**EDUVILLE - SCHOOL MANAGEMENT SYSTEM**

A MINI-PROJECT REPORT

Submitted by

SHREERAM.S 200701234

SURYA.K 200701521

***in partial fulfillment of the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE**

**ANNA UNIVERSITY:CHENNAI**

**MAY 2022**

**BONAFIDE CERTIFICATE**

Certified that this project **“EDUVILLE SCHOOL MANAGEMENT SYSTEM ”** is the bonafide work of **“SHREERAM.S AND SURYA.K”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Name of the HOD** | **Name of the supervisor** |
| **MRS.REVATHI** | **MRS.AKILADEVI** |
| Dept. of Computer Science and Engg, | Dept. of Computer Science and Engg, |
| Rajalakshmi Engineering College,Chennai | Rajalakshmi Engineering College,Chennai |

Submitted for the End semester practical examination Mini-Project work viva voce held on \_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ABSTRACT**

School Management System helps headmasters to get the most accurate information to make more effective decisions. Teachers and headmasters gain time saving administrative tools, parents gain immediate access to their children’s grades and students can track their own progress. School Management System equipped features makes it possible to generate schedules and reports in minutes and to retrieve attendance records, grade checks, report cards, transcripts, and form letters in just a few clicks. School Management Systems helps Teachers to complete grade book, track students attendance, input class notes, create lesson plans and detailed reports, and communicate with other staff members, students, and parents all via e-mail. It also helps Students to access assignments and tests, and view attendance records, grades, report cards, and progress reports all online. They also can communicate through mail and forums with teachers and other students online.

My School (SMS) is a web enabled application developed in Python Django-Web Framework and powerful SQLite3 database backend. To implement My School application, schools do not need expensive hardware and software, they just need an internet connection and desktops. Our system works as a centralized database and application that schools can easily access the system from anywhere based on the login credentials. My School is a platform independent system that virtually any user can access from anywhere through a standard internet accessible system. We can also customize My School for individual school needs.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chariperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance

No Words Of Gratitude Will Suffice For The Unquestioning Support Extended To Us By Our Head Of The Department **MRS.REVATHI.,**For Being Ever Supporting Force During Our Project Work

We also extend our sincere and hearty thanks to our internal guide **MRS.AKILADEVI.,** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of information technology.

SHREERAM.S

SURYA.K

**TABLE OF CONTENTS**

CHAPTER NO. TITLE PAGE

**ABSTRACT iv**

**1 INTRODUCTION 1**

1.1 introduction 1

1.2 SCOPE OF THE WORK 2

1.3 PROBLEM STATEMENT 3

1.4 AIM AND OBJECTIVES OF THE PROJECT 5

**2 SYSTEM SPECIFICATIONS 6**

2.1 HARDWARE SPECIFICATIONS 6

2.2 SOFTWARE SPECIFICATIONS 6

**3 MODULE DESCRIPTION 7**

**4 SYSTEM DESIGN 9**

4.1 ARCHITECTURE DIAGRAM 9

4.2 FLOW CHART 10

**5 CODING 12**

**6 SCREEN SHOTS 17**

**7** **CONCLUSION AND FUTURE ENHANCEMENT 24**

**REFERENCES 25**

**CHAPTER 1**

**INTRODUCTION**

# INTRODUCTION

Nowadays education plays a great role in development of any country. Many of education organizations try to increase education quality. One of the aspects of this improvement is managing of school resources. Our system is a major tool through this managing process by making a continuous communication between headmaster, teacher, parents and student. So in order to achieve that goal, we need a Website that covers the needs of all users at the same time.

For Students, they can view their subject's grades, contact with the headmaster and teachers for any complaint, recommendation or an absence permission, and they also up to date with all school's news or posts that publish by the other users. For Headmasters, they have a full control on the system, like they can add a new parents, teachers and students with their subjects.

For Teachers, they can add student’s grades or edit it for their own subjects only, and they have a direct connection with students and their parents. For Parents, they have an access for their sons/daughters grades without any possibility to edit on it, and they can directly contact with teachers and headmasters.

