**Vidya Bhawan Public School**

**Project report on**

**Student Management System**

**Session: 2022-23**

**Student Details:**

**Name :** Karan Suthar

**Class :** 12th

**Roll no :**

**Subject :** Computer Science(083)

**Sub. Teacher :** Mrs. Neha Kataria

# **CERTIFICATE**

This is to certify that **JAYDEEP SHRIMALI** a student of Class XII Science has satisfactorily completed the project Entitled “**Grocery Store Management**”, under my **Mrs. NEHA KATARIA Ma’am** guidance and supervision during the session 2022-2023.

## I appreciate his keen interest and sincere efforts in covering all details of the project in a very systematic manner. I am very pleased with his project.

**SUBJECT TEACHE PRINCIPAL EXAMINER**

# **ACKNOWLEDGEMENT**

## I would like to express my special thank of gratitude to my Computer Science teacher **Mrs. NEHA KATARIA Ma’am** for her able guidance and support and letting me work on this wonderful project.

I would also like to extend my deep gratitude towards our honourable Principal **Mrs. NEERJA JAIN Ma’am** for providing us all the required facilities and services.

**Karan Suthar**

XII(Science)

**INTRODUCTION**

The **Student Data Management System** is designed to efficiently handle student records within an educational institution. It streamlines administrative tasks related to student data, including **adding new student profiles**, **updating existing information**, and **removing outdated records**. Through this system, administrators can maintain accurate and up-to-date student details, ensuring smooth operations and effective communication.

**Working:**

* The basics functionality of this is based on mysql which facilates a database and tkinter which provides an gui for better interaction with user .
* This programme provides the an interface for both student and admin where student can see his/her data whereas an admin can manipulate the data of students which includes adding new student profiles, updating existing information, and removing outdated records.

**Structure of project**

1. **Login Window:**
   * Users (both students and administrators) access the system through the login window.
   * The login window prompts users to enter their credentials (username and password).
   * Upon successful authentication, the system grants access to the appropriate interface.
2. **Student Interface:**
   * After logging in, students are directed to their personalized interface.
   * Here, students can view their personal information (name, roll number, contact details, etc.).
3. **Admin Interface:**
   * Administrators (such as teachers, school staff, or system administrators) access the admin panel.
   * Key functionalities in the admin interface include:
     + Adding New Student Profiles:
       - Admins can register new students by inputting their details.
     + Updating Existing Information:
       - Admins can modify student records.
     + Removing Outdated Records:
       - If a student leaves the institution or graduates, admins can archive or delete their records.

**File structure of project**

**Source code:**

**Backend.py**

import mysql.connector as sql  
from CTkMessagebox import CTkMessagebox  
  
  
def insert\_student\_record(roll, name, class\_number, date, gender, phone\_number, guardian\_name, address):  
 *# if gender == 0:  
 # gender = "Male"  
 # else:  
 # gender = "Female"* query = "INSERT INTO STUDENT\_RECORDS (roll\_number, name, class, Date\_Of\_Birth, Gender, Phone\_Number, Guardians\_Name, Address)" \  
 " VALUES ({}, '{}', {}, '{}', '{}', '{}', '{}', '{}' )".format(str(roll), name,  
 str(class\_number), date, gender,  
 str(phone\_number), guardian\_name,  
 address)  
 try:  
 cursor.execute(query)  
 mydb.commit()  
 CTkMessagebox(message="Data is Inserted successfully.",  
 icon="check", option\_1="Ok")  
 print("data entry success")  
 except Exception as e:  
 CTkMessagebox(title="Error", message=("Something went wrong!!!\n {}".format(e)), icon="cancel")  
  
  
  
def update\_student\_record(roll, field, updated\_value):  
 query = "update student\_records set {} = \"{}\" where Roll\_Number={}".format(field, updated\_value, roll)  
 *#update student\_records set name = where Roll\_Number={}* try:  
 cursor.execute(query)  
 mydb.commit()  
 CTkMessagebox(message="Data is Updated successfully.",  
 icon="check", option\_1="Ok")  
 print("data update success")  
 except Exception as e:  
 print(e)  
 CTkMessagebox(title="Error", message=("Something went wrong!!!\n {}".format(e)), icon="cancel")  
  
def delete\_student\_record(roll):  
 query = "delete from student\_records where roll\_number = {}".format(roll)  
 try:  
 cursor.execute(query)  
 mydb.commit()  
 CTkMessagebox(message="Data is Deleted successfully.",  
 icon="check", option\_1="Ok")  
 print("data delete success")  
 except Exception as e:  
 CTkMessagebox(title="Error", message="Something went wrong!!!", icon="cancel")  
  
  
def get\_all\_data():  
 mydb.commit()  
 cursor.execute('SELECT \* FROM STUDENT\_RECORDS')  
 data = cursor.fetchall()  
 return data  
  
def get\_student\_data(roll):  
 mydb.commit()  
 cursor.execute('SELECT \* FROM STUDENT\_RECORDS')  
 data = cursor.fetchall()  
 student\_data = ()  
 for i in data:  
 if i[0] == int(roll):  
 student\_data = i  
 break  
  
 return student\_data  
  
  
  
mydb = sql.connect(host="localhost", user="root", passwd="root") *# Establishing SQL connection*cursor = mydb.cursor() *# Creating Cursor object*cursor.execute("CREATE DATABASE IF NOT EXISTS STUDENT\_MANAGEMENT\_SYSTEM")  
cursor.execute("USE STUDENT\_MANAGEMENT\_SYSTEM")  
cursor.execute('''  
CREATE TABLE IF NOT EXISTS STUDENT\_RECORDS (  
 Roll\_Number INT PRIMARY KEY,  
 Name VARCHAR(50),  
 Class INT,  
 Date\_Of\_Birth DATE,  
 Gender ENUM('Male', 'Female'),  
 Phone\_Number BIGINT,  
 Guardians\_Name VARCHAR(50),  
 Address VARCHAR(255)  
);  
''')