VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgavi-590 018, Karnataka, India

An Internship Report On

SCHOOL MANAGEMENT SYSTEM

Submitted in Partial Fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted By

SUDEEP JADHAV **1MV22CS411**

Under the Guidance of

Internal Guide:
Ms. Gayithri .N
Associate Professor
Dept. Of CSE, SMVIT

External Guide:
Mr. Shivakumar
Founder and CEO
MIT India

Department of Computer Science Engineering
Sir M Visvesvaraya Institute of Technology
[Affiliated to Visvesvaraya Technological University, Belgaum]
Bengaluru-562157

Sir M Visvesvaraya Institute of Technology

[Affiliated to Visvesvaraya Technological University, Belgaum]
Bangaluru-562157
2023-2024

CERTIFICATE

Certified Internship Project entitled **School Management System** carried out by **Sudeep jadhav** bearing **USN: 1MV22CS411** a bonafide student of **Sir M V Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2023-24. It is certified that all corrections/suggestions indicated for INTERNSHIP havebeen executed under the directions of **Mrs. Savitha P, Mrs. Savitha B. & Ms. Thrisha V.S.** The report has been approved as it satisfies the academic requirements in respect of INTERNSHIP prescribed for the said degree.

Signature of Guide

Mr. …………
Asst. Professor
Dept of CSE
SIR MVIT

Signature of HOD

Dr. T N Anitha
HOD and Professor
Dept of CSE
SIR MVIT

Signature of Principal

Dr. Rakesh S G
Principal
SIR MVIT

MIT
INDIA
It and management consulting
Address: Opp Govt ITI College, MSK Mill Road, Kalaburagi
Phone No.: 91 9880531905 / 7019117403 web: cs.mitindia.in

Ref No: MIT/2023-2024/14 **Date: 23/11/2023**

TO WHOM SOEVER IT MAY CONCERN

This is to certify that **Mr. Sudeep Jadhav** bearing **USN: 1MV22CS411**, a student of Bachelor of Engineering (B.E.) (Computer Science and Engineering) of **Sir M vishweshriah Institute of Technology, Bengaluru** has successfully completed 04 (four) weeks (from 25th Oct 2023 to 23rd Nov 2023) internship programme in our organization. During the period of his internship programme with us he was found punctual, hardworking and inquisitive.

We wish all the best in his future endeavors.

for MIT India

MIT India
IT and Management
Consulting

Authorized signature

DECLARATION

I, **Sudeep jadhav** bearing the USN **1MV22CS411**, student of 6th Semester B.E. Department of Computer Science & Engineering, **Sir M Visvesvaraya Institute of Technology**, Bengaluru declare that the INTERNSHIP work entitled "*SCHOOL MANAGEMENT SYSTEM*", has been duly executed by me under the guidance of Mr.Shivakumar and **Ms. Gayithri .N**, Asst Professor, Department of Computer Science Engineering, Sir. MVIT, Bengaluru. The INTERNSHIP report of the same is submitted in partial fulfilment of the requirement for the award of Bachelor of Engineering degree in Department of Computer Science & Engineering by Visvesvaraya Technological University, Belgaum during the year 2023-2024.

Date: Sudeep jadhav
Place: Bengaluru 1MV22CS411

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the INTERNSHIP report which would be complete only with the mention of the almighty God and the people who made it possible, whose report rewarded the effort with success of project presentation.

We are grateful to **Sir M Visvesvaraya Institute of Technology** for providing us an opportunity to enhance our knowledge through the project.

We express our sincere thanks to **Dr. Rakesh S G**, Principal, Sir MVIT, for providing us an opportunity and means to present the project.

We express our heart full thanks to **Dr. T N Anitha**, Professor and Head, Department of Computer Science & Engineering, SIR MVIT, for encouragement in our Internship work, whose cooperation and guidance helped in nurturing this project report.

We would like to express profound thanks to the INTERNSHIP guide **Ms. Gayithri .N, Asst Professor**, Internship coordinator, Department of Computer Science Engineering for the keen interest and encouragement in our Internship presentation.

We extend our thanks to the MIT india Company Guide for their effort and endurance to bring out the best in us.

Finally, we would like to thank our family members and friends for standing with us through all times.

Sudeep Jadhav

1MV22CS411

ABSTRACT

The School Management System, developed using the Django framework, is a robust web-based application designed to enhance the efficiency of school administrative tasks. This system provides a centralized platform where administrators can seamlessly manage critical school operations. Key features include the ability for administrators to add new students and staff members, create and manage timetables, and handle financial transactions related to tuition fees.

Administrators can easily input and update student and staff information, ensuring that all records are accurate and up-to-date. The system also offers tools for creating detailed timetables, allowing for effective scheduling of classes and activities. This ensures optimal utilization of resources and minimizes scheduling conflicts, contributing to a more organized and efficient school environment.

The financial management module is particularly powerful, enabling the generation and management of invoices, tracking tuition fees paid, and monitoring outstanding balances. This feature helps in maintaining transparent and organized financial records, providing parents and guardians with clear and timely information about payment statuses. Automated notifications and reminders can be sent to ensure that fee payments are made on time, reducing the administrative burden on school staff.

Additionally, the system supports comprehensive data analytics and reporting tools, allowing administrators to generate insights into various aspects of school management, such as student performance, staff productivity, and financial health. These insights facilitate informed decision-making and strategic planning. By leveraging the power and scalability of Django, this School Management System ensures high performance, security, and ease of use, ultimately aiming to streamline school administration and enhance the overall educational experience for students and staff.

TABLE OF CONTENTS

LIST OF FIGRUES

CHAPTER 1
INTRODUCTION

During my internship at MIT India in Full Stack Web Development, I immersed myself in a comprehensive learning experience that spanned both front-end and back-end technologies. Under the mentorship of seasoned professionals, I honed my skills in Python programming, particularly within the Django framework, gaining proficiency in developing robust and scalable web applications. On the front-end, I applied HTML, CSS, and JavaScript to create dynamic and responsive user interfaces that enhance user experience.

1.1About company:

MIT India, established in 2008, stands as a leading provider of IT and consulting solutions with a strong focus on skill development training. Led by Mr. Shivakumar, an alumnus of IIM-A, the company leverages the expertise of its technically skilled professionals who bring extensive experience from prestigious organizations. MIT India excels in sectors such as Information Technology, finance, hospitality, tourism, fashion design, and management education. The company's commitment to innovation is reflected in its mission to adopt cutting-edge technologies and its vision to create globally competitive IT and management solutions. Over the years, MIT India has developed over 500 live websites and applications, along with more than 100 custom software applications. They have successfully trained over 5000 students, offering specialized courses that enhance employability and technological proficiency. MIT India's services include custom web development tailored to meet specific client needs, comprehensive e-commerce solutions, mobile app development across platforms, UI/UX design emphasizing user engagement, and integrated digital marketing strategies. Through their pioneering initiatives, MIT India continues to redefine standards in IT education and solutions, driving technological advancements and industry relevance worldwide.

1.2 About internship:

During my internship at MIT India, our primary goal was to develop and enhance web applications that met high standards of functionality, performance, and user experience. My tasks were diverse and aimed at building my expertise in full stack web development. I was responsible for creating responsive and user-friendly front-end interfaces using HTML, CSS, and JavaScript. This involved designing layouts, implementing interactive elements, and ensuring cross-browser compatibility. On the back-end, I worked extensively with Python and the Django framework to develop robust server-side logic, manage databases, and integrate APIs. My role also included optimizing queries and ensuring data integrity using SQL databases like SQLite and PostgreSQL. Additionally, I participated in Agile development processes, contributing to sprint planning, daily stand-ups, and iterative development cycles. Collaborating with cross-functional teams, I learned to balance client requirements with technical feasibility.

CHAPTER 2

TECHNOLOGIES

2.1 Introduction to Full stack Development:

Full stack development refers to the practice of working on both the front-end and back-end of a web application. Full stack developers possess a broad range of skills, allowing them to handle the entire web development process, from designing user interfaces to managing databases and server-side logic. This versatility makes full stack developers valuable assets in the development process, as they can bridge the gap between different aspects of a project and ensure cohesive integration.

