# Chapter 1 INTRODUCTION

## INTRODUCTION

In the world of education, managing academic projects well is super important. It helps students succeed, lets people work together, and helps schools achieve their goals. With more and more schools using technology, there's a big need for websites that can make managing projects easier. That's where our project comes in. We're making a website that's packed with useful features for both students and teachers. It's like a central place where everyone can handle their academic projects.

In today's fast-moving world, technology is everywhere, but old-fashioned ways of managing projects don't always keep up. Things like paper forms, meetings, and messy ways of talking can slow things down. Especially in big schools, keeping track of students, projects, and talking between teachers and students can be tough. Students might struggle to find the information they need, which can slow them down.

Without a single place to manage projects, things can get messy. Important details might end up scattered across different places, making it hard for everyone to find what they need. And without good ways to manage accounts, it's tricky for teachers to keep everything organized.

Our project wants to fix all these problems. We're creating a website that's easy to use for both students and teachers. It'll have features like signing up and logging in, so everyone can access what they need. Teachers will get a special area where they can manage students, projects, and accounts. And students will have a place to see all their projects and important information.

By bringing everything together in one place, we're making it easier for everyone involved in academic projects. Our website will help students succeed, make teachers' lives easier, and keep everything organized. With the help of technology, we're making managing projects simpler and smoother for schools everywhere.

## PROBLEM DESCRIPTION

In schools, managing academic projects usually involves lots of paperwork and talking in person. But these methods can be slow, make mistakes, and are hard to use for big schools with lots of students. For school staff, it's tough to keep track of all the students, their projects, and making sure everyone gets the right information. And for students, it can be hard to find what they need for their projects, which can be frustrating.

Without a single place to manage projects, things get even trickier. Important details might end up all over the place, making it hard for everyone to find what they need. And without good ways to manage accounts, it's tough for staff to keep everything organized.

Our project wants to fix all these problems. We're creating a website that's easy to use for both students and staff. It'll have features like signing up and logging in, so everyone can find what they need. Staff will get a special area where they can manage students, projects, and accounts. And students will have a place to see all their projects and important information.

By bringing everything together in one place, we're making it easier for everyone involved in academic projects. Our website will help students do better, make things easier for staff, and keep everything organized. With the help of technology, we're making managing projects simpler and smoother for schools everywhere.

## OBJECTIVE

Our main goal is to make a website that helps schools manage academic projects better. This website will let school staff keep track of student records, add, remove, and update project details, and manage accounts smoothly. Students will also be able to use the website to see project information, check out the student catalogue, and manage their own accounts easily. By keeping all the information in one place, our website will make sure everything is reliable, easy to access, and helps students and staff work together better.

# Chapter 2

## LITERATURE SURVEY

In recent years, there has been a lot of interest in creating websites to help manage academic projects. These websites are designed to make it easier for students and teachers to work together on projects. Researchers have looked at different aspects of these websites to see how they can help students and teachers. Some studies talk about how these websites can use technology to understand what users are saying. By using things like natural language processing, the websites can understand and respond to user questions without needing manual input. This makes it easier for everyone to use the website and saves time.

Other studies focus on using artificial intelligence (AI) to make these websites smarter. For example, AI can suggest project ideas to students based on their interests and skills. It can also help students keep track of their progress and deadlines, making it easier for them to manage their projects effectively.

Some researchers emphasize the need for these websites to be flexible and adaptable. They suggest using cloud-based storage and collaboration tools to make it easier for students and teachers to work together from anywhere. By integrating advanced features like machine learning, these websites can predict project outcomes and provide personalized recommendations to improve project performance.

Overall, these studies show that websites for academic project management have the potential to make collaboration easier, improve project outcomes, and support student success. By using technology like AI and natural language processing, these websites can offer innovative solutions to help students and teachers work together more effectively.

# Chapter 3 SYSTEM ANALYSIS

## EXISTING SYSTEM

The current system for managing academic projects typically involves a combination of manual processes and digital tools. In many educational institutions, project management tasks are still heavily reliant on traditional methods such as paperwork, face-to-face meetings, and email communication. These methods, while familiar, can be inefficient and prone to various challenges.

One aspect of the existing system is the use of physical paperwork for documenting project details, student records, and communication logs. Project proposals, progress reports, and other project-related documents are often printed out and stored in filing cabinets or shared folders. While this method may seem straightforward, it can lead to issues such as document loss, difficulty in tracking revisions, and limited accessibility. Moreover, maintaining physical records requires significant administrative effort and storage space, adding to the overall complexity of project management.

Another component of the existing system is face-to-face meetings, which play a crucial role in project planning, coordination, and progress review. These meetings typically involve students, faculty members, and other stakeholders gathering in person to discuss project objectives, timelines, and responsibilities. While face-to-face interactions can facilitate effective communication and collaboration, they can also be time-consuming and logistically challenging, especially in larger educational institutions or when participants are located in different geographic locations. Additionally, scheduling conflicts, last-minute changes, and limited meeting space can further exacerbate these challenges, leading to delays and inefficiencies in project management.

### DISADVANTAGES OF EXISTING SYSTEM

1. Time-Consuming:

* Manual processes and face-to-face meetings take up a lot of time.
* Sorting through paperwork and scheduling meetings can be tedious.

1. Communication Challenges:

* Email communication can lead to misunderstandings and delays.
* Important messages may get lost in overflowing inboxes.

1. Limited Accessibility:

* Physical paperwork and email communication may limit accessibility to project information.
* Students and faculty may struggle to access documents and updates when needed.

1. Coordination Issues:

* Coordinating meetings and project activities can be difficult, especially in larger institutions.
* Scheduling conflicts and last-minute changes can disrupt project timelines.

1. Inefficiency:

* Overall, the existing system is inefficient and prone to errors.
* Manual processes and fragmented communication can lead to delays and challenges in project management.

## PROPOSED SYSTEM

Our project is a website that makes it easy for students and administrators to handle academic projects. It's a big upgrade from the old way of doing things, which involved lots of paperwork and emails.

* For Administrators:

Our website lets administrators easily manage the list of student projects. They can add, delete, and see project details quickly.

Administrators can also handle student accounts smoothly. They can add, remove, or update student accounts whenever needed.

* For Students:

Students can log in to the website and see all the available projects in one place. This makes it simple for them to find projects they're interested in.

They can also manage their own accounts, like updating their information, making it easier for everyone to keep things up-to-date.

* Database Integration:

All the information, like project details and user accounts, is safely stored in a central database. This makes sure everything is organized and easy to find.

Using a database helps prevent mistakes and keeps everything running smoothly.

