A

Project - I Report

on

EDUCATION PORTAL

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Information Technology

(Session 2017-2018)

Guided By -

Mr. Shirish Nagar

(Assistant Professor ,IT Dept.)

Submitted by-

Ritu Jain(PCE15IT045)

Himani Gupta(PCE15IT016)

Subham Sharma(PCE15IT066)

DEPARTMENT OF INFORMATION TECHNOLOGY

POORNIMA COLLEGE OF ENGINEERING, JAIPUR

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Session 2018-2019

Candidate's Declaration

We hereby declare that the work, which is being presented in the **Project - I Report**, titled Education Portal in partial fulfillment for the award of degree of **Bachelor of Technology** in **Information Technology**, and submitted to the Department of **Information Technology, Poornima College of Engineering**, **Jaipur** is a record of my own work/investigations carried under the guidance of Mr. Shirish Nagar, Department of Information technology, **Poornima College of Engineering.**

We have not submitted the matter presented in this Project-I Report any where for the award of any other Degree.

Ritu Jain(PCE/IT/15/045)
Himani Gupta(PCE/IT/15/016)
Subham Sharma(PCE/IT/15/066)

Mr. Shirish Nagar

(Assistant Professor, IT Dept.)

Date: 25 October 2018
Place: Jaipur

POORNIMA
COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that **Project - I** report titled **Education Portal** has been submitted by Ritu Jain (PCE/IT/15/05), Himani Gupta (PCE/IT/15/016), Subham Sharma (PCE/IT/15/066) in partial fulfillment for the award of the Degree of **Bachelor of Technology** in **Information Technology** during the session 2018-19, Odd Semester.

The project work is found satisfactory and approved for submission.

(Amol Saxena)	**(Shirish Nagar)**
HoD - IT	Project Coordinator - IT

ACKNOWLEDGEMENT

The completion of this undertaking could not have been possible without the participation and assistance of so many people whose names may not all be enumerated. Their contribution is sincerely appreciated and gratefully acknowledged.

However, We would like to express our deep appreciation add indebtedness particularly to the following:

Mr. Amol Saxena, Head of IT Department for the encouragement as it was a great boost and motivation for us.

Mr. Shirish Nagar Associate Professor, IT Dept. for helping and guiding through the project.

We thank you all.

Ritu Jain(PCE/IT/15/045)

Himani Gupta(PCE/IT/15/016)

Subham Sharma (PCE/IT/15/066)

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Abstract

The Education Portal system is a web Application system, which is basically used to provide help to manage student and staff Data. This website provides many facilities like Online exam, online Attendance record, Assignments etc. for students. This system is help in finding information of any student in accurate and fast way. This system reduces the paper work in school. The objective of this "Education portal" is to launch an interactive and informative portal focused on education by developing, implementing and enriching a learning platform and content. This System considered as a good first step in implementing performing online based information management system.

Chapter 1

Introduction to the Project

1.1 Introduction

The Education portal has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate an in some cases reduce the hardship faced by this existing system. More over this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

1.2 Purpose

The information described in this document is the features and requirements of the project to be developed for client. The objective of this "Education portal" is to launch an interactive and informative portal focused on education by developing, implementing and enriching a learning platform and content. The document is developed after consultation with the client and considering the complete requirement specifications of the given project.

1.3 Feasibility

In order to make sure that the project is feasible, following feasibility studies have been conducted**: -**

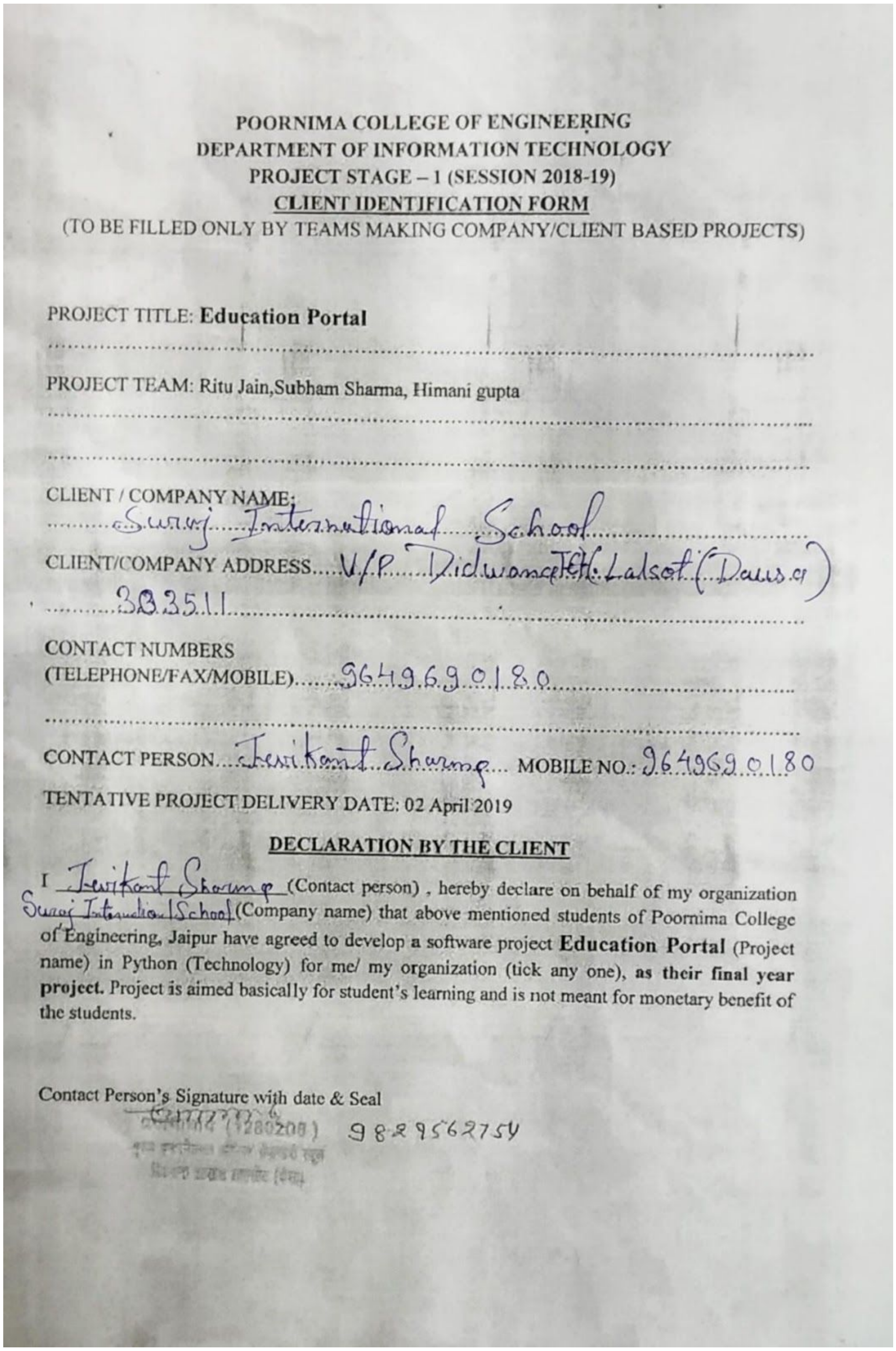
- **Technical feasibility :-**
  It is possible to develop the system using Python (Django Framework) for designing front-end and My SQL for handling back-end of our application.

- **Economic feasibility :-**

This project will be developed at sufficient cost so that it will be acceptable by all clients that need such type of application.

Chapter 2

Client Survey/ results(Photographs)



POORNIMA COLLEGE OF ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
PROJECT STAGE – 1 (SESSION 2018-19)
CLIENT IDENTIFICATION FORM
(TO BE FILLED ONLY BY TEAMS MAKING COMPANY/CLIENT BASED PROJECTS)

PROJECT TITLE: **Education Portal**

PROJECT TEAM: Ritu Jain,Subham Sharma, Himani gupta

CLIENT / COMPANY NAME: Suraj International School

CLIENT/COMPANY ADDRESS V/P Didwana Teh. Lalsot (Dausa) 303511

CONTACT NUMBERS
(TELEPHONE/FAX/MOBILE) 9649690180

CONTACT PERSON Ravikant Sharma MOBILE NO.: 9649690180

TENTATIVE PROJECT DELIVERY DATE: 02 April 2019

DECLARATION BY THE CLIENT

I Ravikant Sharma (Contact person) , hereby declare on behalf of my organization Suraj International School (Company name) that above mentioned students of Poornima College of Engineering, Jaipur have agreed to develop a software project **Education Portal** (Project name) in Python (Technology) for me/ my organization (tick any one), **as their final year project.** Project is aimed basically for student's learning and is not meant for monetary benefit of the students.

