National Institute of Technology Mizoram

Student Result Management System

Project Report

Submitted by:

Vikash Kumar Kharwar

Roll No: BT22CS035

Submitted to:

Mr. Ashish Singh Patel

Under the guidance of:

Mr. Sourav Pandey

Ms. Atoholi Chisho

November 4, 2024

DECLARATION

This project report is submitted to the National Institute of Technology Mizoram, **Respected Mr.Ashish Singh Patel**.I declare that this project report has not been submitted elsewhere for any degree, diploma, or publication.

By:

Vikash Kumar Kharwar (BT22CS035)

Contents

INTRODUCTION

The Student Result Management System is designed to streamline the process of recording, updating, and managing student academic results. The system incorporates modern technologies such as Node.js, Express.js, React.js, and MySQL to provide a robust platform for educational institutions.

The increasing complexity of student data management in educational institutions necessitates a system that not only simplifies processes but also enhances data integrity and accessibility. This project aims to develop an intuitive and efficient application that facilitates the management of student results, attendance, and overall academic performance.

The primary objectives of the system include:

- To automate the processes of recording and retrieving student results.
- To provide a user-friendly interface for students, faculty, and administrators.
- To ensure data security and privacy through role-based access controls.
- To enable the generation of comprehensive reports for analysis and decision-making.

The Student Result Management System (SRMS) simplifies the maintenance of student records for educational institutions. It provides functionalities to manage student information, academic reports, course details, and more. SRMS ensures efficient and accurate data management, aiding in attendance tracking, exam result maintenance, and student performance analysis. It features a secure, online interface for faculty and administrative use. Reports and notifications can be easily generated, improving communication between students and staff.

TECHNOLOGY OVERVIEW

This project employs a stack of modern web technologies to ensure performance, scalability, and maintainability.

Frontend

React.js: A JavaScript library for building user interfaces, allowing for a dynamic and responsive design. React's component-based architecture enhances code reusability and simplifies the development process.

Key Features of React.js

- **Virtual DOM:** Improves performance by minimizing direct manipulation of the DOM.
- **Component Lifecycle:** Provides hooks to manage component states and lifecycle events efficiently.
- **State Management:** Facilitates local and global state management through Context API and Redux.

Backend

Node.js: A JavaScript runtime that enables the development of server-side applications. Node.js is known for its non-blocking I/O model, which makes it lightweight and efficient.

Express.js: A web application framework for Node.js that simplifies API development. Express provides robust features for building web and mobile applications.

Key Features of Express.js

- **Middleware Support:** Easily integrates with third-party middleware for extended functionalities.
- **Routing:** Simplifies the process of defining routes and handling HTTP requests.
- **Error Handling:** Provides built-in error handling mechanisms for a smooth user experience.

Database

MySQL: A relational database management system used for storing student and result data. MySQL is preferred for its reliability, scalability, and ease of use.

Key Features of MySQL

- **ACID Compliance:** Ensures data integrity and reliability through transactions.
- **Structured Query Language (SQL):** Facilitates powerful querying capabilities for data manipulation.
- **Security:** Offers robust security features to safeguard sensitive data.

SYSTEM HIERARCHY

The system hierarchy illustrates the various components of the Student Result Management System and their interactions. This structure ensures a clear organization of responsibilities and data flow.

