

Community Platform for JU Researchers

submitted by
Shrijon Kumar Biswas
Class Roll No: 002210503020
Exam Roll: MCA244019
Master of Computer Application

under the supervision of
Prof. (Dr.) Diganta Saha

in the
Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University

2024

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF RECOMMENDATION

To Whom It May Concern

I hereby recommend that the project report entitled “Community Platform for JU Researchers” has been carried out by Shrijon Kumar Biswas (Reg. No.: 163612 of 2022-23, Roll No.: 002210503020, Exam Roll: MCA244019), under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic year 2023-2024.

Prof. (Dr.) Diganta Saha
Project Supervisor
Dept. of Computer Science and
Engineering
Jadavpur University

Prof. (Dr.) Nandini Mukhopadhyay
Head of the Department
Dept. of Computer Science &
Engineering
Jadavpur University

Prof. Dipak Laha
Dean, Faculty Council
of Engineering & Technology
Jadavpur University, Kolkata-032

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the project report entitled “**Community Platform for JU Researchers**” is a bonafide record of work carried out by SHRIJON KUMAR BISWAS, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted

Signature of Examiner 1

Date :

Signature of Examiner 2

Date :

Faculty of Engineering and Technology
Jadavpur University

DECLARATION OF COMPLIANCE WITH ACADEMIC ETHICS

I hereby declare that this project report entitled "Community Platform for JU Researchers" is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, all materials and results that are not original to this work have been fully cited and referenced.

Shrijon Kumar Biswas

Registration No.: 163612 of 2022-23

Examination Roll No.: MCA244019

Thesis Title :- Community Platform for JU Researchers

Signature with Date

ACKNOWLEDGEMENT

With my most sincere gratitude, I would like to thank Prof. (Dr.) Diganta Saha, Department of Computer Science & Engineering, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestions and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providing a sound knowledge base and cooperation.

I would like to thank all the faculty members of the Department of Computer Science & Engineering of Jadavpur University for their continuous support.

Last, but not the least, I would like to thank my batch mates for staying by my side when I needed them.

Shrijon Kumar Biswas
Class Roll No.: 002210503020
Examination Roll No.: MCA244019
Department of Computer Science & Engineering
Jadavpur University

Contents

Chapter no.	Topic	Page no.
	Abstract	7
Chapter 1	Introduction	8 - 12
	1.1 Problem Analysis	
	1.2 Objective	
	1.3 Challenges	
	1.4 Contribution	
	1.5 Novelty	
Chapter 2	Software Requirements Specification	14 - 18
	2.1 Introduction	
	2.2 Overall Description	
	2.3 System Features	
	2.4 External Interface Requirements	
	2.5 Non Functional Requirements	
	2.6 Other Requirements	
Chapter 3	Proposed Methodology	19 - 48
	3.1 Data Flow Diagram	
	3.2 Use Case Diagram	
	3.3 ER Diagram	

	3.4 Basic Modules	
	3.5 Description of Platform Modules	
	3.6 Technologies Used	
Chapter 4	Security and Testing -----	49 - 55
	4.1 Vulnerability Scan	
	4.2 Testing	
	4.3 Performance Analysis	
Chapter 5	System Analysis -----	56 - 58
	5.1 Advantages	
	5.2 Drawbacks	
Chapter 6	Conclusion & Future Work -----	59 - 60
	6.1 Conclusion	
	6.2 Future Work	
	References -----	61

Abstract

In academia, collaboration has emerged as a cornerstone of progress, allowing researchers to pool their expertise and resources for collective advancement. In response to this growing need, **JU Community** was created as a digital hub to facilitate seamless collaboration among researchers across diverse fields.

At its core, JU Community employs a project group system, enabling administrators to assign members to specific research projects. Within these groups, members can share updates on their work, fostering communication and coordination. Additionally, researchers can personalize their profiles with details like name, email, SCIID, GitHub, and profile pictures, enhancing networking opportunities and community engagement.

JU Community also serves as a repository for resources vital to research, such as datasets and code snippets, which administrators can organize within project groups for easy access by members.

Through this report, we will explore the functionality and impact of JU Community, shedding light on its role in facilitating collaboration and resource-sharing within the research community. By examining its features and gathering user feedback, this study aims to provide insights for further improvement and optimization.

In essence, JU Community offers a platform where researchers can come together, collaborate, and access resources, fostering a culture of collective learning and advancement in academia.

Chapter 1

Introduction

1.1 Problem Analysis

Despite its robust research infrastructure, JU faces several challenges that hinder the full potential of its research community:

1. Lack of a Communication and Collaboration Platform: Currently, there is no dedicated platform for researchers to communicate and collaborate effectively. This limits the exchange of ideas and coordination among researchers from different departments and disciplines.

2. No Support for Showcasing Research Works: Researchers at JU lack a centralized space to display their ongoing projects and completed works. This results in limited visibility and recognition of their contributions within and outside the university.

3. No Way of Creating Research Profiles: There is no unified system for researchers to create and maintain profiles that highlight their expertise, publications, and achievements. This makes it difficult for researchers to network and collaborate with peers.

4. No Centralized Storage of Resources: The absence of a centralized repository for datasets, codes, and other research materials hinders efficient resource sharing and management. Researchers often struggle to access the necessary resources for their projects.

1.2 Objective

To address these issues, the introduction of the **JU Community Platform** is proposed. This platform will serve as a centralized hub for researchers at Jadavpur University, providing several essential features:

1. Centralized Collaboration Platform: The platform will facilitate seamless communication and collaboration among researchers through project groups. Members can post updates, share progress, and discuss findings, fostering a more integrated research environment.

2. Profile Creation and Management: Researchers can set up detailed profiles that include their name, email, SCiID, GitHub, and other relevant information, along with a profile picture. This will help showcase their work and connect with peers more effectively.

3. Resource Sharing: Admins can upload resources like datasets and codes via Google Drive links, making them easily accessible to group members. This centralized storage will streamline resource management and utilization.

4. Admin and User Notifications: Automated email notifications will keep users informed about registration approvals or denials and notify admins about new registration requests. This ensures smooth administrative operations and keeps all researchers informed.

5. SSO and Cookies for Login: Secure Single Sign-On (SSO) and cookie-based login mechanisms will ensure user authentication and a seamless user experience. This will enhance security and convenience for users accessing the platform.

1.3 Challenges

1. Security Vulnerabilities

- Cross-Site Request Forgery (CSRF) and XSS: Implementing effective mechanisms to prevent unauthorized actions and data breaches.
- Data Protection: Ensuring the secure storage and transmission of sensitive user information.

2. User Authentication and Authorization

- Single Sign-On (SSO) Integration: Seamlessly integrating SSO for user login while managing sessions and cookies effectively.
- Role-Based Access Control: Differentiating user permissions to manage access to various features and data.

3. Performance Optimization

- Faster Response Times: Reducing server response times through optimized database queries and backend processes.
- Minimizing Server Requests: Using AJAX and dynamic content loading to reduce server requests and page reloads.

4. Scalability and Reliability

- Handling High Traffic: Ensuring the platform supports a large number of concurrent users without performance issues.
- Cloud Storage Management: Efficiently integrating Cloudflare R2 for scalable data storage and access.

5. User Experience and Interface Design

- Responsive Design: Creating a fully responsive platform for various devices, including smaller screens.
- Intuitive Navigation: Designing a user-friendly interface for easy navigation and interaction.

6. Database Management

- Data Consistency: Maintaining data integrity across modules and user actions.
- Efficient Queries: Optimizing database queries to improve performance and reduce latency.

7. Feature Integration

- Real-Time Collaboration: Adding tools like chat and commenting to facilitate real-time interactions.
- Profile and Project Management: Allowing seamless creation, editing, and management of user profiles and projects.

8. Testing and Debugging

- Comprehensive Testing: Conducting extensive testing to identify and fix bugs, ensuring platform reliability.
- User Feedback Integration: Continuously incorporating user feedback to refine and enhance the platform.

9. Regulatory Compliance

- Data Privacy Laws: Ensuring compliance with regulations such as GDPR to protect user data.
- Content Moderation: Implementing mechanisms to monitor and moderate content according to community guidelines.

These challenges required a meticulous and multifaceted approach, emphasizing secure, efficient, and user-centric development practices to create a robust research collaboration platform for Jadavpur University.

1.4 Contribution

The JU Community Platform will play a crucial role in enhancing the research ecosystem at Jadavpur University. Here are the key contributions of JU Community:

1. Enhanced Collaboration: By providing a centralized platform for communication and collaboration, the JU Community Platform will enable researchers

to work together more effectively. This will lead to more interdisciplinary projects and innovative solutions.

2. Increased Visibility: The platform will offer a space for researchers to showcase their work, increasing visibility and recognition of their contributions. This can attract more funding and collaboration opportunities.

3. Efficient Resource Management: Centralized storage of research materials will streamline resource sharing and management, ensuring that researchers have easy access to the necessary tools and data for their projects.

4. Improved Networking: The ability to create and maintain detailed research profiles will help researchers connect with peers, fostering a more vibrant research community. This can lead to new collaborations and partnerships.

5. Administrative Efficiency: Automated notifications and secure login mechanisms will enhance the efficiency of administrative processes, ensuring that the platform runs smoothly and securely.

1.5 Novelty

The **JU Community Platform** stands out due to its comprehensive approach to facilitating research collaboration within Jadavpur University. Unlike traditional platforms, it integrates multiple functionalities tailored specifically for academic researchers, such as project group systems, resource sharing via cloud storage, and robust user profile management. Its inclusion of real-time collaboration tools, such as chat and commenting, alongside secure, role-based access and a user-friendly, responsive interface, offers an unprecedented level of interactivity and accessibility. This platform not only enhances research productivity but also fosters a cohesive research community, making it a novel solution in the academic domain.

Chapter 2

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for the development of a research scholars' community website, providing a platform for collaboration, knowledge sharing, and networking among scholars within the university.

1.2 Scope

The system will allow registered users to create their personalized research profiles, join research groups, manage research projects, showcase publications and access resources. The website will also provide administrative features for managing user accounts, groups, and system settings.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- UI: User Interface
- DB: Database
- API: Application Programming Interface
- ACL: Access Control List

1.4 Hardware Requirements:

- No. of PCs required: 01
- Configuration: 4 GB (or more) RAM, Intel i3 (or above) processor, 512Gb storage
- Any other devices(s) required: None
- Operating System: Mac / Windows / Linux
- Stable Internet Connection

1.5 Software Requirements:

- Web Browser : Google Chrome , Safari, Firefox

2. Overall Description

2.1 Product Perspective

The website will function as a standalone system, interacting with users through a web-based UI. It will integrate with a backend server for processing user requests and storing/retrieving data from a database.

2.2 Product Functions

- User Registration and Authentication
- Profile Management
- Create/Edit/Delete Research Project
- Create/Edit/Delete Project Milestones
- Create/Edit/Delete Project Resources
- Add/Remove Project Members
- Admin Panel For Approving Member Requests

2.3 User Classes and Characteristics

- Registered Users: Scholars who have created an account and logged into the system.
- Administrators: System administrators responsible for managing user accounts, groups, and system settings.

2.4 Operating Environment

The system will be accessible via modern web browsers (Chrome, Firefox, Safari, etc.) on desktop and mobile devices. It will be hosted on a web server with adequate resources to handle concurrent user requests.

2.5 Design and Implementation Constraints

- The system will be developed using JavaScript-based technologies for both frontend and backend components.
- The UI will be intuitive and user friendly such that users can easily access the features without any prior training.
- Security measures such as encryption, authentication, and authorization will be implemented to protect user data.

3. System Features

3.1 User Management

- Feature: User Registration
 - Description: Allow users to create a new account by providing necessary information.
 - Inputs: UserId, Username, email, password, Roll, Department, Degree, Pass-out, Biography, Default project and any other information.
 - Outputs: Confirmation message upon successful registration

3.2 Profile Management

- Feature: Profile Editing
- Description: Allow users to edit their profile information, including password change, adding additional Information as Dynamic Fields, etc.
- Inputs: Updated profile information.
- Outputs: Confirmation message upon successful profile update.

3.3 Research Project Management

- Feature: Project Creation
- Description: Allow admins to create new research projects and members if and when necessary.
- Inputs: project Id, name, description, members.
- Outputs: Confirmation message upon successful group creation and member addition.

3.5 Project Milestone Showcase

- Feature: Post Upload
- Description: Allow users to upload and showcase their research publications related to the particular project.
- Inputs: Publication metadata, post heading, description, image upload.
- Outputs: Display of uploaded posts on project updates tab.

3.6 Administrative Tools

- Feature: User Management
- Description: Allow administrators to approve or deny new user requests after checking registration data.
- Inputs: User information, access levels.
- Outputs: Confirmation message upon successful user management action.

3.7 Resource Repository

- Feature: Resource Upload and Sharing
- Description: Each research group will have a dedicated resource repository where administrators can upload and share documents, datasets, code snippets, and other resources relevant to the research project.
- Inputs: Resource files (documents, datasets, code), metadata (title, description, tags).
- Outputs: Display of uploaded resources in the group's repository.

4. External Interface Requirements

4.1 User Interfaces

The system will provide a web-based UI accessible via modern web browsers. The UI will be intuitive and visually appealing.

4.2 Hardware Interfaces

The system will require standard hardware components such as servers, storage devices, and network infrastructure for hosting and accessing the application.

4.3 Software Interfaces

The system will interact with external services for features such as email notifications, authentication, and data storage.

4.4 Communication Interfaces

The system will communicate with users via email notifications and RESTful APIs for backend operations.

5. Non-Functional Requirements

5.1 Performance

The system should be able to handle a large number of concurrent users and maintain acceptable response times under heavy load conditions.

5.2 Security

The system should implement security measures such as encryption, authentication, authorization, and data validation to protect user data and prevent unauthorized access.

5.3 Usability

The UI should be intuitive, easy to navigate, and accessible to users with disabilities. User interactions should be consistent and error-tolerant.

5.4 Reliability

The system should be reliable and available 24/7, with minimal downtime for maintenance and upgrades. Backup and recovery mechanisms should be in place to prevent data loss.

5.5 Scalability

The system should be scalable to accommodate future growth in user base and data volume. It should be able to scale horizontally by adding more servers or instances as needed.

