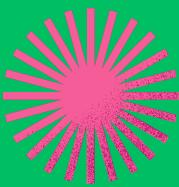


# TABLE OF CONTENTS



Certificate	2
Acknowledgement	3
Declaration	4
Introduction	5
Objective	6
Database Tables	7
Source Code	8
Output Screenshots	15
Bibliography	20

# CERTIFICATE



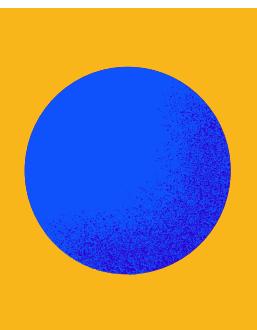
This is to certify that **Ananay Dubey** of class XII-A has prepared the report on the project titled 'Student Assessment System'. This report is the result of his efforts and endeavours. The report is found worthy of acceptance as final project report for the subject Computer Science. The report has been prepared under the guidance of P.G.T. Computer Science, **Mr. Rajat Valecha** within the stipulated time as prescribed by CBSE.

Mr. Rajat Valecha  
**(Internal)**

**(External)**

Mrs. Preeti Poddar  
**(Principal)**

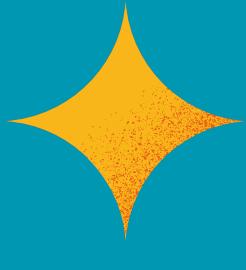
# ACKNOWLEDGEMENT



On completion and submission of the project, I would like to express my deep sense of gratitude to our project guide and teacher **Mr. Rajat Valecha** for his support during the project whenever needed. Without his co-operation, it was impossible to reach this stage.

I am *thankful* to him for giving us this assignment with the purpose of enhancing our knowledge and technical skills. At last, I sincerely regard to my *parents* and *friends* who have directly or indirectly helped me in this project.

# DECLARATION



We declare that the work presented in this project titled "Student Assessment System", submitted to **Mr. Rajat Valecha**, P.G.T. Computer Science. We have not plagiarized or submitted the same work for the award of any other examination. In case this undertaking is found incorrect, we accept that our Certificates may be unconditionally withdrawn.

**Student Name:** Ananay Dubey 12A

# INTRODUCTION



Welcome to my innovative school-level project aimed at revolutionizing student and teacher management! In this comprehensive system, we have developed an efficient database-driven platform to streamline various administrative tasks, ranging from student record management to teacher attendance tracking. Our project integrates a user-friendly interface that facilitates easy data entry, modification, and retrieval. Students can benefit from features like report card generation and personal data updates, while teachers have access to attendance tracking tools. The administration gains insights into financial summaries and teacher statistics. Join us on this journey to enhance the efficiency and organization of school management through our dynamic and user-centric project.

# OBJECTIVES



## Efficiency Enhancement:

- Streamline administrative tasks, reducing manual efforts and saving time for teachers and administrators. Implement a database-driven system to automate student record management, ensuring quick and accurate data handling.

## Data Accuracy and Reliability:

- Establish a platform that prioritizes data accuracy, minimizing errors in student and teacher records. Regularly update and maintain the database to provide reliable and up-to-date information for informed decision-making.

## Insightful Reporting Tools:

- Integrate robust reporting tools for generating student report cards and financial summaries. Provide administrators with valuable insights into academic performance and financial aspects for effective decision-making.

## Attendance Tracking and Financial Transparency:

- Implement a comprehensive attendance tracking system for teachers, promoting accountability in class management. Foster financial transparency by offering administrators a clear overview of earnings from student fees and spending on teacher salaries.

