

Final Project :- Student Performer Tracker

Team Member Name :-

Vishal Mohil , email :- vmohil512@gmail.com , phone no. – 6375665987

Neha Yadav , email :- nehayadav5170@gmail.com , phone no. - 8851701493

Code :-

App.py

```
from flask import Flask, render_template, request, redirect, url_for, flash
import database

app = Flask(__name__)
app.secret_key = "supersecret"

@app.route("/")
def index():
    students = database.list_students()
    stats = {
        "total_students": database.total_students(),
        "total_subjects": database.total_subjects(),
        "total_grades": database.total_grades(),
    }
    return render_template("index.html", students=students, stats=stats)

# ----- Add Student -----
@app.route("/add-student", methods=["GET", "POST"])
def add_student():
    if request.method == "POST":
        roll = request.form.get("roll", "").strip()
        name = request.form.get("name", "").strip()
        try:
            if not roll or not name:
                raise ValueError("Roll number and name are required.")
            database.add_student(roll, name)
            flash("☑ Student added successfully!", "success")
            return redirect(url_for("index"))
        except Exception as e:
```

```

        flash(f"❌ {e}", "error")
        return redirect(url_for("add_student"))
    return render_template("add_student.html")

# ----- Add Grade -----
@app.route("/add-grade", methods=["GET", "POST"])
def add_grade():
    if request.method == "POST":
        roll = request.form.get("roll", "").strip()
        subject = request.form.get("subject", "").strip()
        grade_raw = request.form.get("grade", "").strip()
        try:
            if not roll or not subject or not grade_raw:
                raise ValueError("All fields are required.")
            grade = int(grade_raw)
            if grade < 0 or grade > 100:
                raise ValueError("Grade must be between 0 and 100.")
            database.add_grade(roll, subject, grade)
            flash("✅ Grade added successfully!", "success")
            return redirect(url_for("index"))
        except Exception as e:
            flash(f"❌ {e}", "error")
            return redirect(url_for("add_grade"))
    return render_template("add_grade.html")

# ----- View Student -----
@app.route("/student/<roll>")
def view_student(roll):
    student = database.get_student_by_roll(roll)
    grades = database.get_grades_by_roll(roll)
    if not student:
        flash("❌ Student not found.", "error")
        return redirect(url_for("index"))
    return render_template("view_student.html", student=student, grades=grades)

# ----- Reports -----
@app.route("/reports")
def reports():
    report_data = database.get_full_report()
    return render_template("reports.html", report_data=report_data)

# ----- Subject Toppers -----
@app.route("/subject-topper")
def subject_topper():
    toppers = database.get_subject_toppers()

```

```

        return render_template("subject_topper.html", toppers=toppers)

# ----- Class Average (Chart) -----
@app.route("/class-average")
def class_average():
    data = database.get_class_average()
    subjects = [d["subject"] for d in data]
    values = [d["avg_grade"] for d in data]
    totals = [d["total_students"] for d in data]
    return render_template("class_averages_chart.html", subjects=subjects,
values=values, totals=totals)

if __name__ == "__main__":
    database.init_db()
    app.run(debug=True)

```

cli.py

```

# cli.py
from models import StudentTracker

tracker = StudentTracker()

def menu():
    print("""
===== Student Performance Tracker =====
1. Add Student
2. Add Grade
3. View Student Details
4. List Students
5. Calculate Student Average
6. Subject Topper
7. Class Average
0. Exit
""")

while True:
    menu()
    choice = input("Enter choice: ").strip()

    if choice == "1":
        name = input("Enter name: ")
        roll = input("Enter roll number: ")

```

```

    try:
        tracker.add_student(name, roll)
        print("☑ Student added!")
    except Exception as e:
        print("☒", e)

elif choice == "2":
    roll = input("Roll number: ")
    subject = input("Subject: ")
    try:
        grade = float(input("Grade (0-100): "))
        tracker.add_grade(roll, subject, grade)
        print("☑ Grade added!")
    except Exception as e:
        print("☒", e)

elif choice == "3":
    roll = input("Roll number: ")
    details = tracker.get_student_details(roll)
    if not details:
        print("☒ Student not found.")
    else:
        print("Name:", details["name"])
        print("Roll:", details["roll_number"])
        if details["grades"]:
            for subj, grd in details["grades"].items():
                print(f"  {subj}: {grd}")
        else:
            print("No grades yet.")

elif choice == "4":
    students = tracker.list_students()
    if not students:
        print("No students found.")
    else:
        for s in students:
            print(s["roll_number"], "-", s["name"])

elif choice == "5":
    roll = input("Roll number: ")
    avg = tracker.calculate_average(roll)
    if avg is None:
        print("☒ No grades found for this student.")
    else:
        print(f"Average: {avg:.2f}")

```

```

elif choice == "6":
    subject = input("Subject: ")
    topper = tracker.subject_topper(subject)
    if topper:
        print(f"Topper in {subject}: {topper['name']} ({topper['roll_number']} - {topper['grade']})")
    else:
        print("❌ No data for this subject.")

elif choice == "7":
    subject = input("Subject: ")
    avg = tracker.class_average(subject)
    if avg is None:
        print("❌ No data for this subject.")
    else:
        print(f"Class average in {subject}: {avg:.2f}")

elif choice == "0":
    print("Bye!")
    break
else:
    print("Invalid choice.")

```

database.py

```

import sqlite3
from contextlib import closing

DB_NAME = "student_tracker.db"

def get_connection():
    conn = sqlite3.connect(DB_NAME)
    conn.row_factory = sqlite3.Row # dict-like rows
    return conn

# ----- INIT -----
def init_db():
    with closing(get_connection()) as conn:
        conn.execute("""
            CREATE TABLE IF NOT EXISTS students (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                roll_number TEXT UNIQUE NOT NULL,
                name TEXT NOT NULL
            )
        """)

```

```

    )
    """
    conn.execute("""
        CREATE TABLE IF NOT EXISTS grades (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            student_id INTEGER NOT NULL,
            subject TEXT NOT NULL,
            grade INTEGER NOT NULL,
            FOREIGN KEY(student_id) REFERENCES students(id)
        )
    """)
    conn.commit()

# ----- STUDENTS -----
def add_student(roll_number, name):
    with closing(get_connection()) as conn:
        conn.execute("INSERT INTO students (roll_number, name) VALUES (?, ?)",
            (roll_number, name))
        conn.commit()

def list_students():
    with closing(get_connection()) as conn:
        rows = conn.execute("SELECT id, roll_number, name FROM students ORDER BY
            roll_number").fetchall()
        return [dict(r) for r in rows]

def get_student_by_roll(roll_number):
    with closing(get_connection()) as conn:
        row = conn.execute("SELECT id, roll_number, name FROM students WHERE
            roll_number = ?", (roll_number,)).fetchone()
        return dict(row) if row else None

# ----- GRADES -----
def add_grade(roll_number, subject, grade):
    with closing(get_connection()) as conn:
        stu = conn.execute("SELECT id FROM students WHERE roll_number = ?",
            (roll_number,)).fetchone()
        if not stu:
            raise ValueError("Student not found")
        conn.execute(
            "INSERT INTO grades (student_id, subject, grade) VALUES (?, ?, ?)",
            (stu["id"], subject, grade)
        )
        conn.commit()

```

```

def get_grades_by_roll(roll_number):
    with closing(get_connection()) as conn:
        stu = conn.execute("SELECT id FROM students WHERE roll_number = ?",
(roll_number,)).fetchone()
        if not stu:
            return []
        rows = conn.execute("SELECT subject, grade FROM grades WHERE student_id =
? ORDER BY subject", (stu["id"],)).fetchall()
        return [dict(r) for r in rows]

# ----- ANALYTICS / REPORTS -----
def get_full_report():
    with closing(get_connection()) as conn:
        rows = conn.execute("""
            SELECT s.roll_number, s.name, g.subject, g.grade
            FROM students s
            LEFT JOIN grades g ON s.id = g.student_id
            ORDER BY s.roll_number, g.subject
            """).fetchall()
        return [dict(r) for r in rows]

def get_subject_toppers():
    with closing(get_connection()) as conn:
        rows = conn.execute("""
            SELECT g.subject, s.name, s.roll_number, MAX(g.grade) AS max_grade
            FROM grades g
            JOIN students s ON g.student_id = s.id
            GROUP BY g.subject
            ORDER BY g.subject
            """).fetchall()
        return [dict(r) for r in rows]

def get_class_average():
    with closing(get_connection()) as conn:
        rows = conn.execute("""
            SELECT subject, ROUND(AVG(grade), 2) AS avg_grade, COUNT(*) AS
total_students
            FROM grades
            GROUP BY subject
            ORDER BY subject
            """).fetchall()
        return [dict(r) for r in rows]

# ----- STATS (for dashboard cards) -----
def total_students():