5.6 Compatibility

The system should be compatible with a wide range of web browsers, operating systems, and devices to ensure broad accessibility for users

6. Other Requirements

6.1 Legal and Regulatory Requirements

The system should comply with relevant laws and regulations governing data privacy, intellectual property rights, and accessibility standards.

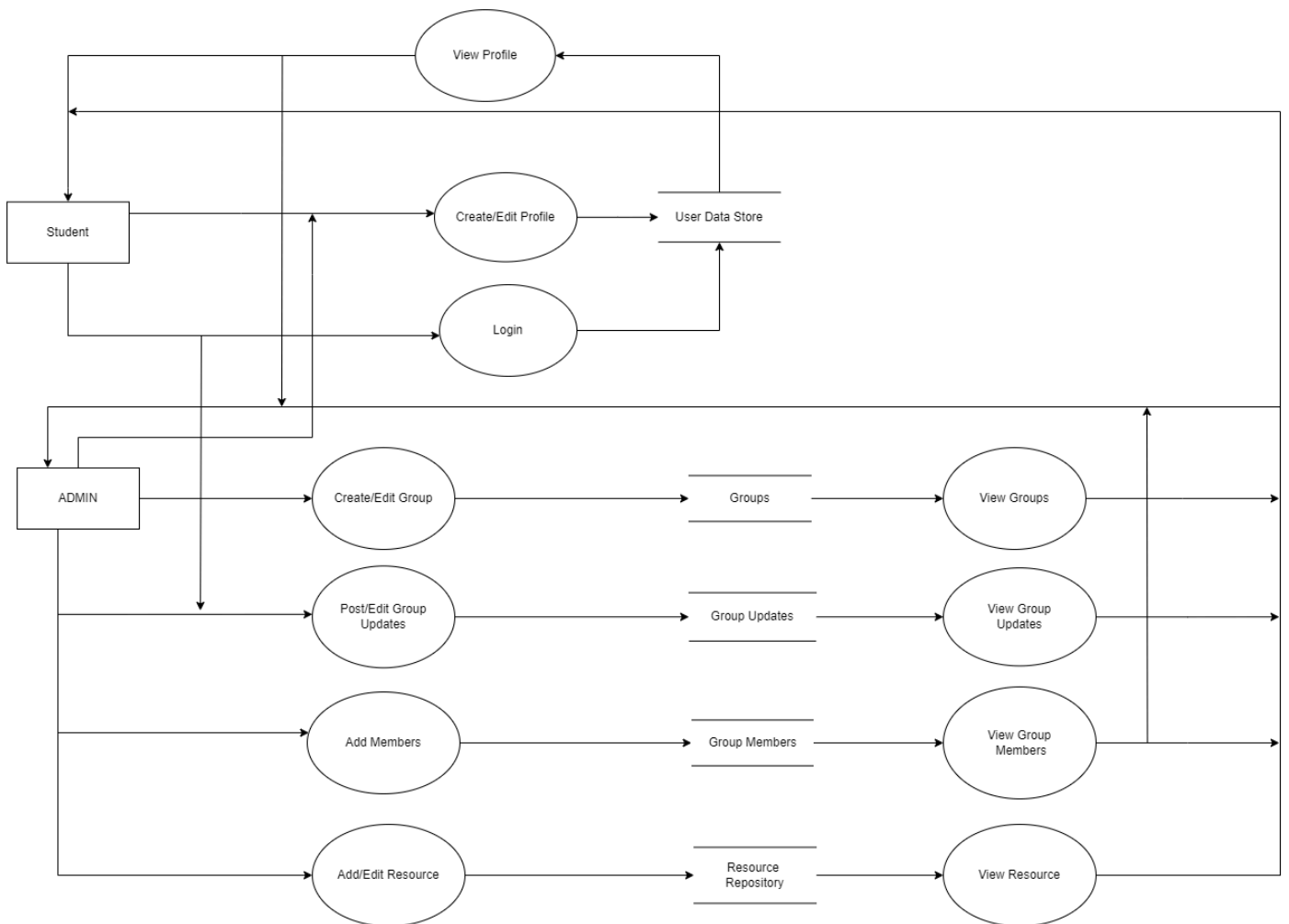
6.2 Documentation Requirements

Comprehensive documentation, including user manuals, technical guides, and system architecture diagrams, should be provided to assist with system usage, maintenance, and troubleshooting.

