

A PROJECT REPORT ON
SCHOOL MANAGEMENT SYSTEM

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

BACHELOR OF COMPUTER APPLICATION (BCA)
TO
GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY ,DELHI

By
Chirag Kapoor
(35290202020)

UNDER THE GUIDENCE OF
Mrs. DIVYA VERMA



SESSION:2020-2023

SRI GURU TEGH BAHUDUR INSTITUTE OF MANAGEMENTAND
INFORMATION TECHNOLOGY, DELHI – 110033
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY,
DELHI)

DECLARATION

I hereby declare that the project entitled is an outcome of our own efforts under the guidance of Mrs. Divya Verma. The Project is submitted to Guru Gobind Singh Indraprastha University. For the Partial Fulfilment of the Bachelor of Computer Application.

I also declare that this project report has not been previously submitted to any other University.

Chirag Kapoor

(35290202020)

CERTIFICATE

Certified that this minor project report titled “School Management System” prepared by Chirag Kapoor is an authentic work carried out by the Sri Guru Tegh Bahadur Institute of Management And Information Technology, under the guidance of Mrs. Divya Verma. This report has not been degree/diploma/certificate.

Date:

Guide-Mrs. Divya Verma

HOD (IT)- Dr. Raj Kumar

ACKNOWLEDGEMENT

I would like to express my special thanks to all the people who helped and supported me during the making of this project. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me throughout the duration of this project.

In the accomplishment of completion of our project on School management System. I would like to express my special gratitude to **Sri Guru Tegh Bahadur Institute of Management & Information Technology** for giving us the opportunity to work under their guidance and help us gain immense professional experience. A sincere thanks to our mentor Mrs. Divya Verma for giving us valuable inputs and ideas right from the selection of Topic for project till its successful completion. The successful completion of our project would not have been possible without the dedicated support from our mentor.

Abstract

A school management system can be defined as a platform designed to enable the efficient running of your institution through digitization and automation of various academic and administrative operations. The software will play the role of a school data management system and allow you complete jobs involving bulk data management flawlessly and quickly.

The pandemic has changed the way the world used to operate. The biggest change has possibly been experienced by educational institutions. This has made it mandatory for all schools to use intelligently designed school management software

.

Table of Content

| S. NO. | Chapter | Page No. |
|--------|--|----------|
| 1 | List of Tables | 8 |
| 2 | List of Figures | 8 |
| 3 | List of Symbols | 9 |
| 4 | <p style="text-align: center;">Introduction</p> <p>1.1) Objective of the Project</p> <p>1.2) Scope of the Project</p> | 10-11 |
| 5 | <p style="text-align: center;">Requirement And Analysis</p> <p>2.1) Introduction</p> <p style="padding-left: 20px;">2.1.1) Project Planning</p> <p>2.2) Software Development Life Cycle</p> <p>2.3) Software Requirement</p> <p>2.4) Hardware Requirement</p> <p>2.5) Technologies Used</p> <p>2.6) Use Case Diagram</p> | 12-16 |
| 6 | <p style="text-align: center;">Software Design</p> <p>3.1) 0 Level DFD</p> <p>3.2) 1 Level DFD</p> | 17-18 |
| 7 | <p style="text-align: center;">Database Design</p> <p>4.1) Tables</p> <p>4.2) ER Diagram</p> | 19-22 |
| 8 | <p style="text-align: center;">Testing</p> <p>5.1) Introduction</p> <p>5.2) Methods Employed for testing</p> <p style="padding-left: 20px;">5.2.1) Unit testing</p> <p style="padding-left: 40px;">5.2.2) Integration testing</p> | 23-26 |

| | | |
|----|--|--------|
| | 5.2.3) Functional testing 5.2.4) System testing 5.3) Test- Cases | |
| 9 | Roles And Responsibilities 6.1) Roles 6.2) Responsibilities | 27 |
| 10 | Conclusion And Future Scope 7.1) Conclusion 7.2) Future Scope | 28 |
| 11 | Appendices 8.1)Code 8.2) Snapshots | 29-100 |
| 12 | Bibliography | 101 |



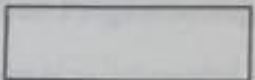
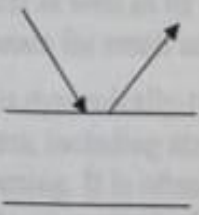
List of Tables

| FIGURE NO. | DESCRIPTION | PAGE NO. |
|-------------------|--------------------|-----------------|
| 1. | Login Details | 19 |
| 2. | Attendance Details | 19 |
| 3. | Student Details | 20 |
| 4. | Teacher Details | 20 |
| 5. | Notice | 20 |

List of Figures

| FIGURE NO. | DESCRIPTION | PAGE NO. |
|-------------------|--------------------|-----------------|
| 1. | USE CASE DIAGRAM | 16 |
| 2. | DFD LEVEL 0 | 17 |
| 3. | DFD LEVEL 1 | 18 |
| 4. | ER DIAGRAM | 22 |

List of symbols

| Symbol | Name | Function |
|---|----------------|---|
|  | Data Flow | Connect process |
|  | Process | Perform some transformation of its input data to yield output data. |
|  | Source or sink | A source of system inputs or sink of system outputs |
|  | Data Store | A repository of data, the arrowhead indicate net input and net outputs to store |

Chapter 1: Introduction

A “School Management System” is an information management system for educational institutions to manage student data. It helps teachers get information about students faster, easier and reduces their workload. School management systems provide skills such as student registration. In addition, school management systems are used to plan the curriculum of students, record their presence and manage the needs of students in the school.

As digital platform, we eliminate the need of pen and paper for keeping the details of students and teachers. The student can see, fill his/her details digitally regarding admission and other details. The Administrator can keep all records of students, teachers

You can use this software for the school to conduct the admission process virtually. The software will make the steps like form submission and admission tracking extremely easy. As a result, the administrator will be able to process more forms and the chances of human errors will be almost zero.

Teachers and the school admin can gain access to admission details and personal information of students within just seconds thanks to School Management Software. You can also use the software to add student registration, add teacher etc.

Python Programming Language

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020.

1.1) Objective of the Project

The School Management System is important because it simplifies the school administration and other activities with its features. It manage all the information about student and teacher. The project is built on administrative point of view and thus only the administrator is guaranteed the access. The purpose of the project is to build the application program to reduce the manual work for managing the Student, Teacher. you can record the attendance, fee etc of students with this software. This provides the users with a single database to store every data in it.it tracks all the details about the student and Teacher.

1.2) Scope of the Project

This System is aimed at total user-friendly as well as efficient management of varied tasks. These tasks may range from registering new students, Teacher to all the essential features necessary for making the administrative division of school effective. In order to cope up with all these factors, the school management system was developed and nowadays, it has even been recognized by most of the Indian schools. As a matter of fact, this system based on smart technology has become an integral part of many schools.

Level of Access:

The System have three level of Access:

- The Admin
- Teacher
- Student

Chapter-2: Requirement and Analysis

If a document that completely describes what the proposed software should do without describing how the software will do it. The basic goal of requirements specific to produce the requirement, which describes the complete external behavior of the proposed software. However, producing the report is said to be done.

2.1) Introduction

With the advent of 21st century, students have become smarter and learn at a faster pace. During such times, it is always a delightful experience for the institution to upgrade from the old traditional model to a platform which is easy to manage as well as smart. Schools, these days, try to adopt a modern learning system with [school management system software](#). This paves way for better administration and help teachers to interact with students on regular basis.

2.1.1) Project Planning

I was assigned the duty for Developing the backend for “School Management System”. Working in team reinstates the team for some common guidelines and standard to be followed by all the team members across all team. For optimum use of practical time it is necessary that every session is planned. Planning of this project will include the following things:

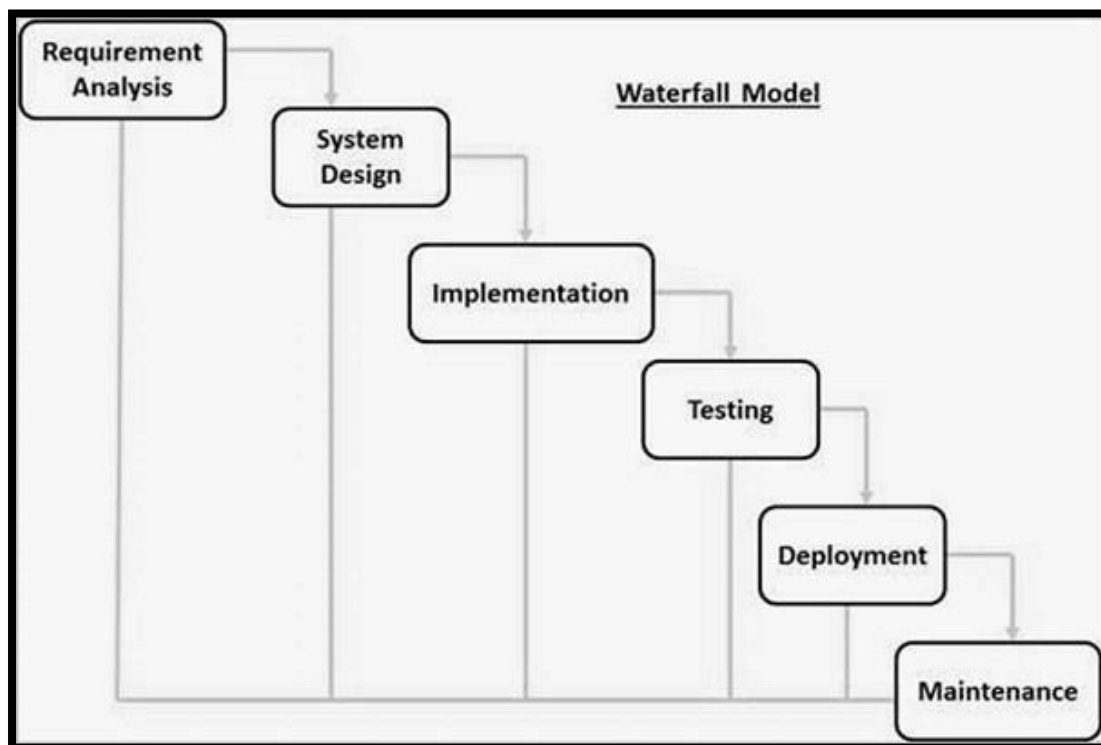
- Topic Understanding
- Module break-up of the system
- Processor logic for each Module
- Database Requirement

2.2) Software Development Life Cycle Model

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality soft wares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

Some popular SDLC models followed in the industry: Waterfall Model, Iterative Model, Spiral Model, V-Model, Big Bang Model etc.

Our Project is based on Waterfall Model.



Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Advantages-

- ❖ The advantages of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
- ❖ The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use. • It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- ❖ In this model, phases are processed and completed one at a time and they do not overlap. Waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages-

- ❖ **It is difficult to estimate time and cost for each phase of the development process.**
- ❖ **Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.**
- ❖ **Not a good model for complex and object-oriented projects.**
- ❖ **Not suitable for the projects where requirements are at a moderate to high risk of changing.**
- ❖ **Phases in Waterfall Model:**

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

2.3) Software Requirements:

