

# **RESULT CALCULATOR**

## **18CSE361T – WEB PROGRAMMING**

### **Mini Project Report**

*Submitted by*

**Yash Kumar [Reg. No.: RA2111031010056]**

**B.Tech. Computer Science and Engineering  
with Specialization in Information Technology**



**SCHOOL OF COMPUTING  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**S.R.M. NAGAR, KATTANKULATHUR – 603 203  
KANCHEEPURAM DISTRICT**

**OCTOBER 2023**

## **BONAFIDE CERTIFICATE**

This is to certify that 18CSE361T – Web Programming mini project report titled “Result Calculator” is the bonafide work of Yash Kumar (Reg. No. RA2111031010056) who undertook the task of completing the project within the allotted time.

### **SIGNATURE**

Dr. M.Anand  
Assistant Professor  
Department of Networking  
and Communications  
SRM Institute of Science  
and Technology

### **SIGNATURE**

Dr. Annapurani Panaiyappan K  
Professor and Head  
Department of Networking  
and Communications  
SRM Institute of Science  
and Technology

## **TABLE OF CONTENTS**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
	Abstract	
1	Introduction	
2	Module Description	
3	Database Design	
4	Implementation	
5	Results	
6	Conclusion	
	References	

## **Abstract**

The "Result Calculator" project is a software solution designed for educational institutions to automate and optimize the result management process. This system simplifies tasks related to student result computation and management. It allows administrators to input and manage student information, define grading systems, and enter marks or grades for courses. The system then calculates final grades, grade points, and GPAs based on predefined grading criteria, generating individual result sheets for students. Moreover, it offers data analysis and reporting, aiding institutions in making data-driven decisions for academic improvements.

The key benefits of this project include increased efficiency, reduced errors, improved transparency, and the automation of administrative tasks. It enhances the academic management process, saves time, and ensures accuracy in result calculation. Additionally, it provides mobile accessibility and data security features. The "Result Calculator" is a valuable tool for educational institutions seeking to streamline their result management processes and improve overall administrative efficiency.

# **CHAPTER 1**

## **INTRODUCTION**

The "Result Calculator" project is a comprehensive software solution developed to address the intricate and critical task of managing academic results in educational institutions. This innovative system has been meticulously designed to streamline and revolutionize the entire result management process. It caters to schools, colleges, and universities, aiming to alleviate the challenges associated with manual result calculations and data management.

At its core, the project encompasses several key features, including student information management, course and grading system configuration, result data entry, automated result calculation, result publication, data analysis and reporting, and robust security measures. These components work in harmony to simplify the process of result generation, reduce the risk of human errors, and enhance the overall administrative efficiency of educational institutions.

The "Result Calculator" not only benefits administrators and educators by saving time and ensuring data accuracy but also empowers students by providing transparent and accessible access to their academic results. With the ability to analyze data trends and make informed decisions, this project stands as a pivotal asset in modernizing educational institutions and fostering data-driven academic enhancements.

## CHAPTER-2

### MODULE DESCRIPTION

A "Result Calculator" project typically consists of various modules that collectively make the software functional and effective. Below is a module descriptor outlining the key modules of the "Result Calculator" project:

**1. User Management Module:**

- Description: This module manages user accounts and their access rights within the system. It includes roles for administrators, teachers, and students, allowing them to interact with the software based on their permissions.

**2. Student Information Management Module:**

- Description: This module handles the storage and management of student information, including personal details, enrollment, courses, and academic history.

**3. Course and Grading System Configuration Module:**

- Description: This module allows administrators to define and configure courses, subjects, and grading systems. It includes options for setting up grading scales, grading points, and pass/fail criteria.

**4. Result Entry Module:**

- Description: Professors and teachers use this module to input students' marks or grades for various subjects, assignments, and exams. The module performs validation checks to ensure data accuracy.

**5. Result Calculation Module:**

- Description: This module automatically computes final grades, grade points, and GPAs based on the entered data and predefined grading criteria. It applies algorithms for result calculation.

**6. Result Publication Module:**

- Description: This module generates and publishes individual student result sheets or transcripts. It offers options for secure online access and, if required, for printing physical result sheets.

**7. Data Analysis and Reporting Module:**

- Description: Educational institutions use this module to access statistical reports and analyses, including pass/fail rates, class

averages, and subject-wise performance. It supports data-driven decision-making for academic improvements.