All the system users can publish whatever they need within the educational process on the last updates section, and these posts is visible for all the users.

* 1. **SCOPE OF THE WORK**

Scope School Management System will organize work inside school and proposed system will do the following tasks:

* Insert student’s information such as student name, student number, address etc.
* Insert employee’s information such as employee name, number, address etc.
* Insert section’s information such as section name etc.
* Insert marks for each student in each subject he/she taken.
* View data about certain student and can edit it like adding or removing a student.
* View data about certain employee and can edit it like adding or removing an employee records including blood donation, blood requests and blood stock for the administration.
  1. **PROBLEM STATEMENT**

The academic achievement for many students has decline, because of lake of care of them from their parents, and this refer to that their parents do not have a free time to come to school . Headmasters and Teachers are facing problems at the start of every new academic year, because of distribution process for courses and classes, in addition to this , through and at the end of every year another problem is facing them, which is the complexities of the grades entry process for their students . Lack of communication after a school day between Headmasters, Teachers, Parents and Students, which has a bad reflection on the educational process.

**1.4 AIM AND OBJECTIVES OF THE PROJECT**

1. To build a responsive website to manage the different school activities.
2. To track student’s grades from their parents.
3. To facilitate distribution process of courses and classes for teachers
4. To facilitate grades entry process for students by teachers.
5. To make a virtual community between the members of educational process

**CHAPTER 2**

**SYSTEM SPECIFICATIONS**

**2.1 HARDWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| Processor | **:** | Pentium IV Or Higher |
| Memory Size | **:** | 256 GB (Minimum) |
| HDD | **:** | 40 GB (Minimum) |

**2.2 SOFTWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| Operating System | **:** | WINDOWS 10 Or XP |
| Front – End | **:** | HTML,CSS & JS |
| Back - End | **:** | Python with Django Web Framework |
| Language | **:** | Python |
| Data Base | **:** | SQLITE3 |

**CHAPTER 3**

**MODULE DESCRIPTION**

#### **ABOUT ONLINE SCHOOL MANAGEMENT SYSTEM DJANGO PROJECT**

In particular, this school management system project in Python Django focuses mainly on dealing with student-teacher records. Also, the system displays all the available dues amounts. In addition, the system allows managing attendance records. Evidently, this project is divided into three categories: Student, Teacher, and Admin Panel. In an overview of this web application, a student can simply register and start using it. Initially, this action requires approval from the admin. In fact, he/she has to apply for it. However, each application requires action from the administrator which depends upon their approval or disapproval. After confirmation of a student’s account, the user can view his/her details, public notice from admin and teachers. Besides, the student can view their own personal attendance record.

#### **TEACHER’S PANEL**

Likewise, a user can apply himself/herself as a position of teacher. However, each registration requires action from the administrator which depends upon their approval or disapproval. Meanwhile, the user can view the list of available students. In fact, the teacher’s account has the right to manage the attendance for each student. Before stepping on to, teachers can make an announcement from the notice section. With it, the notices are published in public where every user can view them using the system. Talking about the attendance section, the system allows taking it as a whole from a class. Meaning, the user has to select any of the classes, their students, and action for it. Before submitting attendance, the teacher has to provide an attendance date too. Although the role of a teacher is a bit minor, it plays a vital role in maintaining the flow within the system.

#### **ADMIN PANEL**

On the other hand, an admin has full control of the system. An admin manages the proper flow of the system, unlike every other user. The user can have an overview of each data. The admin has the right to approve/decline the various requests. By the way, the admin can only manage students and teachers. Nevertheless, the admin can also maintain the student’s attendance. For this, every step is mentioned above. Having said that, the administrator can also view students’ fees details with their dues. Above all, the user can list out attendance records too. Apart from these, the administrator has the right to publish notices. In fact, every section has its own number of records which can be overseen by the admin panel regarding its total number, pending, and dues.

Last but not least, a clean and simple dashboard is presented with various color combinations for greater user experience while using this School Management System Project in Python Django Framework. For its UI elements, a free open-source CSS framework; Bootstrap is on board with some Vanilla CSS too. Presenting a new Online School System Project in Python Django which includes an admin panel with a student and tocher panel that contains all the essential features to follow up, and a knowledgeable resource for learning purposes.