# DATABASE TABLES



id	name	class	section	scholar_number	gender	attendance	phy	chem	maths	eng	comp	total_marks	phone_number	height	weight	fee
8	Ishan Gupta	12	C	1001	M	174	90	80	81	85	56	420	+911058987673	179.2	73.0	8000
11	Divyanshi Mishra	12	A	1002	F	200	80	84	83	85	95	418	+919540711760	154.8	57.4	8000
4	Ananay Dubey	12	A	1003	M	184	85	85	95	90	93	450	+919857176774	174.3	65.6	8000
22	Aditya Kumar	11	C	1004	M	178	82	94	92	85	94	447	+910774591167	171.5	68.9	8000
40	Arunjay Sachan	12	B	1005	M	156	89	91	85	79	92	436	+910904839188	163.8	47.4	8000
24	Shikhar Yadav	12	B	1006	F	167	90	91	94	94	76	445	+917473962843	156.6	58.7	8000
1	Arush Nigam	12	A	1007	M	164	92	80	89	84	83	428	+911576627352	159.4	70.9	8000
3	Shashwat Agnihotri	10	C	1008	M	177	84	86	79	88	83	420	+916697681332	150.8	75.2	8000
9	Faria Hasan	11	E	1009	F	159	91	84	83	91	79	428	+917342514915	178.7	44.1	8000
38	Amoghya Mishra	12	B	1010	M	194	95	95	87	89	91	457	+914758917430	166.8	43.7	8000
2	Arpit Bala	12	A	1011	M	189	83	92	76	89	82	422	+917111964796	174.8	66.0	8000
25	Nirupaksh Mishra	10	A	1012	F	177	86	75	86	87	82	416	+918348161635	164.7	78.1	8000
14	Aakash Patel	12	B	1013	M	163	77	80	87	88	84	416	+911034873887	162.7	42.4	8000
15	Krishna Gera	11	C	1014	F	190	79	80	94	75	84	412	+916837130718	163.4	70.8	8000
37	Hunar Sarna	10	A	1015	F	169	85	87	79	79	86	416	+917416368766	155.1	75.6	8000
10	Mohd Ahad	11	B	1016	F	154	79	88	94	81	86	428	+917784034642	171.4	77.9	8000
39	Lakshya Kumari	10	B	1017	F	163	94	76	84	76	81	411	+914382604421	165.2	45.0	8000
13	Nirvaan Koshy	10	B	1018	F	187	82	89	90	77	75	413	+910851483631	179.6	62.8	8000
31	Jaskeerat Singh	10	B	1019	F	154	76	91	86	79	90	422	+913708047701	151.2	69.0	8000
26	Vihaan Ramaswamy	11	B	1020	M	150	88	77	80	90	88	423	+911077391440	154.2	49.8	8000
12	Anvi	11	B	1021	F	178	84	77	80	76	90	407	+919556757056	174.4	50.6	8000
20	Cameron Welsh	12	B	1022	M	170	87	78	75	91	84	415	+916058710195	166.6	52.8	8000
36	Anvi Vohra	12	R	1023	F	199	81	78	84	93	81	417	+918829362636	152.8	63.3	8000

## STUDENT

id	name	attendance	salary
1	Teacher1	180	40000
2	Teacher2	190	45000
3	Teacher3	200	50000
4	Teacher4	180	56000
5	Teacher5	190	55000
6	Teacher6	200	44000