**8. Security and Access Control Module:**

- Description: Role-based access control is implemented to ensure that only authorized personnel can access and modify the data. This module includes security measures to protect sensitive information.

**9. Data Backup and Recovery Module:**

- Description: This module performs regular backups of result data to ensure data integrity and the ability to recover in case of system failures.

**10. User-Friendly Interface Module:**

- Description: The software provides an intuitive and user-friendly interface for administrators, professors, and students, making interaction with the system easy and efficient.

**11. Scalability and Integration Module:**

- Description: This module is designed to make the system scalable and capable of integration with existing student information systems and databases used by educational institutions.

**12. Mobile Accessibility Module:**

- Description: This module ensures that users can access the system through mobile devices, making it convenient for both students and faculty.

Each of these modules plays a crucial role in the "Result Calculator" project, contributing to the efficiency, accuracy, and overall effectiveness of result management in educational institutions.

Is this conversation helpful so far?

## Database Design

**Users Table**

Field	Type	Description
User_ID	Primary Key	User identifier
Username	Text	User's username
Password	Text	User's password
Role	Text	User's role (admin, teacher, student)

**Students Table**

Field	Type	Description
Student_ID	Primary Key	Student identifier
First_Name	Text	Student's first name
Last_Name	Text	Student's last name



Field	Type	Description
Date_of_Birth	Date	Student's date of birth
Registration_Number	Text	Student's registration number
Course_ID	Foreign Key	Course associated with the student

**Courses Table**

Field	Type	Description
Course_ID	Primary Key	Course identifier
Course_Name	Text	Name of the academic course

**Subjects Table**

Field	Type	Description
Subject_ID	Primary Key	Subject identifier

Field	Type	Description
Course_ID	Foreign Key	Course associated with the subject
Subject_Name	Text	Name of the subject

**Grading\_Systems Table**

Field	Type	Description
Grading_System_ID	Primary Key	Grading system identifier
System_Name	Text	Name of the grading system

**Grades Table**

Field	Type	Description
Grade_ID	Primary Key	Grade identifier
Grading_System_ID	Foreign Key	Grading system associated with the grade

Field	Type	Description
Grade_Name	Text	Name of the grade
Grade_Point	Decimal	Corresponding grade point
Minimum_Score	Numeric	Minimum score required for the grade
Maximum_Score	Numeric	Maximum score for the grade

## Results Table

Field	Type	Description
Result_ID	Primary Key	Result identifier
Student_ID	Foreign Key	Student associated with the result
Subject_ID	Foreign Key	Subject associated with the result
Marks	Numeric	Marks obtained by the student

Field	Type	Description
Grade	Text	Grade awarded for the subject
Year	Numeric	Academic year
Semester	Text	Academic semester

### User\_Student\_Relationship Table

Field	Type	Description
Relationship_ID	Primary Key	Relationship identifier
User_ID	Foreign Key	User associated with a student
Student_ID	Foreign Key	Student associated with the user

**User\_Course\_Relationship Table**

Field	Type	Description
Relationship_ID	Primary Key	Relationship identifier
User_ID	Foreign Key	User associated with a course
Course_ID	Foreign Key	Course associated with the user

**User\_Security\_Logs Table**

Field	Type	Description
Log_ID	Primary Key	Log identifier
User_ID	Foreign Key	User associated with the log entry
Log_Date	Date/Time	Date and time of the log entry
Activity_Description	Text	Description of the security-related activity

This tabular format outlines the tables, fields, data types, and descriptions for the "Result Calculator" project's database. It provides a clear structure for storing and managing data related to users, students, courses, grading systems, grades, and results.

## IMPLEMENTATION

### HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Result Calculator</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="calculator">
    <h1>Result Calculator</h1>
    <label for="subject1">Subject 1:</label>
    <input type="number" id="subject1" placeholder="Enter marks for subject 1">

    <label for="subject2">Subject 2:</label>
    <input type="number" id="subject2" placeholder="Enter marks for subject 2">

    <label for="subject3">Subject 3:</label>
    <input type="number" id="subject3" placeholder="Enter marks for subject 3">