#### **Available Features:**

* Student Panel
* Teacher Panel
* Admin Panel
* Manage Requests
* Student Management
* Teacher Management
* Student Attendance Management
* View Fees, Pending
* Publish Notice

**CHAPTER 4**

**SYSTEM DESIGN**

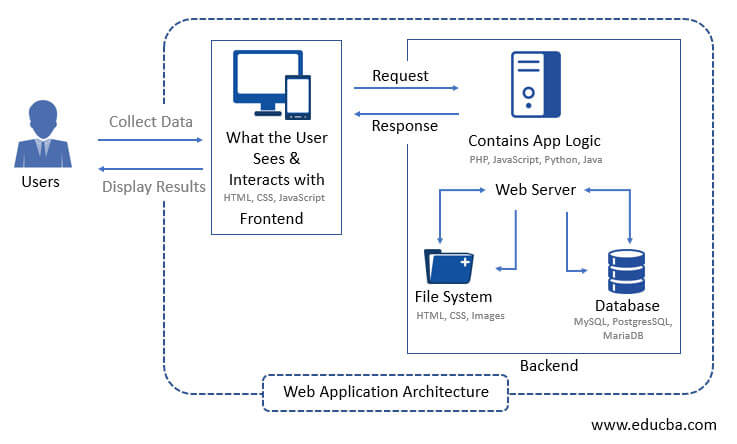
**4.1 ARCHITECTURE DIAGRAM**

The term system architecture is used to describe the overall design and structure of a computer network or system. A method is required to organize and connect these items together in a cohesive manner. The term is also used to describe complex computer\_ software tools. Systems Architecture is a generic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of the objects or it is the conceptual model that defines the structure, behavior, and more views of a system.

There are three main components to any system architecture of the system theses includes: storage, connectivity, and user experience. It is important to note that system architecture must be flexible and able to meet changing needs quickly. Processing power is the brain of the system.

To installing the correct allocation of processors to the system must be based on the software specifications, number of concurrent users, strength of the connection, and applications.

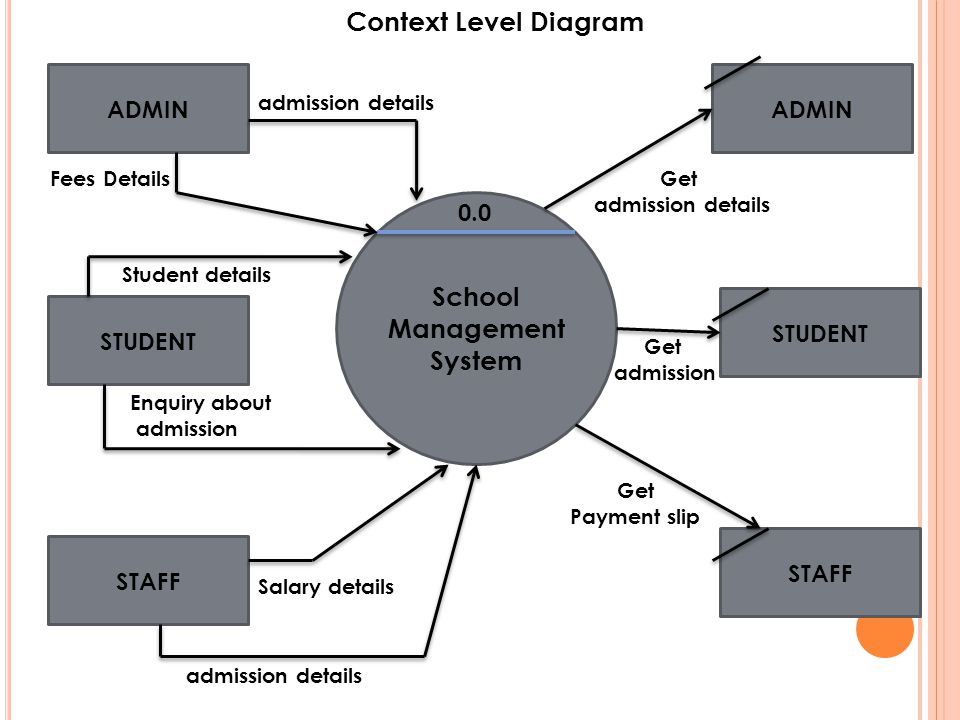
The following architecture shows how the system looks like to implement.