- 1) Python IDLE
- 2) Database: MySQL
- 3) PHP
- 4) CSS

2.4) Hardware Requirement

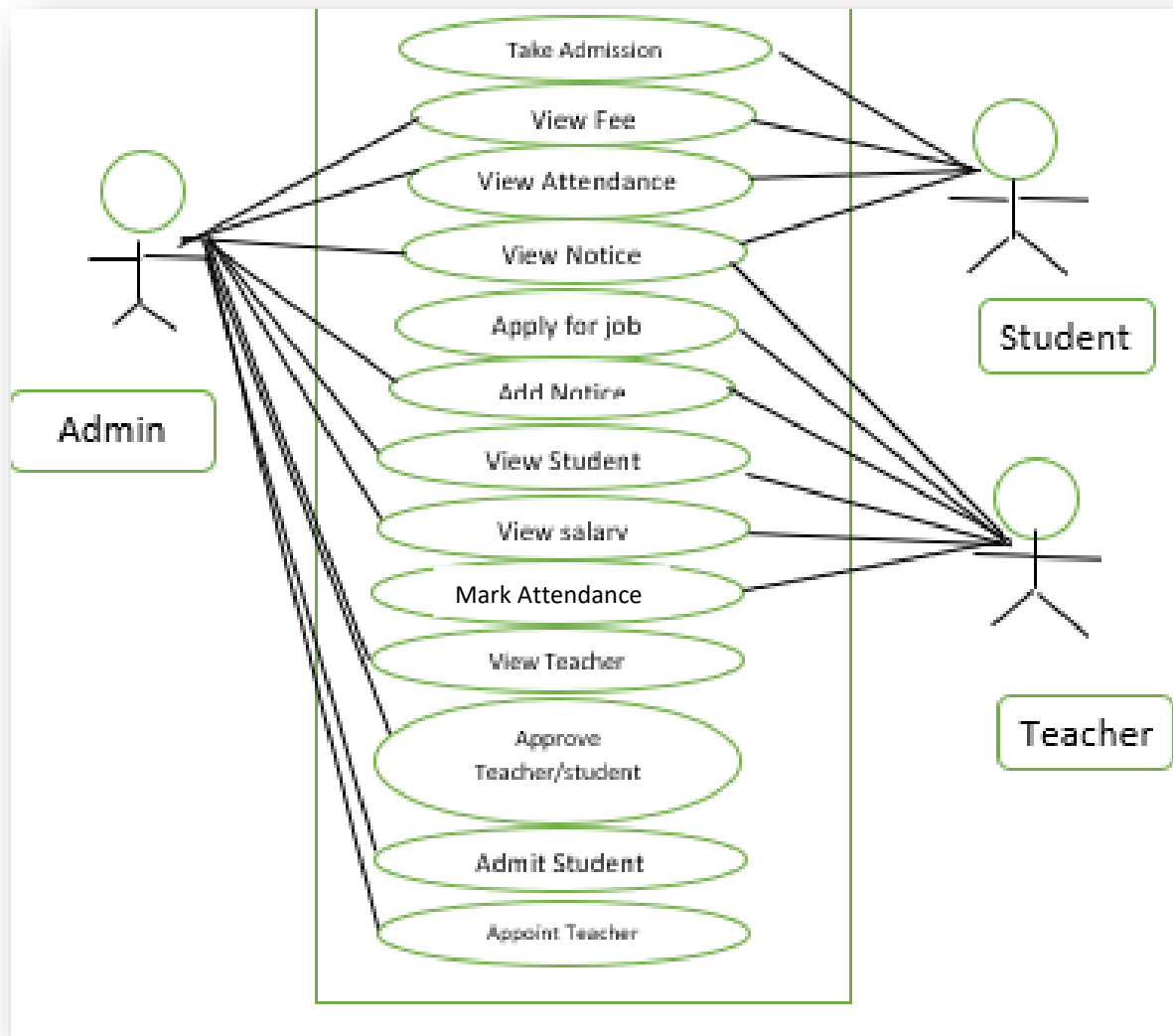
- 1) Device: PC/Laptop 32/64 Bit System
- 2) Storage Size: 512 GB
- 3) Ram: 8GB
- 4) Space On Disc: 16GB
- 5) Processor: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz

2.5) Technologies Used:

| | |
|------------------|----------------------------|
| Operating System | Window 10 |
| Technologies | IDLE Python 3.9.10(64 bit) |
| Tools | Python IDLE |
| Database | SQLite3 |

2.6) Use Case Diagram:

Use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

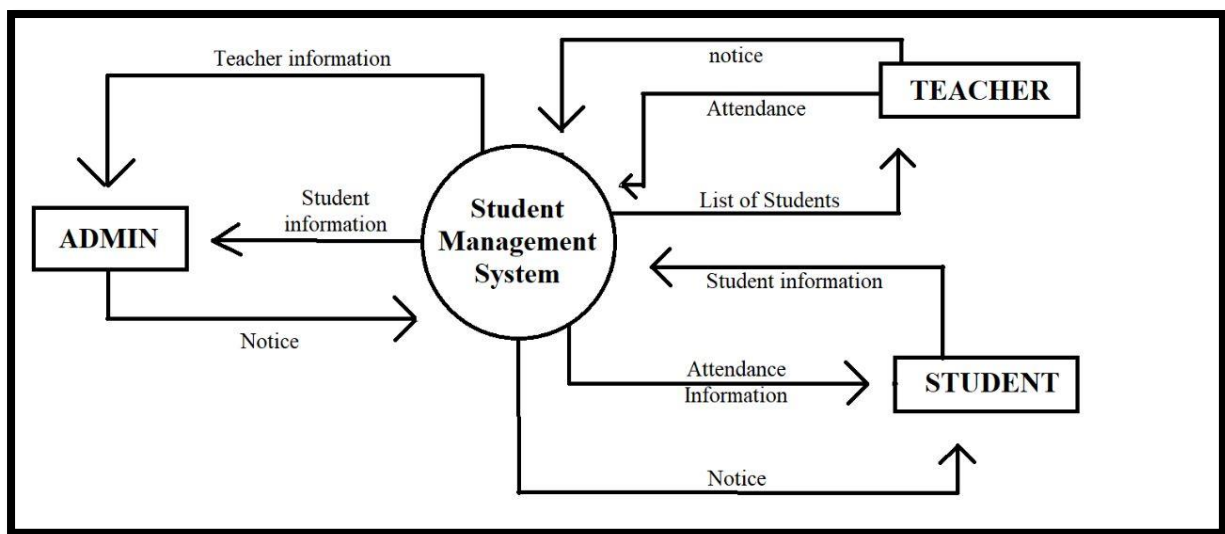


Chapter-3: Software Design

This chapter will focus on the design of the system using diagram to illustrate graphically certain section of the software system.

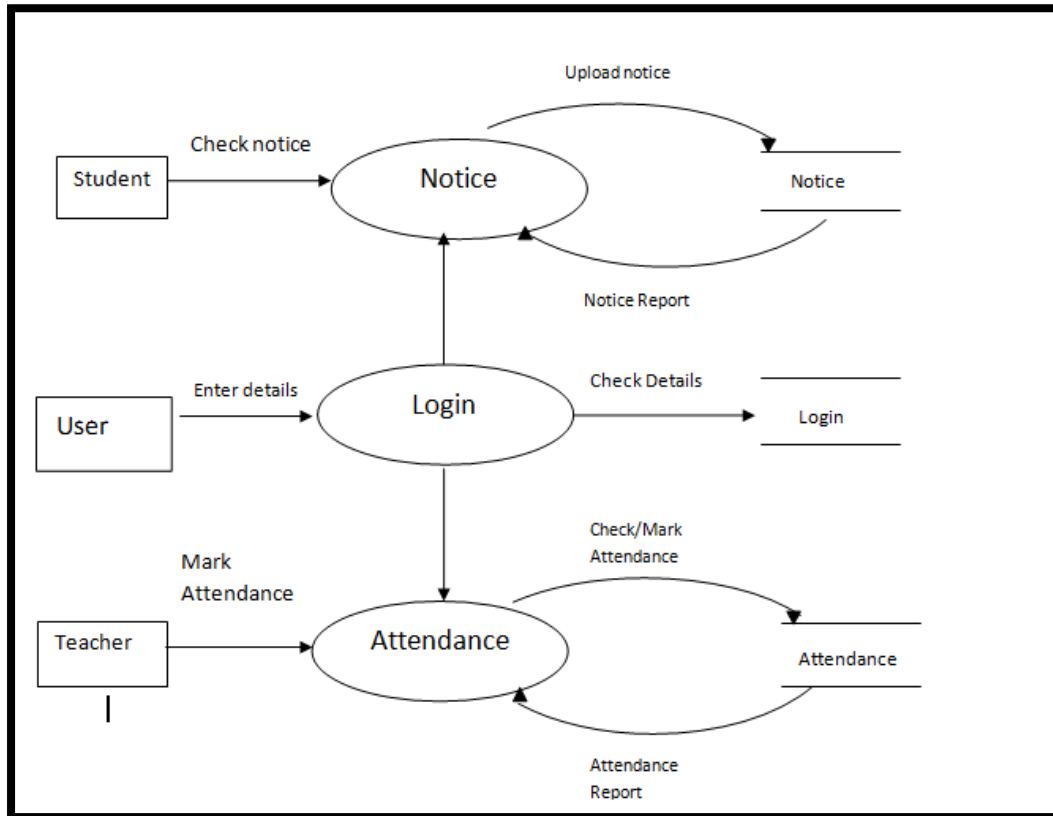
3.1) 0 Level DFD

It is also known as a context diagram .It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.



3.2) 1 Level DFD

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes. The level 1 of the Data Flow Diagram gives explain in detail about packet watching system which was marked under as 0 in the previous level.



Chapter 4: Database Design

4.1) Introduction

This article/tutorial will teach the basis of relational database design and explain how to make a good database design. It is a rather long text, but we advise to read all of it. Designing a database is in fact fairly easy, but there are a few rules to stick to. It is important to know what these rules are, but more importantly is to know why these rules exist, otherwise you will tend to make mistakes!

Database Fields Specification:

A small write-up on the database, giving the fields, explaining each etc, should be written. This write-up can be evaluated on the following points:

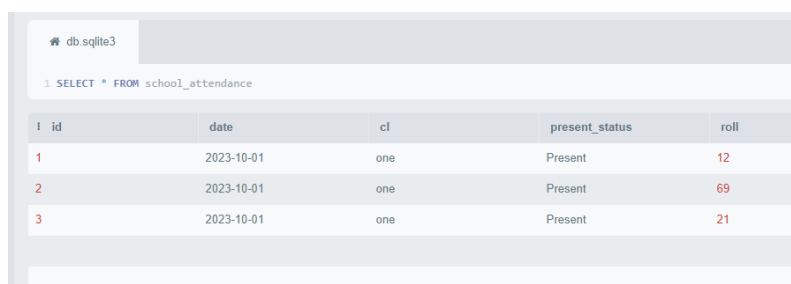
1. Clarity and conciseness of the database design .Like ,whether key is defined, whether any redundant fields are there etc.
2. Whether data-storage calculations are made, and if so ,whether they are done properly etc.
3. Whether any data-backup/recovery mechanism is discussed or being thought of.

4.2) Tables:

SQL TABLES all tables



| # | id | password | last_login | is_superuser | username | first_name | email | is_staff | is_active | date_joi... | last_name |
|----|----|-------------|-------------|--------------|-----------|------------|-------|----------|-----------|-------------|-----------|
| 1 | 1 | pbkdf2_s... | 2023-01-... | 0 | chirag@1 | chirag | | 0 | 1 | 2023-01-... | kapoor |
| 2 | 2 | pbkdf2_s... | NULL | 0 | sahiltut | Sahil | | 0 | 1 | 2023-01-... | tuteja |
| 3 | 3 | pbkdf2_s... | 2023-01-... | 0 | hardika@1 | Hardika | | 0 | 1 | 2023-01-... | khurana |
| 4 | 4 | pbkdf2_s... | 2023-01-... | 0 | maddy | madhav | | 0 | 1 | 2023-01-... | sharma |
| 5 | 5 | pbkdf2_s... | 2023-01-... | 0 | hardika26 | Hardika | | 0 | 1 | 2023-01-... | Khurana |
| 6 | 6 | pbkdf2_s... | 2023-01-... | 0 | kush33 | Kush | | 0 | 1 | 2023-01-... | Jain |
| 7 | 7 | pbkdf2_s... | 2023-01-... | 0 | Simran | Harsimran | | 0 | 1 | 2023-01-... | Kaur |
| 8 | 8 | pbkdf2_s... | 2023-01-... | 0 | saloni69 | saloni | | 0 | 1 | 2023-01-... | mittal |
| 9 | 9 | pbkdf2_s... | 2023-01-... | 0 | anu2 | anu | | 0 | 1 | 2023-01-... | sharma |
| 10 | 10 | pbkdf2_s... | 2023-01-... | 0 | vani | Vanshika | | 0 | 1 | 2023-01-... | Garg |



| # | id | date | cl | present_status | roll |
|---|----|------------|-----|----------------|------|
| 1 | 1 | 2023-10-01 | one | Present | 12 |
| 2 | 2 | 2023-10-01 | one | Present | 69 |
| 3 | 3 | 2023-10-01 | one | Present | 21 |

db.sqlite3

```
1 SELECT * FROM school_notice
```

| id | date | message | by |
|----|------------|-----------------|------|
| 1 | 2023-01-10 | hello students. | Kush |
| 2 | 2023-01-18 | hello students. | Kush |

db.sqlite3

```
1 SELECT * FROM school_studentextra
```

| id | Fathers_name | user_id | cl | fee | mobile | status |
|----|--------------|---------|-------|-------|-------------|--------|
| 1 | Raj Kapoor | 1 | one | 10000 | 1234567890 | 1 |
| 2 | Shiva | 4 | six | 15000 | 97896748775 | 1 |
| 3 | Vijay | 7 | six | 15000 | 9664567867 | 1 |
| 4 | Vishnu | 8 | one | 10000 | 86764656856 | 1 |
| 5 | Vihaan | 9 | one | 10000 | 8767565786 | 1 |
| 6 | Hemant | 10 | three | 10000 | 8767568221 | 0 |

db.sqlite3

```
1 SELECT * FROM school_teacherextra
```

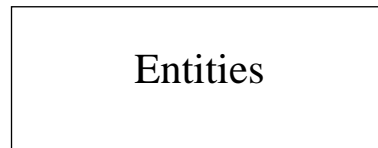
| id | joindate | mobile | user_id | status | salary |
|----|------------|------------|---------|--------|--------|
| 1 | 2023-01-09 | 8756445678 | 6 | 1 | 30000 |

4.3) ER Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

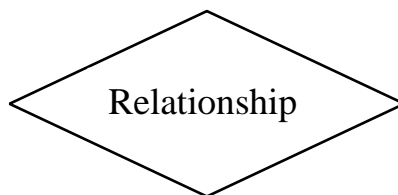
Entities

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.