Contact Person's Signature with date & Seal
[illegible] (1280208) 9829562754
[illegible]

Chapter 3

Software Requirement Specification

3.1. Introduction

3.1.1 Purpose

The information described in this document is the features and requirements of the project to be developed for client. The objective of this “Education portal” is to launch an interactive and informative portal focused on education by developing, implementing and enriching a learning platform and content. The document is developed after consultation with the client and considering the complete requirement specifications of the given project.

3.1.2 Feasibility

In order to make sure that the project is feasible, following feasibility studies have been conducted**: -**

- **Technical feasibility:-**
  It is possible to develop the system using IDE Eclipse Juno for designing front-end and My SQL for handling back-end of our application.

- **Economic feasibility:-**
  This project will be developed at sufficient cost so that it will be acceptable by all clients that need such type of application.

3.2. Functional /Nonfunctional Requirements

3.2.1Functional Requirements

1. A system for the Employees to determine the analysis of the students.
2. User can get their academic details on the website.
3. The facility to update the personal information through the Google forms request.
4. Students as well as parents will receive SMS alerts regarding their examination performance.
5. The facility to check the student’s records and to analyze the weaker section of the student by their parents.
6. Employee can view the report of the students to improve the system

7. Admin can monitor every activity which is performed by system and improve the system by receiving feedback from the users.
8. Provides a mechanism to post queries regarding organization.

3.2.2 Nonfunctional Requirements

Performance Requirements

This software should be able to handle the following task:

- At least 20 instructors/staff can login in on average of 4hours a day for five days a week.
- At least 100 students can log into their accounts for 3 hours for 5 days of the week.
- It should be able to handle the SQLite xerial database of 100 instructors and 4000 students.

3.2.3 Safety Requirements

This software will ease the process of student grading. At the end of every semester each student will receive a grade sheet generated by the administration using the data uploaded by the course instructor on this software. All important details should be maintained in hard copy as well.

Security Requirements

This software will,

- Authenticate each user, who logs in;
- When the user performs any action, authorize him/her to perform the actions allowed for the user and displays an error message if found to be unauthorized.

3.2.4 Software Quality Attributes

- **Scalability:** We can increase the number of organization as required.
- **Manageability:** Activity status will show to the user whenever it will be updated.
- **Reliability:** The system must be backed up on a configurable schedule. Back up requirement will need to be determined. The system should be backed up with a frequency that ensures system and all data ease protected.
- **Extensibility:** System can be further extended to any level like it can include live tracking and it can be large scale project.
- **Usability:** Student can check any activity schedule through this portal instead of wasting time to go to the admin person.

- **Availability:** The system should be available 24 hours a day, 7 days a week. Our website should be efficient enough to manage the load all the time.
- **Portability:** Our website should be portable and responsive enough so that every organization can easily manage the admin panel.

3.2.5 Technical Requirments (Hardware /Software)

Operating Environment

Our Website could be operated on various browsers which include- Google Chrome, Mozilla Firebox, Internet Explorer, Mobile interface, and it also includes database servers which are deployed on BI interface. All this runs effectively with internet connectivity to update and manage activities.

3.2.6 Hardware Interfaces

Server side

The web application will be hosted on one of the department's window servers and connecting to one of the school/college database server. The web server is listening on the web standard port, port80.

Client side

The system is web based application, client are requiring using a modern web browser such as Mozilla Firebox, Internet Explorer and enable cookies. The computer must have an internet connection in order to be able to access the system.

3.2.7 Software Interfaces

Server side

The Django, a required framework to host a Django web application. An Apache server will accept all requests from the client. A development database will be hosted locally (using MySQL); the production database is hosted centrally using (using sqlite xerial).

Client side

An Os is capable of running a modern web browser which supports HTML version 5, CSS3, Bootstrap.

Communications Interfaces

This will be a web application and as it will be communicating with a database server and so will be making use of network and HTTP to communicate.

3.3 System Features

Module 1: User Login and Registration System

Description and Priority

The main purpose of this module is to provide Security by allowing access to only valid users. In order to access information about all the members of management (including students, teachers and all the staff) a registration page is maintained which feeds user information required to access education portal into the registration table in database so as to store all user personal information at one place. This way a valid user can easily get the details.

Stimulus/Response Sequences

1. The data is stored and manage by our database management system.
2. User can interact with other modules only after this module is verified.
3. As user gives input according to required fields, the data is send to secure records through database and during login the stored data is retrieved to verify the username and password.

Module 2: Teachers

Description and Priority

The teacher will manage the time table of the different classes so that the student can view that timetable according to their respective classes. They will also import Exam result and details of different events which will be organized in respective year. They will update the details of attendance and import the assignment for the student practice.

Stimulus/Response Sequences

1. The details will be seen on the student login portal.
2. These details will be stored in the database.

Module 3: Students

Description and Priority

The student has the alerts of the assignments so that they can do practice to enhance their knowledge. They will get the update of an activity which will be organized in the organization and can view their exam and attendance result. They will manage to see the details regarding their fees. They will update their assignment on this portal.

Stimulus/ Response Sequences
Student will be able to see the details of various activities.

Module 4: Admin

Description and Priority
In main admin it can check the live report of the management of the organization and can also deactivate or block the misleading user. Admin can manage and get the details of the user who are using the website at the current time.

Stimulus/ Response Sequences
1. Admin can edit the details of the students and the staff.
2. Admin gets updated about the connected users.