    <button onclick="calculateResult()">Calculate Result</button>

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

  <script src="script.js"></script>
</body>
</html>
```

## CSS:

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f4f4f4;  
  margin: 0;  
  padding: 0;  
}
```

```
.calculator {  
  background-color: #fff;  
  max-width: 400px;  
  margin: 20px auto;  
  padding: 20px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  box-shadow: 0 0 5px #ccc;  
}
```

```
h1 {  
  text-align: center;  
}
```

```
label, input {  
  display: block;  
  margin-bottom: 10px;  
}
```

```
button {  
  display: block;  
  margin: 10px auto;  
  background-color: #007bff;  
  color: #fff;  
  padding: 10px 20px;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

```
button:hover {  
  background-color: #0056b3;  
}
```

```
#result {  
  text-align: center;  
  font-weight: bold;  
  font-size: 18px;  
  margin-top: 20px;  
}
```

## JAVASCRIPT:

```
function calculateResult() {  
    const subject1 = parseFloat(document.getElementById("subject1").value);  
    const subject2 = parseFloat(document.getElementById("subject2").value);  
    const subject3 = parseFloat(document.getElementById("subject3").value);  
  
    if (isNaN(subject1) || isNaN(subject2) || isNaN(subject3)) {  
        alert("Please enter valid marks for all subjects.");  
        return;  
    }  
  
    const totalMarks = subject1 + subject2 + subject3;  
    const averageMarks = totalMarks / 3;  
  
    const resultElement = document.getElementById("result");  
    resultElement.textContent = `Total Marks: ${totalMarks}, Average Marks:  
${averageMarks.toFixed(2)}`;  
}
```

## RESULT

### Result Calculator

Subject 1:	<input type="text" value="89"/>
Subject 2:	<input type="text" value="70"/>
Subject 3:	<input type="text" value="98"/>
<input type="button" value="Calculate Result"/>	

### Result

Total Marks: 257.00  
Average Marks: 85.67  
Grade: A

Note: Grades are based on the average marks.



## CONCLUSION

In conclusion, the result calculator web programming project demonstrates the creation of a simple web page that allows users to input marks for three subjects and calculates the total marks, average marks, and assigns a grade based on the average marks. Here are some key takeaways from this project:

1. **HTML:** The HTML code provides the structure of the web page. It includes input fields for entering marks, a button for triggering the calculation, and a section for displaying the results.
  2. **CSS:** CSS is used for styling the web page, making it visually appealing. It defines the layout, colors, fonts, and overall design of the page.
  3. **JavaScript:** JavaScript is used to calculate the total marks, average marks, and grade. It provides the interactivity needed for the calculation and updates the result dynamically on the page.
  4. **User Interaction:** Users can input marks for three subjects, and upon clicking the "Calculate Result" button, the page displays the total marks, average marks, and a grade.
  5. **Improvements:** The project evolved through iterations to improve the design, including separate rows for subjects, different colors, and the addition of a green background for the entire page. The final design is visually appealing and informative.
  6. **Grading:** A grading system based on average marks has been incorporated, and it assigns grades such as A+, A, B, etc.
  7. **Note:** A note is included to provide users with additional information about the grading system.
- 

## REFERENCES

1. Online Tutorials and Documentation:
  - MDN Web Docs (Mozilla Developer Network): <https://developer.mozilla.org/>
  - W3Schools: <https://www.w3schools.com/>
2. Online Learning Platforms:
  - Coursera: <https://www.coursera.org/>
  - edX: <https://www.edx.org/>
  - Udacity: <https://www.udacity.com/>
  - Codecademy: <https://www.codecademy.com/>
3. Books:
  - "HTML and CSS: Design and Build Websites" by Jon Duckett

- "JavaScript and JQuery: Interactive Front-End Web Development" by Jon Duckett
- "Learning Web Design" by Jennifer Robbins
- 4. YouTube Tutorials:
  - There are many web development tutorial channels on YouTube. A popular one is "Traversy Media": <https://www.youtube.com/user/TechGuyWeb>
- 5. Forums and Communities:
  - Stack Overflow: <https://stackoverflow.com/>
  - Reddit's r/webdev: <https://www.reddit.com/r/webdev/>
- 6. Courses and Specializations:
  - Check out online courses and specializations in web development on platforms like Coursera and edX.
- 7. Documentation for Web Technologies:
  - Refer to official documentation for HTML, CSS, JavaScript, and other web technologies to get in-depth information.

When using online resources, make sure to verify the credibility of the source, and consider enrolling in courses or programs that suit your learning style and goals. Remember that web development is a broad field, and you can find resources for different aspects of it, such as front-end development, back-end development, full-stack development, and more.

