A Project report on

Navigating Courses at your fingertips with STUDENT 360

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

Rishab Agarwal
(20H51A0547)

P Satwik
(20H51A0543)

Akshith Akkali
(20H51A05M7)

Under the esteemed guidance of

Mr. M . Shiva Kumar

(Assistant Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^{+} Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase I report entitled **"Navigating Courses at your Fingertips with Student 360"** being submitted by Rishab Agarwal (20H51A0547),P Satwik(20H51A0543),Akshith Akkali(20H51A05M7) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr.M. Shiva Kumar	**Dr. Siva Skandha Sanagala**	
Assistant Professor	**Associate Professor and HOD**	**EXTERNAL EXAMINER**
Dept. of CSE	**Dept. of CSE**	

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful **to M. Shiva Kumar, Assistant Professor** , Department of Computer Science & Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. S. Siva Skandha,** Head of the Department of Computer Science & Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful **to Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Dept Name for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

Rishab Agarwal 20H51A0547

P Satwik 20H51A0543

Akshith Akkali 20H51A05M7

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In colleges and Universities, faculty face a lot of problems collecting information from students about the completion and choosing of Open Electives and Professional electives at the beginning of the semester. Also, most of the students select the subjects which were already attained by them, not realizing until the final semester. Thus, resulting in not satisfying the required credit score and taking an extra semester wasting their valuable time. Student360 is a project that focuses on students completing their Open Electives and Professional electives, faculty managing students academic profiles and admins adding subjects elected by students and also regarding their academics. The project works as a reminder for students to check their credit score and wrap up the subjects which were not completed by them previously.

Conversely, the Student Registration System for universities maintains personalized interactions between counselors and students, fostering immediate problem resolution. However, limitations arise in data analysis capabilities and manual processes. Similarly, the College Course Registration System improves accessibility and user experience by enabling online registration. Yet, technical expertise requirements, development costs, and data security concerns pose challenges.

Overall, the adoption of online registration systems represents a significant advancement in educational institution efficiency and user experience. However, addressing technical proficiency gaps and financial obstacles is crucial for sustained benefits.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1. Problem Statement

The current manual course registration system within the educational institution is beset with several challenges, hindering the seamless enrollment process for students. These challenges include:

1**) Complex and Time-Consuming Process**: The manual course registration process is intricate and time-consuming, involving multiple steps such as form filling, approvals, and data entry. This complexity often leads to errors and delays in the registration process.

2) **Limited Accessibility**: Physical registration forms and records are difficult to access for both students and administrative staff.

3) **Limited Scalability:** With a manual system, accommodating a growing number of students becomes increasingly challenging. The manual process cannot efficiently handle a large volume of course registrations, leading to bottlenecks during peak registration periods.

Evaluate Current System Challenges: Understand the specific challenges and limitations faced by the educational institution's current manual course registration system, including issues related to data accuracy, efficiency, and resource utilization. Analyze User Requirements: Identify the specific needs and preferences of both students and administrative staff regarding the course registration process. Assess Data Accuracy and Integrity: Investigate the accuracy and integrity of student and course data processed through the automated system. Analyze Resource Utilization: Evaluate the utilization of human resources and physical space within the institution after implementing the automated system. Examine Cost-Effectiveness: Conduct a cost-benefit analysis comparing the expenses associated with the manual system and the automated Course Registration System.

1.2. Research Objective:

The primary objective of this research is to investigate the implementation and impact of online student course registration systems in educational institutions. Specifically, the study aims to: Assess the effectiveness of the Online Student Course Registration System in streamlining and automating the course registration process, with a focus on reducing administrative burdens and enhancing efficiency.

Evaluate the unique merits and challenges associated with the Student Registration System for universities, particularly regarding personalized interactions between counselors and students, and the limitations in data analysis capabilities and manual processes. Examine the benefits and obstacles of the College Course Registration System in improving accessibility and user experience through online registration, including technical expertise requirements, development costs, and data security concerns. Explore the overall impact of adopting online registration systems on efficiency and user satisfaction within educational institutions, and identify strategies to address technical proficiency gaps and financial obstacles for sustained benefits.

By achieving these objectives, the research aims to provide insights into the effectiveness and challenges of online student course registration systems, offering recommendations for enhancing their implementation and maximizing their benefits in educational settings.

1.3. Project Scope

The Student360 project aims to address the challenges faced by colleges and universities in managing elective selections and academic profiles. It encompasses the following key areas:

1) Elective Selection: The project focuses on assisting students in making informed choices for Open Electives and Professional Electives at the beginning of the semester. It provides a user-friendly platform for students to select electives.

2) Academic Profile Management: Faculty and administrators are empowered with tools to efficiently manage students' academic profiles, including elective choices, progress tracking, and credit scoring.

3) Administrative Functions: Administrators can add elective subjects to the system, maintain academic records, and monitor students' academic performance.

4) Credit Score Monitoring: The system serves as a reminder for students to regularly check their credit scores, helping them ensure they meet academic requirements and avoid the need for an extra semester.

1.4. Project Limitations:

1) **Limited User Acceptance**: Resistance to change among administrative staff or students might hinder the acceptance and adoption of the new system, impacting its effectiveness.

2) **Dependency on Technology**: Relying entirely on technology makes the system vulnerable to outages, power failures, or network issues, disrupting access to essential services.

3) **Maintenance Challenges**: Ensuring the system's long-term functionality requires regular updates, maintenance, and technical support, which can strain the institution's resources.

4) **Scalability Issues**: If the system is not designed to handle a large volume of users and data, it might face scalability challenges as the institution grows, leading to performance issues.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2
BACKGROUND WORK

2.1. Development of Online Student Course Registration System

2.1.1. Introduction

System may be defined as a layered structure that depicts how programs involved would interrelate and communicate. In computers, System may also include actual programs, programming interfaces and tools for managing the larger system. The term system may be used differently in different contexts, but more or less the concept remains the same. Online student course registration system combines multiple systems to construct a combined framework. This framework consists of multiple modules, which further contain different systems along with the implementation of their defined constraints.

Basically, systems are implemented for facilitating complex manual processes and that is exactly what we are trying to achieve. System is implemented as per user requirement such as a manufacturing concern may install a plant for easing out manual processes. We have sought help from computer programming for automation of manual registration system. With the introduction of computers, every aspect of our lives has been revolutionized. When used judiciously, computers can help us save time, secure our personal information, access the required information whenever and wherever required. Keeping all these positive points in mind, we have developed an Online Student Course Registration System for easily managing the semester registration process for the student in an institution.

2.1.2. Merits, Demerits and Challenges

Merits:

Efficiency and Time-Saving: The system significantly streamlines the course registration process, reducing the laborious and time-consuming manual paperwork. Students can register for courses more efficiently, and administrators can manage the process more effectively.

User-Friendly: The online system is designed to be user-friendly, making it accessible to both students and administrative staff. It simplifies the registration process, reducing the chances of errors and confusion.

Automation: Automation of the course registration process eliminates the need for physical paperwork, reducing the chances of lost documents and manual errors. This leads to a more accurate and reliable registration process.

Demerits:

1) Technical Challenges: Some students or staff members may not be technologically proficient, and they may struggle with using the online registration system. This can lead to resistance to change and require additional training.

2) Initial Development and Maintenance Costs: Developing and maintaining an online registration system, including hardware, software, and security measures, can be costly. It requires continuous updates and support.

3) Internet Dependency: The system relies on internet connectivity, which could be a limitation in areas with unreliable or limited access to the internet.

Challenges:

1) User Experience Design: Designing a user-friendly interface that caters to the needs and preferences of both students and administrative staff is critical but can be challenging to achieve effectively.

2) Scaling and Performance: Ensuring that the system can handle a large volume of concurrent users during peak registration periods without performance issues or downtime is a technical challenge.

3) Regulatory Compliance: Depending on the jurisdiction and type of institution, there may be legal and regulatory requirements related to data privacy and accessibility that need to be addressed.

2.1.3. Implementation

An online registration system was developed where a student can register himself. The registration form has been designed to be user friendly and easy to fill and hence leads to saving of time and money as compared to multiple forms filled manually by the students. The software development team ascertained the technical feasibility of the project and concluded that the project can be undertaken with available technology and resources. Although implementation phase might require hardware additions but currently the project is technically feasible and should proceed further. The operational feasibility analysis acknowledged the acceptability of the provided solution to the problem. This analysis verified that the new system will be acceptable and adaptable to the new users. The economic feasibility study perceived that the project will produce long term gains for the institution. The cost benefit analysis proved that benefits of the proposed system undermine the costs involved, hence the system is worth implementing. The utility it provides to the students for completing the registration process and the provision it provides to the faculty for managing the database makes this project feasible to undertake. While designing the web application portal a three tier architecture for application development was followed. The presentation tier occupies the front end design of the application. It relates to every entity with which the user interacts. It accepts user inputs and actions, and then sends this information to the data tier through the application tier for further processing. The student course registration portal accepts input in the form of student's mind the authenticity concerns of different users.

