

A
Project Report On
School MANAGEMENT SYSTEM

Submitted by

Nilesh Mishra

Submitted to

Upskill Campus

MCA (Master of Computer Application)
Second Semester

In

Awadhesh Pratap Singh University Rewa
Madhya Pradesh

2023-2024

Table of Contents

1. Acknowledgements
2. Executive Summary
3. Introduction
4. Objectives of the Internship
5. Profile
6. Project Description
 - Project Overview
 - Project Scope
 - Methodology
7. Work Done
 - Tasks Assigned
 - Technologies Used
 - Challenges Faced

- Solutions

8. Learning Outcomes

9. Conclusion

10. Recommendations

11. References

12. Appendices

1. Acknowledgements

I would like to express my sincere gratitude to APSU Rewa especially my supervisor Ajeet Mishra, for providing me with the opportunity to work on the “School Management System” project. I also extend my thanks to my

academic advisor, Ajeet Mishra, for their guidance and support throughout the internship.

2. Executive Summary

This report presents the work completed during my internship at Upskill campus from 06/05/2024 to 16/06/2024. The focus of my project was to develop a “School Management System” aimed at automating various school administrative tasks. This report outlines the objectives, methodologies, tasks completed,

challenges faced, and the outcomes of the project.

3. Introduction

The internship at Upskill campus provided a valuable opportunity to gain hands-on experience in developing management systems for educational institutions. The duration of the internship was [Duration], during which I was able to apply my academic knowledge to real-world scenarios.

4. Objectives of the Internship

The primary objectives of the internship were:

- To understand the requirements of school management systems.
- To develop and implement a comprehensive system to manage school operations.
- To learn and apply new technologies and tools relevant to school management.

5. Company Profile

Upskill Campus is a leading software development company specializing in educational technology solutions. Established in 2024, the company provides innovative software products to schools and educational institutions worldwide, enhancing their administrative efficiency and educational delivery.

6. Project Description

Project Overview

The “School Management System” project was designed to automate various administrative tasks within a

School, including student enrollment, attendance tracking, grade management, and scheduling. The system aimed to streamline operations and improve communication among staff, students, and parents.

Project Scope

The scope of the project included:

- Requirement analysis and system design.
- Development of modules for student management, attendance, grades, and scheduling.

- Integration with existing school databases.
- Testing and deployment of the system.
- Training school staff to use the new system.

Methodology

The project was executed using the Agile methodology, which allowed for iterative development and continuous feedback. The key phases included:

- Requirement Gathering
- System Design
- Development

- Testing
- Deployment
- User Training

7. Work Done

Tasks Assigned

During the internship, I was responsible for:

- Conducting a needs assessment and gathering requirements.
- Designing the system architecture.
- Coding and developing the application.
- Integrating the system with existing databases.

- Conducting user acceptance testing.
- Creating user manuals and training materials.

Technologies Used

The following technologies and tools were used in the project:

- Programming Languages: Python, JavaScript
- Database: MySQL
- Framework: Django
- Tools: Git, JIRA

Challenges Faced

Some challenges encountered during the project included:

- Ensuring data security and privacy.
- Integrating with legacy systems.
- Managing user roles and permissions effectively.

Solutions Implemented

To overcome these challenges, the following solutions were implemented:

- Implementing robust encryption and authentication mechanisms.
- Conducting thorough testing to ensure integration success.

- Designing a flexible user role management system.

8. Learning Outcomes

Through this internship, I gained:

- Practical experience in software development and project management.
- Knowledge of educational administration processes.
- Skills in problem-solving and troubleshooting.

9. Conclusion

The internship at Upskill campus was an enriching experience that enhanced my technical skills and provided a deeper understanding of school management systems. The “School Management System” project not only met the set objectives but also provided valuable insights for future improvements.

10. Recommendations

For future internships, I recommend:

- More frequent check-ins with supervisors for feedback.
- Additional hands-on workshops on relevant technologies.

- Enhanced documentation of the Existing systems for better understanding.

