K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

STUDENT FEE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

MOHAMED FARMANULLAH B (8115U23ME027)

in partial fulfillment of requirements for the award of the course

MGB1201 - PYTHON PROGRAMMING

in

DEPARTMENT OF MECHANICAL ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER - 2024

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

TRICHY-621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"STUDENT FEE MANAGEMENT SYSTEM"** is the bonafide work of **MOHAMED FARMANULLAH B (8115U23ME027)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE
Dr. T. M. NITHYA, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
ASSOCIATE PROFESSOR
Department of CSE
K.Ramakrishnan College of Engineering (Autonomous)
Samayapuram–621112.

SIGNATURE
Mrs.S.RAJESWARI M.E.
SUPERVISOR
ASSISTANT PROFESSOR
Department of CSE
K.Ramakrishnan College of Engineering (Autonomous)
Samayapuram–621112.

Submitted for the End Semester Examination held on…….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **"STUDENT FEE MANAGEMENT SYSTEM"** the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **MGB1201 – PYTHON PROGRAMMING**

.

Signature

_____

MOHAMED FARMANULLAH B

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Engineering (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs.S.RAJESWARI M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions

MISSION OF THE INSTITUTION

M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies. M3: To provide education for developing high-quality professionals to transform the society.

VISION OF THE DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF THE DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successfulcareer.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

ABSTRACT

A student fee management system in Python is designed to automate the process of fee collection, management, and tracking within educational institutions. This system typically includes functionalities such as student registration, fee payment recording, generating receipts, and tracking outstanding balances. The program allows administrators to efficiently manage fee-related tasks, reducing paperwork and manual errors. It often incorporates a user-friendly interface for both administrators and students, facilitating easy access to fee information and transaction histories. Security measures are also implemented to protect sensitive financial data, ensuring compliance with privacy regulations. Overall, a well-implemented student fee management system in Python enhances efficiency, transparency, and accountability in educational fee management processes. Key features of a Python-based student fee management system include database integration to store student and fee-related data securely. The system enables automated notifications for fee reminders and overdue payments, enhancing communication between the institution and students or parents. It supports various payment methods, such as online payments and bank transfers, making fee transactions convenient and accessible. Additionally, the system generates comprehensive reports and analytics on fee collection trends, outstanding payments, and financial summaries, aiding administrators in decision-making and financial planning

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
A Python-based student fee management system streamlines fee collection, management, and tracking for educational institutions. It features student registration, fee payment recording, receipt generation, and outstanding balance tracking. With database integration, it securely stores data, facilitates notifications for reminders and overdue payments, supports multiple payment methods, and generates detailed reports. This system enhances efficiency, transparency, and decision-making while ensuring data security and compliance.	**PO1** **PO2** **PO3** **PO12**	**PSO1 -3** **PSO2 -3**

Note: 1- Low, 2-Medium, 3- High

SUPERVISOR **HEAD OF THE DEPARTMENT**

TABLE OF CONTENTS

CHAPTER -1

INTRODUCTION

1.1 Objective

To design and implement a Student Fee Management System using Python that simplifies the management of student fees for an educational institution. The system should be user-friendly, efficient, and capable of handling core functionalities like fee collection, tracking payments, generating reports, and managing student data.

1.2 Overview

A Student Fee Management System is a software tool to automate the process of recording, updating, and managing student fee details. This system helps reduce manual workload, minimize errors, and provide transparency in fee-related transactions.

The Student Fee Management System is a software solution designed to automate and simplify the process of managing student fee records in educational institutions. It ensures efficiency, accuracy, and transparency by replacing traditional manual processes with a digital, centralized system. This system aims to handle all aspects of fee management, including student registration, fee structure creation, payment recording, and reporting.

1.3 Python Programming Concepts

1. Object-Oriented Programming (OOP):

Classes and objects to model students, fees, and transactions.

Encapsulation for securing sensitive data like payment records.

2. File Handling:

To store student and fee data persistently in text or CSV files.

Read, write, and update files efficiently.

3. Functions:

Modularize the system by creating reusable functions for different operations (e.g., adding a student, generating reports).

4. Error Handling:

Ensure the system handles invalid inputs or unexpected errors gracefully.

5. Conditional Statements and Loops:

Manage flow control for navigating the menu, handling user input, and processing fee records.

6. Data Structures:

Use dictionaries for quick access to student data.

Lists for maintaining collections like transactions.

