An Internship Project Report on

**Student Result Management System**

Third Year of Engineering

In

Computer Engineering

By

|  |  |
| --- | --- |
| Khushboo Rathod | TE4 – 48 |
| Vinay Savla | TE4 – 52 |
| Divyaa Thaivalappil | TE4 – 62 |
| Nikhil Viashya | TE4 – 65 |



**Department of Computer Engineering**

**Shah and Anchor Kutchhi Engineering College, Chembur**

**University of Mumbai**

**2022-2023**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Name** | **Page No.** |
| **1** | **Problem Definition** | **3** |
| **2** | **Modules** | **4** |
| **3** | **Database** | **6** |
| **4** | **Implementation** | **7** |
| **5** | **Results** | **11** |
| **6** | **Conclusion** | **15** |
| **7** | **References** | **16** |

**Chapter 1**

**Problem Definition**

The manual entry and calculation of student marks is time-consuming and prone to errors. This can lead to discrepancies in academic performance evaluations and delays in providing students with their academic records. A centralized SMS can improve the accuracy and efficiency of student mark management. It can also help to identify trends and patterns in academic performance.

A Student Result Management System (SRMS) is a web application that allows admin to add departments and their admins, subjects, students, and teachers for each department. It also allows admins to create notices that will be displayed on the student and faculty side. It allows teachers to add marks for students in each subject, and students to view the marks. The system is secure and prevents unauthorized access to student marks and results. We used the MERN stack, as it is a good choice for developing this type of system because it is a modern, scalable, and easy-to-use stack.

Here is the problem definition in short form:

* The system should allow the admin to add departments, faculties, and students in each department.
* The system should allow teachers to add marks for each student in each subject for each of the following:
  + Internal Assessment 1 (IA1)
  + Internal Assessment 2 (IA2)
  + End-of-semester exam (ESE)
  + Term work
  + Practical exam
* The system should allow students to view their own marks.
* The system should be secure.
* The system should be easy to use.

**Chapter 2**

**Modules**

The Student Result Management System (SRMS) contains three modules namely Admin, Faculty, and Student. Each of these modules includes profile viewing as a common feature. According to the authorization given to each user, this system adds functionalities for them.

**Admin**

The system administrator manages the overall system using this admin module. The admin can add or delete departments, admin per department, subjects per department, teachers, and students. This will allow them to handle all departments as a common admin. They can also create notices for teachers and students. This is useful for communicating important information to teachers and students, such as changes to the syllabus or important deadlines. They can add or delete subjects related to their department. As mentioned above admins can add admins, which means each department will also have its own admin, which will handle departmental activities. Admin or departmental admin can add subjects year-wise.

**Faculty**

Teachers create tests and add marks for each student using the faculty module. The test creation will include the creation of tests like Internal Assessment 1 (IA1), Internal Assessment 2 (IA2), End Semester Examination (ESE), Term Work (TW) during the whole syllabus, and Practical Examination. This test will give access to add marks for the same. It will ask for maximum marks for the test while creating it. Teachers have access to create tests and upload marks for the same only.

**Student**

Students can view their results and see subject lists using the student module. This is the only feature available for students.

**Chapter 3**

**Database**

MongoDB is a document-oriented database that stores data in JSON-like documents. This makes it a good choice for storing large amounts of unstructured data, such as student records and test scores. In our database, we have a total of six to seven tables:

**Admins:** This table stores information about the administrators of the system. The fields in this table include their username, password, registration number, name, email address, contact number, date of birth, joining year, and department. This is a common table for both admin and departmental admin.

**Departments:** This table stores information about the departments. The fields in this table include the department name and code.

**Faculties:** This table stores information about the faculties. The fields in this table include their username, password, name, gender, designation, department, contact number, email address, date of birth, joining year, and registration number.

**Marks:** This table stores information about the marks for each student in each subject. The fields in this table include the student ID, exam ID, and total marks.

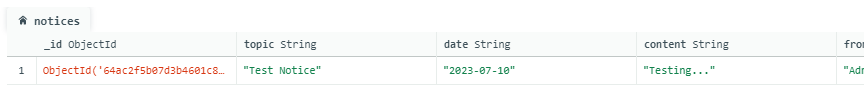
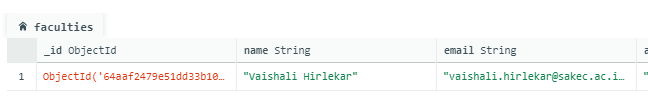
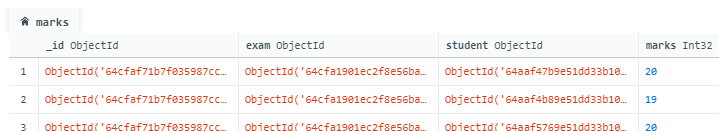
**Notices:** This table stores information about the notices that have been created. The fields in this table include the notice topic, content, from details, notice-for details, and date created.