## TEACHER

7



```
import mysql.connector

connection = mysql.connector.connect(
    host="127.0.0.1",
    user="root",
    password="pass",
    database="student"
)

cursor = connection.cursor()

cursor.execute("""
    CREATE TABLE IF NOT EXISTS Student (
        id INT AUTO_INCREMENT UNIQUE,
        name VARCHAR(255) DEFAULT '-',
        class INT DEFAULT 12,
        section VARCHAR(1) DEFAULT '-',
        scholar_number INT PRIMARY KEY,
        gender VARCHAR(1) DEFAULT '-',
        attendance INT DEFAULT 0,
        phy INT DEFAULT 0,
        chem INT DEFAULT 0,
        maths INT DEFAULT 0,
        eng INT DEFAULT 0,
        comp INT DEFAULT 0,
        total_marks INT,
        phone_number VARCHAR(15) DEFAULT '-',
        height VARCHAR(5) DEFAULT '-',
        weight VARCHAR(5) DEFAULT '-',
        fee INT DEFAULT 8000
    )
""")

cursor.execute("""
    CREATE TABLE IF NOT EXISTS Teachers (
        id INT AUTO_INCREMENT UNIQUE,
        name VARCHAR(255) DEFAULT '-',
        attendance INT DEFAULT 50,
        salary INT DEFAULT 50000,
        PRIMARY KEY (id)
    )
""")
```



```
def insert_student_data():
    name = input("Name: ").capitalize()
    class_ = int(input("Class(*): "))
    section = input("Section: ").upper()
    scholar_number = int(input("Scholar Number(*): "))
    gender = input("Gender (M/F): ").upper()
    attendance = int(input("Attendance(/200*): "))
    phone_number = input("Phone Number: ") or 'xxxxxxxxxx'
    height = float(input("Height(cm*): "))
    weight = float(input("Weight(kg*): "))
    phy = int(input("Physics: "))
    chem = int(input("Chemistry: "))
    maths = int(input("Maths: "))
    eng = int(input("English: "))
    comp = int(input("Computer: "))
    total_marks = phy + chem + maths + eng + comp
    fee = 7000

    insert_query = """
        INSERT INTO Student (name, class, section,
        scholar_number, gender, attendance, phy, chem,
        maths, eng, comp, total_marks, phone_number,
        height, weight, fee)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s)
    """
    data = (name, class_, section, scholar_number, gender,
            attendance, phy, chem, maths, eng, comp,
            total_marks, phone_number, height, weight, fee)
    cursor.execute(insert_query, data)
    connection.commit()
    print("Data inserted successfully.")

def view_report_card():
    student_id = int(input("Enter Scholar No.(1000-1040): "))
    query = """
        SELECT name, class, section, attendance, phy,
        chem, maths, eng, comp, total_marks
        FROM Student
        WHERE scholar_number = %s
    """
    cursor.execute(query, (student_id,))
    result = cursor.fetchone()
    total = 100
    if result:
        c = (result[9] / (5 * total)) * 100
        classec = f"{result[1]}{result[2]}"
        print("\n----- REPORT CARD -----")
        print(f"Name: {result[0]}")
        print(f"Class: {classec}")
        print(f"Chemistry: {result[5]}/{total}")
        print(f"Maths: {result[6]}/{total}")
```



```
print(f"English: {result[7]}/{total}")
print(f"Computer: {result[8]}/{total}")
print(f"Total Marks: {result[9]}/{5 * total}")
print(f"Percentage: {c}")
print("-----")
else:
    print("Student not found.")

def modify_entry():
    student_id = int(input("Enter Scholar Number: "))
    print("Choose a field to modify:")
    print("1. Name")
    print("2. Class")
    print("3. Section")
    print("4. Scholar Number")
    print("5. Gender")
    print("6. Attendance")
    print("7. Physics")
    print("8. Chemistry")
    print("9. Maths")
    print("10. English")
    print("11. Computer")
