Exp.No – 12

Mini Project

Aim:

To Design a student database (mini project) using Python package tkinter as frontend and MySQL as a backend database

Code:

```
import mysql.connector
import tkinter as tk
from tkinter import messagebox

# Connect to MySQL
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Vidhya@12.",
    database="student_db"
)
cursor = db.cursor()

# Function to add a new student
def add_student():
    name = name_entry.get().strip()
    age = age_entry.get().strip()
    grade = grade_entry.get().strip()
    if name and age and grade:
        sql = "INSERT INTO students (name, age, grade) VALUES (%s, %s, %s)"
        val = (name, age, grade)
        cursor.execute(sql, val)
        db.commit()
        messagebox.showinfo("Success", "Student added successfully!")
    else:
        messagebox.showwarning("Input Error", "Please enter all details.")

# Function to retrieve all students
def get_students():
    cursor.execute("SELECT * FROM students")
    students = cursor.fetchall()
    student_info = "Student ID\tName\tAge\tGrade\n"
    for student in students:
        student_info +=
f"{student[0]}\t{student[1]}\t{student[2]}\t{student[3]}\n"
    messagebox.showinfo("View Students", student_info)

# Function to update student details
```

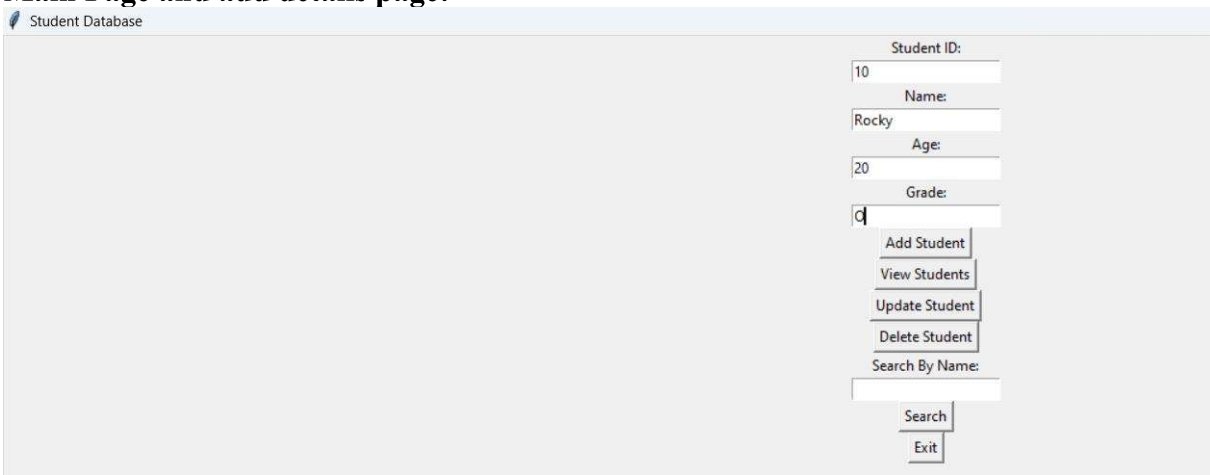
```
def update_student():
    student_id = id_entry.get().strip()
    name = name_entry.get().strip()
    age = age_entry.get().strip()
    grade = grade_entry.get().strip()
    if student_id and name and age and grade:
        sql = "UPDATE students SET name = %s, age = %s, grade = %s WHERE
student_id = %s"
        val = (name, age, grade, student_id)
        cursor.execute(sql, val)
        db.commit()
        messagebox.showinfo("Success", "Student details updated
successfully!")
    else:
        messagebox.showwarning("Input Error", "Please enter all details.")

# Function to delete a student
def delete_student():
    student_id = id_entry.get().strip()
    if student_id:
        sql = "DELETE FROM students WHERE student_id = %s"
        val = (student_id,)
        cursor.execute(sql, val)
        db.commit()
        messagebox.showinfo("Success", "Student deleted successfully!")
    else:
        messagebox.showwarning("Input Error", "Please enter student ID.")

# Function to search students by name
def search_students_by_name():
    name = search_entry.get().strip()
    if name:
        sql = "SELECT * FROM students WHERE name LIKE %s"
        val = ("%" + name + "%",)
        cursor.execute(sql, val)
        students = cursor.fetchall()
        if students:
            student_info = "Student ID\tName\tAge\tGrade\n"
            for student in students:
                student_info +=
f"{student[0]}\t{student[1]}\t{student[2]}\t{student[3]}\n"
            messagebox.showinfo("Search Results", student_info)
        else:
            messagebox.showinfo("Search Results", "No students found with that
name.")
    else:
        messagebox.showwarning("Input Error", "Please enter a name to
search.")
```

```python
# Tkinter window
root = tk.Tk()
root.title("Student Database")

# Labels and Entry fields for student details
tk.Label(root, text="Student ID:").pack()
id_entry = tk.Entry(root)
id_entry.pack()

tk.Label(root, text="Name:").pack()
name_entry = tk.Entry(root)
name_entry.pack()

tk.Label(root, text="Age:").pack()
age_entry = tk.Entry(root)
age_entry.pack()

tk.Label(root, text="Grade:").pack()
grade_entry = tk.Entry(root)
grade_entry.pack()

# Buttons for add, view, update, and delete operations
add_button = tk.Button(root, text="Add Student", command=add_student)
add_button.pack()

get_button = tk.Button(root, text="View Students", command=get_students)
get_button.pack()

update_button = tk.Button(root, text="Update Student", command=update_student)
update_button.pack()

delete_button = tk.Button(root, text="Delete Student", command=delete_student)
delete_button.pack()

# Search bar and button for searching students by name
tk.Label(root, text="Search By Name:").pack()
search_entry = tk.Entry(root)
search_entry.pack()

search_button = tk.Button(root, text="Search",
command=search_students_by_name)
search_button.pack()

exit_button = tk.Button(root, text="Exit", command=root.quit)
exit_button.pack()

root.mainloop()
```