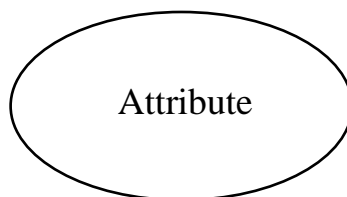
Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box

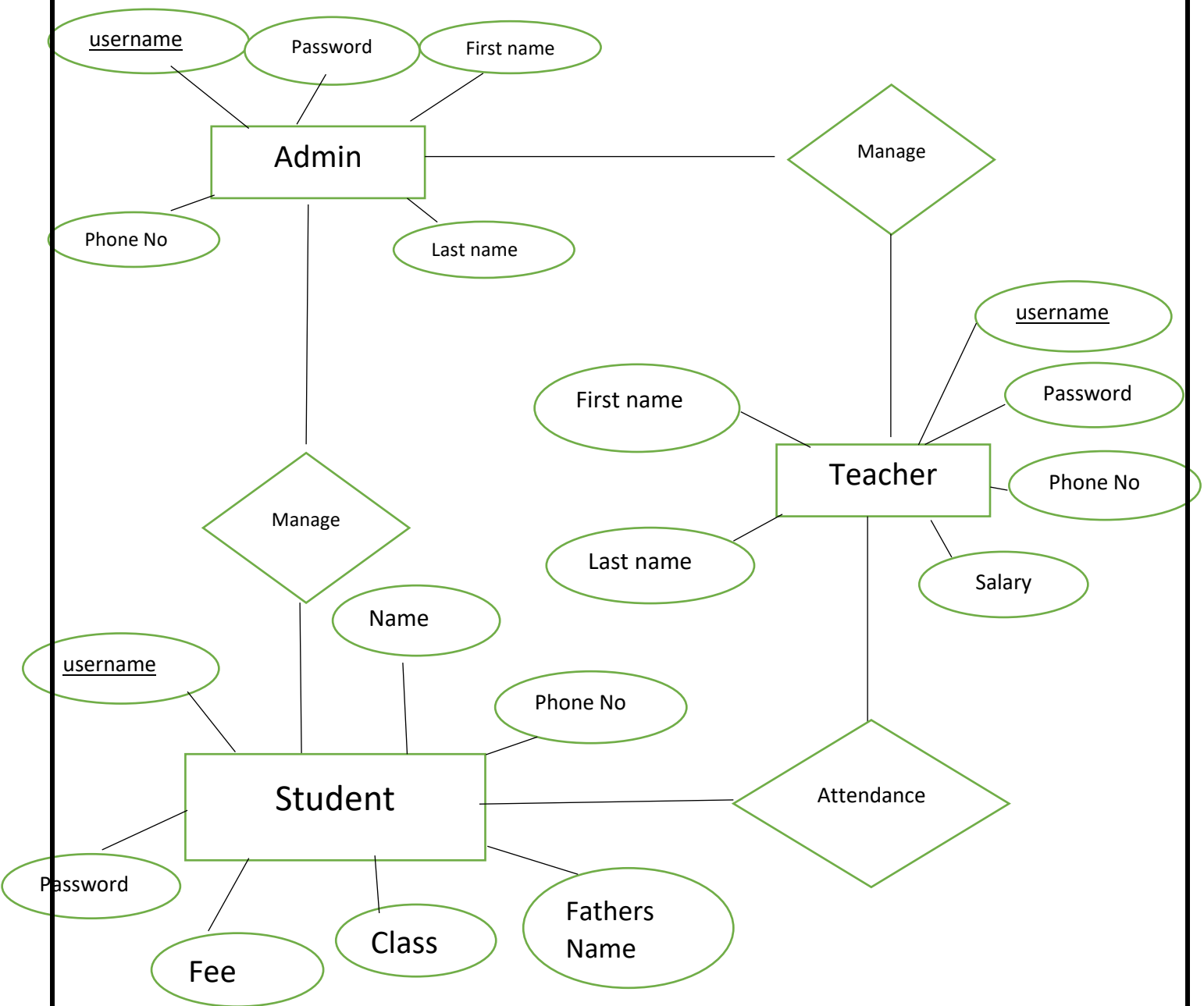


Attributes

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



ER DIAGRAM-



Chapter 5: Testing

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

Types of Testing:

- **Manual Testing:**
Manual testing is a type of testing in which we do not take the help of any tools (automation) to perform the testing. In this testing, testers make test cases for the codes and test the software and give the final report about that software. Manual testing is a time consuming testing because it is done by humans and there is a chance of human errors.
- **Automation Testing:**
Automation testing is a type of testing in which we take the help of tools (automation) to perform the testing. It is faster than manual testing because it is done by some automation tools. There is no chance of any human errors.

5.2) Methods employed for Testing

5.2.1) Unit Testing

Unit testing refers to the testing of individual software components or modules. In Unit testing, the source code is divided into modules, which in turn are divided into smaller units called units. The test done on these units of code is called Unit test. Unit test depends upon the language on which the project is developed. Unit test ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2.2) Integration Testing

In Integration testing, the modules are combined and tested as a group to verify the combined functionality of modules after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. Integration Testing follows unit testing.

5.2.3) Functional Testing

Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

5.2.4) System Testing

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is defined as a series of different tests whose sole purpose is to exercise the full computer-based system.

System testing consists of the following steps:

- Program Testing
- System Testing
- System Documentation
- User Acceptance Testing

5.3 Test-Cases(TC)

| Testing Case ID | Description | Expected Result | Actual result | Remarks |
|-----------------|---|------------------|------------------|---------|
| TC01 | Login (Student): a) Enter your Details b) Password | Login Successful | Login Successful | Pass |
| TC02 | Login with wrong Id But details are right | Login successful | Login Successful | Pass |
| TC03 | Login (Teacher): a) Enter your Details b) Password | Login Successful | Login Successful | Pass |
| TC04 | Admin: a) Sign Up b) Login | Login Successful | Successful | Pass |
| TC05 | View Fees: a) Student Login b) Password | Login Successful | Login Successful | Pass |
| TC06 | View Student Attendance: a) Student Login b) Password | Successful | Successful | Pass |
| TC07 | Teacher View: a) Teacher Login b) Password | Login Successful | Login Successful | Pass |

| | | | | |
|------|---------------------------------------|------------------|------------------|------|
| TC08 | Teacher Salary | Login Successful | Login Successful | Pass |
| TC09 | Notice: (For Student and Teachers) | Login Successful | Login Successful | Pass |

Chapter 6: Roles and Responsibilities

6.1) Roles:

Work as Developer(back-end)

6.2)Responsibilities:

Make sure to achieve all the back end designing and requirement of the software.

Chapter 7: Conclusion And Future Scope

7.1) Conclusion

This school management system is necessary because it is an efficient way of managing colleges, schools, and other educational institutions, allowing schools to reduce the weight of their processes. School administration software makes it simple and effective to keep track of everything. A school management system is a platform created to facilitate the smooth functioning of your institution by digitising and automating a variety of academic and administrative processes. Overall, the school management system has a significant positive impact on the lives of administrators, teachers, and students. Better productivity under good management leads to greater development advancement. As a result, they may concentrate more on assuring the welfare of the teachers, enhancing the performance of the entire team, and making better hiring decisions.

7.2) Future Scope

Anything done to improve the quality of education at any stage may be ranging from the supply of material, human and financial resources to the highest cultural or academic needs-comes under the scope of educational management. This scope of school management is very vast. The use of software for school management will allow students to interact with learning communities from across the world. Students across the world with similar learning styles and similar interests can connect with the help of the app, thus creating a diverse and interactive learning culture. This will not only give students a wider exposure to different cultures, but it will lead to a collaboration of ideas across borders. The advanced school ERP will make school administration in the future easier than it has ever been. Schools will no longer have to struggle with heaps of paper records, as they will be able to maintain all records of the institution in a digital format thus making its access and management really easy. This paper-free environment will make it easier for doing jobs such as data syncing, reports generation and report sharing. Additionally, the school ERP will ensure that digital data can be tracked and retrieved with the click of a button.

Thus for a successful school management software, it is essential that organization consider a mobile app that can be easily accessed by students, teacher, parents and school staff alike

Chapter-8 Appendices

➤ PHP and CSS Coding-

Navbar

```
<!DOCTYPE html>

{ % load static % }

<html lang="en">

<head>

<meta charset="utf-8">

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<style type="text/css">

    .bs-example {
        margin: 0px;
    }

    .navbar-brand {
        font-size: 20px;
        font-family: sans-serif;
    }

</style>

</head>

<body>
```

```

<div class="bs-example">
<nav class="navbar navbar-expand-md bg-dark navbar-dark fixed-top">
<a href="/" class="navbar-brand">School Management</a>
<button type="button" class="navbar-toggler" data-toggle="collapse" data-
target="#navbarCollapse">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">
<div class="navbar-nav">
<a href="/" class="nav-item nav-link active">Home</a>
<a href="/adminclick" class="nav-item nav-link">Admin</a>
<a href="/teacherclick" class="nav-item nav-link">Teacher</a>
<a href="/studentclick" class="nav-item nav-link">Student</a>
</div>
<div class="navbar-nav">
<a href="/aboutus" class="nav-item nav-link">About Us</a>
<a href="/contactus" class="nav-item nav-link">Contact Us</a>
</div>
</div>
</nav>
</div>
</body>
</html>

```

Index

```

<!DOCTYPE html>
{ % load static % }
<html lang="en" dir="ltr">

```

```

<head>
<meta charset="utf-8">
<style media="screen">
.jumbotron {
    margin-top: 0px;
    margin-bottom: 0px;
    background-image: url('{ % static "images/bg2.jpg" % }');
    background-size: cover;
    background-repeat: no-repeat;
}
.jumbotron h1 {
    text-align: center;
}
.alert {
    margin: 0px;
}
.glow {
    font-size: 70px;
    color: #ffffff;
    text-align: center;
    -webkit-animation: glow 1s ease-in-out infinite alternate;
    -moz-animation: glow 1s ease-in-out infinite alternate;
    animation: glow 1s ease-in-out infinite alternate;
}
@-webkit-keyframes glow {
    from {
        text-shadow: 0 0 10px #eeeeee, 0 0 20px #000000, 0 0 30px #000000, 0 0 40px
#000000, 0 0 50px #9554b3, 0 0 60px #9554b3, 0 0 70px #9554b3;
    }
    to {

```

```

        text-shadow: 0 0 20px #eeeeee, 0 0 30px #ff4da6, 0 0 40px #ff4da6, 0 0 50px #ff4da6, 0
0 60px #ff4da6, 0 0 70px #ff4da6, 0 0 80px #ff4da6;
    }
}
</style>
</head>
<body>
    {% include "school/navbar.html" %}
<br>
<br>
<div class="jumbotron" style="margin-bottom: 0px;margin-top: 0px;">
<br>
<h1 class="display-4 glow">Welcome</h1>
<br><br>
<p>Knowledge is key to success.</p>
<br><br><br><br><br>
<p class="lead">
<a class="btn btn-primary btn-lg" href="/studentsignup" role="button">Take Admission</a>
</p>
<br><br>
</div>
<br>
<br>
<br><br>
    {% include "school/admin_teacher_student_card.html" %}
<br><br>
<br>
<br>
<br>
<br>
    {% include "school/footer.html" %}

```



```
</body>
```

```
</html>
```

Home base

```
<!DOCTYPE html>
```

```
<html lang="en" dir="ltr">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<style media="screen">
```

```
.jumbotron {
```

```
    margin-top: 0px;
```

```
    margin-bottom: 0px;
```

```
}
```

```
.jumbotron h1 {
```

```
    text-align: center;
```

```
}
```

```
.alert {
```

```
    margin: 0px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
{% include "school/navbar.html" %}
```

```
{%block content%}
```

```
{%endblock content%}
```

```
{% include "school/footer.html" %}
```

```
</body>
```

```
</html>
```

Admin_teacher_student_card

```
<!DOCTYPE html>
```

```
{% load static %}

<html>

<head>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

<style>

.card {

    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

    max-width: 300px;

    margin: auto;

    text-align: center;

    font-family: arial;

}

.title {

    color: grey;

    font-size: 18px;

}

button {

    border: none;

    outline: 0;

    display: inline-block;

    padding: 8px;

    color: white;

    background-color: #000;

    text-align: center;

    cursor: pointer;

    width: 100%;

    font-size: 18px;

}
```

```

button:hover,
a:hover {
    opacity: 0.7;
}

.grid-container {
    display: grid;
    grid-template-columns: auto autoauto;
    padding: 10px;
}

a:link {
    text-decoration: none;
}

a {
    color: white;
}

</style>
</head>
<body>
<div class="grid-container">
<div class="grid-item">
<div class="card">
<imgsrc="{ % static "images/admin.png" % }" alt="John" style="width:100%">
<p class="title">ADMIN</p>
<p><button><a href="/adminclick">View</a></button></p>
</div>

</div>

<div class="grid-item">
<div class="card">

