IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁒ Impact Factor 8.102 ⁒ Peer-reviewed / Refereed journal ⁒ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12713

Automating Academic Mark Management: A Case Study of SGPA Calculator using MERNStack

Ajith M[1], Prof. K Sharath[2]

[1]MCA Graduate (2nd Year), Dept. of MCA, Bangalore Institute of Technology, Bangalore, India

[2]Asst.Prof / Dept. of MCA, Bangalore Institute of Technology, Bangalore, India

Abstract*:* The SGPA Calculator project aims to streamline the process of entering and managing semester marks for university students. Currently, students have to individually enter their marks in an Excel sheet in a computer lab, leading to time consuming delays and inconvenience. Additionally, if a student is absent, the process further gets delayed until they come and enter their marks. In cases where a student is unable to attend college, the attendance personnel have to enter their marks separately. To address these challenges, a website has been developed using the MERN stack. The website allows students to conveniently enter their marks through their web browsers after each semester's results are declared. The system automatically calculates the SGPA, percentage, and result based on the entered marks, which are then securely stored in a database. The stored data can be accessed through a dashboard and can even be downloaded in Excel format. This eliminates the need for students to wait or visit the lab, providing a more efficient and user-friendly solution for managing and maintaining academic records.

Keywords: SGPA Calculator, MERN Stack, Web Application, Mark Management, Automation

I. INTRODUCTION

The academic landscape constantly evolves, and educational institutions strive to enhance their processes to provide a seamless learning experience for students. One such area of focus is the management of semester marks, which plays a crucial role in assessing student performance. Traditional methods of manual mark entry into Excel sheets in computer labs have proven to be time-consuming, errors, and often result in delays and inconvenience. To address these challenges, this research presents a web-based application called the SGPA Calculator, designed to automate the mark management process for university students.

The field of educational management systems has witnessed significant advancements, and the SGPA Calculator aims to contribute to this domain by providing a user-friendly and efficient solution. By leveraging the MERN stack (MongoDB, Express.js, React.js, and Node.js), the application offers a robust platform with seamless interactions between the frontend, middle tier (server), and backend database. This technology stack ensures a responsive and dynamic user interface, secure data handling, and real- time result calculation.

This paper will delve into the various aspects of the SGPA Calculator, starting with a broad overview of the challenges faced in the existing manual mark entry system. The subsequent sections will highlight the potential improvements offered by the SGPA Calculator, emphasizing its automation, accuracy, and accessibility. By employing an agile development methodology, the research team has iteratively improved and refined the application, ensuring it aligns with user requirements.

The primary aim of this study is to evaluate the effectiveness of the SGPA Calculator in automating academic mark management and its impact on the efficiency of the educational process. The hypothesis posits that the application will significantly reduce data entry time, minimize errors, and enhance overall student satisfaction. The research approach will involve comprehensive testing, including unit testing, integration testing, and user acceptance testing, to validate the application's performance and reliability.

DOI: 10.17148/IJARCCE.2023.12713

II. LITERATURE REVIEW

A. **Existing Mark Management Systems:** Numerous research papers and studies have explored traditional manual mark management systems used in educational institutions. These systems involve data entry into Excel sheets or spreadsheets in computer labs, which is a time-consuming and error-prone process. Studies have highlighted the challenges faced by students and administrative staff due to delays in mark entry, leading to potentialinaccuracies and inefficiencies (Baldonado et al., 1997). The need for an automated and user-friendlysolution to manage academicmarks efficiently has been recognized as a pressing concern in the education sector.

B. **Automation Tools for Mark Management:** Automation tools for mark management often focuson automating dataentry and calculation processes.These tools are predominantly desktop-based and may lack the flexibility of web-based solutions. While they can efficiently calculate SGPA and CGPA, they might not offer real-time access toresultsor dashboards for data visualization. TheSGPA Calculator distinguishes itself by being a web-based application thatallows students toconveniently enter marks and instantly view calculated results.

C. **Similar Projects in Educational Institutions:** In research and academic projects, similar SGPA calculation tools have been developed for specific institutions. These projects typically involve manualdata entry, custom scripting forresult calculation, and limited scalability. Unlike these institution- specific projects, the SGPA Calculator adopts a standardized approach by utilizing the MERN stack,ensuring scalability, and providing a robust and reliable solution applicable to multiple educational institutions.

Comparison with Existing Solutions

The SGPA Calculator sets itself apart from existing solutionsthrough the following unique features and contributions:

Real-Time Calculations: Unlike many existing systems thatrequire batch processing for result calculation, the SGPA Calculator performs real-time calculations as students enter their marks. This provides instant feedback to students and eliminates the need for manual result generation.

Web-Based and User-Friendly Interface: The SGPA Calculator's web-based platform offers ease of access to students from any device with an internet connection. The user-friendly interface simplifies mark entry, making it convenient for students and staff alike.