```

```

    with closing(get_connection()) as conn:
        row = conn.execute("SELECT COUNT(*) AS c FROM students").fetchone()
        return row["c"]

def total_subjects():
    with closing(get_connection()) as conn:
        row = conn.execute("SELECT COUNT(DISTINCT subject) AS c FROM
grades").fetchone()
        return row["c"]

def total_grades():
    with closing(get_connection()) as conn:
        row = conn.execute("SELECT COUNT(*) AS c FROM grades").fetchone()
        return row["c"]

```

models.py

```

import database

class StudentTracker:
    def __init__(self):
        database.init_db()

    def add_student(self, roll, name):
        database.add_student(roll, name)

    def list_students(self):
        return database.list_students()

    def add_grade(self, roll, subject, grade):
        database.add_grade(roll, subject, grade)

    def get_student_details(self, roll):
        return database.get_student_details(roll)

    def class_average(self):
        return database.get_class_averages()

    def subject_topper(self):
        return database.get_subject_toppers()

```


profile

web: gunicorn app:app

README.md

```
# student-performance-tracker
```

Requirements.txt

```
Flask==3.1.2  
gunicorn==21.2.0
```

tempCodeRunn

```
from flask import Flask, render_template, request, redirect, url_for, flash  
import database  
  
app = Flask(__name__)  
app.secret_key = "supersecret"  
  
@app.route("/")  
def index():  
    students = database.list_students()  
    stats = {  
        "total_students": database.total_students(),  
        "total_subjects": database.total_subjects(),  
        "total_grades": database.total_grades(),  
    }  
    return render_template("index.html", students=students, stats=stats)  
  
# ----- Add Student -----  
@app.route("/add-student", methods=["GET", "POST"])  
def add_student():  
    if request.method == "POST":  
        roll = request.form.get("roll", "").strip()  
        name = request.form.get("name", "").strip()  
        try:  
            if not roll or not name:  
                raise ValueError("Roll number and name are required.")  
            database.add_student(roll, name)  
            flash("👤 Student added successfully!", "success")  
            return redirect(url_for("index"))
```

```

        except Exception as e:
            flash(f"❌ {e}", "error")
            return redirect(url_for("add_student"))
        return render_template("add_student.html")

# ----- Add Grade -----
@app.route("/add-grade", methods=["GET", "POST"])
def add_grade():
    if request.method == "POST":
        roll = request.form.get("roll", "").strip()
        subject = request.form.get("subject", "").strip()
        grade_raw = request.form.get("grade", "").strip()
        try:
            if not roll or not subject or not grade_raw:
                raise ValueError("All fields are required.")
            grade = int(grade_raw)
            if grade < 0 or grade > 100:
                raise ValueError("Grade must be between 0 and 100.")
            database.add_grade(roll, subject, grade)
            flash("✅ Grade added successfully!", "success")
            return redirect(url_for("index"))
        except Exception as e:
            flash(f"❌ {e}", "error")
            return redirect(url_for("add_grade"))
    return render_template("add_grade.html")

# ----- View Student -----
@app.route("/student/<roll>")
def view_student(roll):
    student = database.get_student_by_roll(roll)
    grades = database.get_grades_by_roll(roll)
    if not student:
        flash("❌ Student not found.", "error")
        return redirect(url_for("index"))
    return render_template("view_student.html", student=student, grades=grades)

# ----- Reports -----
@app.route("/reports")
def reports():
    report_data = database.get_full_report()
    return render_template("reports.html", report_data=report_data)

# ----- Subject Toppers -----
@app.route("/subject-topper")
def subject_topper():

```



```

                <label class="form-label fw-bold">Roll Number <span
class="text-danger">*</span></label>
                <div class="input-group">
                    <span class="input-group-text bg-light"><i
class="fas fa-hashtag text-success"></i></span>
                    <input type="text" class="form-control"
name="roll" id="rollNumber" required
placeholder="Enter student roll number"
list="rollSuggestions">
                </div>
                <datalist id="rollSuggestions">
                    {% for student in students %}
                    <option value="{ student.roll_number }">
                    {% endfor %}
                </datalist>
            </div>

            <div class="col-md-4 mb-3">
                <label class="form-label fw-bold">Subject <span
class="text-danger">*</span></label>
                <div class="input-group">
                    <span class="input-group-text bg-light"><i
class="fas fa-book text-success"></i></span>
                    <select class="form-select" name="subject"
required>
                        <option value="">Select Subject</option>
                        <option
value="Mathematics">Mathematics</option>
                        <option value="Physics">Physics</option>
                        <option value="Chemistry">Chemistry</option>
                        <option value="Biology">Biology</option>
                        <option value="English">English</option>
                        <option value="Computer Science">Computer
Science</option>
                        <option value="other">Other Subject</option>
                    </select>
                </div>
                <input type="text" class="form-control mt-2"
name="subject_custom" id="subjectCustom"
placeholder="Enter custom subject"
style="display: none;">
            </div>

            <div class="col-md-4 mb-3">

```

```

        <label class="form-label fw-bold">Grade <span
class="text-danger">*</span></label>
        <div class="input-group">
            <span class="input-group-text bg-light"><i
class="fas fa-percent text-success"></i></span>
            <input type="number" class="form-control"
name="grade" id="gradeInput"
min="0" max="100" step="0.01" required
placeholder="0-100">
        </div>
        <div class="form-text">Enter percentage (0 to
100)</div>
    </div>
</div>

<div class="row">
    <div class="col-md-4 mb-3">
        <label class="form-label fw-bold">Semester</label>
        <div class="input-group">
            <span class="input-group-text bg-light"><i
class="fas fa-calendar text-success"></i></span>
            <select class="form-select" name="semester">
                <option value="">Select Semester</option>
                <option value="1">Semester 1</option>
                <option value="2">Semester 2</option>
                <option value="3">Semester 3</option>
                <option value="4">Semester 4</option>
                <option value="5">Semester 5</option>
                <option value="6">Semester 6</option>
            </select>
        </div>
    </div>
</div>

<div class="col-md-4 mb-3">
    <label class="form-label fw-bold">Exam Type</label>
    <div class="input-group">
        <span class="input-group-text bg-light"><i
class="fas fa-file-alt text-success"></i></span>
        <select class="form-select" name="exam_type">
            <option value="">Select Exam Type</option>
            <option value="Midterm">Midterm</option>
            <option value="Final">Final</option>
            <option value="Quiz">Quiz</option>
            <option
value="Assignment">Assignment</option>

```

```

        <option value="Project">Project</option>
    </select>
</div>
</div>

<div class="col-md-4 mb-3">
    <label class="form-label fw-bold">Grade Date</label>
    <div class="input-group">
        <span class="input-group-text bg-light"><i
class="fas fa-calendar-day text-success"></i></span>
        <input type="date" class="form-control"
name="grade_date"
            value="{{ today }}">
    </div>
</div>
</div>

<div class="mb-3">
    <label class="form-label fw-bold">Remarks</label>
    <textarea class="form-control" name="remarks" rows="2"
        placeholder="Any additional remarks about this
grade..."></textarea>
</div>

<!-- Grade Preview -->
<div class="card bg-light mt-4" id="gradePreview"
style="display: none;">
    <div class="card-body">
        <h6 class="card-title"><i class="fas fa-eye me-
2"></i>Grade Preview</h6>
        <div class="row">
            <div class="col-md-6">
                <p class="mb-1"><strong>Student:</strong>
<span id="previewRoll">-</span></p>
                <p class="mb-1"><strong>Subject:</strong>
<span id="previewSubject">-</span></p>
                <p class="mb-1"><strong>Grade:</strong> <span
id="previewGrade">-</span></p>
            </div>
            <div class="col-md-6">
                <p class="mb-1"><strong>Status:</strong>
<span id="previewStatus" class="badge">-</span></p>
                <p class="mb-1"><strong>Grade Point:</strong>
<span id="previewPoint">-</span></p>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>

<div class="d-flex gap-2 mt-4">
    <button type="submit" class="btn btn-success px-4 py-2">
        <i class="fas fa-save me-2"></i> Save Grade
    </button>
    <button type="reset" class="btn btn-outline-secondary px-
4 py-2">
        <i class="fas fa-redo me-2"></i> Reset Form
    </button>
</div>
</form>
</div>
</div>

<!-- Quick Stats -->
<div class="row mt-4">
    <div class="col-md-3">
        <div class="card bg-light border-0 text-center">
            <div class="card-body">
                <i class="fas fa-user-graduate fa-2x text-success mb-
2"></i>
                <h4 class="mb-0">{{ stats.total_students if stats else 0
}}</h4>
                <small class="text-muted">Total Students</small>
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="card bg-light border-0 text-center">
            <div class="card-body">
                <i class="fas fa-book-open fa-2x text-info mb-2"></i>
                <h4 class="mb-0">{{ stats.total_subjects if stats else 0
}}</h4>
                <small class="text-muted">Subjects</small>
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="card bg-light border-0 text-center">
            <div class="card-body">
                <i class="fas fa-chart-bar fa-2x text-warning mb-2"></i>

