.min.js"*></script>

<!--===============================================================================================-->

<script src=*"js/js/main.js"*></script>

</body>

</html>

**controller package**

Controller layer acts as an interface between View and Model. It receives requests from the View layer and processes them, including the necessary validations**.**

**AddMessage.java**

**package** sms.controller;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.sql.Timestamp;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** sms.db.ChatMessage;

**import** sms.model.ChatMessageDao;

/\*\*

\* Servlet implementation class AddMessage

\*/

@WebServlet("/sms.controller.AddMessage")

**public** **class** AddMessage **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession();

String sender = (String) session.getAttribute("emailid");

String receiver = (String) session.getAttribute("receiverEmailid");

String message = request.getParameter("message");

System.***out***.println(message);

System.***out***.println(receiver);

System.***out***.println(sender);

Timestamp time = **new** Timestamp(System.*currentTimeMillis*());

ChatMessage msg = **new** ChatMessage();

msg.setMessage(message);

msg.setReceiver(receiver);

msg.setSender(sender);

msg.setTime(time);

**int** status = ChatMessageDao.*addMessage*(msg);

**if** (status > 0) {

response.sendRedirect("chatPage.jsp");

}

}

}

**AddStudent.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.Student;

**import** sms.model.StudentDao;

@WebServlet("/sms.controller.AddStudent")

**public** **class** AddStudent **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

Student s = **new** Student();

s.setName(request.getParameter("name"));

s.setFname(request.getParameter("fname"));

s.setBranch(request.getParameter("branch"));

s.setMobileno(request.getParameter("mobileno"));

s.setDob(request.getParameter("dob"));

s.setGender(request.getParameter("gender"));

s.setEmailid(request.getParameter("emailid"));

s.setPermanent\_country(request.getParameter("permanent\_country"));

s.setPermanent\_district(request.getParameter("permanent\_district"));

s.setPermanent\_landmark(request.getParameter("permanent\_landmark"));

s.setPresent\_country(request.getParameter("present\_country"));

s.setPresent\_district(request.getParameter("present\_district"));

s.setPresent\_landmark(request.getParameter("present\_landmark"));

s.setRegdno(request.getParameter("regdno"));

s.setMname(request.getParameter("mname"));

s.setSemester(request.getParameter("semester"));

s.setAdhaarno(request.getParameter("adhaarno"));

s.setBloodgroup(request.getParameter("bloodgroup"));

s.setSession(request.getParameter("session"));

s.setPermanent\_state(request.getParameter("permanent\_state"));

s.setPermanent\_city(request.getParameter("permanent\_city"));

s.setPermanent\_pincode(request.getParameter("permanent\_pincode"));

s.setPresent\_state(request.getParameter("present\_state"));

s.setPresent\_city(request.getParameter("present\_city"));

s.setPresent\_pincode(request.getParameter("present\_pincode"));

**int** status = StudentDao.*addStudent*(s);

**if** (status > 0)

response.sendRedirect("addStudent.jsp?msg=Registration successful ");

**else**

response.sendRedirect("addStudent.jsp?msg=Registration Failed ");

}

}

**AddSubject.java  
package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.Student;

**import** sms.db.Subject;

**import** sms.model.AdminDao;

**import** sms.model.StudentDao;

@WebServlet("/sms.controller.AddSubject")

**public** **class** AddSubject **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

Subject sub = **new** Subject();

sub.setSubjectid(request.getParameter("subjectid"));

sub.setSubjectname(request.getParameter("subjectname"));

sub.setTeacherregdno(request.getParameter("teacherregdno"));

sub.setSemester(request.getParameter("semester"));

sub.setBranch(request.getParameter("branch"));

**int** status = AdminDao.*addSubject*(sub);

**if** (status > 0)

response.sendRedirect("addSubject.jsp?msg=Added Succesful");

**else**

response.sendRedirect("addSubject.jsp?msg=Added Failed");

}

}

**AddTeacher.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.Teacher;

**import** sms.model.StudentDao;

**import** sms.model.TeacherDao;

/\*\*

\* Servlet implementation class AddTeacher

\*/

@WebServlet("/sms.controller.AddTeacher")

**public** **class** AddTeacher **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

Teacher s = **new** Teacher();

s.setName(request.getParameter("name"));

s.setFname(request.getParameter("fname"));

s.setMname(request.getParameter("mname"));

s.setDepartment(request.getParameter("department"));

s.setMobileno(request.getParameter("mobileno"));

s.setDob(request.getParameter("dob"));

s.setGender(request.getParameter("gender"));

s.setEmailid(request.getParameter("emailid"));

s.setPermanent\_country(request.getParameter("permanent\_country"));

s.setPermanent\_district(request.getParameter("permanent\_district"));

s.setPermanent\_landmark(request.getParameter("permanent\_landmark"));

s.setPresent\_country(request.getParameter("present\_country"));

s.setPresent\_district(request.getParameter("present\_district"));

s.setPresent\_landmark(request.getParameter("present\_landmark"));

s.setRegdno(request.getParameter("regdno"));

s.setMname(request.getParameter("mname"));

s.setQualification(request.getParameter("qualification"));

s.setAdhaarno(request.getParameter("adhaarno"));

s.setBloodgroup(request.getParameter("bloodgroup"));

s.setJoindate(request.getParameter("joindate"));

s.setPermanent\_state(request.getParameter("permanent\_state"));

s.setPermanent\_city(request.getParameter("permanent\_city"));

s.setPermanent\_pincode(request.getParameter("permanent\_pincode"));

s.setPresent\_state(request.getParameter("present\_state"));

s.setPresent\_city(request.getParameter("present\_city"));

s.setPresent\_pincode(request.getParameter("present\_pincode"));

**int** status = TeacherDao.*addTeacher*(s);

**if** (status > 0)

response.sendRedirect("addTeacher.jsp?msg=Registration successful ");

**else**

response.sendRedirect("addTeacher.jsp?msg=Registration Failed ");

}

}

**AddTeacherMessage.java  
package** sms.controller;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.sql.Timestamp;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** sms.db.ChatMessage;

**import** sms.model.ChatMessageDao;

/\*\*

\* Servlet implementation class AddMessage

\*/

@WebServlet("/sms.controller.AddTeacherMessage")

**public** **class** AddTeacherMessage **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession();

String sender = (String) session.getAttribute("emailid");

String receiver = (String) session.getAttribute("receiverEmailid");

String message = request.getParameter("message");

System.***out***.println(message);

System.***out***.println(receiver);

System.***out***.println(sender);

Timestamp time = **new** Timestamp(System.*currentTimeMillis*());

ChatMessage msg = **new** ChatMessage();

msg.setMessage(message);

msg.setReceiver(receiver);

msg.setSender(sender);

msg.setTime(time);

**int** status = ChatMessageDao.*addMessage*(msg);

**if** (status > 0) {

response.sendRedirect("chatTeacherPage.jsp");

}

}

}

**AdminChangePassword.java**

**package** sms.controller;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** sms.db.Admin;

**import** sms.model.AdminDao;

/\*\*

\* Servlet implementation class AdminChangePassword

\*/

@WebServlet("/sms.controller.AdminChangePassword")

**public** **class** AdminChangePassword **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

PrintWriter out = response.getWriter();

response.setContentType("text/html");

// get data from presentation

String oldpassword = request.getParameter("oldpassword");

String newpassword = request.getParameter("newpassword");

String confirmpassword = request.getParameter("confirmpassword");

HttpSession session = request.getSession();

String emailid = (String) session.getAttribute("emailid");

// Set data to POJO class

Admin a = **new** Admin();

a.setEmailid(emailid);

a.setPassword(oldpassword);

**int** status = AdminDao.*validate*(a);

**if** (status > 0) {

**if** (newpassword.equals(confirmpassword)) {

a.setPassword(newpassword);

**int** result = AdminDao.*updatePassword*(a);

response.sendRedirect("adminWelcome.jsp?msg=Password Changed Sucessfully");

} **else**

response.sendRedirect("adminWelcome.jsp?msg=New Password or Confirm Password mismatch");

} **else**

response.sendRedirect("adminWelcome.jsp?msg=Invalid Old Password Entered");

}

}

**AdminForgotPassword.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.Admin;

**import** sms.model.AdminDao;

**import** sms.service.GenerateOtp;

**import** sms.service.SendPassword;

/\*\*

\* Servlet implementation class AdminForgotPassword

\*/

@WebServlet("/sms.controller.AdminForgotPassword")

**public** **class** AdminForgotPassword **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String emailid = request.getParameter("emailid");

Admin a = **new** Admin();

a.setEmailid(emailid);

**char**[] otp = GenerateOtp.*adminForgotPassword*(5);

String otp1 = String.*valueOf*(otp);

a.setOtp(otp1);

SendPassword p = **new** SendPassword();

p.sendAdminForgotPassword(a);

**int** status = AdminDao.*setOtp*(a);

response.sendRedirect("adminVerify.jsp?emailid=" + emailid);

}

}

**AdminLogin.java**

**package** sms.controller;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** sms.db.Admin;

**import** sms.model.AdminDao;

/\*\*

\* Servlet implementation class AdminLogin

\*/

@WebServlet("/sms.controller.AdminLogin")

**public** **class** AdminLogin **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String emailid = request.getParameter("emailid");

String password = request.getParameter("password");

Admin a = **new** Admin();

a.setEmailid(emailid);

a.setPassword(password);

System.***out***.println("No error...");

**int** status = AdminDao.*validate*(a);

**if** (status > 0) {

HttpSession session = request.getSession();

session.setAttribute("emailid", emailid);

response.sendRedirect("adminWelcome.jsp?msg=Login Success");

} **else** {

response.sendRedirect("index.jsp?msg=Invalid Email or Password");

}

}

}

**AdminLogout.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class AdminLogout

\*/

@WebServlet("/sms.controller.AdminLogout")

**public** **class** AdminLogout **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

HttpSession session = request.getSession();

session.invalidate();

response.sendRedirect("index.jsp?msg=Logout Successful");

}

}

**AdminStudentAttendance.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

@WebServlet("/sms.controller.AdminStudentAttendance")

**public** **class** AdminStudentAttendance **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String attendancedate = request.getParameter("attendancedate");

HttpSession session = request.getSession();

session.setAttribute("attendancedate", attendancedate);

session.setAttribute("semester", request.getParameter("semester"));

session.setAttribute("subject", request.getParameter("subject"));

session.setAttribute("branch", request.getParameter("branch"));

System.***out***.println(request.getParameter("attendancedate"));