```

```

<imgsrc="{ % static "images/teacher.png" % }" alt="John" style="width:100%">
<p class="title">TEACHER</p>
<p><button><a href="/teacherclick">View</a></button></p>
</div>
</div>
<div class="grid-item">
<div class="card">
<imgsrc="{ % static "images/student1.png" % }" alt="John" style="width:100%">
<p class="title">STUDENT</p>
<p><button><a href="/studentclick">View</a></button></p>
</div>
</div>
</div>
</body>
</html>

```

Footer

```

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
<style>
/*-----
Social section
-----*/
footer {
padding: 0px 0px0px0px;
background-color: black;
margin: 0px;
}

```

```
.fa {  
    padding: 20px;  
    font-size: 23px;  
    width: 60px;  
    text-align: center;  
    text-decoration: none;  
    margin: 5px 2px;  
    border-radius: 50%;  
}
```

```
.fa:hover {  
    opacity: 0.5;  
    text-decoration: none;  
}
```

```
.fa-facebook {  
    background: #3B5998;  
    color: white;  
    margin-top: 30px;  
}
```

```
.fa-whatsapp {  
    background: #25d366;  
    color: white;  
}
```

```
.fa-twitter {  
    background: #55ACEE;  
    color: white;  
}
```


Contact us

```
<!DOCTYPE html>

{ % load static % }

<html lang="en" dir="ltr">

<head>

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

</head>

<body>

    { % include "school/navbar.html" % }

<br><br>

<center>

<h3 class='alert alert-success'>Send Us Your Valuable Feedback !</h3>

<form method="POST">

<!-- Very Important csrf Token -->

    { % csrf_token % }

<div class="form-group">

<p>

<h3>{ { form.as_p } }</h3>

</p>

<br>

<input type="submit" value="Send Message" class="btn btn-primary btn-lg">

</div>

</form>

</center>

    { % include "school/footer.html" % }

</body>

</html>
```

About us

```
<!DOCTYPE html>
```

```

{% load static %}

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<style media="screen">

.jumbotron {
    margin-top: 0px;
    margin-bottom: 0px;
}

.jumbotron h1 {
    text-align: center;
}

.alert {
    margin: 0px;
}

</style>

<title></title>

</head>

<body>

    {% include "school/navbar.html" %}

<br><br>

<center>

<h3 class='alert alert-success' style="margin-bottom:0px;">About Us !</h3>

</center>

<div class="jumbotron" style="margin-bottom: 0px;margin-top: 0px;">

```



```

<h1 class="display-4">Hello</h1>

<p class="lead">A service dedicated to Admin, Student and Teacher. We provide online
resources, professional support

    and guidance to all our students,teacher whenever, and from wherever they have chosen
to study.</p>

<hr class="my-4">

<p>Explore our Website.</p>

<p class="lead">

<a class="btn btn-primary btn-lg" href="/" role="button">HOME</a>

</p>

</div>

{% include "school/footer.html" %}

</body>

</html>

```

Admin add Student

```

{% extends 'school/adminbase.html' %}

{% load widget_tweaks %}


{% block content %}


<head>

<style media="screen">

a:link {

    text-decoration: none;

}

.note {

    text-align: center;

    height: 80px;

    background: -webkit-linear-gradient(left, #0072ff, #8811c5);

    color: #fff;

```

```

        font-weight: bold;
        line-height: 80px;
    }

    .form-content {
        padding: 5%;
        border: 1px solid #ced4da;
        margin-bottom: 2%;
    }

    .form-control {
        border-radius: 1.5rem;
    }

    .btnSubmit {
        border: none;
        border-radius: 1.5rem;
        padding: 1%;
        width: 20%;
        cursor: pointer;
        background: #0062cc;
        color: #fff;
    }
</style>

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

```

```

<!------- signup page for student by admin(sumit) ----->
<form method="post">

    { % csrf_token % }

<div class="container register-form">
<div class="form">
<div class="note">
<p>New Admission of Student In School</p>
</div>
<div class="form-content">
<div class="row">
<div class="col-md-6">
<div class="form-group">
        { % render_field form1.first_name class="form-control" placeholder="First Name"
        % }
    </div>
<div class="form-group">
        { % render_field form1.username class="form-control" placeholder="Username" % }
    </div>
<div class="form-group">
        { % render_field form2.mobile class="form-control" placeholder="Mobile" % }
    </div>
<div class="form-group">
        { % render_field form2.cl class="form-control" placeholder="Class" % }
    </div>
</div>
<div class="col-md-6">
<div class="form-group">

```

```

        {% render_field form1.last_name class="form-control" placeholder="Last Name"
    % }
</div>
<div class="form-group">
        {% render_field form1.password class="form-control" placeholder="Password" % }
</div>
<div class="form-group">
        {% render_field form2.fee class="form-control" placeholder="Fee" % }
</div>
<div class="form-group">
        {% render_field form2.roll class="form-control" placeholder="Roll" % }
</div>
</div>
</div>
</div>
<button type="submit" class="btnSubmit">Submit</button>
</div>
</div>
</div>
</form>
{% endblock content % }
Admin Add Teacher {% extends 'school/adminbase.html' % }

{% load widget_tweaks % }

{% block content % }
<head>
<style media="screen">
a:link {
    text-decoration: none;
}

```

```
.note {  
    text-align: center;  
    height: 80px;  
    background: -webkit-linear-gradient(left, #0072ff, #8811c5);  
    color: #fff;  
    font-weight: bold;  
    line-height: 80px;  
}
```

```
.form-content {  
    padding: 5%;  
    border: 1px solid #ced4da;  
    margin-bottom: 2%;  
}
```

```
.form-control {  
    border-radius: 1.5rem;  
}
```

```
.btnSubmit {  
    border: none;  
    border-radius: 1.5rem;  
    padding: 1%;  
    width: 20%;  
    cursor: pointer;  
    background: #0062cc;  
    color: #fff;  
}
```

```
</style>
```

```

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

<!------- signup page for teacher by admin ----->
<form method="post">
    { % csrf_token % }
<div class="container register-form">
<div class="form">
<div class="note">
<p>Hire New Teacher To School</p>
</div>
<div class="form-content">
<div class="row">
<div class="col-md-6">
<div class="form-group">
        { % render_field form1.first_name class="form-control" placeholder="First Name"
% }
    </div>
<div class="form-group">
        { % render_field form1.username class="form-control" placeholder="Username" % }
    </div>
<div class="form-group">
        { % render_field form2.mobile class="form-control" placeholder="Mobile" % }
    </div>
</div>
</div>
<div class="col-md-6">
<div class="form-group">

```

```

        {% render_field form1.last_name class="form-control" placeholder="Last Name"
% }
</div>
<div class="form-group">
        {% render_field form1.password class="form-control" placeholder="Password" % }
</div>
<div class="form-group">
        {% render_field form2.salary class="form-control" placeholder="Salary" % }
</div>
</div>
</div>
<button type="submit" class="btnSubmit">Hire</button>
</div>
</div>
</div>
</form>
{% endblock content % }

```

Admin approve student

```

{% extends 'school/adminbase.html' % }
{% block content % }
<head>
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>

<style media="screen">
a:link {
    text-decoration: none;
}

```

```

    h6 {
        text-align: center;
    }

    .row {
        margin: 100px;
    }
</style>

</head>
<div class="container">
<div class="row">
<div class="panel panel-primary">
<div class="panel-heading">
<h6 class="panel-title">Students</h6>
</div>
<table class="table table-hover" id="dev-table">
<thead>

<tr>
<th>Name</th>
<th>Class</th>
<th>Fee</th>
<th>Mobile</th>
<th>Approve</th>
<th>Delete</th>
</tr>
</thead>
    {% for s in students %}
<tr>
<td> {{ s.get_name }} </td>

```



```

<td>{ {s.cl} }</td>

<td>{ {s.fee} }</td>

<td>{ {s.mobile} }</td>

<td><a class="btn btn-primary btn-xs" href="{ % url 'approve-student' s.id % }"><span
class="glyphicon glyphicon-ok"></span></a></td>

<td><a class="btn btn-danger btn-xs" href="{ % url 'delete-student' s.id % }"><span
class="glyphicon glyphicon-trash"></span></a></td>

</tr>

    { % endfor % }

</table>

</div>

</div>

</div>

{ % endblock content % }

Admin approve teacher

{ % extends 'school/adminbase.html' % }

{ % block content % }


<head>

<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>

<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>

<style media="screen">

a:link {

    text-decoration: none;

}

h6 {

    text-align: center;

}

.row {

```

```

        margin: 100px;
    }
</style>
</head>
<div class="container">
<div class="row">
<div class="panel panel-primary">
<div class="panel-heading">
<h6 class="panel-title">Teachers</h6>
</div>
<table class="table table-hover" id="dev-table">
<thead>
<tr>
<th>Name</th>
<th>Mobile</th>
<th>Salary</th>
<th>Joining Date</th>
<th>Approve</th>
<th>Delete</th>
</tr>
</thead>
        { % for t in teachers % }
<tr>
<td> {{ t.get_name }}</td>
<td> {{ t.mobile }}</td>
<td> {{ t.salary }}</td>
<td> {{ t.joindate }}</td>
<td><a class="btn btn-primary btn-xs" href="{ % url 'approve-teacher' t.id % }"><span
class="glyphicon glyphicon-ok"></span></a></td>
<td><a class="btn btn-danger btn-xs" href="{ % url 'delete-teacher' t.id % }"><span
class="glyphicon glyphicon-trash"></span></a></td>

```

```

</tr>
    {% endfor %}
</table>
</div>
</div>
</div>
{% endblock content %}
Admin attendance
{% extends 'school/adminbase.html' %}

```

```

{% block content %}

```

```

<head>
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<style media="screen">
a:link {
    text-decoration: none;
}

a:link {
    text-decoration: none;
}

h6 {
    text-align: center;
}

```

```

.row {
    margin: 100px;
}
</style>

</head>
<div class="container">

<div class="row">

<div class="panel panel-primary">
<div class="panel-heading">
<h6 class="panel-title">Attendance</h6>

</div>

<table class="table table-hover" id="dev-table">
<thead>

<tr>
<th>Class Name</th>
<th>Take Attendance</th>
<th>View Attendance</th>
</tr>
</thead>

<tr>
<td> One</td>
<td><a class="btn btn-primary btn-xs" href="{ % url 'admin-take-attendance' 'one'
% } "><span class="glyphicon glyphicon-plus"></span></a></td>

```

```
<td><a class="btn btn-danger btn-xs" href="{ % url 'admin-view-attendance' 'one' % }"><span class="glyphicon glyphicon-eye-open"></span></a></td>
```

```
</tr>
```

```
<tr>
```

```
<td> Two</td>
```

```
<td><a class="btn btn-primary btn-xs" href="{ % url 'admin-take-attendance' 'two' % }"><span class="glyphicon glyphicon-plus"></span></a></td>
```

```
<td><a class="btn btn-danger btn-xs" href="{ % url 'admin-view-attendance' 'two' % }"><span class="glyphicon glyphicon-eye-open"></span></a></td>
```

```
</tr>
```

```
<tr>
```

```
<td> Three</td>
```

```
<td><a class="btn btn-primary btn-xs" href="{ % url 'admin-take-attendance' 'three' % }"><span class="glyphicon glyphicon-plus"></span></a></td>
```

```
<td><a class="btn btn-danger btn-xs" href="{ % url 'admin-view-attendance' 'three' % }"><span class="glyphicon glyphicon-eye-open"></span></a></td>
```

```
</tr>
```

```
<tr>
```

```
<td> Four</td>
```

```
<td><a class="btn btn-primary btn-xs" href="{ % url 'admin-take-attendance' 'four' % }"><span class="glyphicon glyphicon-plus"></span></a></td>
```

```
<td><a class="btn btn-danger btn-xs" href="{ % url 'admin-view-attendance' 'four' % }"><span class="glyphicon glyphicon-eye-open"></span></a></td>
```

```
</tr>
```

```
<tr>
```

```
<td> Five</td>
```

```
<td><a class="btn btn-primary btn-xs" href="{ % url 'admin-take-attendance' 'five' % }"><span class="glyphicon glyphicon-plus"></span></a></td>
```

```
<td><a class="btn btn-danger btn-xs" href="{ % url 'admin-view-attendance' 'five' % }"><span class="glyphicon glyphicon-eye-open"></span></a></td>
```