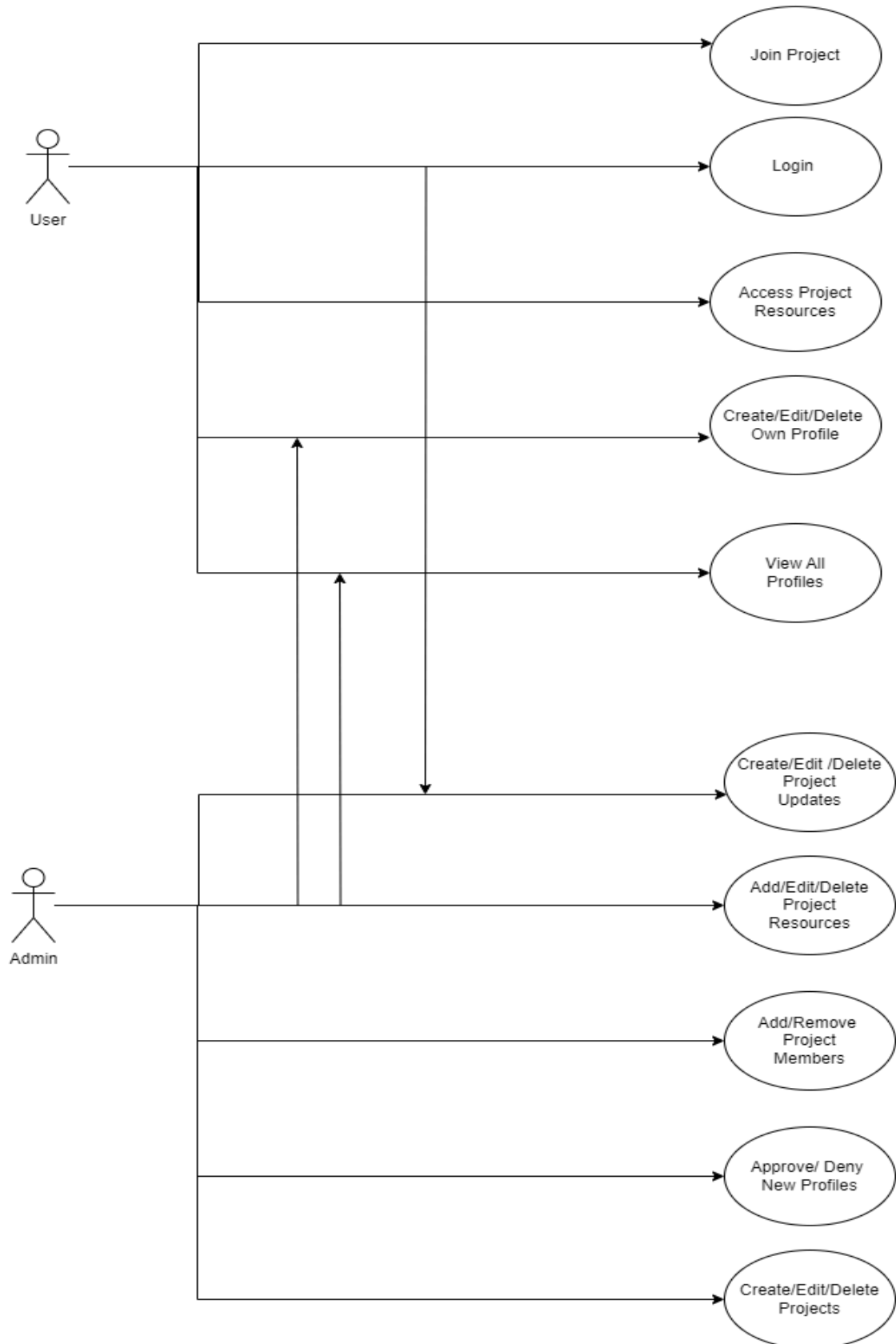
Chapter 3

Proposed Methodology

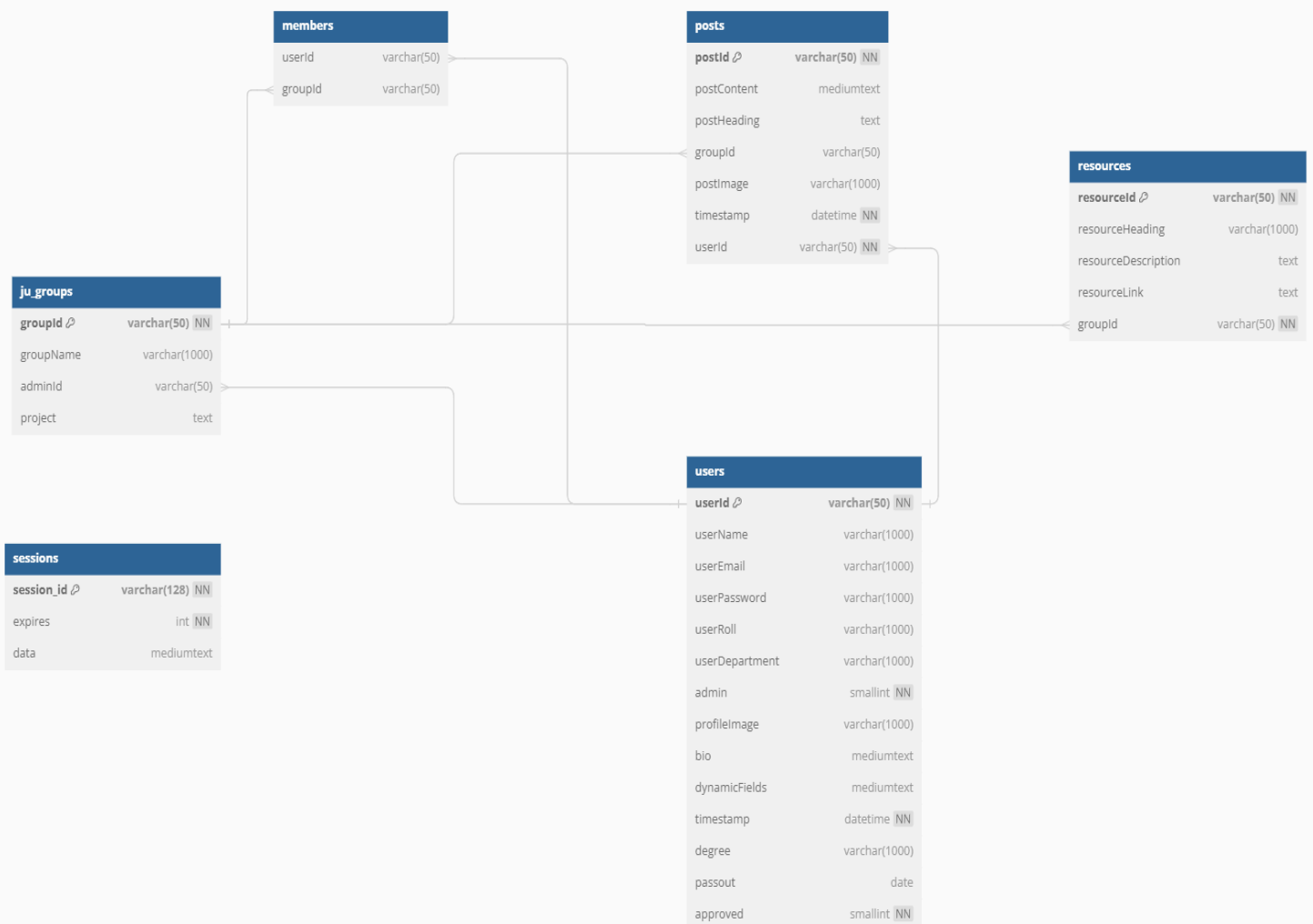
Data Flow Diagram



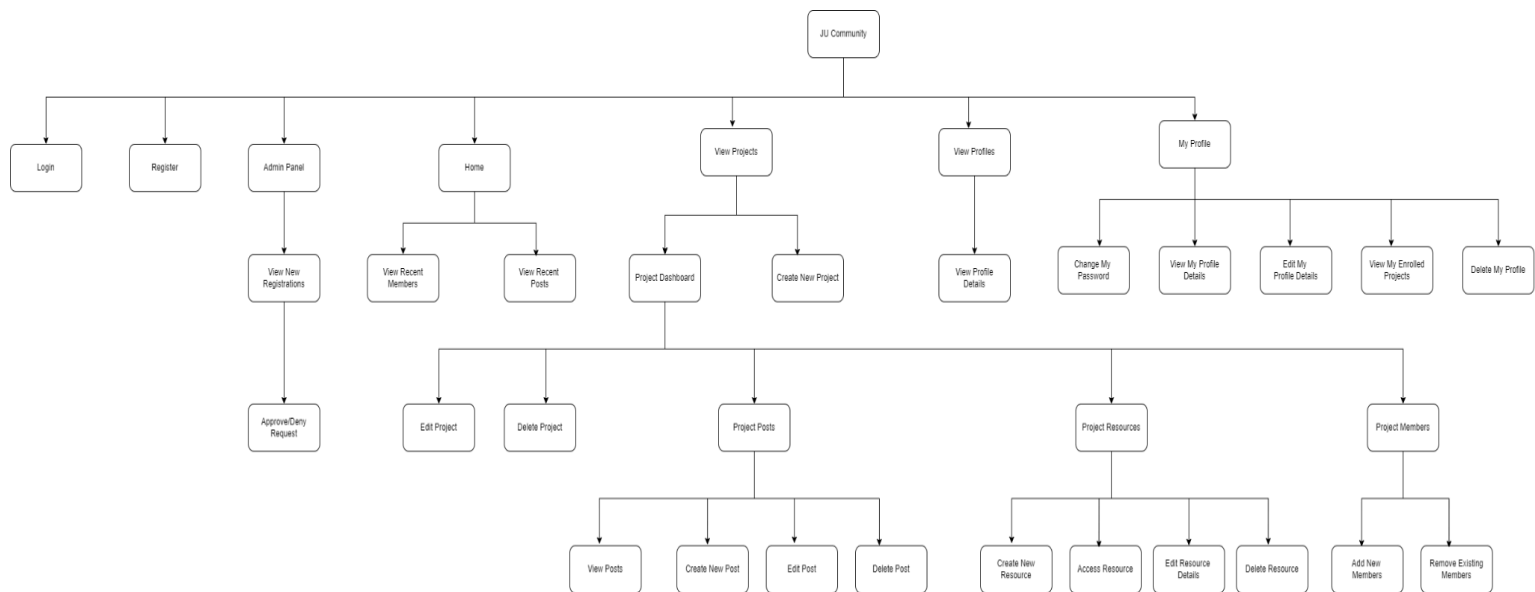
Use Case Diagram



Entity Relationship (ER) Diagram

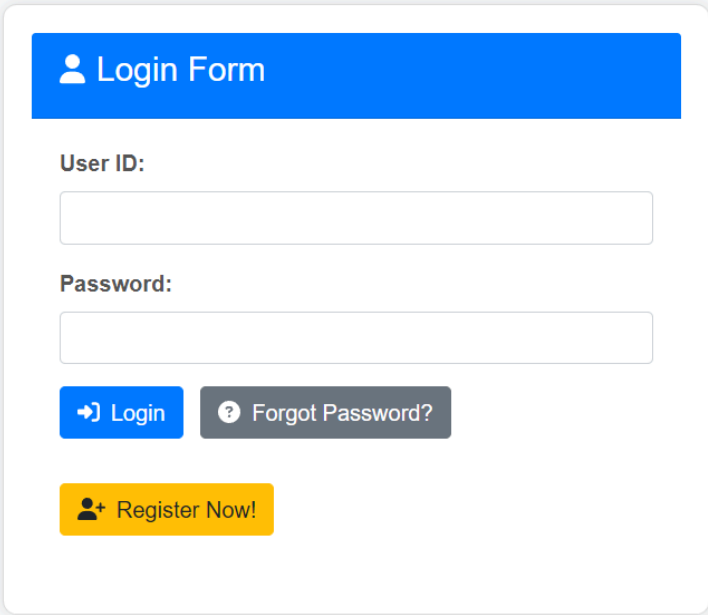


Basic Modules



Detailed Description of JU Community Platform Modules

1. Login Module



Features: -

- **Login:** Users can log in to the community using Single Sign-On (SSO) which simplifies the login process by allowing users to authenticate with their institutional credentials or other SSO providers. This also ensures secure access.
- **Cookies:** Utilizes cookies to maintain user sessions, providing a smoother and persistent login experience.

Preconditions:

- Users must be registered in the system.
- Users must have valid credentials.

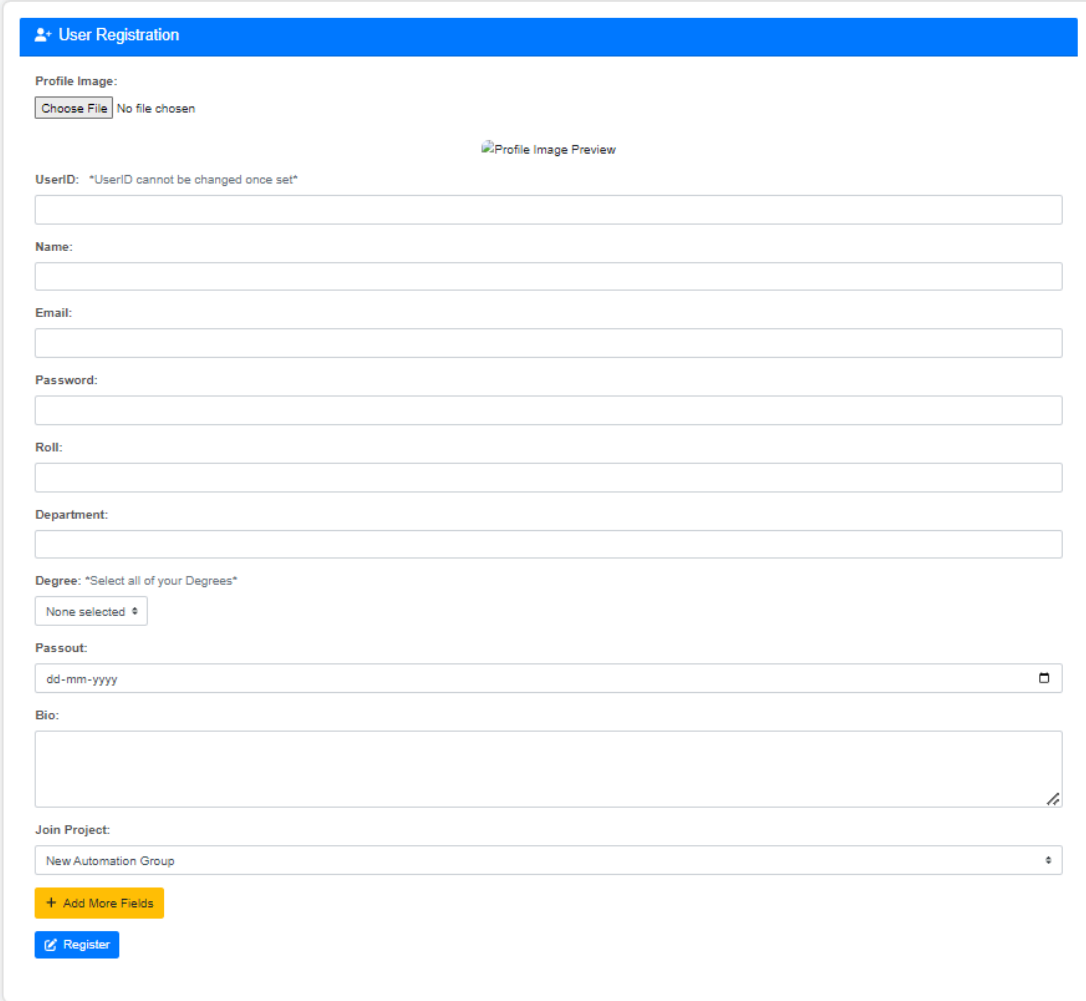
Input:

- Username/Email
- Password

Output:

- User authentication status.
- Access to the community homepage if authentication is successful.

2. Registration Module



The image shows a 'User Registration' form with a blue header bar containing a user icon and the title 'User Registration'. The form fields include: 'Profile Image' with a 'Choose File' button and 'No file chosen' text; a 'Profile Image Preview' placeholder; 'UserID' with a warning '*UserID cannot be changed once set*'; 'Name', 'Email', 'Password', 'Roll', and 'Department' text inputs; 'Degree' with a dropdown menu showing 'None selected'; 'Passout' with a date input (dd-mm-yyyy) and a calendar icon; 'Bio' with a large text area and a rich text editor icon; and 'Join Project' with a dropdown menu showing 'New Automation Group'. At the bottom, there is a yellow '+ Add More Fields' button and a blue 'Register' button with a checkmark icon.

Features:

- **Register:** New users can register for the community by filling out necessary details. Upon registration, an email notification is sent to the admin for approval.

- **Email Notifications:** As part of the registration process, new users receive notifications about the status of their registration, and admins are alerted when a new registration request is made.

Preconditions:

- User should not already have an account.

Input:

- User ID
- Name
- Email
- Password
- University Roll
- Degree
- Passout Year
- Department
- Bio
- Any Extra Information

Output:

- Registration request sent to the admin panel.
- Email notification sent to the admin about the new registration.

3. Admin Panel

JU Community

Home

View Projects

View Profiles

AdminPanel

My Profile

Logout

Registration Requests

Name	Email	Roll	Department	Degree	Passout	Bio	Project ID	Dynamic Fields	Action
Tester	test@gmail.com	002210503056	CSE	["B.Tech","M.Tech"]	2024-05-24	Hii	Group3	View DFs	Approve Deny

Features:

- **View New Registrations:** This feature allows administrators to review all new user registration requests.
- **Approve/Deny Request** Administrators have the authority to approve or deny registration requests, thereby controlling access to the community.
- **Email Notifications:** When a new user registers, an email notification is sent to the admin. Additionally, users receive email notifications regarding the status of their registration (approval or denial) based on the admin's action.

a. View New Registrations

Preconditions:

- Admin is logged into the system.

Input:

- None (admin accesses the registration requests page)

Output:

- List of pending registration requests.

b. Approve/Deny Request

Preconditions:

- Registration request is pending.
- Admin has viewed the registration details.

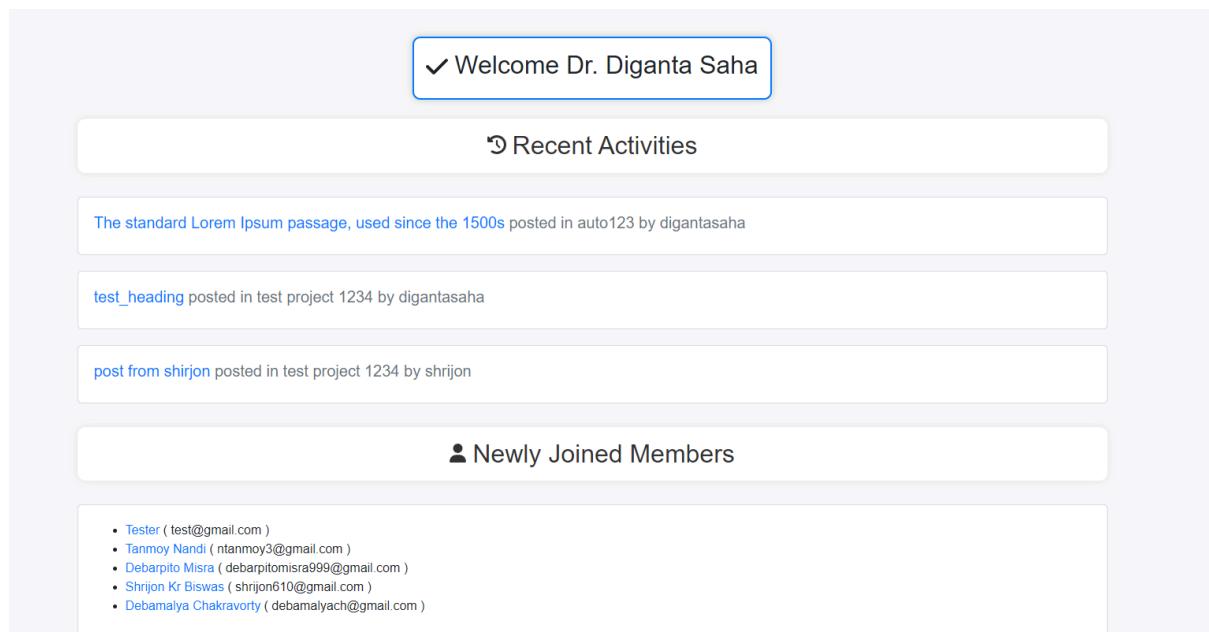
Input:

- Admin's approval or denial action.

Output:

- User receives email notification about the status of their profile.
- Updated profile status in the system.

4. Home Module



Features:

- **View Recent Members:** Displays a list of the most recently registered members, helping the community stay informed about new additions.
- **View Recent Posts:** Shows the latest posts made by members, providing an overview of recent activities and progress within the community.

a. View Recent Members

Preconditions:

- User is logged in.

Input:

- None (system automatically shows recent members).

Output:

- List of recently registered members.

b. View Recent Posts

Preconditions:

- User is logged in.

Input:

- None (system automatically shows recent posts).

Output:

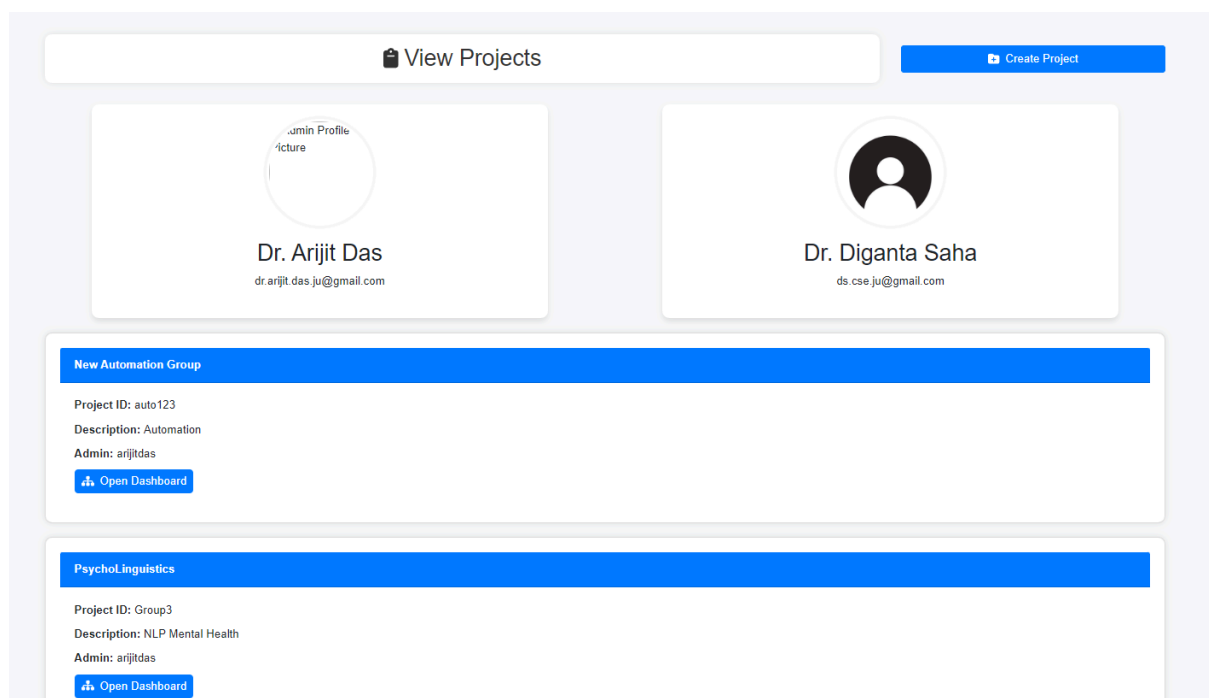
- List of recent posts from various project groups.

