PROJECT CATALYST

A PROJECT REPORT

For

Major Project (KCA451)
Session (2023-24)

Submitted by

Aditya Sharma (Roll No. 2200290140012)
Aayush Chaudhary (Roll No. 2200290140002)
Abhishek Pawar (Roll No. 2200290140011)
Aditya Singh (Roll No. 2200290140013)

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of

Ms. Neelam Rawat
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(MAY 2024)

DECLARATION

I hereby declare that the work presented in report entitled “Project Catalyst” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University of Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Aditya Sharma

Roll No.: 2200290140012

Name: Aayush Chaudhary

Roll No.: 2200290140002

Name: Abhishek Pawar

Roll No.: 2200290140011

Name: Aditya Singh

Roll No.: 2200290140013

(Candidate Signature)

CERTIFICATE

Certified that Aditya Sharma 2200290140012, Aditya Singh 2200290140013, Abhishek Pawar 2200290140011, Aayush Chaudhary 2200290140002 have carried out the project work having "Project Catalyst" (Major Project-KCA451) for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Aditya Sharma	**2200290140012**
Aayush Chaudhary	**2200290140002**
Abhishek Pawar	**2200290140011**
Aditya Singh	**2200290140013**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Ms. Neelam Rawat
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Project Catalyst

Aditya Sharma, Abhishek Pawar, Aditya Singh, Aayush Chaudhary

ABSTRACT

The "**Project Catalyst**" is a web-based platform designed to facilitate collaboration between students and faculty members within an educational institution. This platform simplifies the process of sharing and approving project ideas. Students can browse a list of available projects, each with a title, description, and faculty member's name. They can then propose projects they're interested in, submitting detailed plans and motivations. Faculty members, on the other hand, have the authority to add project ideas, approve or reject student proposals, and mark projects as unavailable once approved.

The "**Project Catalyst**" is built using **HTML-5**, **CSS, PHP,** and **Bootstrap** for a user-friendly and responsive interface. It addresses the need for transparency and efficiency in project management within educational institutions, streamlining the project selection process.

In summary, the "Project Catalyst" SRS provides a concise overview of a web platform that aims to enhance collaboration and innovation by connecting students and faculty members through a centralized hub for project ideation and management.

ACKNOWLEDGEMENT

I extend my deepest gratitude to my thesis supervisor, Ms. Neelam Rawat (Associate Professor), for her guidance, help, and encouragement throughout my project work. Her enlightening ideas, comments, and suggestions have guided me significantly in completing this project successfully.

Special thanks go to Ms. Sangeeta Arora (Associate Professor) and Dr. Shashank Bharadwaj (Associate Professor), who served as valuable project coordinators, offering invaluable guidance, support, and constructive feedback. Their expertise and insights have significantly contributed to the successful completion of this endeavor.

I am also thankful to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends who have helped me a lot in many critical conditions.

Finally, my heartfelt thanks go to my family members and all those who have directly and indirectly provided me moral support and other kinds of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Aditya Sharma (2200290140012)

Aayush Chaudhary (2200290140002)

Abhishek Pawar (2200290140011)

Aditya Singh (2200290140013)

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1 INTRODUCTION

The "Project Catalyst" is a transformative web-based platform poised to revolutionize manual workflows in education, specifically addressing the process of project ideation, allocation, and performance management. This software redefines how students and faculty members collaborate, shifting from manual processes to a streamlined digital platform.

2 OBJECTIVE

The primary objective of the "Project Catalyst" is to automate and enhance the entire project lifecycle, from idea generation to project completion. By providing a centralized space, it aims to simplify project sharing, approval, allocation, and progress tracking, fostering a more efficient and collaborative educational environment.

3 WORKFLOW

The workflow of the "Project Catalyst" unfolds in a series of seamless steps:

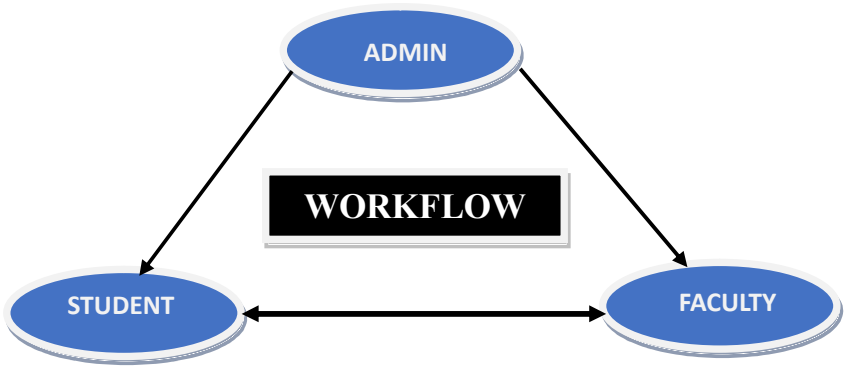

Fig 1.1 Workflow of Project Catalyst

Faculty members initiate the process by logging into the platform and submitting diverse project ideas. These submissions encompass crucial details such as project title, objectives, methodologies, and anticipated outcomes.

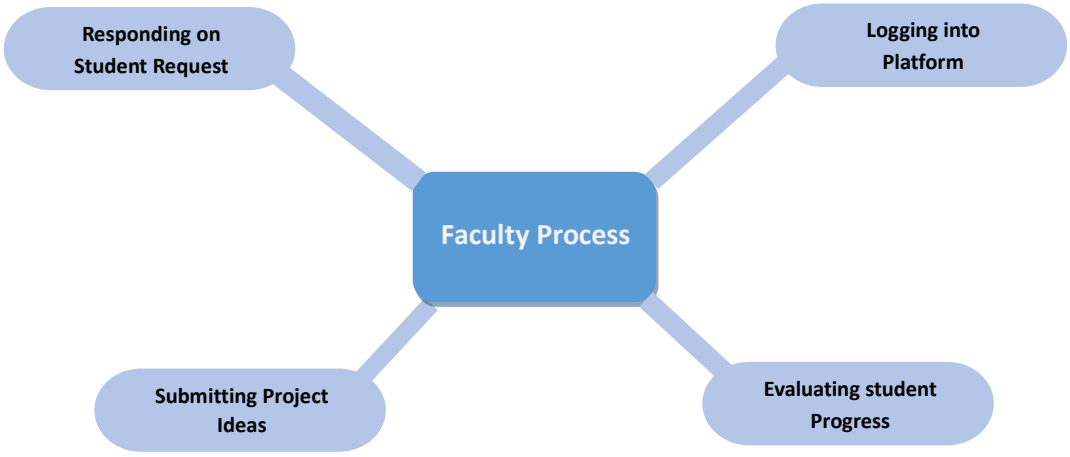

Fig 1.2 Faculty Process

Students, upon logging into the platform, explore the array of available project ideas put forth by faculty members. They meticulously select a project of interest and submit a formal request to work on the chosen project.

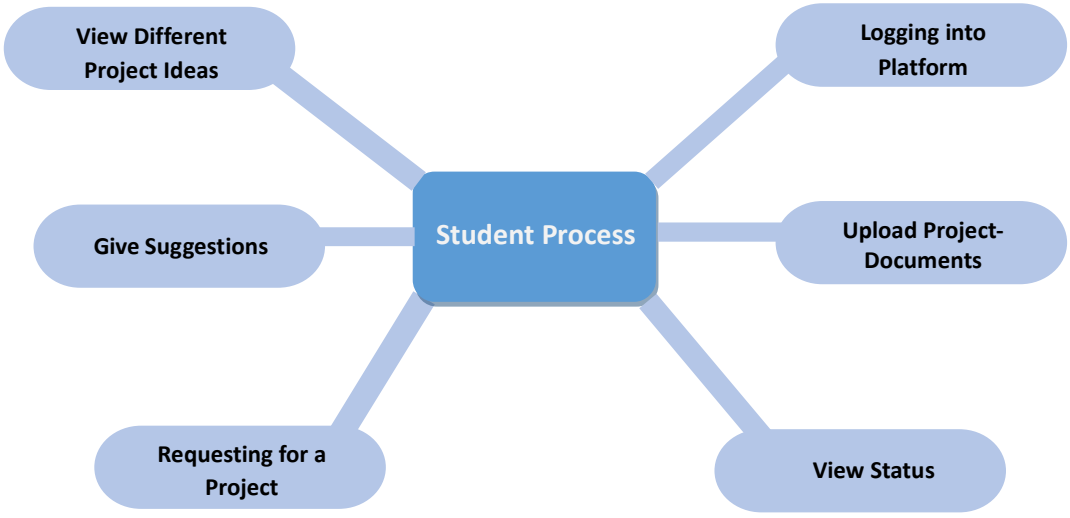

Fig 1.3 Student Process

The "Project Catalyst" platform includes a robust set of features tailored for administrators to efficiently manage the system and its users. Admins can add and remove users, ensuring that only authorized individuals have access to the platform. They can send notifications to users, both individually and in groups, to communicate important updates or reminders. Additionally, admins can generate comprehensive reports that provide insights into user activities and system usage, including student status, faculty status, outgoing communications, request statuses, and suggestions. The admin dashboard serves as a centralized hub, offering a user-friendly interface for overseeing all administrative tasks, monitoring system performance, and maintaining the overall integrity and functionality of the platform. Robust security measures and performance optimization ensure that the platform operates smoothly and securely, even with a substantial number of concurrent users.

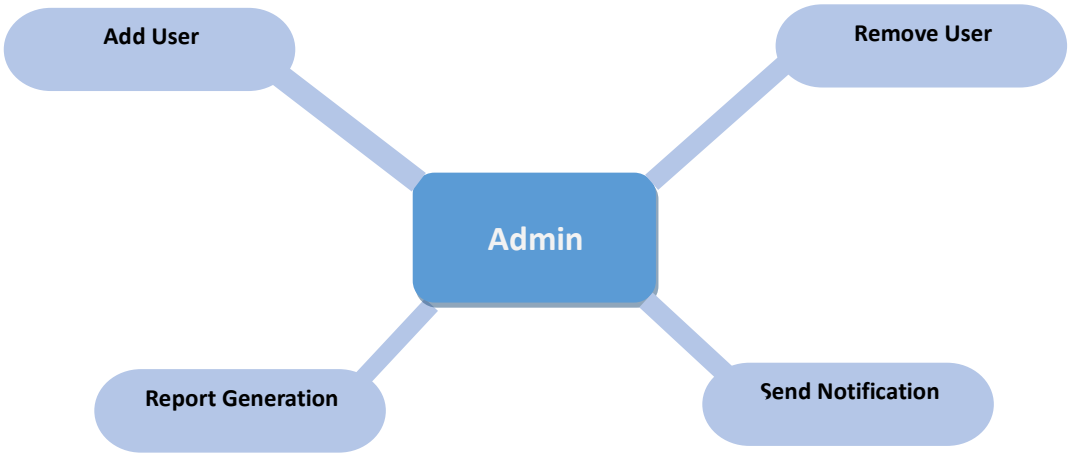

Fig 1.4 Admin Features

The "Project Catalyst" platform includes specialized features for coordinators to streamline the management of student and faculty interactions and overall project oversight. Coordinators can oversee the registration process, ensuring that both students and faculty members are correctly registered and active within the system. They have the ability to monitor and manage project proposals, including tracking the approval and rejection processes. Coordinators can also facilitate communication between students and faculty, ensuring that feedback and updates are effectively shared. Additionally, coordinators can generate reports on project statuses and user activities, providing valuable insights to improve project outcomes. Their dashboard offers a centralized view of all ongoing projects, pending tasks, and critical system alerts, allowing them to maintain smooth and efficient operations while supporting both students and faculty in achieving their project goals. Robust security and performance measures ensure that coordinators can manage these responsibilities efficiently and securely.

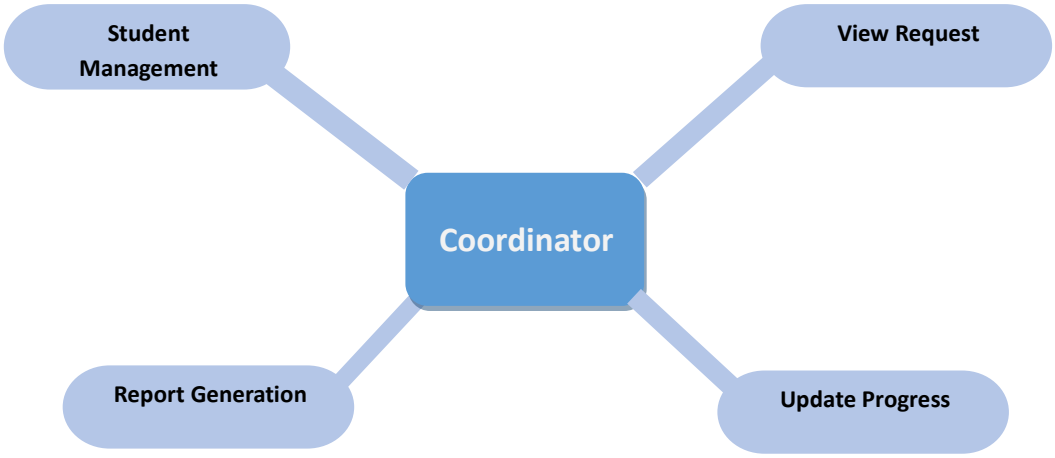

Fig 1.5 Coordinator Features

Faculty members, in turn, receive and review these student requests. The evaluation considers factors such as student qualifications and project suitability. Based on this assessment, faculty members make decisions to either approve or deny the student's request.

Upon approval, the project is officially allocated to the student. Subsequently, the student gains access to a personalized dashboard tailored to the allocated project.

Within their dedicated dashboards, students actively contribute to project progression by uploading essential documents. These documents may include project proposals, research findings, and periodic progress reports. Students leverage their dashboards to stay informed about deadlines and receive valuable feedback from faculty members.

Faculty members, on the other hand, utilize the platform to monitor and assess student progress. They review documents submitted by students, offer constructive feedback, and closely track the overall development of each project.

In a collaborative twist, students are empowered to suggest new project ideas directly to faculty members through the platform. This feature contributes to an environment that nurtures collective ideation and innovation.

Faculty members, being central figures in project management, actively oversee the projects they have proposed. They have the capability to edit project details, monitor student progress, and provide ongoing guidance throughout the project lifecycle.

Lastly, the platform incorporates an admin oversight layer. The admin, possessing overarching control, manages all activities on the platform. This includes the ability to add or remove users, ensuring the smooth functioning and integrity of the "Project Catalyst."

4 **FEATURES**

- Project Idea Submission: Faculty members contribute project ideas, creating a diverse pool for students to choose from.
- Student Request System: Students can request specific projects, aligning their interests with available opportunities.
- Approval Workflow: Faculty members review and approve/deny student requests, ensuring a controlled and transparent project allocation process.
- Student Dashboard: Once a project is approved, students gain access to a dedicated dashboard for document uploads and progress tracking.
- Progress Monitoring: Faculty members can efficiently monitor student progress and evaluate document submissions through the platform.
- Idea Suggestion: Students have the ability to suggest new project ideas, fostering a collaborative environment for innovation.
- Admin Control: An admin login provides overarching control, allowing management of all activities and user permissions.