Web development also involves other critical aspects such as version control, testing, and deployment. Version control systems like Git help developers track changes and collaborate effectively. Testing ensures that applications function correctly and are free from bugs. Deployment involves making web applications available to users, typically through hosting on web servers.

In summary, web development is a dynamic and multifaceted field that requires a blend of creative and technical skills. It plays a crucial role in the digital age, enabling the creation of websites and applications that serve various purposes, from personal blogs to complex e-commerce platforms and enterprise-level applications. As technology continues to evolve, web development remains at the forefront of innovation, driving the creation of engaging and functional digital experiences.

2.2 Proposed System and its Features

The proposed system is a comprehensive School Management System developed using the Django framework. This system aims to streamline various administrative and academic processes within a school, providing an integrated platform for managing students, staff, timetables, invoices, and tuition fees.

Key Features:

1. **User Management:**
   - **Admin Panel:** The system includes a robust admin panel where administrators can manage the entire system efficiently. This includes adding, updating, and deleting records for students and staff.s

- **Role-Based Access Control:** Different user roles (admin, teacher, student, parent) with specific permissions ensure that users can only access relevant information and perform appropriate actions.

2. **Student and Staff Management:**
   - **Student Enrollment:** Admins can easily add new students, including personal details, academic information, and contact details. They can also update or remove student records as needed.
   - **Staff Management:** Similar functionalities are available for managing staff members, including their roles, departments, and contact information.
   - **Attendance Tracking:** The system allows for tracking and recording attendance for both students and staff, ensuring accurate records.
   - **Leave Management:** Features for managing leave requests and approvals for students and staff.
3. **Timetable Management:**
   - **Class Schedules:** The system allows for the creation and management of class schedules, ensuring that teachers and students have access to up-to-date timetables.
   - **Room Allocation:** Admins can allocate classrooms for different periods and subjects, optimizing the use of school facilities.
   - **Teacher Assignment:** Assign teachers to specific classes and subjects efficiently.
   - **Period Management:** Manage periods and breaks to ensure smooth scheduling.
4. **Fee Management:**
   - **Invoice Generation:** The system can generate invoices for tuition fees and other charges, simplifying the billing process for the school administration.
   - **Payment Tracking:** It provides a detailed overview of each student's fee status, including how much has been paid and how much is due. This helps in maintaining accurate financial records and following up on outstanding payments.
   - **Online Payments:** Integration with online payment gateways allows parents to pay tuition fees conveniently through the system.
   - **Financial Reporting:** Generate detailed financial reports to track income and expenses related to school fees.
5. **Academic Records:**
   - **Grade Management:** Teachers can input and manage students' grades, which are then available for review by students and parents.
   - **Report Cards:** The system can generate report cards and academic transcripts, providing a comprehensive view of students' academic performance over time

CHAPTER 3
SYSTEM REQUIREMENT AND SPECIFICATION

3.1 Hardware requirements :

- Computer with a 1.6 GHz or faster processor
- Windows 7/Win 10 (x86 & x64) - all editions except Starter Edition
- Windows® XP (x86 & x64) with Service Pack 2 or later - all editions except Starter Edition
- 2GB of RAM
- 100 GB or More hard disk space
- 1024 x 768 or higher-resolution display

3.2 Software requirements:

Operating System:

- **Server Side:** Linux (Ubuntu Server, CentOS) or Windows Server (Windows Server 2016 or later)
- **Client Side:** Windows 10, macOS, or Linux for admin, teachers, and staff computers
- **Mobile Devices:** iOS (version 11 or later) or Android (version 8 or later)

Web Server:

- Apache HTTP Server or Nginx for serving web applications

Database:

- SQLite3 (included with Django by default for development purposes)

Backend Framework:

- Django (version 3.x or later)
- Python (version 3.8 or later)

Frontend Technologies:

- HTML5, CSS3 for designing user interfaces
- JavaScript (including frameworks like jQuery) for client-side interactivity
- Bootstrap or other responsive design frameworks for front-end development

CHAPTER 4

SYSTEM ANALYSIS

 Current System Assessment:

- **Functionality Review:** Evaluate existing processes for managing students, staff, timetables, fees, and academic records.
- **Strengths and Weaknesses:** Identify current system strengths (e.g., operational efficiency) and weaknesses (e.g., lack of integration).
- **User Feedback:** Gather feedback from administrators, teachers, parents, and students regarding usability and functionality.

 Requirements Gathering:

- **Stakeholder Interviews:** Conduct interviews with stakeholders to understand their specific needs and expectations from the system.
- **Use Case Development:** Develop use cases to outline system interactions and scenarios, ensuring all functional requirements are captured.
- **Data Collection:** Analyze data requirements for student records, attendance, grades, financial transactions, etc.

 System Design Evaluation:

- **Architecture Review:** Assess the current system architecture and propose enhancements if necessary (e.g., scalability, security).
- **Interface Design:** Review user interface designs to ensure they are intuitive and user-friendly for all stakeholders.
- **Data Flow Diagrams:** Create diagrams to visualize data flow within the system, including inputs, processes, and outputs.

 Feasibility Analysis:

- **Technical Feasibility:** Evaluate the technological capabilities required for implementing proposed system enhancements (e.g., hardware, software).
- **Operational Feasibility:** Assess the feasibility of implementing the system within the organization's operational constraints and budget.

 Security Assessment:

- Conduct a thorough security assessment to identify potential vulnerabilities in data storage, user authentication, and access control mechanisms.
- Implement encryption standards for sensitive data and ensure compliance with data protection regulations (e.g., GDPR, HIPAA).

 Integration Capability:

- Evaluate the system's ability to integrate with existing school systems (e.g., Learning Management Systems, HR systems) and third-party applications (e.g., payment gateways, communication tools).
- Define API specifications and protocols for seamless data exchange between different modules of the school management system.

 Scalability Planning:

- Assess scalability requirements to accommodate future growth in student enrollment, staff size, and data volume.
- Design a scalable architecture that can handle increased workload without compromising performance or user experience.

 Mobile Compatibility:

- Ensure the system's user interface is responsive and optimized for mobile devices, allowing stakeholders to access key functionalities from smartphones and tablets.
- Develop a mobile app or a progressive web app (PWA) to enhance accessibility and usability for on-the-go users.

 Usability Testing and UX Design:

- Conduct usability testing sessions with representative users to gather feedback on interface design, navigation flows, and overall user experience.
- Iterate on UI/UX designs based on feedback to improve usability and accessibility for diverse user groups (administrators, teachers, parents, and students).

- **Economic Feasibility:** Conduct a cost-benefit analysis to determine the economic viability of system improvements.

 Risk Assessment:

- **Risk Identification:** Identify potential risks (e.g., data security breaches, system downtime) associated with system implementation and operation.
- **Risk Mitigation:** Develop strategies to mitigate identified risks through proactive measures and contingency planning.

 Performance Evaluation:

- **Performance Metrics:** Define performance metrics (e.g., response times, system availability) to measure the effectiveness of system improvements.
- **Testing Plan:** Develop a comprehensive testing plan to validate system functionalities and ensure they meet stakeholder requirements.