5. Projects Module

Features:

- **View Projects:** Serves as a central hub for managing and monitoring all projects. Users can see an overview of active projects and their statuses.
- **Create New Project:** Allows admins to initiate new projects. This feature includes setting up project details and defining goals and objectives.
- **Edit Project:** Provides functionality for admins to modify existing project details, including updating goals, timelines, and team members.
- **Delete Project:** Enables admins to remove projects that are no longer active or needed.

a. View Projects



Preconditions:

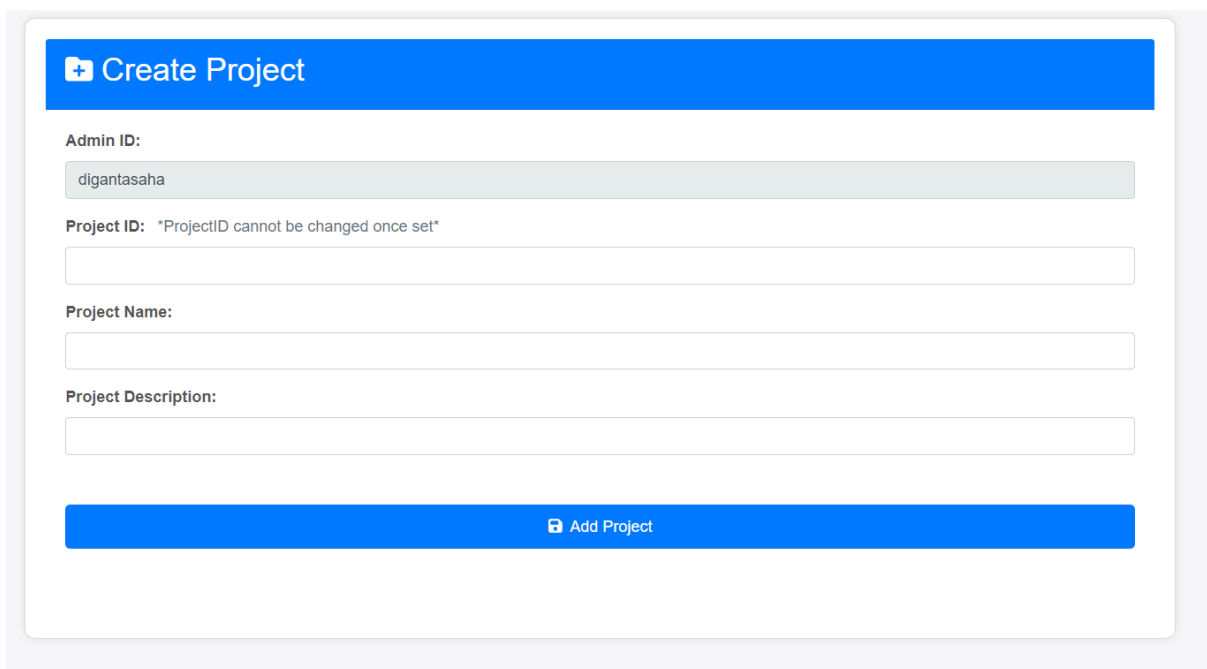
- User is logged in.

Input:

- None (system displays user-related projects).

Output:

- List of all projects in the community.

b. Create New Project

The screenshot shows a web form titled "Create Project" with a blue header bar. Below the header, there are four input fields: "Admin ID" (containing "digantasaha"), "Project ID" (with a warning message "*ProjectID cannot be changed once set*"), "Project Name", and "Project Description". At the bottom, there is a blue button labeled "Add Project".

Create Project

Admin ID:
digantasaha

Project ID: *ProjectID cannot be changed once set*

Project Name:

Project Description:

Add Project

Preconditions:

- Admin is logged in.

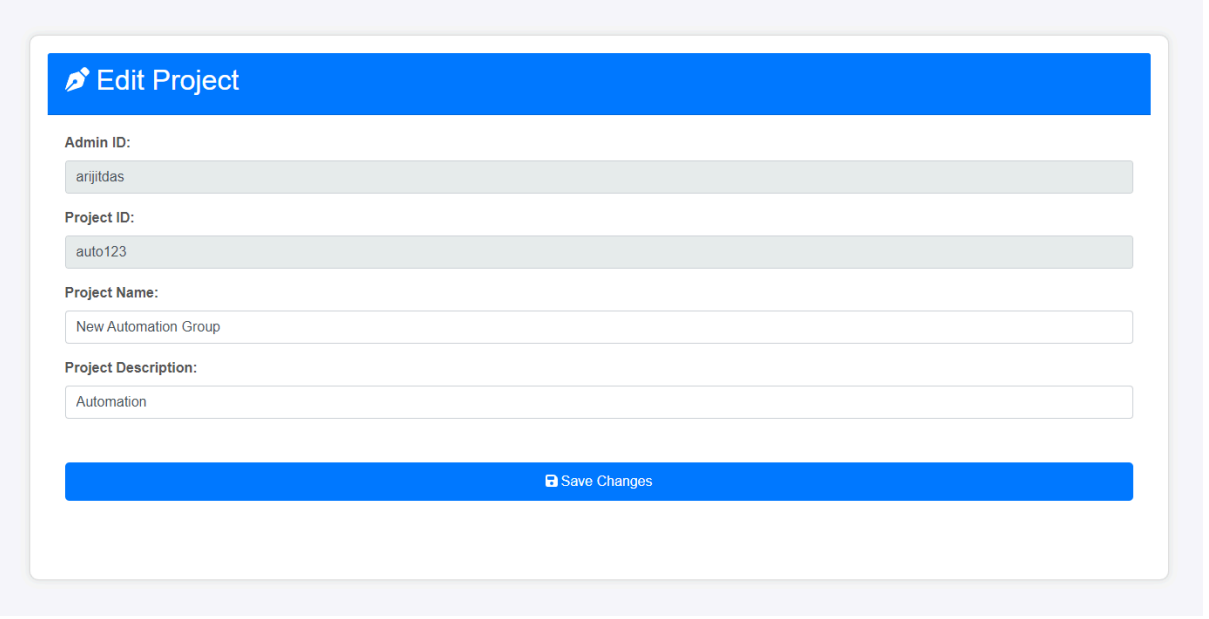
Input:

- Project ID
- Project name
- Description

Output:

- New project created and added to the system.

c. Edit Project



The screenshot shows a web interface for editing a project. At the top is a blue header bar with a pencil icon and the text 'Edit Project'. Below this, the form is organized into sections. The 'Admin ID' section has a grey input field containing 'arjitdas'. The 'Project ID' section has a grey input field containing 'auto123'. The 'Project Name' section has a white input field containing 'New Automation Group'. The 'Project Description' section has a white input field containing 'Automation'. At the bottom of the form is a blue button with a floppy disk icon and the text 'Save Changes'.

Edit Project

Admin ID:
arjitdas

Project ID:
auto123

Project Name:
New Automation Group

Project Description:
Automation

 Save Changes

Preconditions:

- Admin is logged in.

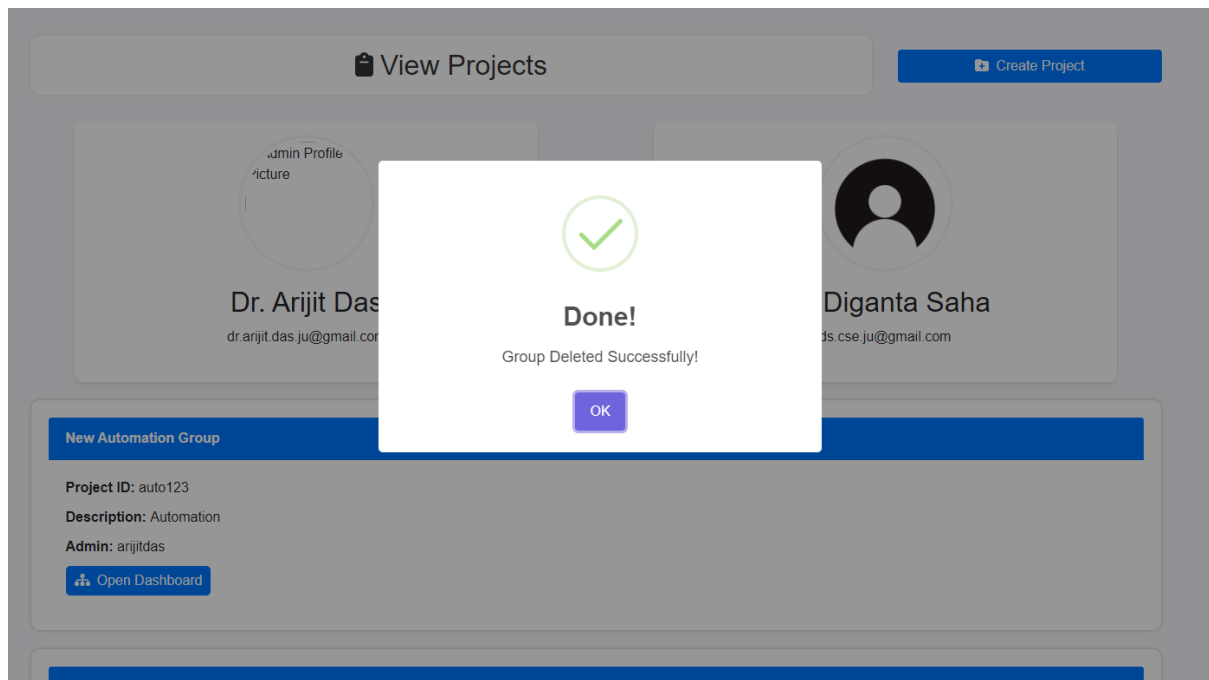
Input:

- Project identifier
- Updated project details (e.g., name, description, members)

Output:

- Project details updated in the system.

d. Delete Project



Preconditions:

- Admin is logged in.

Input:

- Project identifier

Output:

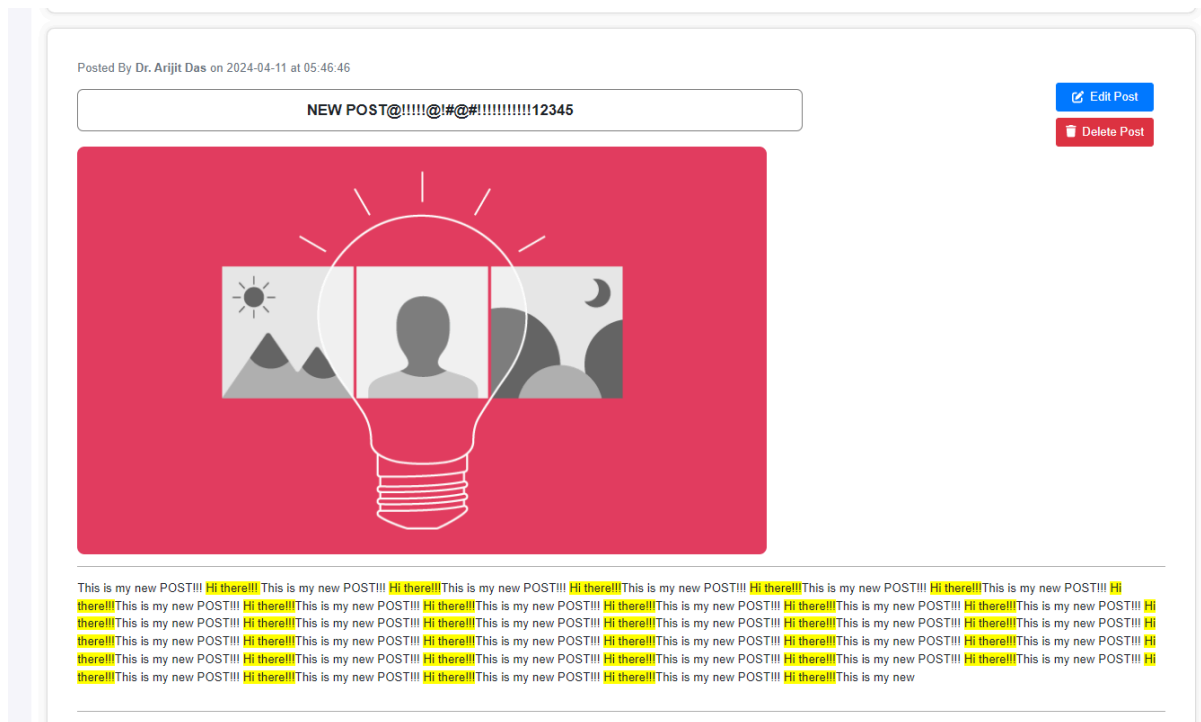
- Project removed from the system.
- Project Members Deallocated

6. Project Posts Module

Features:

- **View Posts:** Users can view all posts related to their projects, which helps in tracking the progress and updates shared by team members.
- **Create New Post:** Allows users to add new posts within a project, sharing updates, findings, or relevant information with their project group.
- **Edit Post:** Provides the option to modify existing posts to correct errors or update information.
- **Delete Post:** Enables users to remove their posts that are outdated or no longer relevant.

a. View Posts



Preconditions:

- User is logged in.

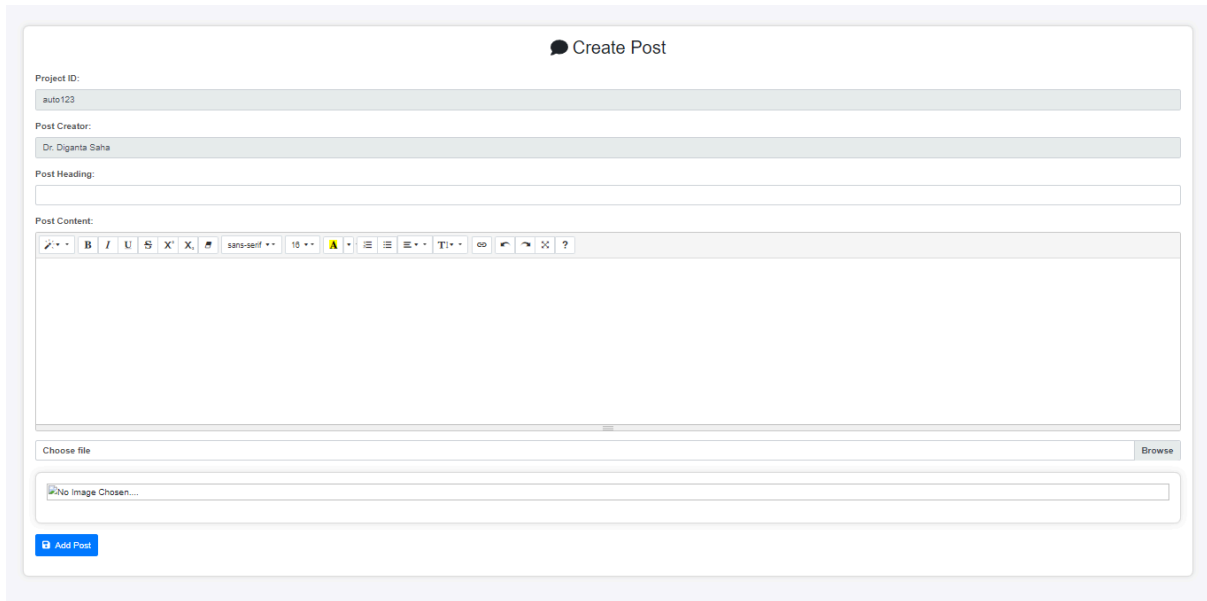
Input:

- None (system shows project-specific posts).

