Software Design & Architecture

Project Report

For

School Management System

Group Members:

M. Ahsan Nadeem 10812 (Leader)

M. Muneeb 10731

Abstract:

Overview

Our Management System is developed for the Schools to ensure all academic processes or management in an organized manners. The basic objective of this system is to facilitate the administrative and students. The system created to solve the problems and to provide a standardized means for the students. The basic purpose to of school management system is to make easy to management to store large amount of data/information as it is difficult to manage that much data manually or in papers.

Problems Statement/Disadvantages of not using Management Systems

The current problem in high school in Pakistan is that the management didn't have systematic data arrangement.

- Lack Of Data Arrangement that is record by using manual system (paper, registers, etc.) to record the students information, result and performance.
- The manual system is difficult to search or maintain properly about the students information.
- Some information released by the school is not known by the parents or teachers. (notices, etc.).

Objective:

The objectives of this system are:

- To record all the student academic information for future reference and enhancement and to update/search student record including personal information, result, performance, etc.
- To record all the Employee information for future reference and enhancement and to update/search employee record including personal information, performance, etc.
- To notify through SMS,MAIL to parents, students and teachers about the occurring meeting or activity.

It also perform all the functionalities and all the complex functions that is required for the schools to work more efficiently and fast.

Introduction

School Management System is a management project which is very important for the current educational institutes as this systems allows parents / guardian to check real-time progress of their children. SMS offers many features that help to enhance the performance of schools with minimum efforts. Due to this schools avoids all the paper work that is difficult to maintain.

Online School Management System is developing for general purpose and used to replace old paper work system and PUMS. OSMS is to build upon the existing information system PUMS in order to efficiently provide student information to teachers and school administration .This increase in efficiency of result making, provide result to parents, give feedback to student, finally, publication and email student result. It provides a mechanism to edit the student information form which makes the system flexible and efficient.

The purpose of why we are building this project is that it help the schools to overcome all the problems or difficulties that occurs while maintaining the large records on paper or manually.

This project is useful in such circumstances when there is a need to maintain students , teachers, employees record or keep track of them. This project is multi-user software as it is used by the admin as well as the students , parent , and other faculty members.

Methodology:

In this project, we use an **Architectural pattern (3-Tier Architecture)** that is explain as below

Our development methodology comprises of 3-Tier Architecture which is a type of software architecture which is composed on three layers. They are mainly used in application as a specific type of client server.

Presentation Layer:

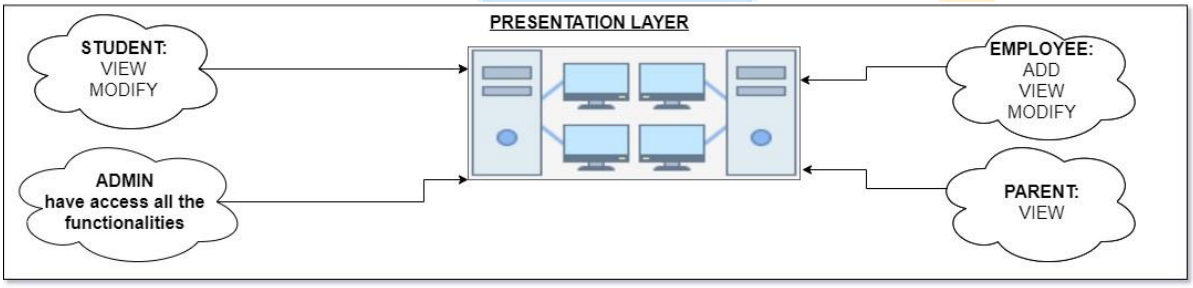
The presentation layer is front end layer in three tier system and consist of user interface. This user interface is often a graphical one accessible through a web-based app to display the content.

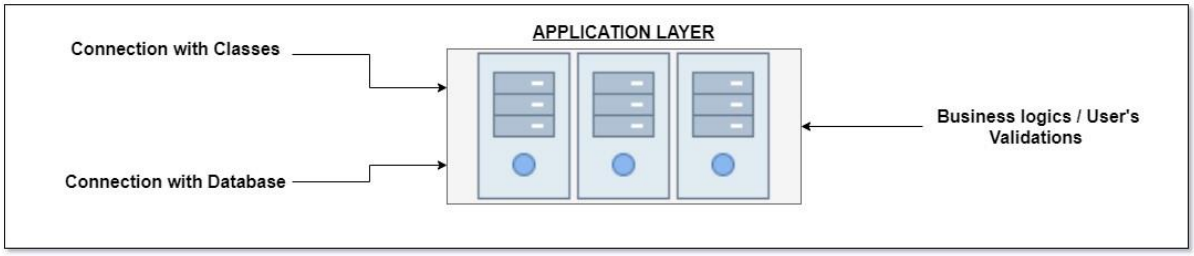
Application Layer:

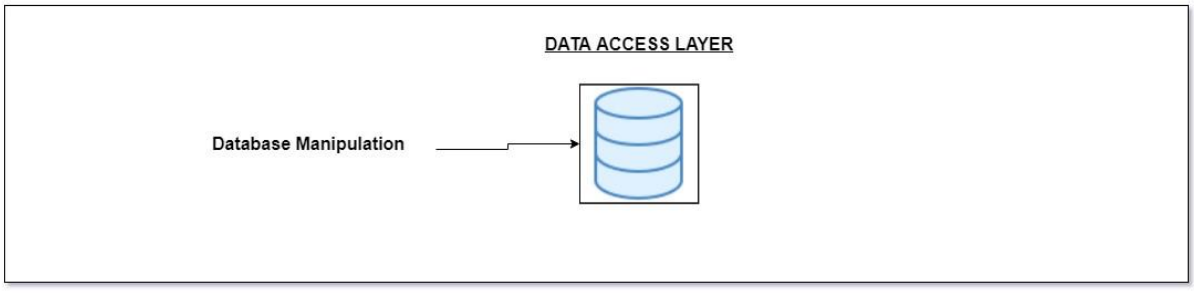
The application layer contains the functional business logics with drives an applications core capabilities. It's often written in C#. Python, Java, C++ etc.

Data Layer:

The data layer comprises of the databases / data storage system and data access layer using database query languages. Data is accessed by the application layer via API calls.







Steps Followed:

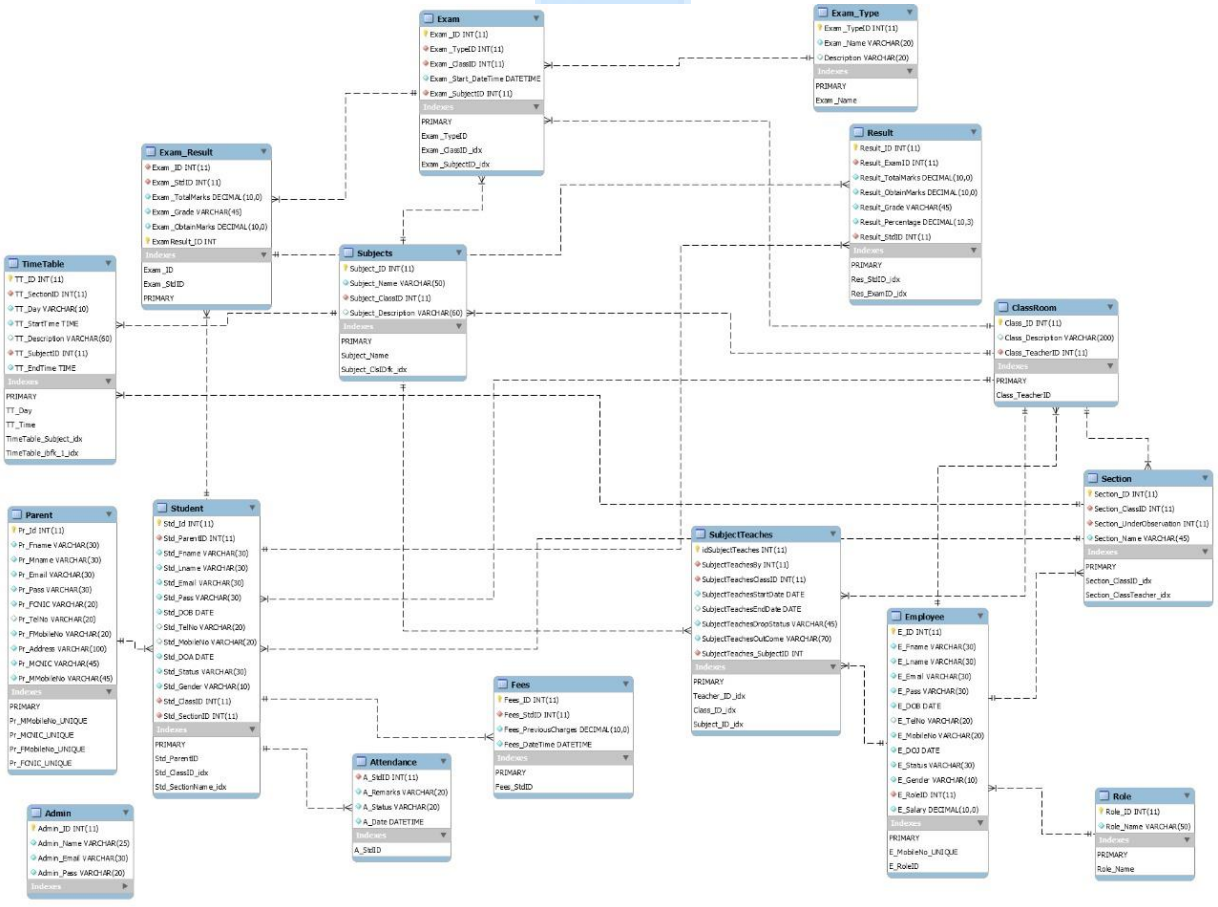
Following are the steps that is followed to build the project.