3.4 Analysis Diagrams

3.4.1 Use Case Diagram

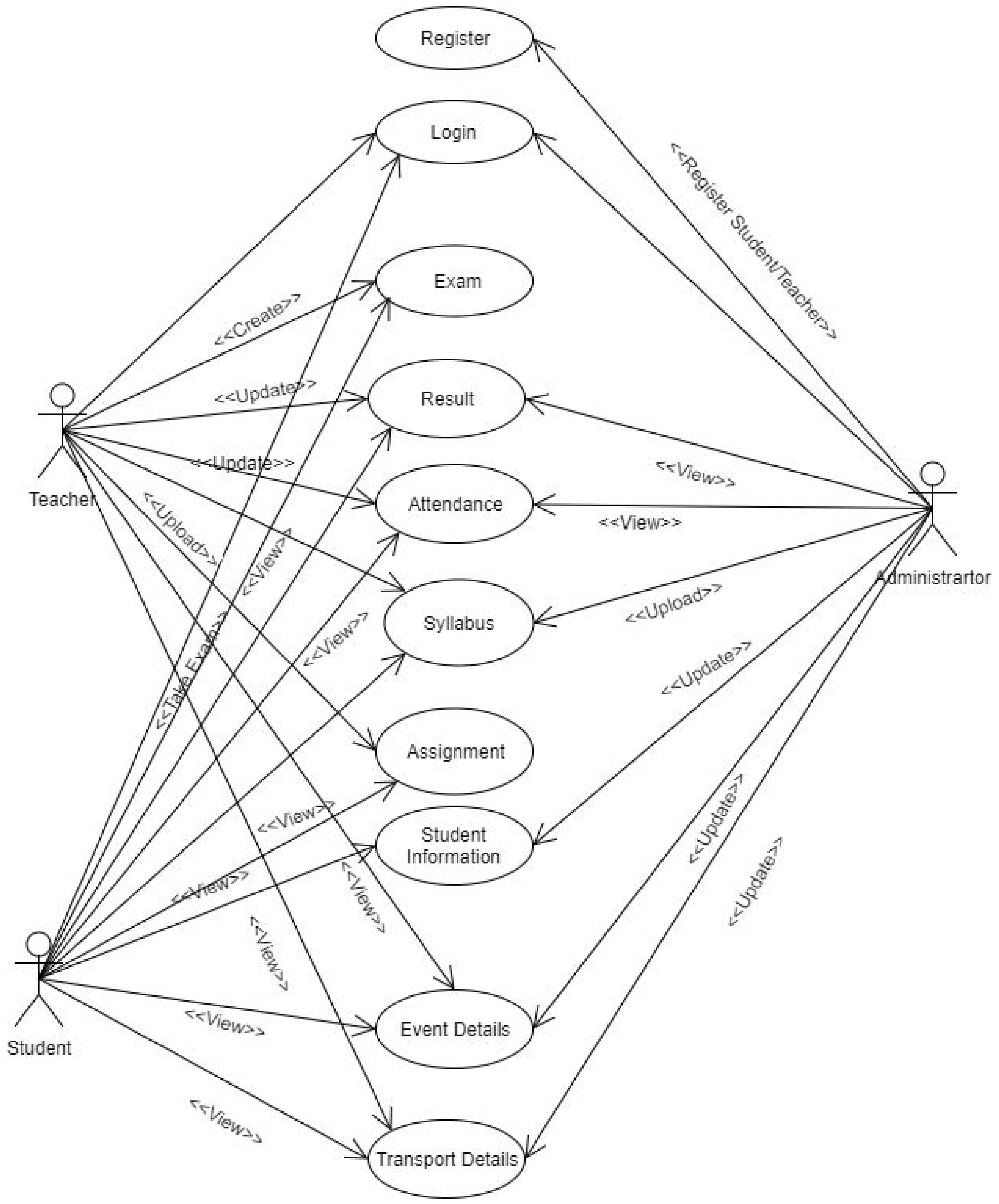


Fig 3.4.2 Use Case Diagram

3.4.2 Sequence Diagram

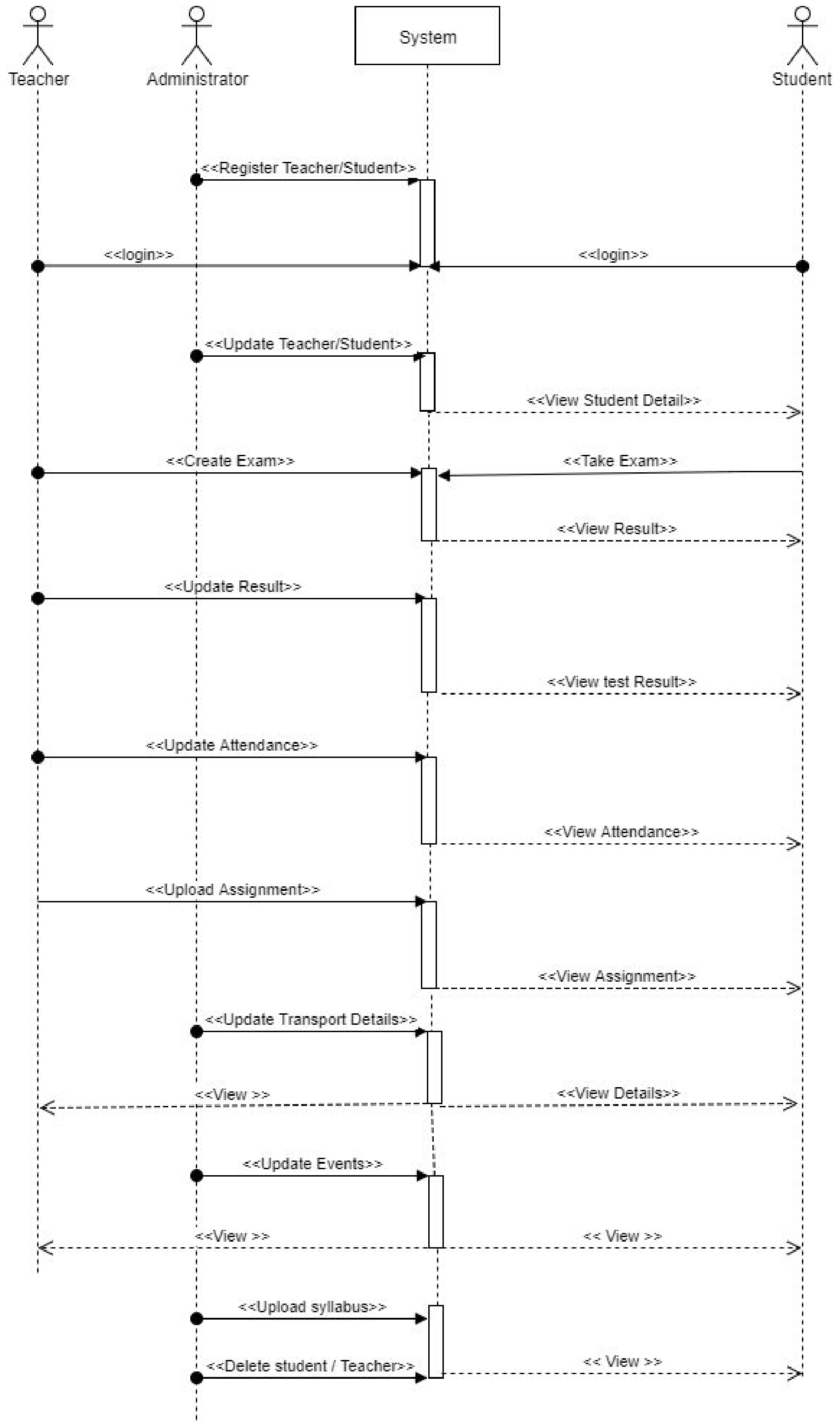


Fig 3.4.2 Sequence Diagram

3.4.3 Component Diagram

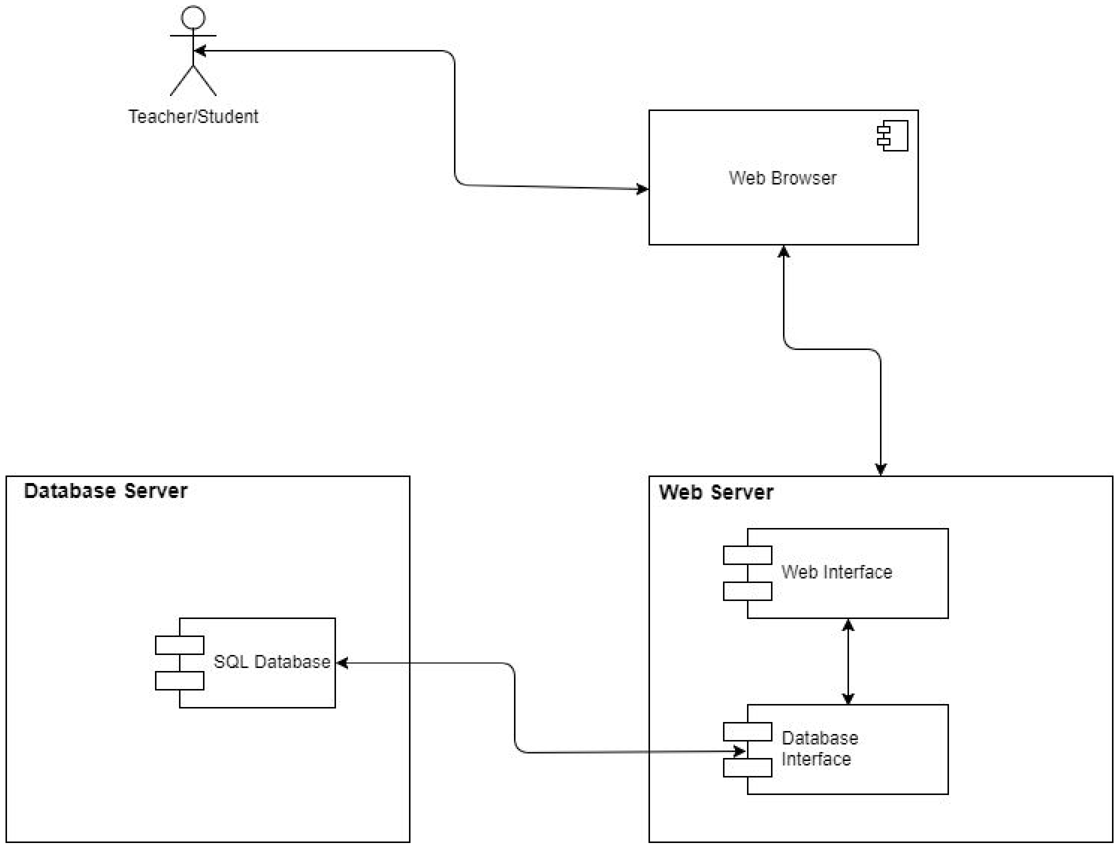


Fig 3.4.3 Component Diagram

3.4.4 ER Diagram

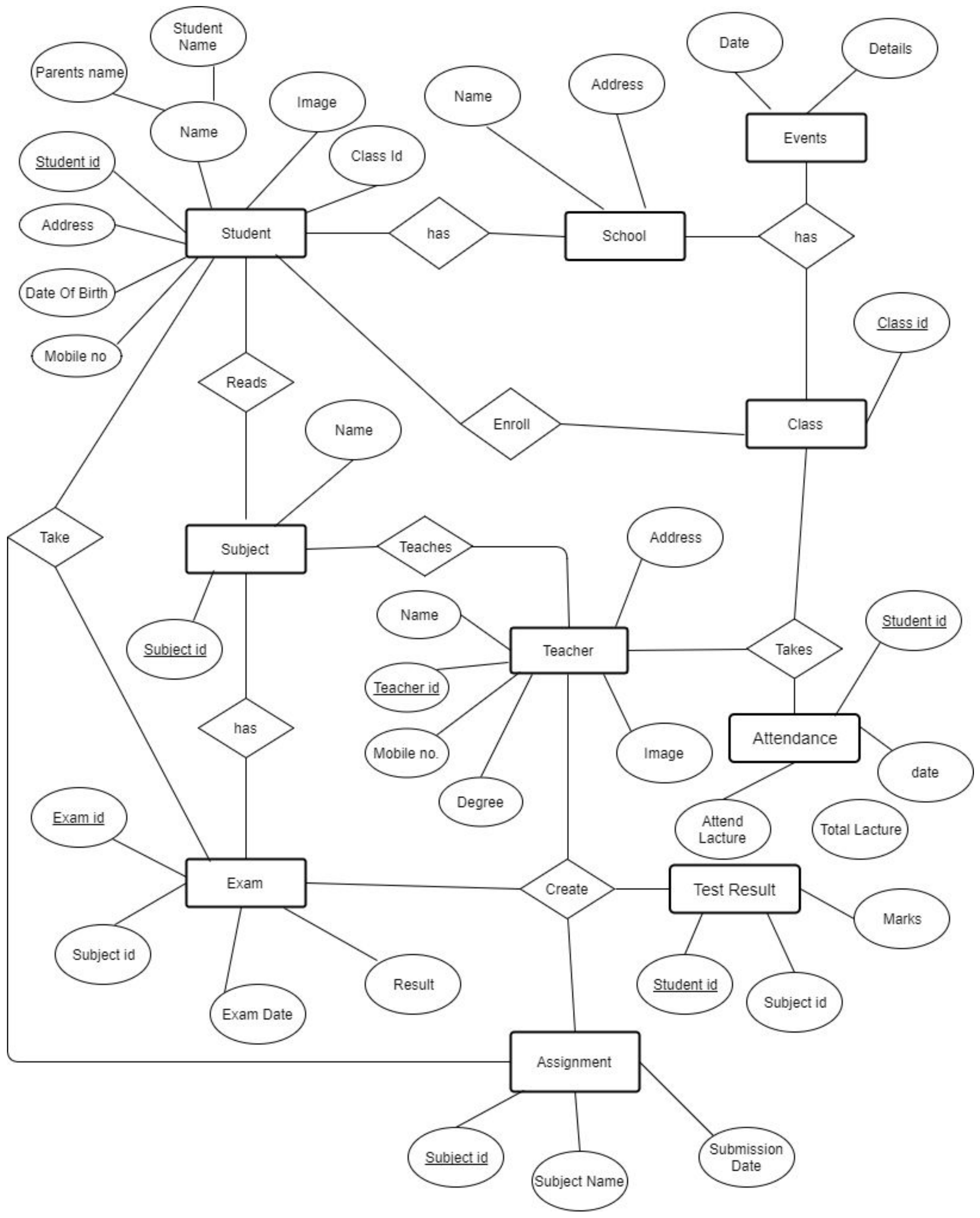


Fig 3.4.4 ER Diagram

3.4.5 Activity Diagram

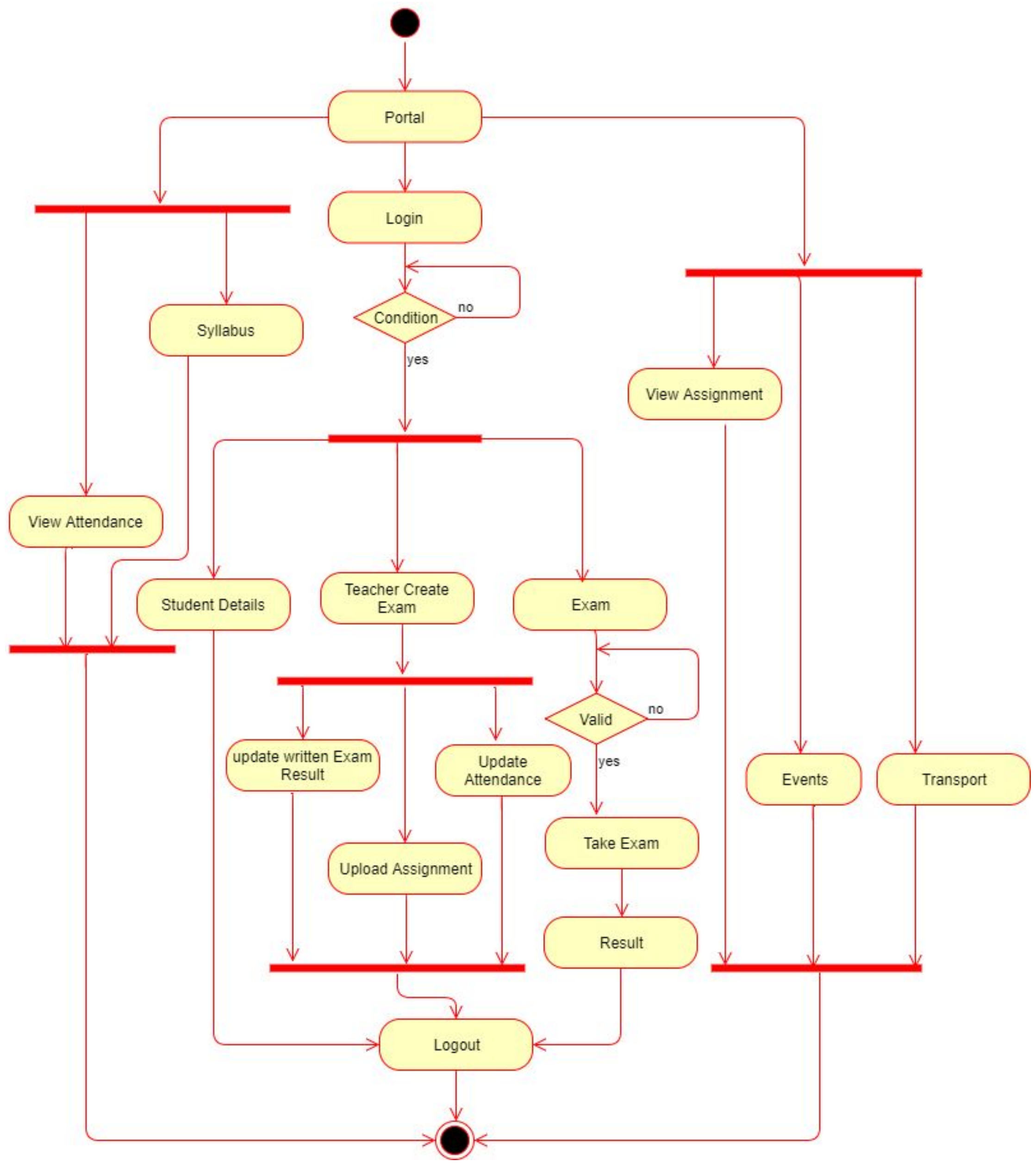


Fig 3.4.5 Activity Diagram

3.4.6 Architecture Diagram

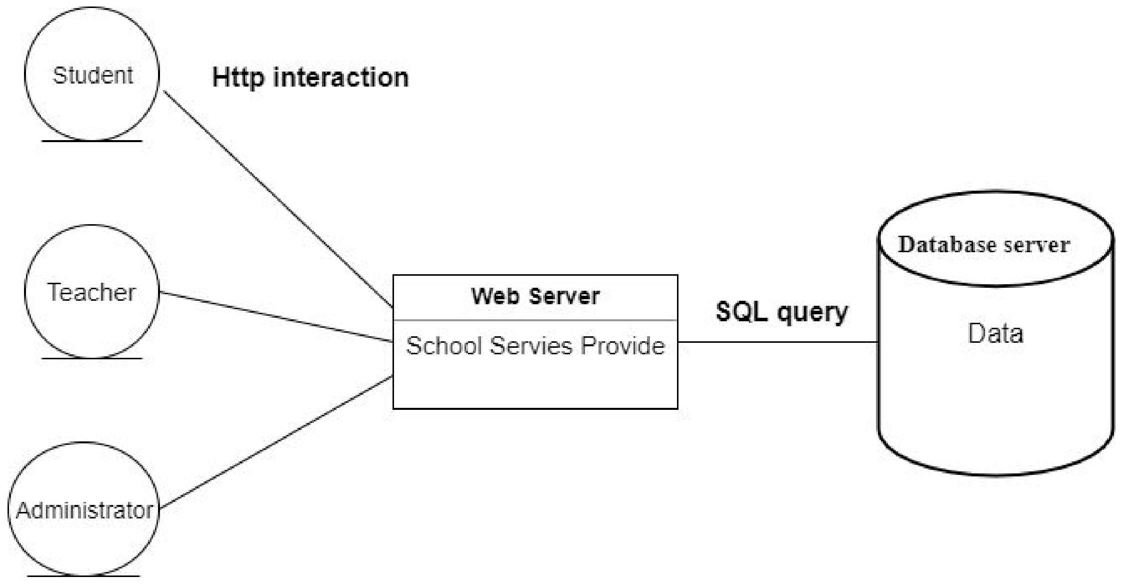


Fig 3.4.6 Architecture Diagram

3.5 Glossary

S. No	Abbreviations	Full Forms
1.	HTML	Hyper Text Markup Language