Output:

- List of posts related to the project.

b. Create New Post



The image shows a web form titled "Create Post" with a speech bubble icon. The form contains several input fields and a rich text editor. The "Project ID" field is pre-filled with "auto123". The "Post Creator" field is pre-filled with "Dr. Diganta Saha". The "Post Heading" field is empty. The "Post Content" field is a rich text editor with a toolbar containing icons for bold, italic, underline, strikethrough, link, unlink, text color, background color, bulleted list, numbered list, indent, outdent, undo, redo, and help. Below the rich text editor is a "Choose file" section with a "Browse" button. Below that is a "No Image Chosen..." placeholder. At the bottom left is a blue "Add Post" button.

Create Post

Project ID:
auto123

Post Creator:
Dr. Diganta Saha

Post Heading:

Post Content:

Choose file Browse

No Image Chosen...

Add Post

Preconditions:

- User is a member of the project.

Input:

- Post title
- Content
- Post Image

Output:

- New post added to the project.

c. Edit Post

[illegible]

Preconditions:

- User is admin OR User is the creator of the post

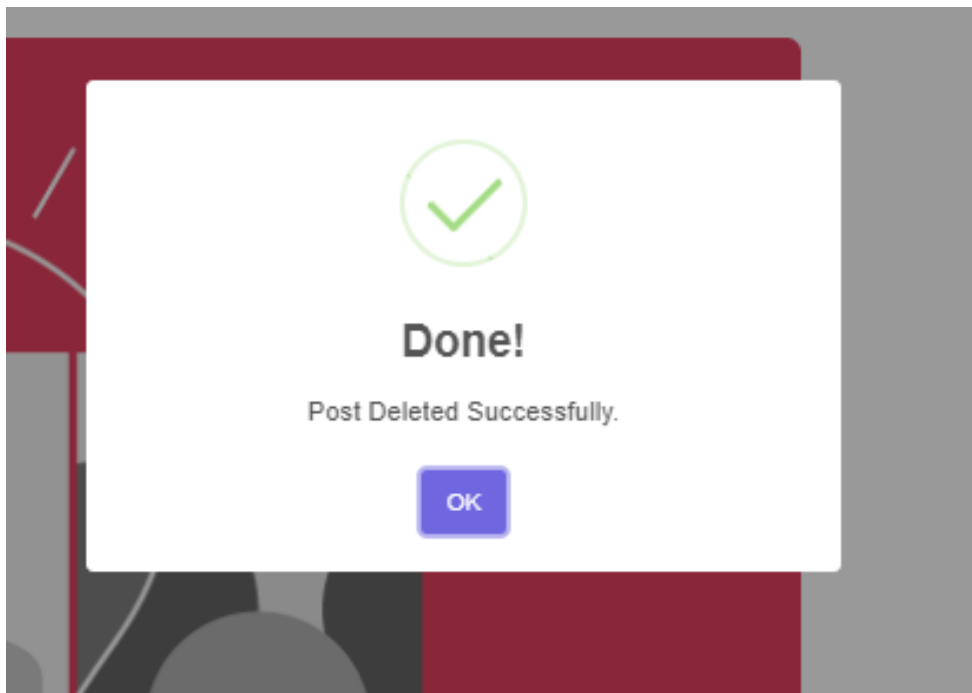
Input:

- Updated post content

Output:

- Post updated in the project.

d. Delete Post



Preconditions:

- User is admin OR User is the creator of the post

Input:

- Post identifier

Output:

- Post removed from the project.

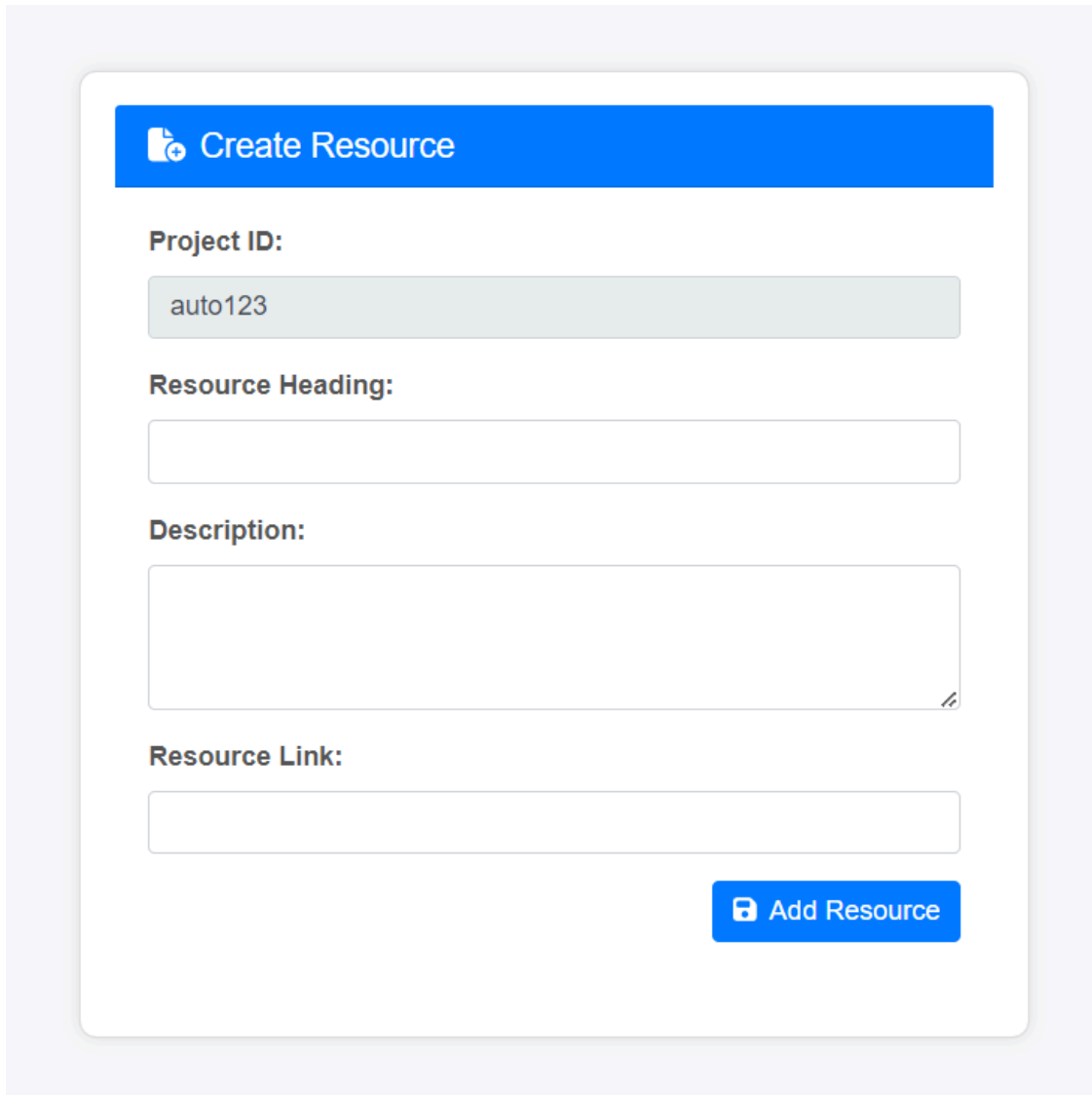
7. Project Resources Module

Features:

- **Create New Resource:** Administrators can add new resources such as datasets, code snippets, or documents that are relevant to specific projects by uploading them as Google Drive links.
- **Access Resource:** Members can access and utilize the resources added to their project groups via the provided Google Drive links, facilitating easier collaboration and data sharing.
- **Edit Resource Details:** Administrators can update the details of existing resources, ensuring that the most current and accurate information is available.

- **Delete Resource:** Administrators can remove resources that are no longer needed or relevant.

a. Create New Resource

A screenshot of a web form titled "Create Resource". The form is contained within a white rounded rectangle on a light gray background. At the top is a blue header bar with a document icon and the text "Create Resource". Below this, the form has four sections: "Project ID:" with a text input containing "auto123"; "Resource Heading:" with an empty text input; "Description:" with a larger empty text area; and "Resource Link:" with an empty text input. At the bottom right of the form is a blue button with a document icon and the text "Add Resource".

Create Resource

Project ID:

auto123

Resource Heading:

Description:

Resource Link:

Add Resource

Preconditions:

- Admin is logged in.

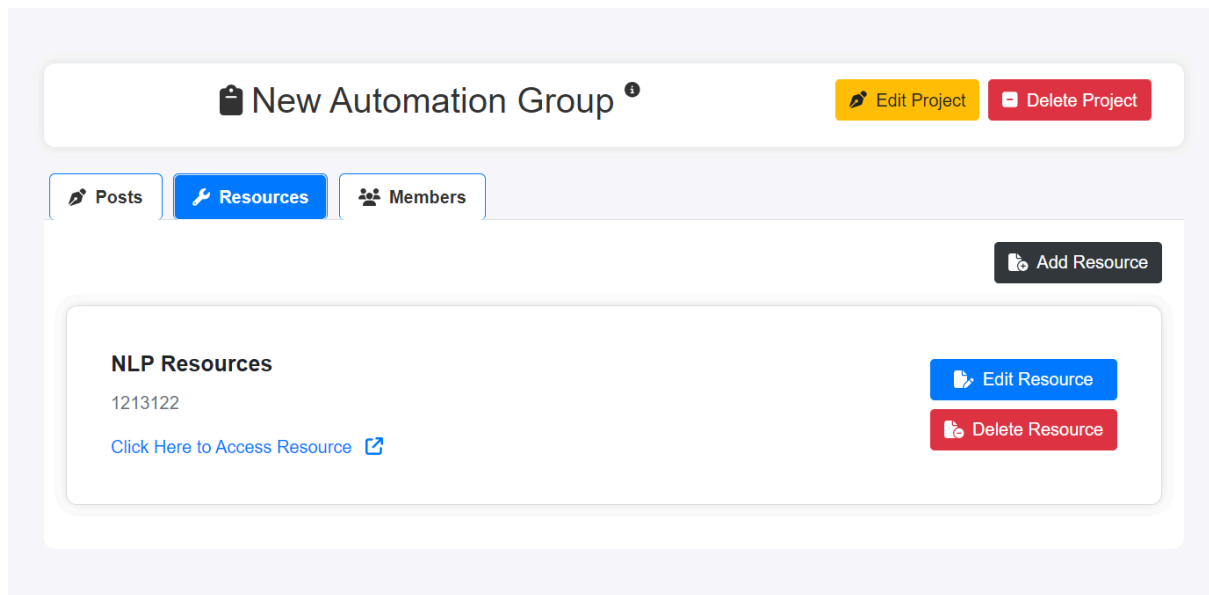
Input:

- Resource details
- Google Drive link

Output:

- Resource added to the project.

b. Access Resource



Preconditions:

- User is logged in and Member of the project

Input:

- Resource Identifier

Output:

- Access to the Google Drive link.

c. Edit Resource Details

 Edit Resource

Project ID:

auto123

Resource ID:

ea630d2d-e645-48f8-b9f3-a65534e87c23

Resource Heading:

NLP Resources

Description:

1213122

Resource Link:

<https://drive.google.com/drive/folders/1kZc-A8HoRxS3bCl3hcE3ocoOz-mg>

Save Changes

Preconditions:

- Admin is logged in.

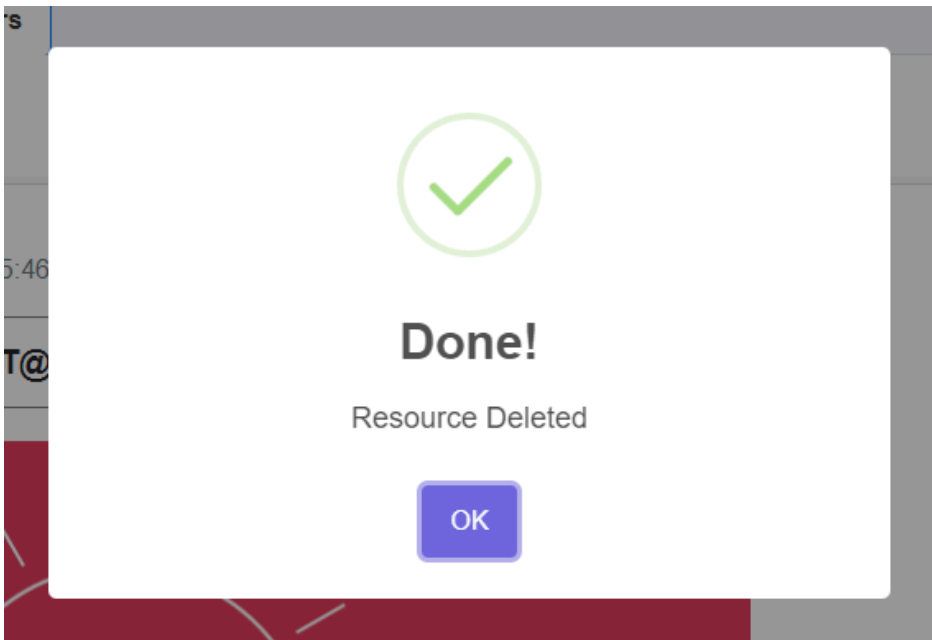
Input:

- Updated resource details
- Updated Google Drive link

Output:

- Resource details updated.

d. Delete Resource



Preconditions:

- Admin is logged in.