Output:

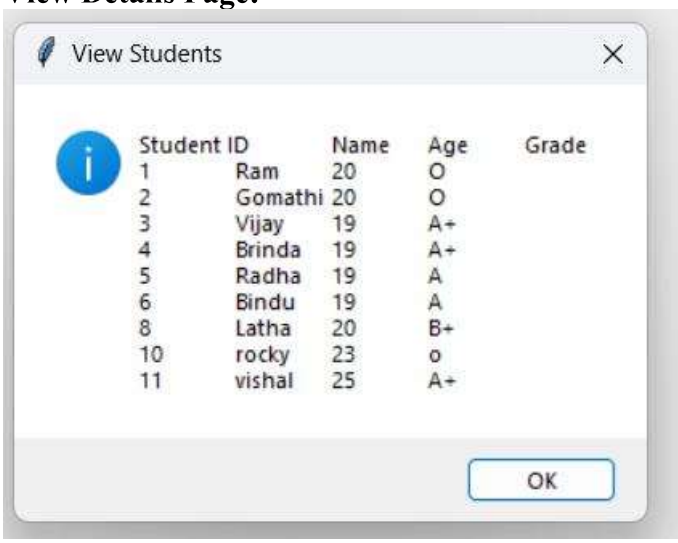
Frontend Design:

Main Page and add details page:



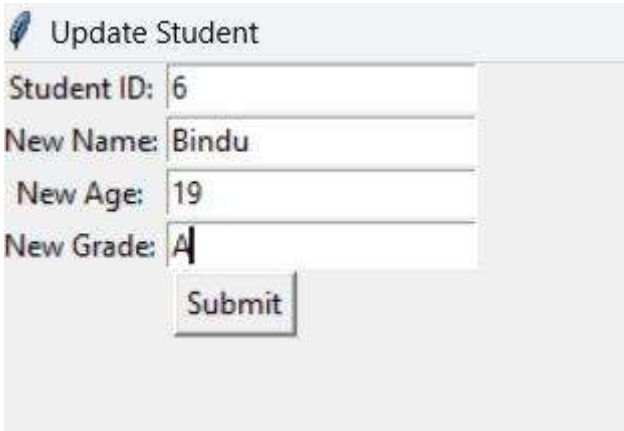
This page allows us to add ,view, update, delete and search the particular student details by their name

View Details Page:



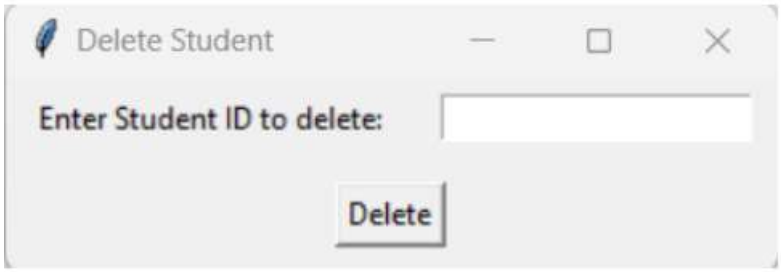
By using this page, we can see all the student's records.

Update Details Page:



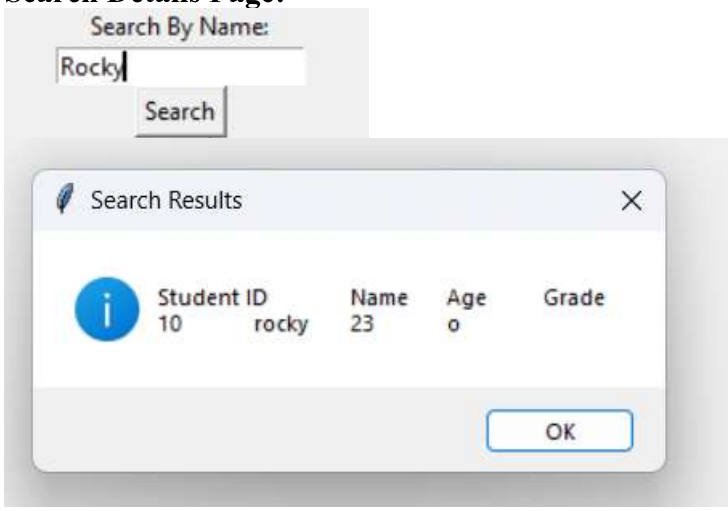
This is the update page. We can update the student's details using their ID on this page.

Delete Details Page:



This is the delete page. We can delete the details of a specific student using their ID.

Search Details Page:



This is the search page. We can search the details of the student by their names. And it fetches the details of the particular student which is stored in student database

Backend Design:

```
1 •  SELECT * FROM student_db.students;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

student_id	name	age	grade
1	Ram	20	O
2	Gomathi	20	O
3	Vijay	19	A+
4	Brinda	19	A+
5	Radha	19	A
6	Bindu	19	A
8	Latha	20	B+
10	rocky	23	o
NULL	NULL	NULL	NULL

MySQL has been used for the backend database. This Stores and displays the data entered in frontend

Result:

Thus the Student Database(mini project) is executed successfully