2.	CSS	Cascading Style Sheet
3.	JS	Java Script
4.	SQL	Structured Query Language

5.	ER Diagram	Entity Relationship Diagram
6.	SRS	Software Requirement Specification
7.	HTTP	Hyper Text Transfer Protocol

3.6 Reference:

Books:

1. Python programming
2. software engineering - McGraw-Hill Education

Website:

1. www.w3schools.com
2. www.stackoverflow.com
3. www.idsPrime.com

Chapter 4

Software Design Documents

4.1 Introduction

The information described in this document is the features and requirements of the project to be developed for client. The objective of this “Education portal” is to launch an interactive and informative portal focused on education by developing, implementing and enriching a learning platform and content. The document is developed after consultation with the client and considering the complete requirement specifications of the given project.

4.2 System Flow Diagram

Reference : SRS (Activity Diagram)

4.3 UML Diagram

4.3.1 Class Diagram

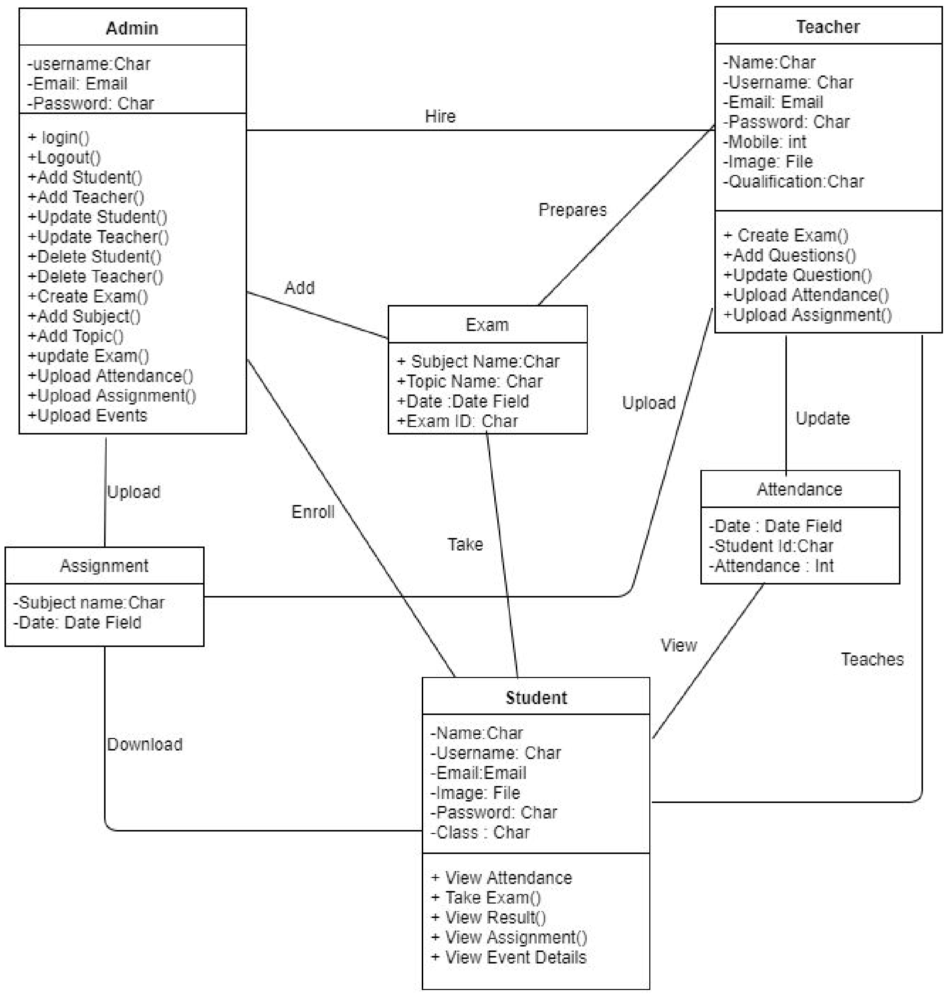


Fig 4.3.1 Class Diagram

4.3.2 Deployment Diagram

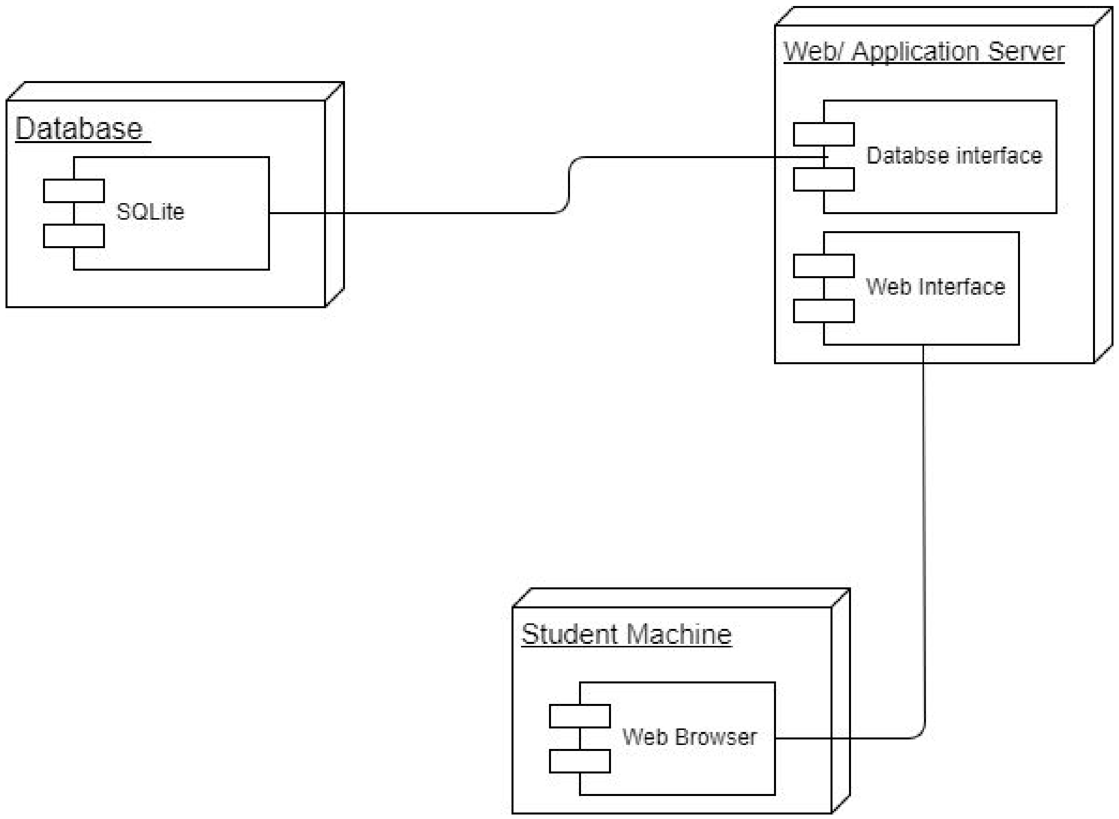


Fig 4.3.2 Deployment Diagram

4.4 Database Table

4.4.1 Student Information

Attribute	**Data Type**	**Constraint**
Student Name	Varchar(50)	-
Username	Varchar(20)	Primary key
Password	Varchar	-
Image	Image Field	-
Email_Id	Varchar(50)	-