Input:

- Resources

Output:

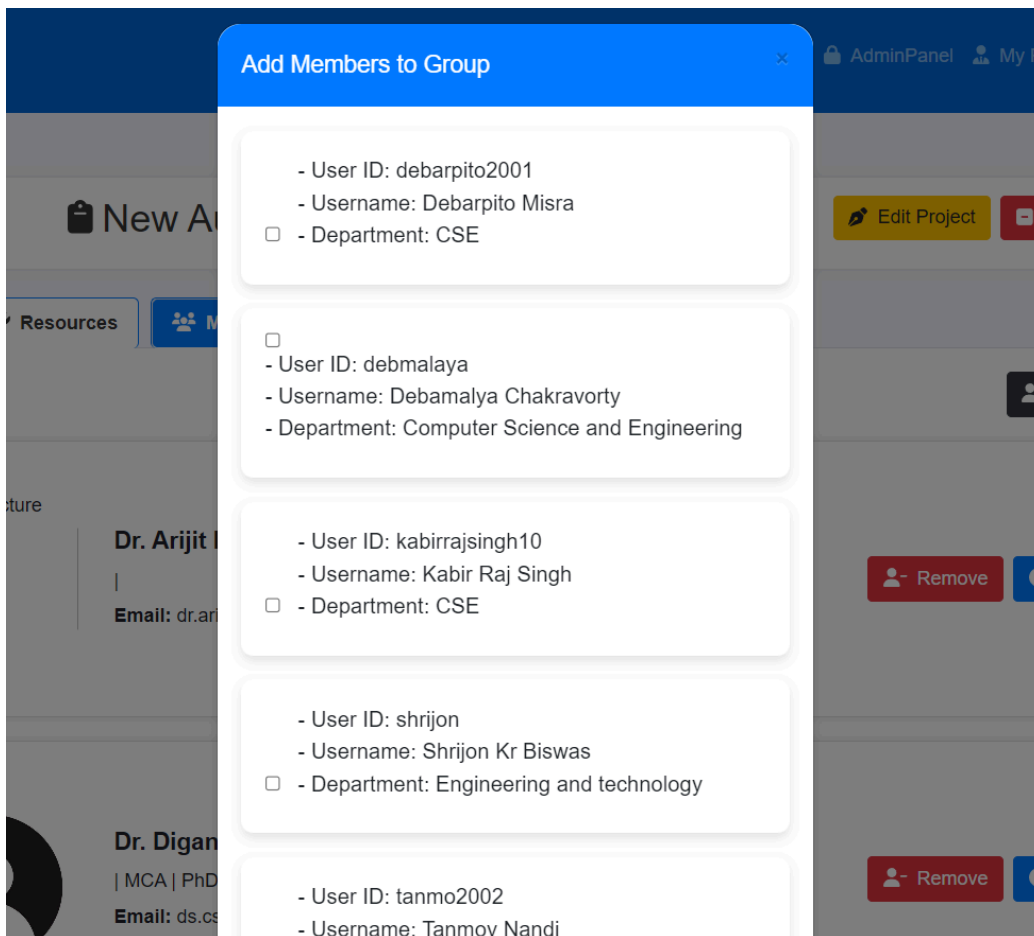
- Resource removed from the project.

8. Project Members Module

Features:

- **Add New Members:** Allows administrators to add new members to specific project groups, ensuring that the right people have access to the project.
- **Remove Existing Members:** Enables administrators to remove members from project groups as necessary, maintaining the relevance and effectiveness of the team.

a. Add New Members



Preconditions:

- Admin is logged in.

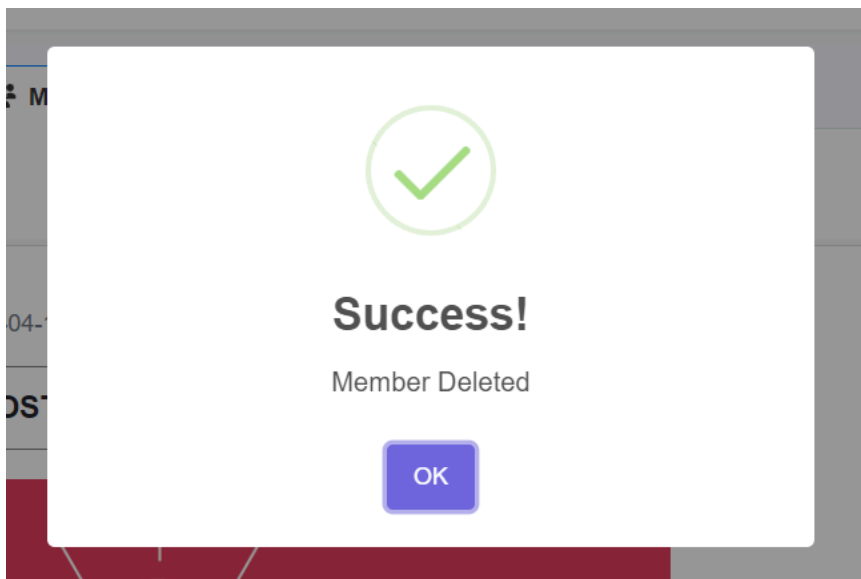
Input:

- Select profiles to be added from profile list

Output:

- New members added to the project.

b. Remove Existing Members



Preconditions:

- Admin is logged in.

Input:

- Member identifier

Output:

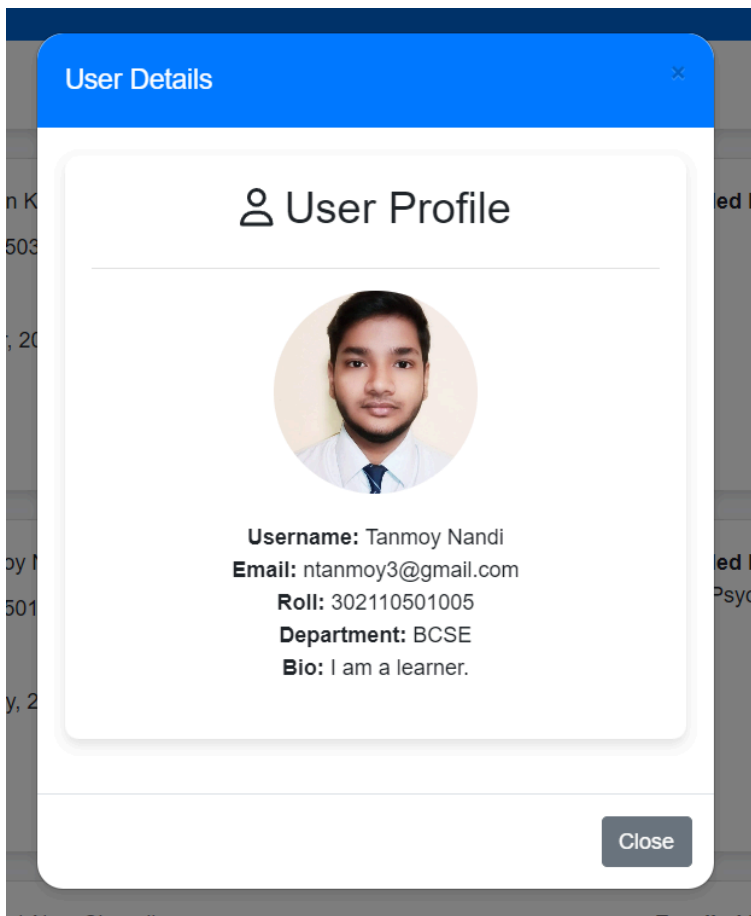
- Member removed from the project.

9. View Profiles Module

Features:

View Profile Details: Users can view detailed profiles of other members, which include information such as name, email, SCIID, GitHub, and profile picture. This feature helps in networking and understanding the expertise of other members.

a. View Profile Details



Preconditions:

- User is logged in.

Input:

- Profile Identifier.

Output:

- Profile information displayed.

10. My Profile Module

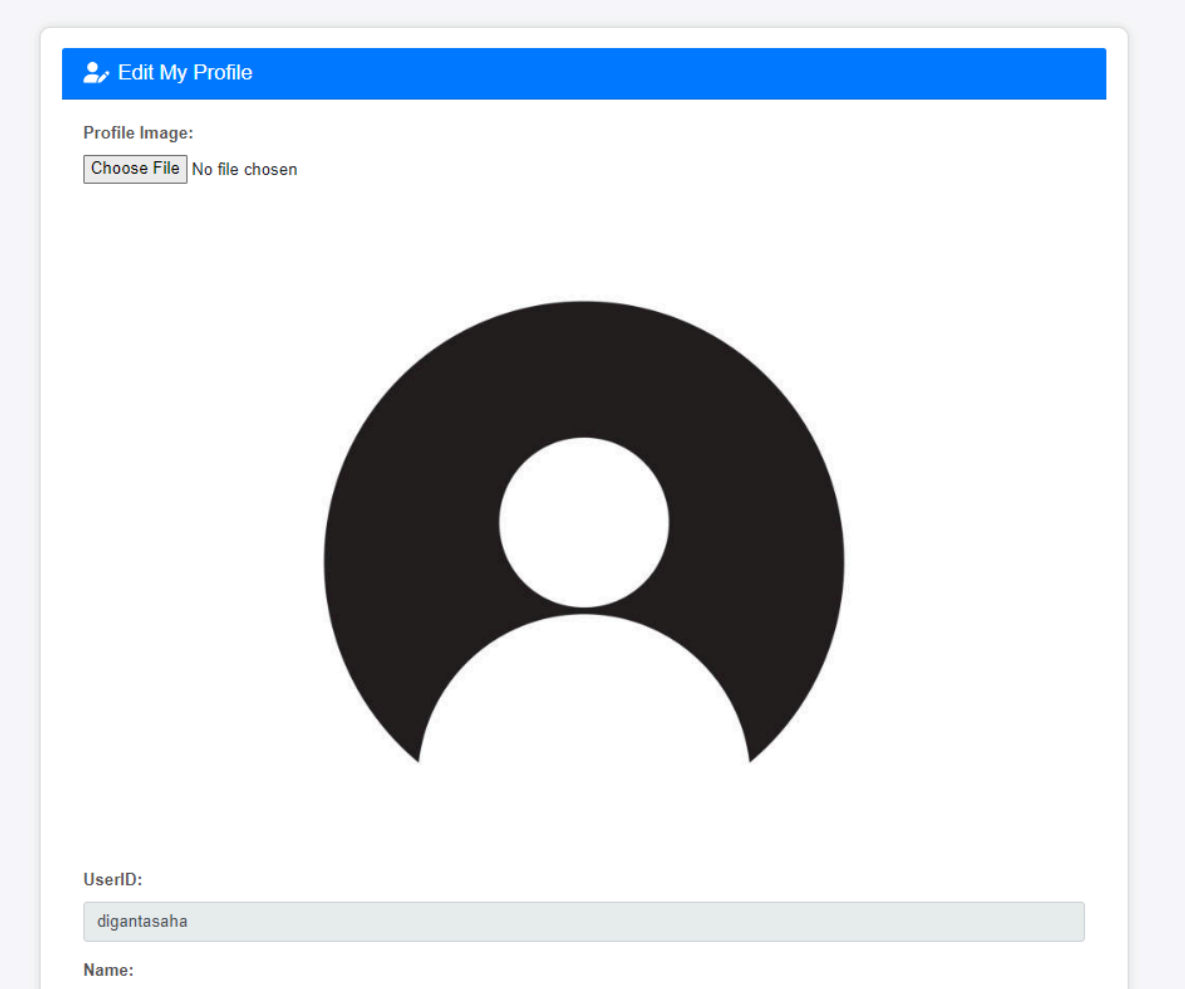
Features:

- **Change My Password:** Users can update their password to maintain account security.
- **Edit My Profile Details:** Users can modify their profile information, including updating their name, email, SCIID, GitHub, and profile picture. Additionally, users

can add more social media links or academic identifiers beyond GitHub and SCIID.

- **View My Enrolled Projects:** Displays a list of projects that the user is currently participating in, providing a quick reference to their active collaborations.
- **Delete My Profile:** Allows users to delete their profile from the community, removing all their data from the system.

a. Edit My Profile Details



The screenshot shows a web form titled "Edit My Profile" with a blue header bar. Below the header, there is a "Profile Image:" section with a "Choose File" button and the text "No file chosen". In the center of the form is a large, dark gray circular placeholder for a profile picture, featuring a white circle in the middle. Below this, there are two input fields: "UserID:" with the value "digantasaha" and "Name:" which is currently empty.

Preconditions:

- User is logged in.

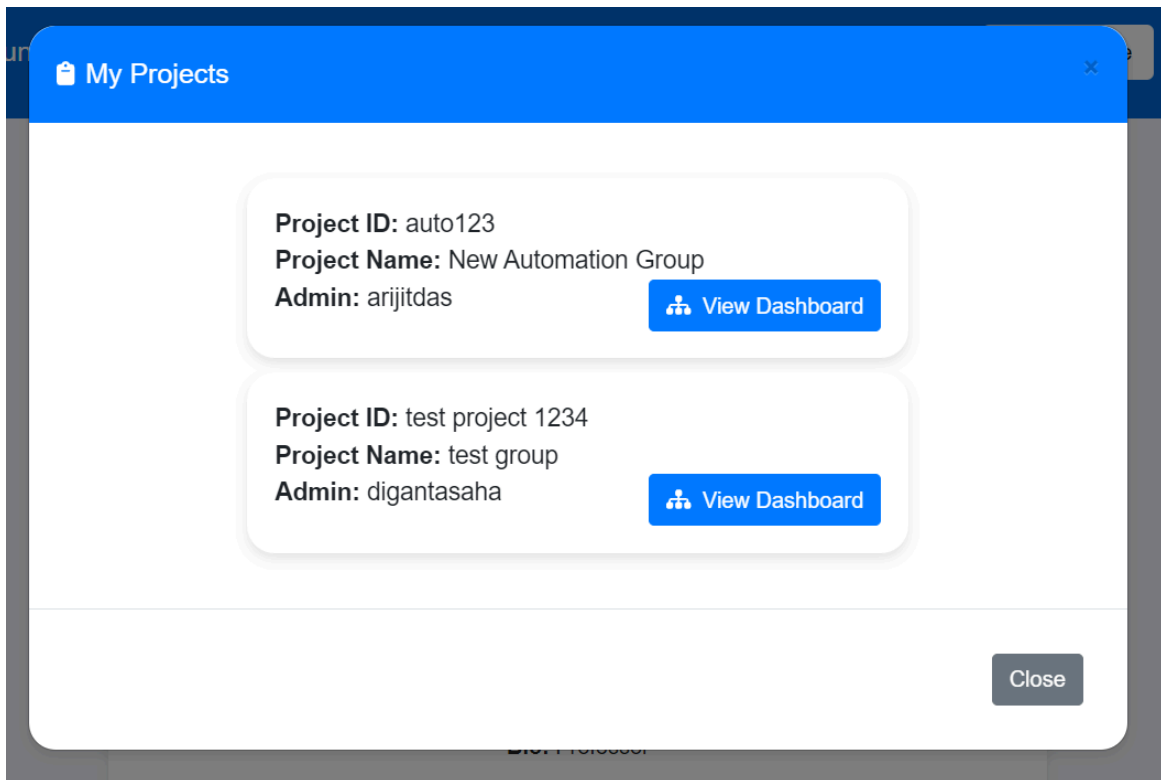
Input:

- Updated profile information

Output:

- Profile details updated.

b. View My Enrolled Projects



Preconditions:

- User is logged in.

