

**COLLEGE CODE : 9222**

**COLLEGE NAME : THENI KAMMAVAR SANGAM COLLEGE OF TECHNOLOGY**

**DEPARTMENT : B.TECH(IT)**

**STUDENT NM-ID : FEEFB73EB0DDFF6AEB749DEF14E540EA ROLL NO : 922223205004**

**DATE : 19/09/2025**

**Completed The Project Named As PHASE 5**

**NAME : IBM-NJ-STUDENT GRADING SYSTEM**

**SUBMITTED BY,**

**MOBILE NO : 8667799800**

**NAME : ASWATHAMAN**



**NAME ASWATHAMAN M**

**MOBILE NO : 8667799800**

**Project Demonstration & Documentation**

**Title:Student Greading System**

## Final Demo Walkthrough

 **A complete walkthrough of the Student Grading System.**

 **Show all key features: student registration, grade entry, grade calculation, report generation, etc.**

 **Include a live demonstration or a video recording link (if applicable).**

 **Mention technologies used (e.g., frontend, backend, database).**

 **Highlight user roles (Admin, Teacher, Student)**

# Project Report

 **Introduction: Brief overview of the project and its purpose.**

 **Objective: What the system aims to solve (automated grading, result management, etc.).**

 **Technology Stack: Tools and frameworks used (e.g., React, Node.js, MySQL).**

 **System Architecture: High-level design or architecture diagram.**

 **Features: List and explanation of all major features.**

 **Conclusion: Summary of outcomes and future improvements.**

# Program

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<title>Student Grading System</title>**

**<style> body {**

**background-image:url("https://media.istockphoto.com/id/172413295/photo/an-up-close-picture-of-report-card-**

**grades.jpg?s=612x612&w=0&k=20&c=d95S74oUPLZA98yn6QKG0-6OEbgAqgroKzQWH5GxwKA="); background-size: cover;**

**font-family: Arial, sans-serif;**

**padding: 20px; background-color: #f2f2f2; align-items: center;**

**justify-content: center;**

**}**

**.container {**

**background-color: #fff; padding:50px;**

**border-radius: 10px;**

**max-width: 500px; margin: auto;**

**box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);**

**}**

**h2 {**

**text-align: center; color: #333;**

**}**

**label {**

**display: block; margin: 10px 0 5px;**

**}**

input {

width: 100%; padding: 8px;

margin-bottom: 15px; border: 1px solid #ccc; border-radius: 5px;

**}**

button { width: 100%;

padding: 10px;

background-color: #4CAF50; color: white;

border: none; border-radius: 5px; cursor: pointer;

**}**

button:hover { background-color:green;

**}**

.result {

margin-top: 20px; font-weight: normal;

**}**

</style>

</head>

<body>

<div class="container">

<h2>Student Grading System</h2>

<label for="name">Student Name:</label>

<input type="text" id="name" placeholder="Enter student name">

<label for="subject1">Subject 1 Marks:</label>

<input type="number" id="subject1" placeholder="Enter marks out of 100">

<label for="subject2">Subject 2 Marks:</label>

<input type="number" id="subject2" placeholder="Enter marks out of 100">

<label for="subject3">Subject 3 Marks:</label>

<input type="number" id="subject3" placeholder="Enter marks out of 100">

<label for="subject4">Subject 4 Marks:</label>

<input type="number" id="subject4" placeholder="Enter marks out of 100">

<label for="subject5">Subject 5 Marks:</label>

<input type="number" id="subject5" placeholder="Enter marks out of 100">

<button onclick="calculateGrade()">Calculate Grade</button>

<div class="result" id="result"></div>

</div>

<script>

function calculateGrade() {

const name = document.getElementById('name').value.trim(); const subjects = [ parseFloat(document.getElementById('subject1').value), parseFloat(document.getElementById('subject2').value), parseFloat(document.getElementById('subject3').value), parseFloat(document.getElementById('subject4').value), parseFloat(document.getElementById('subject5').value)

**];**

const resultDiv = document.getElementById('result');

// Validation

if (!name || subjects.some(mark => isNaN(mark) || mark < 0 || mark > 100)) { resultDiv.innerHTML = "⚠ Please enter a valid name and marks (0-100) for all subjects."; resultDiv.style.color = 'red';

return;