4.1 SYSTEM ARCHITECTURE

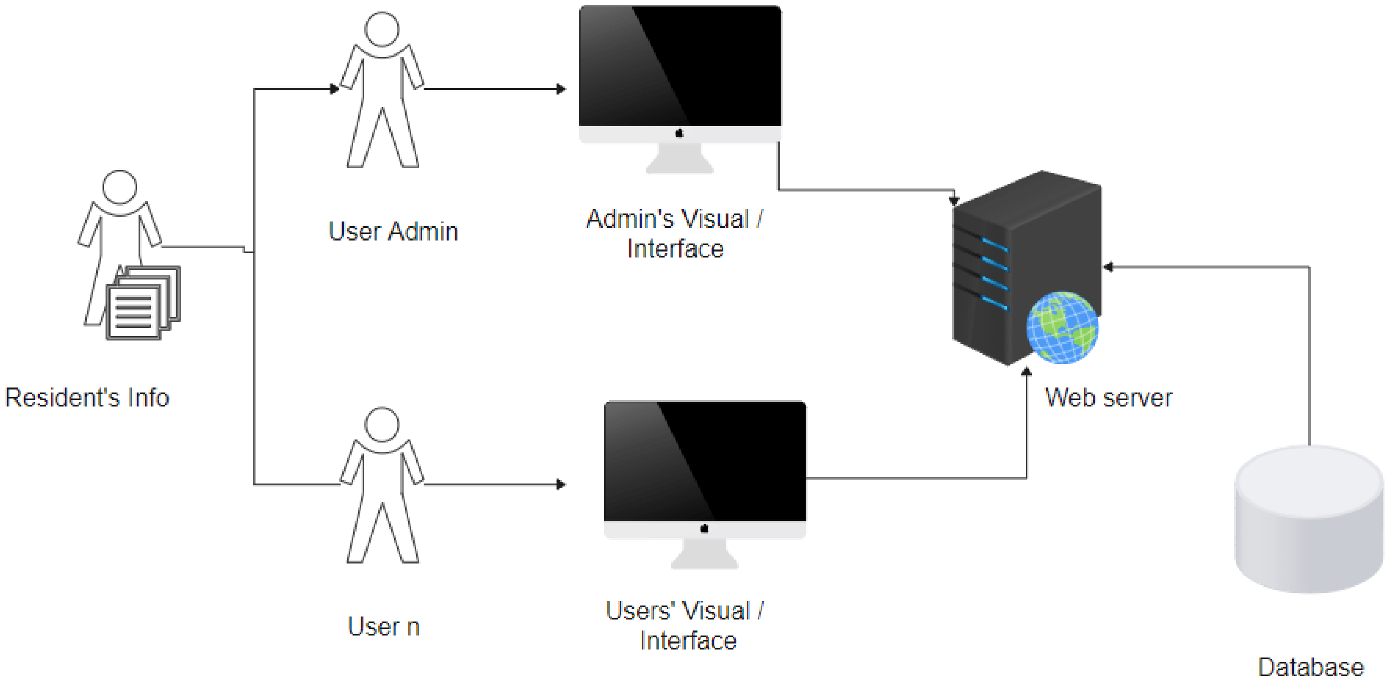


Fig 4.1 : System Architecture

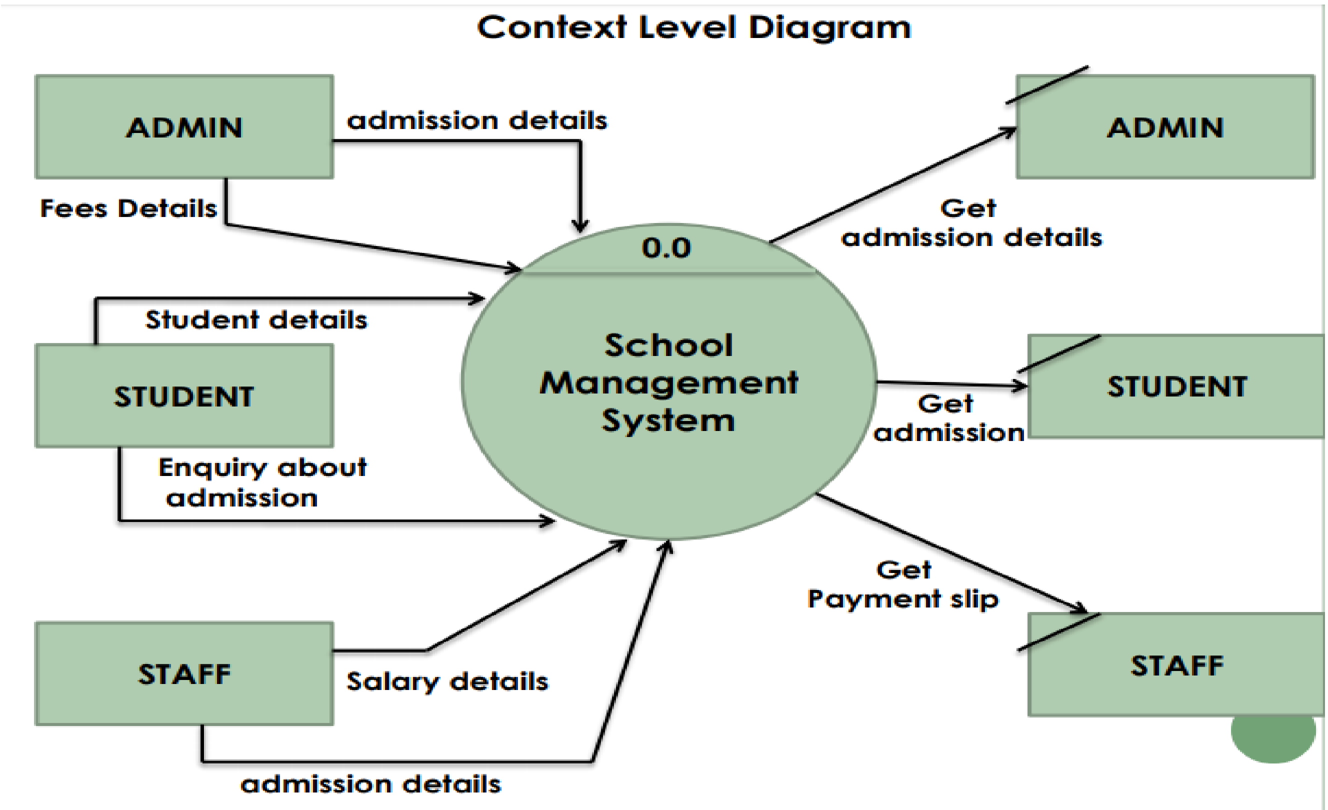


Fig 4.2 : Data flow diagram

The School Management System (SMS) is designed to streamline and manage various administrative and academic operations within a school. This Context Level Diagram represents the high-level interactions between the School Management System and its primary external entities: Admin, Student, and Staff.

Entities and Interactions

1. **Admin**
   - **Inputs to SMS:**
     - Admission Details: Admin provides detailed information about new admissions.
     - Fees Details: Admin inputs and updates information regarding fee structures and payments.
   - **Outputs from SMS:**
     - Get Admission Details: Admin retrieves and reviews admission details from the system.
2. **Student**
   - **Inputs to SMS:**
     - Enquiry about Admission: Students or prospective students submit inquiries regarding the admission process and requirements.
   - **Outputs from SM**

- Get Admission: Students can check their admission status and receive relevant information.

3. **Staff**

   ○ **Inputs to SMS:**

   - Salary Details: Staff members provide or update their salary-related information.
   - Admission Details: Staff might also assist in inputting admission details.

   ○ **Outputs from SMS:**

   - Get Payment Slip: Staff can retrieve their payment slips and salary details from the system.