5 ARCHITECTURE

The architecture of the "Project Catalyst" is designed as a client-server model with a web-based interface.

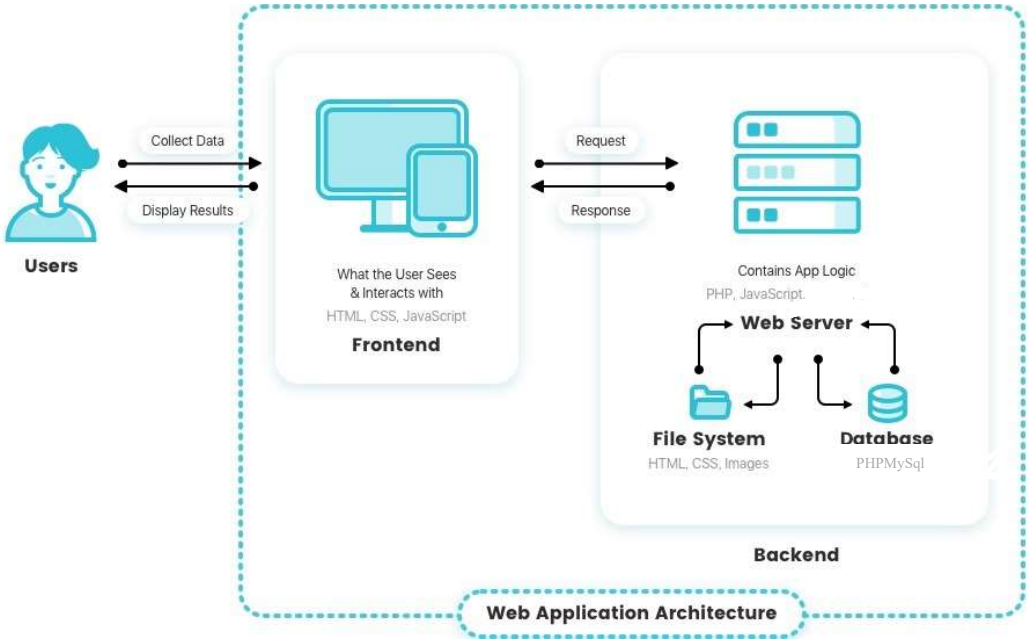

Fig 1.6 Client-Server Architecture Diagram

The front end is built using HTML, CSS, and Bootstrap for a responsive and user-friendly experience. The backend is implemented using PHP, ensuring dynamic content generation and seamless interactions. The system's database, managed through MySQL, stores project details, user information, and progress tracking data.

CHAPTER 2

Literature Review

In the dynamic landscape of education, the transformative power of digital platforms cannot be overstated. These platforms have evolved into indispensable tools that not only facilitate collaboration but also serve as catalysts for innovative project ideation and management, fostering a seamless connection between students and faculty members. A comprehensive exploration of existing research highlights the compelling positive impact these platforms have on project outcomes and student engagement.

Efficiency and transparency, perennial concerns in the realm of project management, have found a promising solution in the form of the "Project Catalyst." This innovative hub acts as a centralized hub for project ideation, submission, and approval processes, aligning seamlessly with institutional goals. By addressing these crucial aspects, the "Project Catalyst" emerges as a beacon of streamlined project management in the educational sphere.

The significance of fostering collaboration between students and faculty cannot be overstated. Research consistently demonstrates that collaborative projects not only enhance interpersonal interactions but also contribute to improved learning outcomes. The "Project Catalyst" strategically capitalizes on this finding by promoting and facilitating meaningful collaboration through its advanced digital platform.

As technology continues to weave its way into the fabric of education, the "Project Catalyst" stands out as a prime example of leveraging cutting-edge tools. By employing HTML, CSS, PHP, and Bootstrap, this hub exemplifies the fusion of technology and education, creating a user-friendly environment that enhances the overall educational experience.

In essence, the "Project Catalyst" not only aligns with but also propels forward the current trends in educational technology. This review underscores the hub's relevance and potential impact by providing an efficient, transparent, and collaborative project management system tailored for educational institutions. As we navigate the ever-evolving landscape of education, the "Project Catalyst" emerges as a pivotal player, poised to shape the future of collaborative learning.

CHAPTER-3

SYSTEM REQUIREMENTS AND SPECIFICATION

3.1 Purpose

The "Project Catalyst" is a web-based platform designed to facilitate the sharing, approval, and management of project ideas for students and faculty members within an educational institution.

3.2 Scope

3.2.1 In-Scope

- User registration and authentication.
- Students can view available project ideas.
- Students can propose project ideas.
- Students can check the status of their proposed projects.
- Faculty members can add, edit, or delete project ideas.
- Faculty members can approve or reject student project proposals.
- Faculty members can mark projects as unavailable.

3.2.2 Out-of-Scope

- Advanced project collaboration features (e.g., team formation).
- Financial transactions.
- Integration with external systems.

3.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- HTML: HyperText Markup Language

- CSS: Cascading Style Sheet
- PHP: Hypertext Preprocessor
- Bootstrap: A front-end framework for web development

3.4 System Overview

3.4.1 System Architecture

The system will consist of a front-end developed using HTML, CSS, and Bootstrap. The back end will use PHP for server-side processing and interact with a MySQL database for data storage.

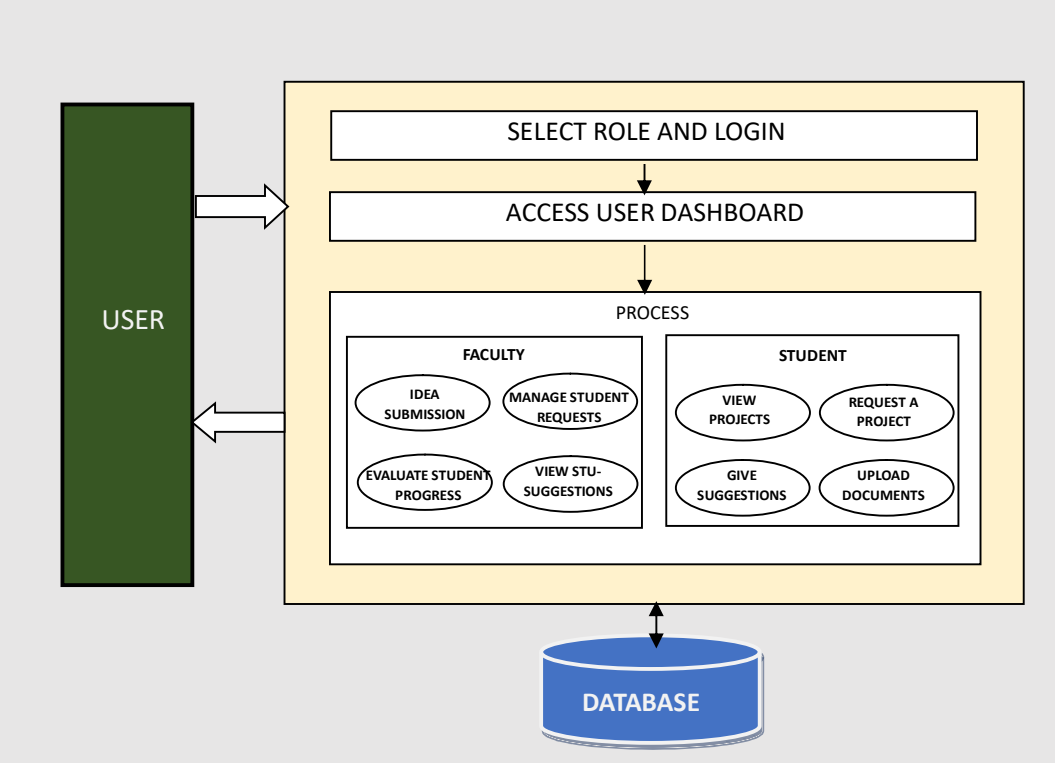

Fig 3.1 System Architecture

3.4.2 Users

- **Students:** Users who can browse, propose, and track project ideas.

- **Faculty Members:** Users with additional privileges to manage project ideas and approve student proposals.

3.5 Functional Requirements

3.5.1 User Registration and Authentication

Description:

- Users need to register and log in to access the system**.**

Functional Requirements:

Registration:

- Users must provide their name, email, and password during registration.
- The system should validate the email format and check for duplicate emails.
- Passwords should be securely hashed and stored in the database.

Authentication:

- Registered users must log in using their email and password to access the system.
- Implement secure authentication mechanisms to protect user credentials.
- Include features for password recovery and reset.

3.5.2 Student Features

3.5.2.1 View Available Projects

Description:

Students need to view a list of available project ideas to choose from**.**

Functional Requirements:

Display a list of project ideas with the following details for each project:

- Title: The name of the project.
- Description: A brief overview of the project idea.
- Faculty Member: The name of the faculty member overseeing the project.
- Request Button: A button that students can click to propose the project.
- Status: Indicates whether the project is available or unavailable.
- Allow students to filter and search projects based on keywords or faculty members.

3.5.2.2 Propose a Project

Description:

Students should be able to propose an available project to a faculty member.

Functional Requirements:

- Students can select a project from the available list and submit a proposal.
- The proposal should be sent to the corresponding faculty member for their approval.
- Include a confirmation message upon successful proposal submission.
- Track the status of the proposal (pending, approved, rejected).

3.5.2.3 Check Proposal Status

Description:

Students need to track the status of their project proposals.

Functional Requirements:

Display a list of submitted proposals with their status:

- Approved: The proposal has been accepted by the faculty member.
- Rejected: The proposal has been declined by the faculty member.
- Pending: The proposal is awaiting review by the faculty member.

For rejected proposals, display feedback or comments provided by the faculty member.

Notify students of status changes via email or in-system notifications.

3.5.2.4 Give Suggestions

Description:

- Students should have the ability to give suggestions related to the projects they are working on or proposing.
- Suggestions can include ideas for project improvement, additional requirements, or general feedback.

Functional Requirements:

- A text input field should be provided for students to enter their suggestions.
- Students should be able to attach files or documents related to their suggestions.
- Each suggestion should be linked to a specific project.

Security and Validation:

- Ensure suggestions are stored securely.
- Validate input to prevent malicious content.

3.5.2.5 Check Suggestion Status

Description:

- Students should be able to track the status of the suggestions they have made.
- Statuses can include reviewed, in-progress, or addressed.

Functional Requirements:

- A section in the student dashboard to view all submitted suggestions.
- Each suggestion should display its current status and any feedback or comments from faculty members.
- Notifications should be sent to students when the status of a suggestion changes.

3.5.2.6 Manage Dashboard

Description:

Students need a personalized dashboard to manage their activities within the system.

Functional Requirements:

- Display an overview of active projects, pending proposals, and suggestions.
- Provide links to detailed views of proposals and suggestions.
- Include notifications, upcoming deadlines, and important announcements.

User Interface:

- The dashboard should be intuitive and easy to navigate.
- Widgets or sections for each functional area (projects, proposals, suggestions, notifications).

3.5.2.7 Upload Suggestions

Description:

Students should have the capability to suggest new project ideas to a faculty member if a particular project idea is not listed.

Functional Requirements:

- Provide a form where students can submit their new project idea suggestions.

- Fields should include project title, detailed description, and the faculty member they wish to propose the idea to.
- Allow students to attach relevant documents or files to support their project idea suggestion.
- The suggestion will be sent to the selected faculty member for review and approval.

User Interface:

- A suggestion submission form with input fields for project title, description, and faculty selection.
- An upload button or drag-and-drop area to add supporting documents.
- A submit button to send the suggestion to the chosen faculty member.

Security and Validation:

- Ensure that the suggestion content and attached files are securely stored.
- Validate input fields to prevent malicious content and ensure completeness (e.g., title and description should not be empty).

3.5.2.8 View Notifications

Description:

Students should receive notifications for various actions and updates within the system.

Functional Requirements:

- Display notifications for proposal status changes, suggestion feedback, project updates, and system announcements.
- Notifications should be real-time and accessible from the dashboard.
- Allow students to mark notifications as read or delete them.

User Interface:

- Notifications should be easily accessible, possibly through an icon or section on the dashboard.
- Provide clear and concise information in each notification.

3.5.3 Faculty Member Features

3.5.3.1 Add Project Ideas

Description:

- Faculty members can add new project ideas to the system.

Functional Requirements:

- Provide fields for project title, description, and status (available).
- Allow faculty to edit or delete their project ideas.
- Display a confirmation message upon successful addition.

3.5.3.2 Approve/Reject Student Proposals

Description:

- Faculty members need to review and decide on student project proposals.

Functional Requirements:

- Display a list of student proposals with options to approve or reject.
- Provide a comment box for feedback when rejecting a proposal.
- Update the proposal status and notify the student.

3.5.3.3 Mark Projects as Unavailable

Description:

Faculty members can mark projects as unavailable once they are no longer open for new proposals.

Functional Requirements:

- Allow faculty to change the status of a project to unavailable.
- Ensure unavailable projects are not visible to students.
- Notify students who proposed the project of the status change.

3.5.3.4 Check Student Proposal

Description:

Faculty members need the ability to review detailed student proposals.

Functional Requirements:

- Provide a detailed view of each proposal, including student name, project title, description, and attached documents.
- Display the proposal's current status and any previous feedback.

3.5.3.5 Manage Faculty Dashboard

Description:

Faculty members should have a personalized dashboard for managing their projects and interactions with students.

Functional Requirements:

- Display summaries of active projects, pending proposals, and recent suggestions.
- Provide quick links to add projects, review proposals, and send notifications.
- Include notifications and alerts for pending actions.

3.5.3.6 View Student Documents

Description:

- Faculty members need to access and review documents submitted by students.

Functional Requirements:

- Display a list of documents associated with a project.
- Provide options to download or view documents within the system.
- Organize documents by project and submission date.

3.5.3.7 View Suggestions

Description:

Faculty members should be able to view new project suggestions made by students.

Functional Requirements:

- Display a list of suggestions with details such as content, student name, and associated project.
- Allow sorting and filtering of suggestions based on status (new, reviewed).

3.5.3.8 Approve or Deny Suggestions

Description:

Faculty members can approve or deny new project suggestions made by students.

Functional Requirements:

- Provide options to approve or deny suggestions with a comment box for feedback.
- Update the suggestion status and notify the student of the decision.

3.5.3.9 Send Notifications

Description:

Faculty members need to send notifications to students regarding project updates and proposal statuses.

Functional Requirements:

- Create and send notifications to individual students or groups.
- Include fields for notification title, message content, and recipient selection.
- Track and confirm delivery of notifications.