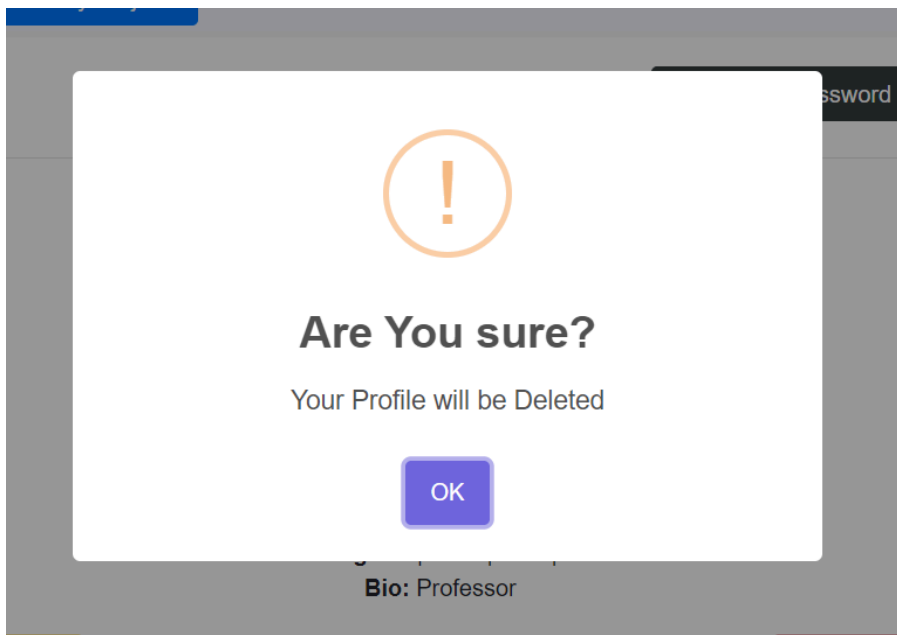
Input:

- None (system displays enrolled projects).

Output:

- List of projects the user is enrolled in.

c. Delete My Profile



Preconditions:

- User is logged in.

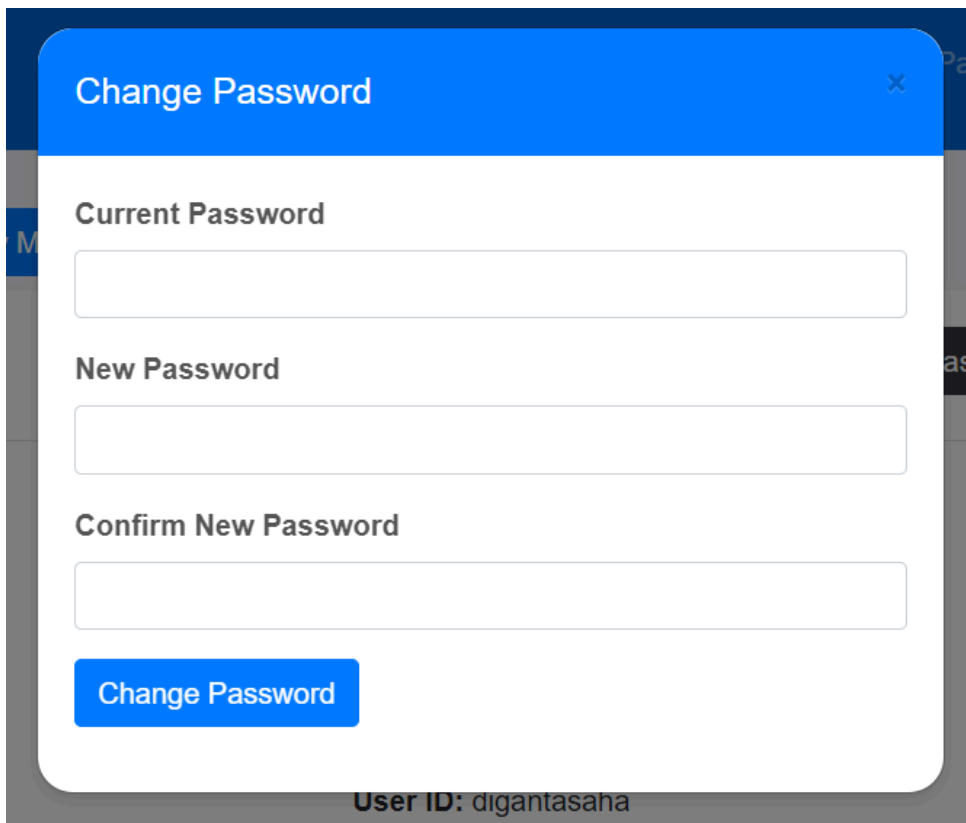
Input:

- Confirmation to delete profile

Output:

- User profile deleted from the system.

d. Change My Password



A screenshot of a 'Change Password' dialog box. The dialog has a blue header bar with the title 'Change Password' and a close button (X) on the right. Below the header, there are three input fields: 'Current Password', 'New Password', and 'Confirm New Password'. Each field is a simple white rectangle with a thin grey border. At the bottom of the dialog, there is a blue button with the text 'Change Password' in white. The dialog is set against a dark blue background. At the bottom of the background, the text 'User ID: digantasaha' is visible.

Change Password

Current Password

New Password

Confirm New Password

Change Password

User ID: digantasaha

Preconditions:

- User is logged in.

Input:

- Current password
- New password

Output:

- Password updated.

Technologies Used

The JU Community Platform leverages a variety of technologies to provide a robust, efficient, and user-friendly environment for researchers and students at Jadavpur University. Here's a detailed overview of the technologies employed:

Frontend Technologies:

1. HTML (HyperText Markup Language)

- Description: HTML is the standard markup language used to create the structure of web pages. It defines the elements and layout of a web page.
- Usage: Used to structure the content on the web pages of the JU Community Platform.

2. CSS (Cascading Style Sheets)

- Description: CSS is used to style and design web pages, controlling the layout, colors, fonts, and overall presentation.
- Usage: Applied to enhance the visual appearance of the web pages, ensuring a consistent and visually appealing design.

3. Bootstrap

- Description: Bootstrap is a powerful front-end framework that includes HTML, CSS, and JavaScript components for creating responsive and mobile-first web pages.
- Usage: Utilized to ensure the platform is responsive, making it accessible on various devices and screen sizes. It also speeds up development with pre-designed components.

4. EJS (Embedded JavaScript)

- Description: EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.
- Usage: Used for server-side rendering of HTML, allowing dynamic content generation based on the application's state and user input.

Backend Technologies

1. Node.js

- Description: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, designed for building scalable network applications.
- Usage: Acts as the runtime environment for executing JavaScript code on the server side, handling asynchronous operations and network requests efficiently.

2. Express.js

- Description: Express is a fast, unopinionated, minimalist web framework for Node.js.
- Usage: Provides the core infrastructure for the backend, handling routing, middleware, and HTTP requests/responses.

3. bcrypt

- Description: bcrypt is a password hashing library.
- Usage: Ensures secure storage of user passwords by hashing them before saving to the database.

4. dotenv

- Description: dotenv is a zero-dependency module that loads environment variables from a `.env` file into `process.env`.
- Usage: Manages configuration and secrets, ensuring sensitive information like database credentials and API keys are not hard-coded.

5. express-session

- Description: express-session is a middleware for managing sessions in Express applications.
- Usage: Handles user sessions, keeping track of user data across multiple requests.

6. multer

- Description: Multer is a middleware for handling `'multipart/form-data'`, primarily used for uploading files.
- Usage: Manages file uploads within the platform, such as profile pictures and project resources.

7. nodemailer

- Description: Nodemailer is a module for Node.js applications to send emails.
- Usage: Sends email notifications for various actions, such as registration approval or project updates.

8. sanitize-html

- Description: sanitize-html is a library for sanitizing user input to prevent cross-site scripting (XSS) attacks.
- Usage: Ensures all HTML content input by users is sanitized to prevent security vulnerabilities.

Cloud Storage

1. Cloudflare R2

- Description: Cloudflare R2 is an object storage service that offers high availability and low latency.
- Usage: Provides scalable and reliable storage for user-uploaded files, such as research documents and datasets.

Database

1. Aiven MySQL

- Description: Aiven offers managed MySQL services in the cloud.
- Usage: Stores and manages the relational data for the platform, such as user information, project details, posts, and resources.

Website Hosting

1. Vercel

- Description: Vercel is a platform for frontend developers, providing hosting services optimized for React, Next.js, and static websites.
- Usage: Hosts the frontend of the JU Community Platform, ensuring fast and efficient delivery of web pages to users.

Integration and Workflow

The integration of these technologies ensures a seamless workflow from frontend to backend, providing a cohesive and efficient user experience. HTML and CSS, enhanced by Bootstrap, build the structure and design of the web pages. EJS dynamically generates HTML content based on user data, while Node.js and Express manage the server-side logic and routing. User security is handled by bcrypt and express-session, while file uploads and email notifications are managed by multer and nodemailer, respectively. Sanitizing user input with sanitize-html ensures a secure platform, and Cloudflare R2 and Aiven MySQL provide robust storage solutions. Finally, Vercel ensures the frontend is delivered quickly and efficiently to users.

This comprehensive technology stack enables the JU Community Platform to offer a powerful, secure, and user-friendly environment for researchers and students to collaborate and share resources effectively.

Chapter 4

Security and Testing

4.1 Vulnerability Scan

Introduction

The JU Community Platform, designed to facilitate communication and collaboration among researchers, must address several security vulnerabilities to ensure the safety and integrity of user data and interactions. This report identifies current vulnerabilities, evaluates their potential impact, and provides recommendations for mitigation.

Passive Web Application Vulnerabilities

1. Absence of Anti-CSRF Tokens

- Threat Level: Medium
 - Description: Without Cross-Site Request Forgery (CSRF) tokens, the platform is susceptible to unauthorized commands transmitted from a user that the website trusts. This vulnerability can allow attackers to perform actions on behalf of authenticated users without their consent.
 - Impact: Unauthorized transactions, data manipulation, and potential account takeovers.

2. Missing Anti-clickjacking Header

- Threat Level: Medium
 - Description: The absence of an anti-clickjacking header leaves the platform vulnerable to clickjacking attacks, where malicious actors trick users into clicking on elements that perform unintended actions.
 - Impact: Unauthorized actions, such as changing settings or initiating transactions, leading to potential data breaches and loss of user trust.

3. CSP: Wildcard Directive and CSP Header Not Set

- Threat Level: Medium
 - Description: A poorly configured Content Security Policy (CSP) or the absence of a CSP allows for the injection of malicious content, such as scripts and styles, from unauthorized sources.
 - Impact: Cross-Site Scripting (XSS) attacks, data theft, and compromised site integrity.

4. X-Content-Type-Options Header Missing

- Threat Level: Low
 - Description: The missing `X-Content-Type-Options` header allows browsers to perform MIME type sniffing, which can lead to the execution of malicious files disguised as benign ones.

- Impact: Execution of malicious files, leading to potential data compromise and system exploitation.

5. Cross-Domain JavaScript Source File Inclusion

- Threat Level: Low
- Description: Including JavaScript files from external domains without validation can introduce vulnerabilities if those external sources are compromised.
- Impact: Execution of malicious scripts, leading to data theft, session hijacking, and unauthorized actions.

6. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)

- Threat Level: Low
- Description: Disclosing server technology through the 'X-Powered-By' header can provide attackers with information that aids in targeting specific exploits.
- Impact: Increased risk of targeted attacks exploiting known vulnerabilities of the disclosed technology.

7. Cookie Without Secure Flag

- Threat Level: Low
- Description: Cookies transmitted without the 'Secure' flag are vulnerable to interception over unencrypted connections.
- Impact: Exposure of sensitive session information, leading to session hijacking and unauthorized access.

8. Private IP Disclosure

- Threat Level: Low
- Description: Exposure of private IP addresses can provide useful information to attackers, aiding in further network penetration efforts.
- Impact: Network reconnaissance by attackers, potentially leading to targeted attacks on internal systems.

9. Strict-Transport-Security Header Not Set

- Threat Level: Low
- Description: Without the Strict-Transport-Security (HSTS) header, the platform does not enforce the use of HTTPS, increasing the risk of man-in-the-middle attacks.
- Impact: Interception and manipulation of data in transit, leading to potential data breaches and compromised user privacy.

10. Cookie without SameSite Attribute

- Threat Level: Low
- Description: Cookies lacking the 'SameSite' attribute are more susceptible to being sent along with cross-site requests, increasing the risk of CSRF attacks.

- Impact: Unauthorized actions on behalf of users, leading to data integrity issues and potential account compromise.

Network Vulnerabilities

1. Missing 'Secure' Cookie Attribute (HTTP)

- Threat Level: Medium
 - Description: Cookies set without the 'Secure' attribute can be transmitted over unencrypted HTTP connections, exposing them to interception.
 - Impact: Exposure of sensitive session information, leading to session hijacking and unauthorized access.

2. Open TCP Ports: 443 and 80

- Threat Level: Low
 - Description: Open TCP ports, especially 443 (HTTPS) and 80 (HTTP), are necessary for web services but can be entry points for attacks if not properly secured.
 - Impact: Potential exploitation through known vulnerabilities in web services, leading to unauthorized access and data breaches.