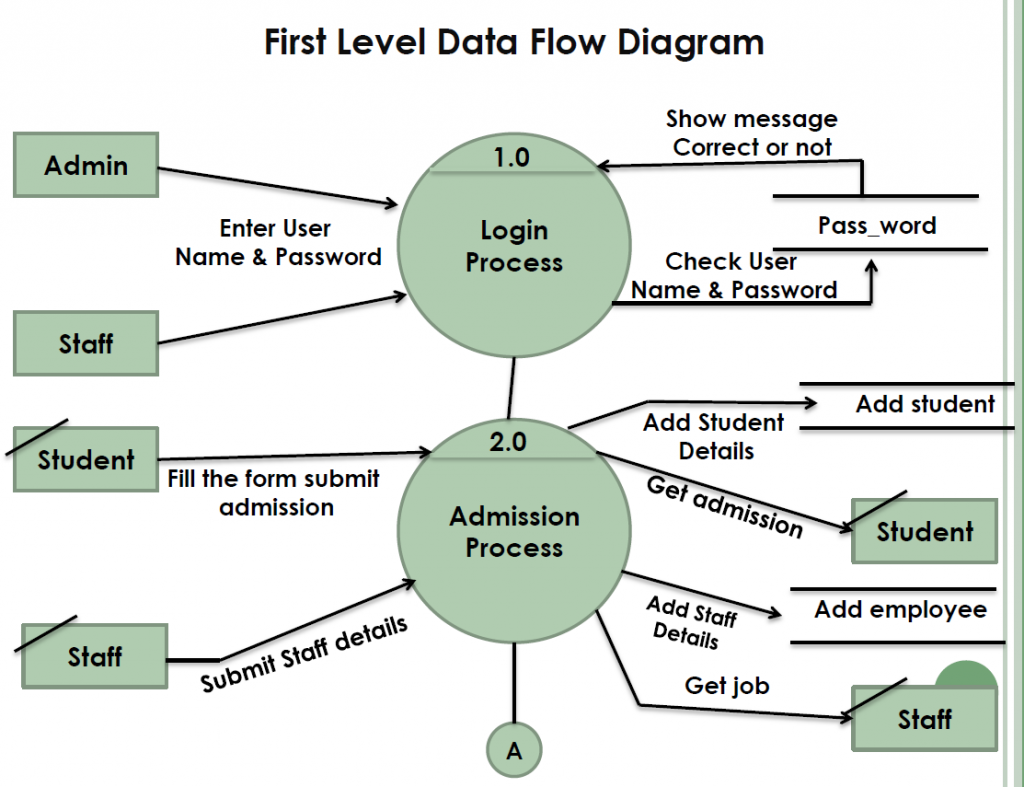


System Architecture of the system

**4.2 FLOW CHART**

**Data Flow Diagram - Context Diagram**





**CHAPTER 5**

**SAMPLE CODING**

**Manage.py**

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks."""

import os

import sys

def main():

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'schoolmanagement.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**settings.py**

import os

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

TEMPLATE\_DIR = os.path.join(BASE\_DIR,'templates')

STATIC\_DIR=os.path.join(BASE\_DIR,'static')

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'k0ujs9pcw+7qohwas!o7\_ept20$c@$)-b=qco8sgviy\_f)((bc'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'school',

'widget\_tweaks',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'schoolmanagement.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [TEMPLATE\_DIR,],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'schoolmanagement.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

# Password validation

# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC\_URL = '/static/'

STATICFILES\_DIRS=[

STATIC\_DIR,

]

LOGIN\_REDIRECT\_URL='/afterlogin'