    print("12. Phone Number")
    print("13. Height")
    print("14. Weight")
    choice = int(input("Enter your choice: "))

    fields = [
        "name", "class", "section", "scholar_number", "gender",
        "attendance", "phy", "chem", "maths", "eng", "comp",
        "phone_number", "height", "weight"
    ]

    if 1 <= choice <= len(fields):
        field_name = fields[choice - 1]
        new_value = input(f"Enter new value for {field_name}: ")

        update_query = f"""
            UPDATE Student
            SET {field_name} = %s
            WHERE scholar_number = %s
        """
        cursor.execute(update_query, (new_value, student_id))
        connection.commit()
        print("Entry modified successfully.")
    else:
        print("Invalid choice.")
```



```
def view_full_table():
    query = "SELECT * FROM Student"
    cursor.execute(query)
    results = cursor.fetchall()

    if not results:
        print("No records found.")
    else:
        print("\n----- FULL STUDENT TABLE -----")
        print(''{:<5} {:<20} {:<5} {:<5} {:<8} {:<5} {:<12} {:<5}{:<5}
              {:<5} {:<5} {:<5} {:<15} {:<10} {:<10}''.format(
            "ID", "Name", "Class", "Sec", "Schl No.", "Gender",
            "Attendance", "Phy", "Chem", "Maths", "Eng", "Comp",
            "Total", "Phone", "Height", "Weight"
        ))
        print("-" * 120)

        for row in results:
            print(''{:<5} {:<20} {:<5} {:<5} {:<10} {:<5}
                  {:<12} {:<5} {:<5} {:<5} {:<5} {:<5} {:<15}
                  {:<10} {:<10}''.format(
                row[0], row[1], row[2], row[3], row[4], row[5],
                row[6], row[7], row[8], row[9], row[10], row[11],
                row[12], row[13], row[14], row[15]
            ))

def delete_student_record():
    scholar_number = int(input("Enter Scholar Number to delete: "))
    delete_query = "DELETE FROM Student WHERE scholar_number = %s"
    cursor.execute(delete_query, (scholar_number,))
    connection.commit()
    print("Student record deleted successfully.")

def search_student():
    search_field = input("Enter the field to search (name/id/class/gender/): ")
    search_value = input(f"Enter the {search_field} to search: ")
    search_query = f"SELECT id, name, class, section, gender FROM Student
    WHERE {search_field} = %s"
    cursor.execute(search_query, (search_value,))
    results = cursor.fetchall()

    if not results:
        print("No records found.")
    else:
        print("\n----- SEARCH RESULTS -----")
        for row in results:
            print(row)
```



```
def average_marks():
    average_query = '''SELECT AVG(phy), AVG(chem), AVG(maths),
    AVG(eng), AVG(comp) FROM Student'''
    cursor.execute(average_query)
    averages = cursor.fetchone()

    print("\n----- AVERAGE MARKS -----")
    print(f"Physics: {averages[0]}")
    print(f"Chemistry: {averages[1]}")
    print(f"Maths: {averages[2]}")
    print(f"English: {averages[3]}")
    print(f"Computer: {averages[4]}")

def show_teachers_table():
    query = "SELECT * FROM Teachers"
    cursor.execute(query)
    results = cursor.fetchall()

    if not results:
        print("No records found in Teachers table.")
    else:
        print("\n----- TEACHERS TABLE -----")
        print("{:<5} {:<20} {:<12} {:<5}".format("ID", "Name",
        "Attendance", "Salary"))
        print("-" * 40)

        for row in results:
            print("{:<5} {:<20} {:<10} {:<5}".format(row[0],
            row[1], row[2], row[3]))

def teacher_attendance_tracker():
    teacher_name = input("Enter teacher's name: ").capitalize()