3.5.4 Admin Features

3.5.4.1 Add New Users

Description:

Admins need the ability to add new users (students and faculty members).

Functional Requirements:

- Provide fields for user name, email, role (student or faculty), and initial password.
- Validate email format and check for duplicates.
- Send a welcome email with login instructions.

3.5.4.2 Remove Any Existing User

Description:

Admins must be able to remove any user from the system.

Functional Requirements:

- Display a list of users with search and filter options.
- Provide a delete button next to each user entry with a confirmation prompt.
- Remove all associated data of the user.

3.5.4.3 Send Notifications

Description:

Admins need to send notifications to individual users or groups.

Functional Requirements:

- Create and send notifications with fields for recipient selection, title, and message content.
- Track sent notifications and confirm their delivery status.

3.5.4.4 Generate Reports

Description:

Admins should generate various reports to monitor system usage and user activity.

Sub-Features:

4.1 Student Status Report:

- Show the status of all students, including active, inactive, and pending registrations.
- Include details such as student name, email, registration date, and current project involvement.

4.2 Faculty Status Report:

- Show the status of all faculty members, including active, inactive, and pending registrations.
- Include details such as faculty name, email, and the number of projects they oversee.

4.3 Out Status Report:

- Provide an overview of all outgoing communications and actions taken by the system.
- Include details such as message type, recipients, and delivery statuses.

4.4 Request Status Report:

- Track all project proposals and their statuses (approved, rejected, pending).
- Include details such as student name, project title, faculty assigned, and proposal dates.

4.5 Suggestion Report:

- Show all suggestions made by students, their statuses (reviewed, pending), and outcomes (approved, denied).
- Include details such as suggestion content, student name, associated project, and faculty comments.

4.6 Master Report:

- Comprehensive report that includes an overview of all the above reports.
- Include summary statistics and detailed breakdowns.

3.5.4.5 Manage Admin Dashboard

Description:

Admins require a centralized dashboard to manage system activities and monitor overall performance.

Functional Requirements:

- Display key metrics and summaries such as the number of active users, pending requests, and recent notifications.
- Provide quick links to user management, notification center, and report generation.
- Include widgets or sections for recent activities, system alerts, and task reminders.

3.6 Non-Functional Requirements

3.6.1 Performance

The system should handle a large number of concurrent users without significant performance degradation.

Response times should be within acceptable limits.

3.6.2 Security

- User data, especially passwords, should be stored securely and transmitted over HTTPS.
- Authentication and authorization mechanisms should be robust to prevent unauthorized access.

3.6.3 Usability

- The user interface should be intuitive and responsive, compatible with different devices and browsers.

3.7 Constraints

- The project must adhere to all relevant data protection and privacy regulations.
- Development should use HTML, CSS, PHP, Bootstrap, and MySQL as specified.

3.8. Project Timeline

25th May 2024

3.9 Conclusion

This Software Requirements Specification outlines the essential features and functionality of the "Project Catalyst" project. It will serve as a foundation for the development team to design and implement the system according to the specified requirements.

CHAPTER-4

ANALYSIS

Systems analysis is the process by which an individual studies a system such that an information system can be analysed, modelled, and a logical alternative can be chosen. Systems analysis projects are initiated for three reasons: problems, opportunities, and directives.

4.1 Requirement Gathering

The success of the "Project Catalyst" platform relies on a comprehensive understanding of user needs and system functionalities. Requirements gathering involves extensive communication with stakeholders to identify and document key features and specifications. The following categories of requirements were identified:

4.1.1 User Requirements

User Registration: The system must allow students and faculty members to register securely, providing unique user credentials.

Project Proposal Submission: Students should be able to submit detailed project proposals, including objectives, documentation, and any supporting materials.

Proposal Review: Faculty members need the ability to review submitted proposals, providing feedback to students for refinement.

Approval/Rejection: The system should facilitate a straightforward process for faculty members to approve or reject project proposals based on predefined criteria.

Project Management: Faculty members must be able to manage approved projects efficiently, with functionalities for addition, editing, and deletion.

4.1.2 Functional Requirements

Collaboration Hub: The platform should offer a centralized space for effective collaboration among students and faculty members, promoting communication and idea exchange.

Approval Workflow: A streamlined workflow should be implemented to facilitate efficient project proposal review and approval processes.
User-Friendly Interface: The system's interface must be intuitive and user-friendly, accessible to both students and faculty members, promoting ease of use and engagement.

4.1.3 Non-Functional Requirements

Security Measures: Robust security measures should be implemented to ensure the confidentiality of user data, preventing unauthorized access and safeguarding sensitive information.
Performance Optimization: The system must be optimized to handle a substantial number of concurrent users without compromising response times, ensuring a seamless user experience.
Compliance with Development Technologies: Adherence to specified development technologies, including HTML, CSS, PHP, Bootstrap, and MySQL, is crucial for consistency and compatibility.

4.2 SYSTEM DESIGN

The system design for the "Project Catalyst" platform involves defining the architectural structure, system flow, and technology stack.

Architectural Structure:

- Multi-tier Architecture: Includes presentation, business logic, and data access layers.

System Flow:

- User Interaction: User inputs are processed and validated.
- Data Handling: logic processes data and communicates with the database.

Technology Stack:

- Front-End: HTML CSS and Bootstrap
- Back-End: PHP
- Database: MySql

4.3 DATA MODELING

The data model for "Project Catalyst" involves creating and organizing database tables.

Database Tables:

- User Information: Stores user ID, name, email, and hashed password.
- Project Details: Includes project ID, title, description, and status.
- Suggestion: Contains Suggestion ID, user ID, project ID, and content.

Relationships Between Tables:

- Users and Projects: One-to-one relationship.
- Student and Suggestion: One-to-many relationship.
- Students and Faculty: Many-to-one relationship.

These relationships ensure data integrity and efficient data management.

4.4 MOCKUPS AND PROTOTYPES

Mock-ups and prototypes of the user interface are developed to visually represent the system's look and feel. These visual aids serve as a reference for developers and provide stakeholders with a tangible understanding of the platform's design.

4.5 FEASIBILITY ANALYSIS

A feasibility analysis is conducted to assess the viability of the "Project Catalyst" platform in terms of technical, operational, and economic factors. This involves evaluating the technical capabilities of the chosen technologies, assessing operational requirements, and estimating the overall cost of development and maintenance.

This chapter outlines the process of requirements gathering, system design, data modelling, and feasibility analysis for the "Project Catalyst" platform. The subsequent chapters will delve into the implementation, testing, and deployment phases, providing a comprehensive overview of the system's development lifecycle.

CHAPTER-5

SYSTEM DESIGN

5.1 **Introduction**

The design of the "Project Catalyst" is meticulously crafted to address the dynamic and collaborative nature of academic project management. This section provides a comprehensive overview of the system architecture, modules, data flow, and user interface, emphasizing the underlying principles driving the design decisions.

5.2 **System Architecture**

The "Project Catalyst" adopts a robust client-server architecture to ensure scalability, responsiveness, and efficient data management. The detailed architecture can be shown in the below diagram.

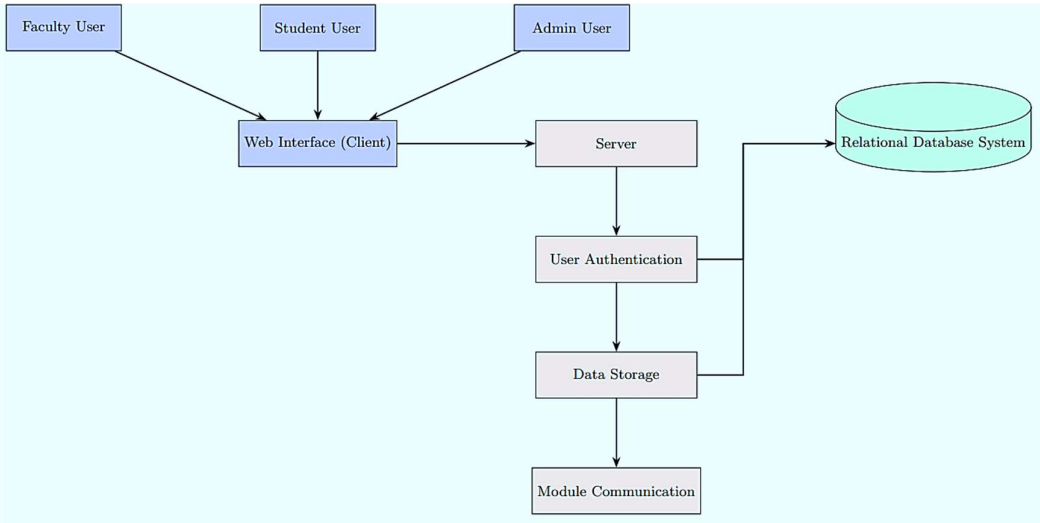

Fig 5.1 System Architecture Diagram

The client-side comprises the web interface accessible to faculty, students, and the admin. The server-side handles the backend processes, including user authentication,

project data storage, and communication between different modules. A relational database system is employed to store and retrieve project-related information efficiently.

5.3 Modules

In the overall design of the "Project Catalyst," the system is organized into several interconnected modules, each serving a specific purpose to facilitate seamless project management within an educational environment. These modules include the User Authentication module, responsible for secure user access; the Project Management module, encompassing functionalities for project idea submissions, student requests, and project allocation; the Progress Monitoring module, allowing faculty members to review student progress and provide constructive feedback; the Ideation and Innovation module, empowering students to propose new project ideas; the Faculty Management module, providing centralized oversight for faculty members; and the Admin Control module, offering overarching control to manage all system activities. The interaction and collaboration between these modules form the backbone of the system, creating an integrated platform that enhances communication, transparency, and efficiency in academic project endeavours.

5.3.1 User Authentication Module

This module employs secure authentication mechanisms, including hashed passwords and session management, to ensure that users (faculty, students, and admin) have authenticated access to the system. Role-based access control is implemented to determine the level of authorization for each user.

5.3.1.1 Two Step Verification: Two-step verification (also known as two-factor authentication, or 2FA) adds an additional layer of security to the user authentication process. Beyond just entering a username and password, users must provide a second piece of information to verify their identity. This ensures a higher level of security by requiring something the user knows (password) and something the user has (a verification code).

5.3.2 Project Management Module

5.3.2.1 Project Idea Submission: Faculty members engage in a user-friendly interface to submit detailed project ideas. The submission process includes fields such as project title, objectives, methodologies, and anticipated outcomes. These submissions are securely stored in the database.

5.3.2.2 Student Project Request: Students explore a diverse range of project ideas presented by faculty members. They meticulously select projects of interest and submit formal requests. This initiates an evaluation process by faculty members.

5.3.2.3 Project Allocation: Upon faculty approval, the system officially allocates the project to the student. This triggers the creation of a dedicated dashboard for the student, linking the project to their account.

5.3.2.4 Student Dashboard*:* Students access a personalized dashboard that serves as a central hub for project-related activities. It facilitates document uploads, progress tracking, and communication with faculty.

5.3.2.5 Document Management*:* This sub-module ensures secure and organized storage of project-related documents. It allows students to upload project proposals, research findings, and periodic progress reports, fostering collaborative feedback exchange.

5.3.3 **Progress Monitoring Module**

This module provides faculty members with tools to actively monitor and assess student progress. It includes:

5.3.3.1 Document Review: Faculty members can review and provide constructive feedback on documents submitted by students, fostering a continuous feedback loop.

5.3.3.2 Progress Tracking: This sub-module offers tools to track the overall development of each project, ensuring accountability and timely intervention if necessary.

5.3.4 Ideation and Innovation Module

This module aims to promote collaboration beyond project-specific interactions:

5.3.4.1 Idea Suggestion*:* Students are empowered to propose new project ideas directly to faculty members through the platform. This feature contributes to an environment that nurtures collective ideation and innovation.

5.3.5 Faculty Management Module

Centralized administrative functions for faculty members include:

5.3.5.1 Project Oversight: Faculty members have the capability to edit project details, monitor student progress, and provide ongoing guidance throughout the project lifecycle.

5.3.6 Admin Control Module

This module provides the admin with comprehensive control over the system, ensuring that all aspects of user and system management are effectively handled to maintain smooth operation and security.

5.3.6.1 User Management: The User Management feature allows admins to oversee and maintain the user base of the system. This includes the ability to add new users, remove existing users, and manage their roles and permissions.

5.3.6.1 Notification Management: The Notification Management feature enables admins to create, send, and manage notifications across the system. Notifications keep users informed about important updates, system changes, and other relevant information.

5.4 Data Flow

The data flow within the "Project Catalyst" follows a logical sequence, ensuring efficient and secure transmission:

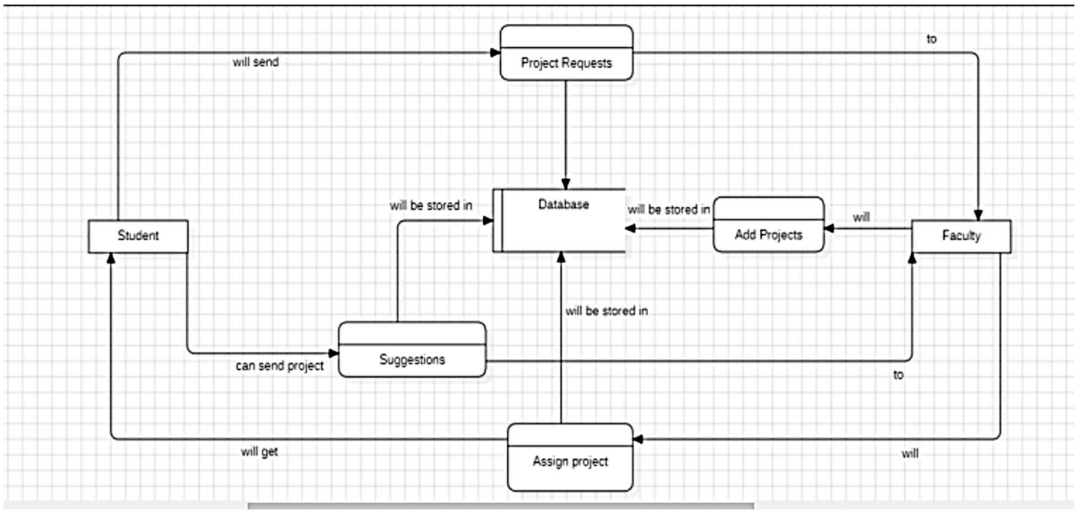

Fig 5.2 Data Flow Diagram

- User authentication initiates access to the appropriate modules based on the user's role.
- Project data flows seamlessly from the Project Management Module to the Progress Monitoring Module, enabling continuous evaluation and feedback.
- Ideation and innovation suggestions from students flow into the Faculty Management Module, fostering a collaborative environment.