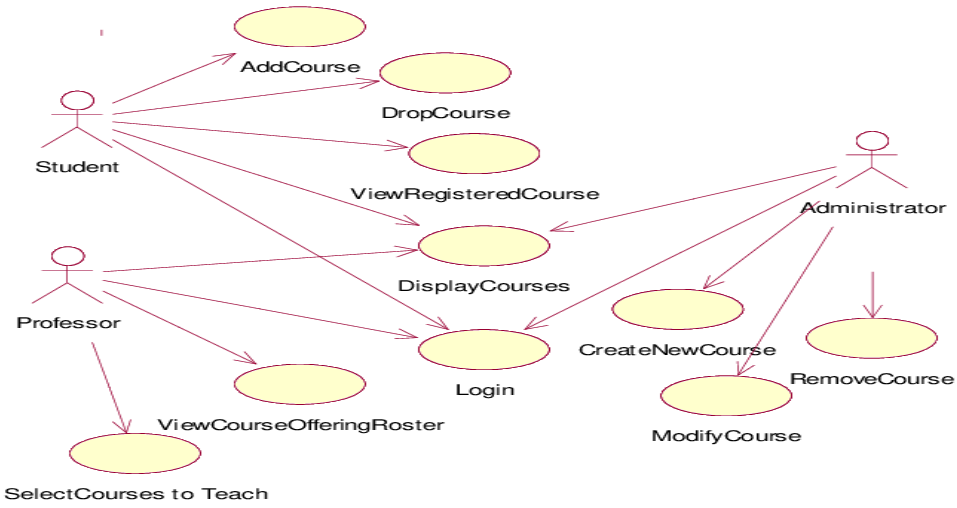


Fig 2.1 Use case Diagram

2.2. Design and Implementation of Student Registration System for Universities

2.2.1. Introduction

Every term the student registration was an indispensable part of work for the universities. Traditionally, the work was done by artificial management, and the registration of the corresponding classes was responsible by counselors from the institutes of colleges. In the process of registration, the counselor should stay in his office all day around, waiting for the students to come to registration. The whole process of registration was very simple. The counselor stamped on the corresponding location of the student's student ID card each term. This process was closed information, manual operation and low efficiency. To sum up, there might be some problems in the old registration process which is shown as follows.

2.2.2. Merits, Demerits and Challenges

Merits:

- Face-to-Face Interaction: The traditional registration process allowed for direct, in-person interaction between students and counselors.
- Immediate Problem Resolution: Any issues or questions that students had during the registration process could be addressed immediately by the counselor, ensuring a quick resolution.
- Personalized Guidance: Counselors had the opportunity to provide personalized guidance to students, helping them make informed decisions about their course selections.

Demerits:

- Lack of Data Analysis: Traditional registration processes provided little opportunity for data analysis, making it difficult for institutions to gather insights on course demand, student preferences, and overall registration trend.
-) Prone to Errors: Manual stamping of ID cards could lead to errors, both in course selection and in the recording of registered courses. This could result in academic and administrative challenges.
- LaMDA requires a large amount of annotated data and computational resources for pre-training and fine-tuning, which may be costly and unsustainable.

Challenges:

- Limited Time Window: Students had to adhere to specific registration days and times, which could create scheduling conflicts for some individuals, making it difficult for them to register for their desired courses
- Overcrowding: The traditional process, especially in larger universities, often led to overcrowding and long lines, causing frustration and stress for both students and counselors.
- Data Accuracy: The reliance on manual processes, such as stamping ID cards, made the system prone to human errors, which could lead to issues with course registration and academic progress tracking

2.2.3. Implementation

Function modules of the student registration system for universities can be divided into four roles, the administrator, the student, Finance Department and Student Affairs Office. The administrator is responsible for the maintenance of the entire system data, as well as the statistical and the analysis of data. The student uses the system to register just by brushing the student card. Finance Department is responsible for updating the information of the students' payments and the information of the students' loans. Student Affairs Office is responsible for importing the basic information data of freshman every school year.

One key area of focus in the background work is the utilization of modern web development technologies and frameworks. Technologies such as React.js, Node.js, Express, and PostgreSQL have gained prominence for their ability to build scalable, responsive, and robust web applications. By leveraging these technologies, educational institutions can develop online registration systems that offer enhanced functionality, performance, and user experience.

Furthermore, the integration of algorithms for seat allocation represents a significant aspect of the background work. Drawing inspiration from industry-leading platforms like BookMyShow, researchers have explored various algorithms and strategies for optimizing seat allocation in the context of online ticketing and event booking systems. These algorithms aim to maximize seat utilization, accommodate user preferences, and minimize seat allocation conflicts.

2.3. Course Registration System

2.3.1. Introduction

A college management registration system empowers colleges and educators to manage regular tasks such as campaigns, student enrolment, admissions, course registration, etc. Registering for courses while taking admission to a college is the most crucial step as it lays the foundation for an entire semester. Providing this information to thousands of students, and that too correctly is a huge task if done manually. Hence, an online College Management System is needed which can provide a list of courses available according to the semester a student selects and talks about the courses one has selected. This system can be used as an integrative tool for many colleges and can be modified according to a college's requirements. This system is much quicker, efficient and the only force used is the clicking of a mouse button.

The process of course registration in educational institutions has traditionally been a complex and time-consuming task, involving manual paperwork, administrative overhead, and potential errors. As educational institutions continue to evolve and adapt to the digital age, there has been a growing emphasis on leveraging technology to streamline administrative processes and enhance user experience.

Several studies and initiatives have explored the challenges and opportunities associated with modernizing the course registration process. These efforts have highlighted the importance of developing user-friendly, efficient, and secure systems to facilitate online course registration. Additionally, research has underscored the need to address issues such as data integrity, security, and user proficiency to ensure the successful implementation and adoption of online registration systems.

2.3.2. Merits, Demerits and Challenges

Merits:

1) Efficiency and Time-Saving: The system significantly streamlines the course registration process, reducing the laborious and time-consuming manual paperwork. Students can register for courses more efficiently, and administrators can manage the process more effectively.

2) User-Friendly: The online system is designed to be user-friendly, making it accessible to both students and administrative staff. It simplifies the registration process, reducing the chances of errors and confusion.

3) Automation: Automation of the course registration process eliminates the need for physical paperwork, reducing the chances of lost documents and manual errors. This leads to a more accurate and reliable registration process.

Demerits:

1) Technical Challenges: Some students or staff members may not be technologically proficient, and they may struggle with using the online registration system. This can lead to resistance to change and require additional training.

2) Initial Development and Maintenance Costs: Developing and maintaining an online registration system, including hardware, software, and security measures, can be costly. It requires continuous updates and support.

3) Internet Dependency: The system relies on internet connectivity, which could be a limitation in areas with unreliable or limited access to the internet

2.3.3. Implementation

The "Course Registration System" is implemented with an HTML and CSS front-end to create a user-friendly interface. This system allows students to search for courses by selecting a semester and displays course information. The "Course Registration" step lets students register for their chosen course by entering the Course ID and Semester. On the back-end, PHP scripts connect with a MySQL database to handle these operations, with thorough exception handling for potential errors, ensuring data integrity and user security. The system is tested rigorously for functionality and security, deployed on a web server, and continuously monitored and updated for optimal performance.

Additionally, studies have examined the challenges and opportunities associated with transitioning from manual to online course registration systems. Factors such as technical infrastructure, institutional culture, stakeholder engagement, and user training play critical roles in the successful implementation and adoption of online registration systems.

Overall, the background work encompasses a comprehensive review of existing literature, research, and technological advancements related to online course registration systems in educational institutions. By building upon this foundation, the proposed thesis aims to develop and evaluate a novel online registration model that addresses the unique needs and challenges of educational institutions in the digital age.

CHAPTER 3
PROPOSED SYSTEM

3.1. Objective of Proposed Model

The objective of the proposed model is to develop an efficient and user-friendly online course registration system for educational institutions, leveraging modern technologies and algorithms. Specifically, the goals of the proposed model include:

Streamlining the Course Registration Process: To design a system that simplifies the course registration process for students, reducing administrative burdens and enhancing overall efficiency. Enhancing User Experience: To create an intuitive and user-friendly interface that improves the experience of students, faculty, and administrative staff during the course registration process. Ensuring Data Integrity and Security: To implement robust data management practices and security measures to safeguard sensitive student information and prevent unauthorized access or data breaches. Leveraging Modern Technologies: To utilize technologies such as React.js, Node.js, Express, and PostgreSQL to develop a scalable and reliable system that meets the evolving needs of educational institutions. Implementing Seat Allocation Algorithm: To incorporate a seat allocation algorithm inspired by industry standards like BookMyShow, optimizing seat allocation to maximize utilization of resources while ensuring customer satisfaction. Facilitating Real-Time Communication: To enable real-time communication and notifications between students, faculty advisors, registrars, and other stakeholders involved in the course registration process. Addressing Technical Proficiency Gaps: To provide training and support to users to address any technical proficiency gaps and ensure smooth adoption and utilization of the proposed system.

3.2. Technologies Used In Proposed Model

REACT JS:

React.js, commonly referred to as React, is an open-source JavaScript library developed by Facebook for building user interfaces (UIs) and single-page applications. It is often used for creating dynamic and interactive web applications with high performance.

Component-Based Architecture: React.js is built around the concept of reusable components. Components are self-contained units of code that encapsulate a specific piece of UI. These components can be composed together to build complex UIs. This approach promotes reusability, maintainability, and scalability in web development.

Virtual DOM (Document Object Model): One of the key features of React is its virtual DOM. Instead of directly manipulating the browser's DOM, React creates an in-memory representation of the DOM known as the virtual DOM. When changes are made to the UI, React first updates the virtual DOM and then efficiently compares it with the real DOM to identify the minimal set of changes needed to be applied. This minimizes DOM manipulation overhead and improves performance.

Declarative Syntax: React uses a declarative approach to describe how the UI should look at any given point in time. Developers specify the desired UI state, and React takes care of updating the DOM to match that state. This is in contrast to imperative programming, where developers have to manually manipulate the DOM to achieve the desired UI changes.

JSX (JavaScript XML): JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript. JSX makes it easier to describe UI components and their structure directly within the JavaScript code. It provides a more concise and expressive way of defining UI elements, which helps improve code readability and maintainability.

Unidirectional Data Flow: In React, data flows in one direction through the component hierarchy. This means that data changes at higher levels of the hierarchy propagate down to child components via props (properties). This unidirectional data flow simplifies the understanding of how data changes affect the UI and makes it easier to debug and maintain large-scale applications.

React Hooks: Introduced in React 16.8, hooks are functions that allow developers to use state and other React features in functional components. Hooks enable functional components to have state and lifecycle methods previously only available in class component.