**Students:** This table stores information about the students. The fields in this table include their username, password, name, contact number, email address, date of birth, current year, subjects, gender, parent details, department, section, batch, and registration number.

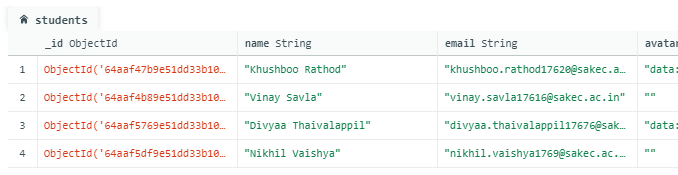
**Subjects:** This table stores information about the subjects that are being offered. The fields in this table include the subject name, subject code, total lectures, year, and department.

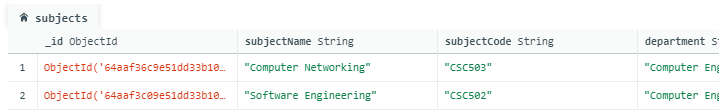
**Tests:** This table stores information about the tests that have been created by the faculties. The fields in this table include the test name, subject code, department, total marks, year, section, and date created.

*Figure 3.1 Admin Table*

*Figure 3.2 Notices Table*

*Figure 3.3 Faculties Table*

*Figure 3.4 Marks Table*

*Figure 3.5 Departments Table*

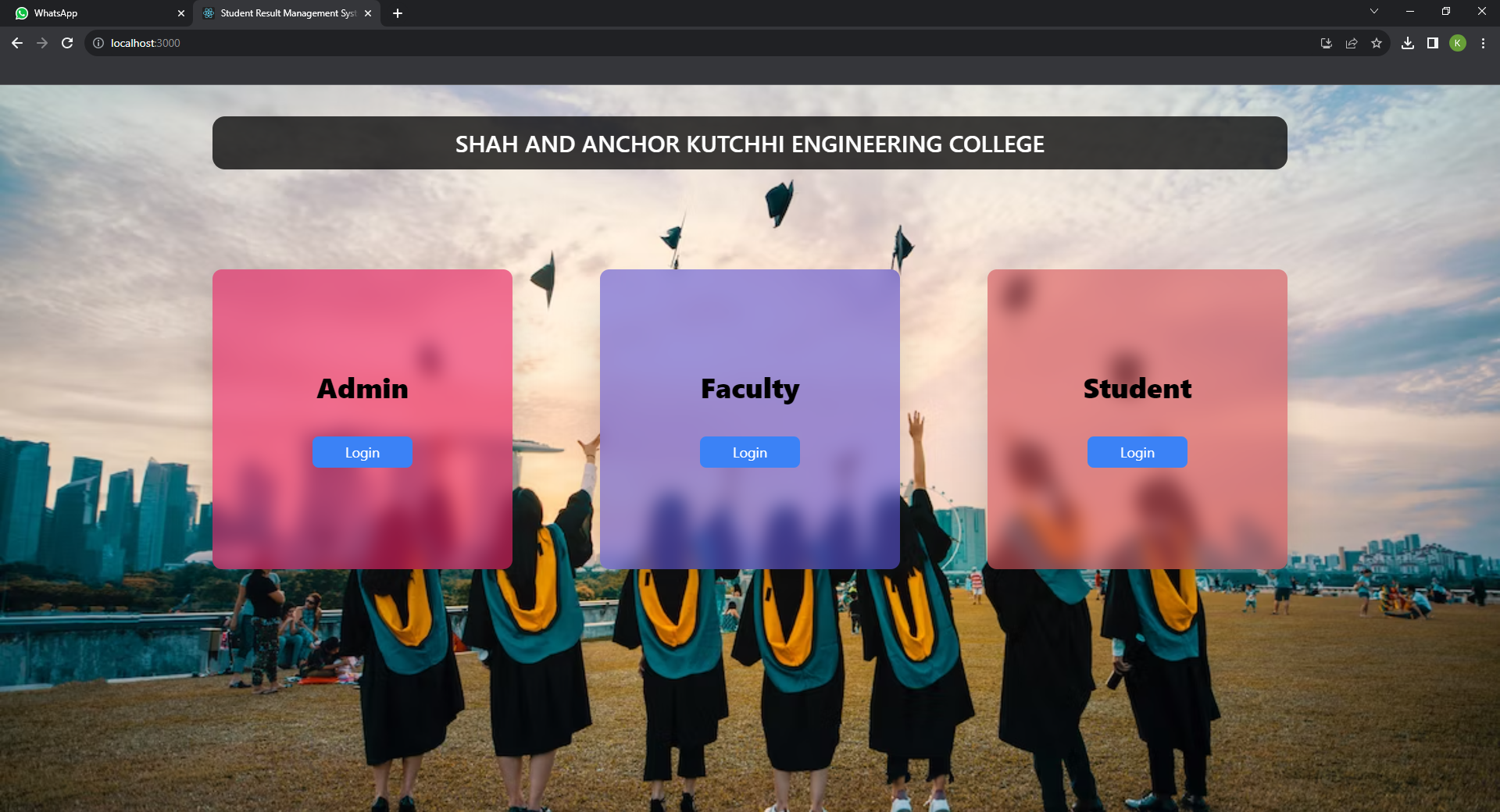
*Figure 3.6 Students Table*

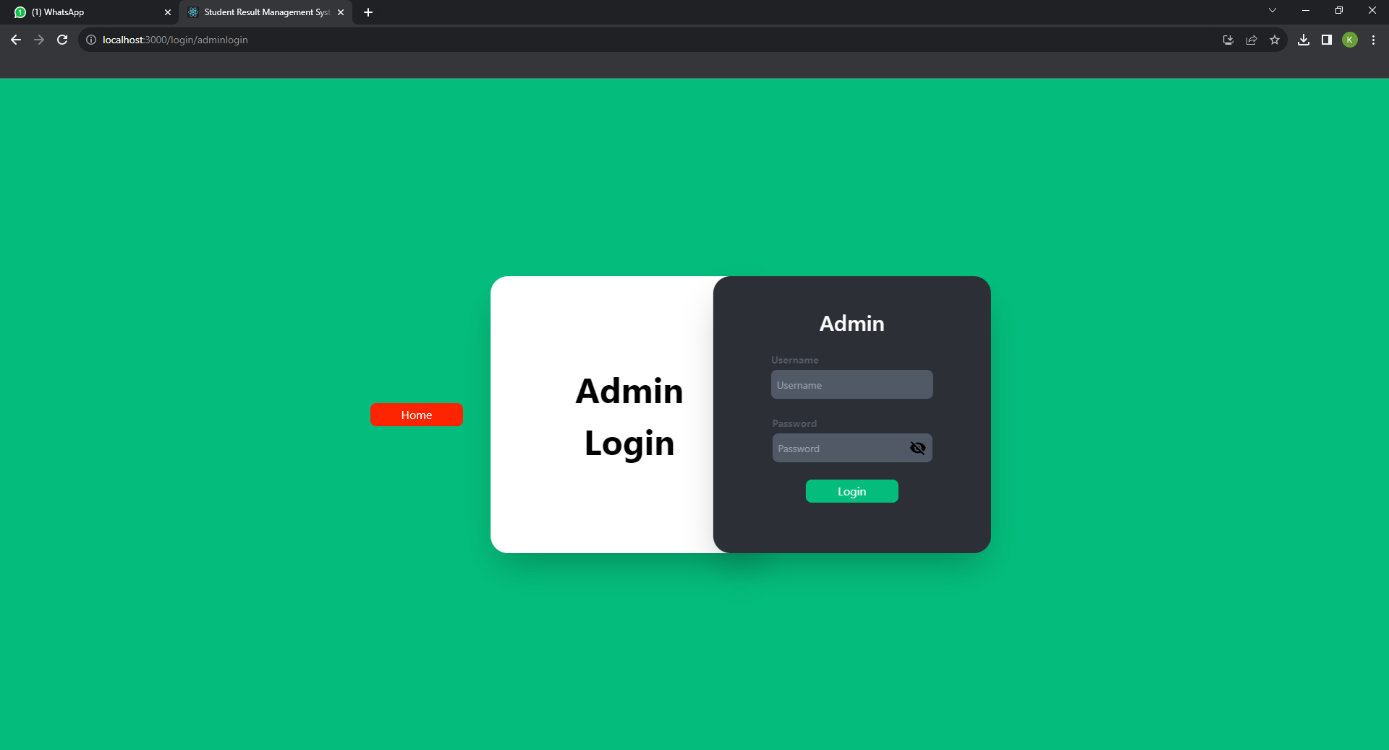
*Figure 3.7 Subjects Table*

**Chapter 4**

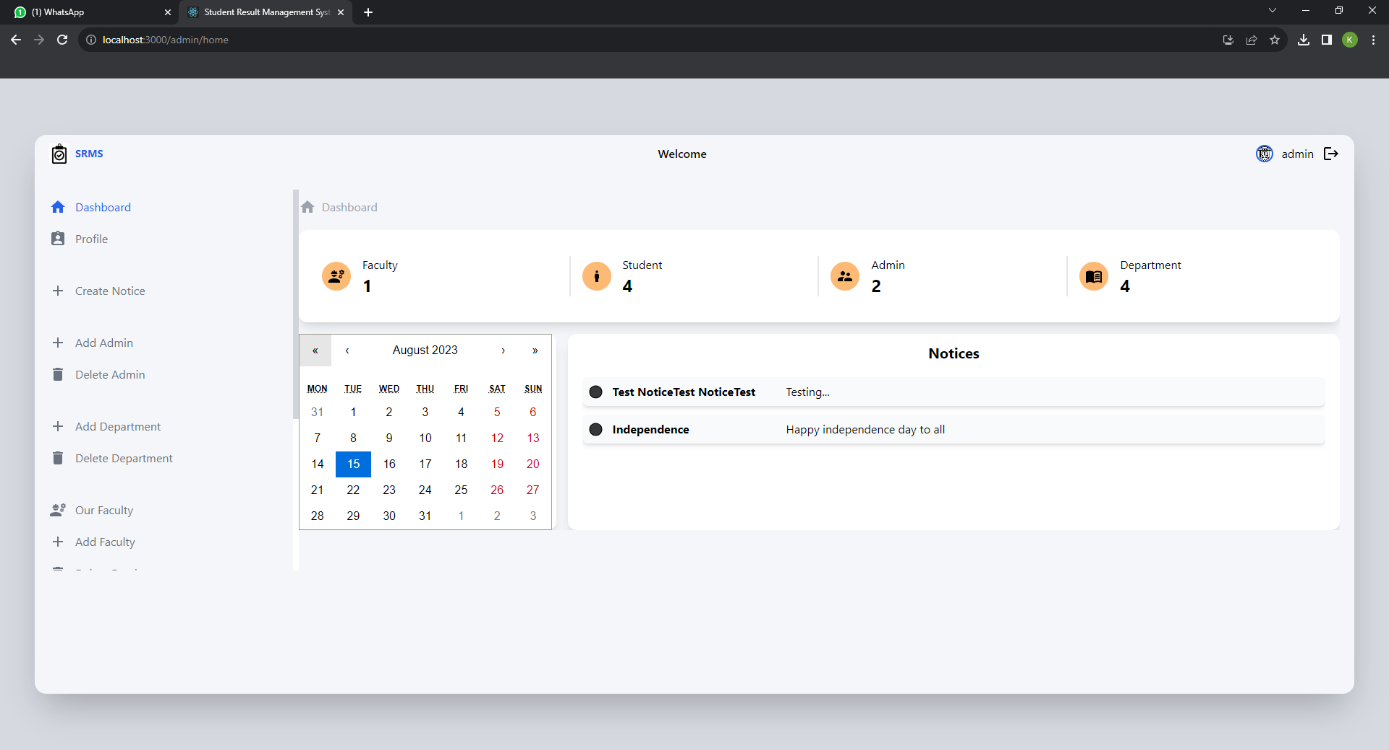
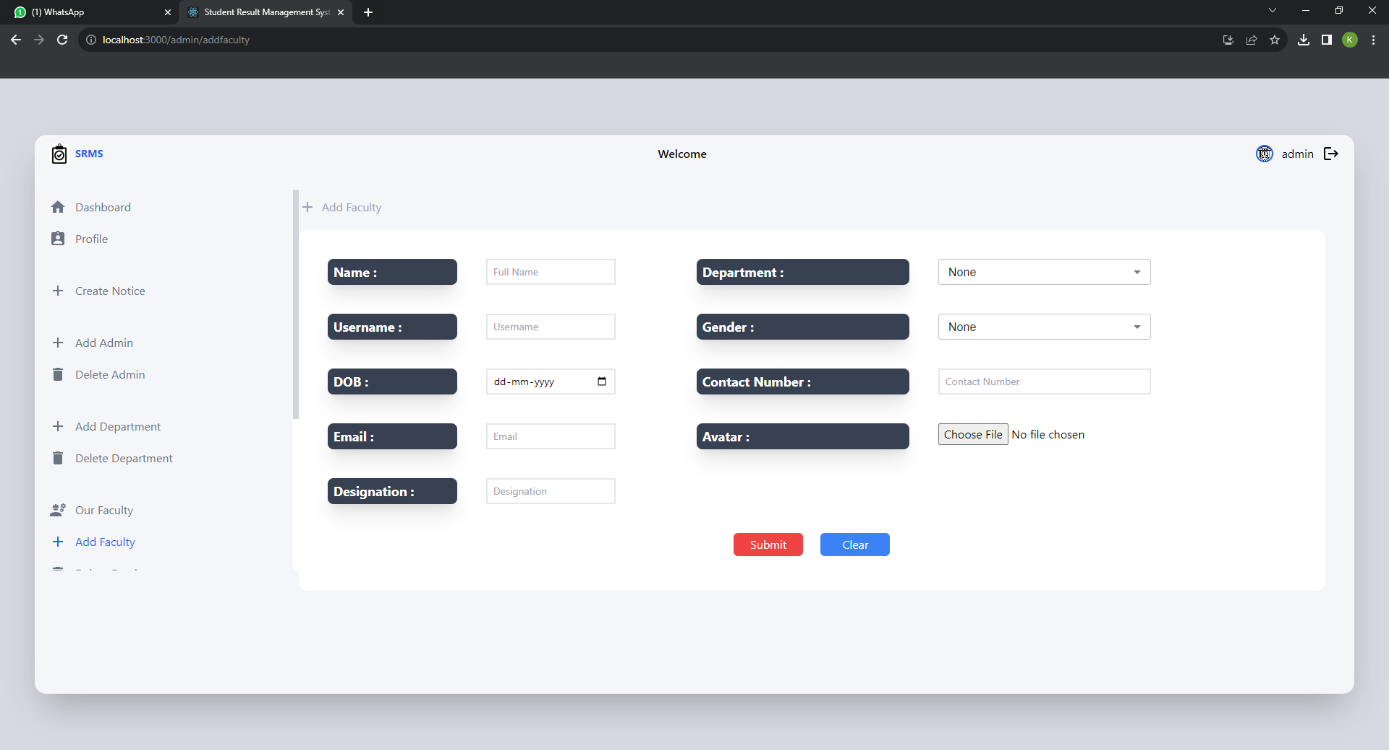
**Implementation**