```

```

        <h4 class="mb-0">{{ stats.total_grades if stats else 0
    }}</h4>

        <small class="text-muted">Grades Recorded</small>
    </div>
</div>
<div class="col-md-3">
    <div class="card bg-light border-0 text-center">
        <div class="card-body">
            <i class="fas fa-percentage fa-2x text-primary mb-2"></i>
            <h4 class="mb-0">85.2%</h4>
            <small class="text-muted">Class Average</small>
        </div>
    </div>
</div>
</div>
</div>
</div>
<!-- JavaScript for enhanced functionality -->
<script>
    document.addEventListener('DOMContentLoaded', function() {
        // Show/hide custom subject input
        const subjectSelect = document.querySelector('select[name="subject"]');
        const subjectCustom = document.getElementById('subjectCustom');

        if (subjectSelect) {
            subjectSelect.addEventListener('change', function() {
                subjectCustom.style.display = this.value === 'other' ? 'block' :
'none';

                if (this.value !== 'other') subjectCustom.value = '';
                updateGradePreview();
            });
        }

        // Grade input validation and preview
        const gradeInput = document.getElementById('gradeInput');
        if (gradeInput) {
            gradeInput.addEventListener('input', function() {
                if (this.value > 100) this.value = 100;
                if (this.value < 0) this.value = 0;
                updateGradePreview();
            });
        }
    })

```



```

// Roll number input for preview
const rollInput = document.getElementById('rollNumber');
if (rollInput) {
    rollInput.addEventListener('input', updateGradePreview);
}

// Update grade preview
function updateGradePreview() {
    const preview = document.getElementById('gradePreview');
    const grade = parseFloat(gradeInput?.value) || 0;
    const subject = subjectSelect?.value === 'other' ?
subjectCustom.value : subjectSelect?.value;
    const roll = rollInput?.value || '';

    if (grade > 0 && subject && roll) {
        preview.style.display = 'block';
        document.getElementById('previewRoll').textContent = roll;
        document.getElementById('previewSubject').textContent = subject
|| '-';

        document.getElementById('previewGrade').textContent = grade +
'%';

        // Calculate grade status
        let status = '';
        let statusClass = '';
        if (grade >= 90) {
            status = 'Excellent';
            statusClass = 'bg-success';
        } else if (grade >= 75) {
            status = 'Good';
            statusClass = 'bg-info';
        } else if (grade >= 60) {
            status = 'Average';
            statusClass = 'bg-warning';
        } else if (grade >= 40) {
            status = 'Pass';
            statusClass = 'bg-primary';
        } else {
            status = 'Fail';
            statusClass = 'bg-danger';
        }

        document.getElementById('previewStatus').textContent = status;
        document.getElementById('previewStatus').className = 'badge ' +
statusClass;

```

```

        // Calculate grade point (simple example)
        const gradePoint = (grade / 10).toFixed(1);
        document.getElementById('previewPoint').textContent = gradePoint;
    } else {
        preview.style.display = 'none';
    }
}

// Form validation
const form = document.getElementById('addGradeForm');
if (form) {
    form.addEventListener('submit', function(e) {
        const grade = parseFloat(gradeInput.value);
        if (grade < 0 || grade > 100) {
            e.preventDefault();
            showError('Please enter a valid grade between 0 and 100!');
            return;
        }
        showSuccess('Grade added successfully!');
    });
}

function showError(message) {
    Swal.fire({
        icon: 'error',
        title: 'Error',
        text: message,
        timer: 2000
    });
}

function showSuccess(message) {
    Swal.fire({
        icon: 'success',
        title: 'Success',
        text: message,
        timer: 2000
    });
}
});
</script>

<style>
.card {

```

```

        border-radius: 12px;
        transition: transform 0.2s ease-in-out;
    }
    .card:hover {
        transform: translateY(-3px);
    }
    .input-group-text {
        border-radius: 8px 0 0 8px;
    }
    .form-control, .form-select {
        border-radius: 0 8px 8px 0;
        transition: all 0.3s ease;
    }
    .form-control:focus, .form-select:focus {
        box-shadow: 0 0 3px rgba(40, 167, 69, 0.15);
        border-color: #28a745;
    }
    .btn {
        border-radius: 8px;
        font-weight: 500;
        transition: all 0.3s ease;
    }
    .btn-success {
        background: linear-gradient(45deg, #28a745, #20c997);
        border: none;
    }
    .btn-success:hover {
        transform: translateY(-2px);
        box-shadow: 0 4px 8px rgba(40, 167, 69, 0.3);
    }
    #gradePreview {
        border-left: 4px solid #28a745;
    }
</style>
{% endblock %}

```

<>add_student.html

```

{% extends "base.html" %}
{% block content %}
<div class="row justify-content-center">
    <div class="col-lg-8">
        <div class="d-flex justify-content-between align-items-center mb-4">

```

```

        <h3 class="mb-0"><i class="fas fa-user-plus me-2 text-
primary"></i>Add New Student</h3>
        <a href="/" class="btn btn-outline-secondary">
            <i class="fas fa-arrow-left me-1"></i> Back to Dashboard
        </a>
    </div>

    <div class="card shadow-lg border-0">
        <div class="card-header bg-primary text-white py-3">
            <h5 class="mb-0"><i class="fas fa-user-graduate me-2"></i>Student
Information</h5>
        </div>
        <div class="card-body p-4">
            <form method="post" id="addStudentForm">
                <div class="row">
                    <div class="col-md-6 mb-3">
                        <label class="form-label fw-bold">Roll Number <span
class="text-danger">*</span></label>
                        <div class="input-group">
                            <span class="input-group-text bg-light"><i
class="fas fa-hashtag text-primary"></i></span>
                            <input type="text" class="form-control"
name="roll" id="rollNumber" required
                                placeholder="Enter unique roll number"
onblur="checkRollNumber()">
                        </div>
                        <div class="form-text">Must be a unique
identifier</div>
                        <div id="rollNumberFeedback" class="invalid-
feedback"></div>
                    </div>

                    <div class="col-md-6 mb-3">
                        <label class="form-label fw-bold">Full Name <span
class="text-danger">*</span></label>
                        <div class="input-group">
                            <span class="input-group-text bg-light"><i
class="fas fa-user text-primary"></i></span>
                            <input type="text" class="form-control"
name="name" required
                                placeholder="Enter full name">
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>

```

```

        <div class="row">
            <div class="col-md-6 mb-3">
                <label class="form-label fw-bold">Class</label>
                <div class="input-group">
                    <span class="input-group-text bg-light"><i
class="fas fa-graduation-cap text-primary"></i></span>
                    <select class="form-select" name="class"
id="classSelect">
                        <option value="">Select Class</option>
                        <option value="X">X</option>
                        <option value="XI">XI</option>
                        <option value="XII">XII</option>
                        <option value="other">Other</option>
                    </select>
                </div>
                <input type="text" class="form-control mt-2"
name="class_custom" id="classCustom"
                    placeholder="Or enter custom class"
style="display: none;">
            </div>

            <div class="col-md-6 mb-3">
                <label class="form-label fw-bold">Section</label>
                <div class="input-group">
                    <span class="input-group-text bg-light"><i
class="fas fa-users text-primary"></i></span>
                    <select class="form-select" name="section"
id="sectionSelect">
                        <option value="">Select Section</option>
                        <option value="A">A</option>
                        <option value="B">B</option>
                        <option value="C">C</option>
                        <option value="D">D</option>
                        <option value="other">Other</option>
                    </select>
                </div>
                <input type="text" class="form-control mt-2"
name="section_custom" id="sectionCustom"
                    placeholder="Or enter custom section"
style="display: none;">
            </div>
        </div>

        <div class="mb-3">
            <label class="form-label fw-bold">Email Address</label>