### ADVANTAGES OF PROPOSED SYSTEM

* + - 1. Streamlined Project Management:
         * Our website simplifies the process of managing academic projects for administrators. With user- friendly tools for adding, deleting, and viewing projects, administrators can efficiently handle project- related tasks.
      2. Enhanced Collaboration:
         * The centralized platform fosters collaboration among students by providing easy access to project details and the student catalogue. This encourages teamwork and knowledge sharing among students.
      3. Improved Accessibility:
         * Students benefit from improved accessibility to project information, as they can conveniently view projects and explore the student catalogue from any location with internet access. This ensures that students can stay informed and engaged in project opportunities.
      4. Empowered User Autonomy:
         * Students have the ability to manage their own accounts, allowing them to update their information and preferences as needed. This empowers students to take control of their academic journey and ensures that their accounts remain accurate and up-to-date.
      5. Efficient Data Management:
         * By securely storing all data in a centralized database, our system facilitates efficient data management and retrieval. This minimizes the risk of errors and data duplication commonly associated with manual processes, ensuring data integrity and reliability.
      6. Time and Cost Savings:
         * The streamlined project management processes and improved collaboration facilitated by our system result in time and cost savings for educational institutions. Administrators and students spend less time

on administrative tasks and communication, allowing them to focus more on academic pursuits.

* + - 1. Scalability and Adaptability:
         * Our system is designed to be scalable and adaptable to the changing needs of educational institutions. As the number of projects and users grows, the system can easily accommodate increased demand without compromising performance or user experience.

## SYSTEM REQUIREMENTS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

## SOFTWARE REQUIREMENTS

Development Environment: Visual Studio

Front-End Technologies: HTML, CSS, JavaScript Back-End Technologies: Node.js

Database System: MongoDB Testing Tool: Postman

Operating System: Windows 11 or higher

## HARDWARE REQUIREMENTS

Processor: i3 or higher RAM: 4GB or more

Network: High-speed internet Connection for efficient data transfer

## SOFTWARE TECHNOLOGIES

* + 1. **NODE.JS:**

Node.js is an open-source, cross-platform JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows developers to run JavaScript code on the server-side, enabling the development of scalable and high-performance web applications.

#### Key Features:

Asynchronous and Event-Driven: Node.js uses a non-blocking, event-driven architecture, allowing for efficient handling of I/O operations. This asynchronous model ensures that the server can handle multiple requests simultaneously without blocking the execution of other tasks.

NPM (Node Package Manager): Node.js comes with a vast ecosystem of libraries and modules available through NPM, the official package manager for Node.js. Developers can easily install, manage, and share reusable code modules to enhance the functionality of their applications.

Scalability: Node.js is well-suited for building scalable applications due to its lightweight and efficient nature. It can handle a large number of concurrent connections with minimal resource consumption, making it ideal for real-time applications and microservices architectures.

Single Language: Node.js enables full-stack JavaScript development by allowing developers to use JavaScript for both frontend and backend development. This eliminates the need to switch between different programming languages, streamlining the development process and improving code maintainability.

Community Support: Node.js has a vibrant and active community of developers, contributors, and users who contribute to its growth and improvement. The community provides extensive documentation, tutorials, and resources, making it easier for developers to learn, troubleshoot, and collaborate on Node.js projects.

## HTML:

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as < img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a corner stone technology of the World Wide Web, alongside HTML andJavaScript.CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content.CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition.

CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple stylesheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading. One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different stylesheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

## JAVASCRIPT:

JavaScript s a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object- orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated Java Script engine to execute it. As a multiparadigm language, JavaScript supports event-driven, functional, and imperative (including object oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in webservers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets. The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries.

Scripts written in Vanilla JS are plain JavaScript code. Google's Chrome extensions, Opera's extensions, Apple's Safari 5extensions, Apple's Dashboard Widgets, Microsoft's Gadgets, Yahoo! Widgets, Google Desktop Gadgets.

## MONGODB:

MongoDB is a popular, open-source NoSQL database management system that provides a flexible, scalable, and high-performance solution for storing and managing unstructured or semi- structured data. It is designed to handle large volumes of data and is particularly well-suited for modern, data-intensive applications.

#### Key Features:

Document-Oriented: MongoDB is a document-oriented database, which means it stores data in flexible, JSON-like documents instead of traditional rows and columns. This allows for easy representation of complex hierarchical data structures and supports dynamic schemas, enabling rapid iteration and development.

Schema less Design: MongoDB's schema less design allows developers to store heterogeneous data types within the same collection without the need for a predefined schema. This flexibility makes it easier to adapt to changing data requirements and accommodate evolving application needs.

High Performance: MongoDB is optimized for performance, with features such as in- memory caching, horizontal scaling, and asynchronous I/O operations. It can handle high throughput and low-latency operations, making it suitable for real-time applications and high- volume workloads.

Scalability: MongoDB supports horizontal scalability through sharding, which allows data to be distributed across multiple servers or clusters. This enables the database to scale seamlessly as data volume and traffic increase, ensuring high availability and performance under heavy load.

Rich Query Language: MongoDB provides a powerful query language that supports a wide range of operations, including CRUD (Create, Read, Update, Delete), aggregation, geospatial queries, and full-text search. Developers can easily query and manipulate data using familiar syntax and operators.

## VISUAL STUDIO CODE:

Visual Studio Code, also commonly referred to as VS Code, is a **source-code editor** made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

# Chapter 4 SYSTEM DESIGN

## INPUT DESIGN:

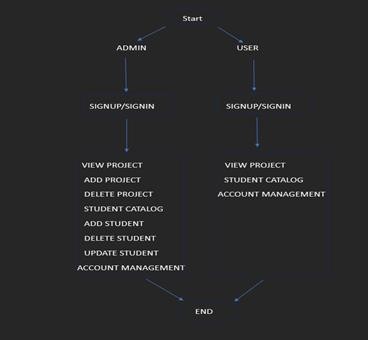
The input design focuses on creating intuitive and user-friendly interfaces for both students and administrators to interact with the system effectively. This involves designing forms and interfaces for functions such as signup, sign in, managing student records, projects, and accounts. For administrators, input design includes forms for adding, deleting, and updating student information and projects. Input validation mechanisms are implemented to ensure that data entered by users is accurate and consistent. Additionally, error handling features are included to guide users in case of incorrect inputs, ensuring a smooth and hassle-free user experience.

## OUTPUT DESIGN:

Output design aims to present information to users in a clear, organized, and visually appealing manner. In your project, output design involves designing interfaces to display project details, student catalogues, and account management information. This includes formatting data for readability, providing relevant visualizations or summaries, and ensuring easy access to information for users. Clear and user-friendly interfaces are designed to enhance the user experience and facilitate efficient navigation through the system. Additionally, feedback mechanisms are incorporated to notify users about the status of their actions, such as successful account creation or project updates. Overall, the output design ensures that users can easily understand and interact with the system's output, leading to a positive user experience.

## SYSTEM ARCHITECTURE

The system architecture of our website is like a blueprint that explains how everything works together. Users interact with the frontend, which is what they see and click on. Behind the scenes, the backend handles all the processing and data storage. The database stores all the information used by the website. To make sure everything runs smoothly, there's a communication layer that connects all the different parts of the system. Overall, these components work together to provide users with a smooth and easy-to-use experience on the website.