System.***out***.println(request.getParameter("semester"));

System.***out***.println(request.getParameter("subject"));

System.***out***.println(request.getParameter("branch"));

response.sendRedirect("adminAttendance2.jsp");

}

}

**ApproveStudent.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.StudentLeave;

**import** sms.model.StudentLeaveDao;

@WebServlet("/sms.controller.ApproveStudent")

**public** **class** ApproveStudent **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String emailid = request.getParameter("emailid");

String status = request.getParameter("status");

String to = request.getParameter("to");

System.***out***.println(status);

StudentLeave sl = **new** StudentLeave();

sl.setEmailid(emailid);

sl.setTo(to);

sl.setStatus(status);

**int** result = StudentLeaveDao.*doApprove*(sl);

**if** (result > 0)

response.sendRedirect("adminViewStudentLeave.jsp");

**else**

response.sendRedirect("adminViewStudentLeave.jsp?msg=Error");

}

}

**ApproveTeacher.java package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.StudentLeave;

**import** sms.db.TeacherLeave;

**import** sms.model.StudentLeaveDao;

**import** sms.model.TeacherLeaveDao;

/\*\*

\* Servlet implementation class ApproveStudent

\*/

@WebServlet("/sms.controller.ApproveTeacher")

**public** **class** ApproveTeacher **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String emailid = request.getParameter("emailid");

String status = request.getParameter("status");

String to = request.getParameter("to");

TeacherLeave tl = **new** TeacherLeave();

tl.setEmailid(emailid);

tl.setTo(to);

tl.setStatus(status);

**int** result = TeacherLeaveDao.*doApprove*(tl);

**if** (result > 0)

response.sendRedirect("adminViewTeacherLeave.jsp");

**else**

response.sendRedirect("adminViewTeacherLeave.jsp?msg=Error");

}

}

**ChatPage.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

@WebServlet("/sms.controller.ChatPage")

**public** **class** ChatPage **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

String name = request.getParameter("name");

String emailid = request.getParameter("emailid");

HttpSession session = request.getSession();

session.setAttribute("receiverEmailid", emailid);

session.setAttribute("receiverName", name);

response.sendRedirect("chatPage.jsp");

}

}

**ChatTeacherPage.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

@WebServlet("/sms.controller.ChatTeacherPage")

**public** **class** ChatTeacherPage **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String name = request.getParameter("name");

String emailid = request.getParameter("emailid");

HttpSession session = request.getSession();

session.setAttribute("receiverEmailid", emailid);

session.setAttribute("receiverName", name);

response.sendRedirect("chatTeacherPage.jsp");

}

}

**ContactUs.java**

**package** sms.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** sms.db.Contact;

**import** sms.model.ContactDao;

/\*\*

\* Servlet implementation class ContactUs

\*/

@WebServlet("/sms.controller.ContactUs")

**public** **class** ContactUs **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

String name = request.getParameter("name");

String emailid = request.getParameter("emailid");

String message = request.getParameter("message");

Contact c = **new** Contact();

c.setName(name);

c.setEmailid(emailid);

c.setMessage(message);

**int** status = ContactDao.*register*(c);

**if** (status > 0)

response.sendRedirect("index.jsp?msg=Contact success");

**else**

response.sendRedirect("index.jsp?msg=Invalid");

}

}

**DeleteStudent.java**

**package** sms.controller;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** sms.db.Admin;

**import** sms.model.AdminDao;

**import** sms.model.StudentDao;

/\*\*

\* Servlet implementation class AdminLogin

\*/

@WebServlet("/sms.controller.DeleteStudent")

**public** **class** DeleteStudent **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String emailid = request.getParameter("emailid");

**int** status = StudentDao.*deleteStudent*(emailid);

**if** (status > 0) {

response.sendRedirect("deleteStudent.jsp?msg=Deleted");

} **else** {

response.sendRedirect("deleteStudent.jsp?msg=Error");

}

}

}

**db package**

This is the data layer which contains business logic of the system, and also represents the state of the application. It's independent of the presentation layer, the controller fetches the data from the Model layer and sends it to the View layer.

**Admin.java**

**package** sms.db;

**public** **class** Admin {

**private** String emailid;

**private** String password;

**private** String otp;

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getOtp() {

**return** otp;

}

**public** **void** setOtp(String otp) {

**this**.otp = otp;

}

}

**ChatMessage.java**

**package** sms.db;

**import** java.sql.Timestamp;

**public** **class** ChatMessage {

**private** String sender, receiver, message, status;

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**private** Timestamp time;

**public** String getSender() {

**return** sender;

}

**public** **void** setSender(String sender) {

**this**.sender = sender;

}

**public** String getReceiver() {

**return** receiver;

}

**public** **void** setReceiver(String receiver) {

**this**.receiver = receiver;

}

**public** String getMessage() {

**return** message;

}

**public** **void** setMessage(String message) {

**this**.message = message;

}

**public** Timestamp getTime() {

**return** time;

}

**public** **void** setTime(Timestamp time) {

**this**.time = time;

}

}

**ChatStudent.java**

**package** sms.db;

**public** **class** ChatStudent {

**private** String name, emailid, password, status;

**private** String sEmail, sName;

**public** String getsEmail() {

**return** sEmail;

}

**public** **void** setsEmail(String sEmail) {

**this**.sEmail = sEmail;

}

**public** String getsName() {

**return** sName;

}

**public** **void** setsName(String sName) {

**this**.sName = sName;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

}

**Contact.java**

**package** sms.db;

**public** **class** Contact {

**private** String name;

**private** String emailid;

**private** String message;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getMessage() {

**return** message;

}

**public** **void** setMessage(String message) {

**this**.message = message;

}

}

**FeedbackStudent.java**

**package** sms.db;

**public** **class** FeedbackStudent {

**private** String regdno, emailid, message;

**public** String getRegdno() {

**return** regdno;

}

**public** **void** setRegdno(String regdno) {

**this**.regdno = regdno;

}

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getMessage() {

**return** message;

}

**public** **void** setMessage(String message) {

**this**.message = message;

}

}

**FeedbackTeacher.java**

**package** sms.db;

**public** **class** FeedbackTeacher {

**private** String regdno, emailid, message;

**public** String getRegdno() {

**return** regdno;

}

**public** **void** setRegdno(String regdno) {

**this**.regdno = regdno;

}

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getMessage() {

**return** message;

}

**public** **void** setMessage(String message) {

**this**.message = message;

}

}

**Provider.java**

**package** sms.db;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**public** **class** Provider {

**public** **static** Connection getConnection() {

Connection con = **null**;

**try** {

**if** (con == **null**) {

Class.*forName*("oracle.jdbc.OracleDriver");

con = DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe", "sms", "sms");

} **else**

**return** con;

} **catch** (Exception e) {

e.printStackTrace();