```

```

        <div class="input-group">
            <span class="input-group-text bg-light"><i class="fas
fa-envelope text-primary"></i></span>
            <input type="email" class="form-control" name="email"
placeholder="student@example.com">
        </div>
    </div>

    <div class="mb-3">
        <label class="form-label fw-bold">Additional
Notes</label>
        <textarea class="form-control" name="notes" rows="3"
placeholder="Any additional information about
the student..."></textarea>
    </div>

    <div class="d-flex gap-2 mt-4">
        <button type="submit" class="btn btn-primary px-4 py-2">
            <i class="fas fa-save me-2"></i> Save Student
        </button>
        <button type="reset" class="btn btn-outline-secondary px-
4 py-2">
            <i class="fas fa-redo me-2"></i> Reset
        </button>
    </div>
</form>
</div>
</div>
</div>
</div>

<!-- JavaScript for enhanced functionality -->
<script>
    // Form validation
    document.getElementById('addStudentForm').addEventListener('submit',
function(e) {
        const rollNumber = document.getElementById('rollNumber').value;
        if (!rollNumber.trim()) {
            e.preventDefault();
            showError('Roll number is required!');
            return;
        }

        // You can add more validations here
        showSuccess('Student added successfully!');
    }

```

```

    });

    // Show/hide custom class input
    document.getElementById('classSelect').addEventListener('change', function()
{
    const customInput = document.getElementById('classCustom');
    customInput.style.display = this.value === 'other' ? 'block' : 'none';
    if (this.value !== 'other') customInput.value = '';
});

    // Show/hide custom section input
    document.getElementById('sectionSelect').addEventListener('change',
function() {
    const customInput = document.getElementById('sectionCustom');
    customInput.style.display = this.value === 'other' ? 'block' : 'none';
    if (this.value !== 'other') customInput.value = '';
});

    function checkRollNumber() {
        const rollInput = document.getElementById('rollNumber');
        const feedback = document.getElementById('rollNumberFeedback');

        if (!rollInput.value.trim()) {
            rollInput.classList.remove('is-invalid');
            feedback.textContent = '';
            return;
        }

        // Simulate checking if roll number exists (you can implement actual AJAX
check)
        if (Math.random() > 0.7) { // Simulate 30% chance of duplicate
            rollInput.classList.add('is-invalid');
            feedback.textContent = 'This roll number already exists!';
        } else {
            rollInput.classList.remove('is-invalid');
            feedback.textContent = '';
        }
    }

    function showError(message) {
        Swal.fire({
            icon: 'error',
            title: 'Error',
            text: message,
            timer: 2000

```

```

    });
}

function showSuccess(message) {
    Swal.fire({
        icon: 'success',
        title: 'Success',
        text: message,
        timer: 2000
    });
}

// Add animation to form elements
document.addEventListener('DOMContentLoaded', function() {
    const formElements = document.querySelectorAll('.form-control, .form-select');
    formElements.forEach(element => {
        element.addEventListener('focus', function() {
            this.parentElement.classList.add('focused');
        });
        element.addEventListener('blur', function() {
            this.parentElement.classList.remove('focused');
        });
    });
});
</script>

<style>
    .card {
        border-radius: 12px;
        transition: transform 0.2s ease-in-out;
    }
    .card:hover {
        transform: translateY(-3px);
    }
    .input-group-text {
        border-radius: 8px 0 0 8px;
    }
    .form-control, .form-select {
        border-radius: 0 8px 8px 0;
        transition: all 0.3s ease;
    }
    .form-control:focus, .form-select:focus {
        box-shadow: 0 0 0 3px rgba(67, 97, 238, 0.15);
        border-color: #4361ee;
    }

```



```

}
.input-group.focused {
  box-shadow: 0 0 0 3px rgba(67, 97, 238, 0.15);
  border-radius: 8px;
}
.btn {
  border-radius: 8px;
  font-weight: 500;
  transition: all 0.3s ease;
}
.btn-primary {
  background: linear-gradient(45deg, #4361ee, #3a0ca3);
  border: none;
}
.btn-primary:hover {
  transform: translateY(-2px);
  box-shadow: 0 4px 8px rgba(67, 97, 238, 0.3);
}
</style>
{% endblock %}

```

<>base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Student Performance Tracker</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
  <style>
    :root {
      --primary: #4361ee;
      --secondary: #3a0ca3;
      --success: #4cc9f0;
      --light-bg: #f8f9fa;
      --dark-bg: #121212;
      --dark-card: #1e1e1e;
      --dark-text: #e0e0e0;
    }
  </style>

```

```

    --shadow: 0 4px 12px rgba(0,0,0,0.08);
    --transition: all 0.3s ease;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    transition: var(--transition);
    background-color: var(--light-bg);
    min-height: 100vh;
    display: flex;
    flex-direction: column;
}

body.dark-mode {
    background-color: var(--dark-bg);
    color: var(--dark-text);
}

/* Sidebar */
.sidebar {
    width: 260px;
    height: 100vh;
    position: fixed;
    top: 0;
    left: 0;
    background: linear-gradient(180deg, var(--primary) 0%, var(--
secondary) 100%);
    color: white;
    padding: 0;
    z-index: 1000;
    box-shadow: var(--shadow);
    transition: var(--transition);
}

.sidebar-header {
    padding: 20px;
    border-bottom: 1px solid rgba(255,255,255,0.1);
}

.sidebar-header h3 {
    margin: 0;
    font-weight: 700;
    display: flex;
    align-items: center;
    gap: 10px;
}

```

```
}

.sidebar-header h3 i {
  font-size: 1.5rem;
}

.nav-menu {
  padding: 20px 0;
}

.nav-item {
  padding: 12px 20px;
  margin: 5px 0;
  display: flex;
  align-items: center;
  gap: 12px;
  transition: var(--transition);
  color: rgba(255,255,255,0.9);
  text-decoration: none;
  font-weight: 500;
}

.nav-item:hover, .nav-item.active {
  background-color: rgba(255,255,255,0.1);
  color: white;
  border-left: 4px solid white;
  padding-left: 16px;
}

.nav-item i {
  width: 20px;
  text-align: center;
}

/* Topbar */
.topbar {
  margin-left: 260px;
  padding: 15px 30px;
  background: white;
  display: flex;
  justify-content: space-between;
  align-items: center;
  box-shadow: var(--shadow);
  transition: var(--transition);
  z-index: 999;
}
```

```
}

body.dark-mode .topbar {
  background-color: var(--dark-card);
}

.search-container {
  width: 50%;
  position: relative;
}

.search-container i {
  position: absolute;
  left: 15px;
  top: 50%;
  transform: translateY(-50%);
  color: #6c757d;
}

.search-container input {
  padding-left: 40px;
  border-radius: 50px;
}

.theme-toggle {
  background: none;
  border: none;
  font-size: 1.2rem;
  cursor: pointer;
  color: #6c757d;
  transition: var(--transition);
}

.theme-toggle:hover {
  color: var(--primary);
}

/* Content */
.content {
  margin-left: 260px;
  padding: 30px;
  flex: 1;
  transition: var(--transition);
}
```

```
.card {
  border: none;
  border-radius: 12px;
  box-shadow: var(--shadow);
  transition: var(--transition);
  margin-bottom: 20px;
}

body.dark-mode .card {
  background-color: var(--dark-card);
}

.card:hover {
  transform: translateY(-5px);
  box-shadow: 0 8px 24px rgba(0,0,0,0.12);
}

/* Footer */
footer {
  margin-left: 260px;
  text-align: center;
  padding: 20px;
  background: white;
  transition: var(--transition);
}

body.dark-mode footer {
  background-color: var(--dark-card);
}

/* Responsive */
@media (max-width: 992px) {
  .sidebar {
    width: 70px;
    overflow: hidden;
  }

  .sidebar-header h3 span, .nav-item span {
    display: none;
  }

  .sidebar-header h3 i {
    font-size: 1.8rem;
  }
}
```

```
.topbar, .content, footer {
  margin-left: 70px;
}

.nav-item {
  justify-content: center;
  padding: 15px;
}

.nav-item i {
  font-size: 1.2rem;
}
}

@media (max-width: 768px) {
  .sidebar {
    width: 100%;
    height: auto;
    position: relative;
    padding: 10px;
  }

  .sidebar-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
  }

  .nav-menu {
    display: flex;
    overflow-x: auto;
    padding: 10px 0;
  }

  .nav-item {
    flex-direction: column;
    padding: 10px;
    font-size: 0.8rem;
    border-left: none;
    border-bottom: 4px solid transparent;
  }

  .nav-item:hover, .nav-item.active {
    border-left: none;
    border-bottom: 4px solid white;
  }
}
```