#for contact us give your gmail id and password

EMAIL\_BACKEND ='django.core.mail.backends.smtp.EmailBackend'

EMAIL\_HOST = 'smtp.gmail.com'

EMAIL\_USE\_TLS = True

EMAIL\_PORT = 587

EMAIL\_HOST\_USER = 'from@gmail.com' # this email will be used to send emails

EMAIL\_HOST\_PASSWORD = 'xyz' # host email password required

# now sign in with your host gmail account in your browser

# open following link and turn it ON

# https://myaccount.google.com/lesssecureapps

# otherwise you will get SMTPAuthenticationError at /contactus

# this process is required because google blocks apps authentication by default

EMAIL\_RECEIVING\_USER = ['to@gmail.com'] # email on which you will receive messages sent from website

**urls.py**

from django.contrib import admin

from django.urls import path

from school import views

from django.contrib.auth.views import LoginView,LogoutView

urlpatterns = [

path('admin/', admin.site.urls),

path('',views.home\_view,name=''),

path('adminclick', views.adminclick\_view),

path('teacherclick', views.teacherclick\_view),

path('studentclick', views.studentclick\_view),

path('adminsignup', views.admin\_signup\_view),

path('studentsignup', views.student\_signup\_view,name='studentsignup'),

path('teachersignup', views.teacher\_signup\_view),

path('adminlogin', LoginView.as\_view(template\_name='school/adminlogin.html')),

path('studentlogin', LoginView.as\_view(template\_name='school/studentlogin.html')),

path('teacherlogin', LoginView.as\_view(template\_name='school/teacherlogin.html')),

path('afterlogin', views.afterlogin\_view,name='afterlogin'),

path('logout', LogoutView.as\_view(template\_name='school/index.html'),name='logout'),

path('admin-dashboard', views.admin\_dashboard\_view,name='admin-dashboard'),

path('admin-teacher', views.admin\_teacher\_view,name='admin-teacher'),

path('admin-add-teacher', views.admin\_add\_teacher\_view,name='admin-add-teacher'),

path('admin-view-teacher', views.admin\_view\_teacher\_view,name='admin-view-teacher'),

path('admin-approve-teacher', views.admin\_approve\_teacher\_view,name='admin-approve-teacher'),

path('approve-teacher/<int:pk>', views.approve\_teacher\_view,name='approve-teacher'),

path('delete-teacher/<int:pk>', views.delete\_teacher\_view,name='delete-teacher'),

path('delete-teacher-from-school/<int:pk>', views.delete\_teacher\_from\_school\_view,name='delete-teacher-from-school'),

path('update-teacher/<int:pk>', views.update\_teacher\_view,name='update-teacher'),

path('admin-view-teacher-salary', views.admin\_view\_teacher\_salary\_view,name='admin-view-teacher-salary'),

path('admin-student', views.admin\_student\_view,name='admin-student'),

path('admin-add-student', views.admin\_add\_student\_view,name='admin-add-student'),

path('admin-view-student', views.admin\_view\_student\_view,name='admin-view-student'),

path('delete-student-from-school/<int:pk>', views.delete\_student\_from\_school\_view,name='delete-student-from-school'),

path('delete-student/<int:pk>', views.delete\_student\_view,name='delete-student'),

path('update-student/<int:pk>', views.update\_student\_view,name='update-student'),

path('admin-approve-student', views.admin\_approve\_student\_view,name='admin-approve-student'),

path('approve-student/<int:pk>', views.approve\_student\_view,name='approve-student'),

path('admin-view-student-fee', views.admin\_view\_student\_fee\_view,name='admin-view-student-fee'),

path('admin-attendance', views.admin\_attendance\_view,name='admin-attendance'),

path('admin-take-attendance/<str:cl>', views.admin\_take\_attendance\_view,name='admin-take-attendance'),

path('admin-view-attendance/<str:cl>', views.admin\_view\_attendance\_view,name='admin-view-attendance'),

path('admin-fee', views.admin\_fee\_view,name='admin-fee'),

path('admin-view-fee/<str:cl>', views.admin\_view\_fee\_view,name='admin-view-fee'),