}

**return** con;

}

}

**Student.java**

**package** sms.db;

**public** **class** Student {

**private** String emailid;

**private** String password;

**private** String otp;

**private** String status;

**private** String name;

**private** String regdno;

**private** String branch;

**private** String dob;

**private** String gender;

**private** String bloodgroup;

**private** String mobileno;

**private** String adhaarno;

**private** String session;

**private** String semester;

**private** String permanent\_landmark;

**private** String permanent\_city;

**private** String permanent\_district;

**private** String permanent\_state;

**private** String permanent\_pincode;

**private** String permanent\_country;

**private** String present\_country;

**private** String present\_landmark;

**private** String present\_city;

**private** String present\_district;

**private** String present\_state;

**private** String present\_pincode;

**private** String fname;

**private** String mname;

**private** String photo;

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getOtp() {

**return** otp;

}

**public** **void** setOtp(String otp) {

**this**.otp = otp;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getRegdno() {

**return** regdno;

}

**public** **void** setRegdno(String regdno) {

**this**.regdno = regdno;

}

**public** String getBranch() {

**return** branch;

}

**public** **void** setBranch(String branch) {

**this**.branch = branch;

}

**public** String getDob() {

**return** dob;

}

**public** **void** setDob(String dob) {

**this**.dob = dob;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** String getBloodgroup() {

**return** bloodgroup;

}

**public** **void** setBloodgroup(String bloodgroup) {

**this**.bloodgroup = bloodgroup;

}

**public** String getMobileno() {

**return** mobileno;

}

**public** **void** setMobileno(String mobileno) {

**this**.mobileno = mobileno;

}

**public** String getAdhaarno() {

**return** adhaarno;

}

**public** **void** setAdhaarno(String adhaarno) {

**this**.adhaarno = adhaarno;

}

**public** String getSession() {

**return** session;

}

**public** **void** setSession(String session) {

**this**.session = session;

}

**public** String getSemester() {

**return** semester;

}

**public** **void** setSemester(String semester) {

**this**.semester = semester;

}

**public** String getPermanent\_landmark() {

**return** permanent\_landmark;

}

**public** **void** setPermanent\_landmark(String permanent\_landmark) {

**this**.permanent\_landmark = permanent\_landmark;

}

**public** String getPermanent\_city() {

**return** permanent\_city;

}

**public** **void** setPermanent\_city(String permanent\_city) {

**this**.permanent\_city = permanent\_city;

}

**public** String getPermanent\_district() {

**return** permanent\_district;

}

**public** **void** setPermanent\_district(String permanent\_district) {

**this**.permanent\_district = permanent\_district;

}

**public** String getPermanent\_state() {

**return** permanent\_state;

}

**public** **void** setPermanent\_state(String permanent\_state) {

**this**.permanent\_state = permanent\_state;

}

**public** String getPermanent\_pincode() {

**return** permanent\_pincode;

}

**public** **void** setPermanent\_pincode(String permanent\_pincode) {

**this**.permanent\_pincode = permanent\_pincode;

}

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getMname() {

**return** mname;

}

**public** **void** setMname(String mname) {

**this**.mname = mname;

}

**public** String getPhoto() {

**return** photo;

}

**public** **void** setPhoto(String photo) {

**this**.photo = photo;

}

**public** String getPermanent\_country() {

**return** permanent\_country;

}

**public** **void** setPermanent\_country(String permanent\_country) {

**this**.permanent\_country = permanent\_country;

}

**public** String getPresent\_country() {

**return** present\_country;

}

**public** **void** setPresent\_country(String present\_country) {

**this**.present\_country = present\_country;

}

**public** String getPresent\_landmark() {

**return** present\_landmark;

}

**public** **void** setPresent\_landmark(String present\_landmark) {

**this**.present\_landmark = present\_landmark;

}

**public** String getPresent\_city() {

**return** present\_city;

}

**public** **void** setPresent\_city(String present\_city) {

**this**.present\_city = present\_city;

}

**public** String getPresent\_district() {

**return** present\_district;

}

**public** **void** setPresent\_district(String present\_district) {

**this**.present\_district = present\_district;

}

**public** String getPresent\_state() {

**return** present\_state;

}

**public** **void** setPresent\_state(String present\_state) {

**this**.present\_state = present\_state;

}

**public** String getPresent\_pincode() {

**return** present\_pincode;

}

**public** **void** setPresent\_pincode(String present\_pincode) {

**this**.present\_pincode = present\_pincode;

}

}

**Teacher.java**

**package** sms.db;

**public** **class** Teacher

{

**private** String emailid;

**private** String password;

**private** String otp;

**private** String status;

**private** String name;

**private** String fname;

**private** String mname;

**private** String regdno;

**private** String joindate;

**private** String dob;

**private** String gender;

**private** String bloodgroup;

**private** String mobileno;

**private** String adhaarno;

**private** String department;

**private** String qualification;

**private** String permanent\_landmark;

**private** String permanent\_city;

**private** String permanent\_district;

**private** String permanent\_country;

**private** String present\_country;

**private** String permanent\_state;

**private** String permanent\_pincode;

**private** String present\_landmark;

**private** String present\_city;

**private** String present\_district;

**private** String present\_state;

**private** String present\_pincode;

**private** String photo;

**public** String getEmailid() {

**return** emailid;

}

**public** **void** setEmailid(String emailid) {

**this**.emailid = emailid;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getOtp() {

**return** otp;

}

**public** **void** setOtp(String otp) {

**this**.otp = otp;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getMname() {

**return** mname;

}

**public** **void** setMname(String mname) {

**this**.mname = mname;

}

**public** String getRegdno() {

**return** regdno;

}

**public** **void** setRegdno(String regdno) {

**this**.regdno = regdno;

}

**public** String getJoindate() {

**return** joindate;

}

**public** **void** setJoindate(String joindate) {

**this**.joindate = joindate;

}

**public** String getDob() {

**return** dob;

}

**public** **void** setDob(String dob) {

**this**.dob = dob;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** String getBloodgroup() {

**return** bloodgroup;

}

**public** **void** setBloodgroup(String bloodgroup) {

**this**.bloodgroup = bloodgroup;

}

**public** String getMobileno() {

**return** mobileno;

}

**public** **void** setMobileno(String mobileno) {

**this**.mobileno = mobileno;

}

**public** String getAdhaarno() {

**return** adhaarno;

}

**public** **void** setAdhaarno(String adhaarno) {

**this**.adhaarno = adhaarno;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** String getQualification() {

**return** qualification;

}

**public** **void** setQualification(String qualification) {

**this**.qualification = qualification;

}

**public** String getPermanent\_landmark() {

**return** permanent\_landmark;

}

**public** **void** setPermanent\_landmark(String permanent\_landmark) {

**this**.permanent\_landmark = permanent\_landmark;

}

**public** String getPermanent\_city() {

**return** permanent\_city;

}

**public** **void** setPermanent\_city(String permanent\_city) {

**this**.permanent\_city = permanent\_city;

}

**public** String getPermanent\_district() {

**return** permanent\_district;

}

**public** **void** setPermanent\_district(String permanent\_district) {

**this**.permanent\_district = permanent\_district;

}

**public** String getPermanent\_country() {

**return** permanent\_country;

}

**public** **void** setPermanent\_country(String permanent\_country) {

**this**.permanent\_country = permanent\_country;

}

**public** String getPresent\_country() {

**return** present\_country;

}

**public** **void** setPresent\_country(String present\_country) {

**this**.present\_country = present\_country;

}

**public** String getPermanent\_state() {

**return** permanent\_state;

}

**public** **void** setPermanent\_state(String permanent\_state) {

**this**.permanent\_state = permanent\_state;

}

**public** String getPermanent\_pincode() {

**return** permanent\_pincode;

}

**public** **void** setPermanent\_pincode(String permanent\_pincode) {

**this**.permanent\_pincode = permanent\_pincode;

}

**public** String getPresent\_landmark() {

**return** present\_landmark;

}

**public** **void** setPresent\_landmark(String present\_landmark) {

**this**.present\_landmark = present\_landmark;

}

**public** String getPresent\_city() {

**return** present\_city;

}

**public** **void** setPresent\_city(String present\_city) {

**this**.present\_city = present\_city;

}

**public** String getPresent\_district() {

**return** present\_district;

}

**public** **void** setPresent\_district(String present\_district) {

**this**.present\_district = present\_district;

}

**public** String getPresent\_state() {

**return** present\_state;

}

**public** **void** setPresent\_state(String present\_state) {

**this**.present\_state = present\_state;

}

**public** String getPresent\_pincode() {

**return** present\_pincode;

}

**public** **void** setPresent\_pincode(String present\_pincode) {

**this**.present\_pincode = present\_pincode;

}

**public** String getPhoto() {

**return** photo;

}

**public** **void** setPhoto(String photo) {

**this**.photo = photo;

}

}

**model package**

This package provides all the business classes and interfaces, some of the classes information given below.

**AdminDao.java**

**package** sms.model;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**import** sms.db.Admin;

**import** sms.db.Provider;

**import** sms.db.Student;

**import** sms.db.Subject;

**public** **class** AdminDao {

**public** **static** **int** validate(Admin a) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "select \* from admin where emailid=? and password=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, a.getEmailid());

pst.setString(2, a.getPassword());

ResultSet rs = pst.executeQuery();

**if** (rs.next())

status = 1;

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** **int** setOtp(Admin a) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update admin set otp=? where emailid=? ";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, a.getOtp());

pst.setString(2, a.getEmailid());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** Admin getOtpAndPassword(String emailid) {

Admin a = **new** Admin();

**try** {

Connection con = Provider.*getConnection*();

String sql = "select password,otp from admin where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

ResultSet rs = pst.executeQuery();

**if** (rs.next()) {

a.setPassword(rs.getString(1));

a.setOtp(rs.getString(2));

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** a;

}

**public** **static** **int** updatePassword(Admin a) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update admin set password=? where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, a.getPassword());

pst.setString(2, a.getEmailid());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** **int** addSubject(Subject sub) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "insert into subject values(?,?,?,?,?)";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, sub.getSubjectid());

pst.setString(2, sub.getSubjectname());

pst.setString(3, sub.getTeacherregdno());

pst.setString(4, sub.getSemester());

pst.setString(5, sub.getBranch());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** ArrayList<Subject> fetchSubject() {

ArrayList<Subject> all = **new** ArrayList<Subject>();

**try** {

Connection con = Provider.*getConnection*();

String sql = "select \* from subject";

PreparedStatement pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

Subject s1 = **new** Subject();

s1.setSubjectid(rs.getString("subjectid"));

s1.setSubjectname(rs.getString("subjectname"));

s1.setTeacherregdno(rs.getString("teacherregdno"));

s1.setSemester(rs.getString("semester"));

s1.setBranch(rs.getString("branch"));

all.add(s1);

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** all;

}

**public** **static** **int** deleteSubject(String subjectid) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "delete from subject where subjectid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, subjectid);

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

}

**ChatMessageDao.java**

**package** sms.model;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**import** sms.db.ChatMessage;

**import** sms.db.Provider;

**public** **class** ChatMessageDao {

**public** **static** ResultSet *rs1* = **null**;

**public** **static** ResultSet *rs2* = **null**;

**public** **static** **int** addMessage(ChatMessage msg) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "insert into chatmessages (sender,receiver,message,time) values (?,?,?,?)";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(3, msg.getMessage());

pst.setString(1, msg.getSender());

pst.setString(2, msg.getReceiver());

pst.setTimestamp(4, msg.getTime());

*rs1* = pst.executeQuery();

**if** (*rs1*.next()) {

System.***out***.println("done : ");

status = 1;

}

*rs1*.close();

pst.close();

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** status;

}

**public** **static** ArrayList<ChatMessage> fetchMessage(ChatMessage msg) {

ArrayList<ChatMessage> all = **new** ArrayList<ChatMessage>();

String sql = "select \* from chatmessages where (sender=? OR sender=?) AND (receiver=? OR receiver=?) order by time asc";

**try** {

Connection con = Provider.*getConnection*();

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, msg.getReceiver());

pst.setString(2, msg.getSender());

pst.setString(3, msg.getReceiver());

pst.setString(4, msg.getSender());

*rs2* = pst.executeQuery();

**while** (*rs2*.next()) {

ChatMessage u = **new** ChatMessage();

u.setSender(*rs2*.getString(1));

u.setReceiver(*rs2*.getString(2));

u.setMessage(*rs2*.getString(3));

u.setTime(*rs2*.getTimestamp(4));

all.add(u);

}

// System.out.println("Hello now closing the ResultSet

// .........................: ");

*rs2*.close();

pst.close();

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** all;

}

}

**ChatStudentDao.java**

**package** sms.model;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**import** sms.db.ChatStudent;

**import** sms.db.Provider;

**import** sms.db.Student;

**import** sms.db.Teacher;

**public** **class** ChatStudentDao {

**public** **static** **boolean** validate(ChatStudent user) {

**boolean** status = **false**;

String sql = "select \* from student where emailid = ? and password = ?";

String sql1 = "update User1 set status=1 where email = ? and password = ?";

**try** {

Connection con = Provider.*getConnection*();

PreparedStatement pst = con.prepareStatement(sql);

PreparedStatement pst1 = con.prepareStatement(sql1);

pst.setString(1, user.getEmailid());

pst.setString(2, user.getPassword());

ResultSet rs = pst.executeQuery();

pst1.setString(1, user.getEmailid());

pst1.setString(2, user.getPassword());

ResultSet rs1 = pst1.executeQuery();

status = rs.next();

// System.out.println(status);

String name = rs.getString(1);

user.setName(name);

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** ArrayList<Student> fetchStudent() {

ArrayList<Student> all = **new** ArrayList<Student>();

String sql = "select \* from student";

**try** {

Connection con = Provider.*getConnection*();

PreparedStatement pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

Student u = **new** Student();

u.setName(rs.getString("name"));

u.setEmailid(rs.getString("emailid"));

u.setPassword(rs.getString("password"));

u.setStatus(rs.getString("status"));

all.add(u);

}

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** all;

}

**public** **static** ArrayList<Teacher> fetchTeacher() {

ArrayList<Teacher> all = **new** ArrayList<Teacher>();