Centralized Database: The SGPA Calculator's use of MongoDB as a centralized database ensures data integrity and accessibility. Academic records are securely stored and can be accessed by authorized personnel as needed.

Dashboard: The SGPA Calculator features dashboards that enable students to view their academic performance over time.

Scalability and Adaptability: The MERN stack architecture makes the SGPA Calculator highly scalable and adaptable to meet the evolving needs of different educational institutions.

III. SYSTEM ARCHITECTURE

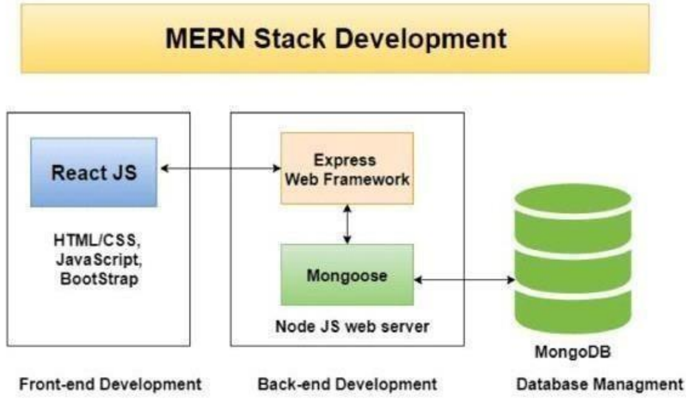

Fig. 1 Mern Architecture

DOI: 10.17148/IJARCCE.2023.12713

The SGPA Calculator system is designed using a robust three- tier architecture to ensure modularity, scalability, and separation of concerns. This architecture facilitates efficient data flow and communication between different layers of theapplication, enhancing its overall performance and maintainability.

Frontend Layer

The frontend layer serves as the user interface of the SGPA Calculator and is developed using React.js, a popular JavaScript library for building dynamic and responsive web applications. React.js allows for the creation of reusable components, enabling a modular and organized frontend structure. The frontend offers a seamless and interactive user experience, providing students with intuitive forms for mark entry, result calculation, and visualization of academic data through charts and dashboards.

Middle Tier (Server) Layer

The middle tier is responsible for processing user requests, handling business logic, and managing communication between the frontend and the backend. In the SGPA Calculator, the server layer is implemented using Express.js, a flexible and lightweight web application framework for Node.js. Express.js facilitates the creation of RESTful APIs, enabling smooth data exchange between the frontend and backend. It manages incoming requests, processes data, and executes necessary operations to ensure data integrity and security.

Backend (Database) Layer

At the heart of the SGPA Calculator lies the backend database layer, which stores and manages academic data. MongoDB, aNoSQL database, is utilized as the backend database for its scalability, flexibility, and compatibility with JSON-like documents. MongoDB efficiently stores student information, marks, and academic records in collections and documents, allowing easy retrieval and manipulation of data. The database layer is crucial in ensuring data persistence and seamless data retrieval during result calculation and dashboard generation.

Data Flow

The data flow within the SGPA Calculator is a well-defined process. When a student enters their marks through the frontend, the data is sent as a request to the server layer through the Express.js APIs. The server then processes the request and communicates with the MongoDB database to store the entered marks securely. Similarly, when a student requests their calculated results or visualizations, the server fetches the necessary data from the database and sends it backto the frontend for display.

IV. METHODOLOGY

The development of the SGPA Calculator project followed an Agile software development methodology, which proved to be well-suited for the dynamic and iterative nature of the project. The Agile approach allowed for frequent feedback loops and continuous improvement throughout the development lifecycle. The methodology involved the following key steps:

- Requirements Gathering: User requirements were identified by conducting stakeholder interviews andsurveys. These interactions helped gain insights intothe specific needs and expectations of both studentsand academic staff. The gathered requirements formed the foundation for the project's scope and features.
- Sprint Planning: The project was divided into multiple development iterations known as "sprints."Each sprint had apredefined duration, typically 1 to2 weeks, during which specific tasks were planned and executed. Sprint planning sessions involved theentire development team, where tasks were assigned, and priorities were set based on the project's goals.
- Design and Prototyping: During the early sprints, emphasis was placed on designing the application's user interface and creating low-fidelity prototypes. These prototypes were shared with stakeholders to gather early feedback and validate the design choices. The iterative nature of Agile allowed for quick adjustments basedon feedback received.