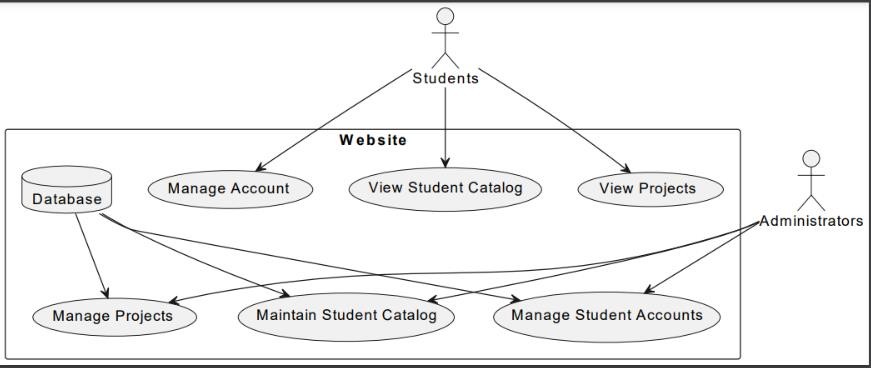


## UML DIAGRAMS FOR PROJECT DESIGN:

UML (Unified Modelling Language) diagrams are like visual blueprints that help to plan and understand how the system works. They use simple symbols and lines to show different parts of the system and how they interact. Diagrams make it easier to design, explain, and build your project.

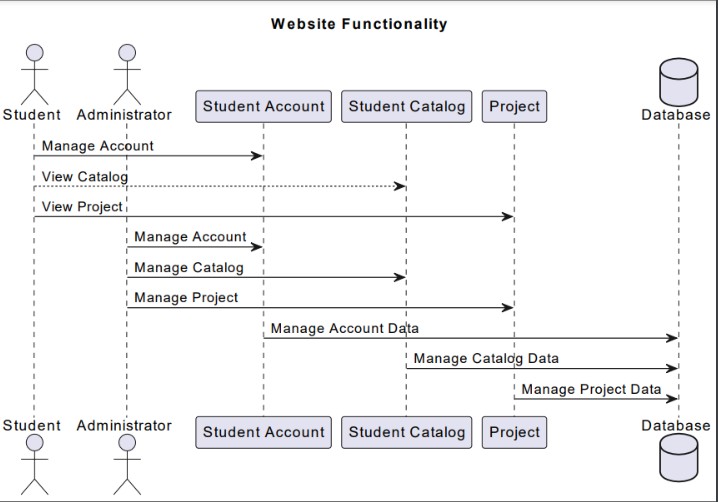
### USE CASE DIAGRAM:

The use case diagram is like a map showing how users interact with the system. It displays the different tasks users can do, such as signing up, logging in, managing student records, and viewing projects. Each task is represented as a box, and lines show how users move from one task to another. This diagram helps understand what the system can do from a user's perspective.



### SEQUENCE DIAGRAM:

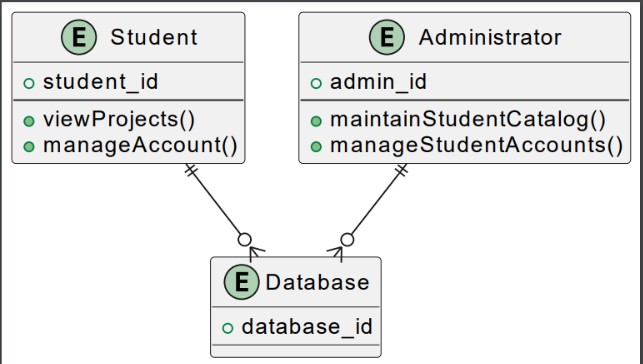
The sequence diagram is like a timeline showing how different parts of the system work together. It shows the order of actions, like a step-by-step guide. For example, it can show how a user signs up for an account, including entering their information, the system processing that information, and finally creating the account. This diagram helps understand how different parts of the system communicate and collaborate.



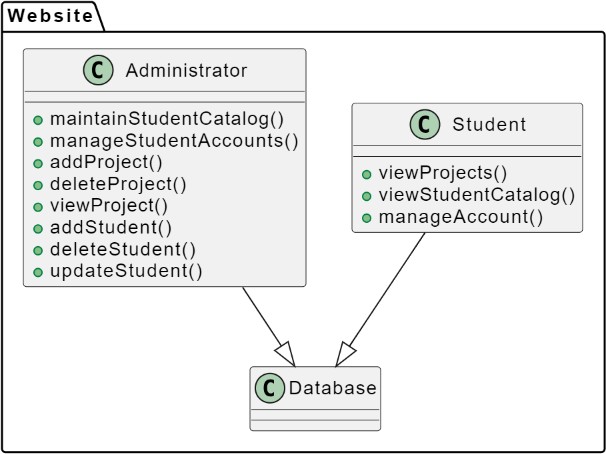
### ENTITY-RELATIONSHIP (ER) DIAGRAM:

The ER diagram is like a blueprint showing how data is organized in the system's database. It shows the different types of data, like students, projects, and accounts, and how they are related to each other. For example, it can show that each student has their own projects, and each project belongs to a student. This diagram helps understand the structure of the database and how

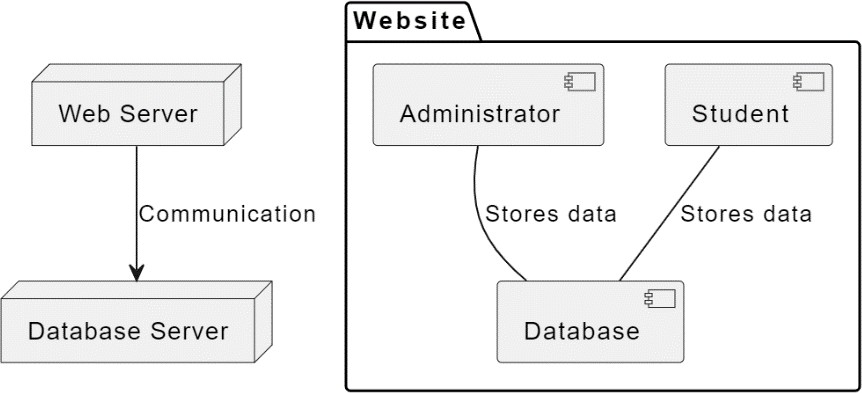
different pieces of information are connected.