Step 01: Identifying the system requirement

Firstly, we recognize or identify all the system requirement including functional and non-functional requirements. We also identify the users who will use the functionalities of system as well as the constraints.

Step 02: Design Database

After identifying system requirement we then start designing of our database which will stores the records. This include tables , stored procedures which is used for the insert and update in respective tables. We also use view to view our table's records. By using stored procedures and views, we are able to complete our database designing more efficiently. Also by using them, the time is also saved which will be utilize in other steps.



Step 03: Identifying Design Specification

In this step we identify all the design requirements including all the design constraints. It also includes the methods or definition of how our system components interact with the interface.

Step 04: Implementing 3-Tier Architecture Pattern

After identifying all the system and design requirement, we then implemented our **Architectural pattern (3-Tier Architecture)**

Implementation of 3-Tier Architecture are as followed

Data Access Layer

In Data Access Layer, we retrieve data from our database using **SQL queries.** We create a functions in which we use SQL attributes to retrieve data from database.

For Example: Employee Data Layer

OEmployee.cs | EEmployee.cs

DAL | DAL.Entities.EEmployee

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace DAL.Entities
8   {
9       public class EEmployee
10      {
11          public int ID { get; set; }
12          public string FNAME { get; set; }
13          public string LNAME { get; set; }
14          public string EMAIL { get; set; }
15          public string PASSWORD { get; set; }
16          public DateTime DOB { get; set; }
17          public string TELEPHONE { get; set; }
18          public string MOBILENO { get; set; }
19          public DateTime DOJ { get; set; }
20          public string GENDER { get; set; }
21          public int ROLE { get; set; }
22          public Decimal SALARY { get; set; }
23          public string STATUS { get; set; }
24      }
25  }
26
```

OEmployee.cs | EEmployee.cs

DAL | DAL.EOperations.OEmployee

```
using DAL.Entities;
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DAL.EOperations
{
    public class OEmployee
    {
        string conn = "Server=192.3.73.34;Database=uhuospdn_practice;Uid=uhuospdn_sql;Pwd=rlL)~~*NJ7t(;";

        public int AddNewEmployee(EEmployee GRD)
        {
            int effectrows;
            using (MySqlConnection con = new MySqlConnection(conn))
            {
                using (MySqlCommand cmd = new MySqlCommand("AddEmployee", con))
                {
                    cmd.CommandType = CommandType.StoredProcedure;
                    cmd.Parameters.AddWithValue("@emp_id", GRD.ID);
                    cmd.Parameters.AddWithValue("@emp_fname", GRD.FNAME);
                    cmd.Parameters.AddWithValue("@emp_lname", GRD.LNAME);
                    cmd.Parameters.AddWithValue("@emp_email", GRD.EMAIL);
                    cmd.Parameters.AddWithValue("@emp_pass", GRD.PASSWORD);
                    cmd.Parameters.AddWithValue("@emp_dob", GRD.DOB);
                    cmd.Parameters.AddWithValue("@emp_telno", GRD.TELEPHONE);
                    cmd.Parameters.AddWithValue("@emp_mobno", GRD.MOBILENO);
                    cmd.Parameters.AddWithValue("@emp_doj", GRD.DOJ);
                    cmd.Parameters.AddWithValue("@emp_status", GRD.STATUS);
                    cmd.Parameters.AddWithValue("@emp_gender", GRD.GENDER);
                    cmd.Parameters.AddWithValue("@emp_roleid", GRD.ROLE);
                    cmd.Parameters.AddWithValue("@emp_salary", GRD.SALARY);
                    con.Open();
                    effectrows = cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            return effectrows;
        }

    }
```

OEmployee.cs | EEmployee.cs

DAL | DAL.EOperations.OEmployee

```
        }

        public int UpdateEmployee(EEmployee GRD, int id)
        {
            int effectrows;
            using (MySqlConnection con = new MySqlConnection(conn))
            {
                using (MySqlCommand cmd = new MySqlCommand("EditEmployee", con))
                {
                    cmd.CommandType = CommandType.StoredProcedure;
                    cmd.Parameters.AddWithValue("@emp_id", id);
                    cmd.Parameters.AddWithValue("@emp_fname", GRD.FNAME);
                    cmd.Parameters.AddWithValue("@emp_lname", GRD.LNAME);
                    cmd.Parameters.AddWithValue("@emp_email", GRD.EMAIL);
                    cmd.Parameters.AddWithValue("@emp_pass", GRD.PASSWORD);
                    cmd.Parameters.AddWithValue("@emp_dob", GRD.DOB);
                    cmd.Parameters.AddWithValue("@emp_telno", GRD.TELEPHONE);
                    cmd.Parameters.AddWithValue("@emp_mobno", GRD.MOBILENO);
                    cmd.Parameters.AddWithValue("@emp_doj", GRD.DOJ);
                    cmd.Parameters.AddWithValue("@emp_status", GRD.STATUS);
                    cmd.Parameters.AddWithValue("@emp_gender", GRD.GENDER);
                    cmd.Parameters.AddWithValue("@emp_roleid", GRD.ROLE);
                    cmd.Parameters.AddWithValue("@emp_salary", GRD.SALARY);
                    con.Open();
                    effectrows = cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            return effectrows;
        }

        public DataSet ViewEmployee(int id)
        {
            string myquery = "Select * from ViewEmployee where E_ID=" + id;
            MySqlConnection con = new MySqlConnection(conn);
            MySqlCommand cmd = new MySqlCommand();
            cmd.CommandText = myquery;
            cmd.Connection = con;
            MySqlDataAdapter da = new MySqlDataAdapter();
            da.SelectCommand = cmd;
            DataSet ds = new DataSet();
            da.Fill(ds);
            return ds;
        }

```

Business Layer

In business logic layer, we uses logics to use our data from data layer

For Example: Employee Business Layer

EOperation.cs | OEmployee.cs | EEmployee.cs

BLL | BLL.BOperations.EOperation

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DAL.Entities;
using DAL.EOperations;
using System.Data;
using MySql.Data.MySqlClient;
using MySql.Web;

namespace BLL.BOperations
{
    public class EOperation
    {
        // Handle to the Employee DBAccess class
        OEmployee EmployeeDb = null;

        public EOperation()
        {
            EmployeeDb = new OEmployee();
        }

        public DataSet GetEmployeeByID(int id)
        {
            return EmployeeDb.ViewEmployee(id);
        }

        public DataSet GetEmployeeList()
        {
            return EmployeeDb.ViewEmployeeList();
        }

        public string GetEmployeeRoleId(string name)
        {
            return EmployeeDb.GetEmployeeRoleId(name);
        }

        public string GetEmployeeRoleName(int id)
        {
            return EmployeeDb.GetEmployeeRoleName(id);
        }
```