</tr>

<tr>

<td> Six</td>

<td></td>

<td></td>

</tr>

<tr>

<td> Seven</td>

<td></td>

<td></td>

</tr>

<tr>

<td> Eight</td>

<td></td>

<td></td>

</tr>

<tr>

<td> Nine</td>

<td></td>

<td></td>

</tr>

```

<tr>

<td> Ten</td>

<td><a class="btn btn-primary btn-xs" href="{ % url 'admin-take-attendance' 'ten' % }"><span
class="glyphicon glyphicon-plus"></span></a></td>

<td><a class="btn btn-danger btn-xs" href="{ % url 'admin-view-attendance' 'ten' % }"><span
class="glyphicon glyphicon-eye-open"></span></a></td>

</tr>

</table>

</div>

</div>

</div>

{ % endblock content % }

Admin student

{ % extends 'school/adminbase.html' % }

```

```

{ % block content % }

```

```

<head>

<meta charset="utf-8">

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

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<link href="http://netdna.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css"
rel="stylesheet">

<style type="text/css">

a:link {

    text-decoration: none;

}

```

```
h6 {
  color: white;
}

.order-card {
  color: #fff;
}

.bg-c-blue {
  background: linear-gradient(45deg, #4099ff, #73b4ff);
}

.bg-c-green {
  background: linear-gradient(45deg, #2ed8b6, #59e0c5);
}

.bg-c-yellow {
  background: linear-gradient(45deg, #FFB64D, #ffcb80);
}

.bg-c-pink {
  background: linear-gradient(45deg, #FF5370, #ff869a);
}

.card {
  border-radius: 5px;
  -webkit-box-shadow: 0 1px 2.94px 0.06px rgba(4, 26, 55, 0.16);
  box-shadow: 0 1px 2.94px 0.06px rgba(4, 26, 55, 0.16);
  border: none;
}
```



```

        margin-bottom: 30px;
        -webkit-transition: all 0.3s ease-in-out;
        transition: all 0.3s ease-in-out;
    }

    .card .card-block {
        padding: 25px;
    }

    .order-card i {
        font-size: 26px;
    }

    .f-left {
        float: left;
    }

    .f-right {
        float: right;
    }
</style>
</head>

<body>
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css" rel="stylesheet">
<div class="container">
<div class="row">
<div class="col-md-4 col-xl-3">
<div class="card bg-c-blue order-card">
<div class="card-block">

```

```
<a href="/admin-view-student">
<h6 class="m-b-20">View All Student</h6>
</a>
<br>
<h2 class="text-right"><i class="fas fa-eye f-left"></i></h2>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col-md-4 col-xl-3">
<div class="card bg-c-green order-card">
<div class="card-block">
<a href="/admin-add-student">
<h6 class="m-b-20">Add Student</h6>
</a>
<br>
<h2 class="text-right"><i class="fas fa-plus-circle f-left"></i></h2>
</div>
</div>
</div>
```

```
<div class="col-md-4 col-xl-3">
<div class="card bg-c-yellow order-card">
<div class="card-block">
<a href="/admin-approve-student">
<h6 class="m-b-20">Approve Student</h6>
</a>
<br>
<h2 class="text-right"><i class="fas fa-check-circle f-left"></i></h2>
</div>
```

</div>

</div>

<div class="col-md-4 col-xl-3">

<div class="card bg-c-pink order-card">

<div class="card-block">

<h6 class="m-b-20">View Student Fee</h6>

<h2 class="text-right"><i class="fas fa-rupee-sign f-left"></i></h2>

</div>

</div>

</div>

</div>

</div>

<script src="http://netdna.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js"></script>

<script type="text/javascript">

</script>

</body>

{% endblock content % }

Admin take attendance

{% extends 'school/adminbase.html' % }

{% load widget_tweaks % }

{% block content % }

<head>

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
```

```
<style media="screen">
a:link {
    text-decoration: none;
}
```

```
h6 {
    text-align: center;
}
```

```
.row {
    margin: 100px;
}
```

```
.btnSubmit {
    text-align: center;
}
```

```
</style>
```

```
</head>
```

```
<form class="" method="post">
```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="panel panel-primary">
```

```
<div class="panel-heading">
```

```
<h6 class="panel-title">Attendance</h6>
```

```
</div>
```

```

<table class="table table-hover" id="dev-table">
<thead>
<tr>
<th>Student Name</th>
<th>Present/Absent</th>
</tr>
</thead>

    {% csrf_token %}

    {% for student in students %}

<tr>
<td>{{ student.get_name }}</td>
<td>{{ aform.present_status }}</td>
</tr>

    {% endfor %}

</table>

    {% render_field aform.date class="form-control" placeholder="Enter Date mm/dd/yyyy" %}

<input type="submit" style="margin-left: 200px;
    margin-top: 20px;
    margin-bottom: 20px; background-color: #ffec00;" name="submit" value="submit">
</div>
</div>
</div>
</form>

{% endblock content %}

Admin teacher

{% extends 'school/adminbase.html' %}

```

```
{% block content %}
```

```
<head>
```

```
<meta charset="utf-8">
```

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

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
```

```
<link href="http://netdna.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css"
rel="stylesheet">
```

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

```
a:link {
```

```
    text-decoration: none;
```

```
}
```

```
h6 {
```

```
    color: white;
```

```
}
```

```
.order-card {
```

```
    color: #fff;
```

```
}
```

```
.bg-c-blue {
```

```
    background: linear-gradient(45deg, #4099ff, #73b4ff);
```

```
}
```

```
.bg-c-green {  
  background: linear-gradient(45deg, #2ed8b6, #59e0c5);  
}
```

```
.bg-c-yellow {  
  background: linear-gradient(45deg, #FFB64D, #ffcb80);  
}
```

```
.bg-c-pink {  
  background: linear-gradient(45deg, #FF5370, #ff869a);  
}
```

```
.card {  
  border-radius: 5px;  
  -webkit-box-shadow: 0 1px 2.94px 0.06px rgba(4, 26, 55, 0.16);  
  box-shadow: 0 1px 2.94px 0.06px rgba(4, 26, 55, 0.16);  
  border: none;  
  margin-bottom: 30px;  
  -webkit-transition: all 0.3s ease-in-out;  
  transition: all 0.3s ease-in-out;  
}
```

```
.card .card-block {  
  padding: 25px;  
}
```

```
.order-card i {  
  font-size: 26px;  
}
```

```

.f-left {
    float: left;
}

.f-right {
    float: right;
}
</style>
</head>

<body>

<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-
awesome.min.css" rel="stylesheet">

<div class="container">

<div class="row">

<div class="col-md-4 col-xl-3">

<div class="card bg-c-blue order-card">

<div class="card-block">

<a href="admin-view-teacher">

<h6 class="m-b-20">View All Teacher</h6>

</a>

<br>

<h2 class="text-right"><i class="fas fa-eye f-left"></i></h2>

</div>

</div>

</div>

<div class="col-md-4 col-xl-3">

<div class="card bg-c-green order-card">

<div class="card-block">

```



```
<a href="admin-add-teacher">
<h6 class="m-b-20">Add Teacher</h6>
</a>
<br>
<h2 class="text-right"><i class="fas fa-plus-circle f-left"></i></h2>
</div>
</div>
</div>
```

```
<div class="col-md-4 col-xl-3">
<div class="card bg-c-yellow order-card">
<div class="card-block">
<a href="admin-approve-teacher">
<h6 class="m-b-20">Approve Teacher</h6>
</a>
<br>
<h2 class="text-right"><i class="fas fa-check-circle f-left"></i></h2>
</div>
</div>
</div>
```

```
<div class="col-md-4 col-xl-3">
<div class="card bg-c-pink order-card">
<div class="card-block">
<a href="/admin-view-teacher-salary">
<h6 class="m-b-20">View Teachers Salary</h6>
</a>
<br>
<h2 class="text-right"><i class="fas fa-rupee-sign f-left"></i></h2>
</div>
</div>
```

```

</div>

</div>

<br><br><br><br><br><br><br>

</div>

<script src="http://netdna.bootstrapcdn.com/bootstrap/4.0.0-
beta/js/bootstrap.min.js"></script>

<script type="text/javascript">

</script>

</body>

{% endblock content %}

Admin update teacher

{% extends 'school/adminbase.html' %}

{% load widget_tweaks %}

{% block content %}


<head>

<style media="screen">

a:link {

    text-decoration: none;

}


.note {

    text-align: center;

    height: 80px;

    background: -webkit-linear-gradient(left, #0072ff, #8811c5);

    color: #fff;

    font-weight: bold;

    line-height: 80px;

}


.form-content {

```

```

padding: 5%;
border: 1px solid #ced4da;
margin-bottom: 2%;
}

.form-control {
border-radius: 1.5rem;
}

.btnSubmit {
border: none;
border-radius: 1.5rem;
padding: 1%;
width: 20%;
cursor: pointer;
background: #0062cc;
color: #fff;
}
</style>

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

<!------- signup page for teacher by admin(sumit) ----->
<form method="post">
{ % csrf_token % }
<div class="container register-form">

```

```

<div class="form">
<div class="note">
<p>Update Teacher Details</p>
</div>
<div class="form-content">
<div class="row">
<div class="col-md-6">
<div class="form-group">
    { % render_field form1.first_name class="form-control" placeholder="First Name"
    % }
</div>
<div class="form-group">
    { % render_field form1.username class="form-control" placeholder="Username" % }
</div>
<div class="form-group">
    { % render_field form2.mobile class="form-control" placeholder="Mobile" % }
</div>
</div>
<div class="col-md-6">
<div class="form-group">
    { % render_field form1.last_name class="form-control" placeholder="Last Name"
    % }
</div>
<div class="form-group">
    { % render_field form1.password class="form-control" placeholder="Password" % }
</div>
<div class="form-group">
    { % render_field form2.salary class="form-control" placeholder="Salary" % }
</div>
</div>
</div>

```

```
<button type="submit" class="btnSubmit">Submit</button>
</div>
</div>
</div>
</form>
{% endblock content %}
```

Admin view attendance page

```
{% extends 'school/adminbase.html' %}
```

```
{% block content %}
```

```
<head>
```

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
```

```
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
```

```
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
```

```
<style media="screen">
```

```
a:link {
```

```
    text-decoration: none;
```

```
}
```

```
h6 {
```

```
    text-align: center;
```

```
}
```

```
.row {
```

```
    margin: 100px;
```

```
}
```

```

</style>

</head>

<div class="container">
<div class="row">
<div class="panel panel-primary">
<div class="panel-heading">
<h6 class="panel-title">Attendance of class {{cl}} and date {{date}}</h6>
</div>
<table class="table table-hover" id="dev-table">
<thead>
<tr>
<th>Student Name</th>
<th>Student Roll</th>
<th>Present/Absent</th>
</tr>
</thead>
    {%for attendancedata,studentdata in mylist %}
<tr>
<td>{{ studentdata.get_name }}</td>
<td>{{ studentdata.roll }}</td>
<td>{{ attendancedata.present_status }}</td>
</tr>
    {%endfor%}
</table>
</div>
</div>
</div>

{% endblock content %}

Admin view fee

{% extends 'school/adminbase.html' %}

```

{% block content %}

<head>

<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>

<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>

<style media="screen">

a:link {

text-decoration: none;

}

h6 {

text-align: center;

}

.row {

margin: 100px;

}

</style>

</head>

<div class="container">

<div class="row">

<div class="panel panel-primary">

<div class="panel-heading">

<h6 class="panel-title">FEE of class {{ cl }} </h6>

</div>

<table class="table table-hover" id="dev-table">

<thead>

```

<tr>
<th>Student Name</th>
<th>Student Roll</th>
<th>Student Mobile</th>
<th>Dues</th>
</tr>
</thead>
    {%for f in feedetails %}
<tr>
<td>{{ f.get_name }}</td>
<td>{{ f.roll }}</td>
<td>{{ f.mobile }}</td>
<td>{{ f.fee }}</td>
</tr>
    {%endfor%}
</table>
</div>
</div>
</div>
{% endblock content %}

```

➤ Python Coding-

Main.py

```

from django.contrib import admin
from .models import Attendance,StudentExtra,TeacherExtra,Notice
class StudentExtraAdmin(admin.ModelAdmin):
    pass
admin.site.register(StudentExtra, StudentExtraAdmin)

class TeacherExtraAdmin(admin.ModelAdmin):