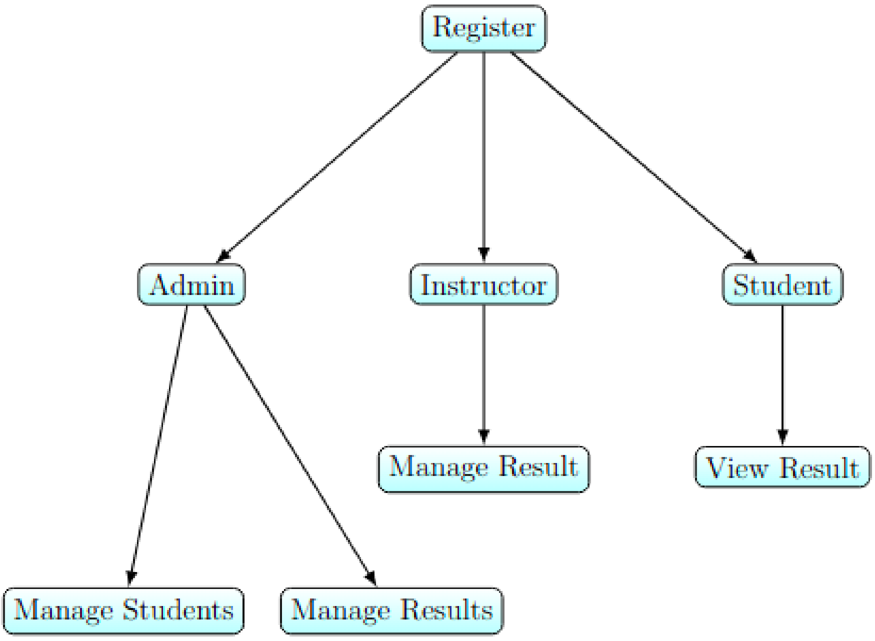
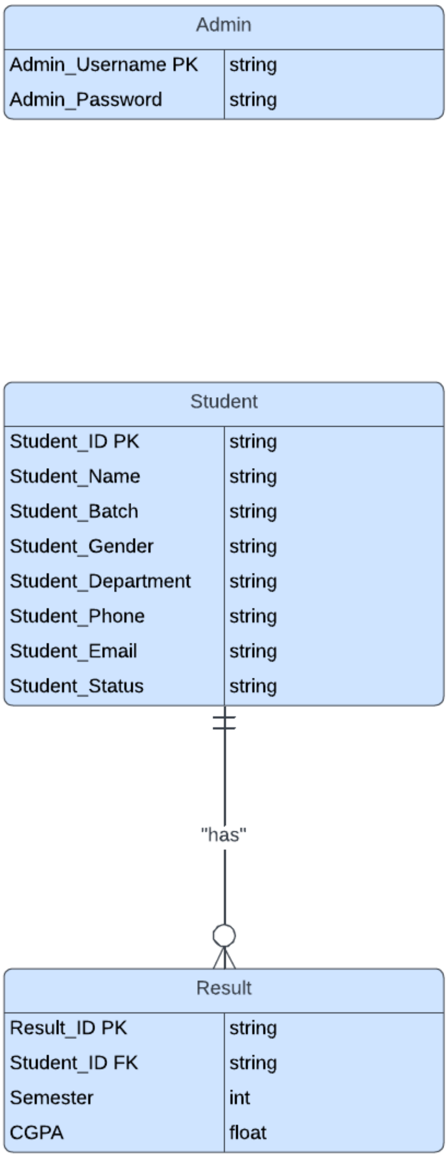


Figure 1: System Hierarchy

Components of the System Modals and Schema

- **Database Layer:** MySQL database that stores all application data securely.



Admin	
Admin_Username PK	string
Admin_Password	string

Student	
Student_ID PK	string
Student_Name	string
Student_Batch	string
Student_Gender	string
Student_Department	string
Student_Phone	string
Student_Email	string
Student_Status	string

Result	
Result_ID PK	string
Student_ID FK	string
Semester	int
CGPA	float

Figure 2: Important Models and schemas

- **Client-Side:** The frontend application built using React.js that interacts with users.
- **Server-Side:** The backend application built on Node.js and Express.js that handles business logic and API requests.

Data Flow

- User interactions on the frontend trigger API requests to the backend.
- The backend processes these requests, performing necessary business logic and interacting with the database.
- Responses from the database are sent back to the frontend, updating the user interface accordingly.

ENTITY-RELATION MODEL

The Entity-Relation (ER) model serves as a blueprint for the database design, illustrating the relationships between different data entities in the system.