11. References

- <https://www.tutorialspoint.com/index.htm>
- <https://www.javatpoint.com>
- <https://www.w3schools.com>

Sample Code for “School Management System”

Here is a sample code snippet using Django (Python) for the “School Management System”:

Models (models.py)

```
From django.db import models from  
django.contrib.auth.models
```

```
Import User
```

```
Class Student(models. Model):
```

```
User =
```

```
Models. OneToOneField(User,  
on_delete=models. CASCADE)
```

```
Date_of_birth =
```


Models.DateField()

Address =

models.CharField(max_length=255)

Phone_number =

models.CharField(max_length=15)

Def _str (self): return

self.user.username

Class Teacher(models. Model):

User =

Models. OneToOneField(User,

on_delete=models.CASCADE)

Subject =

models.CharField(max_length=100)

Phone_number =

```
Models.CharField(max_length=15)
Def _str_(self): return self.user.username
Class Class(models.Model):
    Name =
    models.CharField(max_length=100)
    teacher =
    Models. ForeignKey (Teacher, on_delete=
models.CASCADE)
    Def _str_(self): return self.name
```

```
Class Attendance (models. Model):
    Student =
    Models. ForeignKey (Student,
on_delete=models.CASCADE)
    Date = models. DateField()
```

```
Status =  
models.CharField(max_length=10,  
choices=[('Present', 'Present'), ('Absent',  
'Absent')])
```

```
Def _str_(self):
```

```
Return
```

```
Class Grade (models. Model):
```

```
Student = models. ForeignKey (Student,  
on_delete=models.CASCADE)  
class_assigned =
```

```
Models. ForeignKey (Class,  
on_delete=models.CASCADE)
```

```
Grade =  
models.CharField(max_length=2)
```

```
Def _str_(self):
```

```
Return f'{self.student.user.username}  
{self.class_assigned.name} {self.grade}'
```

```
Views (views.py)
```

```
From django. Shortcuts import
```

```
Render, redirect from
```

```
Django.contrib.auth.decorators
```

```
Import login_required from.models  
import Student, Teacher, Class,  
Attendance, Grade from .forms import  
Attendance Form, GradeForm
```

@login_required

Def dashboard(request):

If request.user.is_staff:

Students =

Student.objects.all()

Teachers =

Teacher.objects.all()

Classes =

Class.objects.all()

```
Return render(request, 'dashboard.html',  
{ 'students':
```

```
Students, 'teachers': teachers, 'classes':  
classes})
```

```
Else:
```

```
Student =
```

```
Student.objects.get(user=request.user)
```

```
Attendance =
```

```
Attendance.objects.filter(student=  
student)
```

```
Grades =
```

```
Grade.objects.filter(student=student)
```

```
Return render(request,  
'student_dashboard.html', {'attendance':  
attendance, 'grades': grades})
```

```
@login_required
```

```
Def mark_attendance (request):
```

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

```
Form =
```

```
Attendance Form (request.POST)
```

```
If form.is_valid():
```

```
Form. Save()
```

```
Return
```

```
Redirect('dashboard')
```

```
Else:
```

```
Form Attendance Form() return render  
(request, 'mark_attendance.html',  
{'form': form})
```

```
@login_required
```

```
Def assign_grade (request):
```

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

```
Form =
```

```
GradeForm(request.POST)
```

```
If form.is_valid():
```

```
Form. Save()
```

```
Return
```

```
Redirect('dashboard')
```

```
Else:
```

```
Form GradeForm()
```



```
Return render(request,  
'assign_grade.html', {'form': form})
```

Forms (forms.py)

```
From django import forms from.models  
import Attendance, Grade
```

```
Class
```

```
AttendanceForm(forms. ModelForm):
```

```
Class Meta: model Attendance
```

```
Fields = ['student',
```

```
'date', 'status']
```

```
Class GradeForm(forms. ModelForm):
```

```
Class Meta:
```

```
Model Grade
```

Fields = ['student', 'class_assigned',
'grade']

Templates

- dashboard.html
- student_dashboard.html
- mark_attendance.html
- assign_grade.html

Each template should be designed to display the respective information and forms as required