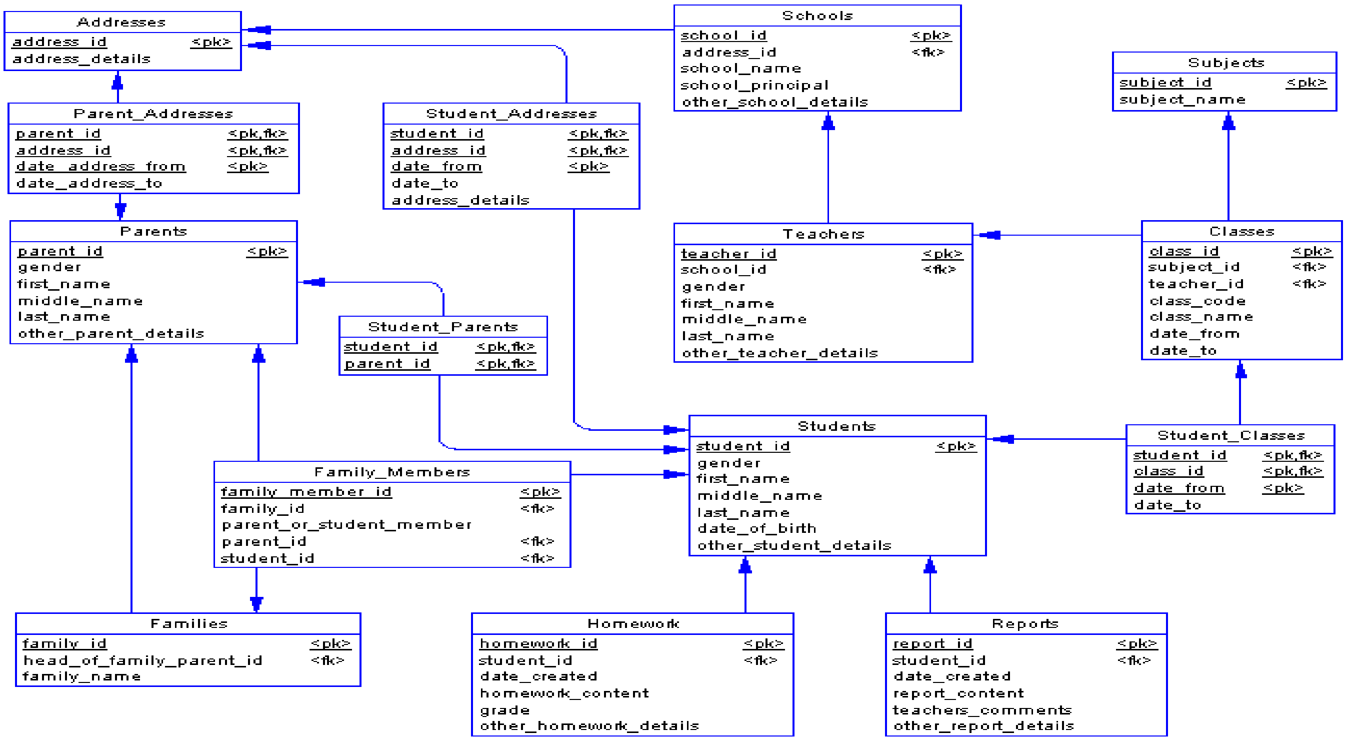


fig 4.3 : ER Diagram

The Physical Data Model for the School Management System encompasses multiple interconnected tables, each representing a distinct aspect of the school's data structure. Key tables include **Students**, **Parents**, **Teachers**, **Classes**, **Subjects**, **Schools**, and **Addresses**. Each of these tables has specific attributes and uses primary and foreign keys to establish relationships with other tables. For example, the **Students** table contains personal details and links to the **Student_Addresses** and **Student_Parents** tables, which store address history and parental information, respectively.

The **Classes** table connects to both the **Teachers** and **Subjects** tables, denoting which teacher instructs which subject in a particular class. Additionally, the **Homework** and **Reports** tables are linked to the **Students** table, allowing tracking of individual student performance and assignments. The **Families** table connects parents and students, encapsulating family structure within the database.

This comprehensive data model ensures that all aspects of the school's administrative and academic operations are thoroughly documented and easily accessible. By maintaining detailed records and facilitating the seamless flow of information, the system supports the efficient management of school operations, enhancing communication and coordination among students, staff, and administrators.

The system analysis for the School Management System involves a comprehensive evaluation of the current administrative processes and the identification of key areas for improvement through automation and integration. This analysis begins with understanding the existing manual processes, which are time-consuming, error-prone, and lack centralization, leading to inefficiencies and data inconsistencies. By mapping out these processes, the analysis identifies critical functions such as student enrollment, staff management, timetable scheduling, fee collection, and academic record maintenance that can be significantly enhanced through a digital solution. It also examines the pain points experienced by different stakeholders, including administrators, teachers, students, and parents, focusing on their specific needs and challenges. The analysis further assesses the technical requirements and feasibility of implementing a Django-based system, ensuring it can handle the institution's growing data and user demands. Additionally, it evaluates the security requirements to protect sensitive information and maintain data integrity. Overall, the system analysis provides a detailed blueprint for developing a robust, scalable, and user-friendly School Management System that streamlines operations, improves data accuracy, and enhances the overall educational experience.

CHAPTER 05
SOURCE CODE

Views.py

```
from django.contrib.messages.views import SuccessMessageMixin
from django.forms import widgets
from django.shortcuts import render
from django.urls import reverse_lazy
from django.views.generic import DetailView, ListView
from django.views.generic.edit import CreateView, DeleteView, UpdateView

from .models import Staff


class StaffListView(ListView):
    model = Staff


class StaffDetailView(DetailView):
    model = Staff
    template_name = "staffs/staff_detail.html"


class StaffCreateView(SuccessMessageMixin, CreateView):
    model = Staff
    fields = "__all__"
    success_message = "New staff successfully added"

    def get_form(self):
        """add date picker in forms"""
        form = super(StaffCreateView, self).get_form()
        form.fields["date_of_birth"].widget = widgets.DateInput(attrs={"type": "date"})
        form.fields["date_of_admission"].widget = widgets.DateInput(
            attrs={"type": "date"}
        )
        form.fields["address"].widget = widgets.Textarea(attrs={"rows": 1})
        form.fields["others"].widget = widgets.Textarea(attrs={"rows": 1})
        return form


class StaffUpdateView(SuccessMessageMixin, UpdateView):
    model = Staff
    fields = "__all__"
    success_message = "Record successfully updated."

    def get_form(self):
        """add date picker in forms"""
        form = super(StaffUpdateView, self).get_form()
        form.fields["date_of_birth"].widget = widgets.DateInput(attrs={"type": "date"})
        form.fields["date_of_admission"].widget = widgets.DateInput(
            attrs={"type": "date"}
        )
        form.fields["address"].widget = widgets.Textarea(attrs={"rows": 1})
        form.fields["others"].widget = widgets.Textarea(attrs={"rows": 1})
        return form


class StaffDeleteView(DeleteView):
    model = Staff
    success_url = reverse_lazy("staff-list")
```

models.py

```
from django.core.validators import RegexValidator
from django.db import models
from django.urls import reverse
from django.utils import timezone


class Staff(models.Model):
    STATUS = [("active", "Active"), ("inactive", "Inactive")]

    GENDER = [("male", "Male"), ("female", "Female")]

    current_status = models.CharField(max_length=10, choices=STATUS, default="active")
    surname = models.CharField(max_length=200)
    firstname = models.CharField(max_length=200)
    other_name = models.CharField(max_length=200, blank=True)
    gender = models.CharField(max_length=10, choices=GENDER, default="male")
    date_of_birth = models.DateField(default=timezone.now)
    date_of_admission = models.DateField(default=timezone.now)

    mobile_num_regex = RegexValidator(
        regex="^[0-9]{10,15}$", message="Entered mobile number isn't in a right format!"
    )
    mobile_number = models.CharField(
        validators=[mobile_num_regex], max_length=13, blank=True
    )

    address = models.TextField(blank=True)
    others = models.TextField(blank=True)

    def __str__(self):
        return f"{self.surname} {self.firstname} {self.other_name}"

    def get_absolute_url(self):
        return reverse("staff-detail", kwargs={"pk": self.pk})
```

urls.py

```
from django.urls import path

from .views import (
    StaffCreateView,
    StaffDeleteView,
    StaffDetailView,
    StaffListView,
    StaffUpdateView,
)

urlpatterns = [
    path("list/", StaffListView.as_view(), name="staff-list"),
    path("<int:pk>/", StaffDetailView.as_view(), name="staff-detail"),
    path("create/", StaffCreateView.as_view(), name="staff-create"),
    path("<int:pk>/update/", StaffUpdateView.as_view(), name="staff-update"),
    path("<int:pk>/delete/", StaffDeleteView.as_view(), name="staff-delete"),
]
```