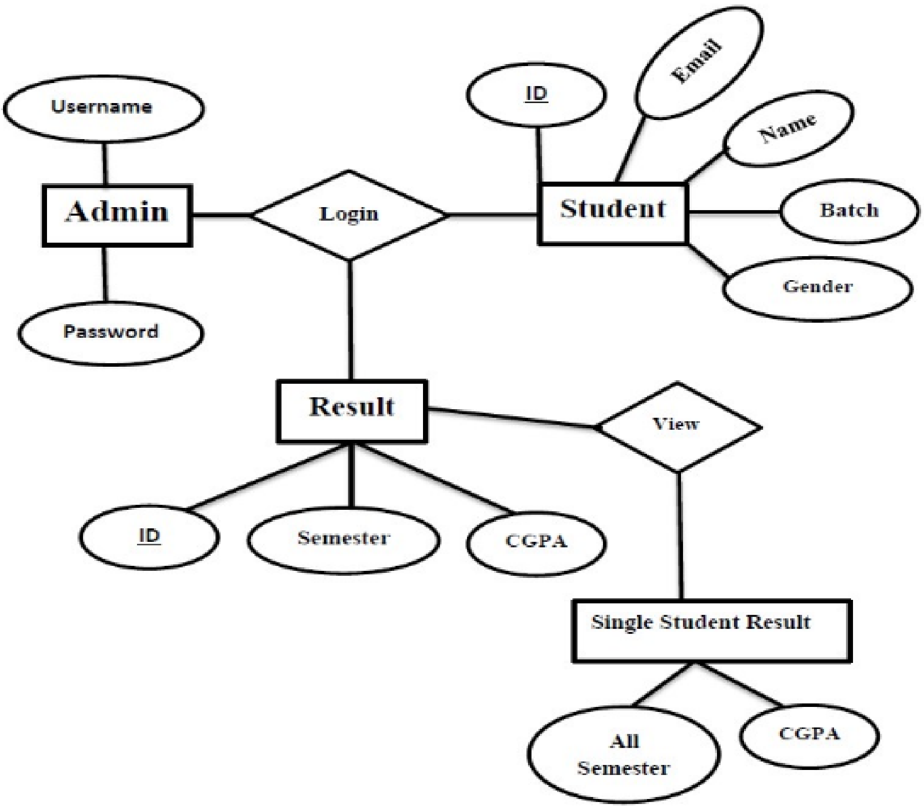


Figure 3: Entity-Relation Model

Entities and Relationships

- **Students:** Contains information about students, including their names, roll numbers, and enrolled courses.
- **Courses:** Represents courses offered, detailing course codes, titles, and credits.
- **Results:** Stores student grades and performance metrics, linked to both students and courses.
- **Instructors:** Holds data about instructors responsible for teaching courses, including their qualifications and contact information.

Normalization

The database schema is normalized to eliminate redundancy and ensure data integrity. This process includes:

- **First Normal Form (1NF):** Ensures that all entries are atomic and that each entry in a column is unique.
- **Second Normal Form (2NF):** Addresses partial dependencies of data on the primary key.
- **Third Normal Form (3NF):** Eliminates transitive dependencies, ensuring that non-key attributes are only dependent on the primary key.

SNAPSHOTS

This section presents snapshots of the key pages in the Student Result Management System.

Register Page

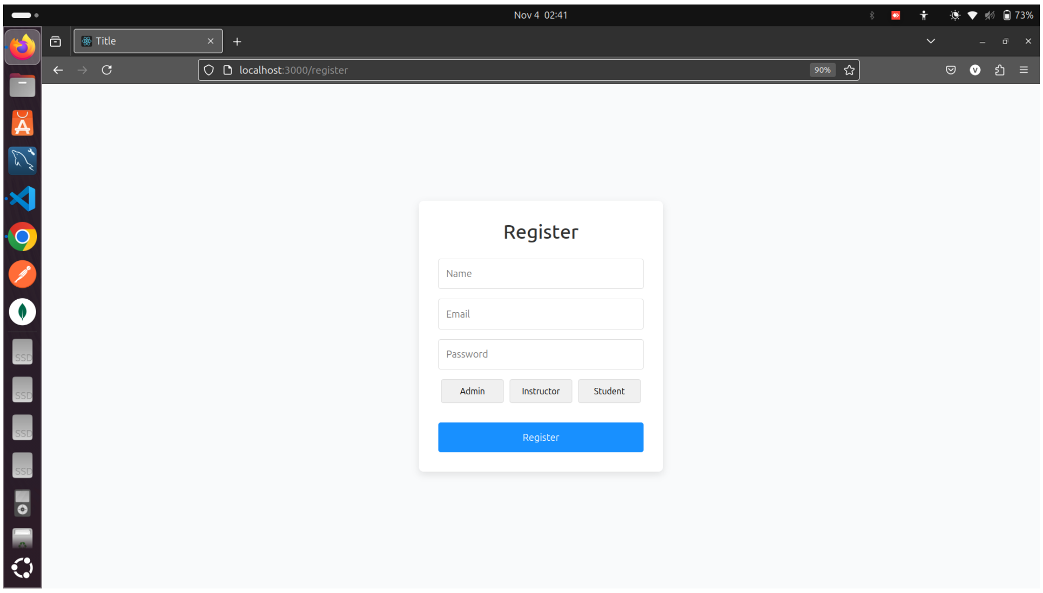
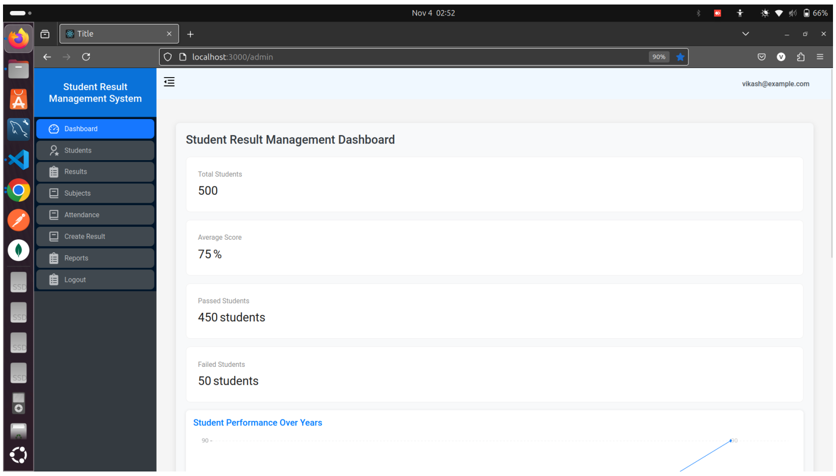