```



```
pass
admin.site.register(TeacherExtra, TeacherExtraAdmin)
```

```
class AttendanceAdmin(admin.ModelAdmin):
    pass
admin.site.register(Attendance, AttendanceAdmin)
```

```
class NoticeAdmin(admin.ModelAdmin):
    pass
admin.site.register(Notice, NoticeAdmin)
```

```
Apps.py
from django.apps import AppConfig
```

```
class SchoolConfig(AppConfig):
    name = 'school'
```

```
forms.py
from django import forms
from django.contrib.auth.models import User
from . import models
```

```
#for admin
class AdminSigupForm(forms.ModelForm):
    class Meta:
        model=User
        fields=['first_name','last_name','username','password']
```

```
#for student related form
class StudentUserForm(forms.ModelForm):
```

```

class Meta:
    model=User
    fields=['first_name','last_name','username','password']
class StudentExtraForm(forms.ModelForm):
    class Meta:
        model=models.StudentExtra
        fields=['roll','cl','mobile','fee','status']
#for teacher related form
class TeacherUserForm(forms.ModelForm):
    class Meta:
        model=User
        fields=['first_name','last_name','username','password']
class TeacherExtraForm(forms.ModelForm):
    class Meta:
        model=models.TeacherExtra
        fields=['salary','mobile','status']
#for Attendance related form
presence_choices=(('Present','Present'),('Absent','Absent'))
class AttendanceForm(forms.Form):
    present_status=forms.ChoiceField( choices=presence_choices)
    date=forms.DateField()

class AskDateForm(forms.Form):
    date=forms.DateField()

#for notice related form
class NoticeForm(forms.ModelForm):
    class Meta:
        model=models.Notice
        fields='__all__'
#for contact us page

```

```
class ContactusForm(forms.Form):
    Name = forms.CharField(max_length=30)
    Email = forms.EmailField()
    Message = forms.CharField(max_length=500,widget=forms.Textarea(attrs={'rows': 3,
'cols': 30})))
```

Models.py

```
from django.db import models
from django.contrib.auth.models import User
# Create your models here.
class TeacherExtra(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    salary = models.PositiveIntegerField(null=False)
    joindate=models.DateField(auto_now_add=True)
    mobile = models.CharField(max_length=40)
    status=models.BooleanField(default=False)
    def __str__(self):
        return self.user.first_name
    @property
    def get_id(self):
        return self.user.id
    @property
    def get_name(self):
        return self.user.first_name+" "+self.user.last_name
```

```
classes=[('one','one'),('two','two'),('three','three'),
('four','four'),('five','five'),('six','six'),('seven','seven'),('eight','eight'),('nine','nine'),('ten','ten')]
class StudentExtra(models.Model):
```

```

user=models.OneToOneField(User,on_delete=models.CASCADE)
roll = models.CharField(max_length=10)
mobile = models.CharField(max_length=40,null=True)
fee=models.PositiveIntegerField(null=True)
cl= models.CharField(max_length=10,choices=classes,default='one')
status=models.BooleanField(default=False)

@property
def get_name(self):
    return self.user.first_name+" "+self.user.last_name

@property
def get_id(self):
    return self.user.id

def __str__(self):
    return self.user.first_name

class Attendance(models.Model):
    roll=models.CharField(max_length=10,null=True)
    date=models.DateField()
    cl=models.CharField(max_length=10)
    present_status = models.CharField(max_length=10)

class Notice(models.Model):
    date=models.DateField(auto_now=True)
    by=models.CharField(max_length=20,null=True,default='school')
    message=models.CharField(max_length=500)

```

Views.py

```

from django.shortcuts import render,redirect,reverse
from . import forms,models
from django.db.models import Sum
from django.contrib.auth.models import Group
from django.http import HttpResponseRedirect

```

```

from django.contrib.auth.decorators import login_required,user_passes_test
from django.conf import settings
from django.core.mail import send_mail

def home_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'school/index.html')
#for showing signup/login button for teacher(by sumit)
def adminclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'school/adminclick.html')
#for showing signup/login button for teacher(by sumit)
def teacherclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'school/teacherclick.html')

#for showing signup/login button for student(by sumit)
def studentclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'school/studentclick.html')

def admin_signup_view(request):
    form=forms.AdminSigupForm()
    if request.method=='POST':
        form=forms.AdminSigupForm(request.POST)
        if form.is_valid():

```

```

        user=form.save()
user.set_password(user.password)
user.save()
my_admin_group = Group.objects.get_or_create(name='ADMIN')
my_admin_group[0].user_set.add(user)

        return HttpResponseRedirect('adminlogin')
        return render(request,'school/adminsignup.html',{'form':form})

def student_signup_view(request):
    form1=forms.StudentUserForm()
    form2=forms.StudentExtraForm()
mydict={'form1':form1,'form2':form2}
    if request.method=='POST':
        form1=forms.StudentUserForm(request.POST)
        form2=forms.StudentExtraForm(request.POST)
        if form1.is_valid() and form2.is_valid():
            user=form1.save()
user.set_password(user.password)
user.save()
            f2=form2.save(commit=False)
            f2.user=user
            user2=f2.save()

my_student_group = Group.objects.get_or_create(name='STUDENT')
my_student_group[0].user_set.add(user)

        return HttpResponseRedirect('studentlogin')
        return render(request,'school/studentsignup.html',context=mydict)
def teacher_signup_view(request):
    form1=forms.TeacherUserForm()

```

```

        form2=forms.TeacherExtraForm()
mydict={'form1':form1,'form2':form2}
    if request.method=='POST':
        form1=forms.TeacherUserForm(request.POST)
        form2=forms.TeacherExtraForm(request.POST)
        if form1.is_valid() and form2.is_valid():
            user=form1.save()
user.set_password(user.password)
user.save()
        f2=form2.save(commit=False)
        f2.user=user
        user2=f2.save()

my_teacher_group = Group.objects.get_or_create(name='TEACHER')
my_teacher_group[0].user_set.add(user)

    return HttpResponseRedirect('teacherlogin')
    return render(request,'school/teachersignup.html',context=mydict)

#for checking user is teacher , student or admin(by sumit)
def is_admin(user):
    return user.groups.filter(name='ADMIN').exists()
def is_teacher(user):
    return user.groups.filter(name='TEACHER').exists()
def is_student(user):
    return user.groups.filter(name='STUDENT').exists()

def afterlogin_view(request):
    if is_admin(request.user):
        return redirect('admin-dashboard')

```

[illegible]


```
studentfee=models.StudentExtra.objects.filter(status=True).aggregate(Sum('fee',default=0))
pendingstudentfee=models.StudentExtra.objects.filter(status=False).aggregate(Sum('fee'))
```

```
notice=models.Notice.objects.all()
```

#aggregate function return dictionary so fetch data from dictionary(by sumit)

```
mydict={
    'teachercount':teachercount,
    'pendingteachercount':pendingteachercount,

    'studentcount':studentcount,
    'pendingstudentcount':pendingstudentcount,

    'teachersalary':teachersalary['salary__sum'],
    'pendingteachersalary':pendingteachersalary['salary__sum'],

    'studentfee':studentfee['fee__sum'],
    'pendingstudentfee':pendingstudentfee['fee__sum'],

    'notice':notice

}
```

```
return render(request,'school/admin_dashboard.html',context=mydict)
```

[illegible]

```
@login_required(login_url='adminlogin')
```

```
@user_passes_test(is_admin)
```

```
def admin_teacher_view(request):
```

```

        return render(request,'school/admin_teacher.html')

@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_add_teacher_view(request):
    form1=forms.TeacherUserForm()
    form2=forms.TeacherExtraForm()
    mydict={'form1':form1,'form2':form2}
    if request.method=='POST':
        form1=forms.TeacherUserForm(request.POST)
        form2=forms.TeacherExtraForm(request.POST)
        if form1.is_valid() and form2.is_valid():
            user=form1.save()
            user.set_password(user.password)
            user.save()

            f2=form2.save(commit=False)
            f2.user=user
            f2.status=True
            f2.save()

    my_teacher_group = Group.objects.get_or_create(name="TEACHER")
    my_teacher_group[0].user_set.add(user)

    return HttpResponseRedirect('admin-teacher')
    return render(request,'school/admin_add_teacher.html',context=mydict)

@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_view_teacher_view(request):

```

```

        teachers=models.TeacherExtra.objects.all().filter(status=True)

        return render(request,'school/admin_view_teacher.html',{'teachers':teachers})

@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_approve_teacher_view(request):
    teachers=models.TeacherExtra.objects.all().filter(status=False)

    return render(request,'school/admin_approve_teacher.html',{'teachers':teachers})


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def approve_teacher_view(request,pk):
    teacher=models.TeacherExtra.objects.get(id=pk)
    teacher.status=True
    teacher.save()

    return redirect(reverse('admin-approve-teacher'))


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def delete_teacher_view(request,pk):
    teacher=models.TeacherExtra.objects.get(id=pk)
    user=models.User.objects.get(id=teacher.user_id)
    user.delete()
    teacher.delete()

    return redirect('admin-approve-teacher')


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def delete_teacher_from_school_view(request,pk):
    teacher=models.TeacherExtra.objects.get(id=pk)

```

```

        user=models.User.objects.get(id=teacher.user_id)
user.delete()
teacher.delete()

        return redirect('admin-view-teacher')


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def update_teacher_view(request,pk):
    teacher=models.TeacherExtra.objects.get(id=pk)
    user=models.User.objects.get(id=teacher.user_id)

    form1=forms.TeacherUserForm(instance=user)
    form2=forms.TeacherExtraForm(instance=teacher)
mydict={'form1':form1,'form2':form2}

    if request.method=='POST':
        form1=forms.TeacherUserForm(request.POST,instance=user)
        form2=forms.TeacherExtraForm(request.POST,instance=teacher)
        print(form1)
        if form1.is_valid() and form2.is_valid():
            user=form1.save()
user.set_password(user.password)
user.save()

            f2=form2.save(commit=False)
            f2.status=True
            f2.save()

            return redirect('admin-view-teacher')

return render(request,'school/admin_update_teacher.html',context=mydict)

```



```

        f2.status=True

        f2.save()

my_student_group = Group.objects.get_or_create(name='STUDENT')
my_student_group[0].user_set.add(user)
    else:
print("form is invalid")
        return HttpResponseRedirect('admin-student')
    return render(request,'school/admin_add_student.html',context=mydict)

@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_view_student_view(request):
    students=models.StudentExtra.objects.all().filter(status=True)
    return render(request,'school/admin_view_student.html',{'students':students})

@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def delete_student_from_school_view(request,pk):
    student=models.StudentExtra.objects.get(id=pk)
    user=models.User.objects.get(id=student.user_id)
    user.delete()
    student.delete()
    return redirect('admin-view-student')
@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def delete_student_view(request,pk):
    student=models.StudentExtra.objects.get(id=pk)
    user=models.User.objects.get(id=student.user_id)
    user.delete()

```

```

student.delete()

    return redirect('admin-approve-student')
@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def update_student_view(request,pk):
    student=models.StudentExtra.objects.get(id=pk)
    user=models.User.objects.get(id=student.user_id)
    form1=forms.StudentUserForm(instance=user)
    form2=forms.StudentExtraForm(instance=student)
mydict={'form1':form1,'form2':form2}

    if request.method=='POST':
        form1=forms.StudentUserForm(request.POST,instance=user)
        form2=forms.StudentExtraForm(request.POST,instance=student)
        print(form1)
        if form1.is_valid() and form2.is_valid():
            user=form1.save()
user.set_password(user.password)
user.save()

            f2=form2.save(commit=False)
            f2.status=True
            f2.save()

            return redirect('admin-view-student')

    return render(request,'school/admin_update_student.html',context=mydict)


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_approve_student_view(request):
    students=models.StudentExtra.objects.all().filter(status=False)

    return render(request,'school/admin_approve_student.html',{'students':students})

@login_required(login_url='adminlogin')
@user_passes_test(is_admin)

```

```
def approve_student_view(request,pk):
    students=models.StudentExtra.objects.get(id=pk)
students.status=True
students.save()

    return redirect(reverse('admin-approve-student'))


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_view_student_fee_view(request):
    students=models.StudentExtra.objects.all()

    return render(request,'school/admin_view_student_fee.html',{'students':students})

#attendance related
viewwwwww(
by sumit)
@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_attendance_view(request):

    return render(request,'school/admin_attendance.html')




@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_take_attendance_view(request,cl):
    students=models.StudentExtra.objects.all().filter(cl=cl)

print(students)
aform=forms.AttendanceForm()

if request.method=='POST':
    form=forms.AttendanceForm(request.POST)

    if form.is_valid():
        Attendances=request.POST.getlist('present_status')

        date=form.cleaned_data['date']
```



```

        for i in range(len(Attendances)):
AttendanceModel=models.Attendance()

        AttendanceModel.cl=cl

AttendanceModel.date=date
AttendanceModel.present_status=Attendances[i]
AttendanceModel.roll=students[i].roll
AttendanceModel.save()

        return redirect('admin-attendance')

    else:
print('form invalid')

    return
render(request,'school/admin_take_attendance.html',{'students':students,'aform':aform})


@login_required(login_url='adminlogin')
@user_passes_test(is_admin)
def admin_view_attendance_view(request,cl):
    form=forms.AskDateForm()
    if request.method=='POST':
        form=forms.AskDateForm(request.POST)
        if form.is_valid():
            date=form.cleaned_data['date']
            attendancedata=models.Attendance.objects.all().filter(date=date,cl=cl)
            studentdata=models.StudentExtra.objects.all().filter(cl=cl)
            mylist=zip(attendancedata,studentdata)

            return
            render(request,'school/admin_view_attendance_page.html',{'cl':cl,'mylist':mylist,'date':date})
        else:
print('form invalid')

    return render(request,'school/admin_view_attendance_ask_date.html',{'cl':cl,'form':form})