EOperation.cs | OEmployee.cs | EEmployee.cs

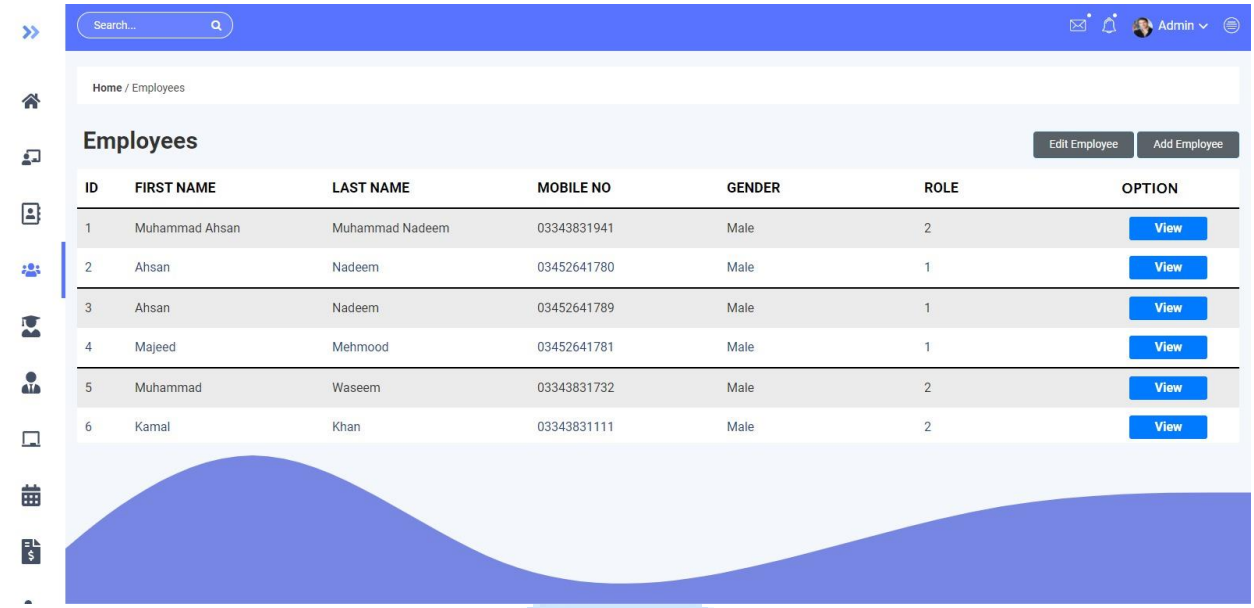
BLL | BLL.BOperations.EOperation

```
16          // Handle to the Employee DBAccess class
17          OEmployee EmployeeDb = null;
18
19          public EOperation()
20          {
21              EmployeeDb = new OEmployee();
22          }
23
24          public DataSet GetEmployeeByID(int id)
25          {
26              return EmployeeDb.ViewEmployee(id);
27          }
28
29          public DataSet GetEmployeeList()
30          {
31              return EmployeeDb.ViewEmployeeList();
32          }
33
34          public string GetEmployeeRoleId(string name)
35          {
36              return EmployeeDb.GetEmployeeRoleId(name);
37          }
38
39          public string GetEmployeeRoleName(int id)
40          {
41              return EmployeeDb.GetEmployeeRoleName(id);
42          }
43
44          public DataSet GetEmployeeRole()
45          {
46              return EmployeeDb.ViewEmployeeRole();
47          }
48
49          public int UpdateEmployee(EEmployee emp, int id)
50          {
51              return EmployeeDb.UpdateEmployee(emp, id);
52          }
53
54          public int AddNewEmployee(EEmployee Employee)
55          {
56              return EmployeeDb.AddNewEmployee(Employee);
57          }
58      }
59  }
60
```

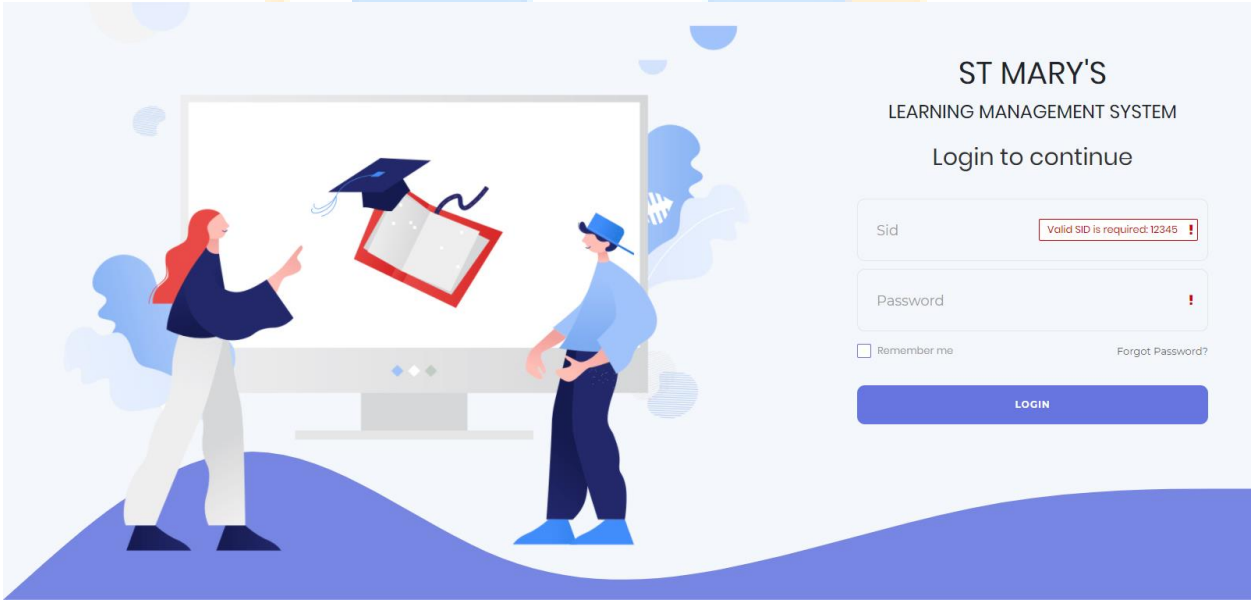
Presentation Layer

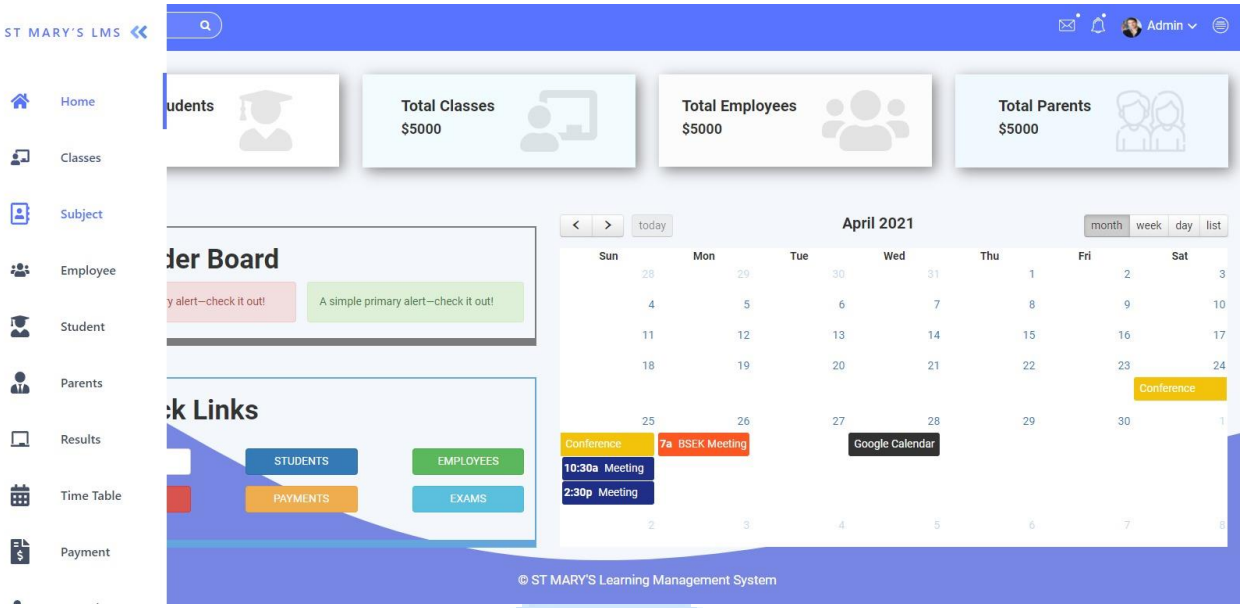
Presentation layer contains all the web pages and validation like check input is in correct format etc.

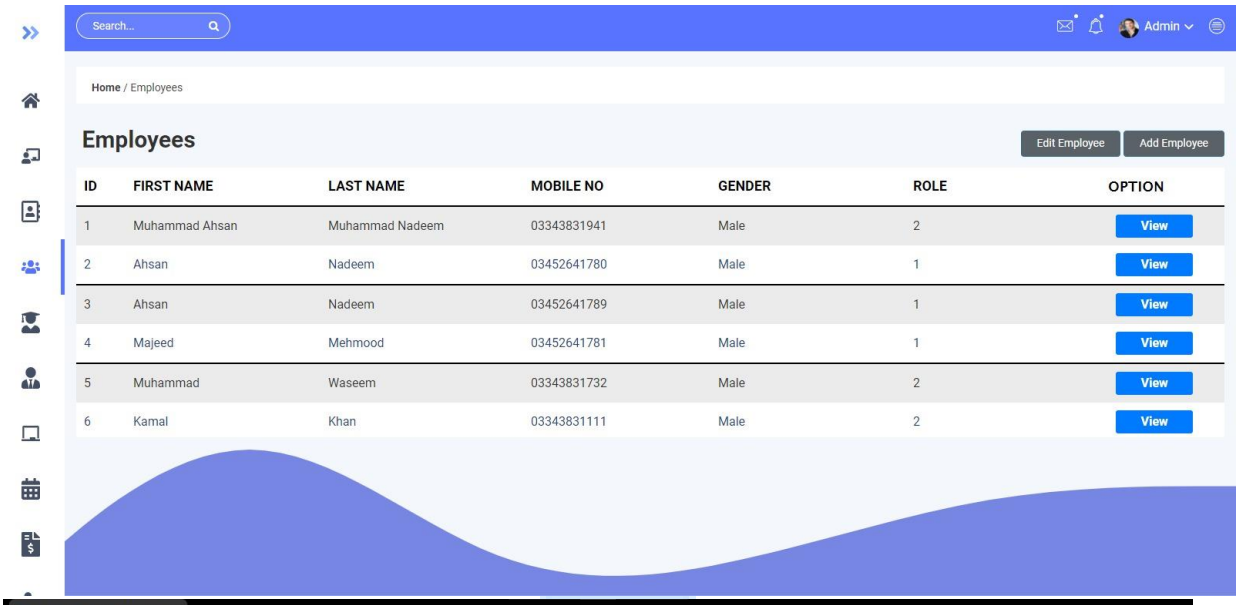
For Example: Employee Presentation Layer