Figure 4: Register Page

Home Page



Student Result Management Dashboard

Total Students
500

Average Score
75 %

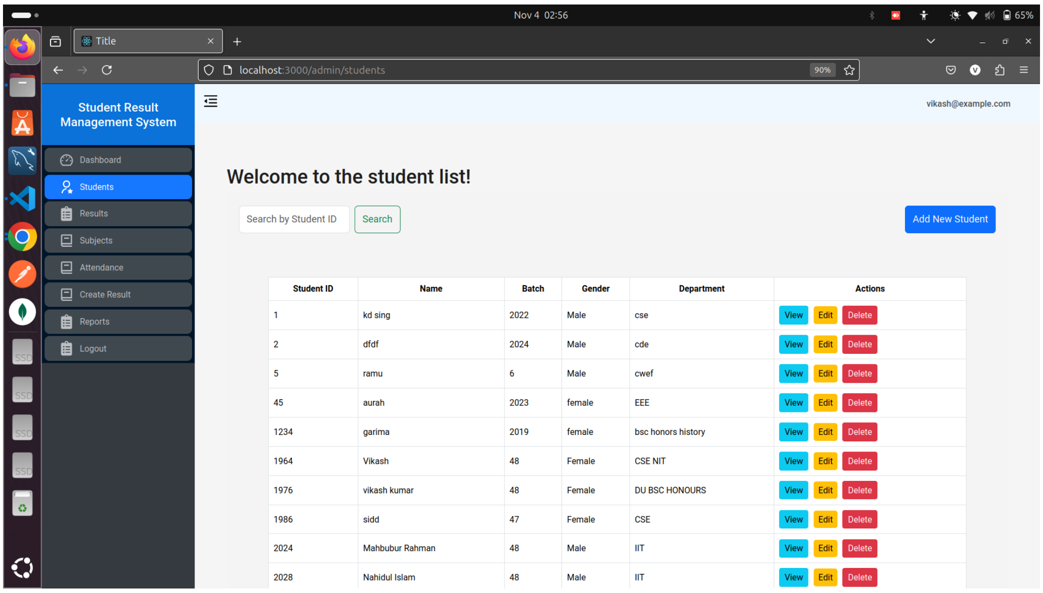
Passed Students
450 students

Failed Students
50 students

Student Performance Over Years

Figure 5: Home Page

Admin-student Page

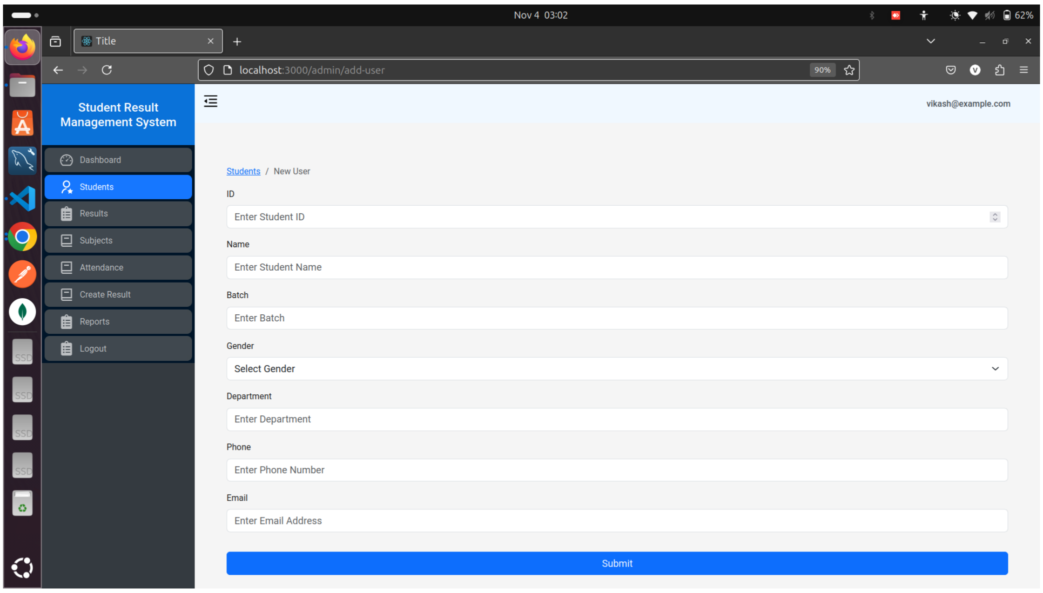


Welcome to the student list!