Apps.py

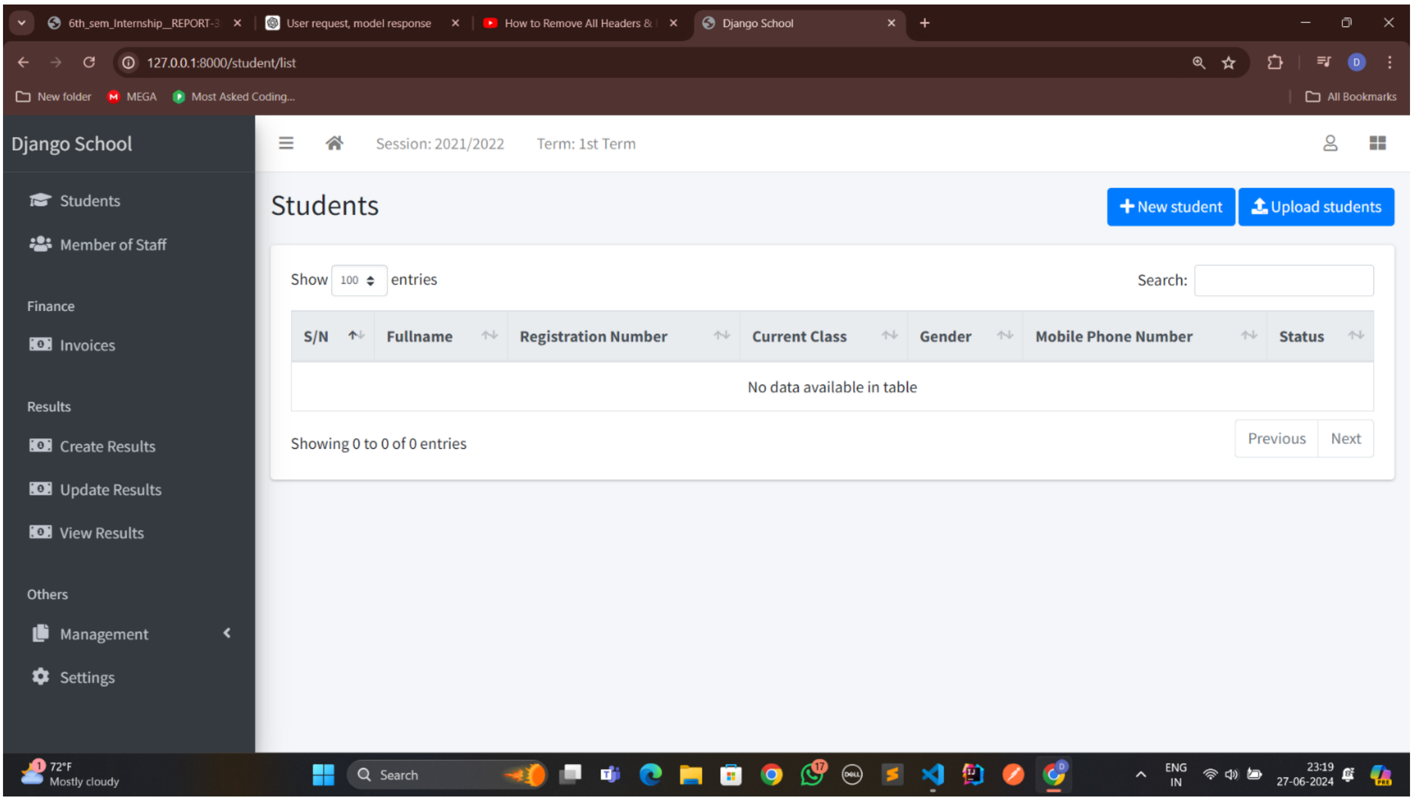
```
from django.apps import AppConfig


class StaffsConfig(AppConfig):
    name = "apps.staffs"
```

asgi.py

```
"""
ASGI config for newapp project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "newapp.settings")

application = get_asgi_application()
```

CHAPTER 6

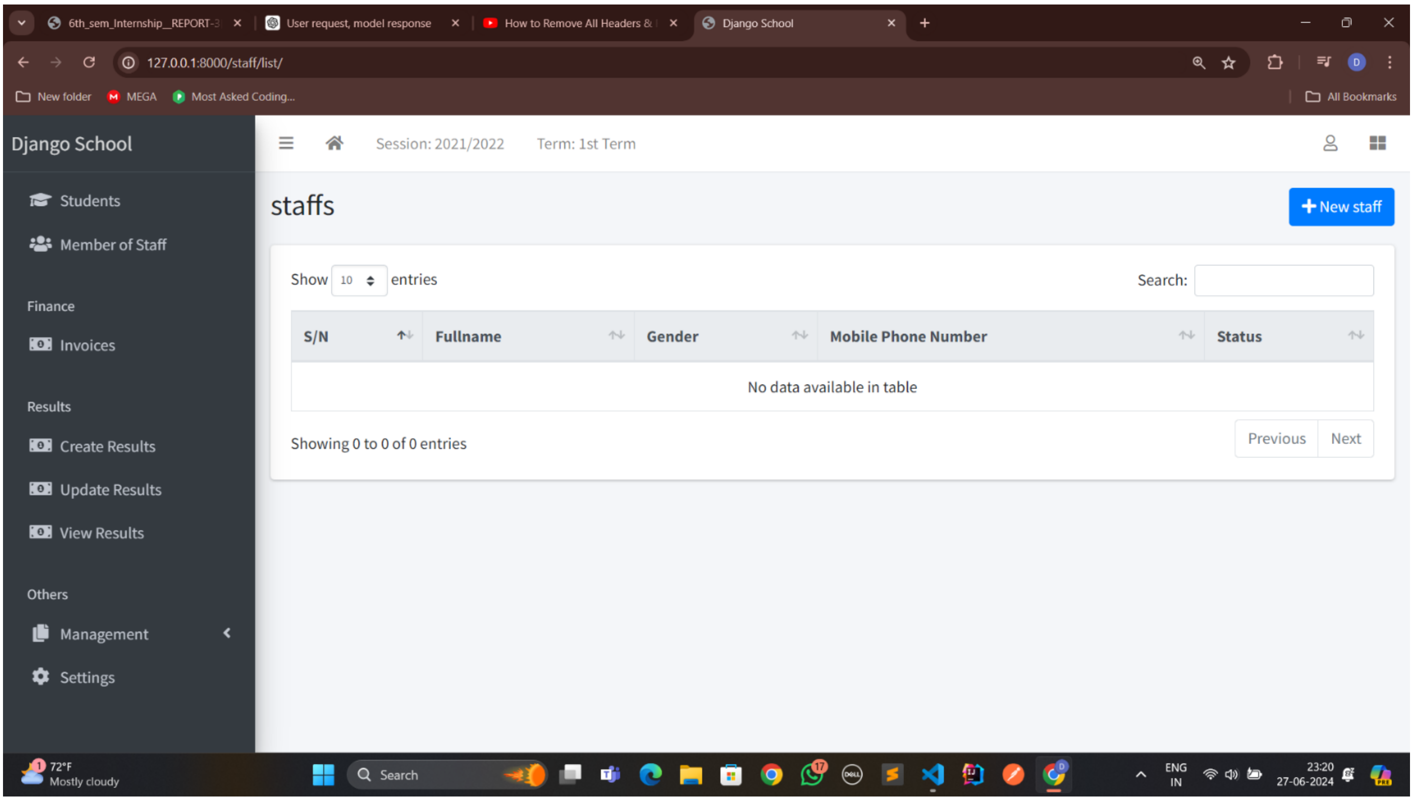
SCREENSHOTS



S/N	Fullname	Registration Number	Current Class	Gender	Mobile Phone Number	Status
No data available in table						