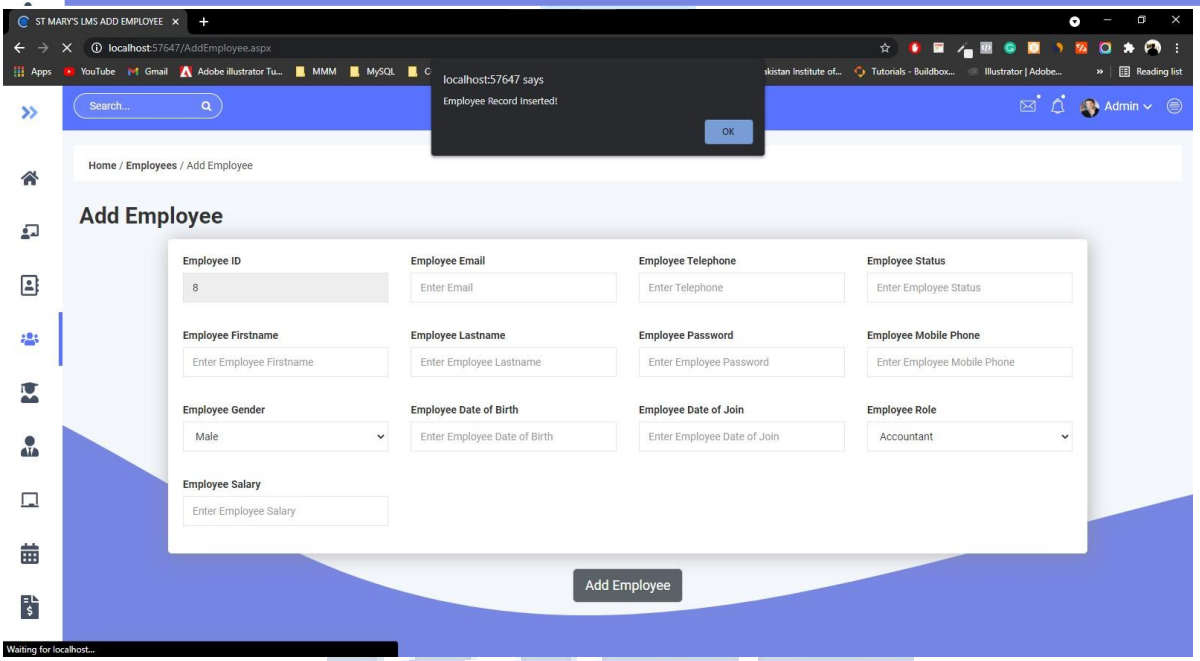

Search...
Admin
Home / Employees
Employees
Edit Employee
Add Employee
ID FIRST NAME LAST NAME MOBILE NO GENDER ROLE OPTION
1 Muhammad Ahsan Muhammad Nadeem 03343831941 Male 2 View
2 Ahsan Nadeem 03452641780 Male 1 View
3 Ahsan Nadeem 03452641789 Male 1 View
4 Majeed Mehmood 03452641781 Male 1 View
5 Muhammad Waseem 03343831732 Male 2 View
6 Kamal Khan 03343831111 Male 2 View

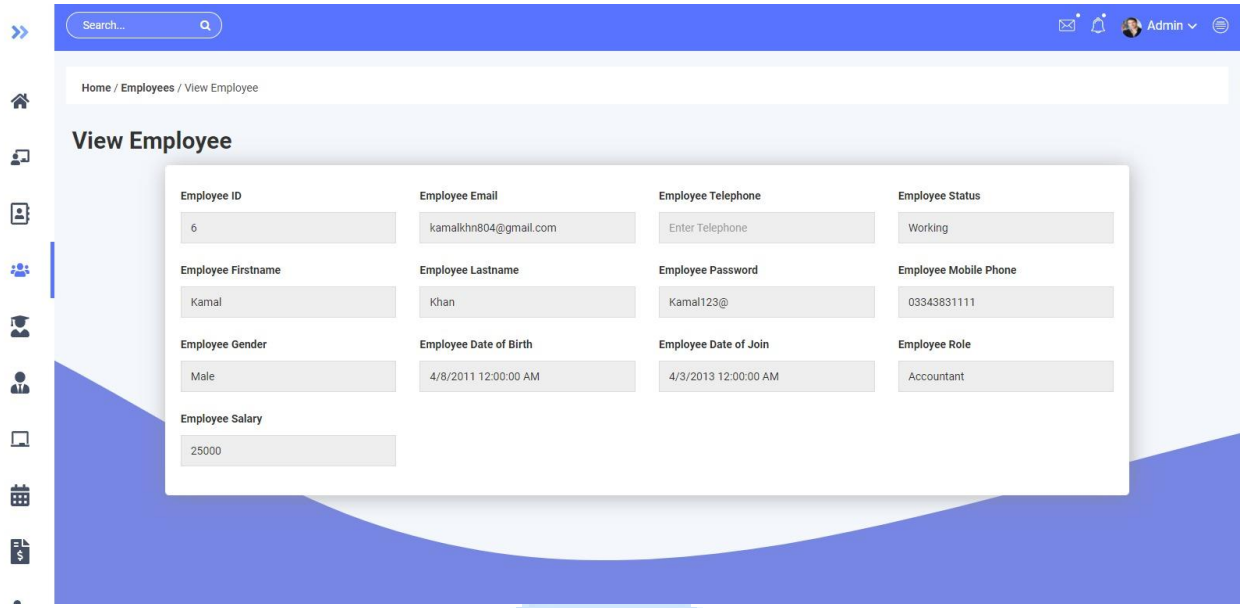
Result

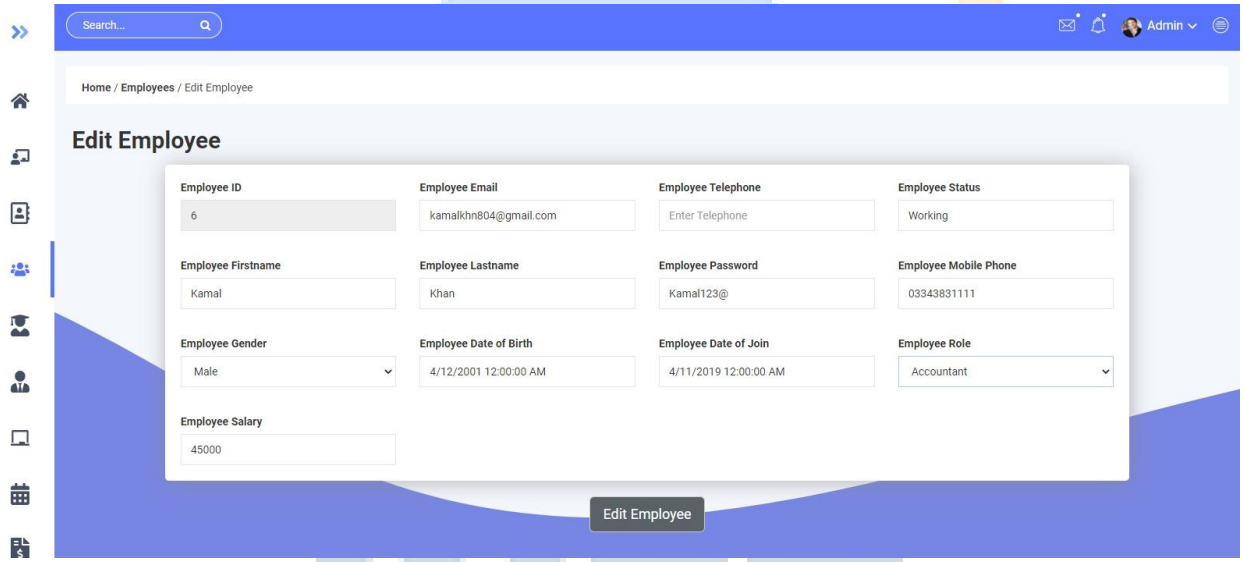

ST MARY'S
LEARNING MANAGEMENT SYSTEM
Login to continue
Sid
Valid SID is required: 12345
Password
Remember me
Forgot Password?
LOGIN