String sql = "select \* from teacher ";

**try** {

Connection con = Provider.*getConnection*();

PreparedStatement pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

Teacher u = **new** Teacher();

u.setName(rs.getString("name"));

u.setEmailid(rs.getString("emailid"));

u.setPassword(rs.getString("password"));

u.setStatus(rs.getString("status"));

all.add(u);

}

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** all;

}

}

**ContactDao.java**

**package** sms.model;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** sms.db.Contact;

**import** sms.db.Provider;

**public** **class** ContactDao {

**public** **static** **int** register(Contact c) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "insert into contact values(?,?,?)";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, c.getName());

pst.setString(2, c.getEmailid());

pst.setString(3, c.getMessage());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

}

**StudentAttendanceDao.java**

**package** sms.model;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**import** sms.db.Provider;

**import** sms.db.Student;

**import** sms.db.StudentAttendance;

**public** **class** StudentAttendanceDao {

**public** **static** **int** takeAttendance(StudentAttendance sa) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "insert into studentattendance values(?,?,?,?,?,?,?,?) ";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, sa.getAttendancedate());

pst.setString(2, sa.getBranch());

pst.setString(3, sa.getSemester());

pst.setString(4, sa.getSubject());

pst.setString(5, sa.getEmailid());

pst.setString(6, sa.getRegdno());

pst.setString(7, sa.getStatus());

pst.setString(8, sa.getName());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** String getStatus(String emailid, String branch, String subject, String semester,

String attendancedate) {

String status = **null**;

**try** {

System.***out***.println("=============================");

System.***out***.println(emailid);

System.***out***.println(branch);

System.***out***.println(subject);

System.***out***.println(semester);

System.***out***.println(attendancedate);

System.***out***.println("======");

Connection con = Provider.*getConnection*();

String sql = "select status from studentattendance where emailid=? and branch=? and attendancedate=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

pst.setString(2, branch);

pst.setString(3, attendancedate);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

status = rs.getString("status");

System.***out***.println(status + "=================");

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** ArrayList<StudentAttendance> fetchStudentAttendence(String emailid) {

ArrayList<StudentAttendance> all = **new** ArrayList<StudentAttendance>();

**try** {

Connection con = Provider.*getConnection*();

String sql = "select \* from studentattendance where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

StudentAttendance sa = **new** StudentAttendance();

sa.setSubject(rs.getString("subject"));

sa.setAttendancedate(rs.getString("attendancedate"));

sa.setStatus(rs.getString("status"));

all.add(sa);

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** all;

}

}

**StudentDao.java**

**package** sms.model;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**import** sms.db.Admin;

**import** sms.db.Provider;

**import** sms.db.Student;

**import** sms.service.SendPassword;

**public** **class** StudentDao {

**public** **static** **int** addStudent(Student s) {

**int** status = 0;

SendPassword p = **new** SendPassword();

String password = p.sendStudentPassword(s);

**try** {

Connection con = Provider.*getConnection*();

String sql = "insert into student(name,fname,branch,mobileno,dob,gender,emailid,permanent\_country,permanent\_district,permanent\_landmark,present\_country,present\_district,present\_landmark,regdno,mname,semester,adhaarno,bloodgroup,ssession,permanent\_state,permanent\_city,permanent\_pincode,present\_state,present\_city,present\_pincode,password) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, s.getName());

pst.setString(2, s.getFname());

pst.setString(3, s.getBranch());

pst.setString(4, s.getMobileno());

pst.setString(5, s.getDob());

pst.setString(6, s.getGender());

pst.setString(7, s.getEmailid());

pst.setString(8, s.getPermanent\_country());

pst.setString(9, s.getPermanent\_district());

pst.setString(10, s.getPermanent\_landmark());

pst.setString(11, s.getPresent\_country());

pst.setString(12, s.getPresent\_district());

pst.setString(13, s.getPresent\_landmark());

pst.setString(14, s.getRegdno());

pst.setString(15, s.getMname());

pst.setString(16, s.getSemester());

pst.setString(17, s.getAdhaarno());

pst.setString(18, s.getBloodgroup());

pst.setString(19, s.getSession());

pst.setString(20, s.getPermanent\_state());

pst.setString(21, s.getPermanent\_city());

pst.setString(22, s.getPermanent\_pincode());

pst.setString(23, s.getPresent\_state());

pst.setString(24, s.getPermanent\_city());

pst.setString(25, s.getPresent\_pincode());

pst.setString(26, password);

status = pst.executeUpdate();

/\*

\* String sql1 =

\* "Insert into ChatStudent (name,emailid,password) values (?,?,?)";

\* PreparedStatement pst1 = con.prepareStatement(sql1);

\* pst1.setString(1,s.getName()); pst1.setString(2,s.getEmailid());

\* pst1.setString(3, s.getPassword()); pst1.executeUpdate();

\*/

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** **int** validate(Student s) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "select \* from student where emailid=? and password=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, s.getEmailid());

pst.setString(2, s.getPassword());

ResultSet rs = pst.executeQuery();

**if** (rs.next())

status = 1;

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** status;

}

**public** **static** **int** setOtp(Student a) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update student set otp=? where emailid=? ";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, a.getOtp());

pst.setString(2, a.getEmailid());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** Student getOtpAndPassword(String emailid) {

Student a = **new** Student();

**try** {

Connection con = Provider.*getConnection*();

String sql = "select password,otp from student where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

ResultSet rs = pst.executeQuery();

**if** (rs.next()) {

a.setPassword(rs.getString(1));

a.setOtp(rs.getString(2));

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** a;

}

**public** **static** **int** updatePassword(Student a) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update student set password=? where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, a.getPassword());

pst.setString(2, a.getEmailid());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** Student getDetails(String emailid) {

Student s = **new** Student();

**try** {

Connection con = Provider.*getConnection*();

String sql = "select \* from student where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

ResultSet rs = pst.executeQuery();

**if** (rs.next()) {

s.setEmailid(rs.getString("emailid"));

s.setName(rs.getString("name"));

s.setRegdno(rs.getString("regdno"));

s.setBranch(rs.getString("branch"));

s.setDob(rs.getString("dob"));

s.setGender(rs.getString("gender"));

s.setBloodgroup(rs.getString("bloodgroup"));

s.setMobileno(rs.getString("mobileno"));

s.setAdhaarno(rs.getString("adhaarno"));

s.setSession(rs.getString("ssession"));

s.setSemester(rs.getString("semester"));

s.setPermanent\_landmark(rs.getString("permanent\_landmark"));

s.setPermanent\_city(rs.getString("permanent\_city"));

s.setPermanent\_district(rs.getString("permanent\_district"));

s.setPermanent\_state(rs.getString("permanent\_state"));

s.setPermanent\_pincode(rs.getString("permanent\_pincode"));

s.setPermanent\_country(rs.getString("permanent\_country"));

s.setPresent\_city(rs.getString("present\_city"));

s.setPresent\_country(rs.getString("present\_country"));

s.setPresent\_district(rs.getString("present\_district"));

s.setPresent\_landmark(rs.getString("present\_landmark"));

s.setPresent\_pincode(rs.getString("present\_pincode"));

s.setPresent\_state(rs.getString("present\_state"));

s.setFname(rs.getString("fname"));

s.setMname(rs.getString("mname"));

s.setPhoto(rs.getString("photo"));

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** s;

}

**public** **static** **int** updateStudent(Student s) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update student set name=?,fname=?,branch=?,mobileno=?,dob=?,gender=?,permanent\_country=?,permanent\_district=?,permanent\_landmark=?,present\_country=?,Present\_district=?,Present\_landmark=?,regdno=?,mname=?,semester=?,adhaarno=?,bloodgroup=?,ssession=?,permanent\_state=?,permanent\_city=?,Permanent\_pincode=?,present\_state=?,present\_city=?,Present\_pincode=? where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, s.getName());

pst.setString(2, s.getFname());

pst.setString(3, s.getBranch());

pst.setString(4, s.getMobileno());

pst.setString(5, s.getDob());

pst.setString(6, s.getGender());

pst.setString(7, s.getPermanent\_country());

pst.setString(8, s.getPermanent\_district());

pst.setString(9, s.getPermanent\_landmark());

pst.setString(10, s.getPresent\_country());

pst.setString(11, s.getPresent\_district());

pst.setString(12, s.getPresent\_landmark());

pst.setString(13, s.getRegdno());

pst.setString(14, s.getMname());

pst.setString(15, s.getSemester());

pst.setString(16, s.getAdhaarno());

pst.setString(17, s.getBloodgroup());

pst.setString(18, s.getSession());

pst.setString(19, s.getPermanent\_state());

pst.setString(20, s.getPermanent\_city());

pst.setString(21, s.getPermanent\_pincode());

pst.setString(22, s.getPresent\_state());

pst.setString(23, s.getPermanent\_city());

pst.setString(24, s.getPresent\_pincode());

pst.setString(25, s.getEmailid());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** ArrayList<Student> fetchStudent() {

ArrayList<Student> all = **new** ArrayList<Student>();

**try** {

Connection con = Provider.*getConnection*();

String sql = "select \* from student";

PreparedStatement pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

Student s1 = **new** Student();

s1.setName(rs.getString("name"));

s1.setRegdno(rs.getString("regdno"));

s1.setEmailid(rs.getString("emailid"));

s1.setPhoto(rs.getString("photo"));

all.add(s1);

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** all;

}

**public** **static** **int** deleteStudent(String emailid) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "delete from student where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** ArrayList<Student> fetchStudentAttendence(String branch, String semester) {

ArrayList<Student> all = **new** ArrayList<Student>();

**try** {

System.***out***.println(branch);

System.***out***.println(semester);

Connection con = Provider.*getConnection*();

String sql = "select \* from student where branch=? and semester=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, branch);

pst.setString(2, semester);

ResultSet rs = pst.executeQuery();

**while** (rs.next()) {

Student s1 = **new** Student();

s1.setName(rs.getString("name"));

s1.setRegdno(rs.getString("regdno"));

s1.setEmailid(rs.getString("emailid"));

all.add(s1);

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** all;

}

**public** **static** **int** setStatus(String emailid) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update student set status=null where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** **int** setStatus1(String emailid) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update student set status='1' where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, emailid);

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

**public** **static** **int** updatePhoto(Student s, String file\_name) {

**int** status = 0;

**try** {

Connection con = Provider.*getConnection*();

String sql = "update student set photo=? where emailid=?";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, s.getPhoto());

pst.setString(2, s.getEmailid());

status = pst.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** status;

}

}

**service package**

This package contains all the classes and interfaces related to services like sms integration or mailing integration, following are some of the classes.