```

        padding-left: 10px;
    }

    .topbar, .content, footer {
        margin-left: 0;
    }

    .search-container {
        width: 100%;
        margin-right: 15px;
    }
}
</style>
</head>
<body>
    <!-- Sidebar -->
    <div class="sidebar">
        <div class="sidebar-header">
            <h3><i class="fas fa-graduation-cap"></i> <span>Student Performance
Tracker</span></h3>
        </div>
        <div class="nav-menu">
            <a href="/" class="nav-item active">
                <i class="fas fa-home"></i> <span>Dashboard</span>
            </a>
            <a href="/add-student" class="nav-item">
                <i class="fas fa-user-plus"></i> <span>Add Student</span>
            </a>
            <a href="/add-grade" class="nav-item">
                <i class="fas fa-chart-line"></i> <span>Add Grade</span>
            </a>
            <a href="/subject-topper" class="nav-item">
                <i class="fas fa-trophy"></i> <span>Subject Topper</span>
            </a>
            <a href="/class-average" class="nav-item">
                <i class="fas fa-calculator"></i> <span>Class Average</span>
            </a>
            <a href="/reports" class="nav-item">
                <i class="fas fa-file-alt"></i> <span>Reports</span>
            </a>
        </div>
    </div>

    <!-- Topbar -->
    <div class="topbar">

```

```

        <div class="search-container">
            <i class="fas fa-search"></i>
            <input class="form-control" type="search" placeholder="Search
students, grades..." aria-label="Search">
        </div>
        <button id="themeToggle" class="theme-toggle">
            <i class="fas fa-moon"></i>
        </button>
    </div>

    <!-- Main Content -->
    <div class="content">
        {% block content %}{% endblock %}
    </div>

    <!-- Footer -->
    <footer>
        <p>&copy; 2025 Student Performance Tracker | <a href="#about">About</a> | 
<a href="#privacy">Privacy Policy</a></p>
    </footer>

    <!-- Dark mode toggle -->
    <script>
        const themeToggle = document.getElementById("themeToggle");
        const themeIcon = themeToggle.querySelector('i');

        // Check saved theme
        let savedTheme = localStorage.getItem("theme");
        if (savedTheme === "dark") {
            document.body.classList.add("dark-mode");
            themeIcon.classList.replace('fa-moon', 'fa-sun');
        }

        themeToggle.addEventListener("click", () => {
            document.body.classList.toggle("dark-mode");

            if (document.body.classList.contains("dark-mode")) {
                localStorage.setItem("theme", "dark");
                themeIcon.classList.replace('fa-moon', 'fa-sun');
            } else {
                localStorage.setItem("theme", "light");
                themeIcon.classList.replace('fa-sun', 'fa-moon');
            }
        });
    </script>

```



```

<!-- SweetAlert flash messages -->
{% with msgs = get_flashed_messages(with_categories=true) %}
{% if msgs %}
<script>
    const items = JSON.parse('{{ msgs | tojson | safe }}');
    items.forEach(([cat, msg])=>{
        Swal.fire({
            icon: cat === 'success' ? 'success' : (cat === 'error' ? 'error'
: 'info'),
            title: msg,
            timer: 1700,
            showConfirmButton: false,
            background: document.body.classList.contains('dark-mode') ?
'#1e1e1e' : '#fff',
            color: document.body.classList.contains('dark-mode') ? '#e0e0e0'
: '#000'
        });
    });
</script>
{% endif %}
{% endwith %}
</body>
</html>

```

<>class_average_chart.html

```

{% extends "base.html" %}
{% block content %}

<div class="container mt-4 text-center">
    <h1 class="mb-4">📊 Class Averages (All Subjects)</h1>

    <!-- Toggle Buttons -->
    <div class="btn-group mb-3" role="group">
        <button class="btn btn-primary" onclick="renderChart('bar')">Bar</button>
        <button class="btn btn-success" onclick="renderChart('pie')">Pie</button>
        <button class="btn btn-info" onclick="renderChart('line')">Line</button>
    </div>

    <!-- Chart Container -->
    <div id="chartBox" class="p-3 bg-light rounded shadow-sm mx-auto">
        <canvas id="avgChart"></canvas>
    </div>

```

```

</div>

<!-- Chart.js include -->
<script
src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>

<script>
  // Flask → JS data
  const subjects = JSON.parse('{{ subjects | tojson | safe }}');
  const values   = JSON.parse('{{ values   | tojson | safe }}');
  const totals   = JSON.parse('{{ totals   | tojson | safe }}');

  const ctx = document.getElementById('avgChart').getContext('2d');
  let chart;

  // Function to render chart with selected type
  function renderChart(type = 'bar') {
    if (chart) chart.destroy(); // destroy old chart before new render
    chart = new Chart(ctx, {
      type: type,
      data: {
        labels: subjects,
        datasets: [{
          label: 'Average Marks',
          data: values,
          backgroundColor: [
            'rgba(72, 201, 176, 0.8)',
            'rgba(255, 159, 64, 0.8)',
            'rgba(153, 102, 255, 0.8)',
            'rgba(255, 99, 132, 0.8)',
            'rgba(255, 205, 86, 0.8)',
            'rgba(54, 162, 235, 0.8)',
            'rgba(201, 203, 207, 0.8)'
          ],
          borderColor: [
            'rgba(72, 201, 176, 1)',
            'rgba(255, 159, 64, 1)',
            'rgba(153, 102, 255, 1)',
            'rgba(255, 99, 132, 1)',
            'rgba(255, 205, 86, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(201, 203, 207, 1)'
          ],
          borderWidth: 1
        }]
      }
    });
  }

```

```

    },
    options: {
      responsive: true,
      maintainAspectRatio: false, // 👉 parent div ki height follow karega
      plugins: {
        tooltip: {
          callbacks: {
            label: function(context) {
              const i = context.dataIndex;
              return `Avg: ${values[i]} | Students: ${totals[i]}`;
            }
          }
        }
      },
      scales: type === 'bar' || type === 'line' ? {
        y: { beginAtZero: true, max: 100, title: { display: true, text:
'Average Marks' } },
        x: { title: { display: true, text: 'Subjects' } }
      } : {}
    }
  });
}

// Load default chart
renderChart('bar');
</script>

<style>
  /* 👉 Fixed chart size */
  #chartBox {
    width: 500px;
    height: 350px;
  }
</style>

{% endblock %}

```

<>index.html

```

{% extends "base.html" %}
{% block content %}
<div class="d-flex justify-content-between align-items-center mb-4">

```

```

<h3 class="mb-0">Dashboard Overview</h3>
<div class="d-flex gap-2">
  <button class="btn btn-outline-primary btn-sm" onclick="exportReport()">
    <i class="fas fa-download me-1"></i> Export Report
  </button>
</div>
</div>

<!-- Stats Cards -->
<div class="row g-4 mb-4">
  <div class="col-xl-3 col-md-6">
    <div class="card stat-card bg-primary-gradient text-white">
      <div class="card-body">
        <div class="d-flex align-items-center">
          <div class="flex-grow-1">
            <h6 class="card-title text-white-50 mb-1">Total
Students</h6>
            <h2 class="mb-0">{{ stats.total_students }}</h2>
            <p class="mb-0 small mt-2"><i class="fas fa-arrow-up me-
1"></i> 12% from last month</p>
          </div>
          <div class="flex-shrink-0">
            <div class="stat-icon">
              <i class="fas fa-user-graduate fa-2x"></i>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-xl-3 col-md-6">
    <div class="card stat-card bg-success-gradient text-white">
      <div class="card-body">
        <div class="d-flex align-items-center">
          <div class="flex-grow-1">
            <h6 class="card-title text-white-50 mb-1">Subjects</h6>
            <h2 class="mb-0">{{ stats.total_subjects }}</h2>
            <p class="mb-0 small mt-2"><i class="fas fa-book me-
1"></i> All categories</p>
          </div>
          <div class="flex-shrink-0">
            <div class="stat-icon">
              <i class="fas fa-book-open fa-2x"></i>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
<div class="col-xl-3 col-md-6">
    <div class="card stat-card bg-warning-gradient text-white">
        <div class="card-body">
            <div class="d-flex align-items-center">
                <div class="flex-grow-1">
                    <h6 class="card-title text-white-50 mb-1">Grades
Entries</h6>
                    <h2 class="mb-0">{{ stats.total_grades }}</h2>
                    <p class="mb-0 small mt-2"><i class="fas fa-chart-line
me-1"></i> 24 new today</p>
                </div>
                <div class="flex-shrink-0">
                    <div class="stat-icon">
                        <i class="fas fa-chart-bar fa-2x"></i>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="col-xl-3 col-md-6">
    <div class="card stat-card bg-info-gradient text-white">
        <div class="card-body">
            <div class="d-flex align-items-center">
                <div class="flex-grow-1">
                    <h6 class="card-title text-white-50 mb-1">Avg
Performance</h6>
                    <h2 class="mb-0">78.5%</h2>
                    <p class="mb-0 small mt-2"><i class="fas fa-trophy me-
1"></i> Top: Physics (85%)</p>
                </div>
                <div class="flex-shrink-0">
                    <div class="stat-icon">
                        <i class="fas fa-trophy fa-2x"></i>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

```

<div class="row g-4">
  <!-- Students Table -->
  <div class="col-lg-8">
    <div class="card shadow-sm">
      <div class="card-header bg-white py-3">
        <h5 class="mb-0">Student Records</h5>
      </div>
      <div class="card-body p-0">
        <div class="table-responsive">
          <table class="table table-hover mb-0">
            <thead class="table-light">
              <tr>
                <th>Roll No</th>
                <th>Name</th>
                <th>Class</th>
                <th>Performance</th>
                <th>Actions</th>
              </tr>
            </thead>
            <tbody>
              {% for s in students %}
              <tr>
                <td><span class="badge bg-primary">{{
s.roll_number }}</span></td>
                <td>
                  <div class="d-flex align-items-center">
                    <div class="avatar-sm bg-light rounded-
circle me-2 d-flex align-items-center justify-content-center">
                      <i class="fas fa-user text-
primary"></i>
                    </div>
                    <div>{{ s.name }}</div>
                  </div>
                </td>
                <td>XII-A</td>
                <td>
                  <div class="progress" style="height: 6px;
width: 80px;">
                    <div class="progress-bar bg-success"
role="progressbar" style="width: 75%;"></div>
                  </div>
                </td>
                <td>
                  <div class="btn-group btn-group-sm">