Showing 0 to 0 of 0 entries

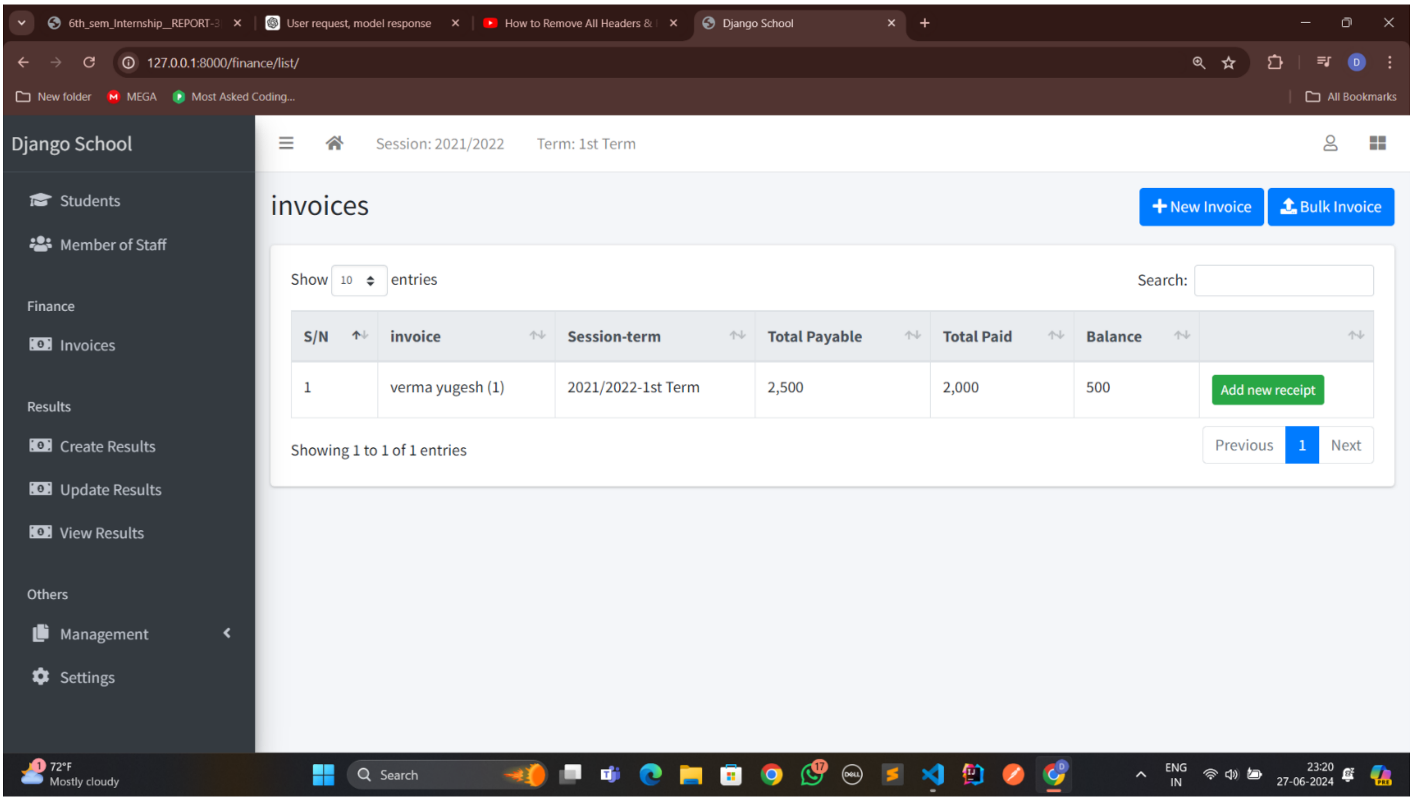
Fig 6.1 : Student details



S/N	Fullname	Gender	Mobile Phone Number	Status
No data available in table				

Showing 0 to 0 of 0 entries

Fig 6.2 : Staff Details



invoices

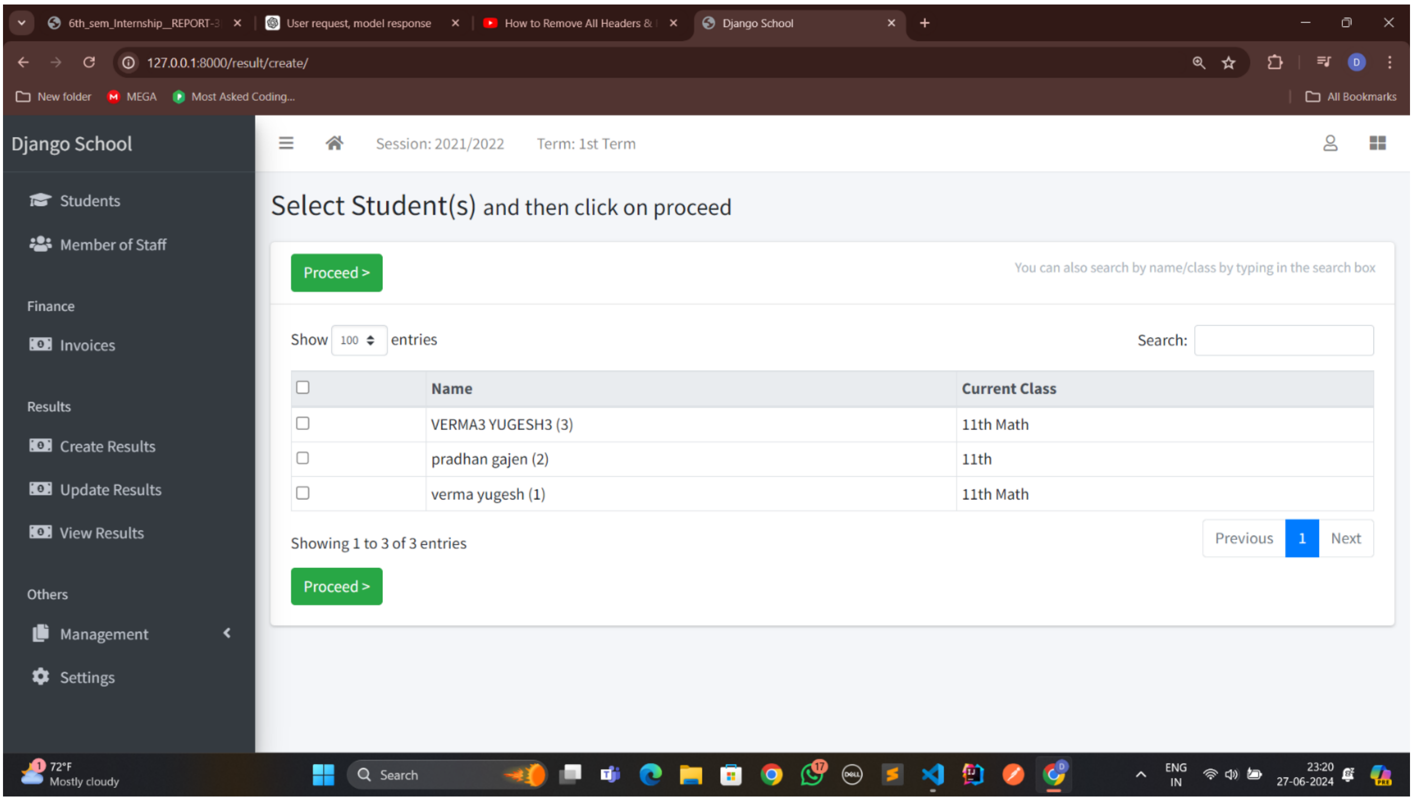
Show 10 entries

Search:

S/N	invoice	Session-term	Total Payable	Total Paid	Balance	
1	verma yugesh (1)	2021/2022-1st Term	2,500	2,000	500	Add new receipt