### CLASS DIAGRAM:



* + 1. **DEPLOYMENT DIAGRAM:**



# Chapter 5

**SYSTEM CODING**

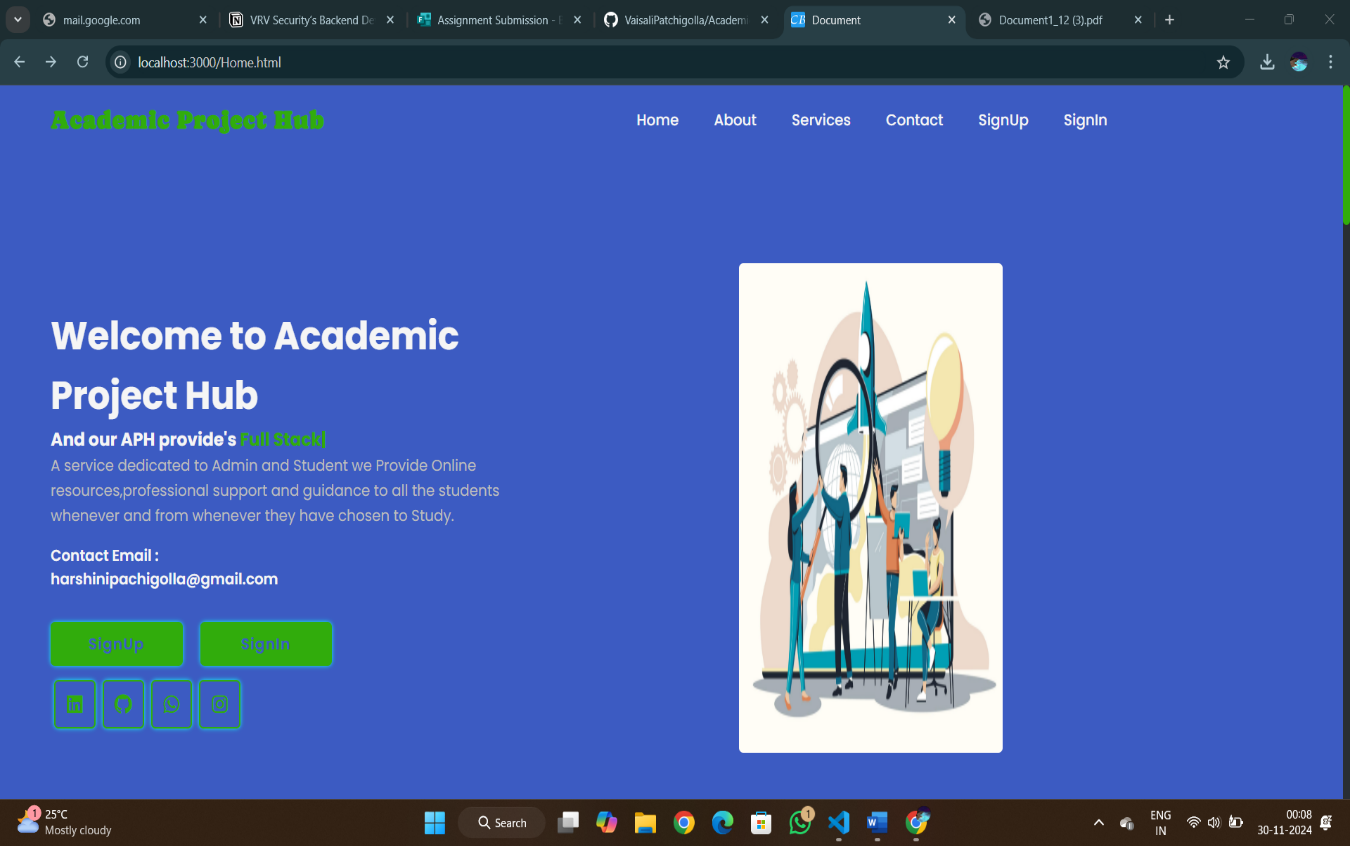
## MODULES:

* + 1. Authentication Module:
       - This module handles the authentication process for both students and administrators.
       - It includes functionalities such as user signup, where new users can create accounts by providing necessary details like username, email, and password.
       - Additionally, the sign in feature allows registered users to log in securely using their credentials.
       - The module ensures authentication security by implementing encryption techniques to protect user data.
    2. Student Management Module:
       - This module is designed to facilitate administrators in managing student records efficiently.
       - It provides functionalities for adding new students to the system, including capturing essential information such as name, email, student ID, etc.
       - Administrators can also delete outdated or inactive student records and update existing information as needed.
       - The module may include features for categorizing students based on criteria such as courses, departments, or academic years.
    3. Project Management Module:
       - This module focuses on managing academic projects within the system.
       - Administrators have the authority to add new projects to the system, providing details such as project title, description, duration, and associated students or groups.
       - The module allows administrators to delete outdated projects or make updates to existing project information.
       - Students can access the project catalogue to browse through available projects, view project details, and express interest or apply to participate in specific projects.
    4. Account Management Module:
       - The account management module oversees the management of user accounts within the system.
       - Administrators have the capability to manage user accounts, including adding new administrators, deleting inactive accounts, and updating account information.
       - The module ensures proper access control, allowing administrators to assign roles and permissions to users based on their responsibilities within the system.
       - It may also include features for password management, allowing users to reset passwords or update account security settings.
    5. Database Management Module:
       - This module handles interactions with the database system, ensuring efficient storage and retrieval of data.
       - It includes functionalities for establishing connections to the database, executing queries to retrieve or modify data, and managing database transactions.
       - The module ensures data integrity and consistency by implementing appropriate database constraints, validations, and error handling mechanisms.
       - It may also include features for database backup and recovery to safeguard against data loss or corruption.