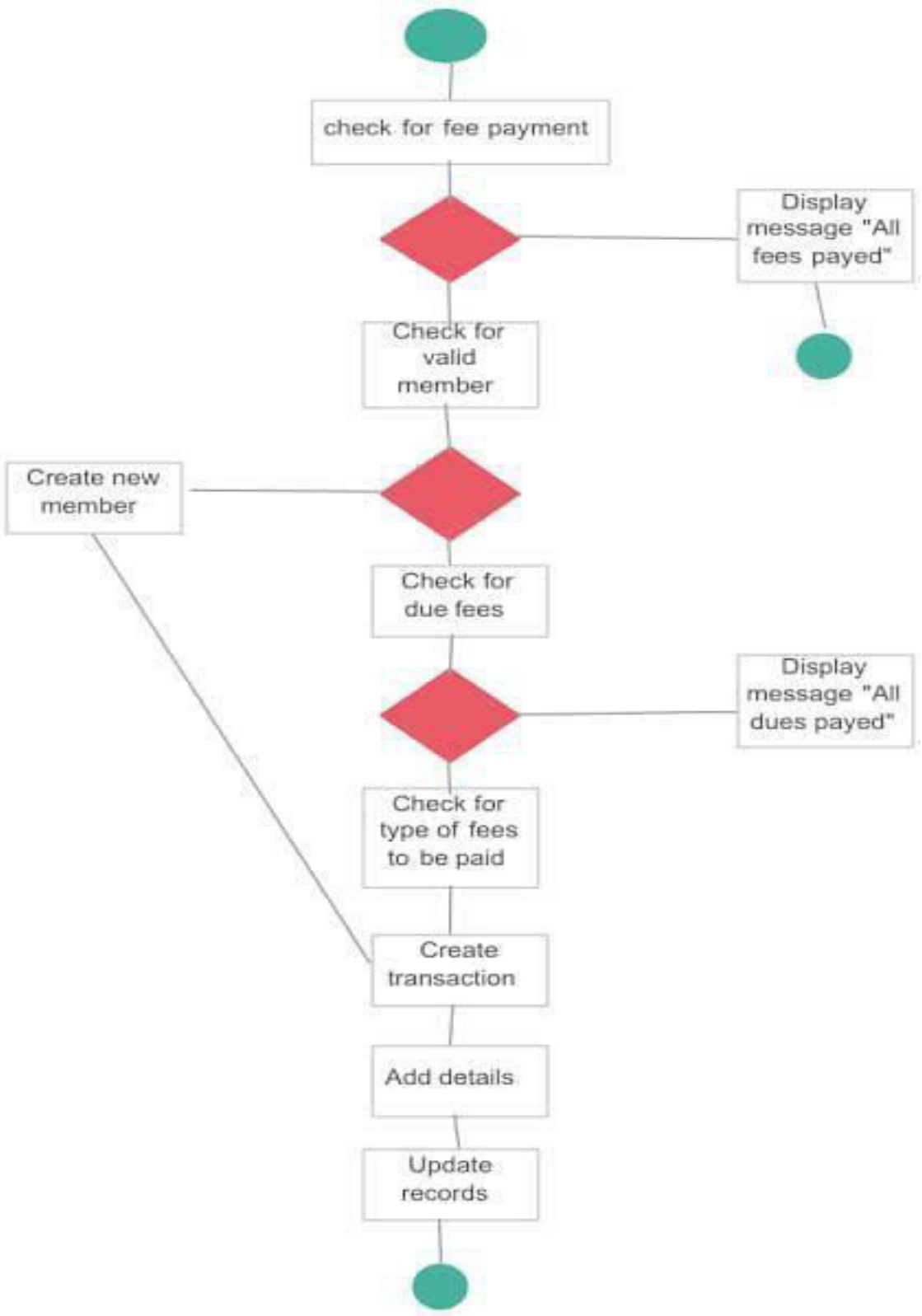
CHAPTER 2
PROJECT METHODOLOGY

2.1Proposed Work

The proposed work involves designing and developing a Student Fee Management System to automate and streamline the management of student fee records in educational institutions. The system will use Python as the programming language and leverage its built-in libraries and functionalities to deliver a robust, efficient, and user-friendly application.

1. **Automate Fee Management:** Reduce manual effort in tracking and recording student fees.
2. **Minimize Errors:** Eliminate errors arising from manual calculations and record-keeping.
3. **Improve Accessibility:** Provide an easy-to-use interface for administrators to manage student fee records efficiently.
4. **Enable Tracking:** Allow for quick and accurate tracking of paid and pending fees.
5. **Generate Reports:** Facilitate the generation of fee-related reports for auditing and analysis.
6. **Student Database Management:** Add, update, and delete student information.Store student data such as name, ID, class, and contact details.
7. **Fee Structure Definition:** Define fee structures for different classes, courses, or programs.Enable adjustments to fee components (e.g., tuition, transportation, library).