```

```

                                <a class="btn btn-outline-primary"
href="{{ url_for('view_student', roll=s.roll_number) }}">
                                <i class="fas fa-eye"></i>
                                </a>
                                <!-- Removed the problematic buttons -->
                                <button class="btn btn-outline-success"
onclick="editStudent('{{ s.roll_number }}')">
                                <i class="fas fa-edit"></i>
                                </button>
                                </div>
                                </td>
                                </tr>
                                {% else %}
                                <tr>
                                <td colspan="5" class="text-center py-5">
                                <div class="py-4">
                                <i class="fas fa-user-graduate fa-3x
text-muted mb-3"></i>
                                <h5 class="text-muted">No students
found</h5>
                                <p class="text-muted mb-0">Get started by
adding your first student</p>
                                <a href="/add-student" class="btn btn-
primary mt-3">
                                <i class="fas fa-plus me-1"></i> Add
Student
                                </a>
                                </div>
                                </td>
                                </tr>
                                {% endfor %}
                                </tbody>
                                </table>
                                </div>
                                </div>
                                {% if students %}
                                <div class="card-footer bg-white py-3">
                                <div class="d-flex justify-content-between align-items-center">
                                <div>Showing {{ students|length }} of {{ stats.total_students
}} students</div>
                                <button class="btn btn-sm btn-outline-primary"
onclick="viewAllStudents()">View All <i class="fas fa-arrow-right ms-
1"></i></button>
                                </div>
                                </div>

```

```

        {% endif %}
    </div>
</div>

<!-- Performance Chart & Quick Actions -->
<div class="col-lg-4">
    <!-- Performance Chart -->
    <div class="card shadow-sm mb-4">
        <div class="card-header bg-white py-3">
            <h5 class="mb-0">Class Performance</h5>
        </div>
        <div class="card-body">
            <canvas id="performanceChart" height="250"></canvas>
        </div>
    </div>

    <!-- Quick Actions -->
    <div class="card shadow-sm">
        <div class="card-header bg-white py-3">
            <h5 class="mb-0">Quick Actions</h5>
        </div>
        <div class="card-body">
            <div class="d-grid gap-2">
                <a href="/add-student" class="btn btn-outline-primary text-
start">
                    <i class="fas fa-user-plus me-2"></i> Add New Student
                </a>
                <a href="/add-grade" class="btn btn-outline-success text-
start">
                    <i class="fas fa-chart-line me-2"></i> Record Grades
                </a>
                <a href="/subject-topper" class="btn btn-outline-warning
text-start">
                    <i class="fas fa-trophy me-2"></i> View Toppers
                </a>
                <a href="/reports" class="btn btn-outline-info text-start">
                    <i class="fas fa-file-pdf me-2"></i> Generate Reports
                </a>
            </div>
        </div>
    </div>
</div>

<!-- Chart.js Script -->

```



```

<script>
  document.addEventListener('DOMContentLoaded', function() {
    var ctx = document.getElementById('performanceChart').getContext('2d');
    var performanceChart = new Chart(ctx, {
      type: 'doughnut',
      data: {
        labels: ['Excellent (A)', 'Good (B)', 'Average (C)', 'Needs
Improvement'],
        datasets: [{
          data: [35, 25, 30, 10],
          backgroundColor: [
            '#4361ee',
            '#4cc9f0',
            '#3a0ca3',
            '#f72585'
          ],
          borderWidth: 0
        }]
      },
      options: {
        responsive: true,
        maintainAspectRatio: false,
        cutout: '70%',
        plugins: {
          legend: {
            position: 'bottom',
            labels: {
              boxWidth: 12,
              padding: 15
            }
          }
        }
      }
    });
  });

  function exportReport() {
    Swal.fire({
      title: 'Exporting Report',
      text: 'Your report is being prepared...',
      icon: 'info',
      timer: 2000,
      showConfirmButton: false
    });
  }

```

```
function editStudent(rollNumber) {
  Swal.fire({
    title: 'Edit Student',
    text: 'Edit functionality for ' + rollNumber,
    icon: 'info',
    confirmButtonText: 'OK'
  });
}

function viewAllStudents() {
  // Redirect to all students page or show all students
  window.location.href = '/';
}
</script>

<style>
  .stat-card {
    border: none;
    border-radius: 12px;
    transition: transform 0.2s;
  }
  .stat-card:hover {
    transform: translateY(-5px);
  }
  .bg-primary-gradient {
    background: linear-gradient(45deg, #4361ee, #3a0ca3);
  }
  .bg-success-gradient {
    background: linear-gradient(45deg, #4cc9f0, #4895ef);
  }
  .bg-warning-gradient {
    background: linear-gradient(45deg, #f72585, #b5179e);
  }
  .bg-info-gradient {
    background: linear-gradient(45deg, #560bad, #7209b7);
  }
  .stat-icon {
    width: 50px;
    height: 50px;
    border-radius: 50%;
    background: rgba(255, 255, 255, 0.2);
    display: flex;
    align-items: center;
    justify-content: center;
  }
```

```

}
.avatar-sm {
  width: 32px;
  height: 32px;
}
.table th {
  border-top: none;
  font-weight: 600;
  text-transform: uppercase;
  font-size: 0.8rem;
  color: #6c757d;
}
</style>
{% endblock %}

```

<>reports.html

```

{% extends "base.html" %}
{% block content %}
<div class="d-flex justify-content-between align-items-center mb-4">
  <h3 class="mb-0"><i class="fas fa-file-alt me-2 text-primary"></i>Student
Reports</h3>
  <div class="d-flex gap-2">
    <button class="btn btn-outline-primary btn-sm" onclick="exportToCSV()">
      <i class="fas fa-download me-1"></i> Export CSV
    </button>
    <button class="btn btn-primary btn-sm" onclick="window.print()">
      <i class="fas fa-print me-1"></i> Print
    </button>
  </div>
</div>

<!-- Filters and Search -->
<div class="card shadow-sm mb-4">
  <div class="card-body">
    <div class="row g-3">
      <div class="col-md-3">
        <label class="form-label fw-bold">Subject Filter</label>
        <select class="form-select" id="subjectFilter">
          <option value="all">All Subjects</option>
          {% for subject in subjects %}
            <option value="{{ subject }}">{{ subject }}</option>
          {% endfor %}
        </select>

```

```

    </div>
    <div class="col-md-3">
        <label class="form-label fw-bold">Grade Range</label>
        <select class="form-select" id="gradeFilter">
            <option value="all">All Grades</option>
            <option value="90-100">90-100% (Excellent)</option>
            <option value="75-89">75-89% (Good)</option>
            <option value="60-74">60-74% (Average)</option>
            <option value="40-59">40-59% (Pass)</option>
            <option value="0-39">0-39% (Fail)</option>
        </select>
    </div>
    <div class="col-md-3">
        <label class="form-label fw-bold">Sort By</label>
        <select class="form-select" id="sortBy">
            <option value="name">Student Name</option>
            <option value="roll">Roll Number</option>
            <option value="subject">Subject</option>
            <option value="grade">Grade (High to Low)</option>
            <option value="grade_asc">Grade (Low to High)</option>
        </select>
    </div>
    <div class="col-md-3">
        <label class="form-label fw-bold">Search</label>
        <div class="input-group">
            <span class="input-group-text"><i class="fas fa-
search"></i></span>
            <input type="text" class="form-control" id="searchInput"
placeholder="Search students...">
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<!-- Stats Overview -->
<div class="row mb-4">
    <div class="col-md-3">
        <div class="card bg-light border-0 text-center">
            <div class="card-body">
                <i class="fas fa-users fa-2x text-primary mb-2"></i>
                <h4 class="mb-0">{{ total_students }}</h4>
                <small class="text-muted">Total Students</small>
            </div>
        </div>
    </div>