5.5 User Interface

The user interface is designed with a focus on user experience, intuitiveness, and visual appeal. Faculty and students navigate through dedicated dashboards that present project-related information in a structured and accessible manner. Responsive design principles are incorporated to ensure accessibility across various devices.

5.6 Suggestion Management Module

The Suggestion Management Module is designed to handle the collection, review, and management of suggestions submitted by students. It ensures that students can provide feedback or suggestions related to projects, and faculty members can review and act upon these suggestions efficiently.

5.6 **Use Cases**

The use case diagram illustrates the various interactions and scenarios between actors (Faculty, Student, Admin) and the "Project Catalyst" system. It provides a visual representation of how each user interacts with the system, outlining essential functionalities such as logging in, submitting project ideas, managing project progress, and overseeing system activities. The use case diagram serves as a valuable tool for understanding the system's functionality from a user-centric perspective.

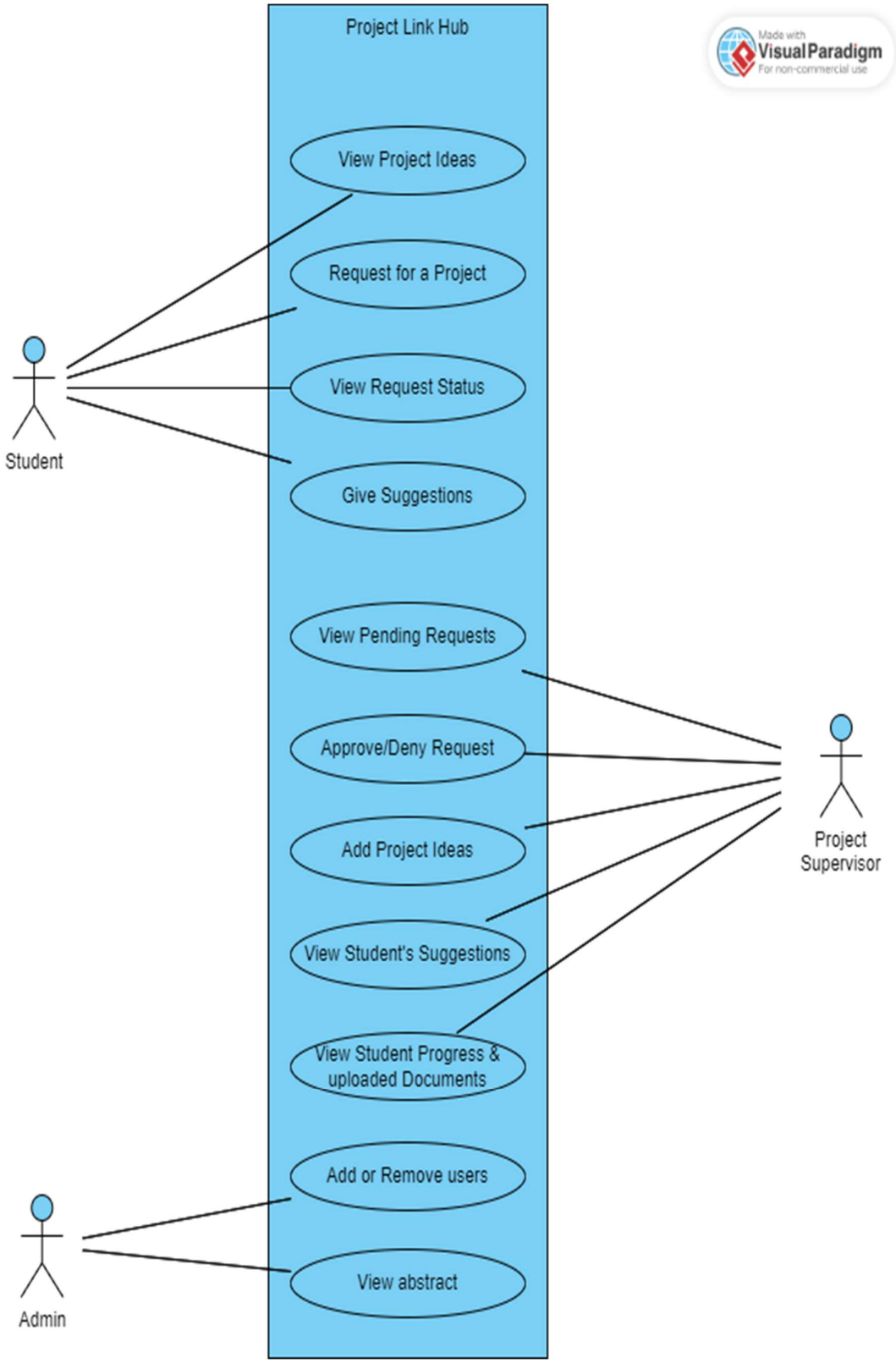

Fig 5.3 System Use Case Diagram

5.6.1 **Student Use Case:**

Logging In:

- Students log into the system using their credentials.
- Upon successful authentication, students are directed to their personalized dashboards.

View Project Ideas:

- Students explore the array of available project ideas submitted by faculty members.
- They meticulously select a project of interest.

Request for a Project:

- Students formally submit requests to work on chosen projects.
- The system notifies faculty members of project requests.

Suggest a Project:

- Students formally submit requests to work on chosen projects.
- The system notifies faculty members of project requests.

Upload Documents:

- Students contribute to project progression by uploading essential documents within their dedicated dashboards.

5.6.2 **Faculty Use Case:**

Logging In:

- Faculty members log into the system using their credentials.
- Successful authentication grants access to the faculty management dashboard.

Project Idea Submissions:

- Faculty members submit diverse project ideas, including project title, objectives, methodologies, and anticipated outcomes.

Reviewing Student Requests:

- Faculty members receive and review student requests, considering factors like qualifications and project fit.

Reviewing Student Suggestions:

- Faculty members receive and review student suggestions, that means students can come forward with their own project proposal then faculty can act on that.

Evaluating Progress:

- Faculty members actively monitor and assess student progress by reviewing uploaded documents and providing feedback.

5.6.3 **Admin Use Case:**

Logging In:

- Admin logs into the system with administrative credentials.
- Authentication provides access to the admin control dashboard.

User Management:

- Admin manages user accounts, adding or removing faculty, students, and other administrative roles.

System Oversight:

- Admin oversees the entire system's functionality, ensuring smooth operation and integrity.

Send notifications:

- Admin send notify the students either individually or collectively to make them inform something about the academics like deadlines.

Set Timer:

- Admin has the power to set the expiry date and time for specific document to be uploaded by the student e.g. Synopis, SRS, Presentation1, Presentation2, Presentation3 and Final Report.

Report Generation:

- Admin can generate different kind of reports like: Student status report, Faculty status report, past participant status report, Student suggestions report, Student requests report etc. to act accordingly.

5.7 Flow Chart

The flowchart for the "Project Catalyst" depicts a structured sequence of actions users undertake within the system. It initiates with users accessing the platform through a login interface, where they input their credentials. Subsequently, the system verifies and authenticates the provided credentials to determine the user's role—either Faculty or Students

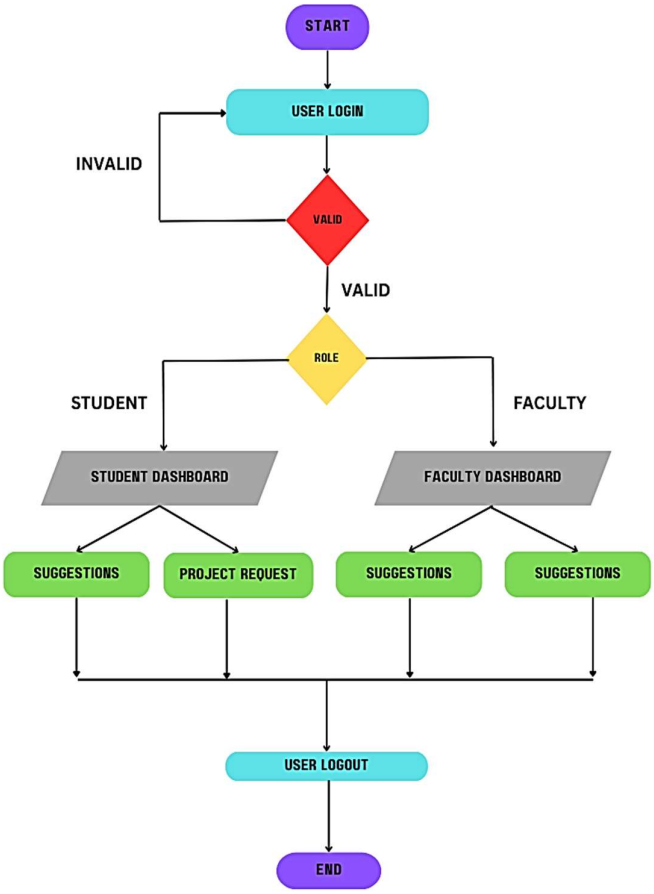

Fig 5.4 System Flow Chart

For Faculty members, the flowchart illustrates a pathway granting access to specific modules tailored to their roles and responsibilities. These modules include functionalities such as adding projects, reviewing project proposal requests submitted by students, approving or rejecting these requests, evaluating student progress, and performing administrative tasks essential for project management.

Conversely, for Students, the flowchart delineates an alternative pathway, allowing access to modules suited to their roles and needs. These modules empower students to request projects, suggest new project ideas, view available projects, and monitor their project progress, fostering active participation and engagement in academic endeavors.

Following the completion of tasks within their respective modules, users have the option to logout from the system. This action terminates the current session, ensuring security and privacy, and redirects users back to the login interface, ready for subsequent interactions with the "Project Catalyst."

In essence, the flowchart offers a visual representation of the user journey within the "Project Catalyst," elucidating the systematic process by which users navigate through the platform's functionalities based on their roles and objectives. Through this structured approach, users can efficiently engage with the system to manage projects, collaborate with peers, and contribute to academic excellence.

5.8 System Diagram

The system architecture is visually represented through a comprehensive diagram, illustrating the flow of information between modules. This visual aid enhances the understanding of the intricate relationships between modules and their interactions

Fig 5.5 System Module Relationship Diagram

Above mentioned a visual representation of the system modules, providing a comprehensive overview of their relationships and interactions.

5.9 Conclusion

The design of the "Project Catalyst" reflects a commitment to fostering collaboration, transparency, and innovation within the academic project management space. By integrating key modules and employing robust architectural principles, the system provides a cohesive and effective platform for faculty and students alike. The

use cases offer insights into the working dynamics of each actor, ensuring a user-centric and efficient project management solution. The visual representation in the form of a system diagram enhances the understanding of the intricate relationships between modules, reinforcing the commitment to a user-centric and efficient project management solution.