    query = "SELECT attendance, salary FROM Teachers WHERE name = %s"
    cursor.execute(query, (teacher_name,))
    result = cursor.fetchone()

    if result:
        print("\n----- TEACHER ATTENDANCE TRACKER -----")
        print(f"Teacher: {teacher_name}")
        print(f"Attendance: {result[0]}")
        print(f"Salary: {result[1]}")
    else:
        print(f"No information found for teacher: {teacher_name}")

def delete_teacher_by_name():
    teacher_name = input("Enter teacher's name to delete: ").capitalize()
    delete_query = "DELETE FROM Teachers WHERE name = %s"
    cursor.execute(delete_query, (teacher_name,))
    connection.commit()
    print(f"Teacher '{teacher_name}' deleted successfully.")
```



```
def admin_portal_menu():
    while True:
        print("\nADMIN PORTAL MENU:")
        print("1. Teachers Attendance Tracker")
        print("2. Finances")
        print("3. Show Teachers Table")
        print("4. Delete a Teacher")
        print("5. Exit")

        admin_choice = input("Select an option (1/2/3/4): ")

        if admin_choice == '1':
            teacher_attendance_tracker()
        elif admin_choice == '2':
            earning_query = '''SELECT IFNULL(SUM(fee), 0)
FROM Student'''
            cursor.execute(earning_query)
            total_earning = cursor.fetchone()[0]

            spending_query = '''SELECT IFNULL(SUM(salary), 0)
FROM Teachers'''
            cursor.execute(spending_query)
            total_spending = cursor.fetchone()[0]

            net_balance = total_earning - total_spending

            print("\n----- FINANCIAL SUMMARY -----")
            print(f"Earnings from Student Fees: {total_earning}")
            print(f"Spending on Teachers' Salaries: {total_spending}")
            print(f"Net Profit/Loss: {net_balance}")

        elif admin_choice == '3':
            show_teachers_table()
        elif admin_choice == '4':
            delete_teacher_by_name()
        elif admin_choice == '5':
            print("Exiting the Admin Portal.")
            break
        else:
            print("Invalid choice. Please enter a valid option.")
```



```
def student_portal_menu():
    while True:
        print("\nSTUDENT PORTAL MENU:")
        print("1. Insert Student Data")
        print("2. View Report Card")
        print("3. Modify Entry")
        print("4. View Full Student Table")
        print("5. Delete A Student")

        student_choice = input("Select an option (1/2/3/4/5): ")

        if student_choice == '1':
            insert_student_data()
        elif student_choice == '2':
            view_report_card()
        elif student_choice == '3':
            modify_entry()
        elif student_choice == '4':
            view_full_table()
        elif student_choice == '5':
            delete_student_record()
        elif student_choice == '6':
            print("Exiting the Student Portal.")
            break
        else:
            print("Invalid choice. Please enter a valid option.")

def main_menu():
    while True:
        print("\nMAIN MENU:")
        print("1. Admin Portal")
        print("2. Student Portal")
        print("3. Exit")

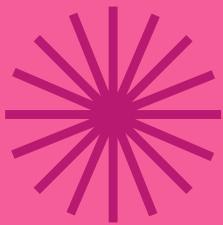
        main_choice = input("Select an option (1/2/3): ")

        if main_choice == '1':
            admin_portal_menu()
        elif main_choice == '2':
            student_portal_menu()
        elif main_choice == '3':
            print("Exiting the program.")
            break
        else:
            print("Invalid choice. Please enter a valid option.")

main_menu()

cursor.close()
connection.close()
```