ST MARY'S LMS
Admin
Home
Classes
Subject
Employee
Student
Parents
Results
Time Table
Payment
udents
Total Classes
$5000
Total Employees
$5000
Total Parents
$5000
der Board
y alert—check it out!
A simple primary alert—check it out!
k Links
STUDENTS
EMPLOYEES
PAYMENTS
EXAMS
today
April 2021
month
week
day
list
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Conference
7a BSEK Meeting
Google Calendar
10:30a Meeting
2:30p Meeting
© ST MARY'S Learning Management System

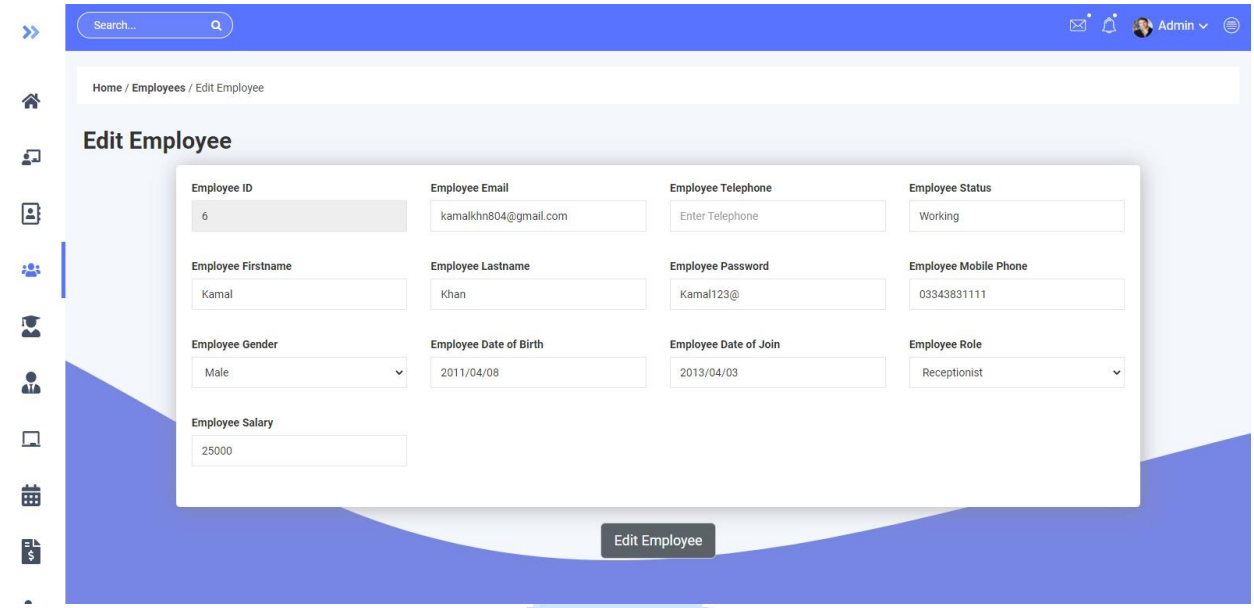

Search...
Admin
Home / Employees
Employees
Edit Employee
Add Employee
ID
FIRST NAME
LAST NAME
MOBILE NO
GENDER
ROLE
OPTION
1 Muhammad Ahsan Muhammad Nadeem 03343831941 Male 2 View
2 Ahsan Nadeem 03452641780 Male 1 View
3 Ahsan Nadeem 03452641789 Male 1 View
4 Majeed Mehmood 03452641781 Male 1 View
5 Muhammad Waseem 03343831732 Male 2 View
6 Kamal Khan 03343831111 Male 2 View


ST MARY'S LMS ADD EMPLOYEE
localhost:57647/AddEmployee.aspx
localhost:57647 says
Employee Record Inserted!
OK
Search...
Admin
Home / Employees / Add Employee
Add Employee
Employee ID
8
Employee Email
Enter Email
Employee Telephone
Enter Telephone
Employee Status
Enter Employee Status
Employee Firstname
Enter Employee Firstname
Employee Lastname
Enter Employee Lastname
Employee Password
Enter Employee Password
Employee Mobile Phone
Enter Employee Mobile Phone
Employee Gender
Male
Employee Date of Birth
Enter Employee Date of Birth
Employee Date of Join
Enter Employee Date of Join
Employee Role
Accountant
Employee Salary
Enter Employee Salary
Add Employee
Waiting for localhost...