CHAPTER-6

IMPLEMENTATION

6.1 User Interface

6.1.1 Home Page

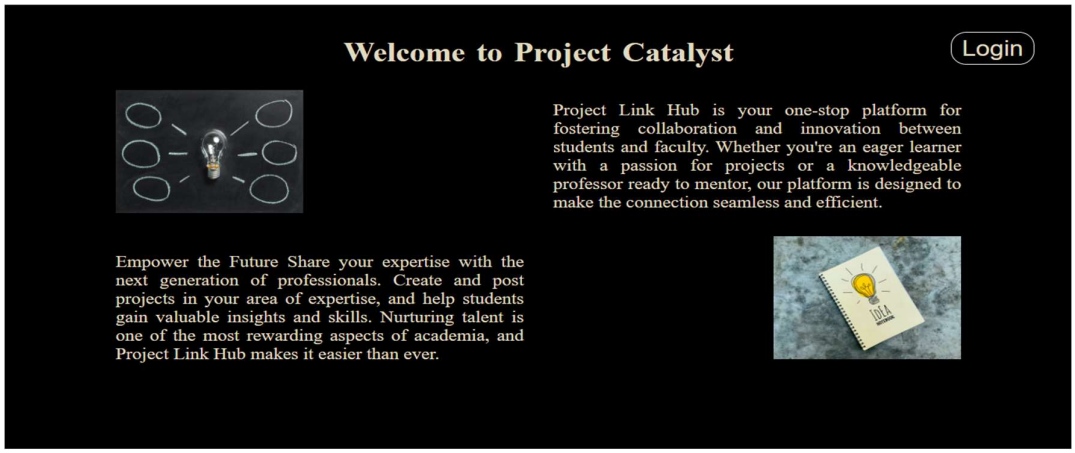

Fig 6.1 Home Page

6.1.2 Login Page

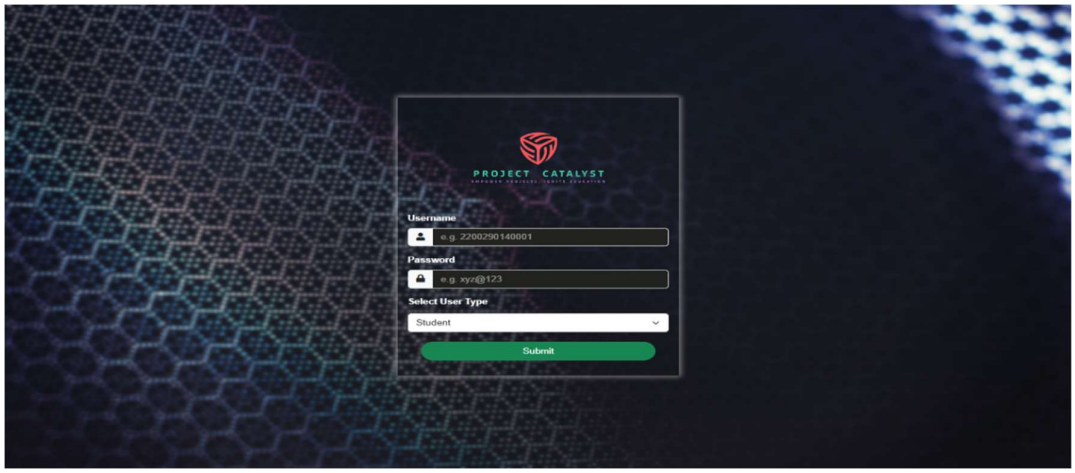

Fig 6.2 Login Page

6.1.3 Faculty Dashboard

6.1.3.1 Home Page

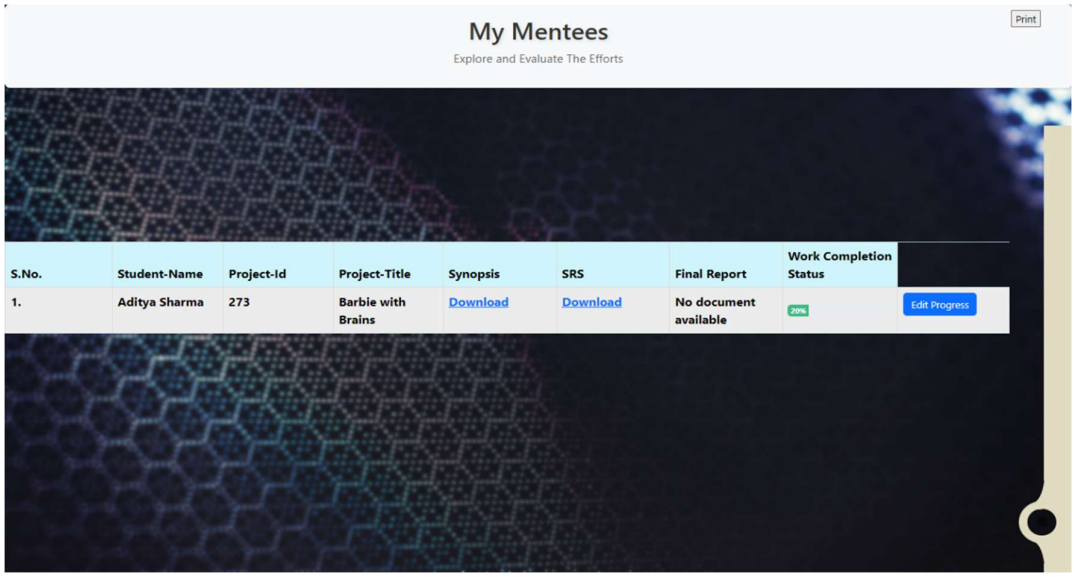

Fig 6.3 Faculty Dashboard

6.1.3.2 Suggestions

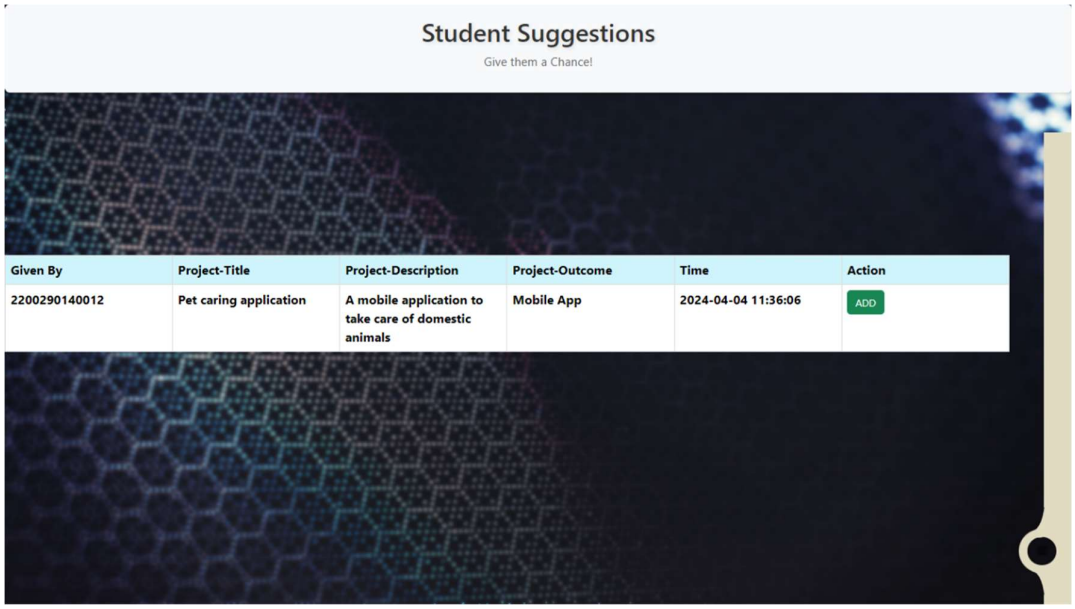

Fig 6.4 Suggestion Received

6.1.3.3 Add Project

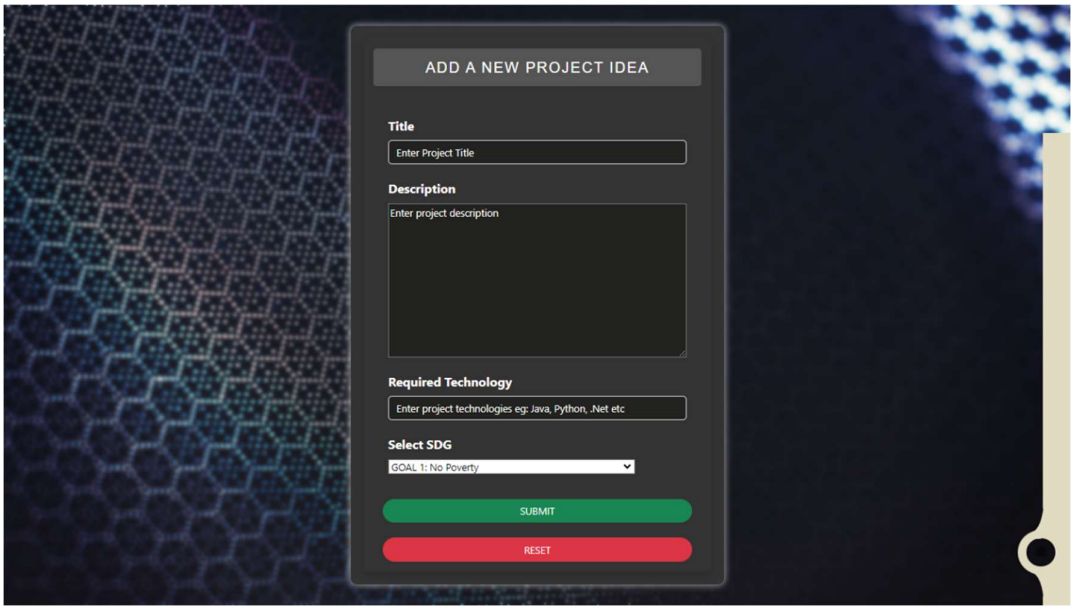

Fig 6.5 Add Project Page

6.1.3.4 Pending Requests

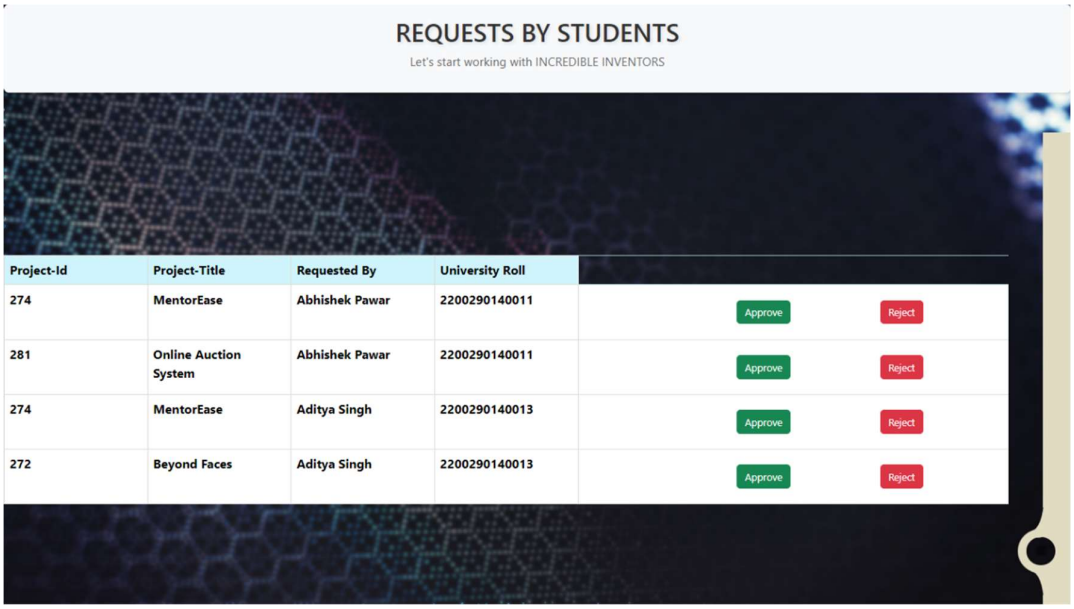

Fig 6.6 Pending Requests Page

6.1.4 Student Dashboard

6.1.4.1 Home Page

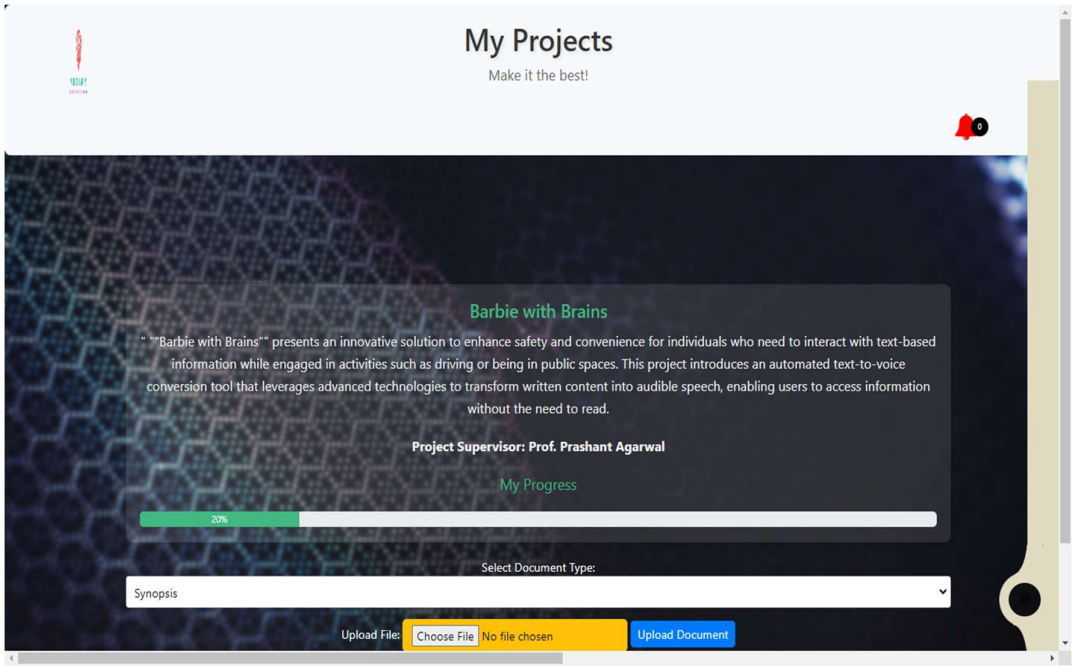

Fig 6.7 Student Dashboard

6.1.4.2 Projects

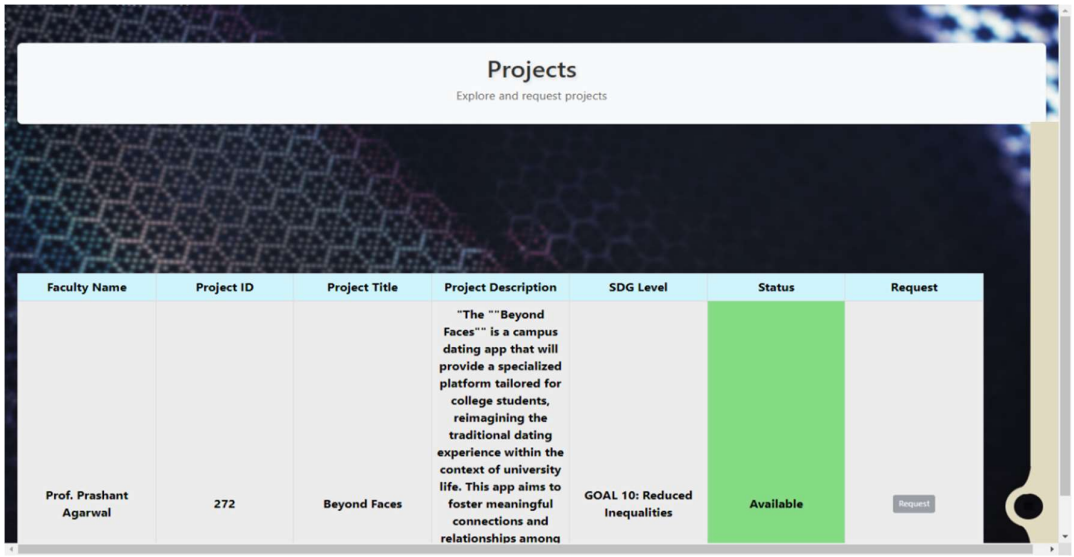

Fig 6.8 Project List Page

6.1.4.3 My Suggestions

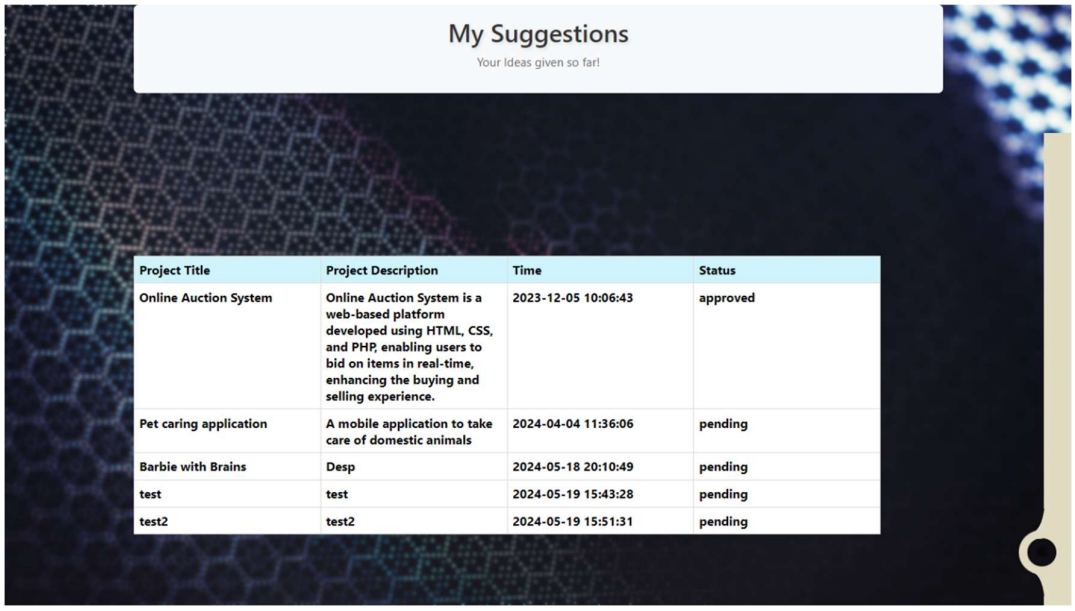

Fig 6.9 My Suggestions Page

6.1.4.4 Give Suggestions

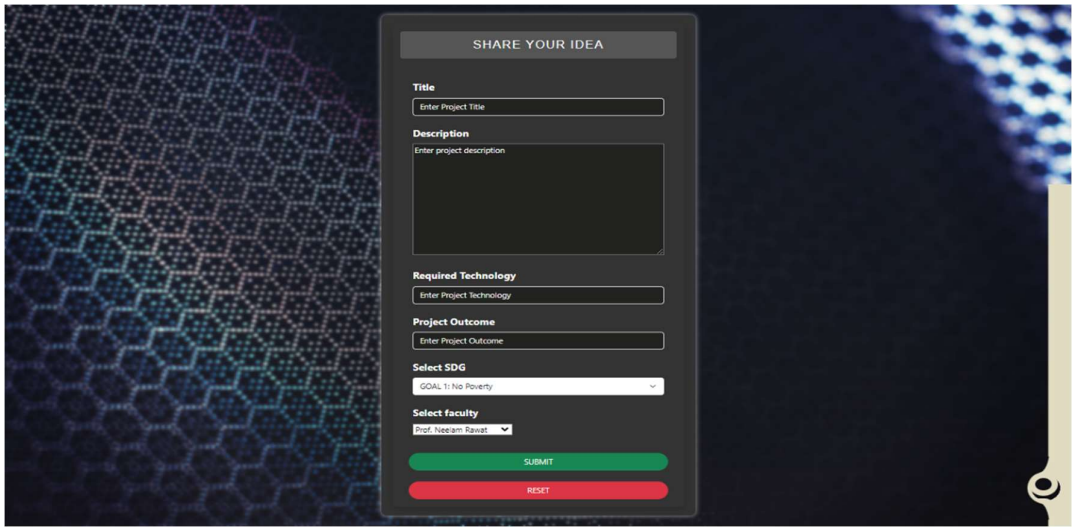

Fig 6.10 Give Suggestions Page

6.1.5 Change Password

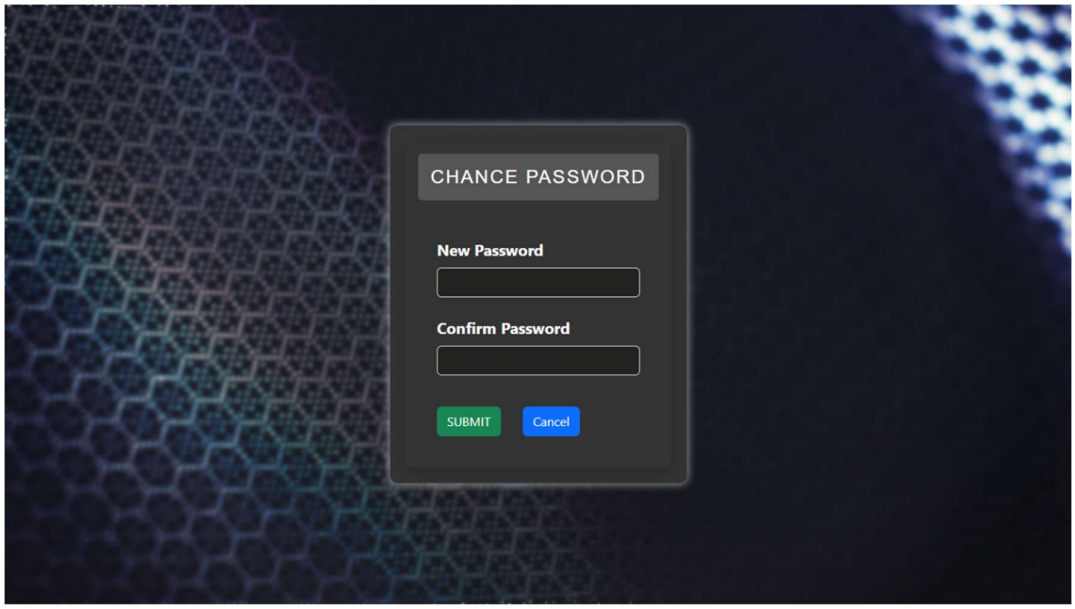

Fig 6.11 Change Password Page

6.1.6 Logout

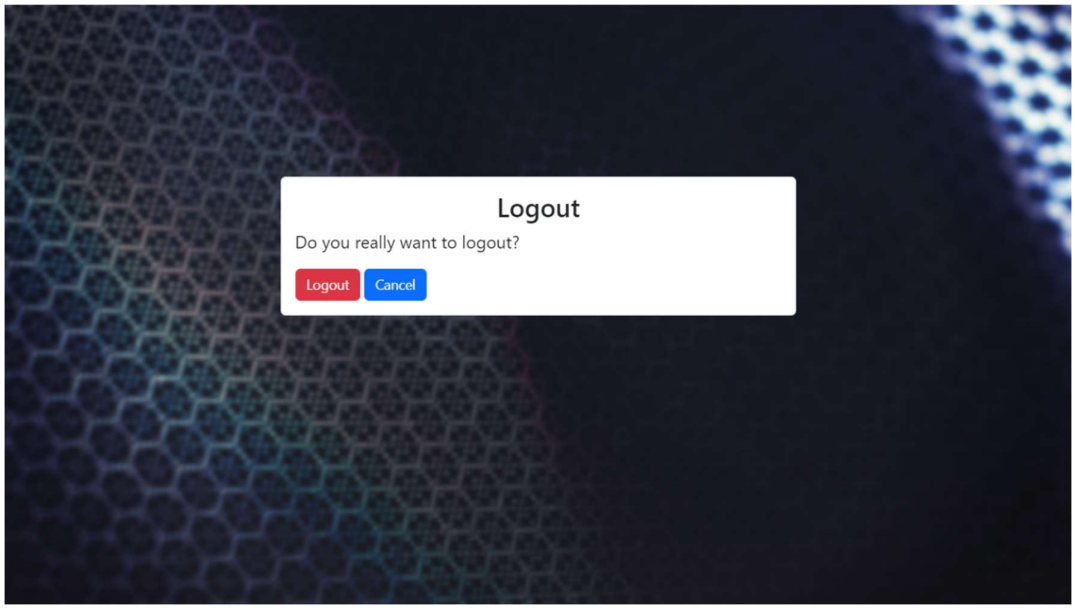

Fig 6.12 Logout Page

6.1.7 Reports

6.1.7.1 Request Report

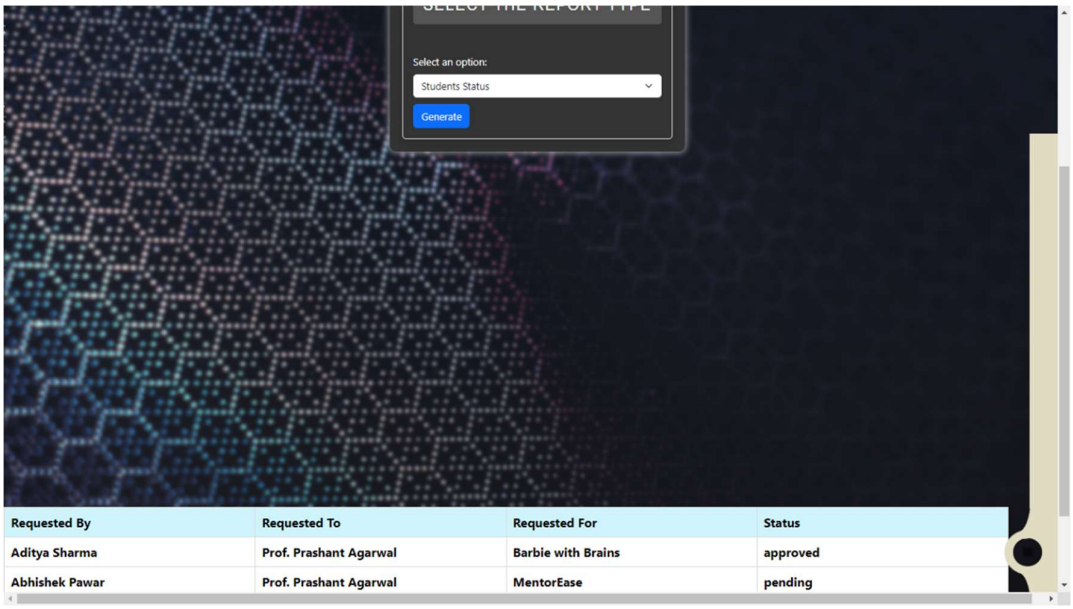

Fig 6.13 Request Status

6.1.7.2 Suggestion Report

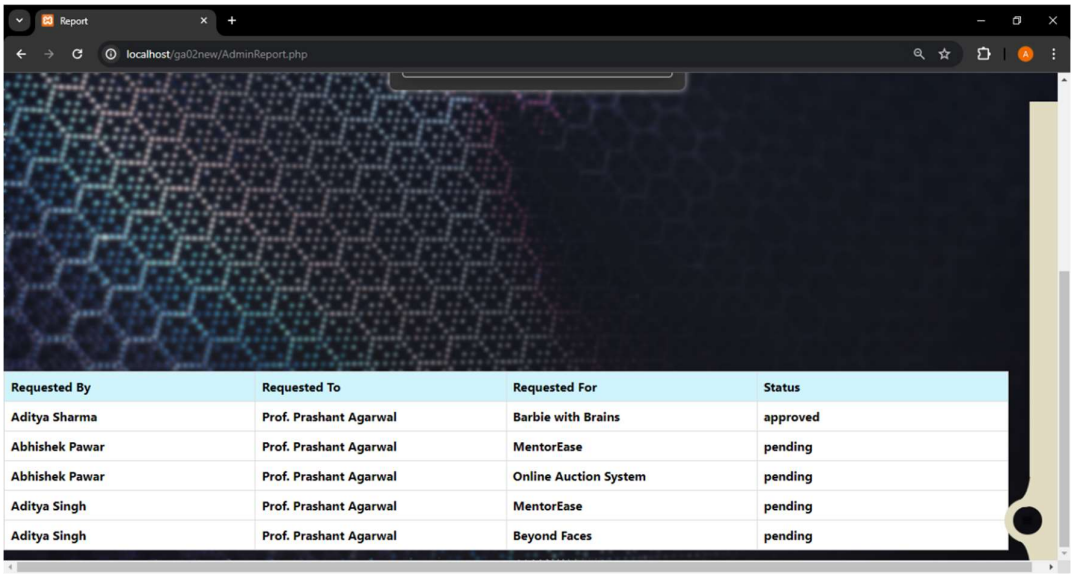

Fig 6.14 Suggestion Status

6.1.7.3 Faculty Report

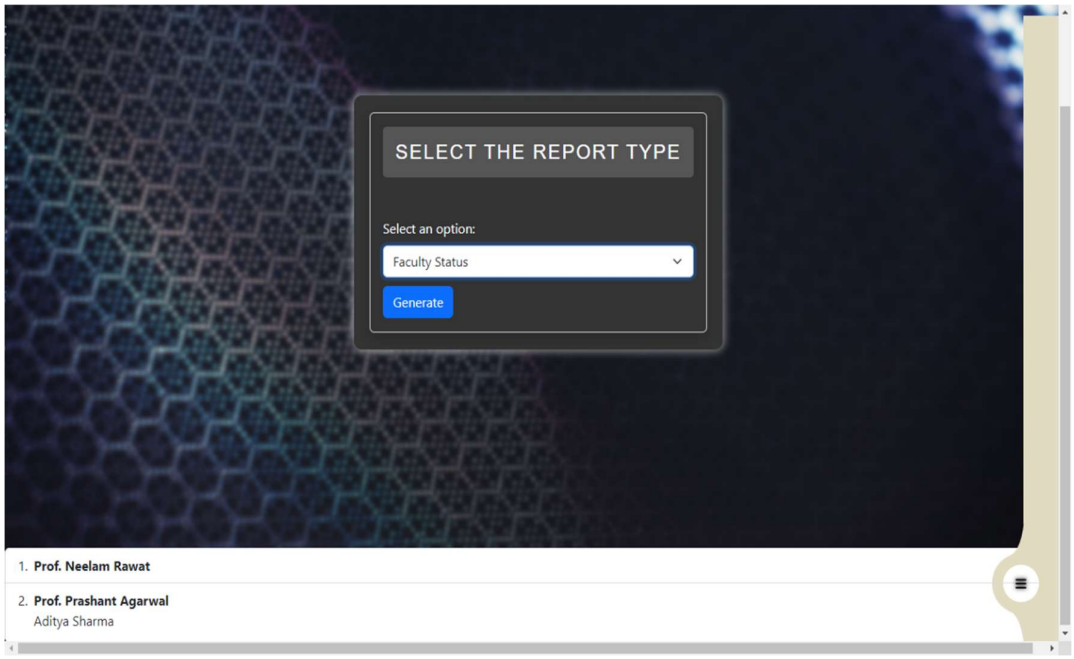

Fig 6.15 Faculty Status

6.1.8 Notifications

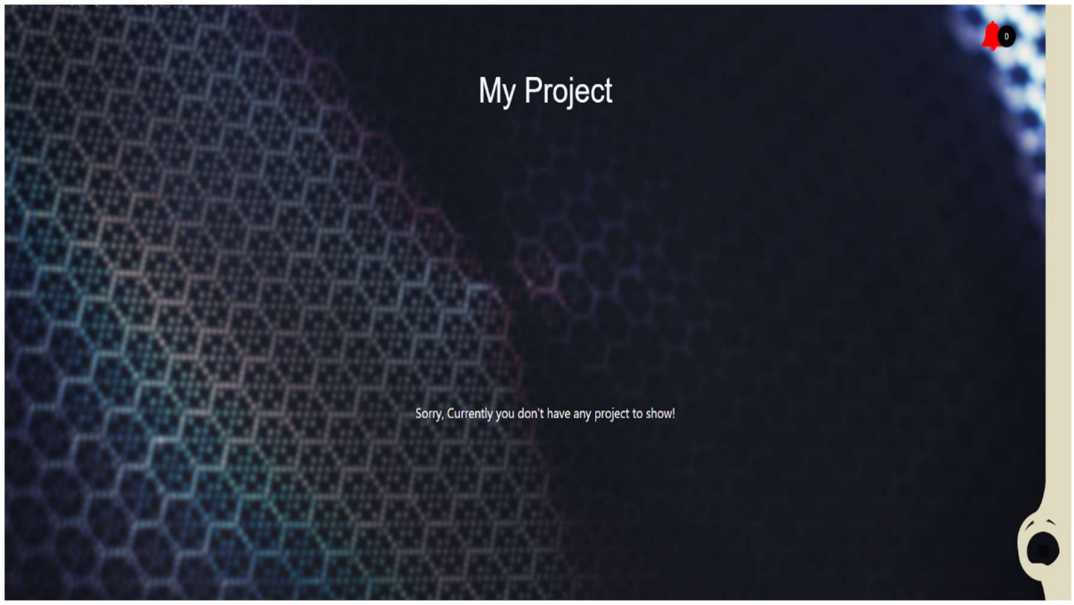

Fig 6.16 Notification Page

6.1.9 Send Notifications

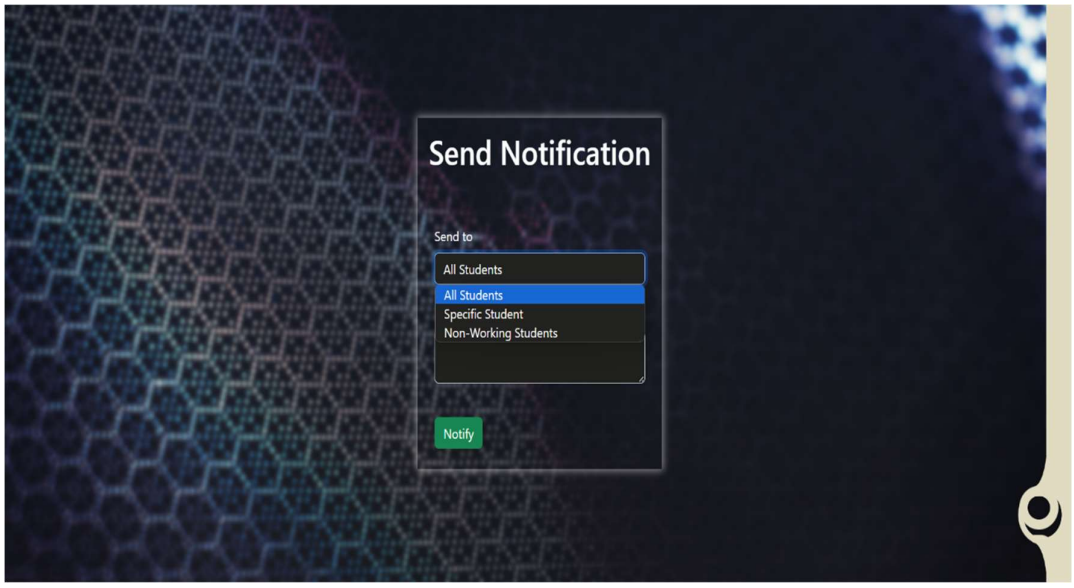

Fig 6.17 Send Notification Page

6.1.10 View Notifications

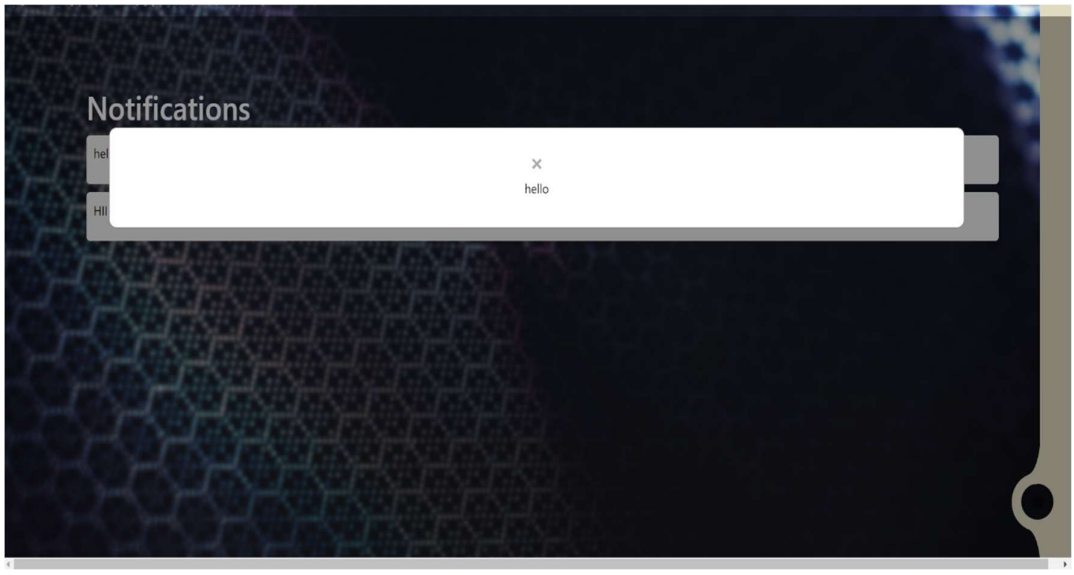

Fig 6.18 View Notification Page

CHAPTER-7

TESTING

7.1 Introduction

The testing phase of the "Project Catalyst" plays a pivotal role in ensuring the system's functionality, reliability, and overall performance. The testing process is meticulously executed, encompassing various levels, including Unit Testing, Integration Testing, and System Testing. This chapter provides an elaborate insight into how each testing level is implemented to guarantee the robustness and effectiveness of the system.

7.2 Unit Testing

Unit Testing involves isolating individual components and verifying their correctness. Each module of the "Project Catalyst" underwent rigorous unit testing to ensure its functionality in isolation. For instance, the User Authentication module was subjected to tests for accurate user verification, secure session management, and proper role-based access control. The Project Management module underwent tests for seamless project idea submission, student request processing, and accurate project allocation.

7.3 Integration Testing

Integration Testing focuses on examining the interaction between different modules to ensure they work seamlessly together. In the "Project Catalyst," integration testing involved validating the communication and data flow between modules. For instance, the integration between the Project Management and Progress Monitoring modules was tested to confirm that project data is accurately transmitted, reviewed, and updated. Similarly, integration tests ensured that the User Authentication module effectively communicates with the Database for secure user access.

7.4 System Testing

System Testing assesses the entire system's behavior to ensure it meets specified requirements. The "Project Catalyst" underwent comprehensive system testing to evaluate end-to-end functionalities. This included testing user journeys such as logging in, submitting project ideas, reviewing progress, and suggesting new project ideas.

7.5 Manual Testing

In addition to automated testing processes, extensive manual testing was conducted to simulate real-world scenarios and user interactions. Test cases were designed to cover a range of inputs, including valid and invalid data, to validate the system's resilience an error handling capability. Manual testing also included exploratory testing to uncover any unforeseen issues and ensure a positive user experience. It includes following test cases:

Testers and Responsibilities

Formally mention who conducted the testing and their specific roles.

Testers:

- **Tester Aayush Chaudhary:** Responsible for designing and executing test cases related to user management and account management modules.
- **Tester Aditya Sharma:** Responsible for designing and executing test cases related to document handling module and overall system integration testing