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 Module 1 : USER MANAGEMENT

Handles user authentication, access control, and user profile management for administrators, staff, students, and parents.Manages user accounts, permissions, and roles to ensure secure access to fee-related functionalities.

3.2 Module 2 : FEE CALCULATION AND BILLING MODULE

Computes fees based on predefined criteria such as tuition, extracurricular activities
,and other charges.Generates fee invoices, calculates totals, and applies discounts or
adjustments as needed.

3.3 Module 3 : PAYMENT PROCESSING MODULE

Facilitates fee payment transactions through various channels such as online payments, bank transfers, or manual methods. Integrates with payment gateways, tracks payment status, sends payment confirmations, and manages payment reversals or refunds.

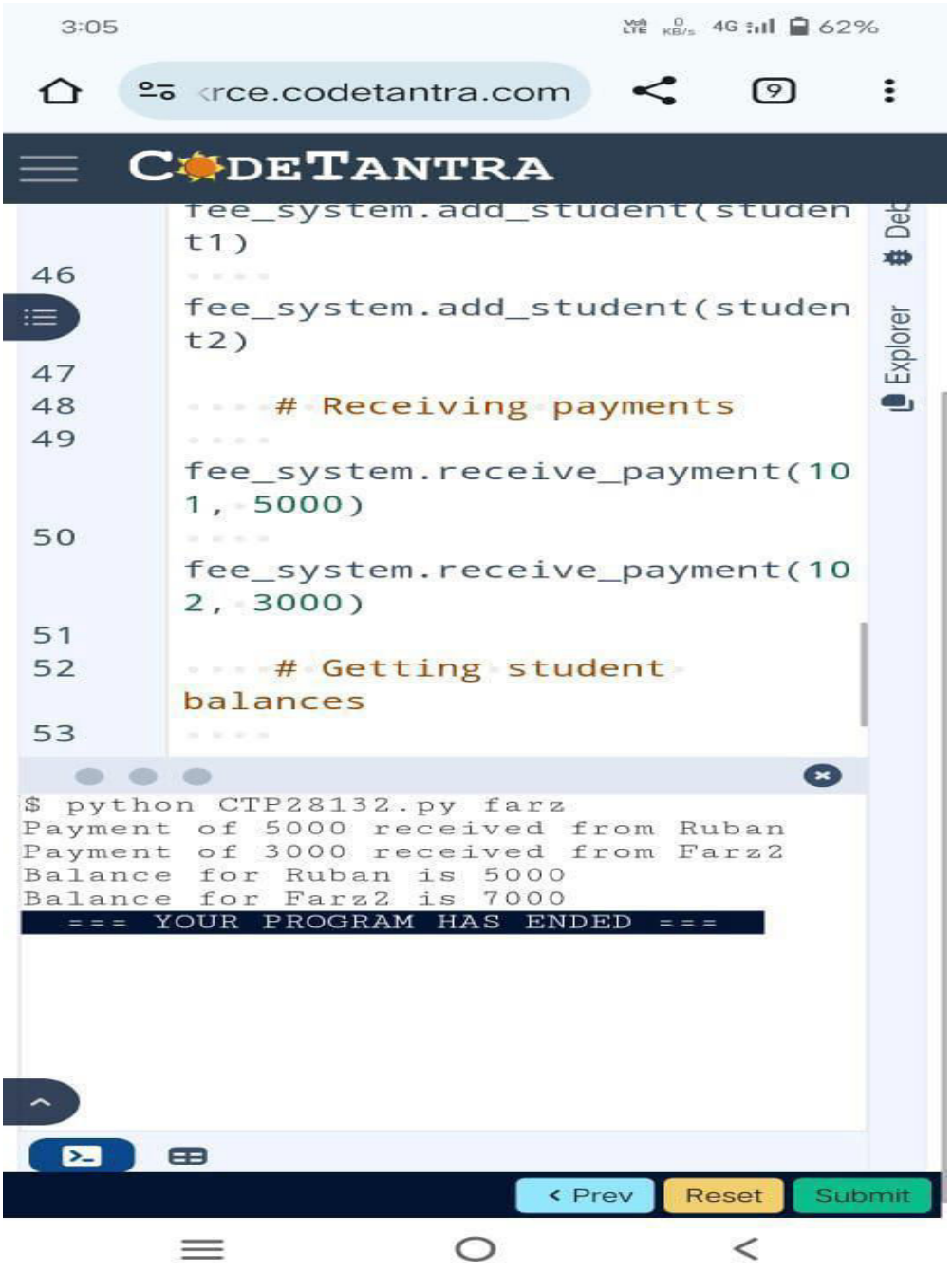
3.4 Module 4 : INTEGRATION WITH STUDENT INFORMATION SYSTEM

Integrates with the institution's SIS to retrieve student information, enrollment status, and academic details relevant to fee calculations. Ensures consistency between fee management and academic records, automates data synchronization, and supports seamless information flow.