```

```

</div>
<div class="col-md-3">
  <div class="card bg-light border-0 text-center">
    <div class="card-body">
      <i class="fas fa-book-open fa-2x text-success mb-2"></i>
      <h4 class="mb-0">{{ total_subjects }}</h4>
      <small class="text-muted">Subjects</small>
    </div>
  </div>
</div>
<div class="col-md-3">
  <div class="card bg-light border-0 text-center">
    <div class="card-body">
      <i class="fas fa-chart-bar fa-2x text-warning mb-2"></i>
      <h4 class="mb-0">{{ total_grades }}</h4>
      <small class="text-muted">Grades Recorded</small>
    </div>
  </div>
</div>
<div class="col-md-3">
  <div class="card bg-light border-0 text-center">
    <div class="card-body">
      <i class="fas fa-percentage fa-2x text-info mb-2"></i>
      <h4 class="mb-0">{{ average_grade }}%</h4>
      <small class="text-muted">Average Grade</small>
    </div>
  </div>
</div>
</div>

<!-- Reports Table -->
<div class="card shadow-sm">
  <div class="card-body p-0">
    <div class="table-responsive">
      <table class="table table-hover mb-0" id="reportsTable">
        <thead class="table-primary">
          <tr>
            <th>Roll No</th>
            <th>Student Name</th>
            <th>Subject</th>
            <th>Grade</th>
            <th>Status</th>
            <th>Actions</th>
          </tr>
        </thead>

```

```

        <tbody>
            {% for r in report_data %}
                <tr class="report-row" data-subject="{{ r.subject if
r.subject else '' }}" data-grade="{{ r.grade if r.grade is not none else 0 }}">
                    <td><span class="badge bg-secondary">{{ r.roll_number
}}</span></td>

                    <td>
                        <div class="d-flex align-items-center">
                            <div class="avatar-sm bg-primary rounded-circle
me-2 d-flex align-items-center justify-content-center">
                                <i class="fas fa-user text-white"></i>
                            </div>
                            <div>{{ r.name }}</div>
                        </div>
                    </td>
                    <td>{{ r.subject if r.subject else "-" }}</td>
                    <td>
                        {% if r.grade is not none %}
                            <span class="fw-bold">{{ r.grade }}%</span>
                        {% else %}
                            <span class="text-muted">-</span>
                        {% endif %}
                    </td>
                    <td>
                        {% if r.grade is not none %}
                            {% if r.grade >= 90 %}
                                <span class="badge bg-success">Excellent</span>
                            {% elif r.grade >= 75 %}
                                <span class="badge bg-primary">Good</span>
                            {% elif r.grade >= 60 %}
                                <span class="badge bg-info">Average</span>
                            {% elif r.grade >= 40 %}
                                <span class="badge bg-warning">Pass</span>
                            {% else %}
                                <span class="badge bg-danger">Fail</span>
                            {% endif %}
                        {% else %}
                            <span class="text-muted">-</span>
                        {% endif %}
                    </td>
                    <td>
                        <div class="btn-group btn-group-sm">
                            <button class="btn btn-outline-primary"
onclick="viewStudent('{{ r.roll_number }}')">
                                <i class="fas fa-eye"></i>

```

```

        </button>
        <button class="btn btn-outline-info"
onclick="editGrade('{{ r.roll_number }}', '{{ r.subject }}')">
            <i class="fas fa-edit"></i>
        </button>
    </div>
</td>
</tr>
{% else %}
<tr>
    <td colspan="6" class="text-center py-5">
        <div class="py-4">
            <i class="fas fa-file-alt fa-3x text-muted mb-
3"></i>

            <h5 class="text-muted">No Report Data
Available</h5>

            <p class="text-muted">Add grades to generate
reports</p>

            <a href="/add-grade" class="btn btn-primary">
                <i class="fas fa-plus me-1"></i> Add Grades
            </a>
        </div>
    </td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
{% if report_data %}
<div class="card-footer bg-white">
    <div class="d-flex justify-content-between align-items-center">
        <div>Showing {{ report_data|length }} records</div>
        <div class="text-muted">Last updated: {{ current_time }}</div>
    </div>
</div>
{% endif %}
</div>

<!-- JavaScript for filtering and functionality -->
<script>
document.addEventListener('DOMContentLoaded', function() {
    const subjectFilter = document.getElementById('subjectFilter');
    const gradeFilter = document.getElementById('gradeFilter');
    const sortBy = document.getElementById('sortBy');

```

```

const searchInput = document.getElementById('searchInput');
const rows = document.querySelectorAll('.report-row');

// Filter functionality
function applyFilters() {
    const subjectValue = subjectFilter.value;
    const gradeValue = gradeFilter.value;
    const searchValue = searchInput.value.toLowerCase();

    rows.forEach(row => {
        const subject = row.getAttribute('data-subject');
        const grade = parseFloat(row.getAttribute('data-grade'));
        const text = row.textContent.toLowerCase();

        let subjectMatch = subjectValue === 'all' || subject ===
subjectValue;
        let gradeMatch = true;
        let searchMatch = text.includes(searchValue);

        // Apply grade filter
        if (gradeValue !== 'all') {
            const [min, max] = gradeValue.split('-').map(Number);
            gradeMatch = grade >= min && grade <= max;
        }

        // Show/hide based on filters
        if (subjectMatch && gradeMatch && searchMatch) {
            row.style.display = '';
        } else {
            row.style.display = 'none';
        }
    });
}

// Add event listeners
subjectFilter.addEventListener('change', applyFilters);
gradeFilter.addEventListener('change', applyFilters);
sortBy.addEventListener('change', applyFilters);
searchInput.addEventListener('input', applyFilters);

// Initial filter application
applyFilters();
});

function viewStudent(rollNumber) {

```



```

    window.location.href = `/view-student/${rollNumber}`;
}

function editGrade(rollNumber, subject) {
    alert(`Edit grade for Roll: ${rollNumber}, Subject: ${subject}`);
    // Implement edit functionality here
}

function exportToCSV() {
    // Simple CSV export implementation
    let csvContent = "Roll Number,Name,Subject,Grade,Status\n";

    document.querySelectorAll('.report-row').forEach(row => {
        if (row.style.display !== 'none') {
            const cells = row.querySelectorAll('td');
            const roll = cells[0].textContent.trim();
            const name = cells[1].textContent.trim();
            const subject = cells[2].textContent.trim();
            const grade = cells[3].textContent.trim();
            const status = cells[4].textContent.trim();

            csvContent +=
`"${roll}","${name}","${subject}","${grade}","${status}"\n`;
        }
    });

    const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
    const link = document.createElement('a');
    link.href = URL.createObjectURL(blob);
    link.download = 'student_reports.csv';
    link.click();
}
</script>

<style>
.avatar-sm {
    width: 32px;
    height: 32px;
    font-size: 0.8rem;
}

.table th {
    border-top: none;
    font-weight: 600;
    text-transform: uppercase;
    font-size: 0.8rem;
}

```

```

        color: #6c757d;
    }
    .report-row:hover {
        background-color: #f8f9fa;
    }
</style>
{% endblock %}

```

<>Students.html

```

{% extends "base.html" %}
{% block title %}Students{% endblock %}
{% block header %}🔍 Students{% endblock %}

{% block content %}
<div class="chart-card">
    <div class="d-flex justify-content-between align-items-center">
        <h5 class="m-0">All Students</h5>
        <a href="/add-student" class="btn btn-sm btn-success">🔍 Add Student</a>
    </div>
    <table class="table table-striped mt-3">
        <thead class="table-dark"><tr><th>Roll
No</th><th>Name</th><th>Action</th></tr></thead>
        <tbody>
            {% for s in students %}
            <tr>
                <td>{{ s.roll_number }}</td>
                <td>{{ s.name }}</td>
                <td>
                    <a href="/student/{{ s.roll_number }}" class="btn btn-sm btn-outline-
primary">View</a>
                </td>
            </tr>
            {% endfor %}
            {% if not students %}
            <tr><td colspan="3" class="text-center text-muted">No students
found.</td></tr>
            {% endif %}
        </tbody>
    </table>
</div>
{% endblock %}