Conclusion

Addressing the identified security vulnerabilities is crucial for enhancing the security posture of the JU Community Platform. Implementing the recommended mitigations will protect the platform against potential threats and ensure the safety and integrity of user data and interactions. Regular security assessments and updates are essential to maintain a robust security framework and adapt to emerging threats.

4.2 Testing

Introduction

Testing is a critical component of the software development lifecycle, ensuring that the JU Community Platform operates as intended and is free of significant bugs and errors. This report details the different types of testing performed to identify and rectify issues, thus enhancing the platform's reliability and security.

Types of Testing

1. Unit Testing

- Purpose: Verify the functionality of individual components in isolation.
- Outcome: Early detection and resolution of bugs within individual functions, simplifying the debugging process.

2. Integration Testing

- Purpose: Ensure that different modules and services work together as expected.
- Outcome: Identification and resolution of interface defects between integrated units, ensuring cohesive functionality across the platform.

3. System Testing

- Purpose: Validate the complete and fully integrated software product.
- Outcome: Confirmation that the platform operates as a whole, with all components working together seamlessly, meeting the specified requirements.

4. User Acceptance Testing (UAT)

- Purpose: Verify that the platform meets business requirements and is ready for deployment.
- Outcome: Validation from the user's perspective, ensuring the platform is user-friendly and fulfills the intended use cases.

5. Performance Testing

- Purpose: Ensure the platform performs well under expected load conditions.
- Outcome: Identification of performance bottlenecks, leading to optimizations that enhance the user experience.

6. Security Testing

- Purpose: Identify vulnerabilities and ensure the platform's data and resources are protected.
- Outcome: Discovery and resolution of security loopholes, strengthening the platform's security posture.

7. Compatibility Testing

- Purpose: Ensure the platform is compatible with various devices, operating systems, and browsers.
- Outcome: Consistent and reliable user experience across different platforms and devices.

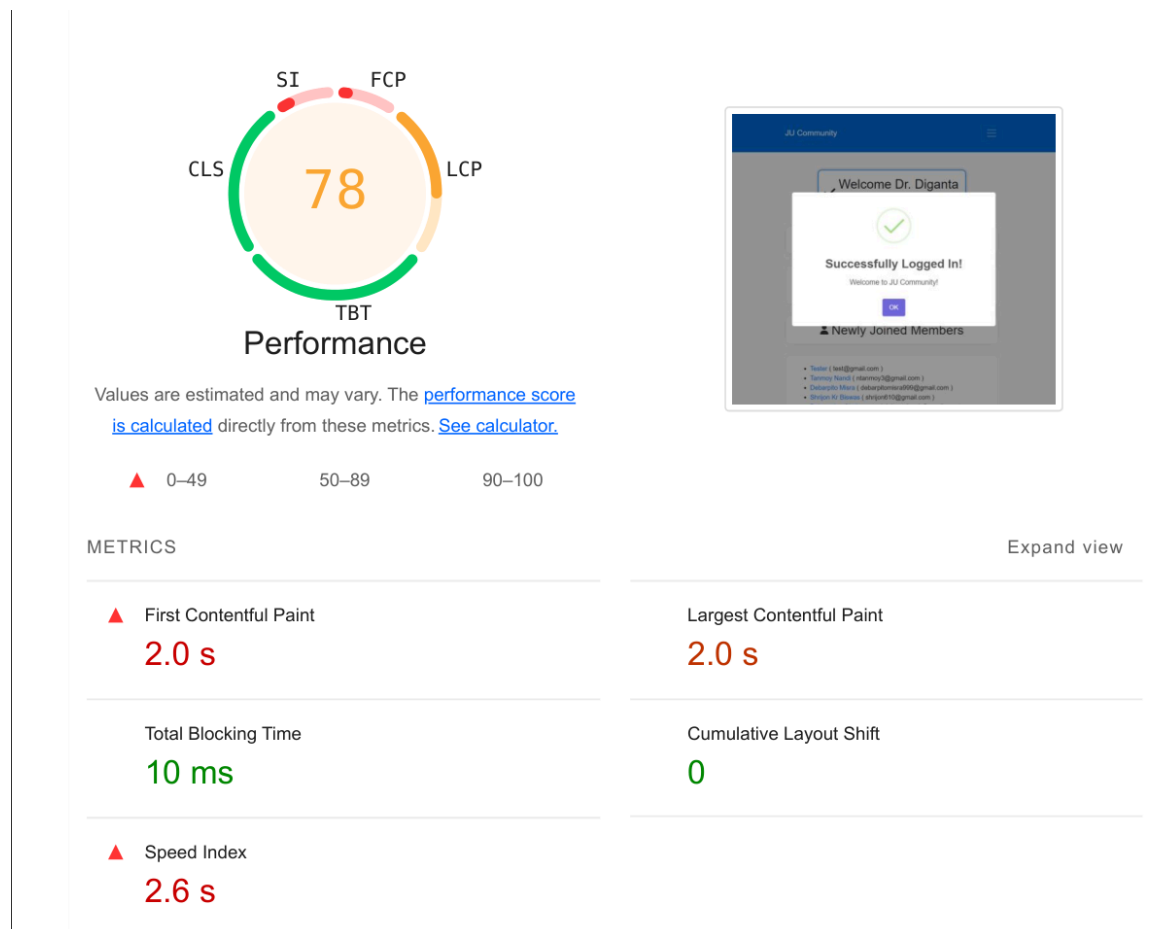
8. Regression Testing

- Purpose: Ensure that new changes do not adversely affect existing functionality.
- Outcome: Maintenance of platform stability and reliability by verifying that new updates do not introduce new bugs.

Results of the Testing Performed

- **Unit Testing:** Uncovered several bugs in individual functions, which were promptly fixed, leading to more robust modules.
- **Integration Testing:** Revealed issues in data flow between modules, which were addressed to ensure smooth interaction between different parts of the platform.
- **System Testing:** Confirmed that the platform met all specified requirements, with no major issues found during end-to-end testing.
- **User Acceptance Testing (UAT):** End-users reported the platform as user-friendly and intuitive, with minor usability issues that were subsequently resolved.
- **Performance Testing:** Identified performance bottlenecks under high load conditions. Optimizations were made to improve response times and handle higher traffic.
- **Security Testing:** Detected several vulnerabilities, including XSS and CSRF, which were fixed to enhance the platform's security.
- **Compatibility Testing:** Ensured the platform functioned consistently across multiple devices and browsers, with no significant compatibility issues.
- **Regression Testing:** Validated that new updates did not introduce new bugs, maintaining the platform's overall stability.

4.3 Performance Analysis



Performance Overview

The overall performance score of the website is 78, which falls into the "Needs Improvement" category (50-89). This score is derived from several key metrics that assess various aspects of the website's performance.

Key Metrics

1. First Contentful Paint (FCP)

- Time: 2.0 seconds
- Status: Needs Improvement
 - Description: FCP measures the time it takes for the first piece of content to be rendered on the screen. At 2.0 seconds, this is on the higher side, indicating the initial load time can be improved.

2. Largest Contentful Paint (LCP)

- Time: 2.0 seconds
- Status: Needs Improvement
- Description: LCP measures the time it takes for the largest piece of content to be visible. Similar to FCP, an LCP of 2.0 seconds suggests that users may experience a noticeable delay before the main content is fully loaded.

3. Total Blocking Time (TBT)

- Time: 10 milliseconds
- Status: Good
- Description: TBT quantifies the total time that the main thread was blocked and unable to respond to user input. A TBT of 10ms is excellent, indicating minimal delays in response times.

4. Speed Index (SI)

- Time: 2.6 seconds
- Status: Needs Improvement
- Description: The Speed Index measures how quickly the content is visually displayed during page load. With a score of 2.6 seconds, there is room for improvement to ensure faster visual completion.

5. Cumulative Layout Shift (CLS)

- Score: 0
- Status: Good
- Description: CLS measures the visual stability of the page. A score of 0 indicates that there are no unexpected layout shifts, ensuring a stable and pleasant user experience.

Summary

The website performs well in terms of Total Blocking Time and Cumulative Layout Shift, indicating good responsiveness and visual stability. However, there are areas that need improvement, particularly in the initial load times as indicated by the First Contentful Paint, Largest Contentful Paint, and Speed Index metrics.

Chapter 5

System Analysis

5.1 Advantages Of JU Community:

- 1. Enhance Research Collaboration:** Create a centralized platform where researchers can easily collaborate, share ideas, and work together on various projects.
- 2. Streamline Project Management:** Develop features that allow for efficient project group creation, member allocation, and progress tracking within individual research groups.
- 3. Facilitate Resource Sharing:** Integrate a system where admins can upload and manage resources like datasets and code, making them easily accessible to group members.
- 4. User Profile Management:** Enable users to set up comprehensive profiles, showcasing their research interests, publications, and other professional details.
- 5. Ensure Security and Privacy:** Implement robust security measures to protect user data and maintain privacy, including secure authentication and authorization protocols.
- 6. Scalability and Performance:** Develop the platform to handle high traffic efficiently, ensuring quick response times and minimal server requests.
- 7. Support Guest Access:** Allow guest users to view community projects and member profiles, increasing visibility and engagement.

By achieving these objectives, the JU Community Platform aims to foster a vibrant and collaborative research environment at Jadavpur University, promoting innovation and academic excellence.

5.2 Drawbacks of JU Community

- 1. Lack of Responsive Web Design:** The platform's current lack of responsive design poses a significant drawback, especially in today's mobile-centric world. Without optimization for smaller screens, accessing the platform on mobile devices may result in a subpar user experience, hindering accessibility and usability for a growing number of users.
- 2. Security Vulnerabilities:** Despite ongoing efforts to enhance security protocols, the platform may still be susceptible to emerging threats due to outdated or insufficient security measures. Without robust encryption and regular audits, user data and sensitive information remain at risk, potentially undermining trust in the platform.

3. Limited Interaction Features: The absence of commenting options restricts user engagement and interactivity within the community. Without the ability to comment on posts, researchers miss out on opportunities for dynamic discussions and collaboration, limiting the platform's effectiveness as a communication tool.

4. Underdeveloped Collaboration Tools: While the platform aims to facilitate collaboration, the lack of real-time chatting features hinders effective communication among researchers. Without immediate communication channels, coordinating projects and sharing ideas may be more cumbersome and less efficient, impeding collaboration efforts.

5. Suboptimal Performance: The platform's performance may suffer from slow response times and excessive server requests, leading to a frustrating user experience. Without optimization measures in place, such as minimizing page reloads and backend process optimization, users may encounter delays and inefficiencies while navigating the platform.

6. Complex User Interface: The platform's UI may lack intuitiveness and user-friendliness, making it challenging for users to navigate and utilize its features effectively. Without a streamlined and intuitive interface, users may struggle to find relevant information or perform necessary actions, reducing overall user satisfaction and engagement.

7. Absence of Event Management Functionality: Without dedicated features for managing and promoting research events, users may miss out on valuable opportunities to stay informed and engage with relevant academic activities. The lack of event management functionality limits the platform's utility as a comprehensive resource for academic networking and professional development.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The **JU Community Platform** has been meticulously designed to enhance research collaboration and resource sharing among the academic community at Jadavpur University. The platform's implementation addresses significant gaps in communication, project management, and resource accessibility. By providing a centralized hub for research activities, the platform facilitates greater visibility, efficiency, and collaboration, essential for fostering academic excellence and innovation.

Comprehensive testing has ensured that the platform is robust, reliable, and secure, establishing a solid foundation for its adoption and long-term success. The utilization of modern technologies, including Node.js for the backend, EJS for templating, and MySQL for database management, contributes to a scalable and efficient system. The platform's diverse features, from user registration and project management to resource sharing and profile management, collectively streamline the research process and promote interdisciplinary collaboration.

6.2 Future Work

To maintain the platform's relevance and effectiveness, several future enhancements are planned:

- 1. Responsive Web Design:** Ensuring the platform is fully responsive for smaller screens, enhancing accessibility and user experience on mobile devices.
- 2. Security Enhancements:** Ongoing updates and improvements to security protocols to protect against emerging threats, including advanced encryption and regular audits.
- 3. Commenting on Posts:** Introducing the ability for users to comment on posts, facilitating more interactive and dynamic discussions within the community.
- 4. Enhanced Collaboration Options:** Adding real-time chatting features to provide researchers with more collaboration tools, enabling immediate and effective communication.
- 5. Performance Optimization:** Improving response times and reducing server requests to ensure a faster, more efficient user experience. This includes minimizing page reloads and optimizing backend processes.
- 6. Intuitive UI:** Continuously refining the platform's design to make it more intuitive and user-friendly, ensuring that users can easily navigate and utilize its features.

7. Event Management: Adding functionality to manage and promote JU research events, allowing users to stay informed about relevant academic activities and opportunities.

8. Guest User Access: Allowing guest users (those not logged in) to view community projects and members, increasing the platform's visibility and engagement.

References

1. Jadavpur University Official Website
- [Jadavpur University](<http://www.jaduniv.edu.in/>)
2. Node.js Official Documentation
- [Node.js Documentation](<https://nodejs.org/en/docs/>)
3. Express.js Official Documentation
- [Express.js Documentation](<https://expressjs.com/>)
4. Bootstrap Official Documentation
- [Bootstrap Documentation](<https://getbootstrap.com/docs/5.0/getting-started/introduction/>)
5. EJS Official Documentation
- [EJS Documentation](<https://ejs.co/#docs>)
6. Aiven for MySQL
- [Aiven MySQL](<https://aiven.io/mysql>)
7. Vercel Documentation
- [Vercel Documentation](<https://vercel.com/docs>)
8. Cloudflare R2 Documentation
- [Cloudflare R2](<https://developers.cloudflare.com/r2/>)
9. Nodemailer Documentation
- [Nodemailer](<https://nodemailer.com/about/>)
10. OWASP Cheat Sheet Series
- [OWASP Cheat Sheets](<https://cheatsheetseries.owasp.org/>)

11. bcrypt Documentation

- [bcrypt Documentation](<https://www.npmjs.com/package/bcrypt>)

12. express-session Documentation

- [express-session](<https://www.npmjs.com/package/express-session>)

13. Multer Documentation

- [Multer](<https://www.npmjs.com/package/multer>)

14. sanitize-html Documentation

- [sanitize-html](<https://www.npmjs.com/package/sanitize-html>)