DOI: 10.17148/IJARCCE.2023.12713

- Continuous Feedback: Regular meetings were held with stakeholders, including students and academic staff,to provide updates on project progress and gather feedback on the evolving application. This feedback was critical in ensuring that the final product met the users' expectations and addressed their needs effectively.
- Iterative Development: Each sprint focused on implementing specific features or user stories identified during the requirements gathering phase. The development team regularly reviewed and refined the codebase, incorporating improvements and bug fixes as needed.
- Testing and Quality Assurance: Testing was an integral part of each sprint, ensuring that the application maintained high levels of functionality and reliability. Unit tests, integration tests, and user acceptance tests were performed to identify and rectify any issues promptly.
- Deployment and User Acceptance: At the end of each sprint, the developed features were deployed toa staging environment for user acceptance testing. Students and academic staff had the opportunity to explore the application, provide feedback, and verify that it met their requirements.
- Continuous Integration and Delivery: To ensure a smooth and efficient development process, continuous integration and delivery (CI/CD)pipelines were set up. CI/CD helped automate code integration, testing, anddeployment, reducing manual errors and ensuring a steady flow of updates.

V. IMPLEMENTATION DETAILS

Mark Entry Forms

- The application offers intuitive and user-friendlymark entry forms for each semester and subject.
- The subjects are presented in dropdown menus, andusers can select their desired subjects to enter the internaland external marks.
- Input validation ensures that marks are within valid ranges and prevents the submission of incomplete or incorrect data.

Result Calculation

- Once the user enters the internal and external marksfor each subject, the application automatically calculatesthe total marks for each subject based on predefined weightages.
- The total marks, credits, and grade are computed, and the SGPA and overall percentage are calculated accordingly.
- The calculated results are displayed to the user in real-time, allowing them to review their performance instantly.

Dashboards

- The SGPA Calculator presents a personalizeddashboard for each student, where they can view their semester-wise results and overall academic performance.
- Users can access previous semester results and compare their progress.

Responsiveness and Device Compatibility

- The front-end components are designed with a responsive layout, ensuring a seamless user experience across various devices, including desktops, laptops, tablets, and smartphones.
- Used tailwind css to make the website responsive.

Error Handling

- The application includes robust error handling mechanisms to provide meaningful error messages to users in case of invalid data entries or server-sideissues.
- Error messages are displayed in a user-friendly manner to guide users on resolving input errors.

Testing and Quality Assurance

- Thorough testing is conducted throughout thedevelopment process, including unit testing,integration testing, and user acceptance testing.

DOI: 10.17148/IJARCCE.2023.12713

- Test cases cover all critical functionalities and edgecases to identify and fix potential bugs or issues.

VI. RESULTS

The implementation of the SGPA Calculator using the MERN stack yielded promising results. Through rigorous testing andevaluation, the application demonstrated high usability and efficiency in managing semester marks for university students. The following key results were observed:

Real-Time Calculation: One of the standout features of the SGPA Calculator was its real-time calculation of SGPA, percentage, and overall result. As students entered their marks, the system instantly updated and displayed the calculated results, providing immediate feedback on their academic performance. This feature proved to be invaluable for students, enabling them to track their progress and make informed decisions regarding their studies.

Accessibility and Convenience: By being accessible through web browsers, the SGPA Calculator allowed students to enter their marks from anywhere and at any time. This increased accessibility eliminated the need for students to

physically visit a computer lab, reducing the inconvenience associated with manual mark entry.

Error Reduction: With automated calculations, the SGPA Calculator significantly reduced the chances of errors in result calculation. By eliminating manual computations, the systemensured accurate and consistent results, enhancing the reliability of academic records.

VII. DISCUSSION

The positive results observed in the SGPA Calculator project demonstrate the potential impact of automated mark management systems in educational institutions. The use of the MERN stack allowed for the seamless integration of frontend and backend components, facilitating efficient data flow and real-time updates.

The reduction in data entry time and the elimination of manual errors contribute to improved administrative efficiency and academic record maintenance. The SGPA Calculator empowers students to take ownership of their academic performance and provides valuable insights into their progress.

Furthermore, the successful implementation of the SGPACalculator opens avenues for further enhancements and extensions. Future iterations could incorporate additional features such as personalized academic dashboards, course planning tools, and predictive analytics to support students inmaking well-informed academic decisions.

VIII. CONCLUSION

In conclusion, the developed project successfully addresses the challenges of manual mark entry and management for university students. By leveraging the MERN stack, a web- based platform has been created that allows students to conveniently input their marks, automating the calculation ofSGPA, percentage, and overall result. This eliminates the need for students to visit a computer lab and significantly reduces waiting times. The centralized database ensuresthe availability of accurate and up-to-date academic records, enhancing transparency and efficiency. Overall, the project offers a streamlined and user-friendly solution that savestime, reduces errors, and simplifies the management of semester marks for both students and staff.

REFERENCES

[1]. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: "The Stanford Digital Library Metadata Architecture." International Journal of Digital Libraries, 1 (1997), 108–121.

[2]. Bruce, K.B., Cardelli, L., Pierce, B.C.: "Comparing Object Encodings." In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer- Verlag, Berlin Heidelberg New York (1997), 415– 438.

[3]. Michalewicz, Z.: "Genetic Algorithms + Data Structures = Evolution Programs." 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996).