```

[illegible]

```
@login_required(login_url='adminlogin')
```

```
@user_passes_test(is_admin)
```

```
def admin_fee_view(request):
```

```
return render(request, 'school/admin_fee.html')
```

```
@login_required(login_url='adminlogin')
```

```
@user_passes_test(is_admin)
```

```
def admin_view_fee_view(request,cl):
```

```
feedetails=models.StudentExtra.objects.all().filter(cl=cl)
```

```
return render(request,'school/admin_view_fee.html',{'feedetails':feedetails,'cl':cl})
```

#notice related

views(b
y sumit)

```
@login_required(login_url='adminlogin')
```

```
@user_passes_test(is_admin)
```

```
def admin_notice_view(request):
```

```
form=forms.NoticeForm()
```

```
if request.method=='POST':
```

```
form=forms.NoticeForm(request.POST)
```

```
if form.is_valid():
```

```
form=form.save(commit=False)
```

```
form.by=request.user.first_name
```

```
form.save()
```

```
return redirect('admin-dashboard')
```

```
return render(request,'school/admin_notice.html',{'form':form})
```

#for TEACHER LOGIN

[illegible]

```

@login_required(login_url='teacherlogin')
@user_passes_test(is_teacher)
def teacher_dashboard_view(request):
    teacherdata=models.TeacherExtra.objects.all().filter(status=True,user_id=request.user.id)
    notice=models.Notice.objects.all()
    mydict={
        'salary':teacherdata[0].salary,
        'mobile':teacherdata[0].mobile,
        'date':teacherdata[0].joindate,
        'notice':notice
    }
    return render(request,'school/teacher_dashboard.html',context=mydict)
@login_required(login_url='teacherlogin')
@user_passes_test(is_teacher)
def teacher_attendance_view(request):
    return render(request,'school/teacher_attendance.html')
@login_required(login_url='teacherlogin')
@user_passes_test(is_teacher)
def teacher_take_attendance_view(request,cl):
    students=models.StudentExtra.objects.all().filter(cl=cl)
    aform=forms.AttendanceForm()
    if request.method=='POST':
        form=forms.AttendanceForm(request.POST)
        if form.is_valid():
            Attendances=request.POST.getlist('present_status')
            date=form.cleaned_data['date']
            for i in range(len(Attendances)):
AttendanceModel=models.Attendance()
                AttendanceModel.cl=cl
AttendanceModel.date=date
AttendanceModel.present_status=Attendances[i]

```

```

AttendanceModel.roll=students[i].roll
AttendanceModel.save()

        return redirect('teacher-attendance')

    else:
print('form invalid')

    return
render(request,'school/teacher_take_attendance.html',{'students':students,'aform':aform})


@login_required(login_url='teacherlogin')
@user_passes_test(is_teacher)
def teacher_view_attendance_view(request,cl):
    form=forms.AskDateForm()
    if request.method=='POST':
        form=forms.AskDateForm(request.POST)
        if form.is_valid():
            date=form.cleaned_data['date']
            attendancedata=models.Attendance.objects.all().filter(date=date,cl=cl)
studentdata=models.StudentExtra.objects.all().filter(cl=cl)
mylist=zip(attendancedata,studentdata)

            return
render(request,'school/teacher_view_attendance_page.html',{'cl':cl,'mylist':mylist,'date':date})
        else:
print('form invalid')

        return render(request,'school/teacher_view_attendance_ask_date.html',{'cl':cl,'form':form})
@login_required(login_url='teacherlogin')
@user_passes_test(is_teacher)
def teacher_notice_view(request):
    form=forms.NoticeForm()
    if request.method=='POST':
        form=forms.NoticeForm(request.POST)
        if form.is_valid():

```



```
studentdata=models.StudentExtra.objects.all().filter(user_id=request.user.id,status=True)

attendancedata=models.Attendance.objects.all().filter(date=date,c1=studentdata[0].c1,roll=stu
dentdata[0].roll)

mylist=zip(attendancedata,studentdata)

        return
render(request,'school/student_view_attendance_page.html',{'mylist':mylist,'date':date})

    else:

print('form invalid')


    return render(request,'school/student_view_attendance_ask_date.html',{'form':form})



# foraboutus and contact ussssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
(by sumit)

def aboutus_view(request):

    return render(request,'school/aboutus.html')




def contactus_view(request):

    sub = forms.ContactusForm()

    if request.method == 'POST':

        sub = forms.ContactusForm(request.POST)

        if sub.is_valid():

            email = sub.cleaned_data['Email']

            name=sub.cleaned_data['Name']

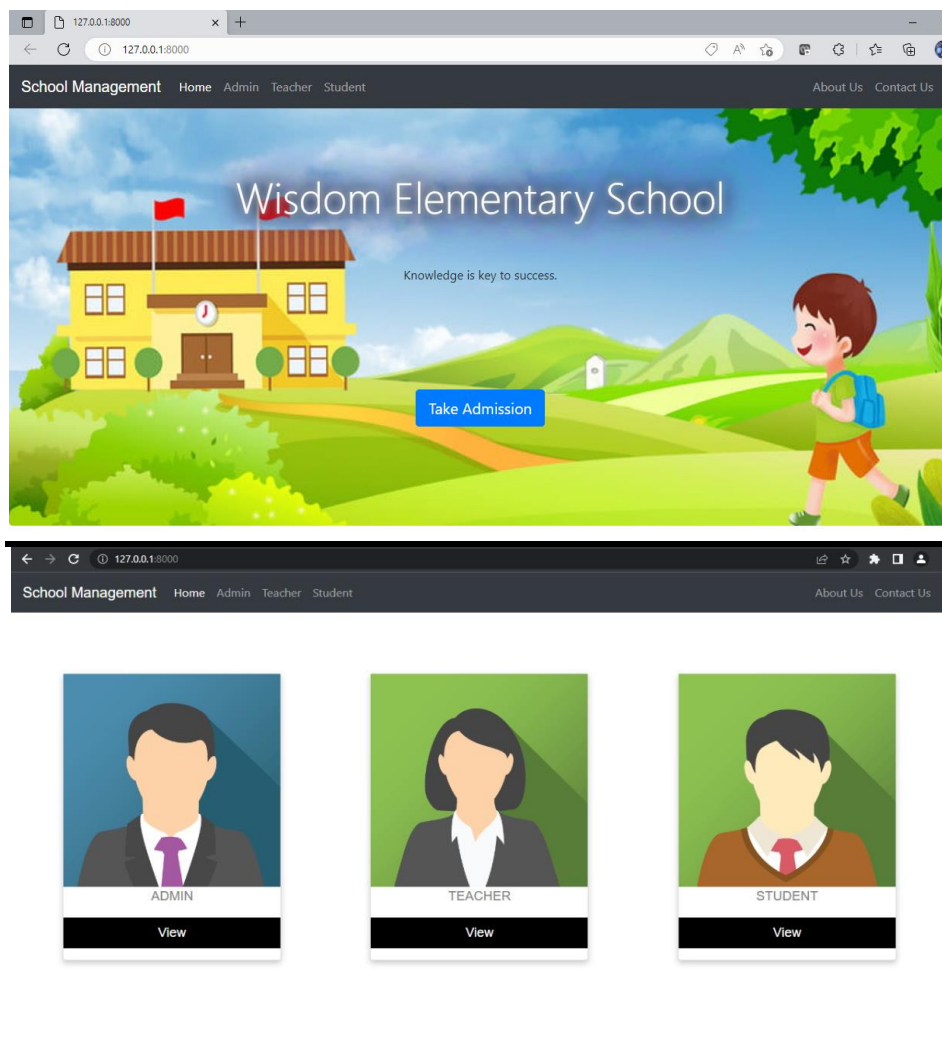
            message = sub.cleaned_data['Message']

send_mail(str(name)+' || '+str(email),message,settings.EMAIL_HOST_USER,
settings.EMAIL_RECEIVING_USER, fail_silently = False)

            return render(request, 'school/contactussuccess.html')

return render(request, 'school/contactus.html', {'form':sub})
```

Sample Outputs:



1

Student-

Student Login to Take Admission-

School Management Home Admin Teacher Student About Us Contact Us

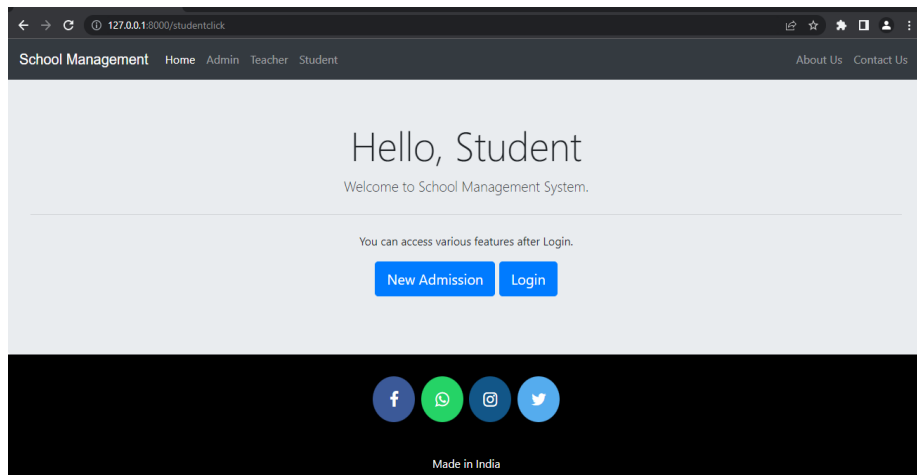
New Admission To School

Fees for
Class 1-5 = 10,000 and Class 6-10 = 15,000

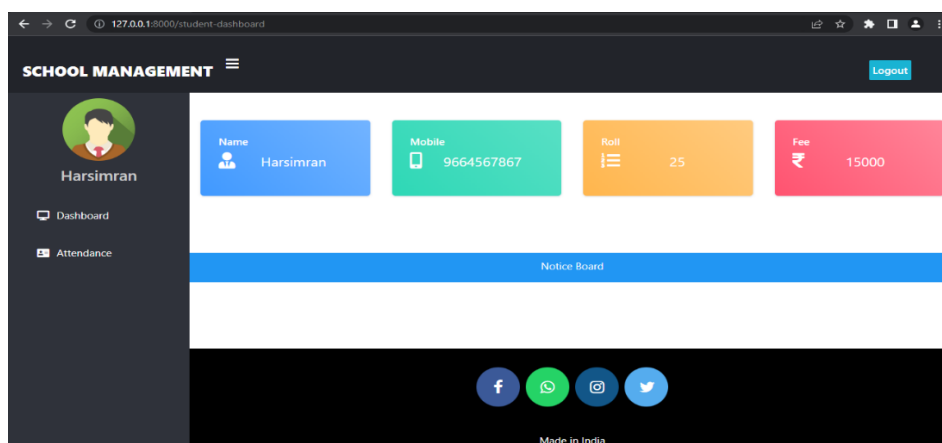
| | |
|------------|--------------|
| First Name | Last Name |
| Username | Password |
| one | Fathers Name |
| Mobile | Fee |

[Take Admission](#)

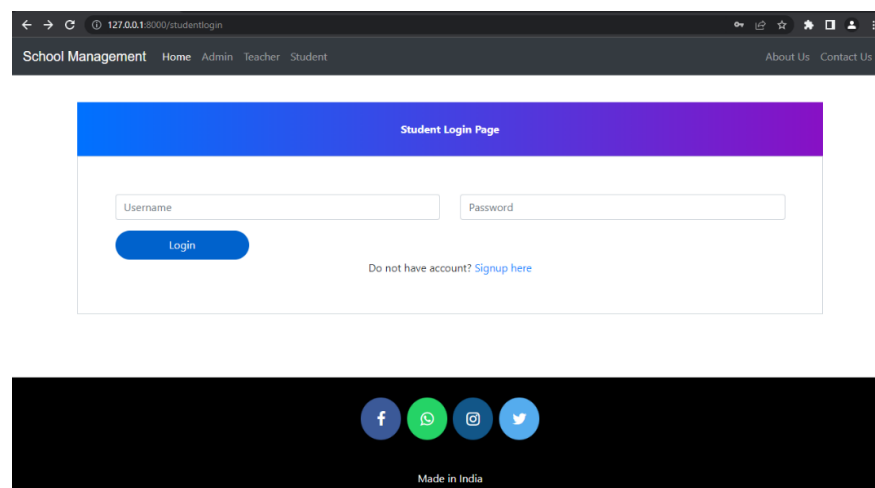
Already have an account? [Login here](#)