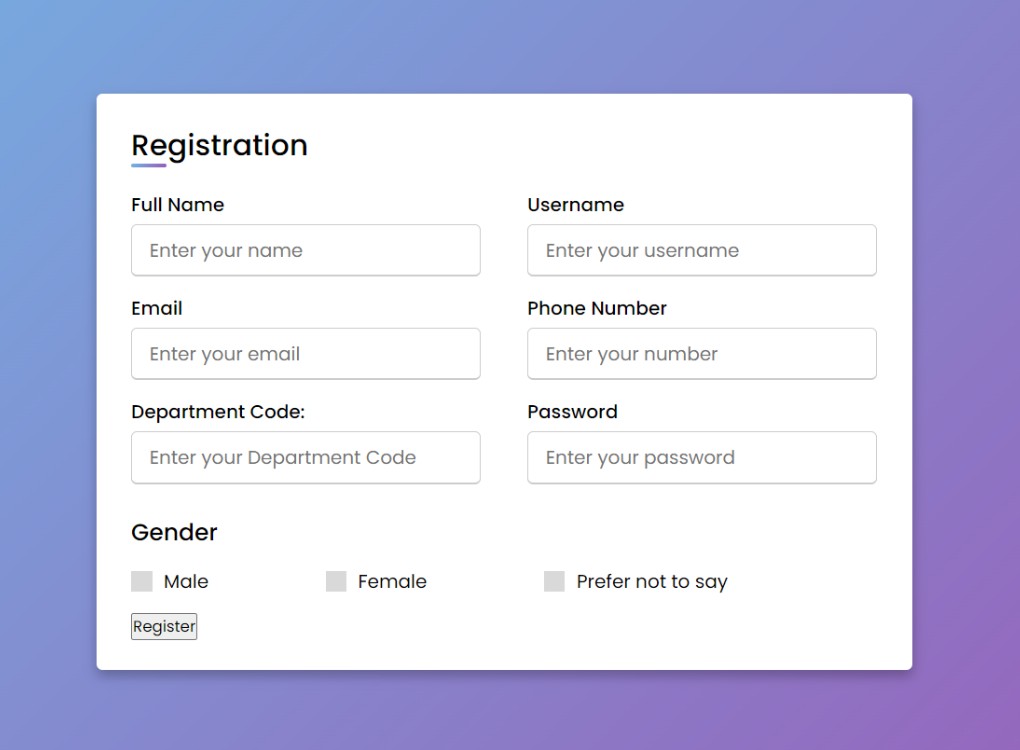
# Chapter 6 SYSTEM TESTING

## SAMPLE OUTPUTS

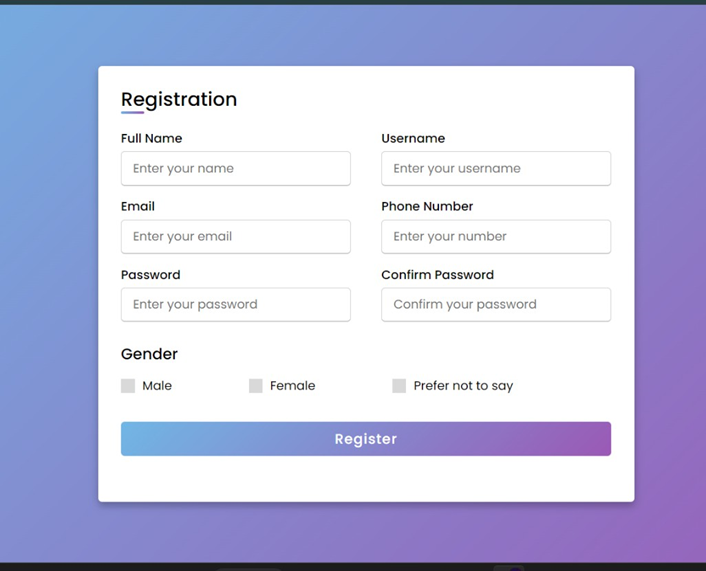
* + 1. Website Home Page:

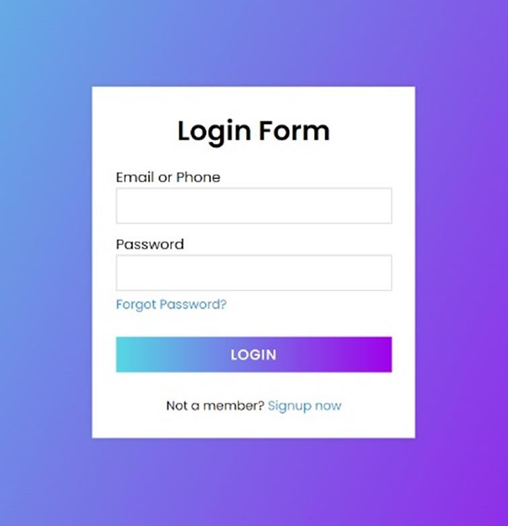


* + 1. Registration for Faculty:

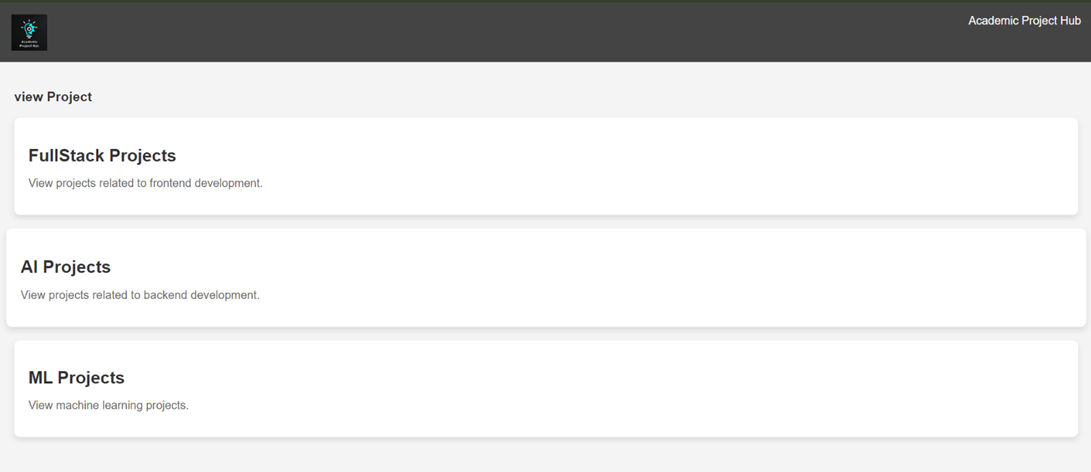


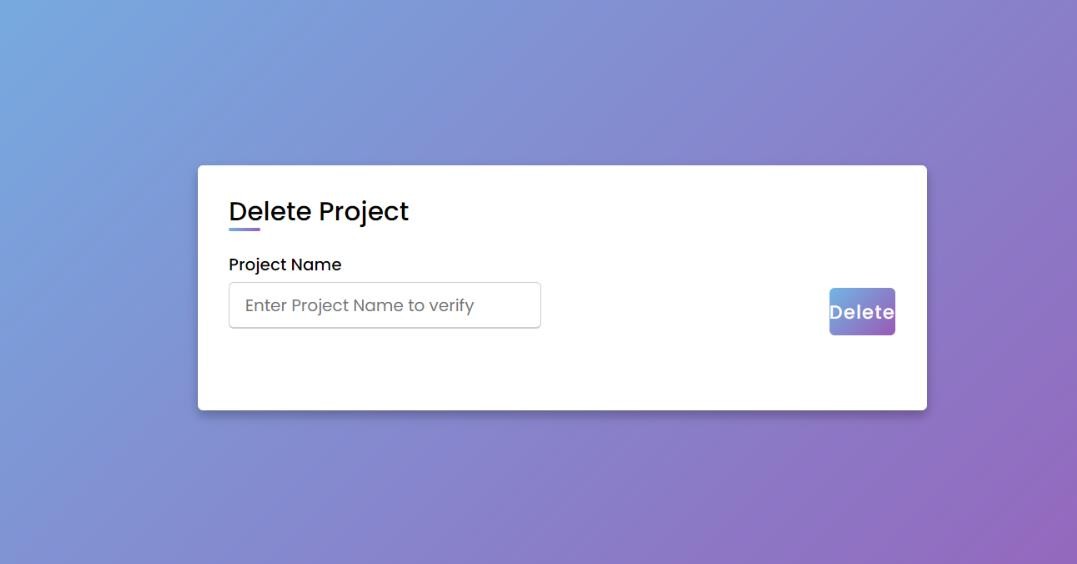
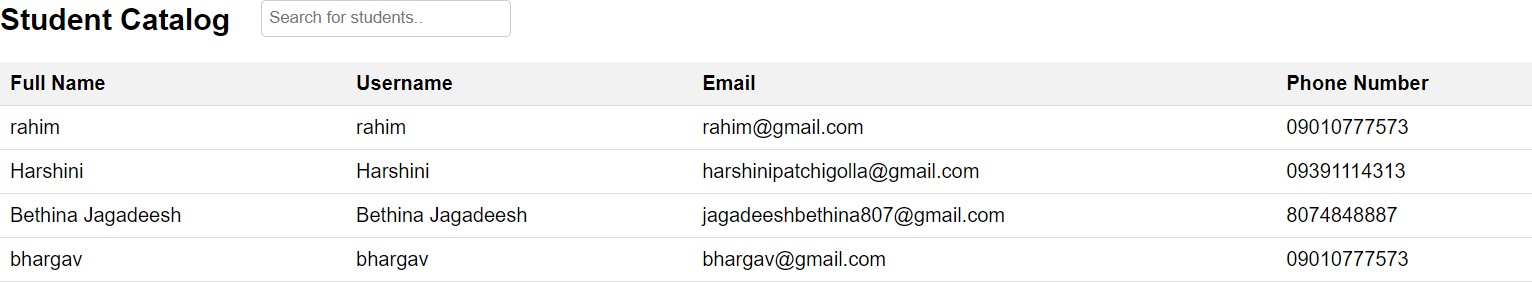
* + 1. Registration for Student:

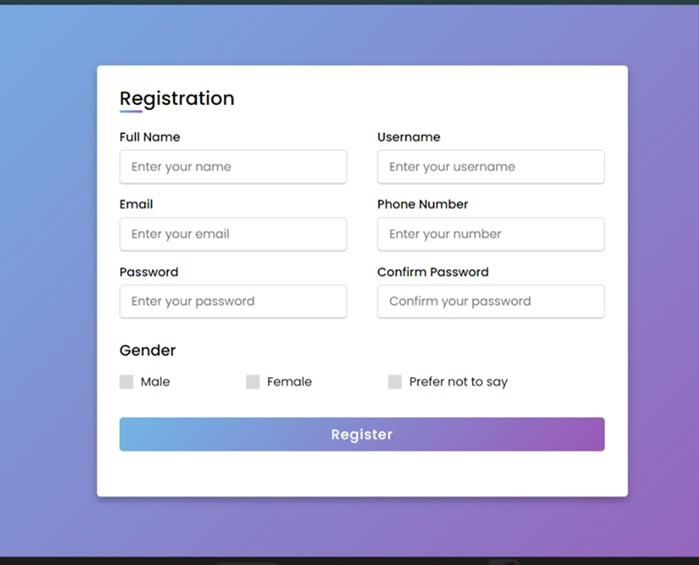
****

* + 1. Login Page:

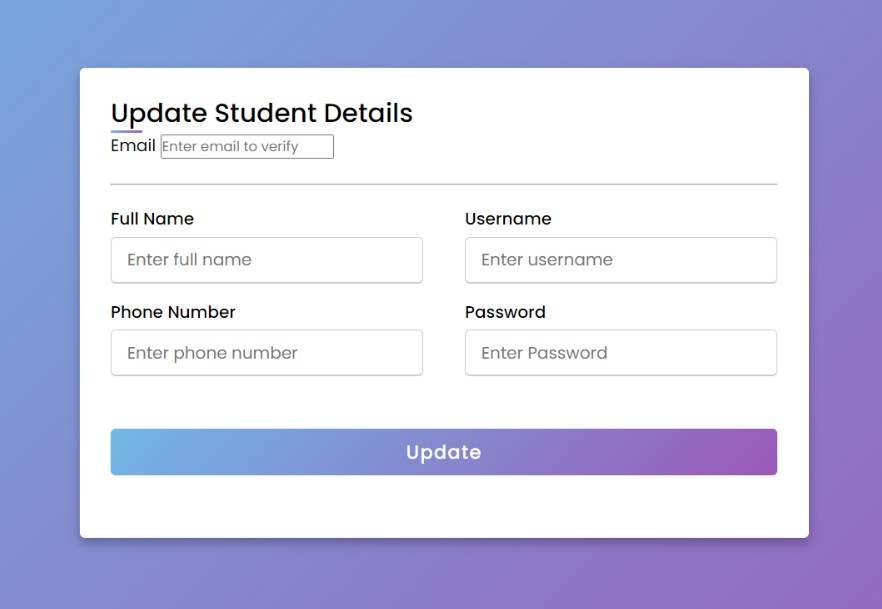
* + 1. Faculty Home Page:
    2. View Projects in Faculty Home Page:

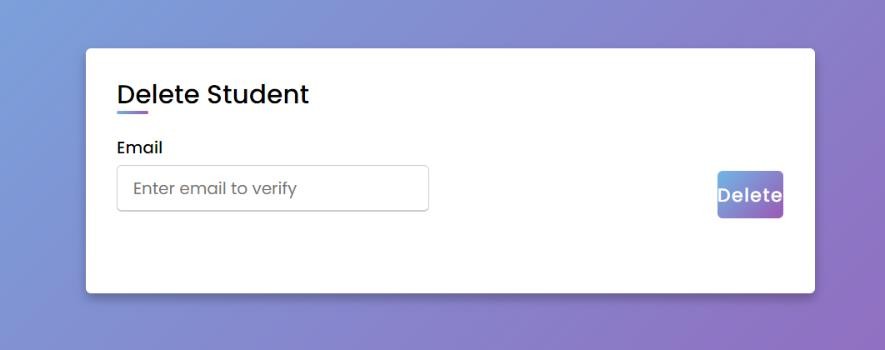
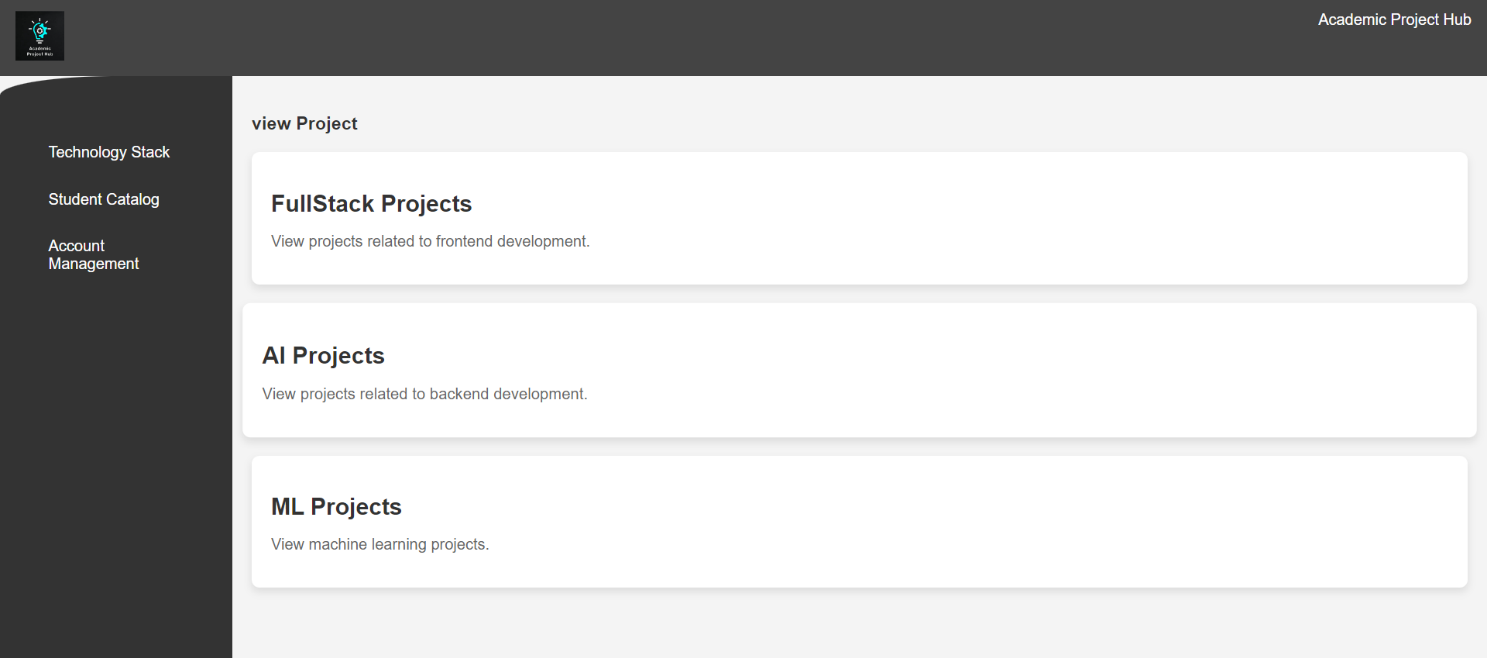
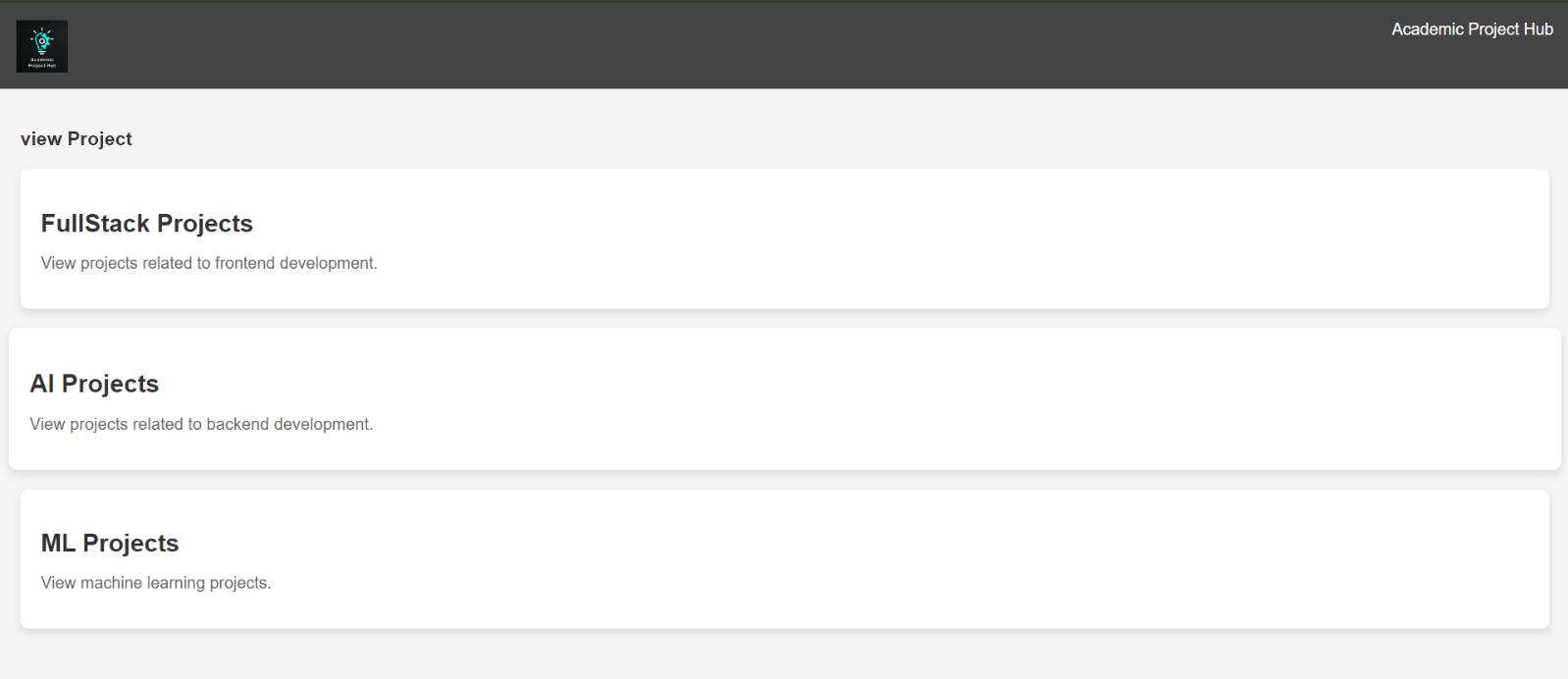


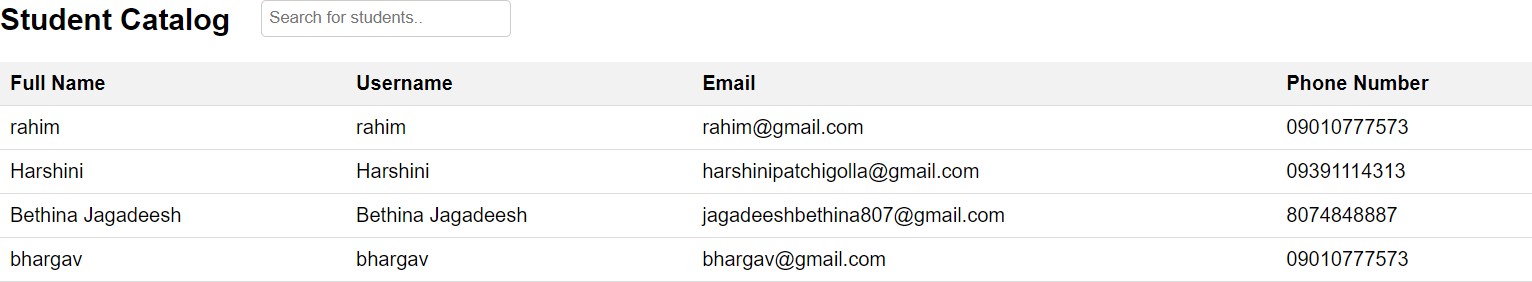
* + 1. Deleting Project in Faculty Home Page:
    2. Student Catalogue in Faculty Home Page:
    3. Adding Student in Faculty Home Page:

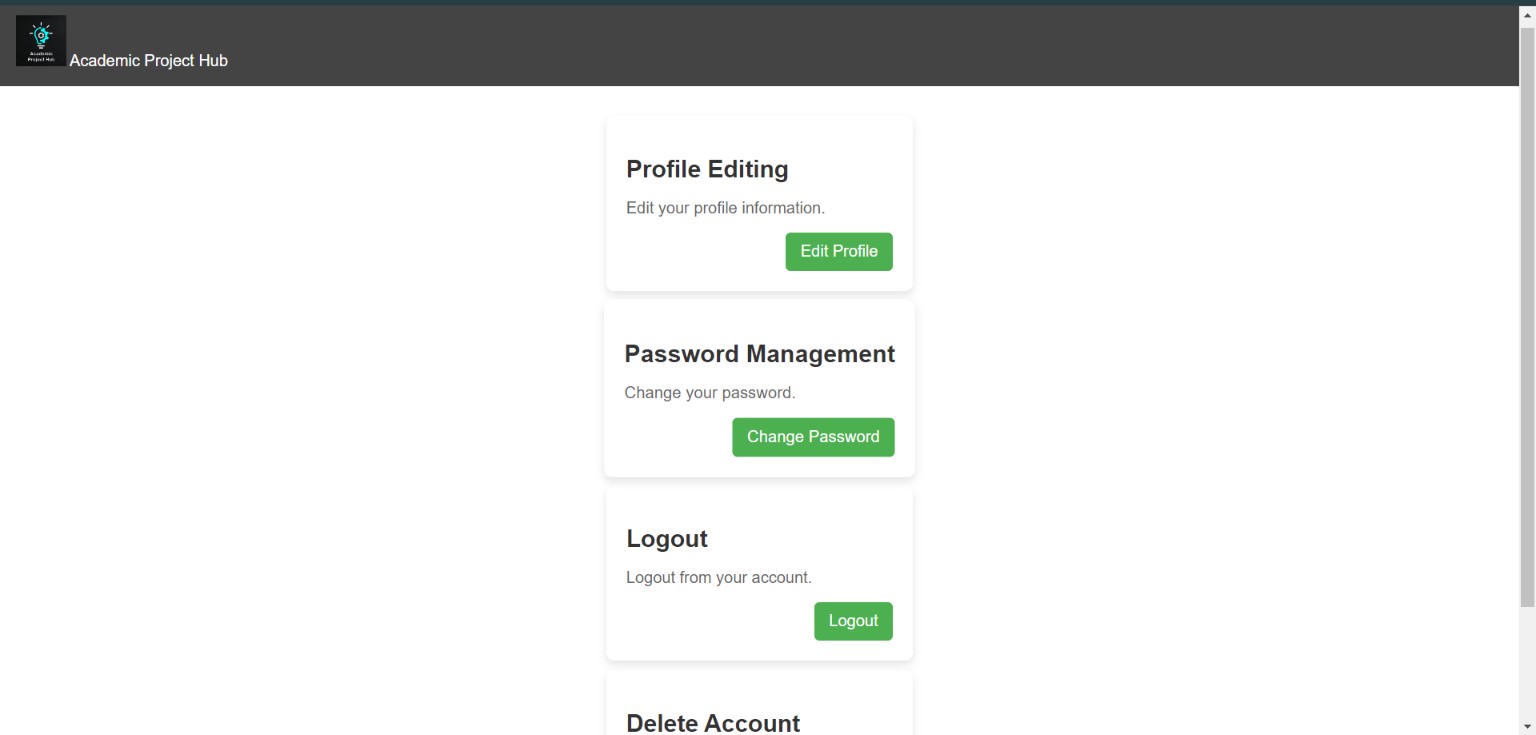


* + 1. Update Student in Faculty Home Page:



* + 1. Delete Student in Faculty Home Page:
    2. User Home Page:
    3. Technology Stack in Student Home Page:
    4. Student Catalogue in Student Home Page:



* + 1. Account Management for both User and Faculty:

# CHAPTER 7 CONCLUSION AND FUTURE WORK

## CONCLUSION:

The culmination of our project signifies a significant step forward in the domain of academic project management. Through the development of a comprehensive website featuring user-friendly signup and sign in functionalities for both students and administrators, as well as modules dedicated to managing student records, projects, and accounts, we have successfully addressed the pressing need for a streamlined and efficient platform within educational institutions. Harnessing the power of cutting-edge technologies such as Node.js and MongoDB, alongside adhering to modern web development practices, has empowered us to craft a robust system that not only meets but surpasses the expectations of our users.

Throughout the development lifecycle, meticulous attention was devoted to ensuring the system's reliability, security, and scalability. Rigorous testing methodologies were employed to validate the system's functionality under diverse scenarios, guaranteeing a seamless user experience. By embracing agile methodologies and incorporating feedback from stakeholders, we iteratively refined and enhanced the system, resulting in a polished and feature-rich platform that resonates with our users.

## FUTURE WORK:

While the current version of the website represents a significant achievement, there are several avenues for future improvement and expansion:

* + 1. Enhanced Collaboration Features: Introduce advanced collaboration features such as real-time messaging, file sharing, and task assignment to facilitate seamless communication and collaboration among project teams.
    2. Integration with External Tools: Explore opportunities for integrating the project management functionalities with popular external tools such as GitHub, Slack, and Google Drive, enabling users to leverage existing workflows and tools seamlessly.
    3. Personalized Recommendations: Implement machine learning algorithms to provide personalized project recommendations based on user interests, skills, and previous project participation, enhancing user engagement and satisfaction.
    4. Accessibility and Localization: Ensure accessibility compliance and provide support for multiple languages to cater to a diverse user base, promoting inclusivity and accessibility.
    5. Performance Optimization: Continuously optimize the performance of the website by identifying and addressing bottlenecks, optimizing database queries, and implementing caching mechanisms to improve response times and scalability.
    6. Analytics and Reporting: Implement comprehensive analytics and reporting functionalities to provide administrators with insights into user engagement, project progress, and system usage, enabling data- driven decision-making and continuous improvement.

By prioritizing these areas of future work, we can further elevate the capabilities and usability of the website, cementing its position as an indispensable tool for academic project management and fostering a culture of collaboration and innovation in educational institutions.

**Thank**

**You**