**Chapter 5**

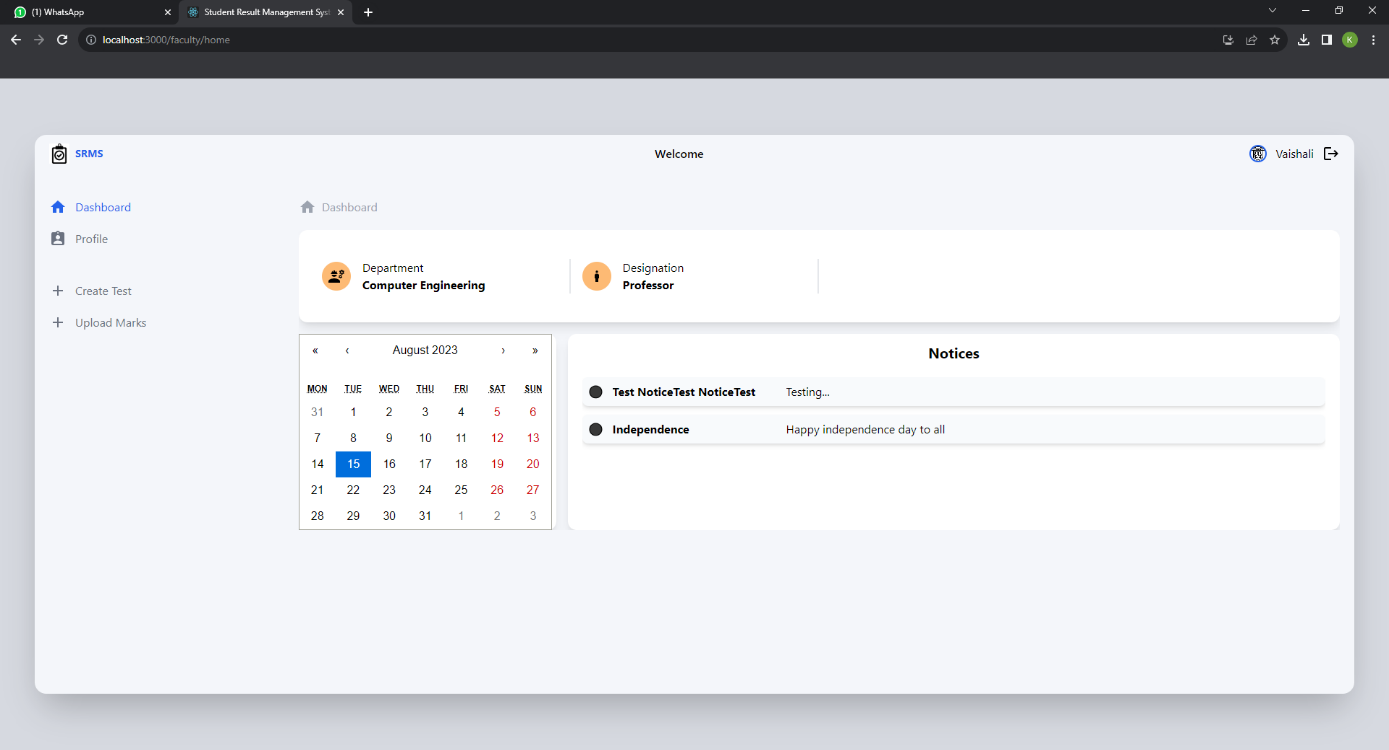
**Results**

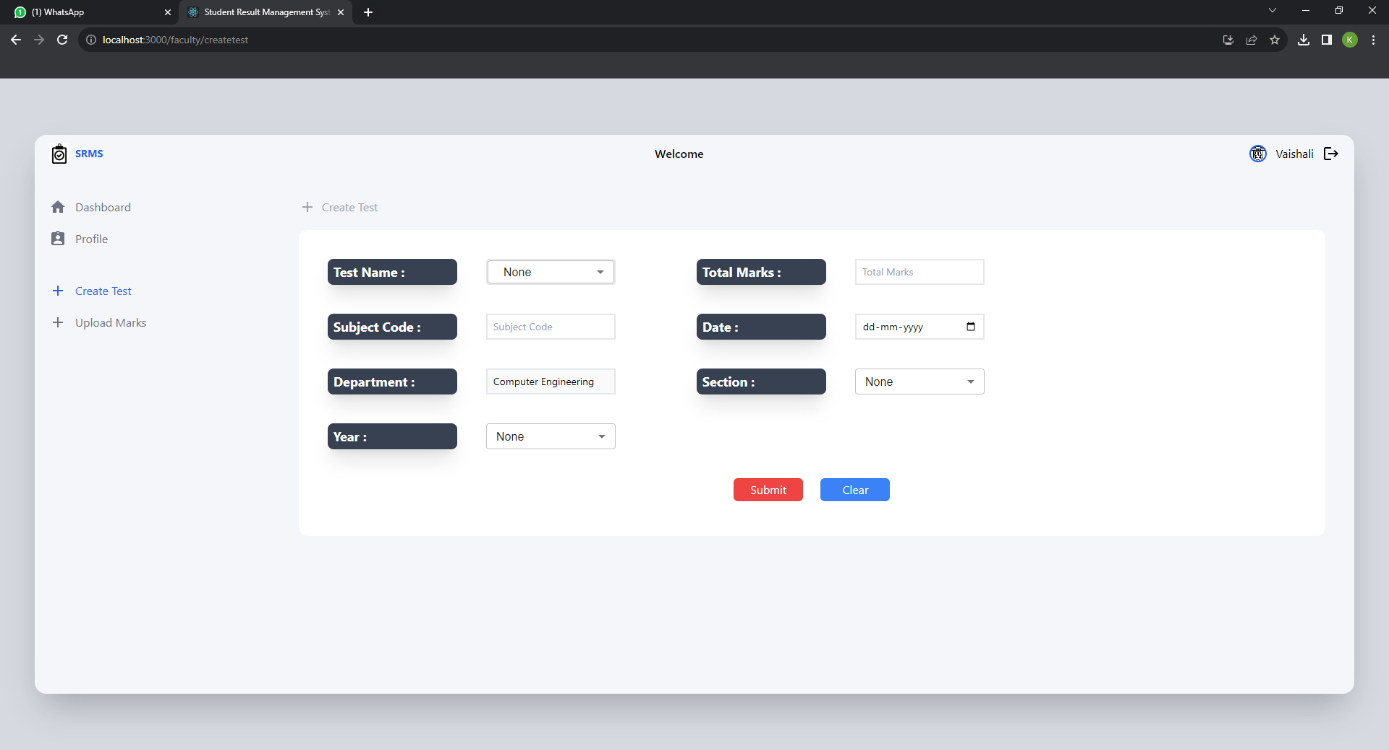
*Figure 5.1 Homepage*

*Figure 5.2 Admin login*

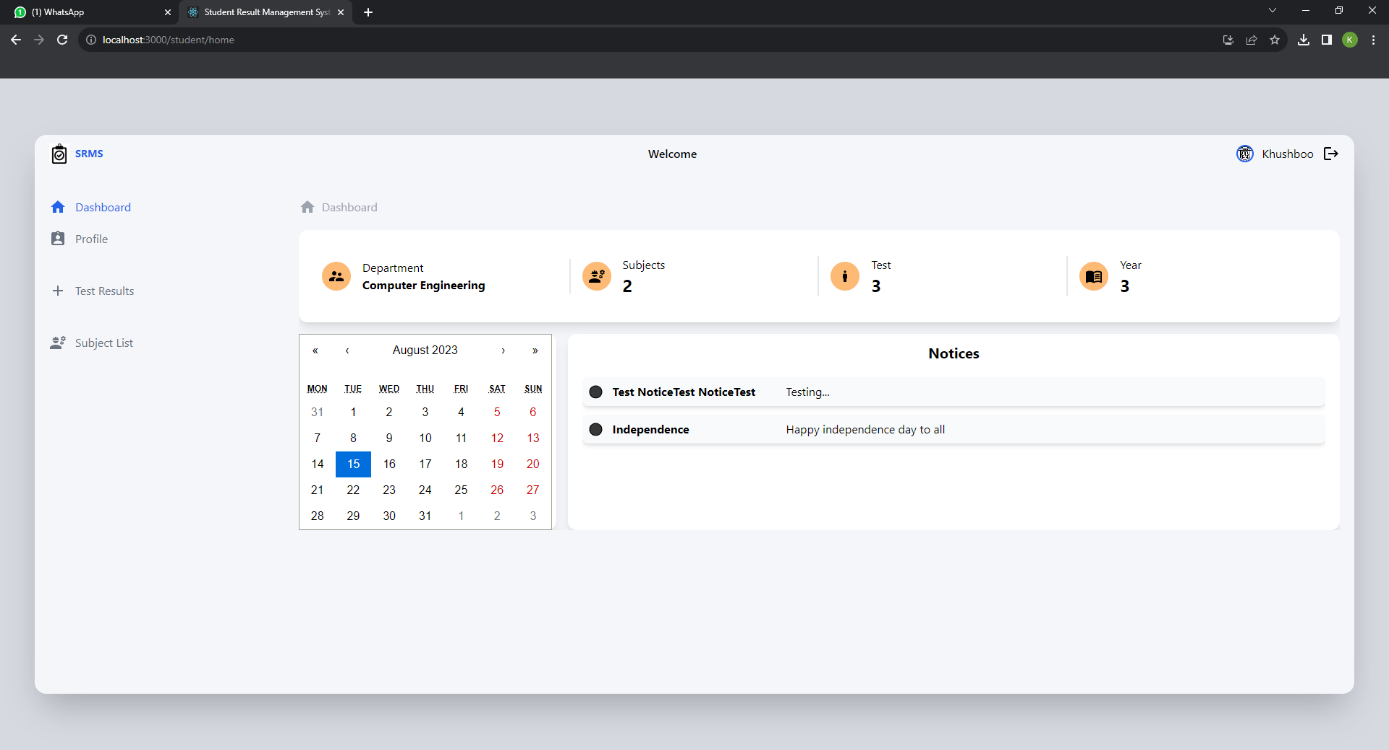
*Figure 5.3 Admin Dashboard*

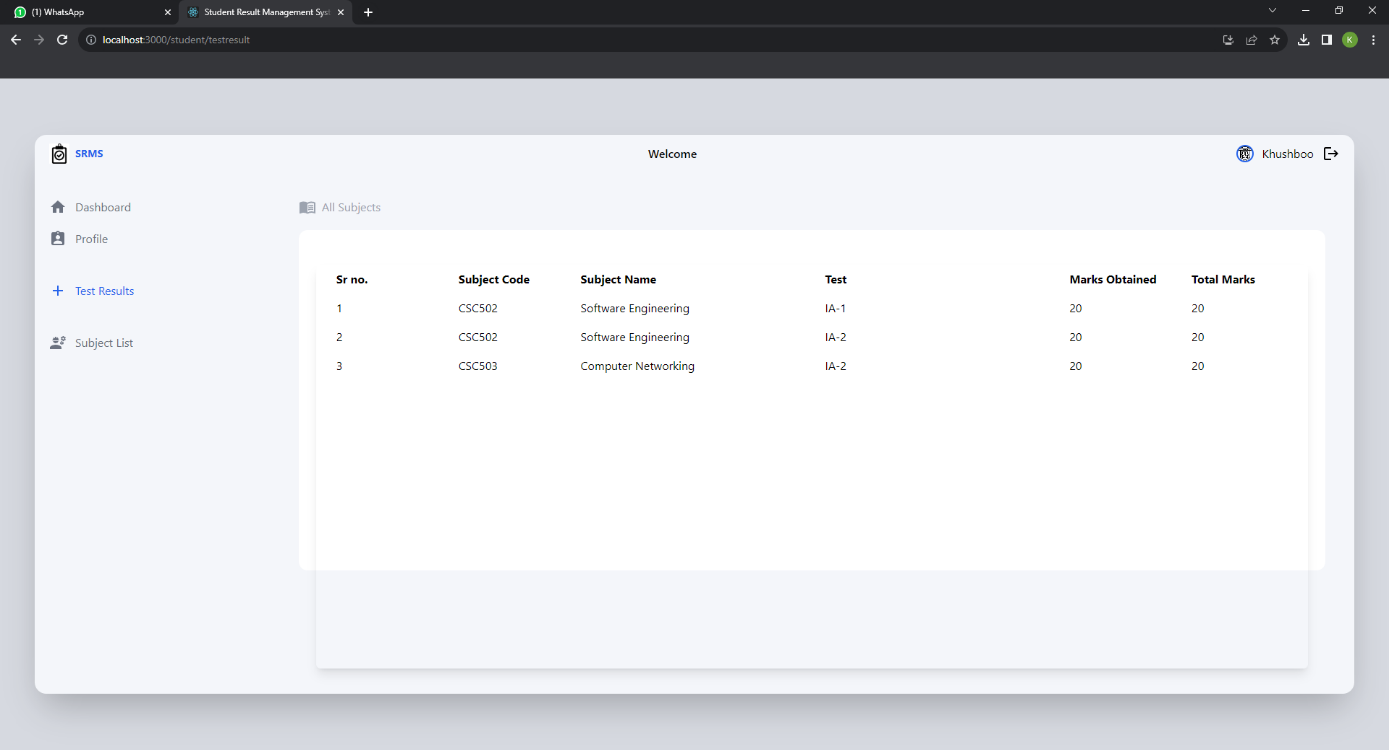
*Figure 5.4 Admin Add-faculty page*

*Figure 5.5 Faculty Dashboard*



*Figure 5.6 Faculty Create-test Page*

*Figure 5.7 Student Dashboard*



*Figure 5.8 Student Test-result page*

**Chapter 6**

**Conclusion**

In conclusion, utilizing the MERN (MongoDB, Express.js, React, Node.js) stack for building a Systematic Review Management System (SRMS) offers a comprehensive and efficient solution. The integration of these technologies addresses various aspects of SRMS development, ensuring seamless data management, user-friendly interfaces, and robust backend capabilities.