7.5.1. Check Login Functionality (for all four :student, faculty, coordinator and admin) there many possible test cases are:

- Test Case 1: Check results on entering valid User Id & Password
- Test Case 2-5: Check results on entering Invalid User ID & Password or otp

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/ Fail
TU01	Check User Login with valid credentials and otp	1. Go to site localhost/GA03/login.php 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = 2200290140002 Password = mca OTP=737291	User should Login into an application	As Expected	Pass
TU02	Check User Login with invalid username valid pass	1. Go to site localhost/GA03/login.php 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = 220029140002 Password = mca	User should not Login into an application	As Expected	Pass
TU03	Check User Login with valid username but invalid password	1. Go to site localhost/GA03/login.php 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = 2200290140002 Password = mcb	User should not Login into an application	As Expected	Pass
TU04	Check User Login with valid credentials but wrong otp	1. Go to site localhost/GA03/login.php 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = 2200290140002 Password = mca OTP=737291	User should not Login into an application	As Expected	Pass

Table 7.1 Check Login Functionality Test Case

7.5.2. Check project status before user press the request button:

- Test Case 1: Check results when project is available
- Test Case 2: Check results when project is unavailable

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Request when Project Status is available	1. Go to site localhost/GA03/dashboard.php 2. Navigate Projects 3. Hit the request button	Userid = 2200290140012 Password = mca	User should be able to request for the selected project	As Expected	Pass
TU02	Request when Project Status is unavailable	1. Go to site localhost/GA03/login.php 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = Aditya111 Password = kiet	User should not be able to request for the selected project	As Expected	Pass

Table 7.2 Check Project Status Test Case

7.5.3. Check Notification module on different events:

1. Test Case 1-2: Verify notification count increment/decrement when a notification arrives/read
2. Test Case 3: Verify that a new notification is displayed in the notification list
3. Test Case 4: Verify that all notifications can be cleared
4. Test Case 5: Verify that notifications persist after logging out and logging back in

Test Case ID	*Test Case Description*	*Test Steps*	*Test Data*	*Expected Results*	*Actual Results*	*Pass/Fail*
TN01	Verify notification count increment when a new notification arrives	1. Log in to the site using valid credentials. 2. Check the current notification count. 3. Trigger an action that generates a new notification 4. Check the notification count again.	Userid: 2200290140012 Password: mca	Notification count should increment by 1 after the new notification arrives.	As Expected	Pass

TN02	Verify notification count decrement when a notification is read	1. Log in to the site using valid credentials. 2. Check the current notification count. 3. Click on the notification icon to view notifications. 4. Mark a notification as read. 5. Check the notification count again.	Userid: 2200290140012 Password: mca	Notification count should decrement by 1 after the notification is marked as read.	As Expected ,	Pass
TN03	Verify that a new notification is displayed in the notification list	1. Log in to the site using valid credentials. 2. Trigger an action that generates a new notification. 3. Click on the notification icon to view notifications. 4. Verify that the new notification is listed.	Userid: 2200290140012 Password: mca	The new notification should be displayed in the notification list.	As Expected	Pass
TN04	Verify that all notifications can be cleared	1. Log in to the site using valid credentials. 2. Click on the notification icon to view notifications. 3. Click on the "Clear All" button. 4. Verify that the notification count is reset to 0. 5. Verify that the notification list is empty.	Userid: 2200290140012 Password: mca	Notification count should be reset to 0 and the notification list should be empty after clearing all notifications.	Missing	Fail