Student ID	Name	Batch	Gender	Department	Actions
1	kd sing	2022	Male	cse	View Edit Delete
2	dfdf	2024	Male	cde	View Edit Delete
5	ramu	6	Male	cwef	View Edit Delete
45	aurah	2023	female	EEE	View Edit Delete
1234	garima	2019	female	bsc honors history	View Edit Delete
1964	Vikash	48	Female	CSE NIT	View Edit Delete
1976	vikash kumar	48	Female	DU BSC HONOURS	View Edit Delete
1986	sidd	47	Female	CSE	View Edit Delete
2024	Mahbubur Rahman	48	Male	IIT	View Edit Delete
2028	Nahidul Islam	48	Male	IIT	View Edit Delete

Figure 6: Admin-student Page

Add-student Page



Students / New User

ID

Enter Student ID

Name

Enter Student Name

Batch

Enter Batch

Gender

Select Gender

Department

Enter Department

Phone

Enter Phone Number

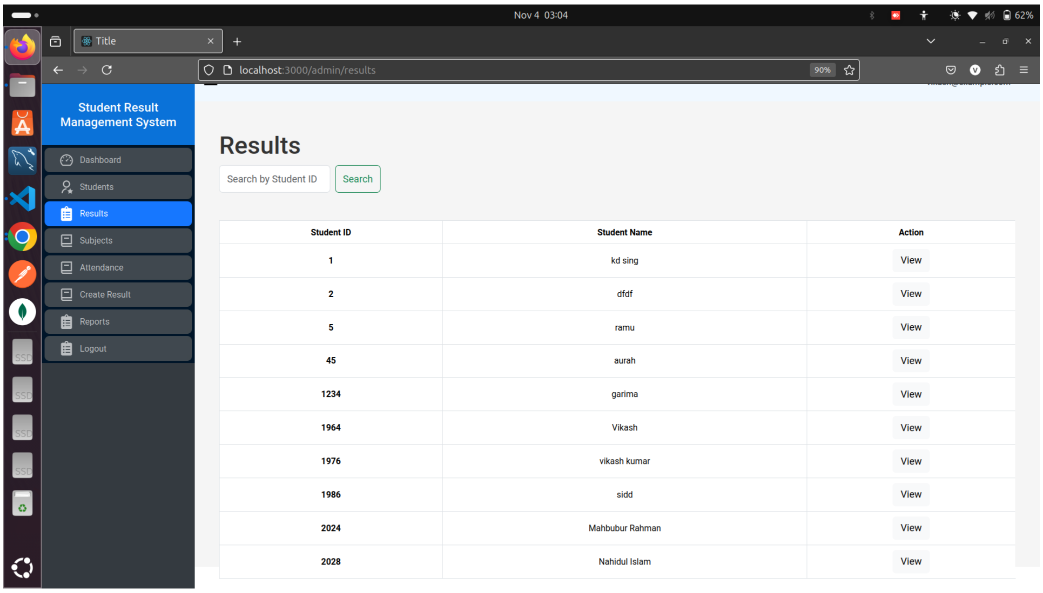
Email

Enter Email Address

Submit

Figure 7: Add-student Page

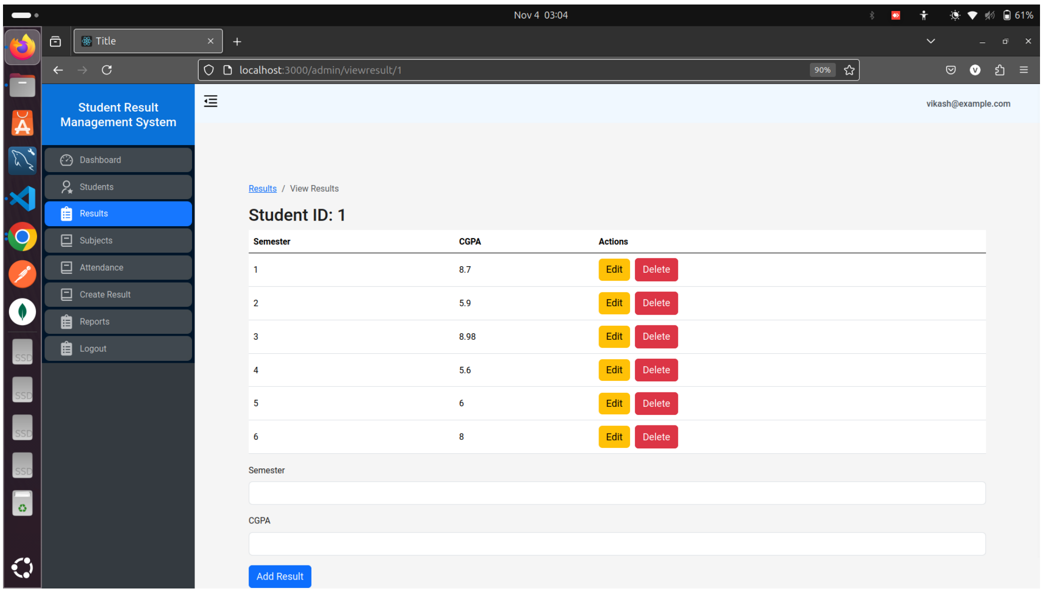
Result Page



Student ID	Student Name	Action
1	kd sing	View
2	dfdf	View
5	ramu	View
45	aurah	View
1234	garima	View
1964	Vikash	View
1976	vikash kumar	View
1986	sidd	View
2024	Mahbubur Rahman	View
2028	Nahidul Islam	View