# OUTPUTS (MENU)



```
PS E:\school> python main.py
```

MAIN MENU:

1. Admin Portal
2. Student Portal
3. Exit

```
Select an option (1/2/3): 1
```

ADMIN PORTAL MENU:

1. Teachers Attendance Tracker
2. Finances
3. Show Teachers Table
4. Delete a Teacher
5. Exit

```
Select an option (1/2/3/4): 5
```

```
Exiting the Admin Portal.
```

MAIN MENU:

1. Admin Portal
2. Student Portal
3. Exit

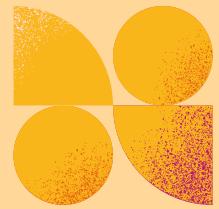
```
Select an option (1/2/3): 2
```

STUDENT PORTAL MENU:

1. Insert Student Data
2. View Report Card
3. Modify Entry
4. View Full Student Table
5. Delete A Student

```
Select an option (1/2/3/4/5): 0
```

# OUTPUTS (ADMIN)



ADMIN PORTAL MENU:

1. Teachers Attendance Tracker ✓
2. Finances
3. Show Teachers Table
4. Delete a Teacher
5. Exit

Select an option (1/2/3/4): 1

Enter teacher's name: Teacher1

----- TEACHER ATTENDANCE TRACKER -----

Teacher: Teacher1

Attendance: 180

Salary: 40000

ADMIN PORTAL MENU:

1. Teachers Attendance Tracker
2. Finances
3. Show Teachers Table
4. Delete a Teacher
5. Exit

Select an option (1/2/3/4): 1

ADMIN PORTAL MENU:

1. Teachers Attendance Tracker ✓
2. Finances ✓
3. Show Teachers Table ✓
4. Delete a Teacher ✓
5. Exit

Select an option (1/2/3/4): 3

----- TEACHERS TABLE -----

ID	Name	Attendance	Salary
2	Teacher2	190	45000
3	Teacher3	200	50000
4	Teacher4	180	56000
5	Teacher5	190	55000

ADMIN PORTAL MENU:

1. Teachers Attendance Tracker
2. Finances
3. Show Teachers Table
4. Delete a Teacher
5. Exit

Select an option (1/2/3/4): 1

1. Teachers Attendance Tracker ✓

2. Finances ✓

3. Show Teachers Table

4. Delete a Teacher

5. Exit

Select an option (1/2/3/4): 2

----- FINANCIAL SUMMARY -----

Earnings from Student Fees: 311000

Spending on Teachers' Salaries: 290000

Net Profit/Loss: 21000

ADMIN PORTAL MENU:

1. Teachers Attendance Tracker
2. Finances
3. Show Teachers Table
4. Delete a Teacher
5. Exit

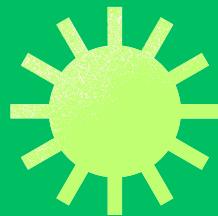
Select an option (1/2/3/4): 3

----- TEACHERS TABLE -----

ID	Name	Attendance	Salary
1	Teacher1	180	40000
2	Teacher2	190	45000
3	Teacher3	200	50000
4	Teacher4	180	56000
5	Teacher5	190	55000
6	Teacher6	200	44000



# OUTPUTS (STUDENT)



```
-- REPORT CARD --
Name: Ananay Dubey
Class: 12B
Attendance: 184/200
Physics: 95/100
Chemistry: 85/100
Maths: 95/100
English: 90/100
Computer: 93/100
Total Marks: 450/500
Percentage: 90.0

STUDENT PORTAL MENU:
1. Insert Student Data ✓
2. View Report Card ✓
3. Modify Entry ✓
4. View Full Student Table
5. Delete A Student
Select an option (1/2/3/4/5): 3
Enter Scholar Number: 1003
Choose a field to modify:
1. Name
2. Class
3. Section
4. Scholar Number
5. Gender
```

11. Computer  
 12. Phone Number  
 13. Height  
 14. Weight  
 Enter your choice: 3  
 Enter new value for section: A  
 Entry modified successfully.

STUDENT PORTAL MENU:  
 1. Insert Student Data  
 2. View Report Card  
 3. Modify Entry  
 4. View Full Student Table  
 5. Delete A Student

Select an option (1/2/3/4/5): 2  
 Enter Scholar No.(1000-1040): 1003

-- REPORT CARD --
 Name: Ananay Dubey
 Class: 12A
 Attendance: 184/200
 Physics: 95/100
 Chemistry: 85/100
 Maths: 95/100
 English: 90/100
 Computer: 93/100
 Total Marks: 450/500
 Percentage: 90.0

STUDENT PORTAL MENU:  
 1. Insert Student Data ✓  
 2. View Report Card ✓  
 3. Modify Entry ✓  
 4. View Full Student Table ✓  
 5. Delete A Student ✓  
 Select an option (1/2/3/4/5): 5  
 Enter Scholar Number to delete: 1039  
 Student record deleted successfully.