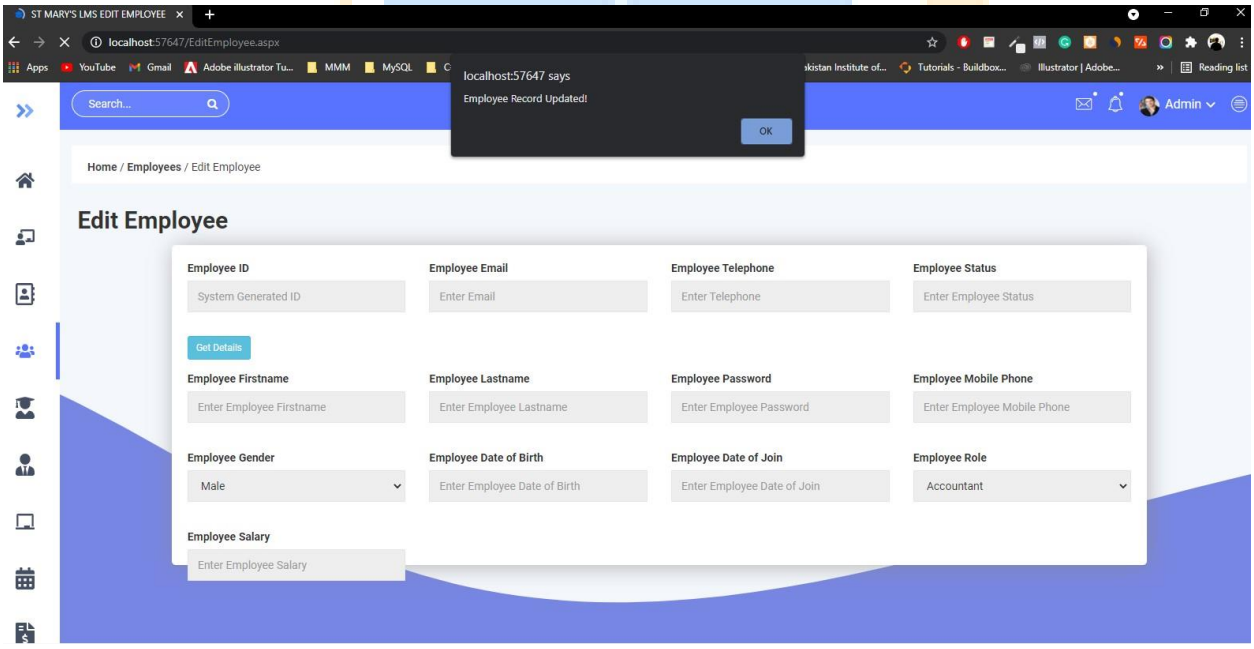


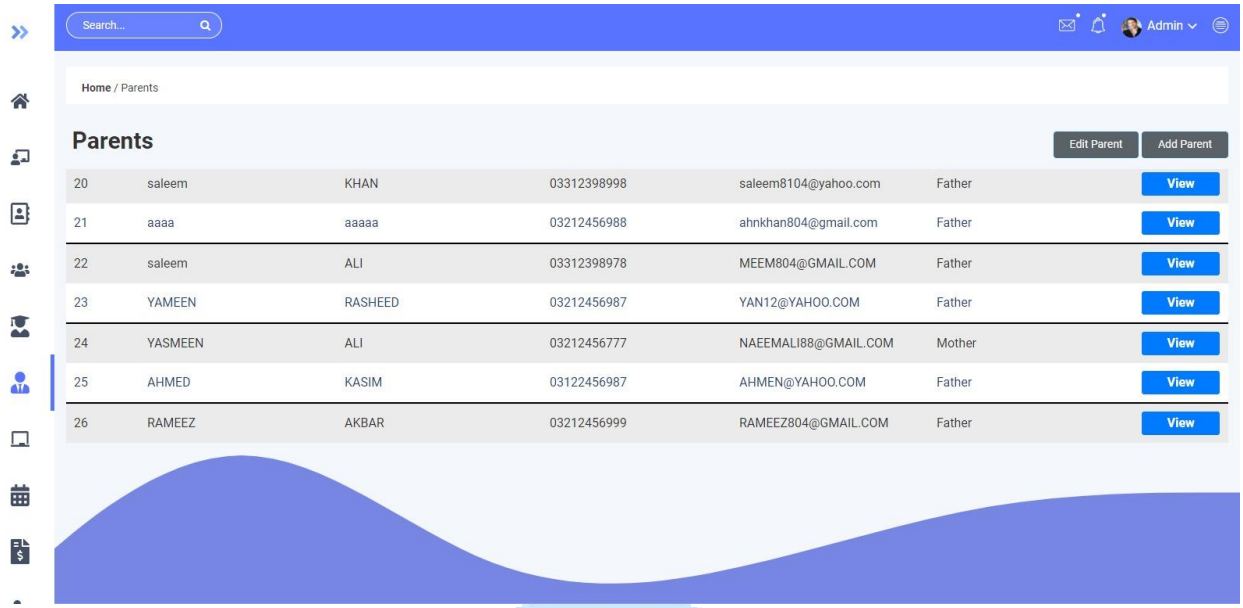


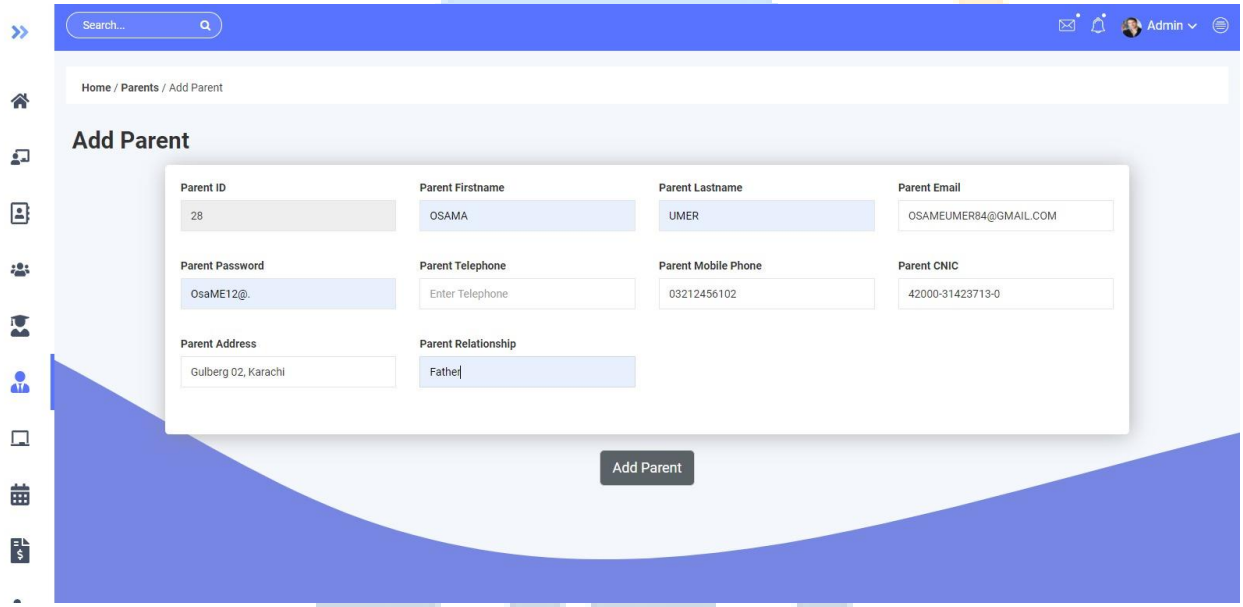
Before Update:

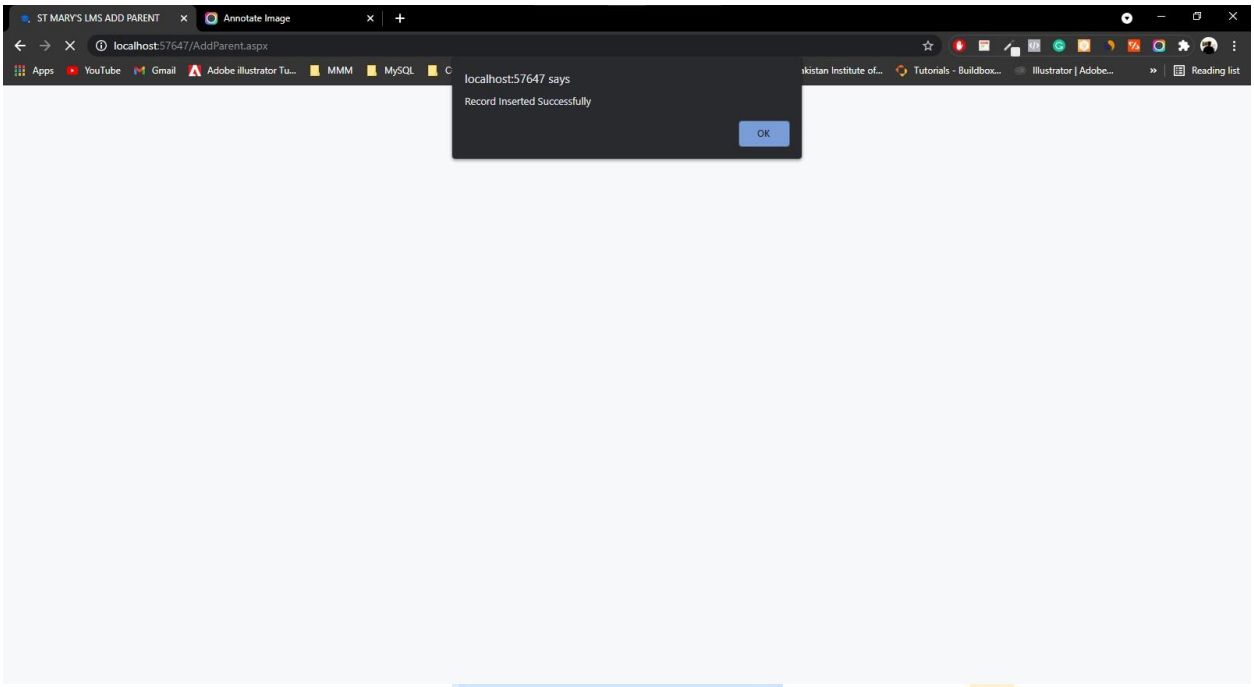


After Update:

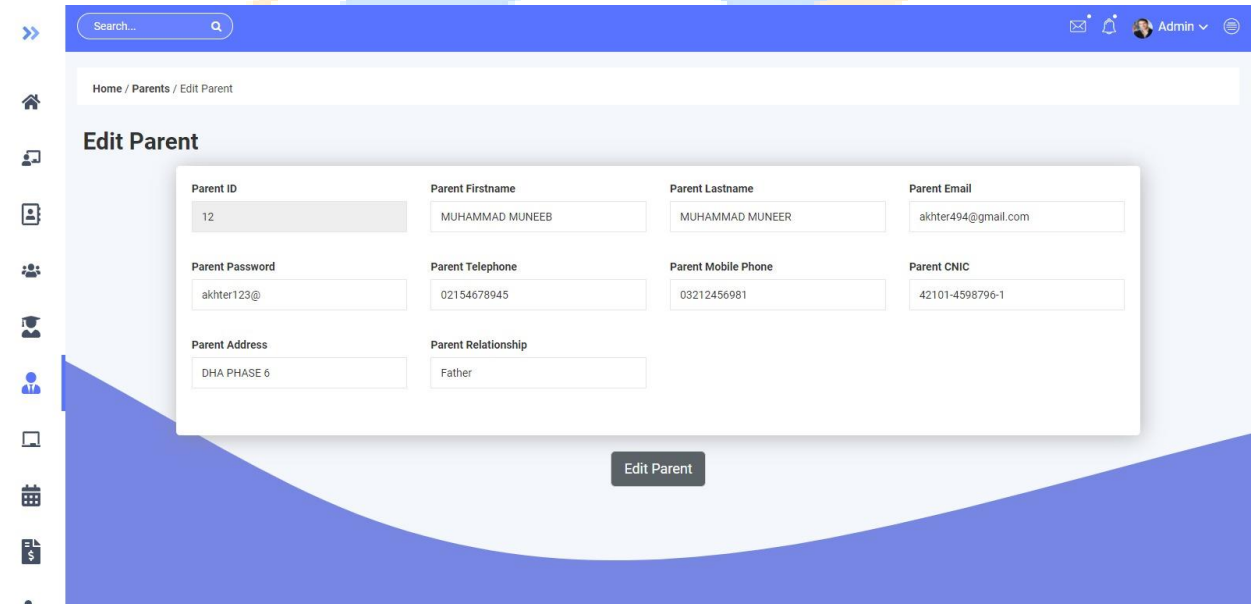



Search...
Admin
Home / Parents
Parents
Edit Parent
Add Parent
20 saleem KHAN 03312398998 saleem8104@yahoo.com Father View
21 aaaa aaaaa 03212456988 ahnkhan804@gmail.com Father View
22 saleem ALI 03312398978 MEEM804@GMAIL.COM Father View
23 YAMEEN RASHEED 03212456987 YAN12@YAHOO.COM Father View
24 YASMEEN ALI 03212456777 NAEEMALI88@GMAIL.COM Mother View
25 AHMED KASIM 03122456987 AHMEN@YAHOO.COM Father View
26 RAMEEZ AKBAR 03212456999 RAMEEZ804@GMAIL.COM Father View