Web-based Integration, we have used MERN stack which involves MongoDB, Express.js, React.js, Node.js. Reasons for using MERN Stack for this project are as follows:

MERN allows developers to use JavaScript for both frontend and backend development. This eliminates the need to switch between different programming languages, resulting in a more streamlined development process and improved code consistency. With MERN, developers can build full-stack web

applications using a single language (JavaScript) and a unified set of technologies. This reduces the complexity of the development environment and facilitates seamless integration between frontend and backend components.

React.js, the frontend library in the MERN stack, follows a component-based architecture. This allows developers to create modular and reusable UI components, leading to faster development, easier maintenance, and consistent user interfaces across the application.

Node.js, the backend runtime in the MERN stack, is known for its non-blocking, event-driven architecture, which enables high scalability and performance. Combined with MongoDB's flexible data model and Express.js's lightweight framework, MERN applications can handle large volumes of concurrent requests efficiently.The MERN stack benefits from a rich ecosystem of libraries, frameworks, and tools, as well as active community support. Developers have access to a wide range of third-party packages and resources, which accelerates development and provides solutions to common challenges.

MERN's simplicity, flexibility, and modular architecture make it ideal for rapid prototyping and iterative development. Developers can quickly build and test prototypes, make iterative improvements based on feedback, and deploy updates seamlessly, leading to faster time-to-market and improved user satisfaction. The MERN stack enjoys widespread adoption in the developer community and industry. Many companies and organizations use MERN for building web applications due to its proven reliability, scalability, and performance, making it a solid choice for projects of all sizes and complexities.

To explain more about each of the Technologies:

PostgreSQL, often abbreviated as Postgres, is a powerful open-source relational database management system (RDBMS) known for its reliability, robustness, and extensibility. Developed by a diverse community of contributors, PostgreSQL has gained widespread adoption across various industries due to its advanced features and adherence to SQL standards.

Here are some key aspects of PostgreSQL:

Open Source: PostgreSQL is released under the PostgreSQL License, a permissive free and open-source license. This means that anyone can use, modify, and distribute PostgreSQL without any licensing fees.Relational Database Management System (RDBMS): PostgreSQL follows the relational model, storing data in tables composed of rows and columns. It supports complex SQL queries, transactions, and ACID (Atomicity, Consistency, Isolation, Durability) properties, making it suitable for handling mission-critical data.

Extensibility: PostgreSQL provides a rich ecosystem for extending its functionality. Users can develop custom data types, functions, and procedural languages using PL/pgSQL, PL/Python, PL/Perl, PL/Java, and other languages. Additionally, PostgreSQL supports the integration of external tools and libraries through various extension mechanisms.

Advanced Features: PostgreSQL offers a wide range of advanced features, including:

JSON and JSONB data types for storing and querying JSON documents.

Full-text search capabilities through the built-in tsvector and tsquery types.

Geospatial data support with PostGIS extension for handling geographic information.

Advanced indexing options such as B-tree, hash, GiST, GIN, and BRIN indexes to optimize query performance.

Multi-Version Concurrency Control (MVCC) for handling concurrent transactions efficiently.

Table partitioning for improved manageability and performance of large tables.

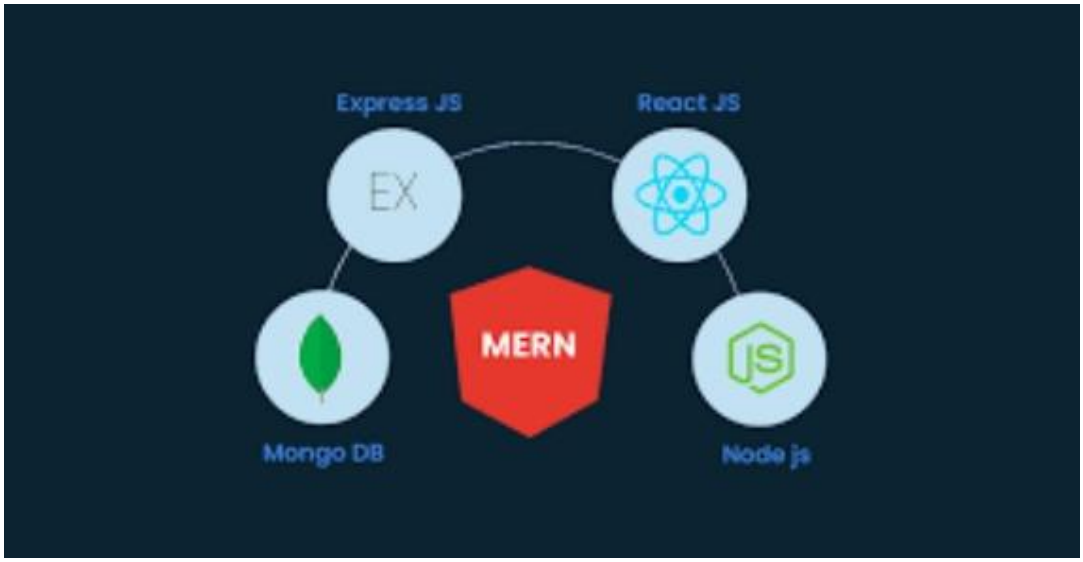
Express.js

Express.js is a web application framework for Node.js. It provides a minimalist and flexible set of features for building web applications and APIs. Express.js simplifies the process of handling HTTP requests, routing, middleware integration, and more. In the MERN stack, Express.js is used on the server-side to handle incoming requests from clients, process data, interact with the database (MongoDB), and send responses back to the client.

Node.js

Node.js is a server-side JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows developers to run JavaScript code outside of a web browser, making it suitable for building scalable and high-performance server-side applications. In the MERN stack, Node.js serves as the backend runtime environment where server-side logic is executed. It enables developers to build fast and efficient web servers, handle concurrent connections, perform I/O operations, and integrate with other technologies in the stack, such as Express.js and MongoDB. .

Fig. 3.1 MERN Stack



3.3 DESIGNING

3.3.1. UML DIAGRAM

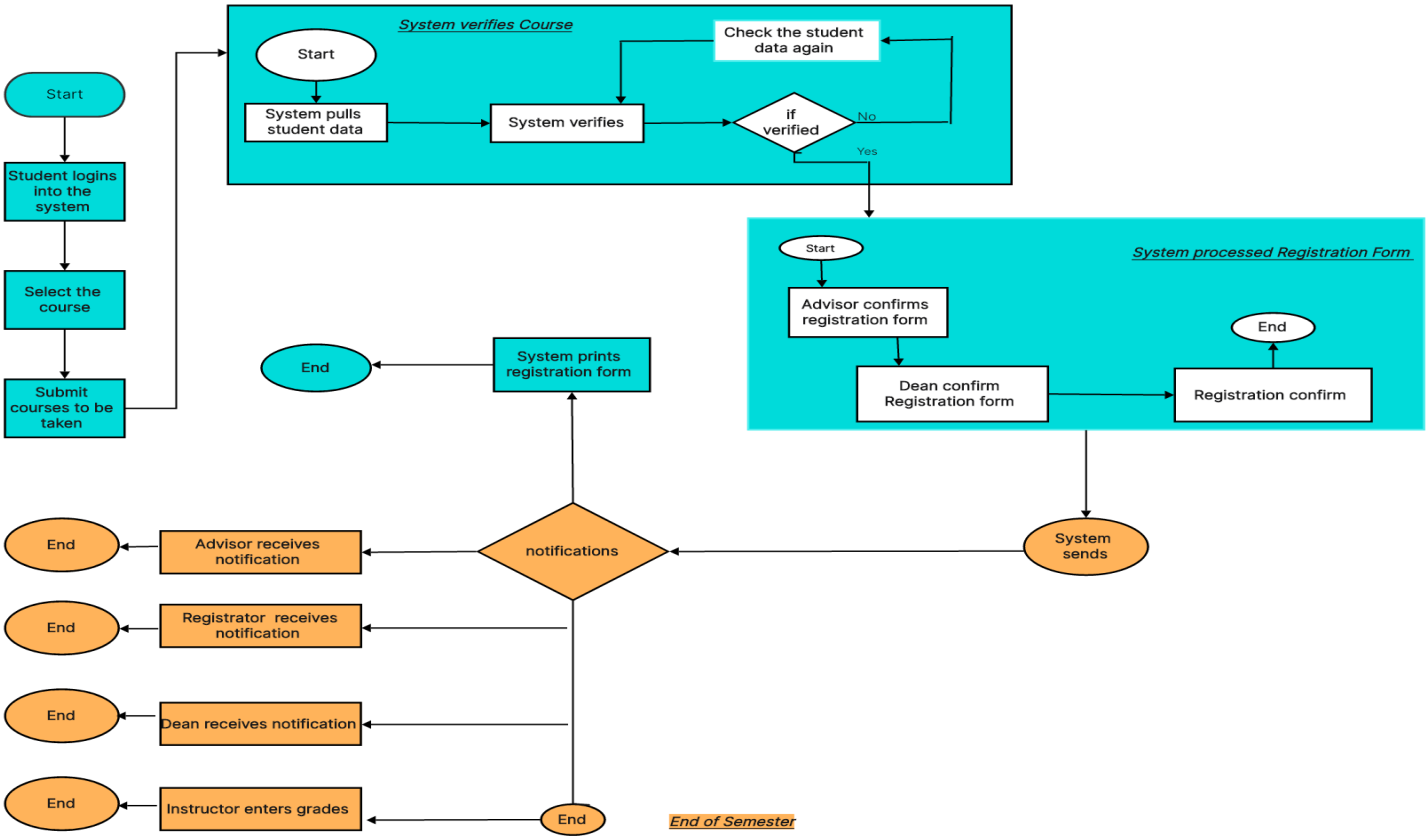


Fig. 3.2 UML Sequence diagram

Flow Description:

Student Flow:

The student initiates the process by logging into the system and selects a desired course.