Class	Varchar(10)	-
Address	Varchar	-
DOB	Date Field	-

4.4.2 Question Model

Attribute	Data Type	Constraint
Topic	Number	Primary key
Question	Varchar	-
Option A	Varchar	-
Option B	Varchar	-
Option C	Date	-
Option D	Varchar	-
Answer	Varchar	-
Marks	Integer	-

4.4.3 Subject Model

Attribute	Data Type	Constraint
Class	Varchar(10)	Primary key
Subject	Varchar(50)	-
Subject Id	Varchar(10)	-

4.4.4 Test Given

Attribute	Data Type	Constraint
Exam Date	Date Field	Primary Key
Exam Time	Time Field	-
Subject Name	Varchar(50)	-
username	Varchar(20)	-
Topic Name	Varchar(50)	-
Total Question	Number	-
Total Attempted	Number	-

Total Correct	Number	-
Total Incorrect	Number	-
Marks Obtained	Number	-
Percent	Number	-

4.4.5 Topics

Attribute	Data Type	Constraint
Subject	Varchar(10)	Primary Key
Topic	Varchar(50)	-

4.4.6 Attendance

Attribute	Data Type	Constraint
Username	Varchar(20)	Primary key
Date	Date Field	-
Attend Lecture	Number	-
Total Lecture	Number	-
Percent	Number	-

4.4.7 Assignment

Attribute	Data Type	Constraint
Subject name	Varchar(50)	Primary Key
Date	Date Field	-
Topic	Varchar(50)	-
Assignment	File Field	-

4.4.8 Events

Attribute	Data Type	Constraint
Event File	Image Field	Primary Key
Date	Date Field	-

4.4.9 Teacher Information

Attribute	Data Type	Constraint
Name	Varchar(50)	-
User ID	Varchar(20)	Primary Key
Email_Id	Varchar(50)	-
Mobile no	Integer(10)	-
Image	Image Field	-
Qualification	Varchar	-
Address	Varchar	-
DOB	Date Field	-

4.5 Database Diagram

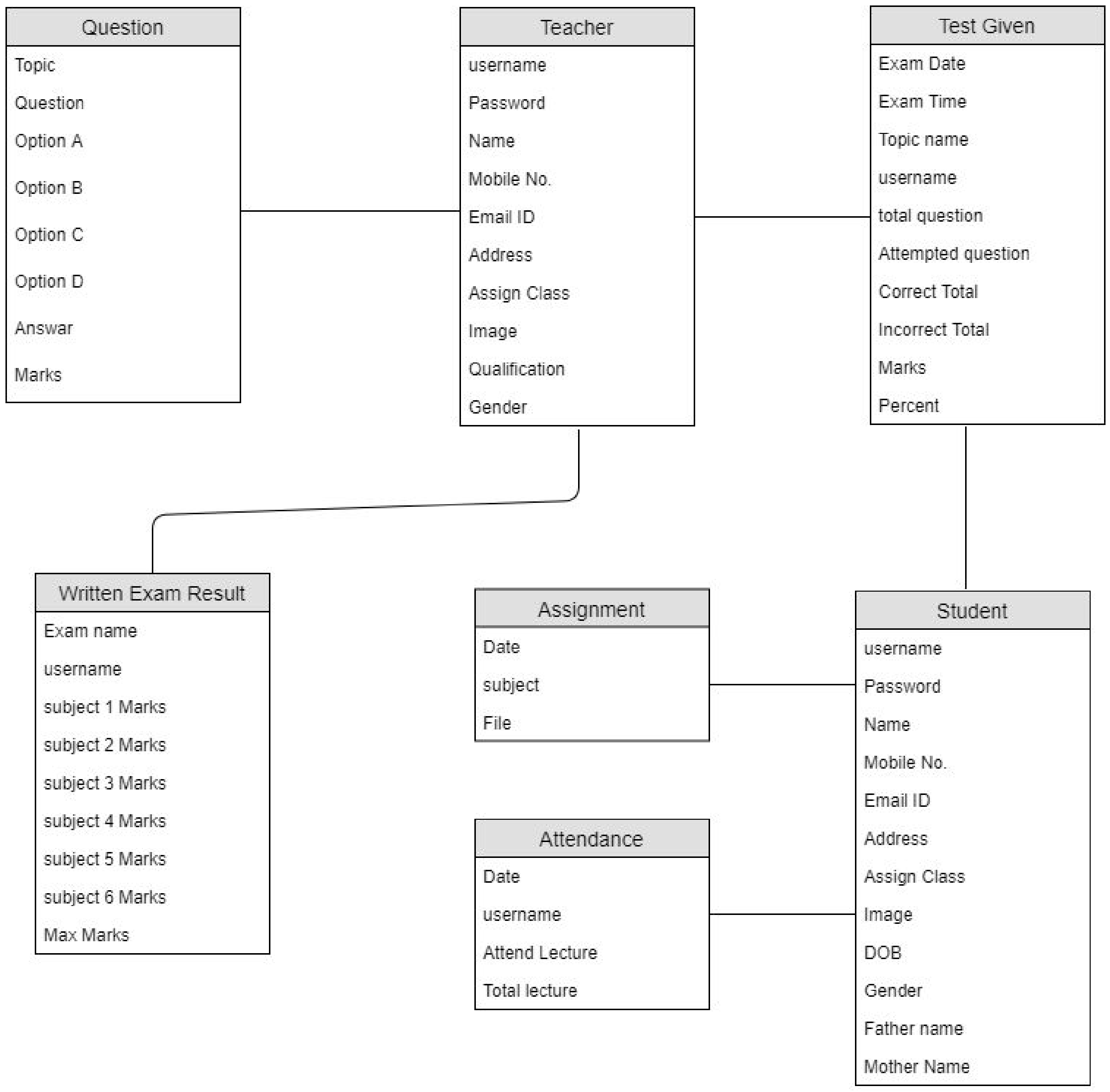
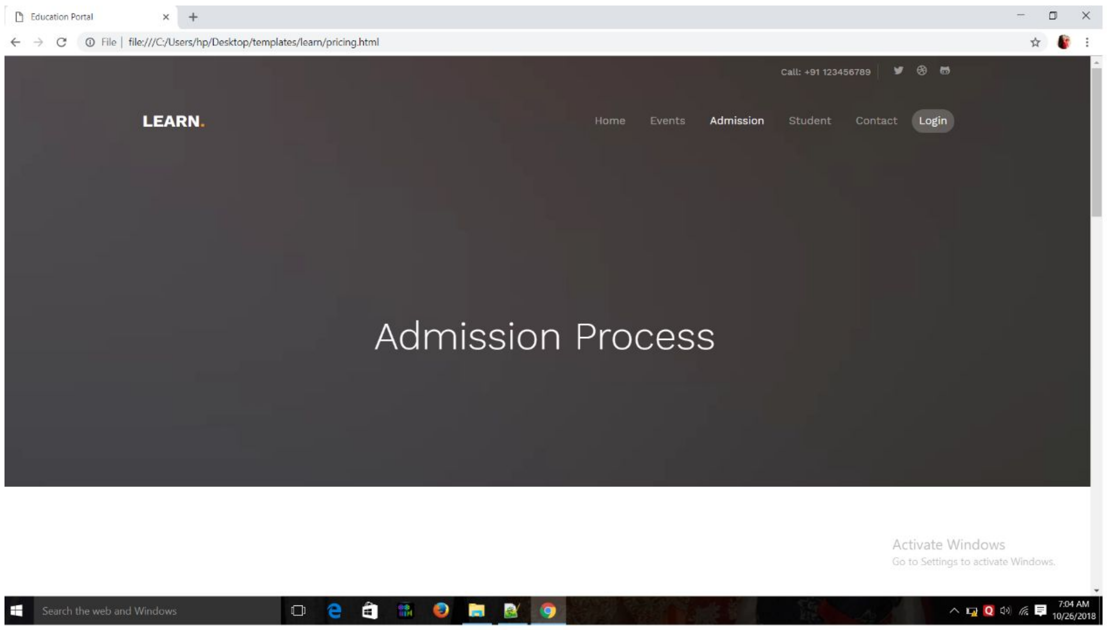
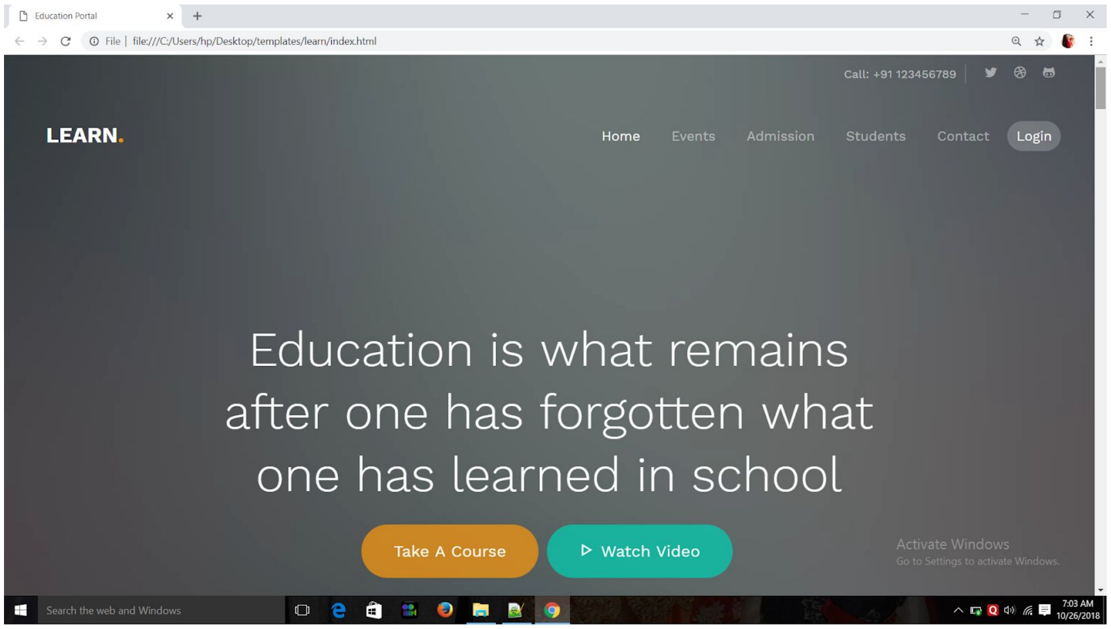