	28	Ekansh Grose	10	A	1054
	17	Darsshit Banik	12	B	1054
	30	Ojas Tripathi	11	A	1035
	35	Ishaan Dada	12	A	1036
	41	Dhanush Tandon	12	A	1037
	-		-	A	1041
	1038		1037	A	1041
	145		1038	F	1041
	145		145	M	1041
	92		92	M	1041
	90		90	F	1041
	94		94	F	1041
	86		86	F	1041
	98		98	M	1041
	460		460	M	1041
	443		443	F	1041
	+919543278960		+919543278960	F	1041
	159.3		159.3	F	1041
	168.1		168.1	M	1041
	52.2		52.2	M	1041
	77.9		77.9	F	1041
	62.0		62.0	F	1041

# OUTPUTS (STUDENT)

Full Student Table													Height	Weight	
ID	Name	Class	Sch1	No.	Gender	Attendance	Phy	Chem	Maths	Eng	Comp	Total	Phone		
8	Ishan Gupta	12	A	1001	M	174	90	80	81	85	56	420	+911058987673	179.2	73.0
11	Divyanshi Mishra	12	A	1002	F	200	80	84	83	85	95	418	+919540711760	154.8	57.4
4	Ananay Dubey	12	B	1003	M	184	85	85	95	90	93	450	+919855717674	174.3	65.6
22	Aditya Kumar	11	C	1004	M	178	82	94	92	85	94	447	+918774591167	171.5	68.9
40	Arunjay Sachan	12	C	1005	M	156	89	91	85	79	92	436	+9109904839188	163.8	47.4
24	Shikhar Yadav	12	C	1006	F	167	90	91	94	94	76	445	+917473962843	156.6	58.7
1	Arush Nigam	12	A	1007	M	164	92	80	89	84	83	428	+911576627352	159.4	70.9
3	Shashwat Agnihotri	10	C	1008	M	177	84	86	79	88	83	420	+916697681332	150.8	75.2
9	Faria Hasan	11	B	1009	F	159	91	84	83	91	79	428	+917342514915	178.7	44.1
38	Amoghya Mishra	12	A	1010	M	194	95	95	87	89	91	457	+914758917430	166.8	43.7
2	Arpit Bala	12	B	1011	M	189	83	92	76	89	82	422	+917111964796	174.8	66.0
25	Nirupaksh Mishra	10	A	1012	F	177	86	75	86	87	82	416	+918348161635	164.7	78.1
14	Aakash Patel	12	B	1013	M	163	77	80	87	88	84	416	+9110348723887	162.7	42.4
15	Krishna Gera	11	C	1014	F	190	79	80	94	75	84	412	+916837130718	163.4	70.8
37	Hunar Sarna	10	A	1015	F	169	85	87	79	79	86	416	+917416368766	155.1	75.6
10	Mohd Ahad	11	B	1016	F	154	79	88	94	81	86	428	+917784034642	171.4	77.9
39	Lakshya Kumari	10	B	1017	F	163	94	76	84	76	81	411	+914382604421	165.2	45.0
13	Nirvaan Koshy	10	B	1018	F	187	82	89	90	77	75	413	+919851485631	179.6	62.8

Weight(kg\*): 62  
 Physics: 92  
 Chemistry: 90  
 Maths: 94  
 English: 86  
 Computer: 98  
 Data inserted successfully.

STUDENT PORTAL MENU:

1. Insert Student Data
2. View Report Card
3. Modify Entry
4. View Full Student Table
5. Delete A Student

Select an option (1/2/3/4/5) : 4

# BIBLIOGRAPHY



- <https://python.org/>
- <https://www.codecademy.com/>
- <https://www.tutorialspoint.com/>
- <https://www.pythonchallenge.com/>
- <https://learnpython.org>
- <https://chat.openai.com>
- <https://youtube.com>

THANK YOU

THANK YOU

THANK YOU

THANK YOU

**THANK YOU**

THANK YOU

THANK YOU

THANK YOU

THANK YOU