MongoDB serves as the database foundation, allowing for flexible and scalable data storage, essential for managing the extensive data associated with systematic reviews. Express.js facilitates the creation of a structured backend, enabling smooth communication between the front end and the database. Node.js supports real-time processing and enhances the overall performance of the system.

On the front, React delivers a dynamic and responsive user interface, enhancing user experience and interaction. Its component-based architecture streamlines development and maintenance efforts, making it easier to create complex features and manage UI elements effectively.

By combining these technologies, an SRMS gains the ability to streamline the review process, manage data efficiently, and provide an intuitive user experience for both researchers and administrators. However, it is crucial to recognize that the success of an SRMS depends not only on technology but also on a well-designed information architecture, robust security measures, and continuous user feedback for improvements.

In essence, adopting the MERN stack for building an SRMS empowers researchers with a comprehensive tool to manage and conduct systematic reviews effectively, ultimately contributing to the advancement of knowledge in various fields.

**References**

[1] S. R. Bharamagoudar, R. B. Geeta, and S. G. Totad, “Web-based student information management system,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 6, 2013.

[2]. Akinmosin James (2014). Automated Students Result Management System using Oracle’s Database, Forms, and Reports. Journal of Information Engineering and Applications.

[3] YongKang Xing, JiaPeng Huang, YongYao Lai, Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development, Conference Paper · February 2019 DOI: 10.1145/3313991.3314021 ResearchGate

[4] Sanchit Aggarwal, Jyoti Verma, Comparative analysis of MEAN stack and MERN stack, Sanchit Aggarwal et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 5, Issue 1, March 2018, pp. 127-132

[5] Suman Chatterjee, Manish Kumar Thakur, Smart Collage Management System International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, www.ijert.org NCRAEM - 2019 Conference Proceedings

[6] Hema Krishnan, Research Scholar, CUSAT, MongoDB – a comparison with NoSQL databases, International Journal of Scientific & Engineering Research, Volume 7, Issue 5, May-2016 ISSN 2229-5518