Figure 8: Results Page

Specific student Result Page



Results / View Results

Student ID: 1

Semester	CGPA	Actions
1	8.7	Edit Delete
2	5.9	Edit Delete
3	8.98	Edit Delete
4	5.6	Edit Delete
5	6	Edit Delete
6	8	Edit Delete

Semester

CGPA

Add Result

Figure 9: Specific student Result Page

Create Result Page

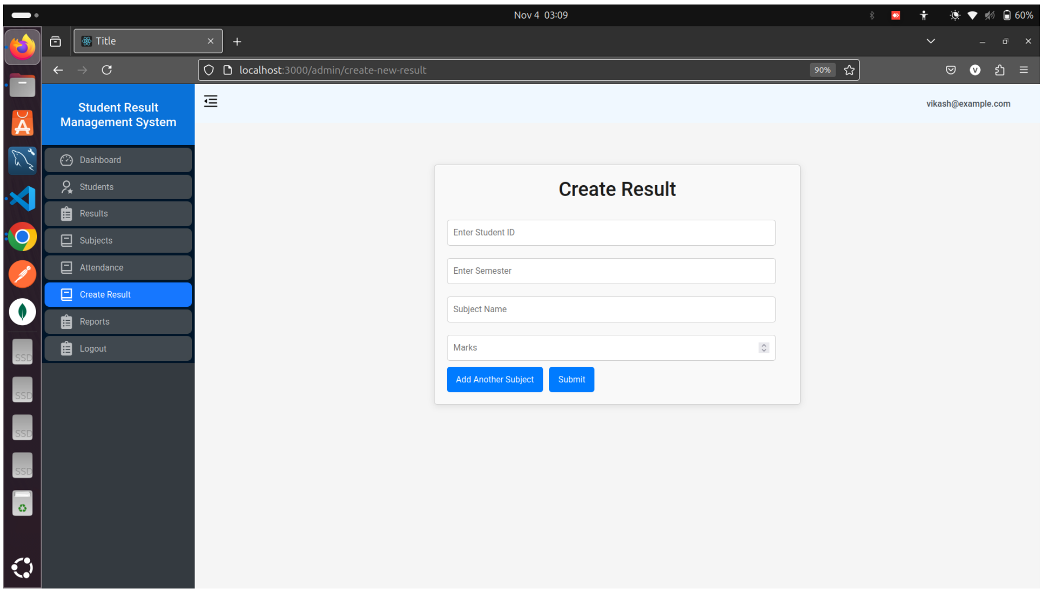
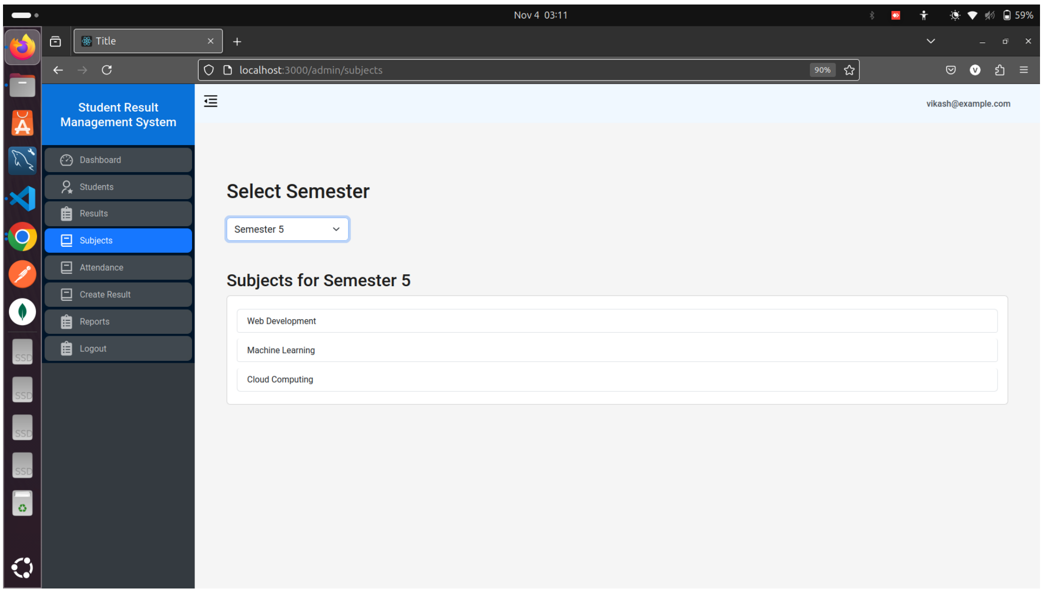