TN05	Verify that notifications persist after logging out and logging back in	1. Log in to the site using valid credentials. 2. Trigger an action that generates a new notification. 3. Log out from the site. 4. Log in again with the same credentials. 5. Click on the notification icon to view notifications. 6. Verify that the previously generated notification is still present.	Userid: 2200290140012 Password: mca	Notifications should persist after logging out and logging back in.	As Expected	Pass

Table 7.3 Check Notification Module Test Case

7.5.4. Check project fields before faculty add it to the database

- Test Case 1: Check results on entering valid project title, description, technology and SDG level
- Test Case 2-5: Check results on entering one of the fields is empty

Test Case ID	*Test Case Description*	*Test Steps*	*Test Data*	*Expected Results*	*Actual Results*	*Pass/ Fail*
TU01	Add Project while filling each field	1. Go to site localhost/GA03/AddProject.php 2. Enter the details for each field 3. Hit the submit button	ProjectTitle:QuizMaker ProjectDesc: Design Quizs ProjectTech:PHP SDG:4	User should be able to add the project	As Expected	Pass
TU02	Add Project while skipping Project Title field	1. Go to site localhost/GA03/AddProject.php 2. Enter the details 3. Hit the submit button	ProjectTitle: ProjectDesc: Design Quizs ProjectTech:PHP SDG:4	User should not be allowed to add the project	As Expected	Pass
TU03	Add Project while skipping Project Desc. field	1. Go to site localhost/GA03/AddProject.php 2. Enter the details 3. Hit the submit button	ProjectTitle: QuizMaker ProjectDesc: ProjectTech:PHP SDG:4	User should not be allowed to add the project	As Expected	Pass
TU04	Add Project while skipping Project Tech. field	1. Go to site localhost/GA03/AddProject.php 2. Enter the details 3. Hit the submit button	ProjectTitle: QuizMaker ProjectDesc: Design Quizs ProjectTech: SDG:4	User should not be allowed to add the project	As Expected	Pass
TU05	Add Project while skipping SDG field	1. Go to site localhost/GA03/AddProject.php 2. Enter the details 3. Hit the submit button	ProjectTitle: QuizMaker ProjectDesc: Design Quizs ProjectTech:PHP SDG:	User should not be allowed to add the project	As Excepted	Pass

Table 7.4 Check Project Fields Test Case

7.5.5. Check project fields at suggestion page before student add it to the database.

- Test Case 1: Check results on entering valid project title, description, technology and SDG level
- Test Case 2-4: Check results on entering one of the fields is empty

Test Case ID	*Test Case Description*	*Test Steps*	*Test Data*	*Expected Results*	*Actual Results*	*Pass/Fail*
TU01	Add Project while filling each field	1. Go to site localhost/GA03/ suggest.php 2. Enter the details for each field 3. Hit the submit button	ProjectTitle:QuizMaker ProjectDesc: Design Quizs ProjectTech:PHP SDG:4 Faculty:Prof. Neelam Rawat Outcome:Mobile App	User should be able to add the suggestion	As Expected,	Pass
TU02	Add Project while skipping ProjectTitle field	1. Go to site localhost/GA03/ suggest.php 2. Enter the details for each field 3. Hit the submit button	ProjectTitle: ProjectDesc: Design Quizs ProjectTech:PHP SDG:4 Faculty:Prof. Neelam Rawat Outcome:Mobile App	User should not be allowed to add the suggestion	As Expected	Pass
TU03	Add Project while skipping Project Desc. field	1. Go to site localhost/GA03/ suggest.php 2. Enter the details 3. Hit the submit button	ProjectTitle: QuizMaker ProjectDesc: ProjectTech:PHP SDG:4 Faculty:Prof. Neelam Rawat Outcome:Mobile App	User should not be allowed to add the suggestion	As Expected	Pass
TU04	Add Project while skipping Project Tech. field	1. Go to site localhost/GA03/ suggest.php 2. Enter the details 3. Hit the submit button	ProjectTitle: QuizMaker ProjectDesc: Design Quizs ProjectTech: SDG:4 Faculty:Prof. Neelam Rawat Outcome:Mobile App	User should not be allowed to add the project suggestion	As Expected	Pass

Table 7.5 Check Suggestion Module Test Case

7.5.6. Testing Progress Bar Functionality with Document Uploads

- Test Case 1: Checking if the progress bar correctly reflects an initial 5% progress upon logging
- Test Case 2: Verifying that uploading a synopsis document advances the progress bar to 15%.
- Test Case 3: Ensuring that uploading an SRS document increments the progress bar to 25%.
- Test Case 4: Confirming that uploading Presentation 1 increases the progress bar to 40%.
- Test Case 5: Validating that uploading Presentation 2 progresses the bar to 55%.
- Test Case 6: Checking if uploading Presentation 3 advances the progress bar to 70%.
- Test Case 7: Verifying that uploading the Final Report completes the progress bar to 100%.