**GenerateOtp.java**

**package** sms.service;

**import** java.util.Random;

**public** **class** GenerateOtp {

**public** **static** **char**[] adminForgotPassword(**int** len) {

System.***out***.println("Generating OTP using random():");

System.***out***.println("Your new OTP is:");

String numbers = "123456789";

Random r = **new** Random();

**char**[] otp = **new** **char**[len];

**for** (**int** i = 0; i < len; i++) {

otp[i] = numbers.charAt(r.nextInt(numbers.length()));

}

**return** otp;

}

}

**SendPassword.java**

**package** sms.service;

**import** java.util.Properties;

**import** java.util.Random;

**import** javax.mail.Authenticator;

**import** javax.mail.Message;

**import** javax.mail.MessagingException;

**import** javax.mail.PasswordAuthentication;

**import** javax.mail.Session;

**import** javax.mail.Transport;

**import** javax.mail.internet.InternetAddress;

**import** javax.mail.internet.MimeMessage;

**import** sms.db.Admin;

**import** sms.db.Student;

**import** sms.db.Teacher;

**public** **class** SendPassword {

**public** String sendStudentPassword(Student s) {

String to = s.getEmailid();

String subject = "OTP Sent";

**char**[] msg1 = otp\_ForStudent(5);

String otp1 = String.*valueOf*(msg1);

System.***out***.println(otp1);

String message = "Dear student Please Login Using " + s.getEmailid() + "& password is:" + otp1;

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

**return** otp1;

}

**private** **char**[] otp\_ForStudent(**int** len) {

System.***out***.println("Generating OTP using random():");

System.***out***.println("Your new OTP is:");

String numbers = "123456789";

Random r = **new** Random();

**char**[] otp = **new** **char**[len];

**for** (**int** i = 0; i < len; i++) {

otp[i] = numbers.charAt(r.nextInt(numbers.length()));

}

**return** otp;

}

**public** String sendTeacherPassword(Teacher s) {

String to = s.getEmailid();

String subject = "OTP Sent";

**char**[] msg1 = otp\_ForTeacher(5);

String otp1 = String.*valueOf*(msg1);

System.***out***.println(otp1);

String message = "Dear student Please Login Using " + s.getEmailid() + "& password is:" + otp1;

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

**return** otp1;

}

**private** **char**[] otp\_ForTeacher(**int** len) {

System.***out***.println("Generating OTP using random():");

System.***out***.println("Your new OTP is:");

String numbers = "123456789";

Random r = **new** Random();

**char**[] otp = **new** **char**[len];

**for** (**int** i = 0; i < len; i++) {

otp[i] = numbers.charAt(r.nextInt(numbers.length()));

}

**return** otp;

}

**public** **void** sendAdminForgotPassword(Admin a) {

String to = a.getEmailid();

String subject = "OTP Sent";

String message = "Dear Admin Please verify OTP " + a.getOtp();

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

}

**public** **static** **void** sendAdminLoginPassword(Admin db) {

String to = db.getEmailid();

String subject = "Password Sent";

String message = "Dear Admin Your Password is " + db.getPassword();

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

}

**public** **void** sendStudentForgotOtp(Student a) {

String to = a.getEmailid();

String subject = "OTP Sent";

String message = "Dear Student Please verify OTP " + a.getOtp();

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

}

**public** **static** **void** sendStudentLoginPassword(Student db) {

String to = db.getEmailid();

String subject = "Password Sent";

String message = "Dear Student Your Password is " + db.getPassword();

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

}

**public** **void** sendTeacherForgotOtp(Teacher a) {

String to = a.getEmailid();

String subject = "OTP Sent";

String message = "Dear Teacher Please verify OTP " + a.getOtp();

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

}

}

**public** **static** **void** sendTeacherLoginPassword(Teacher db) {

String to = db.getEmailid();

String subject = "Password Sent";

String message = "Dear Teacher Your Password is " + db.getPassword();

String from = "seeree.lab.project@gmail.com";

String password = "seereeproject2018";

**try** {

// Authentication with Gmail server

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

Authenticator auth = **new** Authenticator() {

**public** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

};

Session session = Session.*getInstance*(props, auth);

// Composing the message

MimeMessage msg = **new** MimeMessage(session);

msg.setFrom(**new** InternetAddress(from));

msg.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

msg.setSubject(subject);

msg.setText(message);

// Sending message

Transport.*send*(msg);

System.***out***.println("Message delivered successfully , Check your mail ...........");

// response.sendRedirect("mail.jsp?msg=mail delivered");

} **catch** (MessagingException e) {

// **TODO**: handle exception

// throw new RuntimeException(e);

e.printStackTrace();