Fig 4.5 Database Diagram

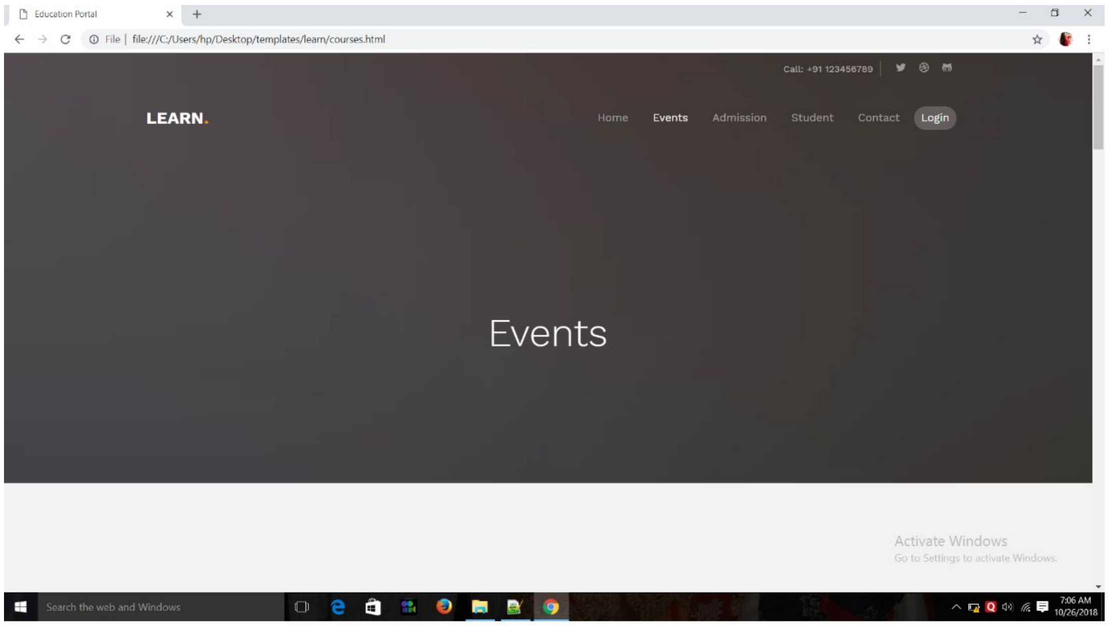
4.6 GUI Design

Screenshots

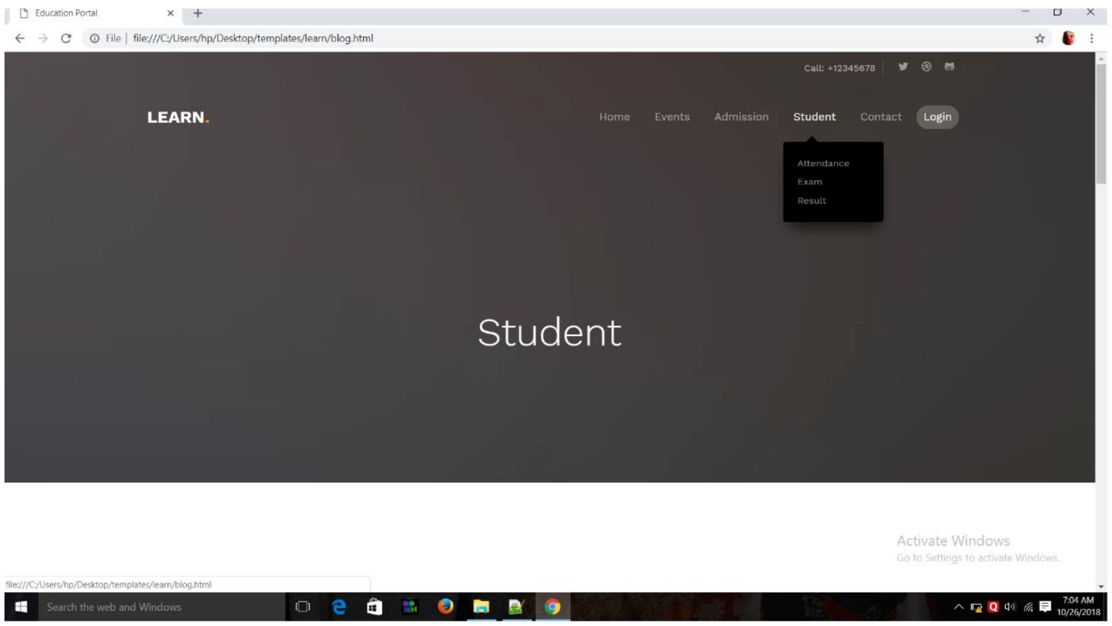




Education Portal
file:///C:/Users/hp/Desktop/templates/learn/index.html
Call: +91 123456789
LEARN.
Home
Events
Admission
Students
Contact
Login
Education is what remains after one has forgotten what one has learned in school
Take A Course
Watch Video



Education Portal
file:///C:/Users/hp/Desktop/templates/learn/courses.html
Call: +91 123456789
LEARN.
Home
Events
Admission
Student
Contact
Login
Events



4.7 API Specification

API	Required Package	Version
1. Rest API	Python	2.7, 3.4, 3.5, 3.6, 3.7
	Django	1.11, 2.0, 2.1
	Markdown	2.1.0
	Django-filter	1.0.1

API Reference

- Built-in class-based views API

Base class-based views can be thought of as *parent* views, which can be used by themselves or inherited from. They may not provide all the capabilities required for projects, in which case there are Mixings which extend what base views can do. Django's generic views are built off of those base views, and were developed as a shortcut for common usage patterns such as displaying the details of an object. They take certain common idioms and patterns found in view development and abstract them so that you can quickly write common views of data without having to repeat yourself.