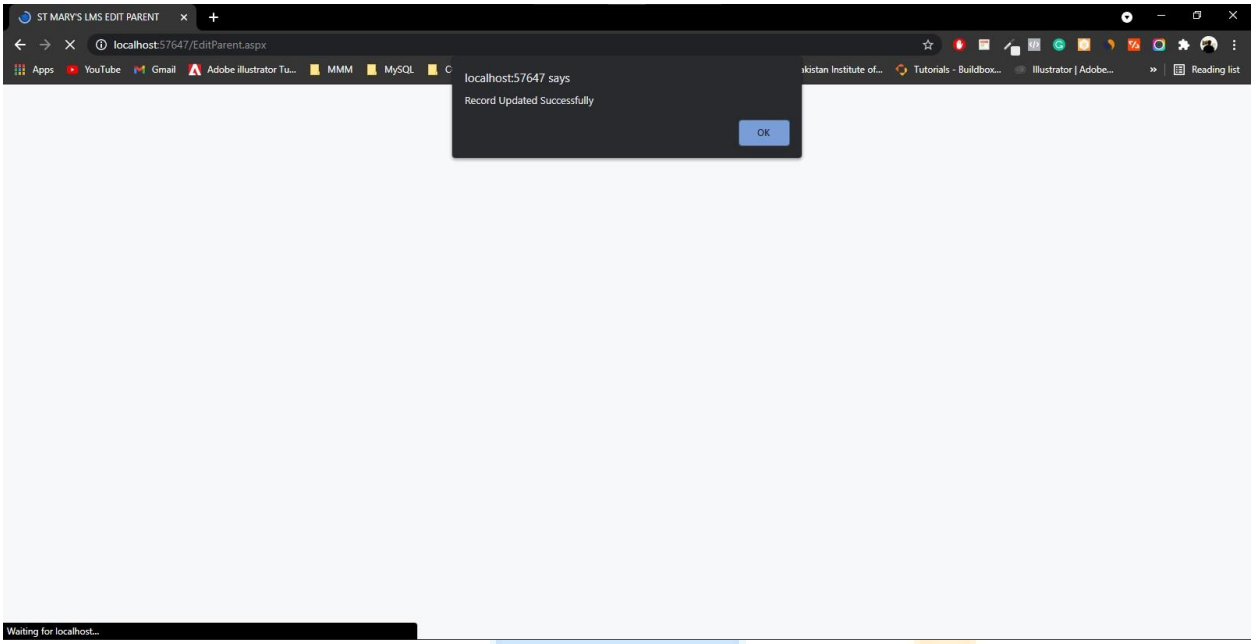

Search...
Admin
Home / Parents / Add Parent
Add Parent
Parent ID
28
Parent Firstname
OSAMA
Parent Lastname
UMER
Parent Email
OSAMEUMER84@GMAIL.COM
Parent Password
OsaME12@.
Parent Telephone
Enter Telephone
Parent Mobile Phone
03212456102
Parent CNIC
42000-31423713-0
Parent Address
Gulberg 02, Karachi
Parent Relationship
Father
Add Parent