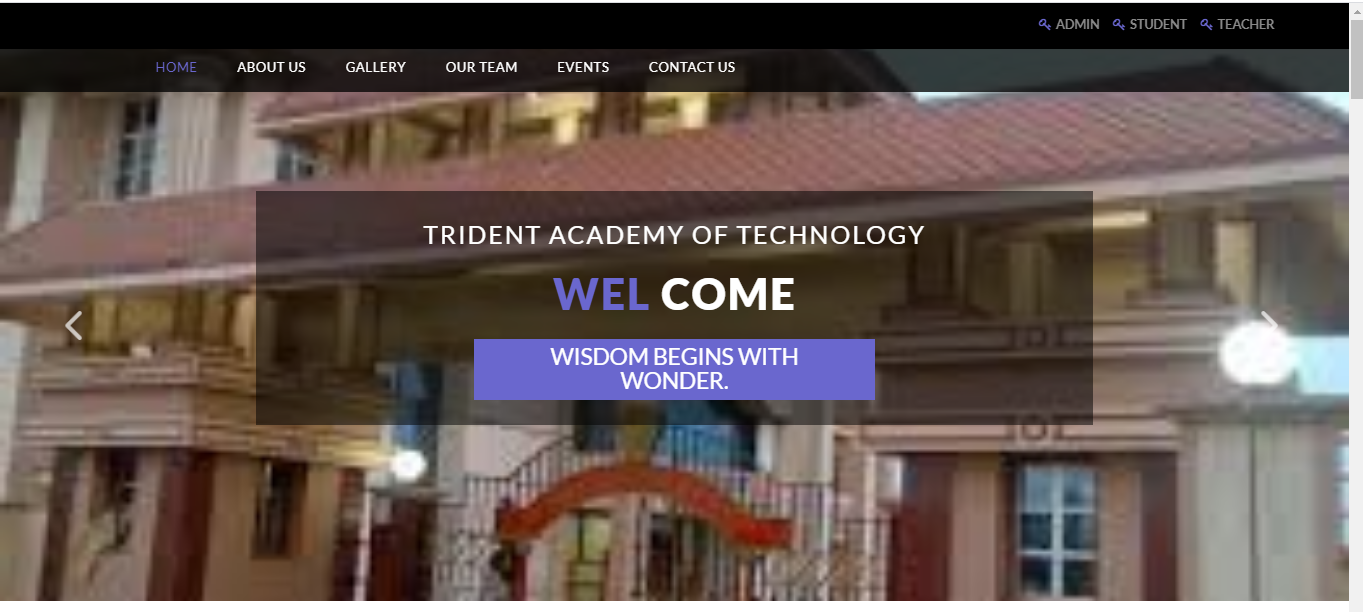
}

}

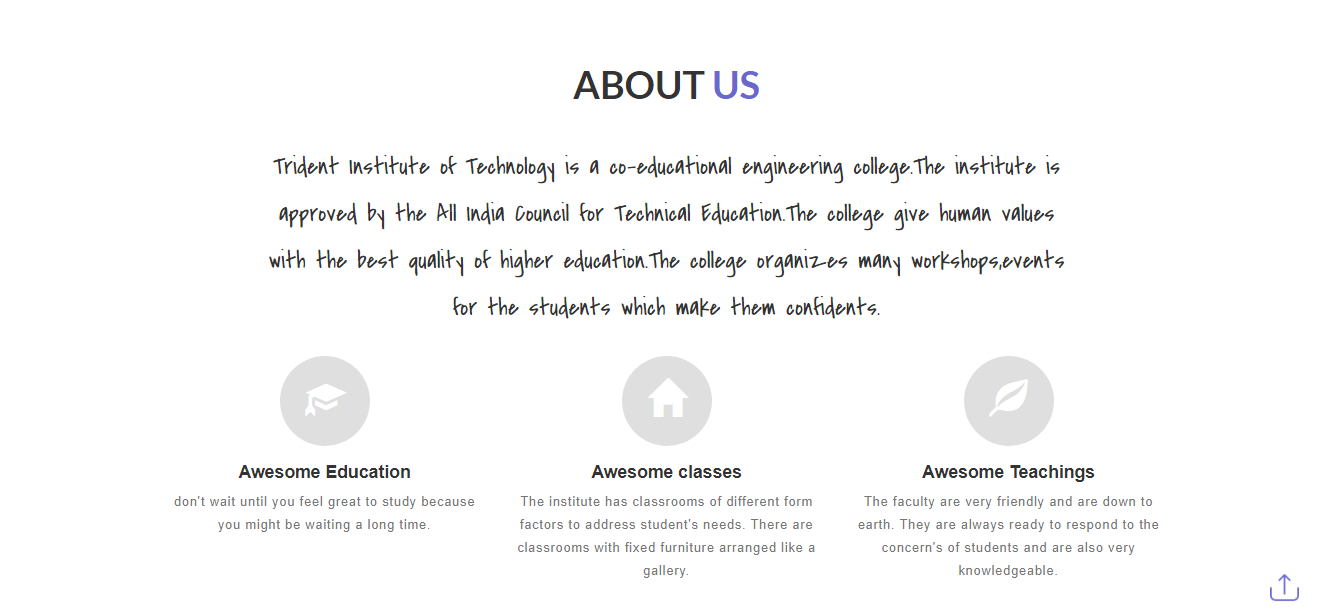
}

**Screen Shots**

**Home Page**

****

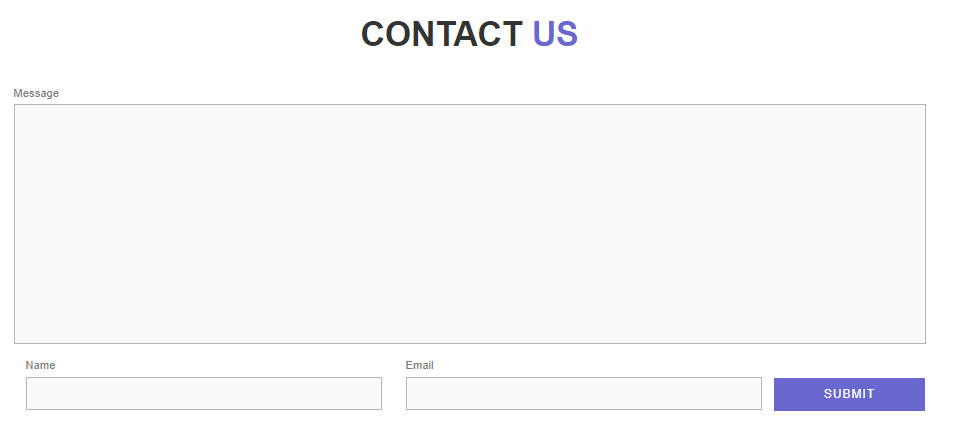
**About Us Page**



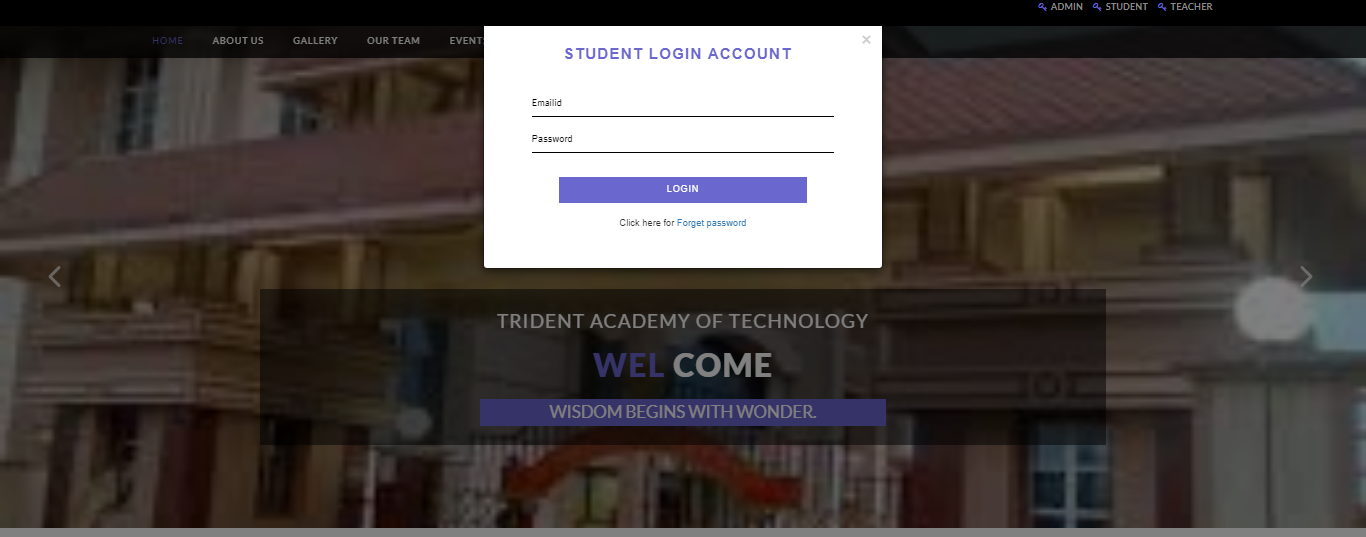
**Gallary Page**



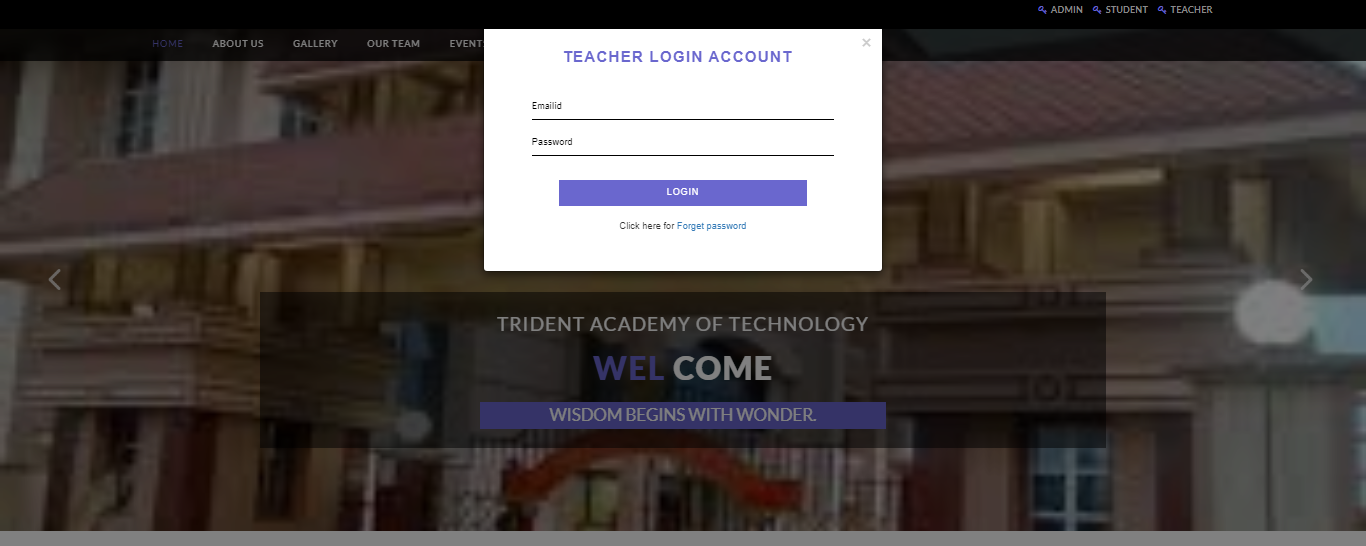
**Contact Us Page**



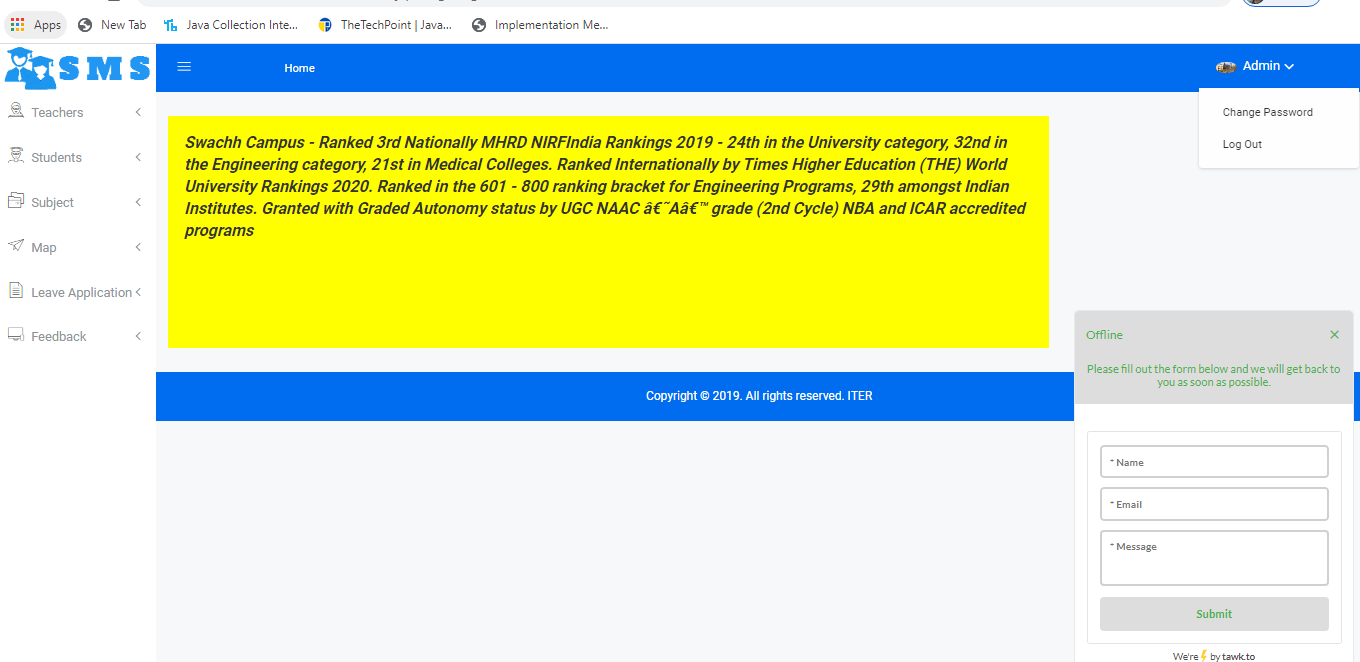
**Student\_Login Page**

****

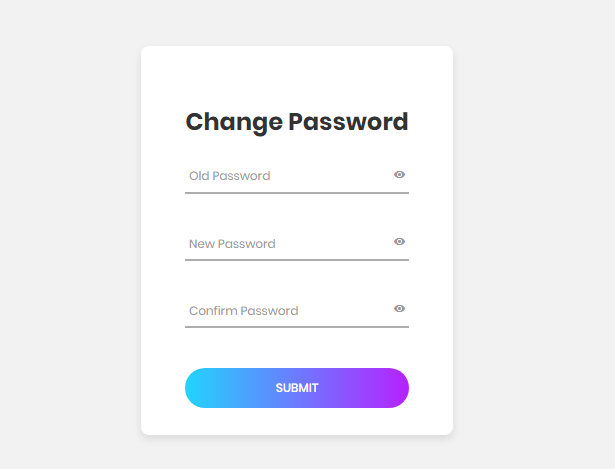
**Teacher\_Login Page**

****

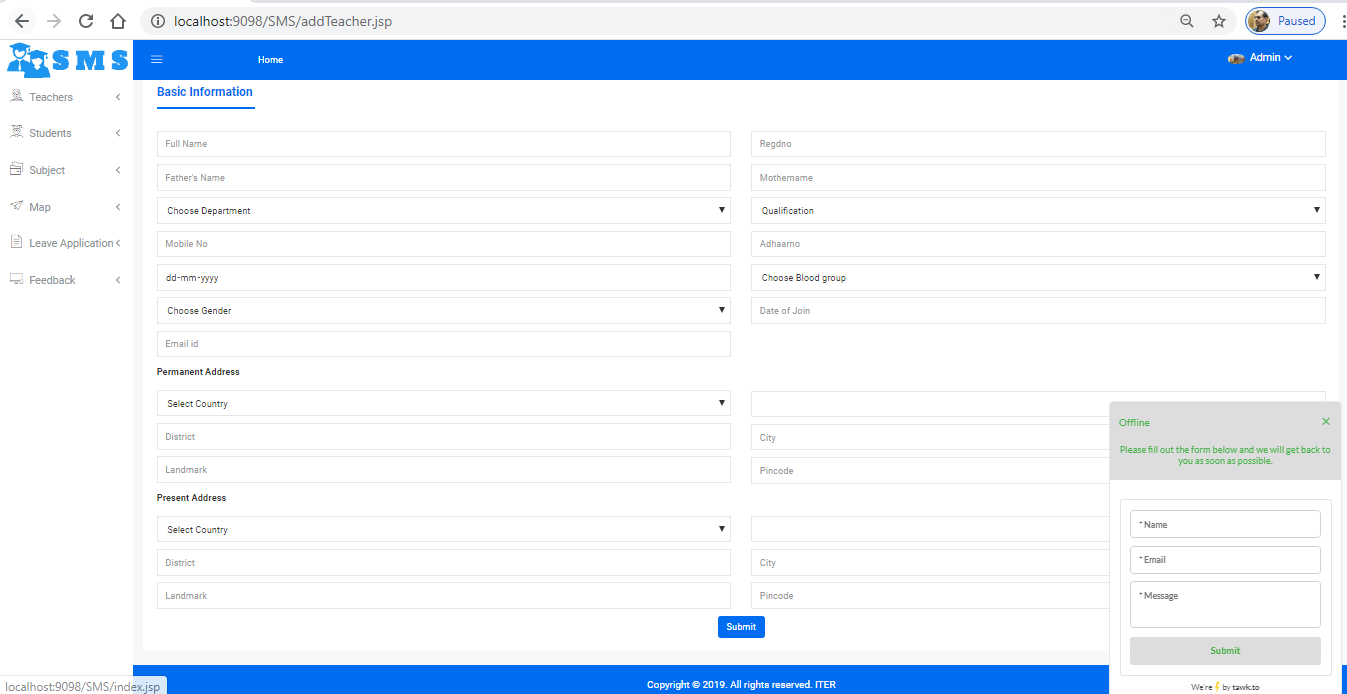
**AdminWelcome Page**



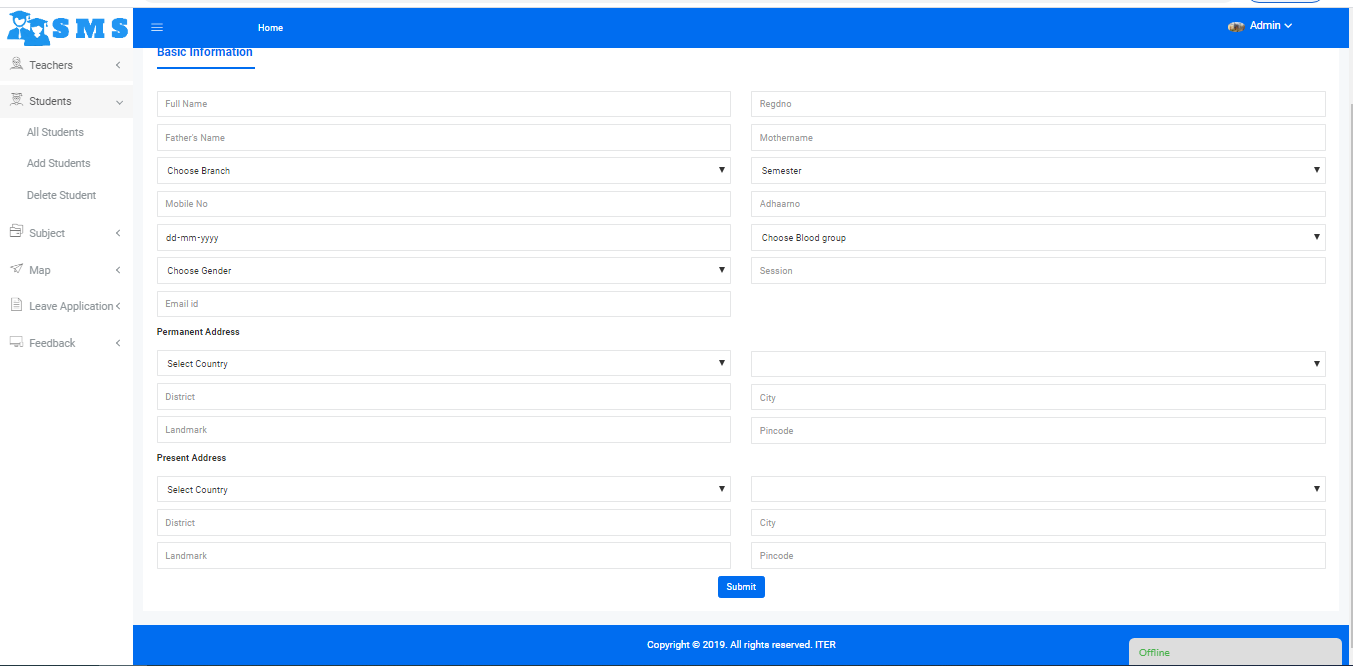
**adminChangePassword Page**



**Add Teacher Page**



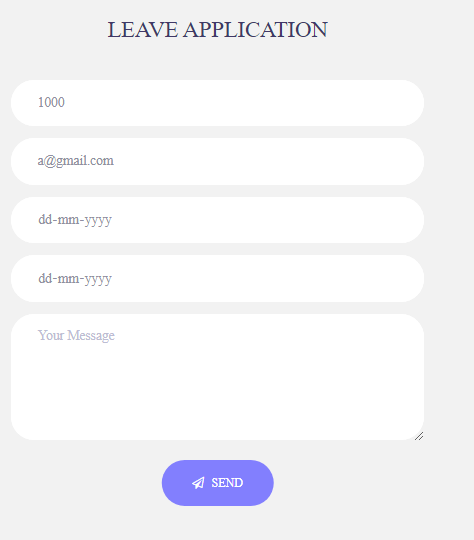
**Add Student Page**



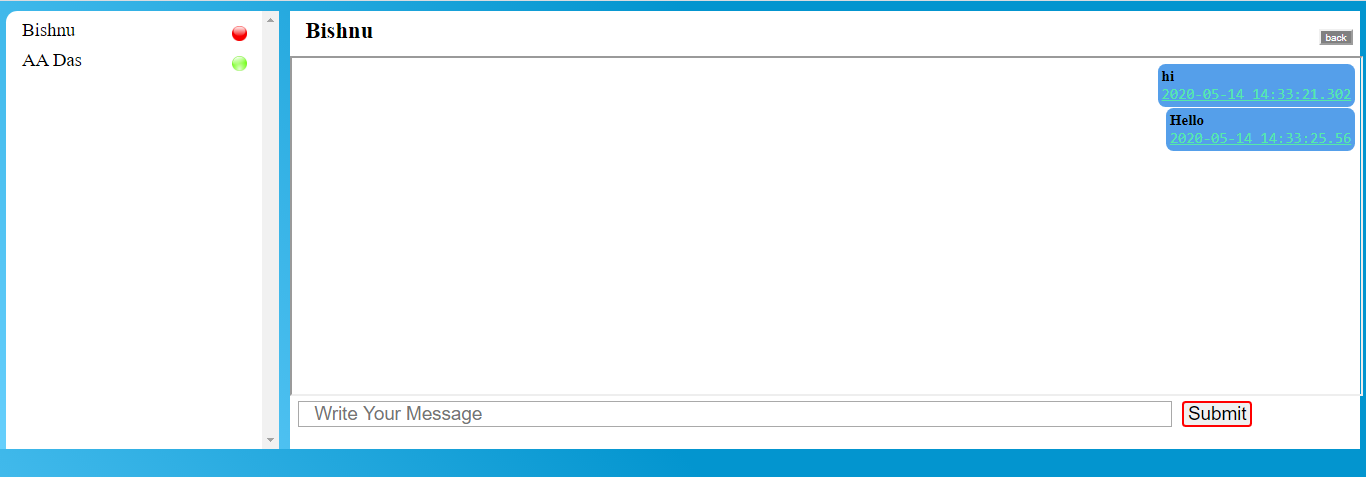
**Edit Student Profile Page**



**Leave Application Page**



**Chat Page**



**Feasibiseereey study**

Feasibiseereey study is a process to check possibiseereeies of system development. It is a method to check various different requirements and availabiseereey of financial & technical resources. Before starting the process, various parameters must be checked like:

* Estimated finance is there or not?
* The man power to operate the system is there or not?
* The man power is trained or not?

All the above conditions must be satisfied to start the project. This is why in-depth analysis of feasibiseereey is carried out. There are three different ways feasibiseereey can be tested

1. Economical Feasibiseereey
2. Technical Feasibiseereey
3. Operational Feasibiseereey.

**Economical Feasibiseereey**

In economical feasibiseereey, analysis of the cost of the system is carried out. The system should be only developed if it is going to give returned the current manual system user can get the price only by purchasing the newspapers. In addition, if he/she wants to see archives of particular