- contrib packages

Django aims to follow Python's "batteries included" philosophy. It ships with a variety of extra, optional tools that solve common Web-development problems.

This code lives in **Django/contrib** in the Django distribution. This document gives a rundown of the packages in **contrib**, along with any dependencies those packages have.

- Databases

  Django attempts to support as many features as possible on all database backend. However, not all database backend is a like, and we've had to make design decisions on which features to support and which assumptions we can make safely.

  This file describes some of the features that might be relevant to Django usage. Of course, it is not intended as a replacement for server-specific documentation or reference manuals

- Django-admin and manage.py

  Django-admin is Django's command-line utility for administrative tasks. This document outlines all it can do.

  In addition, **manage.py** is automatically created in each Django project. **manage.py** does the same thing as **Django-admin** but takes care of a few things for you:

- It puts your project's package on **sys.path**.

- It sets the **DJANGO_SETTINGS_MODULE** environment variable so that it points to your project's **settings.py** file.

- Django Exceptions

  Django core exception classes are defined in **django.core.exceptions**

Some of them are as follows:

1. **AppRegistryNotReady**
2. **ObjectDoesNotExist**
3. **EmptyResultSet**
4. **FieldDoesNotExist**
5. **MultipleObjectsReturned**

4.8 References

Book:

1. python programming
2. software engineering - McGraw-Hill Education
3. Advance Django models
4. Django Book

Website:

1. www.stackoverflow.com
2. www.djangotutorial.com

Chapter 5

Sample Coding

5.1 Model.py

```
from django.db import models
from django.contrib.auth.models import User
from django.db import models
from django.db.models.signals import post_save
from django.dispatch import receiver

class Profile(models.Model):
   user = models.OneToOneField(User, on_delete=models.CASCADE)
   location = models.CharField(max_length=30, blank=True)
   birthdate = models.DateField(null=True, blank=True)
   mobile = models.IntegerField(max_length=10, blank=True)

   def __str__(self):  # __unicode__ for Python 2
      return self.user.username

# Create your models here.
class SignupModel(models.Model):
   name = models.CharField(max_length=250)
   Email = models.EmailField(max_length=250,unique=True)
   Password = models.CharField(max_length=50)
   username = models.CharField(max_length=50,unique=True)

   def __str__(self):
      return self.Email

class Subject(models.Model):
   subname=models.CharField(max_length=50)

         def __str__(self):
      return '%s'%(self.subname)
```

```
class Topics(models.Model):
   subject = models.ForeignKey( Subject, on_delete=models.CASCADE )
   topicname=models.CharField(max_length=80)

   def __str__(self):
      return '%s'%(self.topicname)

class Question(models.Model):
   topics=models.ForeignKey(Topics,on_delete=models.CASCADE)
   question=models.TextField()
   optionA=models.CharField(max_length=50)
   optionB=models.CharField(max_length=50)
   optionC=models.CharField(max_length=50)
   optionD=models.CharField(max_length=50)
   answer=models.CharField(max_length=50)
   marks=models.IntegerField()

   def __str__(self):
      return '%s - %s'%(self.topics.topicname,self.question)

class TestGiven(models.Model):
   exam_date = models.DateField()
   exam_time = models.TimeField()
   user_id = models.IntegerField()
   subject_name = models.CharField(max_length=50)
   topic_name = models.CharField(max_length=50)
   total_question = models.IntegerField()
   total_attempted = models.IntegerField()
   total_correct = models.IntegerField()
   total_incorrect = models.IntegerField()
   marks_obtained = models.FloatField()
   percent = models.FloatField()

   def __str__(self):

        return '%d - %s - %s'%(self.user_id,self.subject_name,self.topic_name)
```

5.2 views.py

```
from django.shortcuts import render, redirect
from apt.models import *
```

```
          user = authenticate(username=request.POST.get('username'),
password=request.POST.get('Password'))
          if user:
             request.session['username'] = model.username
             login(request, user)
             return redirect('logsuccess')
          else:
             return redirect('login')
       else:
          return redirect('login')
   return render(request, 'login.html', {'form': LogInForms(), 'Email': model.Email})

def log_out(request):
   logout(request)
   return render(request, 'login.html', {})

def signup(request):
   form = SignupForms()
   model = SignupModel()
   if request.method == "POST":
      form = SignupForms(request.POST)
      if form.is_valid():
         form.name = request.POST.get('name')   #
         form.Email = request.POST.get('Email')    #
         form.Password = request.POST.get('Password')  #
         form.username = request.POST.get('username')  #

         model.save()

User.objects.create_user(username=form.username, password=form.Password)  #
         return redirect("logsuccess")

      else:
         return HttpResponseRedirect("contact/")
    else:
      return render(request,'signup.html',{'form':form})



def exam(request):
```

```
templates='exam.html'
   context={}
if test_id:
      temp = 1;
context = {'s': s, 't': t, 'name': uname,
              'q_count': q_count, 'l_count': l_count, 'r_count': r_count, 'v_count': v_count,
         for i in range(1, c+1):
            c="q"+str(i)
            print('You did not answer this question!!!')
         elif user_answer == a[i-1]:
            correct+=1
            print(user_answer,' is correct answer')
         else:
            incorrect+=1

            print(user_answer, ' is incorrect answer')
      test = TestGiven()
      test.exam_date = datetime.datetime.now()
      test.exam_time = datetime.datetime.now().time().strftime('%H:%M:%S')
      test.subject_name = subjectname
      test.topic_name = topicname
      test.total_question = c
      test.total_attempted = c-unanswered
      test.total_correct = correct
      test.total_incorrect = incorrect
      test.marks_obtained = correct
      test.percent = round((correct/c)*100,2)
      test.user_id = uid
      test.save()
      messages.info(request,'Test successfully submitted')
      return redirect(exam)
   return redirect(logsuccess)
```

5.3 urls.py

```
"""training URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from django.conf.urls import url
from apt import views
from django.contrib.auth import views as auth_views
urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^login/$',views.Login, name="login"),
    url(r'^logout$', views.log_out, name="logout"),
    url(r'^signup/$', views.signup, name="signup"),
    url(r'^about/$',views.about,name='about'),
    url(r'^contact/$',views.contact,name='contact'),
    url(r'^$',views.home,name='home'),
    url(r'^logsuccess/$',views.logsuccess,name='logsuccess'),
    url(r'^exam/$',views.exam,name='exam'),
    url(r'^startexam/$',views.startexam,name='startexam'),



]
```

Chapter 6

Report from Guide and Recommendation

Chapter 7

Conclusion And Future Scope of work

7.1 Conclusion

The project was designed in such a way that future modification can be done easily. The following conclusions can be reduced from the development of the project.

- This System are being identified as an appropriate method for managing information in schools.
- This System considered as a good first step in implementing performing online based information management system.
- Provides easy, accurate and faster data access.

7.2 Future Scope

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when required in this web application.

- We will add fee portal in the system in future.
- We can add notes and previous year question paper to this web application for students.
- We provide a facility to student forgot password.

.

Chapter 8

FAQ questions about the project

8.1 What is the future scope of your project?

- We will add fee portal in the system in future.
- We can add notes and previous year question paper to this web application for students.
- We provide a facility to student forgot password.

8.2 What are the hardware and software requirements to run your project?

- Web Brower

8.3 What is the core technologies included in project development?

- Python Django

APPENDICES

1. Pycharm Version

We have used python IDE for professional Developer by JetBrains version 2018.2.4 released in September 19,2018.

2. Django Version

The last version to support python 2.7 is Django 1.11 LTS.

3. Database

Django attempts to support as many features as possible on all database backend. However, not all database backend is a like, and we've had to make design decisions on which features to support and which assumptions we can make safely.

This file describes some of the features that might be relevant to Django usage. Of course, it is not intended as a replacement for server-specific documentation or reference manuals.