I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY, KAPURTHALA

Final Year Project Report

ON

“STUDENT RESULT MANAGEMENT SYSTEM”

LUDHIANA GROUP OF COLLEGES, CHAUKIMAAN

Submitted in Partial Fulfillment of the Requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY (B.TECH)

IN

COMPUTER SCIENCE AND ENGINEERING (CSE)

Submitted By: GURMAN SINGH | Roll No: 2219105
Submitted To: ABHISHEK BANSAL | Academic Year: 2024–2025
Semester - 7th

CERTIFICATE

This is to certify that the project titled "STUDENT RESULT MANAGEMENT SYSTEM" submitted by Gurman Singh, Roll No: 2219105, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B.Tech) in Computer Science and Engineering, is a bona fide project work carried out under the supervision of Abhishek Bansal and Head of Department Gundeep Kaur.

Mr. Abhishek Bansal

Guide Signature: ______________________ Date: ____________

Mrs. Gundeep Kaur

Head of Department Signature: ______________________ Date: ____________

DECLARATION

I hereby declare that the project titled "STUDENT RESULT MANAGEMENT SYSTEM" is my own work and has been carried out under the guidance of Abhishek Bansal. All sources used have been acknowledged and this work has not been submitted elsewhere for any other degree or diploma.

GURMAN SINGH (2219105)

Student Signature: ______________________ Date: ____________

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide, Abhishek Bansal, for their continuous support, valuable suggestions, and guidance throughout this project. Their insights and feedback were instrumental in shaping the project and resolving technical challenges.

I also thank the Head of Department, Gundeep Kaur, and the faculty of the Department of Computer Science & Engineering for providing the facilities and encouragement needed to complete this work. Special thanks to the lab staff and classmates who assisted during testing and troubleshooting. Their cooperation made implementation and evaluation smoother.

Finally, my heartfelt thanks go to my family and friends for their support and motivation during the project. Their encouragement helped me stay focused and complete this work on time.

TABLE OF CONTENTS

ABSTRACT

This project presents a desktop-based Student Result Management System developed in Python using Tkinter and SQLite. It allows teachers to enter student marks, compute totals and percentages automatically, view stored records, and export data as needed. This report documents the design, implementation, and evaluation of the system along with selected code snippets and explanations.

CHAPTER 1 - INTRODUCTION

1.1 Background

Managing student academic records and results is a critical administrative task in educational institutions. Manual processes for entering marks are time-consuming and often introduce errors. An automated system reduces mistakes and accelerates result publication.

1.2 Problem Definition

Traditional manual handling leads to arithmetic errors, delays in preparing consolidated reports, and difficulty in maintaining historical data. For large classes, these problems scale significantly.

1.3 Objectives

- To develop a desktop application for accurate entry and management of student results.
- To automate computation of totals and percentages and ensure persistent storage.

1.4 Scope and Significance

The system targets small to medium-sized institutes where a lightweight desktop solution suffices. It demonstrates core software concepts such as GUI design, database integration, and data validation. With modular design, the system can be extended to support additional features like reporting and analytics.

1.5 System Overview

The Student Result Management System provides modules for adding results, viewing and searching records, deleting incorrect entries, and exporting data for reporting. The application follows a modular structure with separated GUI and database layers to facilitate maintenance and future extension.

CHAPTER 2 - METHODOLOGY / DESIGN

2.1 System Architecture

The application follows a modular architecture: presentation layer (Tkinter GUI), business logic (validation and calculations), and data layer (SQLite). This separation improves maintainability and testing.

2.2 Database Design

A single 'results' table stores the roll number (primary key), student name, subject names, marks, total, and percentage. Proper data types and constraints ensure data integrity.

2.3 GUI Design

The interface provides labelled input fields for roll number, name, subject titles, and marks. Buttons for Add, View, Delete, Clear, and Export enable primary functions. The Treeview widget displays stored records in a grid.

2.4 Algorithm & Data Flow

Upon Add, inputs are validated, numerical conversions are performed, totals and averages computed, and insert operation executed. Viewing retrieves records and populates the UI. Deletion removes selected entries by roll number.

2.5 Tools and Libraries

Python 3.x, Tkinter for UI, SQLite for embedded storage, and optional Pandas for exporting data to Excel were chosen for portability and simplicity.

CHAPTER 3 - IMPLEMENTATION & RESULTS

This chapter highlights key implementation modules and shows selected code snippets that capture the main logic: application entry, database functions, saving data, and viewing results.

Python Code With Logics

3.1 Application Entry (main.py):

```
import time
from gui import run_app

def loading_screen():
    print('Loading...')
    for i in range(3):
        print('.', end='', flush=True)
        time.sleep(0.5)

if __name__ == '__main__':
    loading_screen()
    run_app()
```

Explanation: Shows a brief loading message and starts the GUI application.

3.2 Database Core (database.py):

```
import sqlite3

def connect():
    conn = sqlite3.connect('student.db')
    cur = conn.cursor()
    cur.execute('''CREATE TABLE IF NOT EXISTS results(
        roll_no TEXT PRIMARY KEY,
        name TEXT,
        subject1 TEXT, mark1 REAL,
        subject2 TEXT, mark2 REAL,
        subject3 TEXT, mark3 REAL,
        total REAL, percentage REAL
    )''')
    conn.commit()
    conn.close()
```

Explanation: Creates and manages the SQLite table for storing results.

3.3 Save Logic (gui.py):

```
def save_data():
    roll = roll_no_entry.get()
    name = name_entry.get()
    m1 = float(sub1_mark_entry.get())
    m2 = float(sub2_mark_entry.get())
    m3 = float(sub3_mark_entry.get())
    total = m1 + m2 + m3
    percent = total / 3
    database.insert(roll, name, sub1_name_entry.get(), m1,
sub2_name_entry.get(), m2, sub3_name_entry.get(), m3, total, percent)
    messagebox.showinfo('Success', 'Result Added Successfully!')
```

Explanation: Validates inputs, computes aggregates, and persists the record.

3.4 View Logic (gui.py):

```
def view_data():
    records = database.fetch()
    for i in tree.get_children():
        tree.delete(i)
    for record in records:
        tree.insert('', 'end', values=record)
```

Explanation: Retrieves and displays stored records.

CHAPTER 4 - CONCLUSION & FUTURE SCOPE

Conclusion:

The Student Result Management System meets its objectives of automating result entry, calculation, and storage. It simplifies administrative tasks, reduces errors, and produces results that can be exported for reporting. The modular design aids maintenance and future enhancement.

Future Scope:

- Implement authentication and role-based access control.
- Convert to a web application for multi-user access and centralized storage.
- Add analytics and visualization for class performance.
- Integrate PDF generation and scheduled reporting features.

Viva Summary:

In conclusion, this project successfully automates student result management using Python and SQLite. It provides an easy interface for teachers to enter, calculate, and view results without errors. The system saves time, improves accuracy, and demonstrates how programming can simplify academic record-keeping tasks.

REFERENCES

1. **Python Documentation -** https://docs.python.org/3/

2. **Tkinter Documentation** - https://docs.python.org/3/library/tkinter.html

3. **SQLite Documentation -** https://www.sqlite.org/docs.html