Student Account-

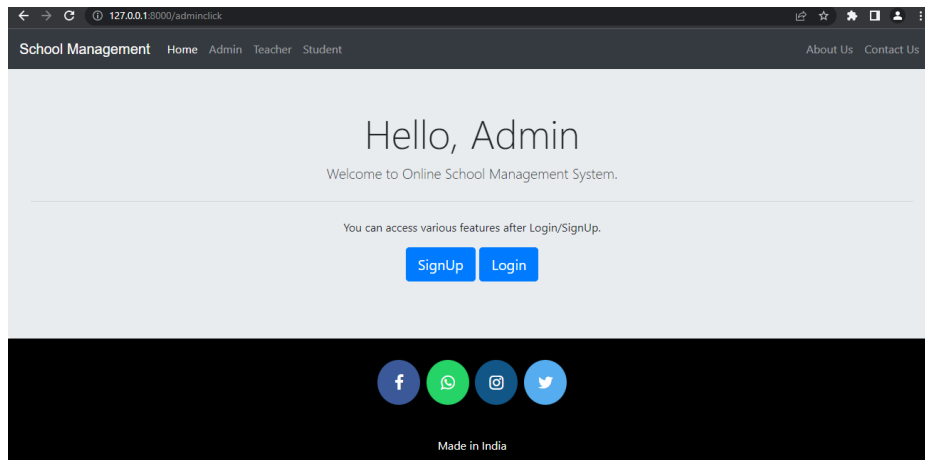


Student Login Page-

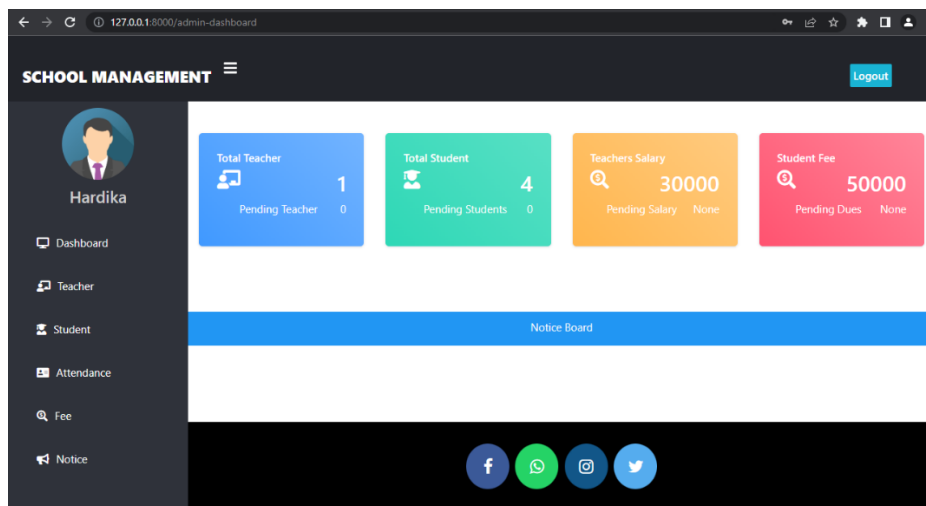


Admin-

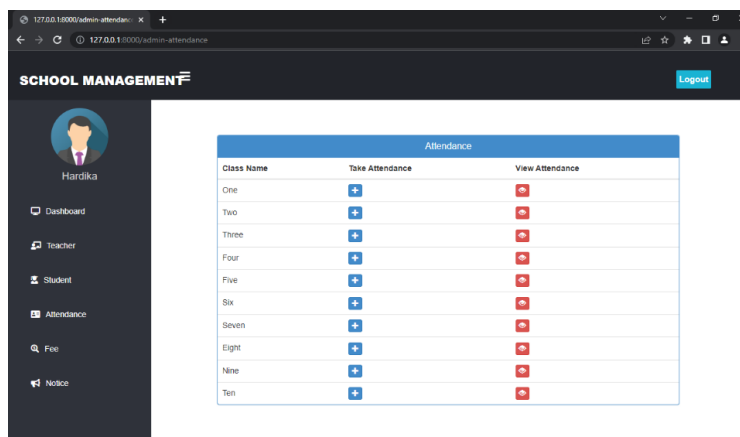
Admin (For verification)-



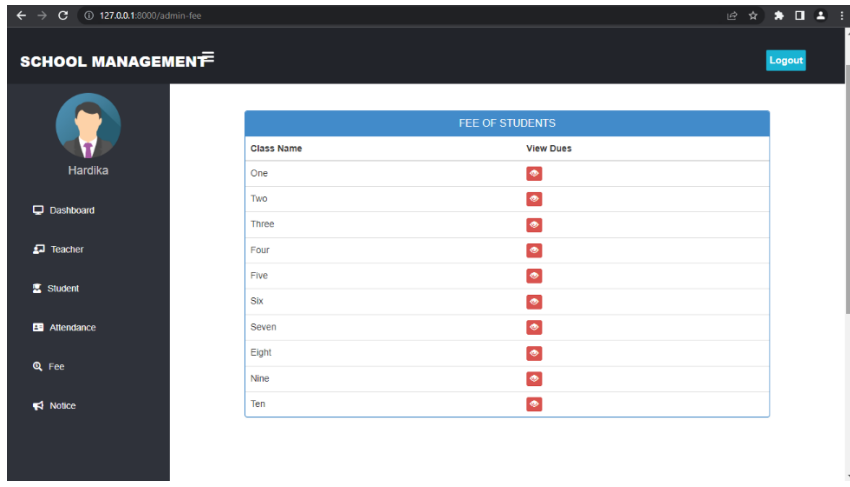
Admin Account-



Admin View Attendance-



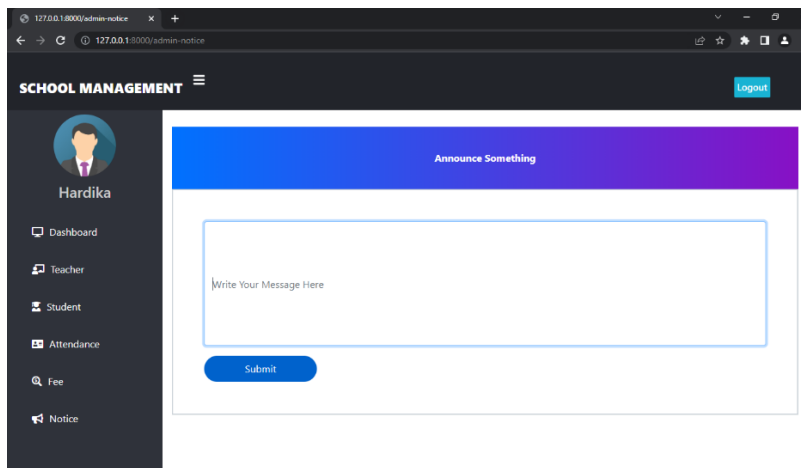
Admin View Student Fee-



The screenshot shows the 'FEE OF STUDENTS' table in the Admin interface. The table has two columns: 'Class Name' and 'View Dues'. The 'View Dues' column contains red icons with a plus sign, indicating that fees are due for each class.

| Class Name | View Dues |
|------------|-----------|
| One | |
| Two | |
| Three | |
| Four | |
| Five | |
| Six | |
| Seven | |
| Eight | |
| Nine | |
| Ten | |

Admin put some Notice-



The screenshot shows the 'Announce Something' form in the Admin interface. The form has a text area labeled 'Write Your Message Here' and a 'Submit' button.

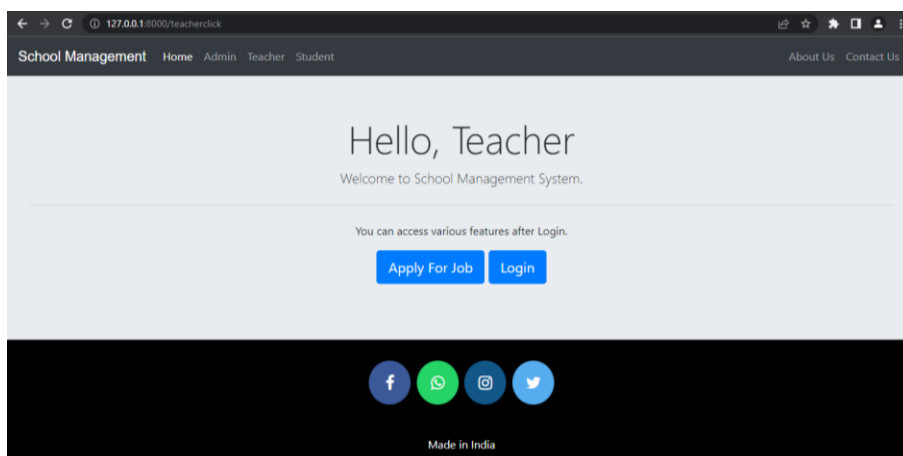
Announce Something

Write Your Message Here

Submit

Teacher

Teacher Apply for Job-



The screenshot shows the Teacher interface. It displays a 'Hello, Teacher' message, a welcome message, and a 'You can access various features after Login.' message. There are two buttons: 'Apply For Job' and 'Login'.

Hello, Teacher

Welcome to School Management System.

You can access various features after Login.

Apply For Job Login

127.0.0.1:8000/teachersignup

School Management Home Admin Teacher Student About Us Contact Us

Apply For Teacher Job

First Name Last Name

Username Password

Mobile Salary


[Apply](#)

Already have an account? [Login here](#)


f WhatsApp Instagram Twitter


127.0.0.1:8000/teacher-dashboard


SCHOOL MANAGEMENT Logout


 Kush

Dashboard Attendance Notice

Name  Kush

Mobile  8756445678

Join Date  Jan. 9, 2023

Salary  ₹ 30000

Notice Board


f WhatsApp Instagram Twitter

Made in India











Teacher can Mark Attendance-

127.0.0.1:8000/teacher-attendance

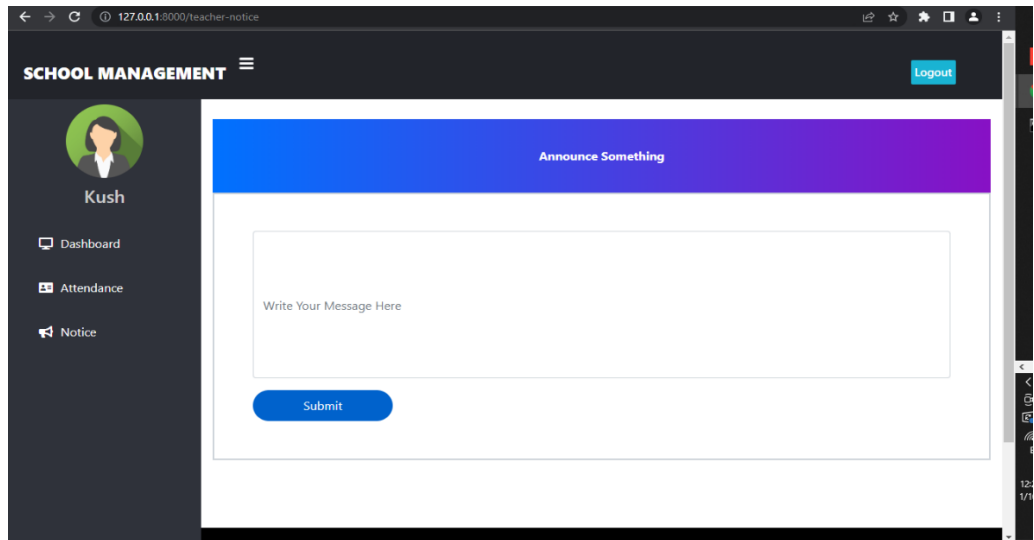
SCHOOL MANAGEMENT Logout

 Kush

Dashboard Attendance Notice

| Attendance | | |
|------------|-------------------|---|
| Class Name | Take Attendance | View Attendance |
| One | + |  |
| Two | + |  |
| Three | + |  |
| Four | + |  |
| Five | + |  |
| Six | + |  |
| Seven | + |  |
| Eight | + |  |
| Nine | + |  |
| Ten | + |  |

Teacher put some Notice-



The screenshot displays a web application titled "SCHOOL MANAGEMENT". The user is logged in as "Kush", indicated by a profile icon and name in the left sidebar. The sidebar also contains links for "Dashboard", "Attendance", and "Notice". The main content area is titled "Announce Something" and features a large text input field with the placeholder "Write Your Message Here". Below the input field is a blue "Submit" button. A "Logout" button is located in the top right corner of the application header. The browser's address bar shows the URL "127.0.0.1:8000/teacher-notice".

Chapter-9: BIBILOGRAPHY-

(1) The Complete Reference Python 3.7

-by Shildit

(2) SQLite3, Black Book

-by Steven Holzner

(2) Understanding SQLite3

– Gruber

(3) [SQL OnLine IDE \(sqliteonline.com\)/](http://sqliteonline.com/)

(4) <http://www.python.org/>

(5) Understanding PHP and CSS with the Online Help

(5) On-line Help of Python.

(6) Various Websites of Discussion Forum and software development activities.

Other than the above-mentioned books, the suggestions and supervision of my teacher and my classmates also helped me to develop this software project.