The system proceeds to verify the student's eligibility for the chosen course by retrieving and analyzing their relevant data.

If the eligibility verification fails, the process loops back to re-verify the student's data.Upon successful verification, the system generates and prints a registration form for the student.

Error Handling:

In the event of unsuccessful verification, the process prompts a reevaluation of the student's data before proceeding further.

Key Elements:

Start/End: Denotes the initiation and termination points of the registration process.

Activities: Represent the specific actions undertaken throughout the process, including logging in, course selection, eligibility verification, and form printing.

Decisions: Depict conditional branches within the process, such as the eligibility verification step.

Notifications: Indicate the dispatch of notifications to designated officials for registration form confirmation.

Loop: Signifies the iterative nature of certain steps, such as the data verification loop.

Below are the some of the screenshots attached of our project

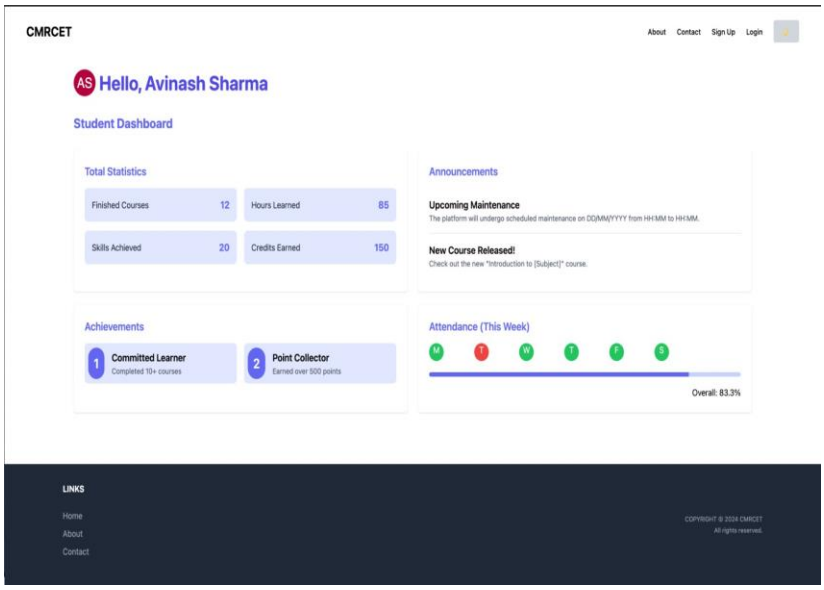


Fig 3.3 Student Dashboard

3.4. Stepwise Implementation and Code

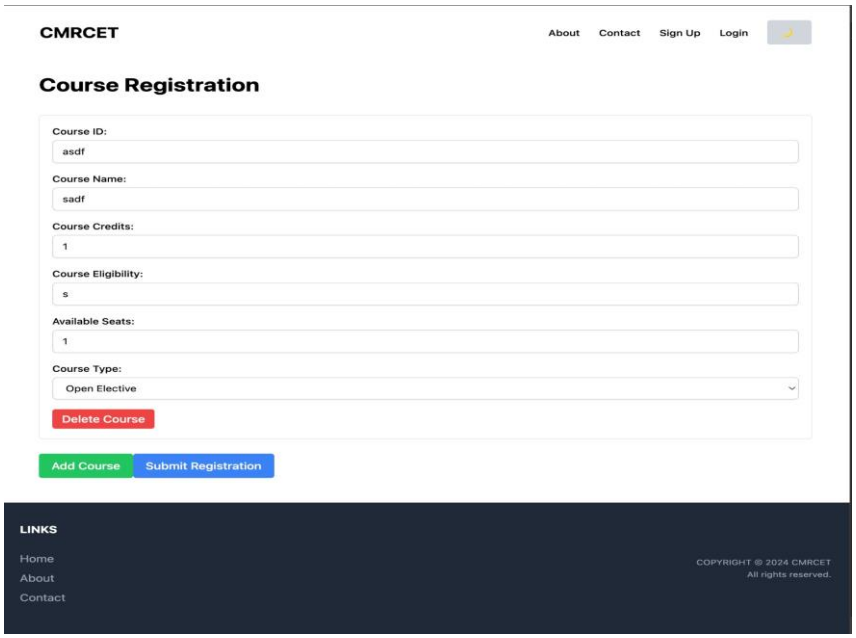


Fig 3.4 Course Registration form

Setup Instructions

1. Clone the Repository

If the project is hosted on a version control system like GitHub, start by cloning the repository. If not, skip to the next step.

git clone https://Course registration app

cd Course registration app

2. Database Setup

Create the Database

First, ensure your PostgreSQL server is running. Then create a new database:

```
CREATE TABLE users (
    user_id VARCHAR(12) PRIMARY KEY,
    email VARCHAR(255) UNIQUE NOT NULL,  -- Unique college email
    password VARCHAR(255) NOT NULL,  -- Store securely hashed password
    role VARCHAR(50) NOT NULL DEFAULT 'student', -- 'student', 'faculty', 'admin', etc.
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_active BOOLEAN DEFAULT TRUE  -- Account activation status
);
CREATE TABLE achievements (
    id SERIAL PRIMARY KEY,
    user_id VARCHAR(255) NOT NULL,
    achievement_summary TEXT NOT NULL,
    date_of_achievement DATE NOT NULL,
    certificate TEXT,
    photos TEXT[],
    is_approved BOOLEAN DEFAULT false
);
CREATE TABLE courses (
        course_id VARCHAR(255) PRIMARY KEY,
    sem VARCHAR(255) NOT NULL,
```

```
    course_name VARCHAR(255) NOT NULL,
    course_credit INT NOT NULL,
    course_eligibility INT NOT NULL,
    course_type VARCHAR(255) NOT NULL,
    faculty_id VARCHAR(255) NOT NULL,
    available_seats INT NOT NULL,
    course_syllabus TEXT NOT NULL
);
CREATE TABLE announcements (
    announcement_id SERIAL PRIMARY KEY,
    message VARCHAR(255) NOT NULL,
    expiry_date TIMESTAMP NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE student_courses (
    id SERIAL PRIMARY KEY,
    user_id VARCHAR(255) NOT NULL,
    course_id VARCHAR(255) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);
CREATE TABLE attendance (
    id SERIAL PRIMARY KEY,
    user_id VARCHAR(255) NOT NULL,
    course_id VARCHAR(255) NOT NULL,
    attendance_date DATE NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (course_id) REFERENCES courses(course_id)
);
```

Initialize the Database

Navigate to the directory containing your database initialization scripts, and run them using psql:

psql -d cmrcet -f path/to/your/database/init/script.sql

3. Backend Setup

Install Dependencies

Navigate to the backend directory and install the necessary npm packages:

cd server

npm ci

Start the Backend Server

Start the server with the following command:

nodemon src/server.js

Your backend should now be running on http://localhost:3001/.

4. Frontend Setup

Install Dependencies

Navigate to the frontend directory and install the necessary npm packages:

cd ui

npm ci

Start the Frontend Application

Start the React application with the following command:

npm start

Your application should now be accessible at http://localhost:3000/.

Implementation of Code:

Login.js:

```
import React, { useState } from "react";
import { Link } from "react-router-dom";
import axios from "axios";
import { useNavigate } from "react-router-dom";

function Login() {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const navigate = useNavigate();
```

```
 const handleSubmit = async e => {
   e.preventDefault();
   try {
     const { data, status } = await axios.post("http://localhost:3001/auth/login", {
       user_id: username,
       password
     });

     if (status === 200) {
       localStorage.setItem("token", data.token);
       localStorage.setItem("data", JSON.stringify(data));
       navigate("/dashboard");
     } else {
       throw new Error("Invalid credentials");
     }
   } catch (error) {
     alert(error.message);
   }
 };
       <Link to="/signup" className="text-sm text-gray-700 hover:text-blue-700">
         Don't have an account? Sign up
       </Link>
     </div>
   </form>
 );
}
export default Login;
```

SignUp.js:

```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
function Signup() {
```

```
 const navigate = useNavigate();
 const [formData, setFormData] = useState({
  user_id: "",
  password: "",
  first_name: "",
  last_name: ""
 });
 const handleChange = event => {
  setFormData({
   ...formData,
   [event.target.name]: event.target.value
  });
 };
 const handleSubmit = event => {
  event.preventDefault();
  fetch("http://localhost:3001/auth/signup", {
   method: "POST",
   headers: { "Content-Type": "application/json" },
   body: JSON.stringify({ ...formData })
  })
   .then(res => res.json())
   .then(data => {
    alert(data.message)
    navigate("/login");
   })
   .catch(error => console.error("Login error:", error));
  console.log("Form data:", formData);
 };
```

CourseDropdown.js:

```
import React, { useState, useEffect } from "react";
function CourseDropdown({ course, onCourseChange }) {
 const [searchTerm, setSearchTerm] = useState("");
```

```
 const [filteredcourse, setFilteredcourse] = useState(course);


 useEffect(() => {
  const filtercourse = () => {
   if (searchTerm.trim() === "") {
    setFilteredcourse(course); // Reset to original list
    return;
   }
   const filtered = course.filter(cls =>
cls.course_name.toLowerCase().includes(searchTerm.toLowerCase()));
   setFilteredcourse(filtered);
  };
  filtercourse();
 }, [searchTerm, course]); // Trigger filtering on search term or course change
 return (
  <div>
   <label htmlFor="search" className="block text-sm font-medium text-gray-700">
    Search course:
   </label>
   <label htmlFor="course-select" className="block text-sm font-medium text-gray-700 mt-2">
    Select course:
   </label>
   <select id="course-select" onChange={e => onCourseChange(e.target.value)} className="mt-
1 block w-full py-2 px-3 border border-gray-300 bg-white rounded-md shadow-sm focus:outline-
none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm">
    {filteredcourse.map(cls => (
     <option key={cls.id} value={cls.id}>
      {cls.course_name} - {cls.course_id}
     </option>
    ))}
   </select>
  </div>
 );
```

```
}
export default CourseDropdown;
```