Test Case ID	**Test Case Description**	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Pass /Fail**
TD01	Verify initial progress is 5%	1. Log in to the fresh student dashboard using valid credentials. 2. Check the initial progress bar value.	Userid: 2200290140012 Password: mca	Initial progress bar should be at 5%.	As Expected	Pass
TD02	Verify progress bar after uploading synopsis	1. Log in to the student dashboard using valid credentials. 2. Upload the synopsis document. 3. Check the progress bar value.	Userid: 2200290140012 Password: mca	Progress bar should change to 15% after uploading the synopsis.	As Expected	Pass
TD03	Verify progress bar after uploading SRS	1. Log in to the student dashboard using valid credentials. 2. Upload the SRS document. 3. Check the progress bar value.	Userid: 2200290140012 Password: mca	Progress bar should change to 25% after uploading the SRS.	As Expected	Pass
TD04	Verify progress bar after uploading Presentation 1	1. Log in to the student dashboard using valid credentials. 2. Upload Presentation 1. 3. Check the progress bar value.	Userid: 2200290140012 Password: mca	Progress bar should change to 40% after uploading Presentation 1.	As Expected	Pass
TD05	Verify progress bar after uploading Presentation 2	1. Log in to the student dashboard using valid credentials. 2.Upload Presentation-2 3. Check the progress bar value.	Userid: 2200290140012 Password: mca	Progress bar should change to 55% after uploading Presentation 2.	As Expected	Pass
TD06	Verify progress bar after uploading Presentation 3	1. Log in to the student dashboard using valid credentials. 2. Upload Presentation-3 3. Check the progress bar value.	Userid: 2200290140012 Password: mca	Progress bar should change to 70% after uploading Presentation 3.	Ordering mismatch	Fail

Table 7.6 Check Progress Bar Test Case

7.5.7. Testing Report Generation Functionality(AdminReport.php and CoordinatorReport,php)

- **Test Case 1:** Verify the generation of the Student Status Report by the admin.
- **Test Case 2:** Validate the generation of the Faculty Status Report by the admin.
- **Test Case 3:** Ensure the generation of the Suggestion Report by the admin.
- **Test Case 4:** Confirm the generation of the Request Report by the admin.
- **Test Case 5:** Verify the generation of the Master Report by the admin.

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/ Fail
TR01	Verify generation of Student Status Report by the admin	1.Log in to the admin panel using valid credentials. 2.Navigate to the report generation section. 3.Select "Student Status Report". 4. Generate the report. 5. Check the generated report.	Admin credentials	The Student Status Report should be generated accurately.	As Expect ed	Pass
TR02	Validate generation of Faculty Status Report by the admin	1. Log in to the admin panel using valid credentials. 2. Navigate to the report generation section. 3. Select "Faculty Status Report". 4. Generate the report. 5. Check the generated report.	Admin credentials	The Faculty Status Report should be generated accurately.	As Expect ed	Pass
TR03	Ensure generation of Suggestion Report by the admin	1. Log in to the admin panel using valid credentials. 2. Navigate to the report generation section. 3. Select "Suggestion Report". 4. Generate the report. 5. Check the generated report.	Admin credentials	The Suggestion Report should be generated accurately.	As Expect ed	Pass
TR04	Verify generation of Master Report by the admin	1. Log in to the admin panel using valid credentials. 2. Navigate to the report generation section. 3. Select "Master Report". 4. Generate the report. 5. Check the generated report.	Admin credentials	The Master Report should be generated accurately.	As Expect ed	Pass

Table 7.7 Check Reports Generation Module Test Case

7.5.8. Testing Add New User Module

- **Test Case 1:** Verify adding a new user with valid details.
- **Test Case 2:** Verify adding a user with missing details.

Test Case ID	*Test Case Description*	*Test Steps*	*Test Data*	*Expected Results*	*Actual Results*	*Pass/ Fail*
AU01	Verify adding a new user with valid details	1. Log in to the admin panel using valid credentials. 2. Navigate to the "Add New User" section. 3. Enter valid user details. 4. Submit the form.	User details: username, password, email, role	New user should be added successfully.	As Expected	Pass
AU02	Verify adding a user with missing details	1. Log in to the admin panel using valid credentials. 2. Navigate to the "Add New User" section. 3. Leave some required fields empty. 4. Submit the form.	Incomplete user details	Error message should be displayed.	As Expected	Pass
AU03	Verify adding a user with existing username	1. Log in to the admin panel using valid credentials. 2. Navigate to the "Add New User" section. 3. Enter a username that already exists. 4. Submit the form.	Existing username	Error message indicating username already exists.	As Expected	Pass

Table 7.8 Check Add Users Module Test Case

7.5.9. Testing Remove Existing User Module

- **Test Case 1:** Verify removing an existing user

Test Case ID	*Test Case Description*	*Test Steps*	*Test Data*	*Expected Results*	*Actual Results*	*Pass/Fail*
RU01	Verify removing an existing user	1. Log in to the admin panel using valid credentials 2. Navigate to the "Remove User" section. 3. Select an existing user. 4. Confirm the removal.	Existing user details	User should be removed successfully.	As Expected	PAss

Table 7.9 Check Remove Users Module Test Case

7.5.10. Testing Set Expiry Time Module for Documents

- **Test Case 1:** Verify setting expiry time for documents
- **Test Case 2:** Verify document upload is restricted after expiry time
- **Test Case 3:** Verify document upload is allowed before expiry time
- **Test Case 4:** Verify error message when uploading a document after the expiry time

Test Case ID	***Test Case Description***	***Test Steps***	***Test Data***	***Expected Results***	***Actual Results***	***Pass/ Fail***
ET01	Verify setting expiry time for documents	1. Log in to the admin panel using valid credentials. 2. Navigate to the "Set Timer" section. 3. Set an expiry time for any document (Synopsis, SRS, Presentations, Final Report). 4. Save the settings.	Admin credentials Expiry time for documents	Expiry time should be set successfully for the specified document.	As Expected	Pass
ET02	Verify document upload is restricted after expiry time	1. Log in to the student dashboard using valid credentials. 2. Attempt to upload a document (Synopsis, SRS, Presentations, Final Report) after the expiry time has passed.	Student credentials Document to upload	The student should not be able to upload the document after the expiry time.	As Expected	Pass
ET03	Verify document upload is allowed before expiry time	1. Log in to the student dashboard using valid credentials. 2. Attempt to upload a document (Synopsis, SRS, Presentations, Final Report) before the expiry time.	Student credentials Document to upload	The student should be able to upload the document before the expiry time.	As Expected	Pass
ET04	Verify error message when uploading a document after the expiry time	1. Log in to the student dashboard using valid credentials. 2. Attempt to upload a document (Synopsis, SRS, Presentations, Final Report) after the expiry time has passed.	Student credentials Document to upload	An appropriate error message should be displayed indicating the document cannot be uploaded.	As Expected	Pass

Table 7.10 Check Set Expiry Time Setter Module Test Case

7.6 User Acceptance Testing (UAT)

User Acceptance Testing involved collaboration with end-users, including students, faculty, and administrators, to validate that the system meets their needs and expectations. This phase allowed stakeholders to provide feedback on the user interface, functionality, and overall usability of the "Project Catalyst."

7.7 Conclusion

The testing chapter demonstrates a comprehensive approach to ensuring the quality and reliability of the "Project Catalyst." Through unit testing, integration testing, system testing, manual testing, and user acceptance testing, the system's various facets were thoroughly examined. This meticulous testing process contributes to the confidence in the system's performance, functionality, and user satisfaction, making the "Project Catalyst" a robust and dependable platform for academic project management.

CHAPTER-8

PERFORMANCE ANALYSIS

8.1. User-Friendly Interface:

- **Objective:** Develop an intuitive and user-friendly interface.
- **Metrics:** Evaluate the responsiveness of the interface using tools like Google Lighthouse or browser developer tools.
- **Optimization Actions:** Optimize client-side code, use asynchronous loading for resources, and ensure efficient CSS and JavaScript practices.

8.2. Collaboration Enhancement:

- **Objective:** Promote effective collaboration among students and faculty members.
- **Metrics:** Measure the time taken for collaborative features, such as sharing and reviewing project ideas.
- **Optimization Actions:** Optimize server-side code to handle concurrent collaboration efficiently, minimizing delays in communication.

8.3. Streamlined Approval Processes:

- **Objective:** Facilitate a streamlined process for faculty members to review, approve, or reject student project proposals.
- **Metrics:** Monitor the time taken for faculty members to review and provide feedback.
- **Optimization Actions:** Optimize server-side logic to ensure timely approval/rejection and constructive feedback.

8.4. Centralized Project Management:

- **Objective:** Establish a centralized system for managing project ideas.
- **Metrics:** Evaluate the response time for project management actions, such as adding, editing, or deleting projects.
- **Optimization Actions:** Optimize database queries and ensure efficient project management functionalities.

8.5. Robust Security Measures:

- **Objective:** Implement robust security measures to safeguard user data.
- **Metrics:** Assess the impact of security measures on system performance.
- **Optimization Actions:** Regularly update security protocols and ensure that security measures do not introduce noticeable latency.

8.6. Performance Optimization:

- **Objective:** Optimize system performance to handle a substantial number of concurrent users.
- **Metrics:** Conduct load testing to identify performance bottlenecks.
- **Optimization Actions:** Optimize code, use caching mechanisms, and consider scaling options (vertical or horizontal) for handling concurrent users.

8.7. Compliance with Development Technologies:

- **Objective:** Adhere to specified development technologies for consistency.
- **Metrics:** Confirm that HTML, CSS, PHP, Bootstrap, and MySQL are well-integrated.
- **Optimization Actions:** Ensure smooth interactions and minimize compatibility issues by following the specified technology stack.

8.8. Adherence to Scope:

- **Objective:** Define and adhere to the project scope to maintain focused development efforts.
- **Metrics:** Prioritize in-scope functionalities related to user registration, project proposal, and approval.
- **Optimization Actions:** Prioritize optimization efforts based on in-scope functionalities, considering future enhancements for out-of-scope elements.

8.9. Overall Assessment:

Conclusion: The performance analysis highlights specific areas for optimization, including user interface responsiveness, collaboration features, approval processes, project management efficiency, and security measures. Continuous monitoring and targeted optimizations during development are crucial to ensuring a seamless user experience for the "Project Catalyst" platform.

CHAPTER-9

CONCLUSION & FUTURE ENHANCEMENT

9.1 Conclusion

The "Project Catalyst" stands as a transformative platform that revolutionizes project management within the educational realm. Throughout its development and deployment, the project has successfully achieved its objectives of enhancing collaboration, streamlining approval processes, and providing centralized project management. The journey of creating the "Project Catalyst" has been marked by dedication, innovation, and collaboration among the development team, stakeholders, and end-users.

As we conclude this phase of the project, it's essential to reflect on the achievements and acknowledge the challenges overcome. The "Project Catalyst" has not only streamlined project ideation and management but has also fostered a culture of innovation and engagement within the academic community. By providing a user-friendly interface and robust features, the platform has empowered students and faculty members alike to contribute meaningfully to academic projects.

Moving forward, it's crucial to recognize that the journey does not end here. Continuous improvement and adaptation are key to ensuring the long-term success and relevance of the "Project Catalyst." Future iterations of the platform will focus on addressing user feedback, incorporating emerging technologies, and expanding functionalities to meet evolving needs.

9.2 Future Enhancements

9.2.1. Advanced Collaboration Features: Incorporate features such as real-time collaboration tools, discussion forums, and messaging systems to facilitate seamless communication and collaboration among users.

9.2.2. Enhanced Data Analytics: Integrate data analytics capabilities to provide insights into project trends, user engagement, and system usage, enabling informed decision-making and strategic planning.

9.2.3. Integration with Learning Management Systems (LMS): Explore integration with existing LMS platforms to streamline project management processes and enhance user experience by leveraging existing educational infrastructure.

9.2.4. Mobile Application Development: Develop mobile applications for iOS and Android platforms to provide users with on-the-go access to the "Project Catalyst," enhancing convenience and accessibility.

9.2.5. Gamification Elements: Implement gamification elements such as badges, leaderboards, and rewards to incentivize active participation and engagement among users, fostering a dynamic and motivating environment.

9.2.6. AI-powered Recommendation System: Introduce an AI-powered recommendation system to suggest project ideas based on user preferences, interests, and past engagements, facilitating personalized project discovery.

9.2.7. Expanded User Roles and Permissions: Enhance role-based access control to accommodate additional user roles and permissions, catering to diverse user groups such as mentors, evaluators, and external collaborators.

9.2.8. Accessibility and Inclusivity: Ensure the platform adheres to accessibility standards and guidelines, making it accessible to users with disabilities and diverse learning needs, thus fostering inclusivity and diversity.

9.2.9. Enhanced Security Measures: Continuously update and enhance security measures to safeguard user data, prevent unauthorized access, and mitigate potential security threats, ensuring the confidentiality and integrity of information.

9.2.10. Community Engagement Initiatives: Launch community engagement initiatives such as webinars, workshops, and user forums to foster a vibrant community of practice, encouraging knowledge sharing, collaboration, and continuous improvement.

By incorporating these future enhancements, the "Project Catalyst" will continue to evolve as a dynamic, user-centric platform that empowers the academic community to thrive in the digital age. Through ongoing collaboration, innovation, and adaptation, the platform will remain at the forefront of transforming project management practices within educational institutions, driving positive outcomes for students, faculty, and administrators alike.

Bibliography

Online Platforms

During the development of the "Project Catalyst," various online platforms were analysed for inspiration and reference. These include:

- Moodle: An open-source learning management system widely used for creating and delivering online courses.
- GitHub: A popular platform for hosting and collaborating on software development projects using version control.
- Canvas: A learning management system designed to support teaching and learning in various educational settings.
- Google Classroom: A digital learning platform developed by Google for schools and educational institutions to manage assignments and communication.
- Blackboard: A leading learning management system used by educational institutions worldwide for course management and online learning.

YouTube Channels

The following YouTube channels were referenced for educational content related to programming and computer science:

- Jenny's Lectures CS IT: A YouTube channel offering lectures and tutorials on computer science and information technology topics.
- Gaurav Sen: A popular Indian programming teacher known for his informative videos on software engineering concepts and interview preparation.
- Apna College: A YouTube channel providing tutorials on programming languages and web development in Hindi.
- CodeWithHarry: A Hindi-language YouTube channel offering tutorials on programming languages like Python, Java, and C++.

Books

The development of the "Project Catalyst" drew insights from various reference books related to web development, collaboration, and project management. These include:

- "Web Development with HTML, CSS, and JavaScript" by Jon Duckett
- "PHP and MySQL Web Development" by Luke Welling and Laura Thomson

- "Agile Project Management with Scrum" by Ken Schwaber and Mike Beedle
- "Collaborative Learning Techniques: A Handbook for College Faculty" by Elizabeth F. Barkley
- "Effective Project Management: Traditional, Agile, Extreme" by Robert K. Wysocki
- "Version Control with Git" by Jon Loeliger and Matthew McCullough

References

1. Duckett, J. "Web Development with HTML, CSS, and JavaScript."
2. Welling, L., & Thomson, L. "PHP and MySQL Web Development."
3. Schwaber, K., & Beedle, M. "Agile Project Management with Scrum."
4. Barkley, E. F. "Collaborative Learning Techniques: A Handbook for College Faculty."
5. Wysocki, R. K. "Effective Project Management: Traditional, Agile, Extreme."
6. Loeliger, J., & McCullough, M. "Version Control with Git."