path('admin-notice', views.admin\_notice\_view,name='admin-notice'),

path('teacher-dashboard', views.teacher\_dashboard\_view,name='teacher-dashboard'),

path('teacher-attendance', views.teacher\_attendance\_view,name='teacher-attendance'),

path('teacher-take-attendance/<str:cl>', views.teacher\_take\_attendance\_view,name='teacher-take-attendance'),

path('teacher-view-attendance/<str:cl>', views.teacher\_view\_attendance\_view,name='teacher-view-attendance'),

path('teacher-notice', views.teacher\_notice\_view,name='teacher-notice'),

path('student-dashboard', views.student\_dashboard\_view,name='student-dashboard'),

path('student-attendance', views.student\_attendance\_view,name='student-attendance'),

path('aboutus', views.aboutus\_view),

path('contactus', views.contactus\_view),

]

**SQLITE3 DATA BASE:**

-- TABLE

CREATE TABLE "auth\_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE);

CREATE TABLE "auth\_group\_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group\_id" integer NOT NULL REFERENCES "auth\_group" ("id") DEFERRABLE INITIALLY DEFERRED, "permission\_id" integer NOT NULL REFERENCES "auth\_permission" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE TABLE "auth\_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content\_type\_id" integer NOT NULL REFERENCES "django\_content\_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename" varchar(100) NOT NULL, "name" varchar(255) NOT NULL);

CREATE TABLE "auth\_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last\_login" datetime NULL, "is\_superuser" bool NOT NULL, "username" varchar(150) NOT NULL UNIQUE, "first\_name" varchar(30) NOT NULL, "email" varchar(254) NOT NULL, "is\_staff" bool NOT NULL, "is\_active" bool NOT NULL, "date\_joined" datetime NOT NULL, "last\_name" varchar(150) NOT NULL);

CREATE TABLE "auth\_user\_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user\_id" integer NOT NULL REFERENCES "auth\_user" ("id") DEFERRABLE INITIALLY DEFERRED, "group\_id" integer NOT NULL REFERENCES "auth\_group" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE TABLE "auth\_user\_user\_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user\_id" integer NOT NULL REFERENCES "auth\_user" ("id") DEFERRABLE INITIALLY DEFERRED, "permission\_id" integer NOT NULL REFERENCES "auth\_permission" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE TABLE "django\_admin\_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "action\_time" datetime NOT NULL, "object\_id" text NULL, "object\_repr" varchar(200) NOT NULL, "change\_message" text NOT NULL, "content\_type\_id" integer NULL REFERENCES "django\_content\_type" ("id") DEFERRABLE INITIALLY DEFERRED, "user\_id" integer NOT NULL REFERENCES "auth\_user" ("id") DEFERRABLE INITIALLY DEFERRED, "action\_flag" smallint unsigned NOT NULL CHECK ("action\_flag" >= 0));

CREATE TABLE "django\_content\_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app\_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL);

CREATE TABLE "django\_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL);

CREATE TABLE "django\_session" ("session\_key" varchar(40) NOT NULL PRIMARY KEY, "session\_data" text NOT NULL, "expire\_date" datetime NOT NULL);

CREATE TABLE "school\_attendance" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "date" date NOT NULL, "cl" varchar(10) NOT NULL, "present\_status" varchar(10) NOT NULL, "roll" varchar(10) NULL);

CREATE TABLE "school\_notice" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "date" date NOT NULL, "message" varchar(500) NOT NULL, "by" varchar(20) NULL);

CREATE TABLE "school\_studentextra" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "roll" varchar(10) NOT NULL, "user\_id" integer NOT NULL UNIQUE REFERENCES "auth\_user" ("id") DEFERRABLE INITIALLY DEFERRED, "cl" varchar(10) NOT NULL, "fee" integer unsigned NULL CHECK ("fee" >= 0), "mobile" varchar(40) NULL, "status" bool NOT NULL);