**}**

const total = subjects.reduce((sum, mark) => sum + mark, 0); const average = total / subjects.length;

let grade;

if (average >= 90) grade = 'A+'; else if (average >= 80) grade = 'A'; else if (average >= 70) grade = 'B'; else if (average >= 60) grade = 'C'; else if (average >= 50) grade = 'D'; else grade = 'F';

resultDiv.style.color = 'black'; resultDiv.innerHTML = `

<strong>${name}</strong><br> Total Marks: ${total} / 500<br> Average: ${average.toFixed(2)}<br> Grade: <strong>${grade}</strong>

**`;**

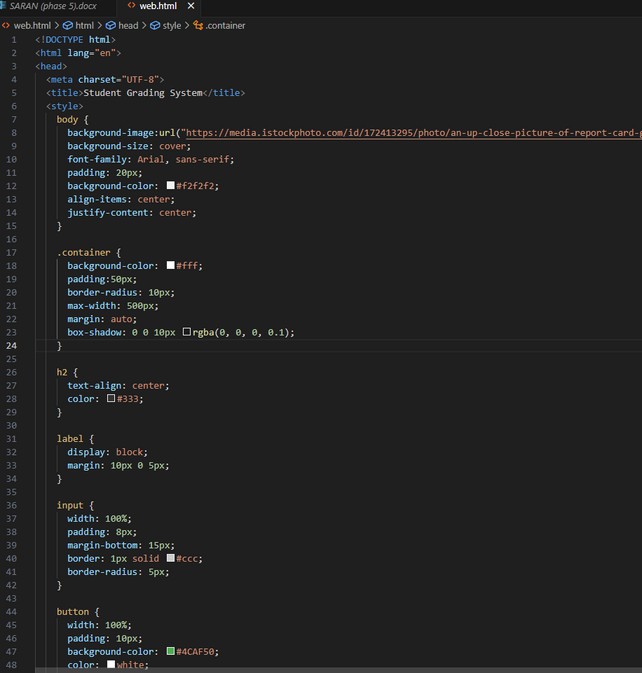
**}**

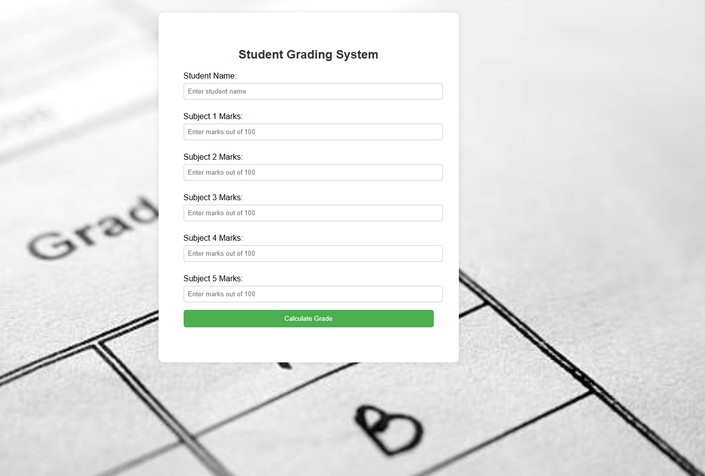
</script>

</body>

</html>

## Screenshots / API Documentation

**



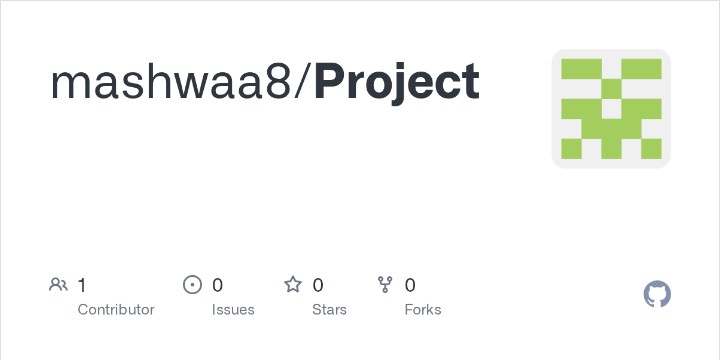
1. *Challenges & Solutions*

 **Challenge 1: Handling dynamic grading scalesSolution: Created a flexible grading logic with configurable rules.**

 **Challenge 2: Securing student dataSolution: Implemented JWT authentication and hashed passwords.**

 **Challenge 3: Deploying with database connectivitySolution: Used environment variables and cloud database (e.g., MongoDB Atlas / Firebase / Railway).**

## GitHub README

**

FINAL SUBMISSION :