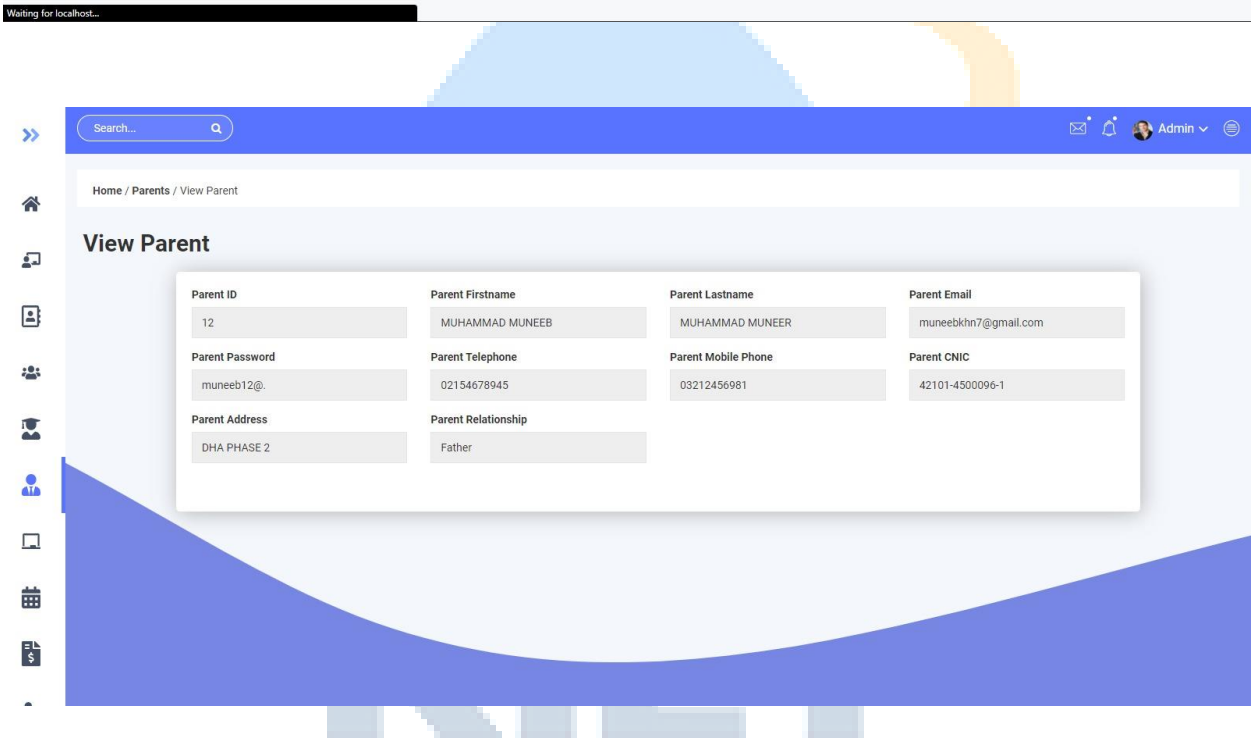


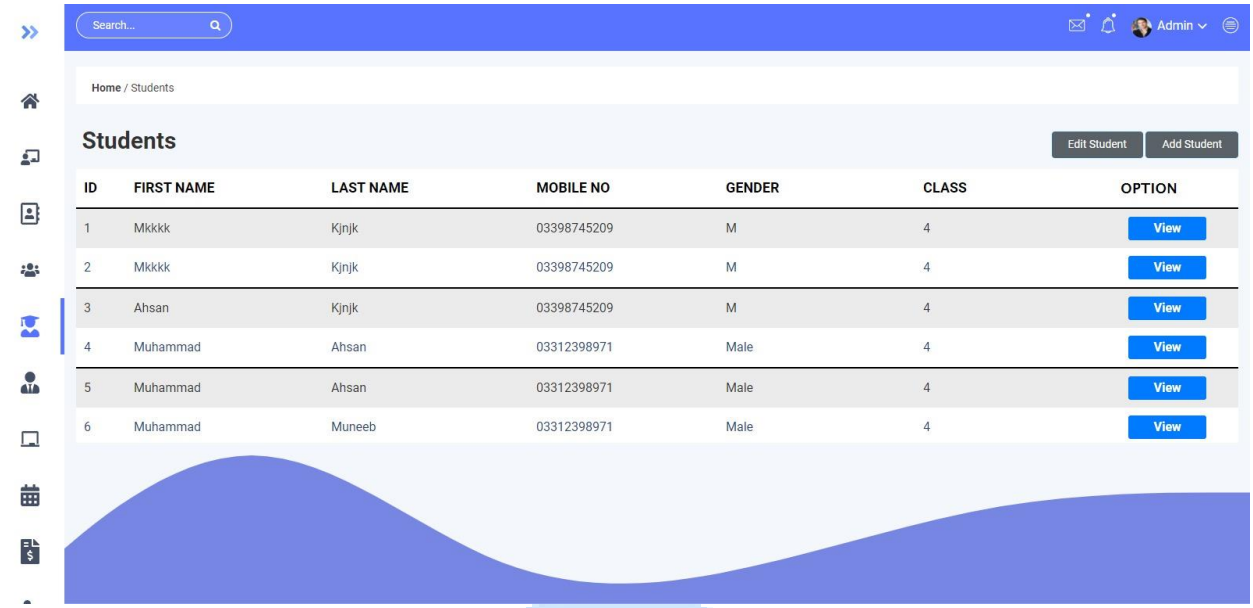
Before Update



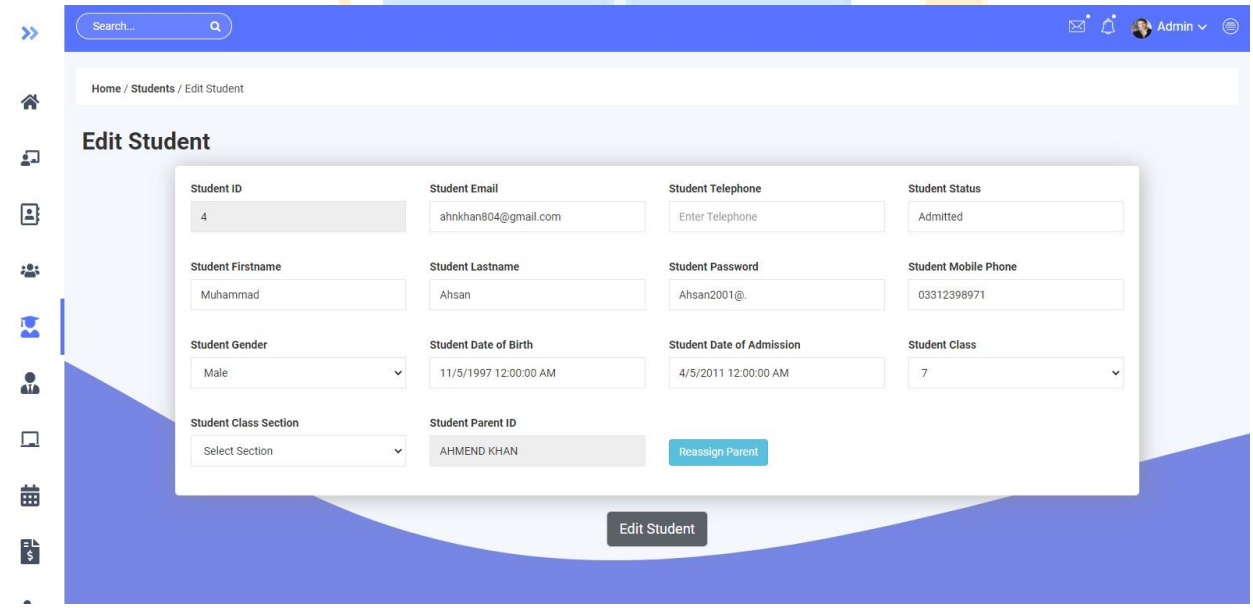
After Update


ST MARY'S LMS EDIT PARENT
localhost:57647/EditParent.aspx
Apps
YouTube
Gmail
Adobe illustrator Tu...
MMM
MySQL
Tutorials - Buildbox...
Illustrator | Adobe...
Reading list
localhost:57647 says
Record Updated Successfully
OK
Waiting for localhost...

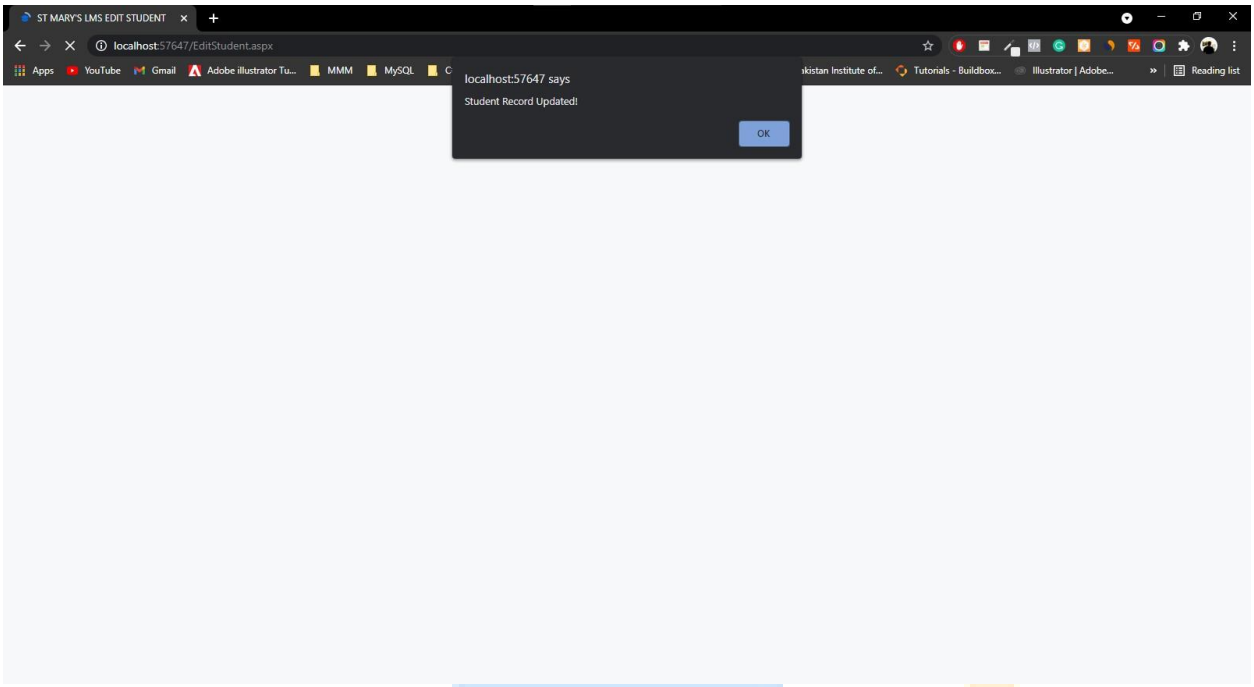

Search...
Admin
Home / Parents / View Parent
View Parent
Parent ID
12
Parent Firstname
MUHAMMAD MUNEEB
Parent Lastname
MUHAMMAD MUNEER
Parent Email
muneebkhn7@gmail.com
Parent Password
muneeb12@.
Parent Telephone
02154678945
Parent Mobile Phone
03212456981
Parent CNIC
42101-4500096-1
Parent Address
DHA PHASE 2
Parent Relationship
Father

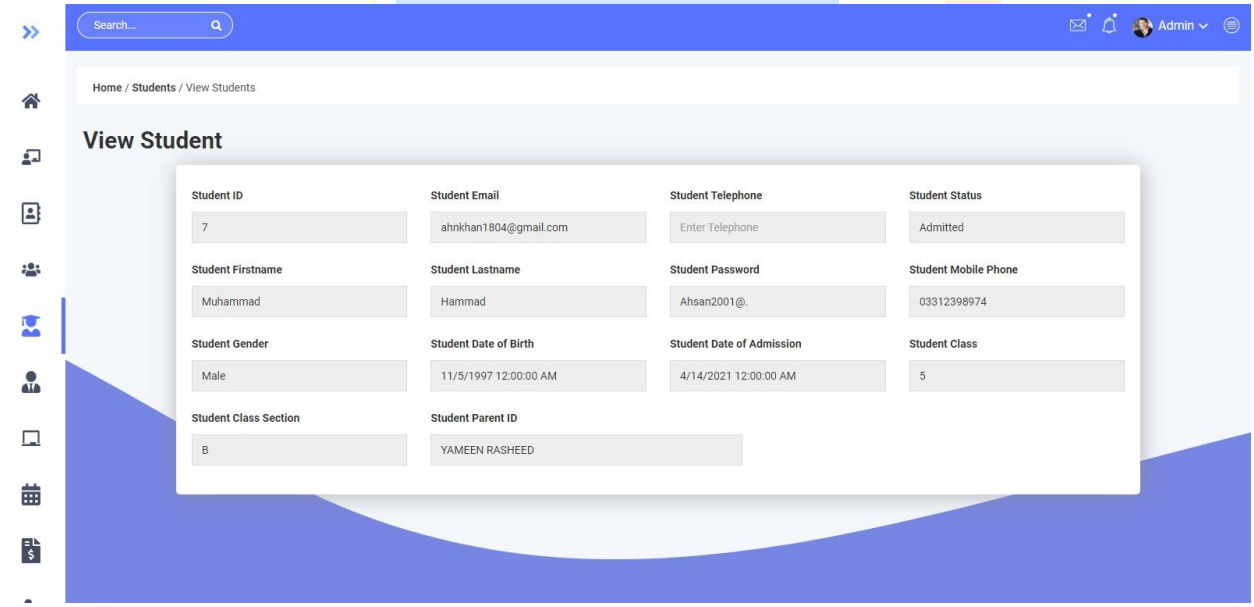


Before Update



After Update





Conclusion

In all the school management system is bringing a great difference in the lives of students, teachers, parents, and the admin. Good management offers better productivity and hence more progress towards development. Seeing its demands and benefits, we have come forward with best-featured school

management software. It helps the school to achieve the target, reduce work, increase efficiency, eliminating error, and monitoring progress.

References

- Software Engineering' by K.K. Aggarwal & Yogesh Singh, New Age Publishing House, 2nd Ed.
- IEEE Recommended Practice for Software Requirements Specifications – IEEE Std 830-1998.
- IEEE Standard for Software Test Documentation – IEEE Std. 829-1998.