Showing 1 to 1 of 1 entries

Fig 6.3 : Invoices details



Select Student(s) and then click on proceed

You can also search by name/class by typing in the search box

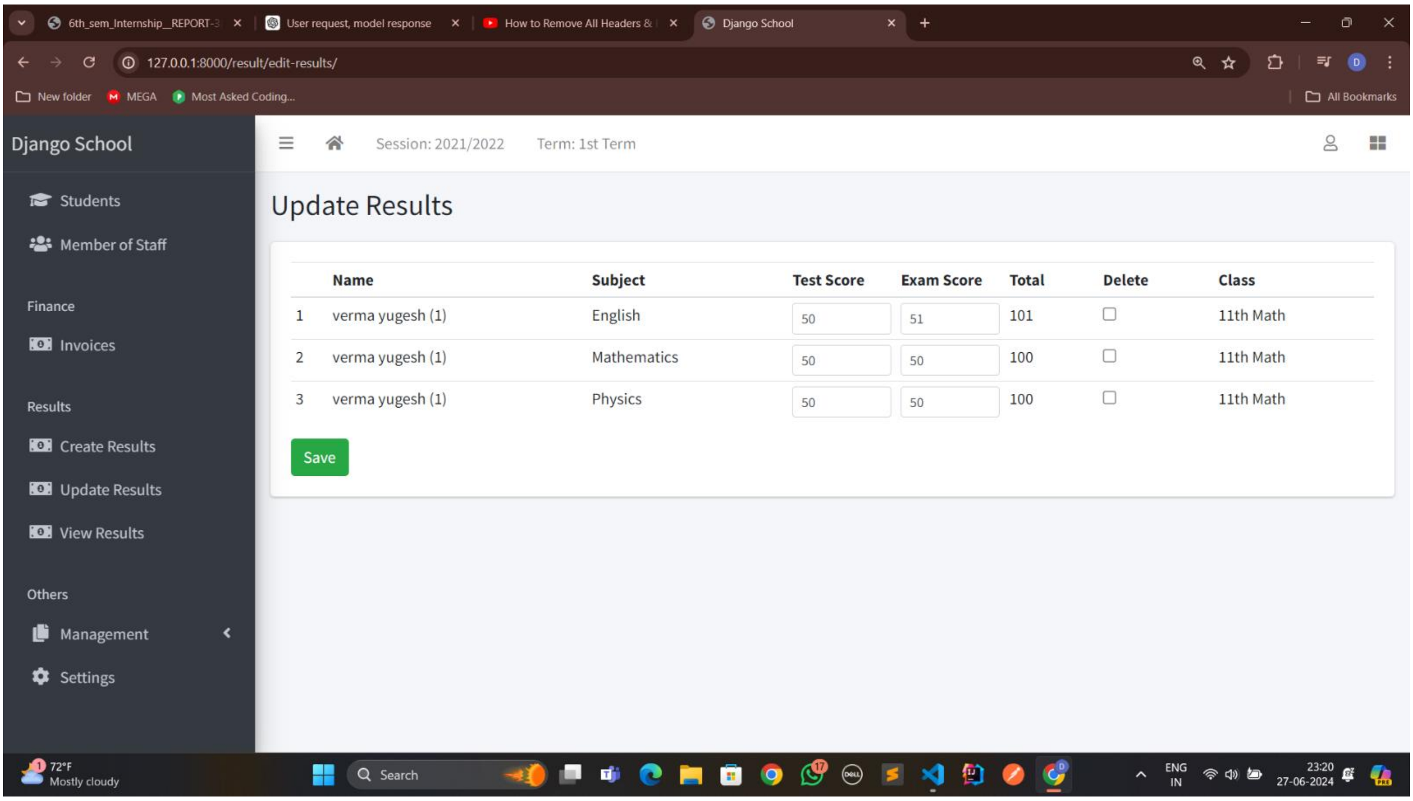
Show 100 entries

Search:

□	Name	Current Class
□	VERMA3 YUGESH3 (3)	11th Math
□	pradhan gajen (2)	11th
□	verma yugesh (1)	11th Math

Showing 1 to 3 of 3 entries

Fig 6.4 : Create Result

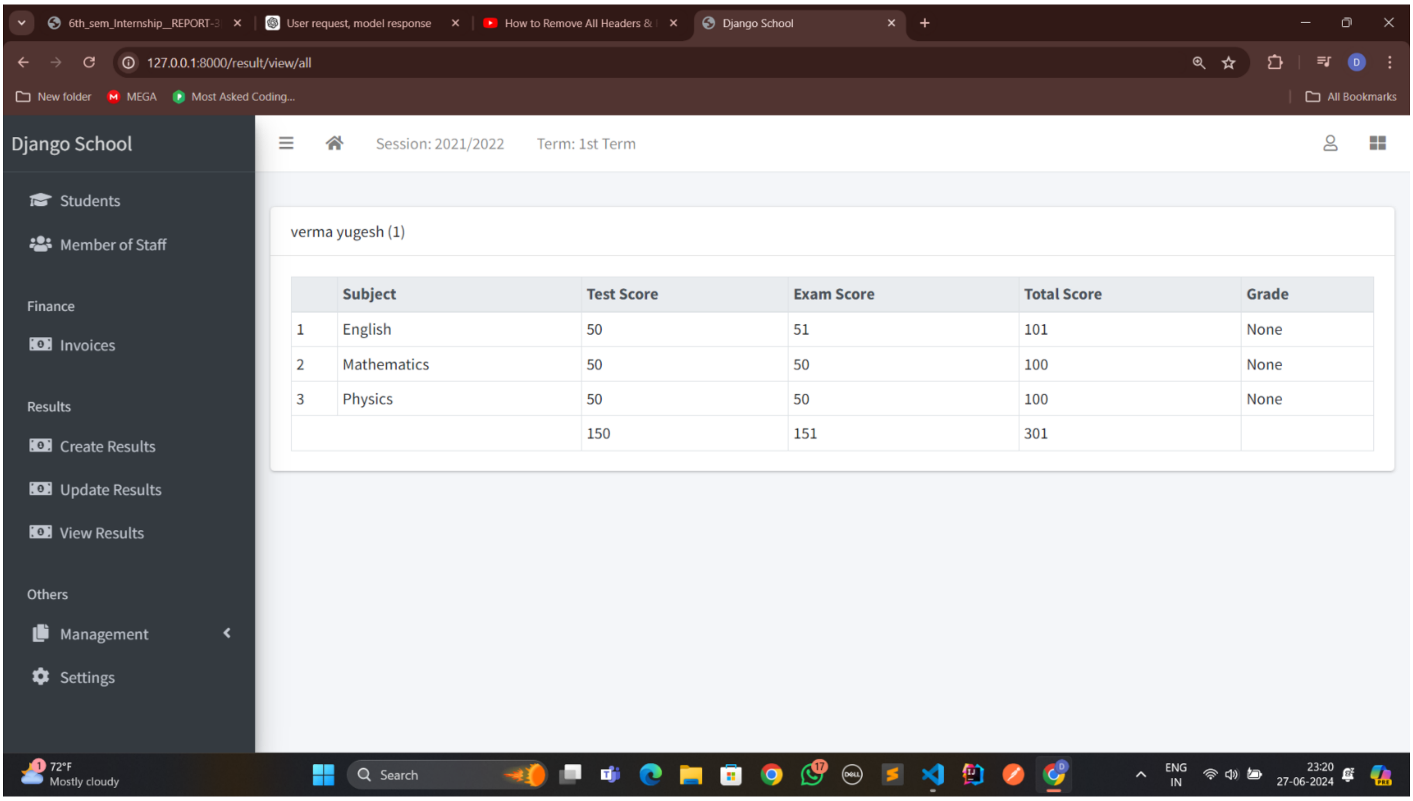


Update Results

	Name	Subject	Test Score	Exam Score	Total	Delete	Class
1	verma yugesh (1)	English	50	51	101	□	11th Math
2	verma yugesh (1)	Mathematics	50	50	100	□	11th Math
3	verma yugesh (1)	Physics	50	50	100	□	11th Math