Figure 10: Create Result Page

Subject Page



Select Semester

Semester 5

Subjects for Semester 5

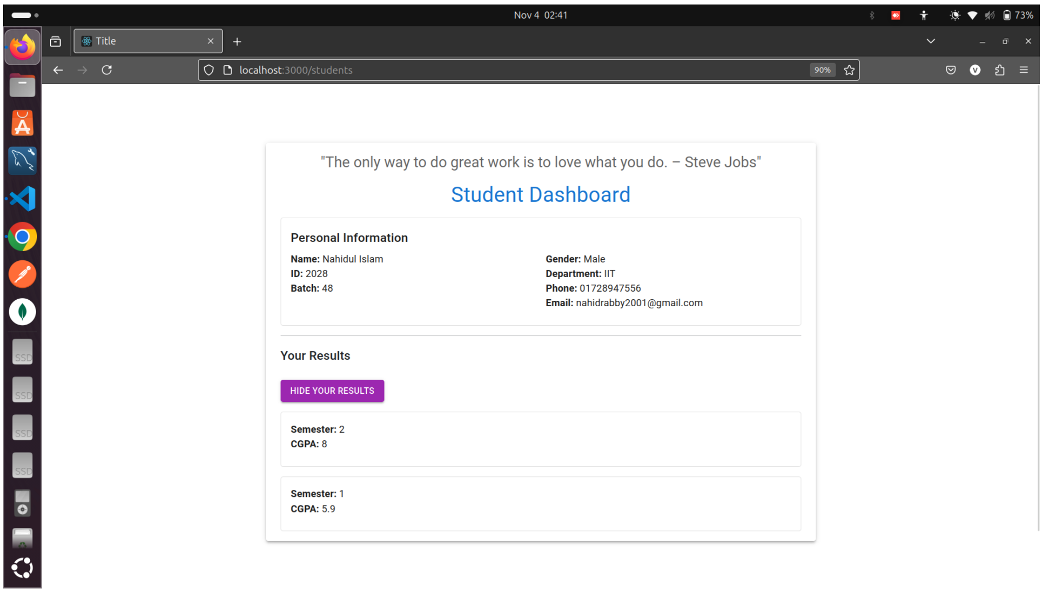
Web Development

Machine Learning

Cloud Computing

Figure 11: Subject Page

Student Dasboard Page



"The only way to do great work is to love what you do. – Steve Jobs"

Student Dashboard

Personal Information

Name: Nahidul Islam
ID: 2028
Batch: 48

Gender: Male
Department: IIT
Phone: 01728947556
Email: nahidrabby2001@gmail.com

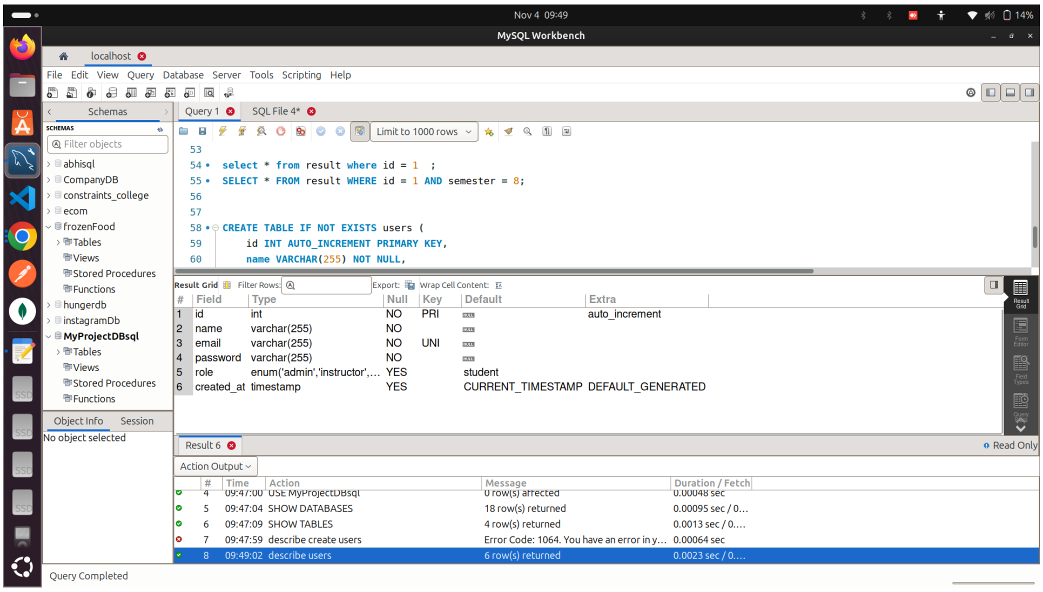
Your Results

Semester: 2
CGPA: 8

Semester: 1
CGPA: 5.9

Figure 12: Student Dasboard Page

Databse Page



```sql

select * from result where id = 1  ;
SELECT * FROM result WHERE id = 1 AND semester = 8;


CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
```