```

<>subjects_topper.html

```
{% extends "base.html" %}
{% block content %}
<div class="d-flex justify-content-between align-items-center mb-4">
    <h3 class="mb-0"><i class="fas fa-trophy me-2 text-warning"></i>Subject
Toppers</h3>
</div>

<!-- Filters -->
<div class="card shadow-sm mb-4">
    <div class="card-body py-3">
        <div class="row g-3">
            <div class="col-md-4">
                <label class="form-label fw-bold">Search Subject</label>
                <div class="input-group">
                    <span class="input-group-text bg-light"><i class="fas fa-
search"></i></span>
                    <input type="text" class="form-control" id="subjectSearch"
placeholder="Type to search...">
                </div>
            </div>
            <div class="col-md-4">
                <label class="form-label fw-bold">Sort By</label>
                <select class="form-select" id="sortSelect">
                    <option value="subject">Subject A-Z</option>
                    <option value="subject_desc">Subject Z-A</option>
                    <option value="marks">Highest Marks</option>
                </select>
            </div>
            <div class="col-md-4">
                <label class="form-label fw-bold">Class Filter</label>
                <select class="form-select" id="classFilter">
                    <option value="all">All Classes</option>
                    <option value="X">Class X</option>
                    <option value="XI">Class XI</option>
                    <option value="XII">Class XII</option>
                </select>
            </div>
        </div>
    </div>
</div>

<!-- Subject Toppers Cards -->
<div class="row" id="toppersContainer">
```

```

    {% for t in toppers %}
    <div class="col-lg-6 col-xl-4 mb-4 subject-card" data-subject="{{
t.subject|lower }}">
        <div class="card shadow-sm h-100 subject-topper-card">
            <div class="card-header bg-warning text-white">
                <h5 class="mb-0">
                    <i class="fas fa-book me-2"></i>{{ t.subject }}
                    <span class="badge bg-white text-warning ms-2">Top: {{
t.max_grade }}%</span>
                </h5>
            </div>
            <div class="card-body">
                <!-- 1st Rank -->
                <div class="topper-item first-rank mb-3 p-3 bg-light rounded-3">
                    <div class="d-flex align-items-center">
                        <div class="rank-badge bg-warning text-white rounded-
circle me-3">
                            <span class="fw-bold">1</span>
                        </div>
                        <div class="flex-grow-1">
                            <h6 class="mb-1 fw-bold">{{ t.name }}</h6>
                            <p class="mb-1 text-muted small">Roll: {{
t.roll_number }}</p>
                            <div class="progress" style="height: 6px;">
                                <div class="progress-bar bg-warning" data-
grade="{{ t.max_grade }}"></div>
                            </div>
                            <small class="text-muted">{{ t.max_grade }}%
Marks</small>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    {% else %}
    <div class="col-12">
        <div class="card shadow-sm text-center py-5">
            <div class="card-body">
                <i class="fas fa-trophy fa-3x text-muted mb-3"></i>
                <h5 class="text-muted">No Toppers Data Available</h5>
                <p class="text-muted">Add grades to see subject toppers</p>
                <a href="/add-grade" class="btn btn-warning">
                    <i class="fas fa-plus me-1"></i> Add Grades
                </a>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
</div>
{% endfor %}
</div>

<!-- JavaScript for interactive features -->
<script>
    document.addEventListener('DOMContentLoaded', function() {
        // Progress bars ko set karo
        const progressBars = document.querySelectorAll('.progress-bar[data-
grade]');
        progressBars.forEach(bar => {
            const grade = parseFloat(bar.getAttribute('data-grade'));
            bar.style.width = grade + '%';
        });

        // Search functionality
        const subjectSearch = document.getElementById('subjectSearch');
        if (subjectSearch) {
            subjectSearch.addEventListener('input', function() {
                const searchTerm = this.value.toLowerCase();
                const cards = document.querySelectorAll('.subject-card');

                cards.forEach(card => {
                    const subject = card.getAttribute('data-subject');
                    if (subject.includes(searchTerm)) {
                        card.style.display = 'block';
                    } else {
                        card.style.display = 'none';
                    }
                });
            });
        }

        // Sort functionality
        const sortSelect = document.getElementById('sortSelect');
        if (sortSelect) {
            sortSelect.addEventListener('change', function() {
                alert('Sorted by: ' + this.value);
            });
        }

        // Filter functionality
        const classFilter = document.getElementById('classFilter');

```

```
        if (classFilter) {
            classFilter.addEventListener('change', function() {
                alert('Filtered by: ' + this.value);
            });
        }
    });
</script>

<style>
.subject-topper-card {
    border-radius: 12px;
    transition: all 0.3s ease;
    border: none;
}

.subject-topper-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 8px 25px rgba(0,0,0,0.15);
}

.card-header {
    border-radius: 12px 12px 0 0;
    border: none;
}

.rank-badge {
    width: 30px;
    height: 30px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 0.9rem;
}

.progress {
    border-radius: 10px;
}

.progress-bar {
    border-radius: 10px;
}

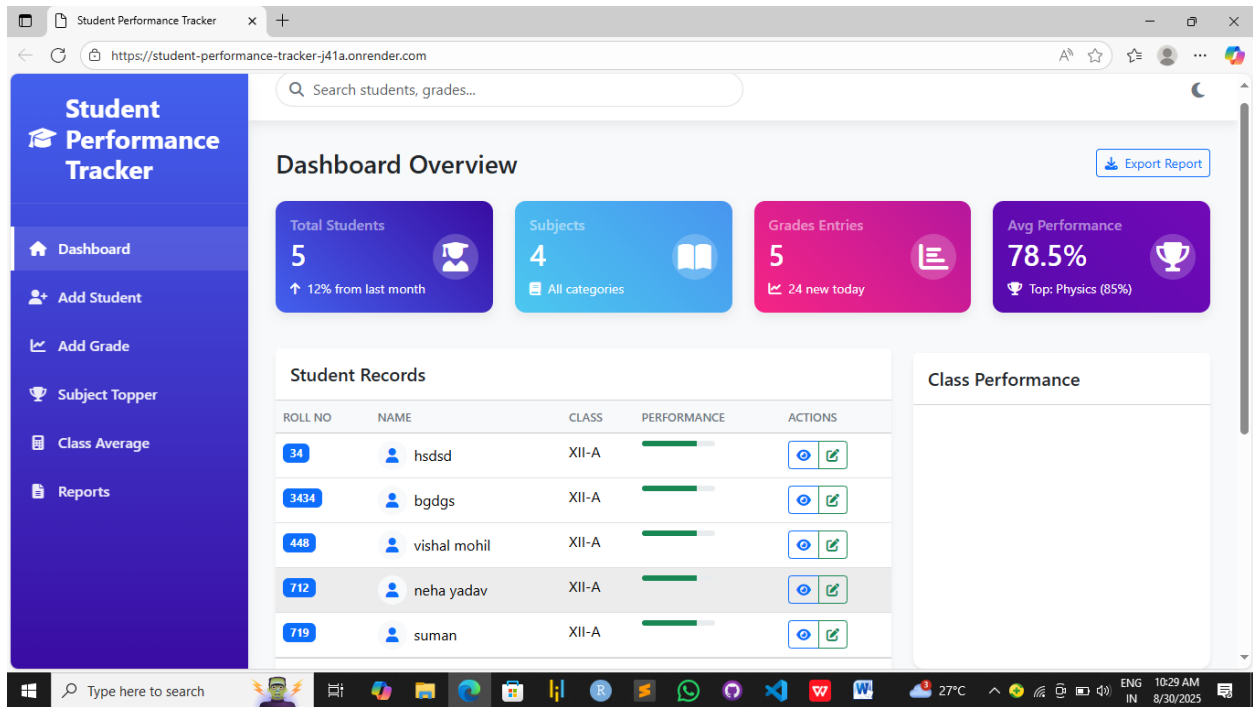
.first-rank {
    background-color: #fff8e1;
    border: 2px solid #ffeaa7;
```

```
}  
</style>  
{% endblock %}
```

<>view_student.html

```
{% extends "base.html" %}  
{% block content %}  
<h3 class="mb-3">Student: {{ student.name }} ({{ student.roll_number }})</h3>  
<div class="card shadow-sm">  
  <div class="table-responsive">  
    <table class="table mb-0">  
      <thead class="table-light">  
        <tr>  
          <th>Subject</th>  
          <th>Grade</th>  
        </tr>  
      </thead>  
      <tbody>  
        {% for g in grades %}  
          <tr>  
            <td>{{ g.subject }}</td>  
            <td>{{ g.grade }}</td>  
          </tr>  
          {% else %}  
            <tr><td colspan="2" class="text-center text-muted py-4">No grades  
yet.</td></tr>  
          {% endfor %}  
        </tbody>  
      </table>  
    </div>  
  </div>  
{% endblock %}
```

OUTPUT:-



Github link:-

Neha yadav:-

<https://github.com/Neha-yadav-Full/student-performance-tracker>

Vishal Mohil:-

<https://github.com/Vishalmohil63/student-performance-tracker>

Project Deployment link:-

<https://student-performance-tracker-j41a.onrender.com/>