CourseForm.js:

```
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
function CourseForm({courses,userId}) {
  const [selectedCourses, setSelectedCourses] = useState({});
  // Categorize courses
  const categories = {
    mandatory: courses.filter(course => course.course_type === "mandatory"),
    openElective: courses.filter(course => course.course_type === "open elective"),
    professionalElective: courses.filter(course => course.course_type === "professional elective"),
  }:
  // Handle change in course selection
  const handleCourseSelection = (e, course) => {
    const { checked } = e.target;
    setSelectedCourses(prev => ({
      ...prev,
      [course.course_id]: checked ? course : undefined,
    }));
  };
  // Handle form submission
  const handleSubmit = async (e) => {
    e.preventDefault();
    const courseIds = Object.entries(selectedCourses)
                        .filter(([_, isSelected]) => isSelected)
                        .map(([courseId, _]) => courseId);

    if (courseIds.length === 0) {
      console.error("No courses selected");
      return;
    }
```

```
  console.log(courseIds);
  async function getAvailableSeats(courseId) {
    try {
      const response = await fetch(`http://localhost:3001/courses/seats/${courseId}`);
      if (!response.ok) {
        throw new Error(`Error: ${response.statusText}`);
      }
      const {available_seats} = await response.json();
      return available_seats;
    } catch (error) {
      console.error('Failed to fetch available seats:', error);
    }
  }
  async function updateSeats(courseId, currentSeats) {
    try {
      const newSeats = currentSeats - 1; // Decrement the current seats by 1
      const response = await fetch(`http://localhost:3001/courses/updateSeats/${courseId}`, {
        method: 'PUT',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ available_seats: newSeats }),
      });
      if (!response.ok) {
        throw new Error(`Error: ${response.statusText}`);
      }
      console.log(`Seats updated successfully for course ID ${courseId}. New available seats:
${newSeats}`);
    } catch (error) {
      console.error('Failed to update seats:', error);
    }}
  async function enrollInCoursesSequential(courseIds) {
    for (const courseId of courseIds) {
```

```
        try {
          const availableSeats = await getAvailableSeats(courseId);
          if (availableSeats != null && availableSeats > 0) {
            await updateSeats(courseId, availableSeats);
            console.log(`Successfully enrolled in course ${courseId}.`);
          } else {
            console.log(`No seats available for course ${courseId}.`);}
        } catch (error) {
          console.error(`Failed to enroll in course ${courseId}:`, error);
        } }  }
    try {
      const response = await fetch('http://localhost:3001/courses/enroll', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          student_id: userId,
          course_ids: courseIds,
        }),
      });
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      const data = await response.json();
      enrollInCoursesSequential(courseIds);
      alert("Enrollment successful");
      // Handle success
    } catch (error) {
        alert("Enrollment failed");
      // Handle errors
    }
  };
```

```
$1').trim()}</h2>
        {courses.map(course => (
          <div key={course.course_id} className="flex items-center mt-2">
            <label className="flex items-center space-x-2">
              <input
                type="checkbox"
                checked={!!selectedCourses[course.course_id]}
                onChange={(e) => handleCourseSelection(e, course)}
                className="form-checkbox h-5 w-5 text-blue-600/>
              <a
              href={`/course/${course.course_id}`}
              target="_blank"
              rel="noopener noreferrer"
              className="text-blue>
              {course.course_name}
              </a>
              <span> {"->"} (Credits: {course.course_credit}) (Available Seats:
{course.available_seats})</span>
            </label>
          </div>
        ))}
      </div>
    ))}
    <button type="submit" className="px-4 py-2 bg-blue-500 text-white rounded-lg shadow
hover:bg-blue-700">Submit</button>
  </form>
 );
}
export default CourseForm;
```

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Comparison of Existing Solutions

Comparison Criteria	**Offline form fill up**	**Student 360**
Online Student Registration	✗	✓
Admin Profile	✗	✓
Online records of the course	✗	✓
Attendance records	✓	✓
Announcement Notifications	✗	✓
Overall Suitability for Engineering Student Needs	✗	✓
Enhancement of User Experience	✗	✓

Table No. 4.1 Comparision Table

The implementation of the Online Student Course Registration System has revolutionized the course registration process, significantly reducing administrative burdens, minimizing errors, and boosting overall efficiency. Noteworthy advantages include its user-friendly interface and seamless automation. However, challenges persist, particularly regarding user proficiency and reliance on internet connectivity.

In contrast, the Student Registration System for universities maintains the invaluable aspect of personalized, face-to-face interactions between counselors and students. This facilitates immediate problem-solving and guidance, adding a distinct touch of personalization to the registration process. Nonetheless, limitations arise in terms of data analysis capabilities and the propensity for manual errors.

Similarly, the College Course Registration System has ushered in efficiency and accessibility, empowering students to register for courses online. This transition simplifies procedures, reduces reliance on physical visits, and elevates the overall user experience. Nevertheless, hurdles such as technical expertise requirements, initial development costs, and data security concerns must be addressed.

Despite these challenges, the adoption of online registration systems signifies a significant stride forward in enhancing efficiency and user satisfaction within educational institutions. However, ongoing efforts are necessary to address technical proficiency gaps and financial obstacles, ensuring a smooth transition and sustained benefits for all stakeholders.

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION

In conclusion, the challenges faced by colleges and universities in gathering elective information from students have been effectively addressed through our innovative solution. By implementing a reminder system that prompts students to review their academic progress and complete any outstanding subjects, we have streamlined the elective selection process. This not only reduces resource wastage but also enhances academic efficiency by ensuring students enroll in relevant courses.

Moreover, the introduction of an online student course registration system has revolutionized the registration process, making it simpler and more accessible. The transition from manual procedures to a computerized system has significantly reduced the time required for process completion and relieved department officials from tedious manual data entry tasks. Utilizing MongoDB for database management has ensured data integrity and minimized the likelihood of errors, while an authentication-based access system has enhanced security measures.

Overall, our project has not only improved the efficiency of elective selection and course registration but also contributed to the overall enhancement of organizational processes within educational institutions. The automated and computerized system we have developed excels in terms of safety, speed, and user-friendliness, marking a significant step forward in optimizing academic administrative processes for the benefit of both students and staff alike.

Furthermore, the implementation of this project underscores our commitment to leveraging technology to address longstanding challenges in higher education administration. By harnessing the power of digital tools, we have created a system that not only simplifies processes but also fosters transparency and accountability. The centralized storage of data, distributed access among departments, and stringent authorization protocols ensure that information is handled securely and efficiently. As we look to the future, the success of this project serves as a testament to the transformative potential of technology in optimizing educational practices. By continuing to innovate and adapt, we can further enhance the academic experience for students and streamline administrative workflows for institutions, ultimately advancing the goals of education in the digital age.

REFERENCES

REFERENCES

[1]. Shafie, E., Al-Ajlan, A., Aldrawiesh, K., Bajahzar, A., & Al-Saawy, Y. (2011, January). "Online registration system". In Advanced Computer Control (ICACC), 2011 3rd International Conference on (pp. 174-179).

[2]. E. Hart-Baldridge, "Faculty Advisor Perspectives of Academic Advising," NACADA J., vol. 40, no. 1, pp. 10–22, 2020.

[3]. M. N. Esraa Afify, "A Proposed Model for a Web-Based Academic Advising System," (International J. Adv. Netw. Appl. (IJANA, vol. 9, no. 2, pp. 3345–3361, 2017.

[4]. L. Keston and W. Goodridge, "AdviseMe: An Intelligent Web-Based Application for Academic Advising," Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 8, 2015.

[5]. Huang Jiancheng Li Weiwen He Qubo Zhang Wanmao, "Design and implementation of the management system of college students' registration and tuition payment," no. 5, vol. 25, China Medical Education Technology, Xi'an, 2011, pp. 535–537

[6]. https://react.dev/learn/adding-interactivity

[7]. https://www.mongodb.com/docs/manual/crud/

[8]. https://nodejs.org/docs/latest/api/module.html

[9]. Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du YaGuang Li, Hongrae Lee Huaixiu Steven Zheng Amin Ghafouri Marcelo Matthew Lamm Viktoriya Kuzmina Joe Fenton Aaron Cohen Rachel Bernstein Ray Kurzweil Blaise Aguera-Arcas Claire Cui Marian Croak Ed Chi Quoc Le, Google, "LaMDA: Language Models for Dialog Applications", arXiv:2201.08239v3 [cs.CL] 10 Feb 2022

[10]. Kaur, Harpreet, & Grover, Dinesh. "Design and Development of Online Hospital Management Information System" International Journal of Computer Science Engineering and Information Technology Research 3.2 79-88 (2013)

[11]. https://www.npmjs.com/package/bcrypt

[12]. https://jwt.io/introduction

[13]. https://www.mongodb.com/json-and-bson

[14]. https://inzint.com/wp-content/uploads/2023/02/Features-of-Mern-stack-development-services-You-Should-Know-768x397.png

GITHUB LINK

https://github.com/pamminasatwik/Major-project

iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59345

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Student 360: Navigating Courses at your Fingertips with Student 360

M. Shiva Kumar[1], Rishab Agarwal[2], Akshith Akkali[3], P. Satwik[4], Dr. S. Kirubakaran[5]
[1]*Assistant Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad,*
[2, 3, 4]*UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*
[5]*Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*

Abstract: ***In colleges and Universities, faculty face a lot of problems collecting information from students about the completion and choosing of Open Electives and Professional electives at the beginning of the semester. Also, most of the students select the subjects which were already attained by them, not realizing until the final semester. Thus, resulting in not satisfying the required credit score and taking an extra semester wasting their valuable time. Student360 is a project that focuses on students completing their Open Electives and Professional electives, faculty managing students academic profiles and admins adding subjects elected by students and also regarding their academics. The project works as a reminder for students to check their credit score and wrap up the subjects which were not completed by them previously.***
Keywords: Engineering Education Tools, Web-based Assistance, Efficiency Enhancement, Technical Information Retrieval, Enhanced Learning, Problem-Solving Acceleration, Innovative Learning Solutions.