CHAPTER 4

RESULTS AND DISCUSSION

Program


3:05
4G 62%
rce.codetantra.com
CODETANTRA
fee_system.add_student(student1)
fee_system.add_student(student2)
# Receiving payments
fee_system.receive_payment(101, 5000)
fee_system.receive_payment(102, 3000)
# Getting student balances
$ python CTP28132.py farz
Payment of 5000 received from Ruban
Payment of 3000 received from Farz2
Balance for Ruban is 5000
Balance for Farz2 is 7000
=== YOUR PROGRAM HAS ENDED ===
Explorer
< Prev
Reset
Submit

DISCUSSION

Encoding techniques play a crucial role in representing data efficiently and securely in computing environments. This discussion delves into the concept of encoding, focusing on Base64 encoding in Python, and explores its significance, applications, and implications in modern software development. Define encoding and its importance in data representation and transmission. Discuss various encoding schemes, including ASCII, UTF-8, and Base64. Explain how encoding transforms data from one format to another, facilitating storage, transmission, and processing. Explain the principles of Base64 encoding and its role in converting binary data into ASCII characters.

Discuss the Base64 alphabet and how it represents binary data using a subset of printable ASCII characters. Explore real-world use cases of Base64 encoding, such as data serialization, cryptographic operations, and data transfer protocols like MIME and HTTP Basic Authentication.

CHAPTER 5

CONCLUSION

In conclusion, the development of a student fee management system program in Python represents a significant advancement in educational administrative tools. This program efficiently handles key tasks such as student registration, fee calculation,and payment tracking, thereby streamlining the management of financial transactions within educational institutions. By leveraging Python's versatility and robust libraries like tkinter for the user interface and SQLite for database management, the system ensures reliability and scalability. Moreover, its intuitive design and user-friendly interface make it accessible to administrators and staff alike, enhancing overall operational efficiency.

Looking into the future, the student fee management system program developed in Python holds immense potential for further enhancement and integration. One promising avenue is the adoption of cloud-based architecture, enabling real-time data access and collaboration across multiple locations. Cloud integration would facilitate seamless updates and backups, ensuring data security and accessibility from any device, which is crucial for modern educational environments.Moreover, the program could benefit from incorporating advanced analytics and machine learning algorithms. By analyzing historical fee data and student payment behaviors, the system could predict cash flow trends, identify potential payment delays, and suggest proactive measures to optimize revenue collection

REFERENCES:

(It may be any Python Books (Refer Syllabus), websites, YouTube links etc..)

1. Anurag Gupta, IPS Jharkhand, GP Biswass, " Python Programming, Problem Solving,Packages and Libraries",McGraw Hill Education (India) Private Limited,2019 ISBN-13:978-93-5316-800-1

2. Reema Theraja , "Python Programming: Using Problem Solving Approach" Oxford Higher Education,2017.

3. Gowrishankar S, Veena A, "Introduction to Python Programming", 1st Edition,CRCPress/Taylor & Francis, 2018. ISBN-13: 978-0815394372

4. https//:w3schools.com

APPENDIX

```
class Student:


    def__init__(self, name, roll_number):
        self.name = name
        self.roll_number = roll_number
        self.fees_paid = 0


    def pay_fee(self, amount):
        self.fees_paid += amount


    def get_balance(self, total_fee):
        return total_fee - self.fees_paid

class FeeManagementSystem:
    def__init__(self):
        self.students = []


    def add_student(self, student):
        self.students.append(student)


    def receive_payment(self, roll_number, amount):
        for student in self.students:
            if student.roll_number == roll_number:
                student.pay_fee(amount)
                print(f"Payment of {amount} received from {student.name}")
                return
        print("Student not found!")
```

```
    def get_student_balance(self, roll_number, total_fee):
        for student in self.students:
            if student.roll_number == roll_number:
                balance = student.get_balance(total_fee)
                print(f"Balance for {student.name} is {balance}")
                return
        print("Student not found!")


# Example usage:
if__name__== "__main__":
    fee_system = FeeManagementSystem()

    # Adding students
    student1 = Student("John", 101)
    student2 = Student("Alice", 102)
    fee_system.add_student(student1)
    fee_system.add_student(student2)

    # Receiving payments
    fee_system.receive_payment(101, 5000)
    fee_system.receive_payment(102, 3000)

    # Getting student balances
    fee_system.get_student_balance(101, 10000)
    fee_system.get_student_balance(102, 10000)
```