CREATE TABLE "school\_teacherextra" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "joindate" date NOT NULL, "mobile" varchar(40) NOT NULL, "user\_id" integer NOT NULL UNIQUE REFERENCES "auth\_user" ("id") DEFERRABLE INITIALLY DEFERRED, "status" bool NOT NULL, "salary" integer unsigned NOT NULL CHECK ("salary" >= 0));

CREATE TABLE sqlite\_sequence(name,seq);

-- INDEX

CREATE INDEX "auth\_group\_permissions\_group\_id\_b120cbf9" ON "auth\_group\_permissions" ("group\_id");

CREATE UNIQUE INDEX "auth\_group\_permissions\_group\_id\_permission\_id\_0cd325b0\_uniq" ON "auth\_group\_permissions" ("group\_id", "permission\_id");

CREATE INDEX "auth\_group\_permissions\_permission\_id\_84c5c92e" ON "auth\_group\_permissions" ("permission\_id");

CREATE INDEX "auth\_permission\_content\_type\_id\_2f476e4b" ON "auth\_permission" ("content\_type\_id");

CREATE UNIQUE INDEX "auth\_permission\_content\_type\_id\_codename\_01ab375a\_uniq" ON "auth\_permission" ("content\_type\_id", "codename");

CREATE INDEX "auth\_user\_groups\_group\_id\_97559544" ON "auth\_user\_groups" ("group\_id");

CREATE INDEX "auth\_user\_groups\_user\_id\_6a12ed8b" ON "auth\_user\_groups" ("user\_id");

CREATE UNIQUE INDEX "auth\_user\_groups\_user\_id\_group\_id\_94350c0c\_uniq" ON "auth\_user\_groups" ("user\_id", "group\_id");

CREATE INDEX "auth\_user\_user\_permissions\_permission\_id\_1fbb5f2c" ON "auth\_user\_user\_permissions" ("permission\_id");

CREATE INDEX "auth\_user\_user\_permissions\_user\_id\_a95ead1b" ON "auth\_user\_user\_permissions" ("user\_id");

CREATE UNIQUE INDEX "auth\_user\_user\_permissions\_user\_id\_permission\_id\_14a6b632\_uniq" ON "auth\_user\_user\_permissions" ("user\_id", "permission\_id");

CREATE INDEX "django\_admin\_log\_content\_type\_id\_c4bce8eb" ON "django\_admin\_log" ("content\_type\_id");

CREATE INDEX "django\_admin\_log\_user\_id\_c564eba6" ON "django\_admin\_log" ("user\_id");

CREATE UNIQUE INDEX "django\_content\_type\_app\_label\_model\_76bd3d3b\_uniq" ON "django\_content\_type" ("app\_label", "model");

CREATE INDEX "django\_session\_expire\_date\_a5c62663" ON "django\_session" ("expire\_date");

-- TRIGGER

-- VIEW

**CHAPTER 6**

**SCREEN SHOTS**

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**CONCLUSION**

Conclusion In recent years, with the pace of technological development, people have become more and more demanding in terms of quality of life, and the schools managers in recent years look to improve a performance in their schools to get the highest rate of knowledge and experience in their student.

**FUTURE ENHANCEMENT**

Some ideas and features can be considered as a future work for this project. These features can be summarized in the following points:

* Let the student perform exams Online.
* Bank of question per subject.
* Let teacher add questions to the bank of question.
* Let admin add new classes.
* Add a medical examination part to the system, with a medical supervisor as a new user.

**REFERENCES**

1. Vikas Kulshreshtha, Sharad Maheshwari. (2011).”Blood Bank Management Information System in India”, International Journal of Engineering, 1,2, 260-263.
2. Rational Unified Process, Best Practices for Software Development Teams. (2012).
3. Core Workflows Retrieved from www.ibm.com/developerworks/rational/.../1251\_bestpractices Noushin Ashrafi, & Hessam Ashrafi. (2008). Object Oriented Systems Analysis and Design, Pearson Higher Ed USA.
4. Lions Blood Bank & Research Foundation. (2012).
5. Retrieved from http://www.lionsbloodbank.net/ Blood Bank India. (2012).
6. Retrieved from http://www.bloodbankindia.net