I. INTRODUCTION

The current manual course registration system within the educational institution is beset with several challenges, hindering the seamless enrollment process for students. These challenges include Complex and Time-Consuming Process, The manual course registration process is intricate and time-consuming, involving multiple steps such as form filling, approvals, and data entry. This complexity often leads to errors and delays in the registration process. Limited Accessibility, Physical registration forms and records are difficult to access for both students and administrative staff. Limited Scalability, With a manual system, accommodating a growing number of students becomes increasingly challenging. The manual process cannot efficiently handle a large volume of course registrations, leading to bottlenecks during peak registration periods. The aim of implementing the College Management System is to use technology to make human work easier. Using this system as a tool, students can search and register for courses of their choice. All colleges can integrate this tool in their existing websites and applications and can make their existing systems faster and more efficient. It will also come in very handy to students and will be easy to use.

II. RELATED WORK

User Requirements Analysis, Identify the specific needs and requirements of educators, administrators, and students concerning student profile management. Determine the pain points and challenges faced in the current manual or digital systems. User Interface and Experience, Design a user-friendly interface that caters to the diverse user roles within educational institutions. Conduct usability testing and gather feedback to optimize the user experience. Performance and Reliability, Test the application's performance under various loads to ensure reliability during peak usage times. Develop strategies for data backup and disaster recovery.

A. An Online Registration System

This system was created so that students may sign up independently. Compared to several forms that students would have to manually fill out, the registration form is designed to be user-friendly and simple to fill out, saving time and money. After determining the project's technical viability, the software development team came to the conclusion that it could be completed given the resources and technology currently in use. While hardware additions may be needed during the implementation phase, the project is currently technically doable and should move forward. The offered remedy to the issue was deemed acceptable by the operational feasibility analysis. A three-tier application development architecture was used in the web application portal's design. The application's front end design is occupied by the presentation tier. It has to do with every object that the user communicates with. After receiving inputs and actions from the user, it routes this data through the application tier to the data tier for additional processing.

B. Assess Data Accuracy and Integrity

Identify the specific needs and preferences of both students and administrative staff regarding the course registration process. Investigate the accuracy and integrity of student and course data processed through the automated system. Analyze Resource. Evaluate the utilization of human resources and physical space within the institution after implementing the automated system.

C. Examine Cost-Effectiveness

Conduct a cost-benefit analysis comparing the expenses associated with the manual system and the automated Course Registration System.

III. PROPOSED METHODS AND ITS ARCHITECTURE

A. High-level Methodology

Student360 is a project that focuses on students completing their Open Electives and Professional electives, faculty managing students academic profiles and admins adding subjects elected by students and also regards about their academics. The login functionality allows users to access the system with specific roles, such as "Admin" or "Student." Upon successful login, the system checks the role and directs the user accordingly. If the user is an admin, they will be directed to the admin dashboard. If the user is a student, they will either see their dashboard or be prompted to register if they haven't done so already. For the student dashboard, the UI should provide an intuitive and user-friendly interface, displaying essential information. The main pages include "Courses Finished," where students can view their completed courses, and "Register Course," which offers a template for all semester courses. Electives are limited, and students can select them on a first-come-first-serve basis. An "Achievements Form" allows students to submit their accomplishments. The "Profile" page captures details like the academic year, password, name, date of birth, father's name, and more. After filling out the registration form, students can submit it, and the form approval process is directed to the admin. Appropriate filters ensure efficient processing. Upon admin approval, students receive an approved copy, and they can download it. This ensures transparency and a record of approved registrations .An "Announcement" page is included for disseminating important information to students. We have used MERN Stack to implement, the functionality of each technology goes like, In this project, MongoDB would be used to store student information, course details, achievements, and other relevant data. Collections could include "Students," "Courses," "Achievements," and "Forms," each storing data specific to its purpose. Express.js would handle routing, middleware, and server-side logic. It facilitates communication between the frontend (React) and the backend (Node.js), handling requests, and responding with the appropriate data or views. React is used for the frontend to create a dynamic and responsive user interface. The student dashboard, various forms, and pages, such as "Courses Finished," "Register Course," "Achievements Form," "Profile," and "Announcement," would be implemented using React components. React allows for the creation of a Single Page Application (SPA), enhancing the user experience by reducing page reloads. Node.js is used to build the backend server. It handles incoming requests from the frontend, interacts with the MongoDB database to retrieve or store data, and communicates with the frontend via the Express.js framework. Node.js supports asynchronous operations, making it suitable for handling multiple requests concurrently.

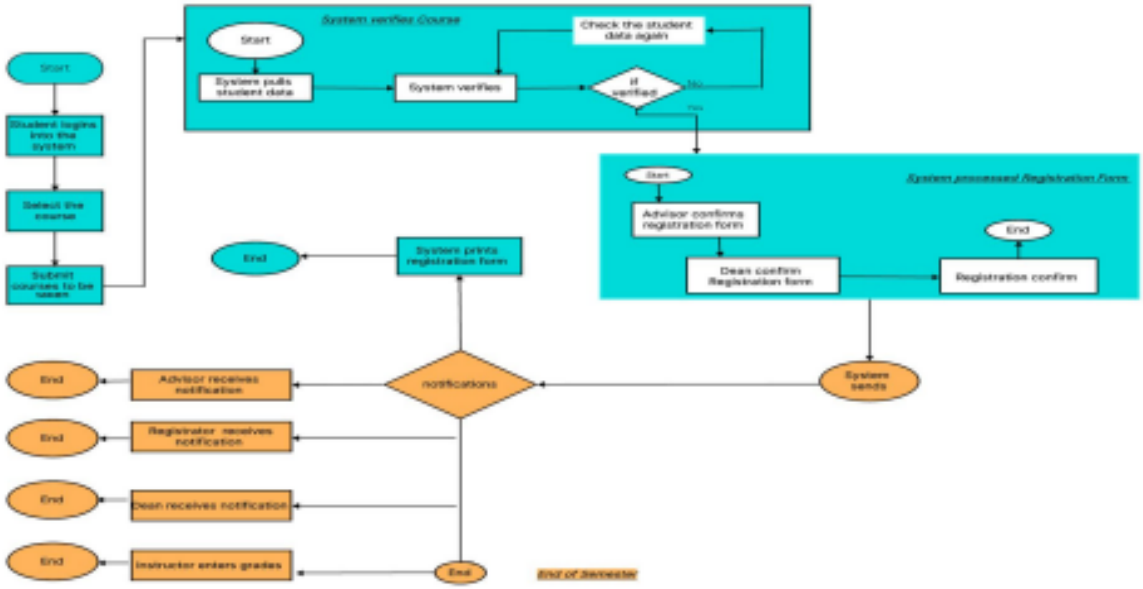


Fig1. Architecture of the Proposed Solution

Figure 1 - The architecture of the course registration system includes an easy-to-use interface that connects to a reliable back-end server. While student, course, and schedule data is managed by a central database, a secure authentication module guarantees safe access.

B. Users (Student, Admin)

Input credentials. The system validates the credentials. If valid, the system redirects the user to the respective dashboard.
Student is redirected to the Student Dashboard Page .Admin is redirected to the Admin Dashboard. If invalid, an error message is displayed, allowing re-entry of credentials.
Course Selection Page: Students view available courses categorized appropriately. Selection of desired courses by the student. Submission of the course selection. The system confirms the course registration. Admin Dashboard: Admin manages CRUD operations (Create, Read, Update, Delete) for courses, students, and faculty. Additions of new courses, updates to existing courses, and deletion of courses if necessary.

C. Faculty Flow: Faculty Dashboard

Faculty members view available courses they can teach. Selection of a course they want to teach. Confirmation of course allocation. End Flow: Users log out or perform other necessary actions. The system session concludes. Open Elective Course Selection: Students view available open elective courses. Criteria for selecting an open elective course: Credit Requirement: Minimum 'x' credits obtained in the previous semester. Attendance Requirement: Attendance in the previous semester above 75 percent. Selection limited to the first 40-45 students meeting the criteria. If criteria are met and the limit not reached, the student selects the desired open elective course. If criteria are not met or the limit is reached, the system provides a notification, prompting the student to select another course.

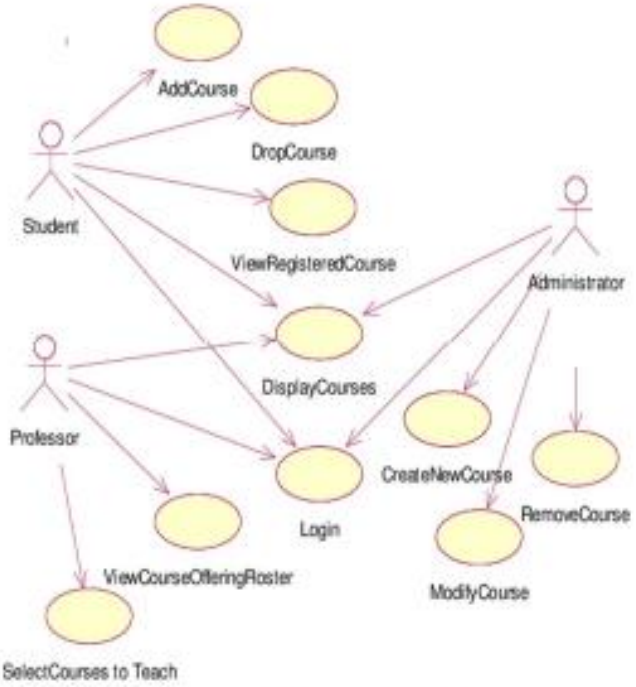


Fig 2. Use Case Diagram

IV. RESULTS AND DISCUSSIONS

The Online Student Course Registration System has streamlined and automated the course registration process, reducing administrative burdens, minimizing errors, and enhancing efficiency. The merits include enhanced user-friendliness and automation. However, there are challenges related to user proficiency and internet dependency.
The Student Registration System for universities maintains the personalized, face-to-face interaction between counselors and students, facilitating immediate problem resolution and guidance. This offers a unique merit in terms of a personal touch in the registration process. However, it has limitations in terms of data analysis capabilities and error-prone manual processes.