Save

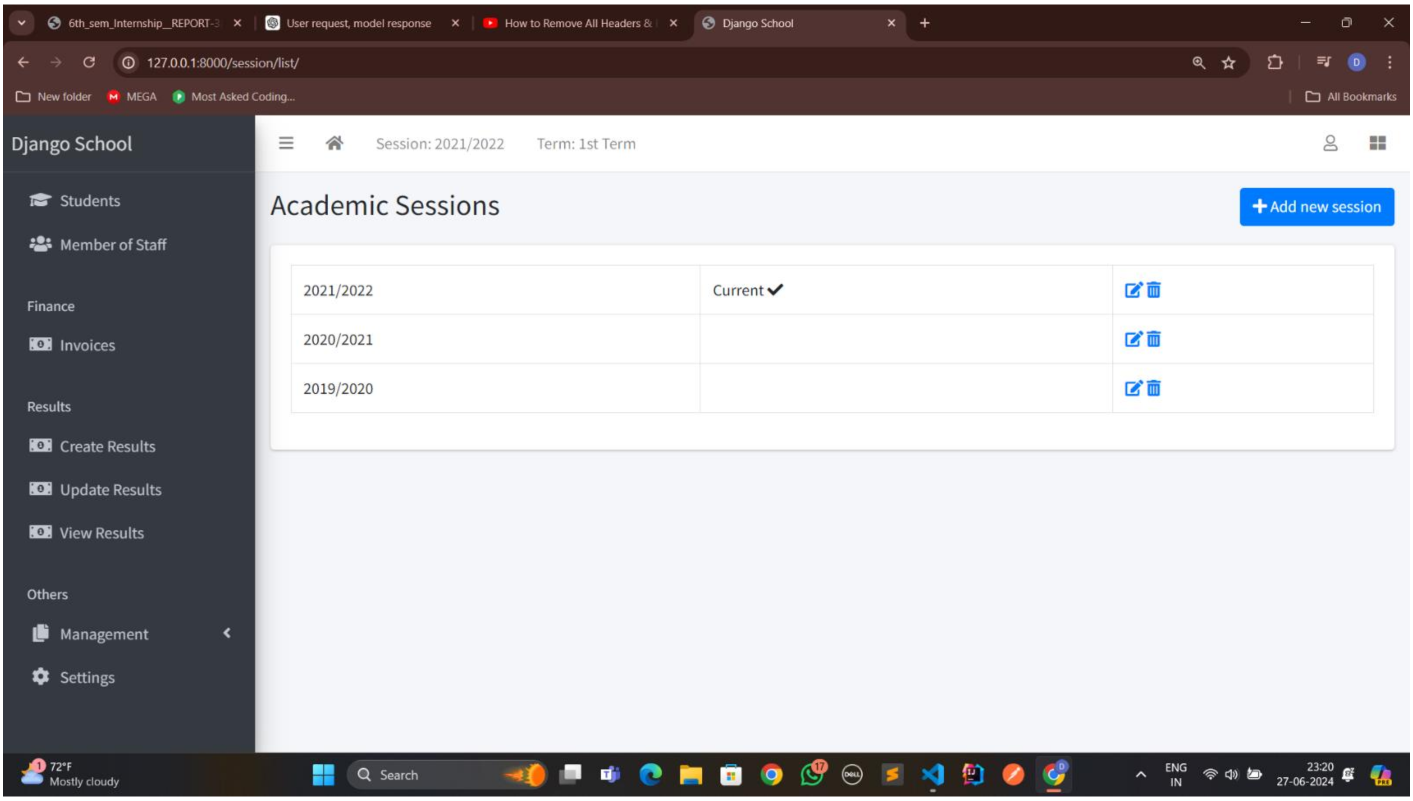
fig 6.5 : Update Result



verma yugesh (1)

	Subject	Test Score	Exam Score	Total Score	Grade
1	English	50	51	101	None
2	Mathematics	50	50	100	None
3	Physics	50	50	100	None
		150	151	301	

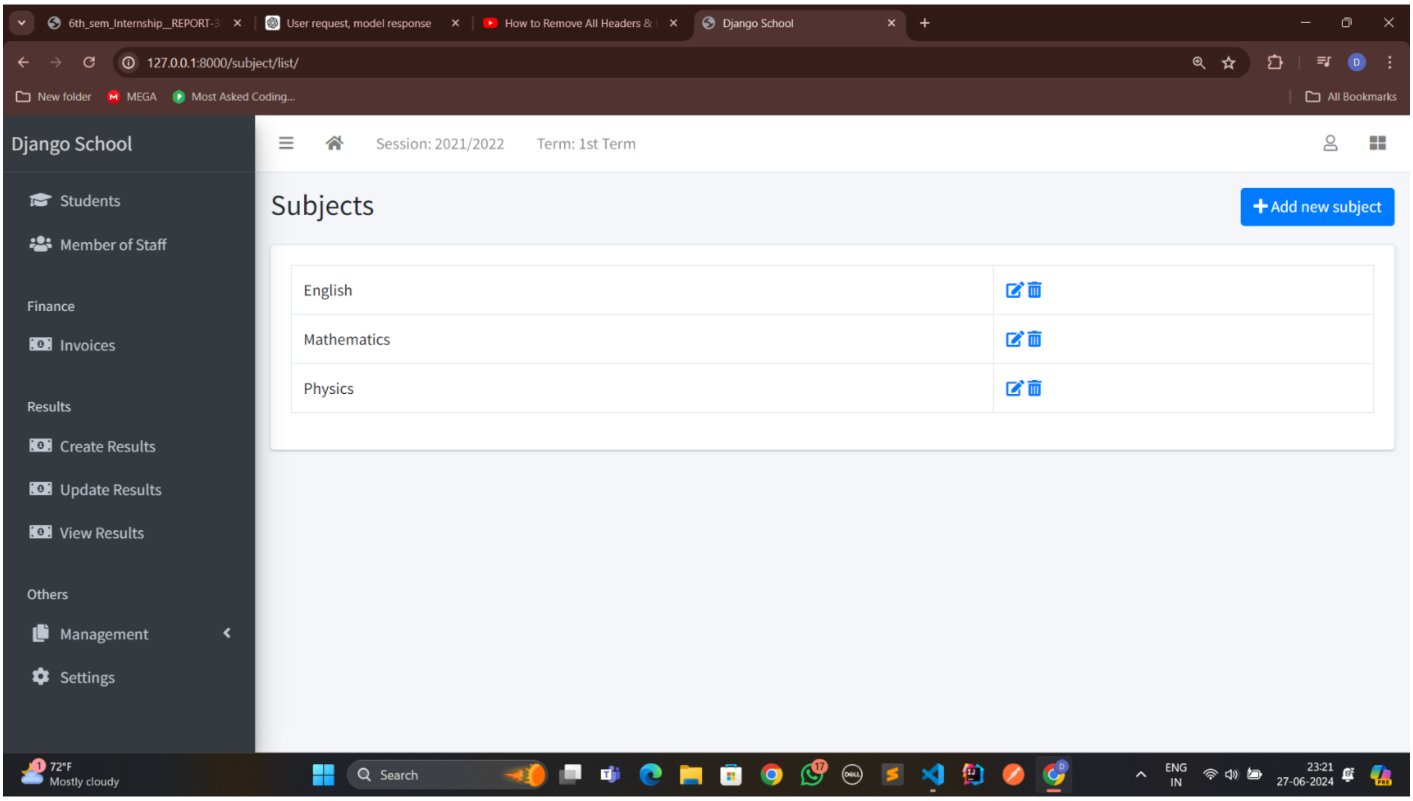
fig 6.6 : View Result



Academic Sessions

2021/2022	Current ✓	
2020/2021		
2019/2020		

fig 6.7 : Academic Sessions



Subjects

English	
Mathematics	
Physics	

fig 6.8 : Subjects

CONCLUSION

In conclusion, the system architecture outlined for the School Management System represents a robust framework designed to meet the diverse needs of educational institutions. By leveraging modern technologies and best practices, including Django for backend development, responsive web interfaces, and secure data management with databases like SQLite3 or PostgreSQL, the architecture ensures scalability, reliability, and security.

The layered approach from client devices to server infrastructure and external integrations facilitates efficient management of student records, academic processes, and administrative tasks. This architecture not only supports seamless interaction among stakeholders—administrators, teachers, students, and parents—but also integrates essential features such as authentication, payment gateways, and communication services for a comprehensive solution.

Overall, this structured system architecture aims to enhance school management efficiency, improve user experience through intuitive interfaces, and maintain data integrity and security, thereby supporting the educational institution in achieving its goals of effective administration and academic excellence.

BIBLIOGRAPHY

1. https://in.search.yahoo.com/
2. https://www.youtube.com/
3. https://codeastro.com/
4. https://projectworlds.in/
5. https://itsourcecode.com/
6. https://in.video.search.yahoo.com/