equity then he has to refer to all the old newspapers. For research reports he has to buy another magazine. So Instead of buying no of magazines user has to just go online and with a single click he can get whatever information he wants. So, our project of online share news passes the test of economical feasibiseereey.

**Technical Feasibiseereey**

It is basically used to see existing computer, hardware and software etc, weather it is sufficient or additional equipments are required? Minimum System Requirement is such that it can be affordable by of the user who is having computer. All the user requires is compatible browser and .net framework installed so our system is fully technical feasible.

**Operational Feasibiseereey**

Once the system is designed there must be trained and expert operator. If there are not trained, they should given training according to the needs of the system. From the user’s perspective our system fully operational feasible as it just requires some knowledge of computer. Operators only need add daily prices of various equities and there are enough validations available so operator does not require any special technical knowledge. So, our system also passes the test of operational feasibiseereey.

**TESTING STRATEGY**

Test More and Test Frequent is organization ‘s tagline for testing. A typical screen in asp.net is tested at four levels before it goes for production. Level 1 is generally the work to be tested by other developers or other interns (this is typical first level of testing where focus is not on requirement but end user testing) Ratio: 0% end user: 100% Technical Level 2 is level where a senior programmer comes into the testing cycle of the screen that was unit tested by the developer in this phase the onus is to test software for technical requirements specified.