The College Course Registration System has introduced efficiency and accessibility, allowing students to register for courses online. This implementation simplifies the process, reduces the need for physical visits, and enhances the overall user experience. Nevertheless, challenges include technical expertise requirements, initial development costs, and data security concerns.

Challenges remain, particularly in the realm of technical proficiency. Some students and staff may find it challenging to adapt to the new digital systems, and additional training may be required to ensure a smooth transition. Furthermore, the initial development and ongoing maintenance costs can be a financial hurdle for institutions. The implementation of these online registration systems marks a significant step forward in improving efficiency and user experience in educational institutions.

A. *Merits*

1) *Efficiency and Accuracy:* The online system streamlines and automates various tasks related to student registration, admission, and course selection, leading to increased efficiency and reduced errors.
2) *Improved Accessibility:* Students can access the system from anywhere with an internet connection, eliminating the need for physical visits to the college, making it more accessible to a broader range of students.

B. *Demerits*

1) *Technical Challenges:* Implementing and maintaining the online system may require technical expertise, and some students and staff may struggle with using it.
2) *Initial Development Costs:* Developing the system, including software, hardware, and security measures, can be costly. Securing funding for the initial development and ongoing support can be challenging.

C. *Challenges*

1) *Data Privacy and Security:* Safeguarding sensitive student information from data breaches and ensuring data privacy is crucial and can be challenging.
2) *Customization:* Adapting the system to the specific needs and requirements of each college may be complex, as it may require customization and integration with existing systems.

Colleges and educators can handle routine duties like advertising, student enrollment, admissions, course registration, etc. with the help of a college management registration system. The most important step in the admissions process is registering for classes since it sets the stage for the entire semester. It would be very difficult to accurately provide this information to thousands of pupils by hand. Therefore, an online college management system that lists courses based on the semester a student chooses and provides information about the courses they have chosen is required.

Many colleges can use this system as an integrative tool, and it can be tailored to meet their specific needs.

Below Figure 3-5, you'll find a series of images showcasing our innovative web project in action. These visuals provide a glimpse into the user interface, features, and functionality of our web application. Take a moment to explore and discover how our project can revolutionize your online experience. From streamlined interactions to intuitive design, witness firsthand the power and potential of our creation. Dive in and envision the possibilities with our web project.

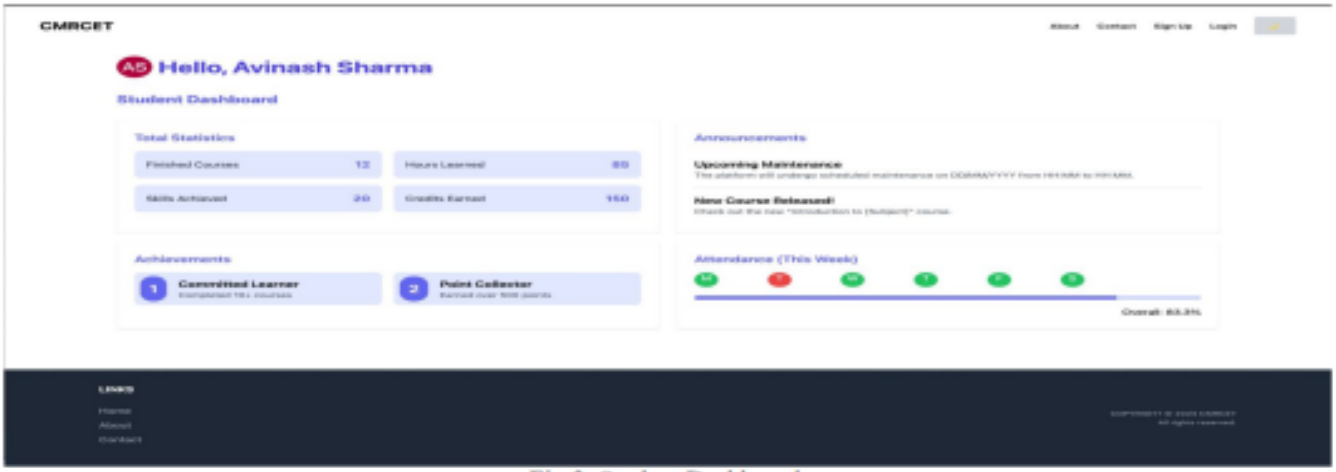


Fig 3. Student Dashboard

Fig4. Admin Dashboard

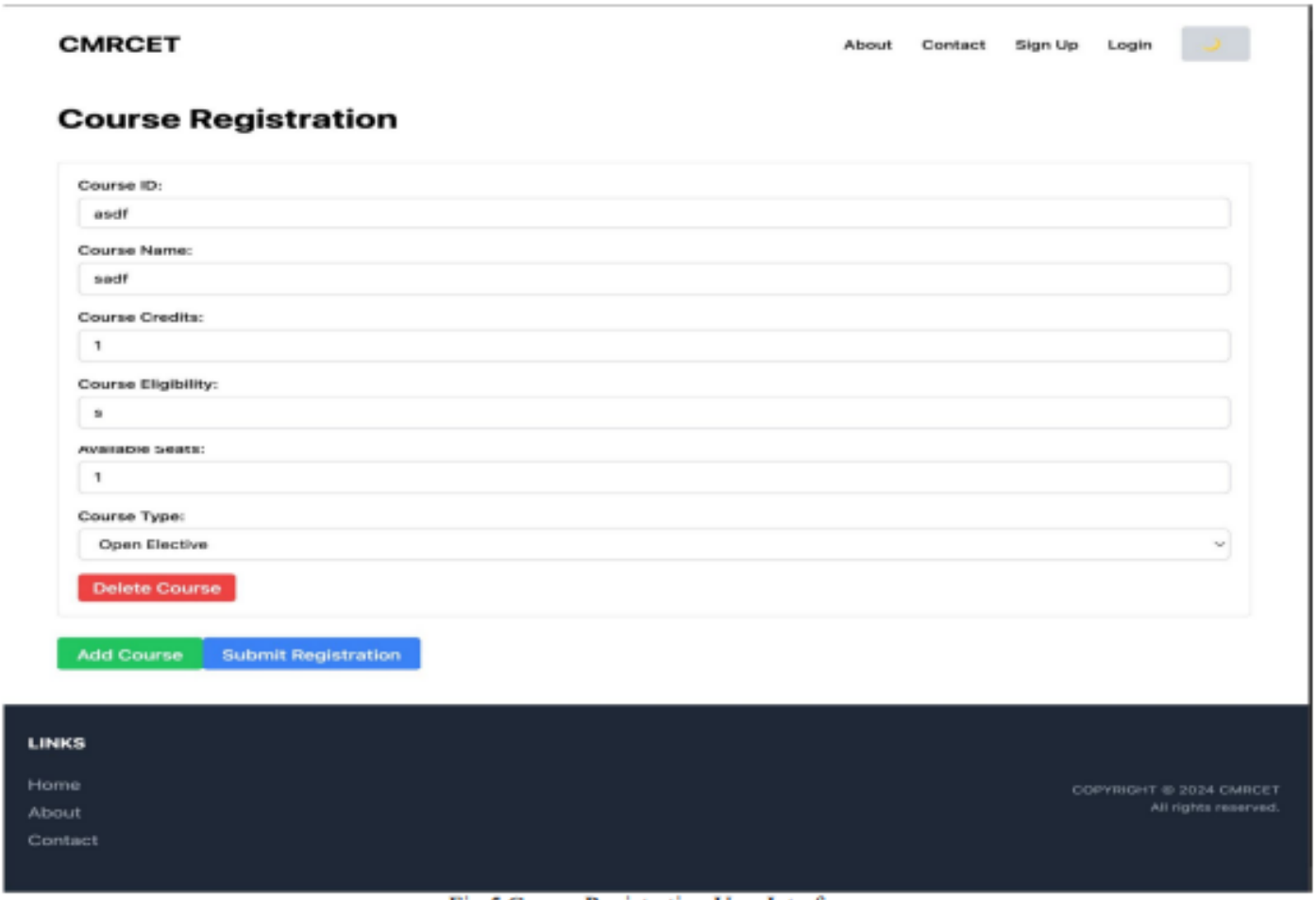


Fig 5.Course Registration User Interface

V. CONCLUSION

The Problem: Colleges and universities encounter challenges in gathering elective information from students. Students often unintentionally choose previously completed subjects.

Our Solution: The project works as a reminder for students to check their credit score and wrap up the subjects which were not completed by them previously.