#	Field	Type	Null	Key	Default	Extra
1	id	int	NO	PRI	NULL	auto_increment
2	name	varchar(255)	NO		NULL	
3	email	varchar(255)	NO	UNI	NULL	
4	password	varchar(255)	NO		NULL	
5	role	enum('admin','instructor',...	YES		student	
6	created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

	#	Time	Action	Message	Duration / Fetch
✓	4	09:47:00	USE MyProjectDBsql	0 row(s) affected	0.00048 sec
✓	5	09:47:04	SHOW DATABASES	18 row(s) returned	0.00095 sec / 0....
✓	6	09:47:09	SHOW TABLES	4 row(s) returned	0.0013 sec / 0....
✗	7	09:47:59	describe create users	Error Code: 1064. You have an error in y...	0.00064 sec
✓	8	09:49:02	describe users	6 row(s) returned	0.0023 sec / 0....

Figure 13: Databse Page

PROJECT UPDATES

Enhanced CRUD Operations

1.1 Enhanced CRUD Operations for Admin

Create Student Results:

- **Functionality:** Allow admin to input the details of the new student to admit to the academic system, and the system will automatically provide a particular ID or the admin may provide it.
    1. Capture student details through a user-friendly form.
    2. Store data in the database, ensuring relationships with the Students and Courses tables.

Retrieve Student Results

Functionality:

- Allow users to retrieve results based on filters such as Roll Number, Course, and Semester.

Update and Delete Operations

Functionality:

- Provide functionalities for updating existing records and deleting students from the database when necessary.

Authorized Access

Functionality:

- Authorized to access all the permissions that have to the instructor and students.

1.2 Enhanced CRUD Operations for Instructor

Create Student Results:

- **Functionality:** Allow the instructor to input the marks for each subject. The system will automatically calculate and store the GPA based on the results.

    1. Capture student details through a user-friendly form.
    2. Store data in the database, ensuring relationships with the Students and Courses tables.

Retrieve Student Results

Functionality:

- Allow instructors to retrieve results based on filters such as Roll Number, Course, and Semester.

Update and Delete Operations

Functionality:

- Provide functionalities for updating existing records and deleting students' results from the database when necessary.

1.3 Enhanced CRUD Operations for Students

Retrieve Student Information

Functionality:

- Allow Students to retrieve their information and result based on their email and student ID

CONCLUSION

The Student Result Management System presents a robust solution for managing academic records efficiently. By leveraging modern web technologies, the system ensures that data management processes are streamlined and user-friendly. This project demonstrates how technology can be harnessed to improve educational administrative processes, ultimately benefiting both students and educators.

Future enhancements could include the integration of machine learning algorithms for predicting student performance trends and an analytics dashboard for real-time insights into academic data.

Future Planning

The following features are planned for future iterations of the Student Result Management System:

- **Email Integration:** Implement an email feature to streamline communication for students. This will enable students to receive important details directly to their email, including notifications and updates. Additionally, provide a "Forgot Password" feature that allows students to securely consult with the admin to reset their password.
- **Attendance Management:** Integrate a feature to track and manage student attendance, providing reports for faculty.
- **Student Performance Analytics:** Implement tools to analyze student performance over time, identifying trends and areas for improvement.
- **Social Networking Features:** Develop a module to facilitate interaction among students and faculty, enhancing collaboration.
- **Online Tracker and Chat Functionality:** Introduce a messaging system for students to communicate with faculty and peers.
- **Online Institution Module:** Expand the system to include functionalities for managing online courses and related resources.

REFERENCES

- ”NodeJS About — Node.js”, Node.js, 2022. [Online]. Available: `nodejs.org`.
- ”Home”, Html.com, 2022. [Online]. Available: `html.com`.
- ”CSS”, W3schools.com, 2022. [Online]. Available: `w3schools.com`.
- ”JavaScript”, Javascript.com, 2022. [Online]. Available: `javascript.com`.
- ”Handlebars”, Handlebarsjs.com, 2022. [Online]. Available: `handlebarsjs.com`.
- ”SQL”, En.wikipedia.org, 2022. [Online]. Available: `wikipedia.org`.