Ratio: 80% Technical: 20% end user Level 3 is where a tester will come into picture. The tester will test the software for both end user as well as technical point of view. The ratio here is: 50% Technical: 50% end user Level 4 is where we make the code at Release-Ready. Here screen is tested to the core and each and every standard must be followed and verified. Ratio here is: 80% User Testing – 20% Technical

This allows us to text a screen at four levels and at the end of four weeks when the screen goes to production, it is generally bug free because more people have looked at this screen from different viewpoints. References available while testing

* Project Plan.
* System Requirements specifications.
* High Level design document.
* Detail design document.
* Development and Test process standards.
* Methodology.
* Low level design

Also, organization has Separate module to store all bugs So, each screen is released for testing as a build and all information for that screen (till release) is maintained using this particular build.

**CONTENT TESTING**

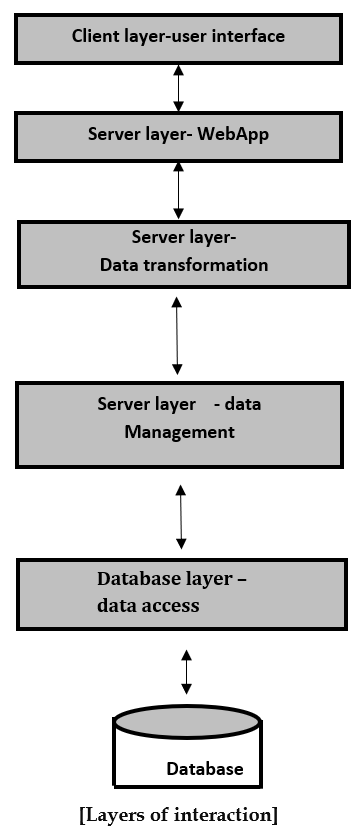
Errors in Web Application content can be as trivial as minor typographical error as incorrect information, improper organization or validation of intellectual property laws. Content Testing attempt to uncover this and many other problems before the user encounter them. There are three types of objectives. To uncover syntactic errors in text-based documents, graphical representation and other media.

* To uncover semantic errors in any content object represented as navigation occurs, and
* To find errors in organization or structure of content that is presented to the end-user

**DATABASE TESTING**

Modern Web Application does much more than present static content objects. In many application domains, Web Application interface with sophisticated database management system and build dynamic content object that are created in real time using the data acquired from a database. Database Testing for Web Application is complicated by a variety of factor.

1. The original client-side request for information is rarely presented in the form that can be input to a database management system.
2. The database may be remote to the server that houses the Web application.
3. RAW data acquired from the database must be transmitted to the Web application Server and properly formatted for subsequent transmittal to the client.
4. The dynamic content objects must be transmitted to the client in a form that can be displayed to the end user.



In figure testing should be ensure that

1. valid information is passed between the client and server from the interface layer
2. The Web application process script correctly and properly extract or formats user data.
3. Queries are passed to a data management layer that communicates with database access routines.
4. User data are passed correctly to a server-side data transformation function that format appropriate queries.

**INTERFACE TESTING**

Interface design model is reviewed to ensure that generic quaseereey criteria established for all user interfaces have been achieved and that application specific interface design issue has been properly addressed.

**Interface testing strategy**

The overall strategy for interface testing is to

1. Uncover error related to specific Interface mechanisms
2. uncover errors in the way the interface implements the semantics of navigation, Web Application functionaseereey, or content display. to accomplish this strategy, a number of objectives must be achieved

Interface futures are tested to ensure that design rules, aesthetics, and related visual content are available for the user without error. Individual interface mechanisms are tested in a manner that is a logous to unit testing For example, tests are designed to exercise all forms, client-side scripting, dynamic HTML.Each interface mechanism is tested within the context of a use-case or NSU for a specific user category The interface is tested within a variety of environments to ensure that it will be compatible.

**Testing Interface Mechanisms**

When a user interacts with a Web Application, the interaction occurs through one or more interface mechanisms

**Links:** Each link is tested to ensure that the proper content object or Function is reached. The Web engineer builds a list of all links associated with interface layout. and then executes each individually.

**Forms:** At a microscopic level, tests are performed to ensure that Labels correctly identified fields within the form and that mandatory fields are identified visually for the user. The server receives all information content within the form and their no data are lost in the transmission between client and serverAppropriate defaults are used when the user does not select from a pull down menu or set of buttons.Browser function don’t corrupt data enter in a form and Scripts that perform error checking on data entered work Properly and provide meaningful error message.

**Client-side scripting:** Black box tests are conducted to uncover any error in processing as the script is executed. These tests are coupled with forms testing because script input is often derived from data provided as part of forms processing

**Dynamic HTML:** Each Web page that contain dynamic HTML is executed to ensure that the dynamic display is correct. In addition, a compatibiseereey test should be conducted to ensure that the dynamic HTML is work properly in the environmental configuration that support the Web application.

**Application specific interface mechanisms:** Test conforms to a checklist of functionaseereey and features that are defined by the interface mechanism. Boundary test minimum and maximum number of items that can be placed in to shopping chart. Test to determine persistence of shopping chart contents. Test to determine whether the Web Application can be record shopping chart content at some future date.

**COMPATIBISEEREEY TESTING**

Web application must operate within environment that differs from one another. Different computer, display device, OS, browser and network connection speed can have significant on Web application operation. Different browser some time produced slightly different results, regardless of the degree of HTML standardization within the Web application. The Web Engineering team derives a series of compatibiseereey, validation tests, derived from existing interface tests, navigation tests, performance tests and security tests.

**TESTING METHODS**

Testing presents an interesting anomaly for the software engineering activities; the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test case that are initiated to "demolish" the software that has been build. Infect, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive.

**Models of Testing:**

There are different Models of testing. On the basis of testing methods there are two types of testing

1. Black-box testing.
2. White-box testing

Black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that integrity of external information is maintained. White-box tests are used to examine the procedural details. It checks the logical paths by test case. It can also check the conditions, loops used in the software coding. It checks that loops are working correctly on defined boundary value.

**WHITE-BOX TESTING**

White-box testing some times called glass-box testing, is a test case design method that users the control structure of the procedural design to drive the test case. Always we are thinking that there is no necessary to execute or checks the loops and conditions. And so large number of errors is uncovered. With using white-box testing methods, we have checked that; All independent paths within a function have been executed at least once. All logical decisions on their true and false side. A11 loops working correctly at their boundary values and within their specified conditions. In our coding we test that all the loops work truly in each module. The one technique of white-box testing is basis path testing. It contains two parts; one is flow graph notation and the second is cyclometer complexity. In flow graph notation we are checking logical control of flow. By using cyclometer complexity, we find complexity of our project structure.

**BLACK-BOX TESTING**

Black-box testing focuses on the functional requirements of the software. That is black-box testing enables the software engineer to drive sets of input conditions that will fully exercise all functional Requirements for the program. Black-box testing is not an alternative to white-box testing techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods. We use in our coding to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in database
* Performance errors
* Initialization and termination errors

Unlike white-box testing, which is performed earlier in the testing process, black-box testing tends to be applied during later stages of testing. Because black-box testing purposely disregards control structure, attention is focused on the information domain. By applying black-box techniques, we derive a set of test cases that satisfy following criteria. Test cases that reduce, by a count that is greater than one, the number of additional test cases must be designed to achieve reasonable testing.

**Level 1 - Build Acceptance Tests -** Other related test cases ensure that adopters received the proper Development Release Document plus other build related information (drop point, etc.). The objective is to determine if further testing is possible. If any Level 1 test case fails, the build is returned to developers un-tested.

**Level 2 - Smoke Tests -** The objective is to determine if further testing is possible. These test cases should emphasize breadth more than depth. All components should be touched, and every major feature should be tested briefly by the Smoke Test. If any Level 2 test case fails, the build is returned to developers un-tested.

**Level 2a - Bug Regression Testing -**Every bug that was “Open” during the previous build, but marked as “Fixed, Needs Re-Testing” for the current build under test, will need to be regressed, or re-tested. Once the smoke test is completed, all resolved bugs need to be regressed. It should take between 5 minutes to 1 hour to regress most bugs

**Level 3 - Critical Path Tests -** Critical Path test cases must pass by the end of every 2-3 Build Test Cycles. They do not need to be tested every drop, but must be tested at least once per milestone. Thus, the Critical Path test cases must all be executed at least once during the Iteration cycle, and once during the Final Release cycle.

**Level 4 - Standard Tests -** Test Cases that need to be run at least once during the entire test cycle for this release. These cases are run once, not repeated as are the test cases in previous levels. Functional Testing and Detailed Design Testing (Functional Spec and Design Spec Test Cases, respectively). These can be tested multiple times for each Milestone Test Cycle (Iteration, Final Release, etc.).

Standard test cases usually include Installation, Data, GUI, and other test areas.

**Bug Regression**

Bug Regression will be a central tenant throughout all testing phases. When a Severity 1 bug fails regression, adopters Testing team should also put out an immediate email to development. The Test Lead will be responsible for tracking and reporting to development and product management the status of regression testing

**FUTURE SCOPE**

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

* Discontinue of particular student eliminate potential attendance.
* Bar code Reader based attendance system.
* Individual Attendance system with photo using Student login.
* Talent management of students based on their performance evaluation can be added.
* Online class functionaseereey can be added.
* Online exam functionaseereey can be added.
* Online resume builder functionaseereey can also be added.

**CONCLUSION**

This paper assists in automating the existing manual system. This is a paperless work. It can be monitored and controlled remotely. It reduces the man power required. It provides accurate information always. Malpractice can be reduced. All years together gathered information can be saved and can be accessed at any time. The data which is stored in the repository helps in taking intelligent decisions by the management. So, it is better to have a Web Based Information Management system. All the stakeholders, faculty and management can get the required information without delay. This system is essential in the colleges/hostels and universities

**BIBLIOGRAPHY**

* CORE JAVA (A Complete Platform for Application Development) by Bishnu Charan Barik
* JAVA The Complete Reference Book (Herbert Schildth,2011)
* An Engineering approach to software engineering (Pankaj Jalote)
* Database design and Organization (Martin)
* HTML & CSS,5th Edition (Thomas Powell,2010)
* SQL Server 2005(Wrox Publication)
* [**www.udemy.com**](http://www.udemy.com)
* [**www.** **khanacademy.com**](http://www.jeevansathi.com)
* [**www.coursera.com**](http://www.coursera.com)
* [**www.open2study.com**](http://www.open2study.com)