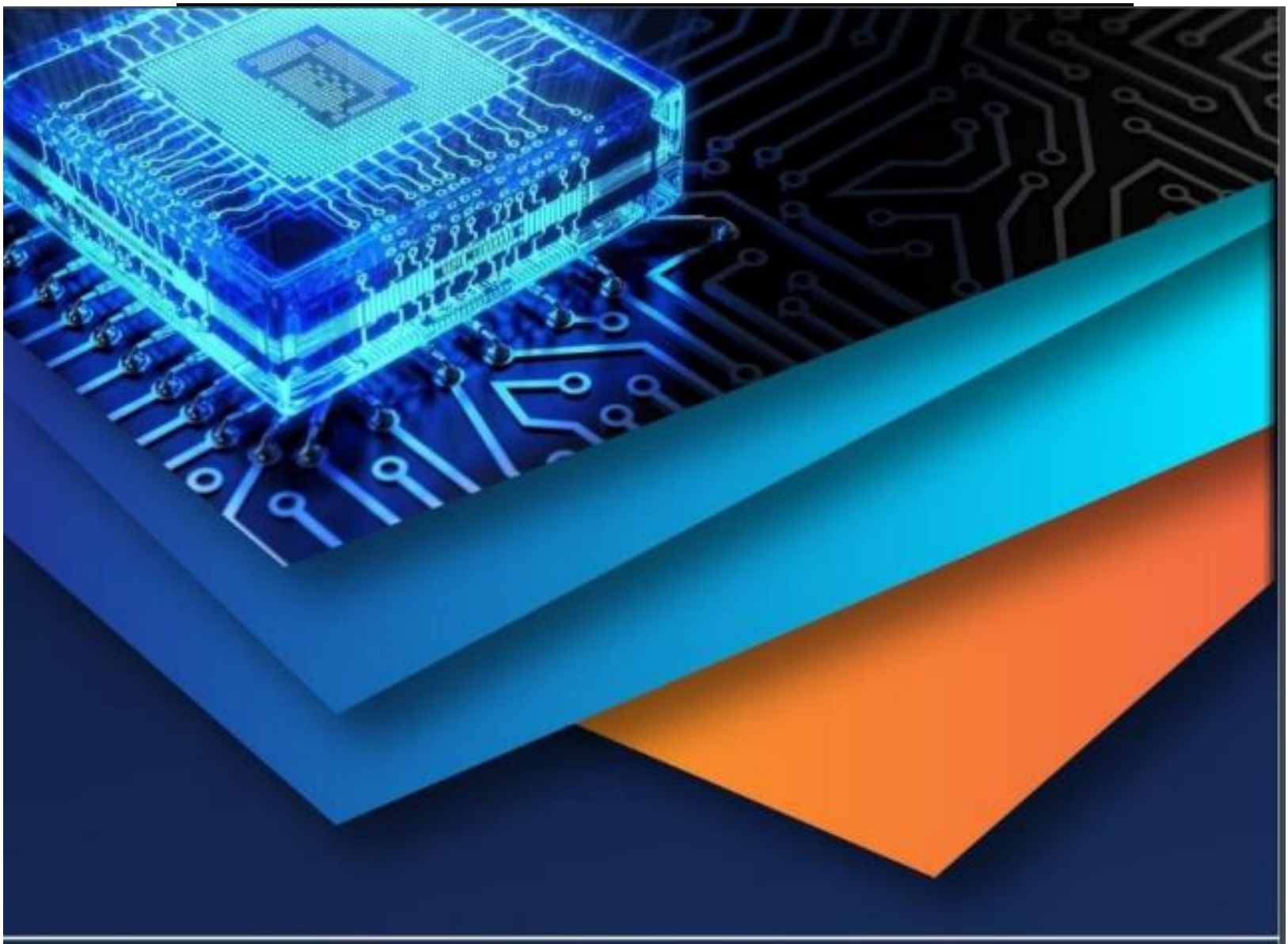
The Impact: Our solution streamlines elective selection, reducing resource wastage and enhancing academic efficiency.

The online student course registration system has made the registration process simpler. In the past, students had to knock on doors to get the relevant officials to accept their documents; however, the newly designed method provides a more effective means of carrying out these tasks. Students can use a computer or a smartphone to visit the registration portal online, fill out the required forms, and submit them for additional review. We are able to access this web application with ease and it is transparent. However, from the perspective of the organization, it supports easy maintenance, data accessibility, uniformity, and openness. When the suggested system is put into place, everyone's workload will be reduced because data can now be managed properly with authorization and authentication rather than being hard copied and available to everybody.

This has significantly reduced the time required for process completion. Upon registration, the database is automatically updated at the conclusion of the process, relieving department officials from the manual data entry burden. Utilizing MongoDB for database management ensures the elimination of data duplication, thus minimizing the likelihood of errors. The system allows for easy retrieval, editing, and printing of data as needed. The authentication-based access system enhances security compared to a manual system. All data is centrally stored, distributed among different departments as required, and backed up on separate servers. Access to the database is strictly authorized, preventing viewing or editing by unauthorized individuals. In summary, this automated and computerized system stands out for its safety, speed, and user-friendly interface.

REFERENCES

[1] Shafie, E., Al-Ajlan, A., Aldrawiesh, K., Bajahzar, A., & Al-Saawy, Y. (2011, January). "Online registration system". In Advanced Computer Control (ICACC), 2011 3rd International Conference on (pp. 174-179).
[2] E. Hart-Baldridge, "Faculty Advisor Perspectives of Academic Advising," NACADA J., vol. 40, no. 1, pp. 10–22, 2020.
[3] M. N. Esraa Afify, "A Proposed Model for a Web-Based Academic Advising System," (International J. Adv. Netw. Appl. (IJANA, vol. 9, no. 2, pp. 3345–3361, 2017.
[4] L. Keston and W. Goodridge, "AdviseMe: An Intelligent Web-Based Application for Academic Advising," Int. J. Adv. Comput. Sci. Appl., vol. 6, no. 8, 2015.
[5] Huang Jiancheng Li Weiwen He Qubo Zhang Wanmao, "Design and implementation of the management system of college students' registration and tuition payment," no. 5, vol. 25, China Medical Education Technology, Xi'an, 2011, pp. 535–537
[6] https://react.dev/learn/adding-interactivity
[7] https://www.mongodb.com/docs/manual/crud/
[8] https://nodejs.org/docs/latest/api/module.html



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

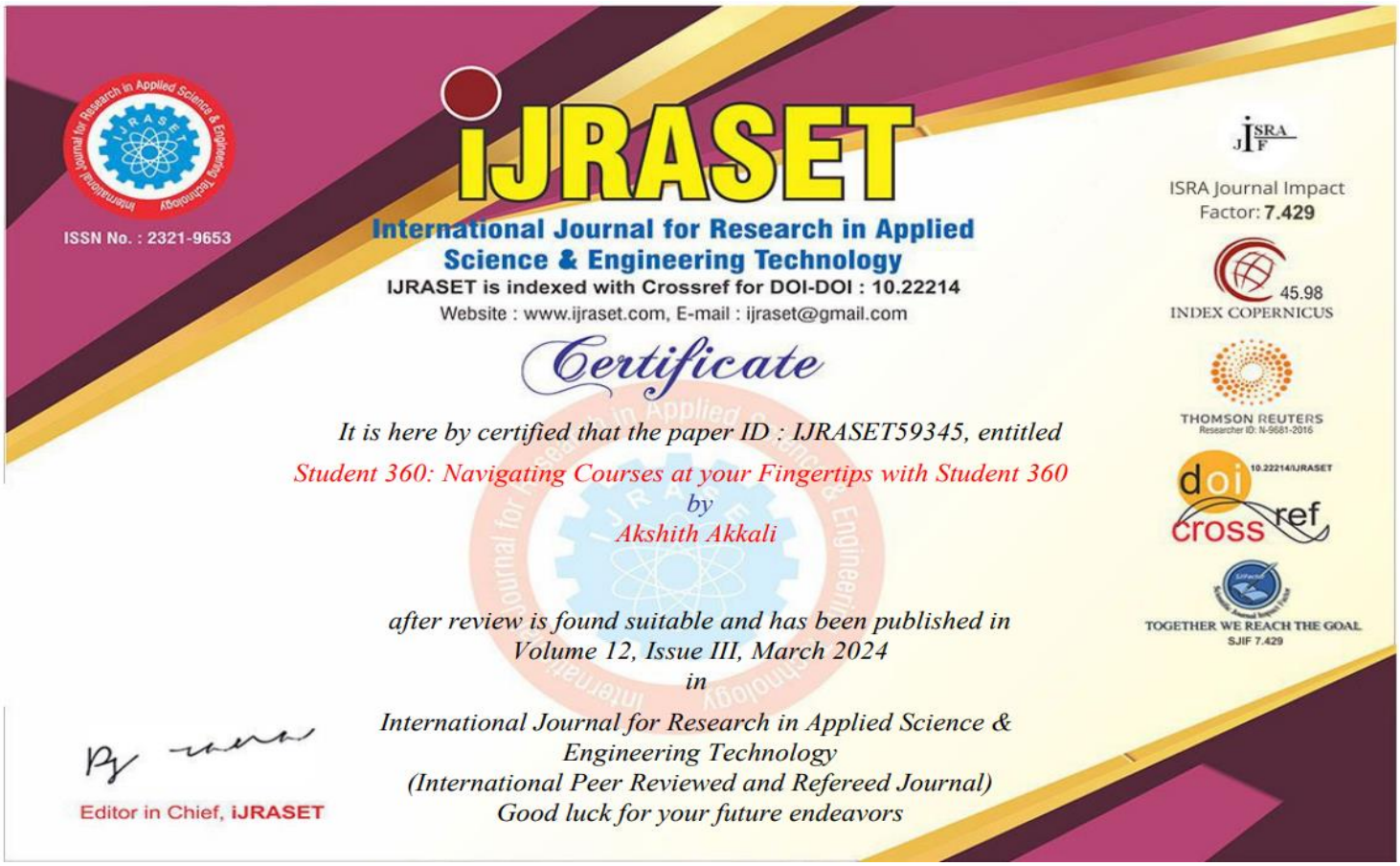
IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
doi 10.22214/IJRASET cross ref
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59345, entitled
Student 360: Navigating Courses at your Fingertips with Student 360
by
M. Shiva Kumar
after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
doi 10.22214/IJRASET cross ref
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59345, entitled
Student 360: Navigating Courses at your Fingertips with Student 360
by
Rishab Agarwal
after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
doi cross ref
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59345, entitled
Student 360: Navigating Courses at your Fingertips with Student 360
by
Akshith Akkali
after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
doi cross ref
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59345, entitled
Student 360: Navigating Courses at your Fingertips with Student 360
by
P. Satwik
after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET