CHAPTER 1

INTRODUCTION

1.1 Problem Statement

1.2 Proposed Solution

1.3 My Roles

1.4 Deliverables

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

The college website currently faces several challenges that impact its overall effectiveness, user experience and functionality. Many colleges rely on static websites that lack interactivity and personalization for different user groups. These websites often require technical expertise to update content, leading to delays and outdated information.

The consequences might affect the current students. They might face difficulties accessing essential resources or updates. Faculty and staff might lose valuable time dealing with outdated information or inefficient communication channels. The college's overall image and reputation could be negatively impacted due to an outdated or poorly maintained website.

The website exhibits an outdated design, which may result in a less engaging and intuitive user experience and mobile responsiveness. The website may not be optimized for mobile devices, resulting in a suboptimal experience for users accessing it from smartphones or tablets. Images, text, or interactive elements do not scale appropriately on smaller screens.

Outdated information is present on various sections of the website, including course details, faculty listings, and event calendars. Lack of a streamlined content management process for regular updates and maintenance. Developing an intuitive, and scalable platform with effective customer support adds complexity.

Reports of potential security vulnerabilities, posing risks to user data and the integrity of the website. A need for an updated security infrastructure to protect against potential threats. Along with-it absence of a robust feedback mechanism for users to report issues, provide suggestions, or seek assistance. Limited channels for users to communicate problems to the website administrators.

1.2 Proposed Solution

This project proposal creates a dynamic college website. A well-designed website is essential for educational institutions to interact with the community in the digital age. In order to meet this need, our suggested solution combines contemporary features and technologies with those that are especially suited to the demands of a college environment. The primary aim of this project is to successfully transition the existing college website to the MERN stack ensuring accuracy, legality, and functionality.

Tailor the website's layout and navigation to guide users, including students, faculty, and visitors. Employ user-friendly interfaces to enhance ease of use and ensure that users can quickly locate the information. Implement a responsive design to ensure a seamless and consistent user experience across various devices.

Ensure quick and easy access to essential information such as course details, academic calendars, faculty profiles, and campus resources. Incorporate dynamic elements and interactive features to engage users, such as live event feeds, multimedia content, and real-time updates.

Portal has three types of users:

1. Student
2. Faculty
3. Admin

Student gets 24/7 access to information, which help them to learn at their own pace. Additionally, a dedicated feedback form and contact us feature facilitate two-way communication, ensuring any questions or concerns are promptly addressed.

Faculty has the following rights in the application:

- Login to the system using their “Email ID” and “Password”.
- They can leverage the website's features.
- **They can upload** material including course pdfs, syllabus, assignments, and other resources.
- **Faculty gain a centralized platform (dashboard)** to manage all their course materials, making them readily accessible to students.

Admin have all rights throughout the application.

- Login to the system using their admin “Email ID” and “Password”.
- Admin manages user access for faculty, staff, and potentially other user groups by providing “Email Id” and “Password”. Admin can assign different levels of access based on user roles.
- Admins can create, edit, publish, and delete various types of content, including:
  1. News and announcements
  2. Events
  3. Academic program information
  4. Faculty and staff profiles
  5. Gallery images
  6. Other website content

1.3 My Roles

1. Design of Web Pages: I was actively involved in designing the user interface (UI) and user experience (UX) of the web pages for the “EduHub" college website. This involved creating intuitive and responsive layouts that catered to the needs of both students and administrators.

2. Design of Database Schema: I played a key role in designing the database schema for the grievance redressal system. This included defining the database structure, relationships between entities, and optimizing data storage for efficient retrieval and management of grievances.

3. **Development of Backend:** I took charge of developing the backend logic and functionality of the grievance redressal system using the MERN (MongoDB, Express.js, React.js, Node.js) stack. This involved implementing server-side logic, APIs, and integrating with the frontend to ensure seamless data flow and functionality.

4.**Testing (Unit Testing and Integration Testing):** I conducted comprehensive testing of the system, including unit testing and integration testing. This involved writing and executing test cases to

ensure each component of the system functioned as expected and integrated smoothly with other modules.

Faculty Module:

- **Faculty Profile Management:** Responsible for creating and editing faculty profiles. This could involve gathering information like name, title, department, research interests, and contact details.
- **Content Creation and Curation:** Involved in creating content for faculty profiles, such as experience, achievements, seminar, qualification, etc. Worked with faculty members to create compelling content for their profiles, such as biographies, teaching philosophies, or research summaries. Assist faculty members in formatting and uploading content (e.g., publications list, awards).

Images Module:

- **Image Uploading and Management:** Involved in setting up a system for uploading, editing, and managing images used throughout the website. This could involve ensuring images are optimized for web display (size, format) and have proper captions and alt text.
- **Image Integration:** Responsible for integrating images into various sections of the website, such as gallery, faculty profiles, notices descriptions, or event announcements.
- **Image Gallery:** Responsible for designing a picture gallery for all the major college related activities.

1.4 Deliverables

The list of deliverables is provided in Table 1.1

Table 1.1: List of deliverables

S. No.	Phase	Deliverables
1.	Requirement Analysis	Use Case
2.	Design	DFD Diagrams DFD Level 0 DFD Level 1 E-R Diagram Activity Diagram
3.	Unit Testing	Test Results

CHAPTER 2

PROJECT DESCRIPTION

CHAPTER-2: PROJECT DESCRIPTION

2.1 System Interfaces

The overall System has the following interfaces:

- **Login Page for Admin/Faculty:**
  - This interface allows faculty/admin to login to the system.
  - Provides Form based authentication.
  - Performs the authentication and authorization process.

- **Home Page:**
  - This is the website's home page covering all the features.
  - This page is Freely Accessible and available to public without any barriers.
  - This section displays all the tabs like Notice Board, Events, Gallery, Faulty Information, Contact Us etc.

- **Dashboard:**
  - This page is only for Admin Users having general layout, and specific features can be customized based on the college's needs.
  - This section allows the admin to manage faculty's information.
  - Provides quick access to different admin sections (e.g., User Management, Content Management, Settings).
  - Displays interactive modules for content updates and additions.
  - Managing user accounts, including adding, deleting users, and assigning roles.
  - This is page acts as a central hub for managing website content and functionalities.

- **Contact Us page:**
    - This page offers a user-friendly way for visitors to submit inquiries directly through the website.
    - The sole purpose of the page is to gather feedback or inquiries from users.
    - Users don't need to navigate through multiple options like phone numbers or email addresses, making it faster to submit feedback.

- **Faculty Information Page:**
    - Display the faculty member's name and their academic title (e.g., Professor, Associate Professor).
    - Indicate the qualifications and working experience of the faculty member.
    - Include an email address for students to reach the faculty member directly.
    - A professional photo that can help students identify the faculty member.
    - Highlighting any awards or recognition received by the faculty member can showcase their achievements.
    - Briefly highlight the faculty member's research interests and areas of expertise.
    - List recent publications or provide a link to a faculty profile on a platform where their publications are showcased.

- **Notices Page:**
    - This page is reached in case any user clicks on notice board.
    - For a large number of notices, consider offering filtering options by category to help users refine their search.

- Implement a system for categorizing notices (e.g., Academics, Activities, Examination, Placement, etc). This helps users quickly find relevant information.

- **Course Material Page:**
  - Display a clear list of all offered courses, potentially categorized by department or semester.
  - Allow users to easily navigate to specific course material pages. Consider options like dropdown menu of semesters.
  - Each course page clearly displays the subject name, syllabus (downloadable PDFs), assignments, examination papers and learning objectives.

- **Gallery page:**
  - Utilize a visually appealing layout that showcases the images in an organized and engaging way. Consider a grid layout depending on the type and volume of images.
  - Include clear and concise names for each image.
  - This page helps to visually communicates the essence of your institution and attracts prospective students, faculty, and the wider community.

- **Other Static Pages:**
  - **About Us:** Briefly highlight the college's history, mission, vision, and core values. Include accreditation information and affiliations.
  - **Director's Message:** Feature a welcome message from the director, outlining the institution's goals and unique offerings.
  - **Training & Placement:** Showcase career guidance services, placement stats, and success stories of alumni. Include

information on recruiting companies, internship opportunities and skill development programs.

- **Campus:** Provide a virtual tour with images and video showcasing the college's facilities, classrooms, libraries, labs, common areas, and student life.

- **Founder's Message:** A notable founder of college includes a message or story highlighting their vision and inspiration.

- **Student Council:** Introduce the student council, its members, and their roles in representing student interests and organizing events.

- **Hostel:** Provide details on hostel life, including accommodation types, facilities, amenities, rules and regulations, and the application process.

- **Library:** Showcase the library's collection, online resources, research assistance services, and operating hours.

- **Additional Pages:** Consider adding pages for specific programs, departments, student clubs, alumni relations, scholarships, admissions process, FAQs, and contact information.

2.2 System Specification

2.2.1 Hardware Requirements:

- Processor: Pentium-4 class processor or Above
- RAM: 2GB of RAM or Above
- Operating System: Windows XP or Above

2.2.2 Software Requirements:

- Operating System: Windows, Mac, Linux
- Database Server: MongoDB
- Client: Microsoft Internet Explorer or any other web browser

2.2.3 Technologies Used:

- Front-End: HTML, CSS, React, Bootstrap
- Back-End: Node.js

2.3 Methodology and Tools used:

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and
important stakeholders.

The agile software development emphasizes on four core values:

- Individual and team interactions over processes and tools
- Working software over comprehensive documentation

- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile methodology, a dynamic software development approach, prioritizes iterative progress and collaboration with customers. Projects are broken down into small sprints, typically lasting one to four weeks, fostering adaptability within cross-functional teams. Continuous feedback from stakeholders ensures alignment with evolving requirements.

Agile promotes transparency and early delivery of functional software. It prioritizes individuals and interactions, valuing customer satisfaction and adaptability over strict adherence to plans.

2.3.1 Requirement Phase

This requirement phase provides a foundational understanding of the college website's goals, target audience, functionalities, and technical considerations. It details the functionalities, target audience needs, and technical considerations for the project.

- **Project Goals and Objective**

    1. Enhance communication and information dissemination for students, faculty, staff, and the wider community.

    2. Establish a dynamic and user-friendly website to serve as the college's primary online presence.

    3. Improve accessibility to college resources, programs, and services.

    4. Streamline content management and website administration.

    5. Foster a positive and engaging online experience for all users.

    6. Showcase the college's unique offerings and achievements to attract prospective students and faculty.

- **Target Audience Analysis**

    1. **Primary Audience:**
        a. Prospective Students
        b. Current Students
        c. Faculty and Staff
        d. Parents and Guardians
        e. Alumni

    2. **Secondary Audience:**
        a. The General Public
        b. Employers
        c. External Collaborators

    Understanding the needs of each audience segment is crucial for tailoring content and functionalities to their specific interests and requirements.

- **Functional Requirements**

    1. **Dashboard:** A user-friendly dashboard allowing authorized personnel to easily create, edit, and publish website content.

    2. **Information Architecture:** A clear and logical website structure that facilitates intuitive navigation for users.

    3. **Content Pages:**

        a. About Us
        b. Admissions
        c. Academics (Programs, Courses)
        d. Faculty and Staff Information
        e. Student Life (Events, Clubs, Activities)
        f. Training & Placement
        g. Campus Life (Virtual Tour, Facilities)
        h. Library Resources
        i. News & Announcements
        j. Notices & Deadlines

k. Contact Us
l. Additional pages specific to departments or programs

4. **User Interaction Features:**
    a. Contact Us Form
    b. Calendar Integration (for events, deadlines)
    c. Social Media Sharing Buttons
    d. Online Applications

5. **Mobile-Responsive Design:** The website should adapt seamlessly to various screen sizes for optimal viewing on desktops, tablets, and mobile devices.

- **Non-Functional Requirements**

1. **Performance:** The website should load quickly and function efficiently even with high traffic volumes.

2. **Security:** Robust security measures are essential to protect user data and website content from unauthorized access.

3. **Scalability:** The website's design and technology stack should accommodate future growth in content and user traffic.

4. **Dashboard User Interface:** This should be intuitive and user-friendly for authorized personnel to manage website content without requiring extensive technical expertise.

- **Technology Stack**

1. **Front-End:** Consider mentioning the proposed front-end technologies like HTML, CSS, JavaScript frameworks (React.js).

2. **Back-End:** Briefly mention the proposed back-end technologies like Node.js, and database (MongoDB) if the MERN stack is chosen.

3. Technology stack selection can be finalized during the design phase based on project requirements and development expertise.

2.3.2 Design Phase

The requirements were frozen and finalized before the design phase began. The design phase looks for different system entities and the behavior that go along with them, as well as potential system interfaces. They created the system's ER diagram and data flow diagram. This step also included identifying the database tables and their relationship.

2.3.3 Development Phase

It outlines the chosen approach, key stages, and considerations specific to the MERN stack (MongoDB, Express.js, React.js, Node.js).

During the development phase, a structured approach was adopted using a 3-tier architecture with a bottom-up strategy. The development process involved defining distinct layers and utilizing specific tools to implement the system functionalities effectively.

1. **Data Access Layer:**
    a. A data access layer was established to manage connections to the MongoDB database.

    b. A database context class was defined within this layer to handle the flow of information between the application and the MongoDB database.

    c. Frameworks and libraries were utilized to streamline data operations and ensure efficient data retrieval and manipulation.

2. **Business Logic Layer:**

    a. The business logic layer served as an intermediary between the presentation layer (user interface) and the data access layer.

    b. All business rules, validation logic, and modification operations were implemented within this layer.

c. The business logic layer encapsulated core functionalities of "EduHub" college website, ensuring separation of concerns and modularity.

3. **Presentation Layer:**

   a. The presentation layer defined the user interface through which users interacted with the Grievance Redressal System.

   b. Classes and components within this layer were responsible for presenting data retrieved from the business logic layer in the required formats (e.g., web pages, forms).

   c. User interactions and input validations were handled within the presentation layer to provide a seamless user experience.

- **Development Methodology**

  The development process will follow an iterative approach, potentially incorporating the following stages:

  1. **Front-End Development:**

     a. **User Interface (UI) Design and Development:** Utilize React.js and related libraries to design and build visually appealing and interactive user interface components.

     b. **Interactive Features:** Implement functionalities like search bars, form submissions, and dynamic content updates using JavaScript and React functionalities.

     c. **API Integration:** Establish seamless communication between the front-end and back-end by integrating with the APIs developed for data fetching and manipulation.

  2. **Back-End Development:**

     a. **API Development:** Employ the Express.js framework to develop robust APIs that handle user requests and responses effectively.

b. **Database Management:** Utilize MongoDB to connect with the database, ensuring efficient storage, retrieval, and manipulation of various content types (text, images, documents).

c. **Authentication and Authorization (Optional):** If required, implement secure measures for user login and access control based on roles (e.g., admin, student).

d. **Third-Party Integration (Optional):** Integrate with external services like email notification tools to enhance functionalities if needed.

3. **Dashboard Development:**

a. Design and develop a user-friendly interface within the website for authorized personnel to manage content (create, edit, publish).

b. Leverage MERN stack-specific libraries or frameworks designed for functionalities.

- **Development Tools Used**

1. **Microsoft Visual Studio 2017 Express for Web:**

a. Visual Studio was used as the primary integrated development environment (IDE) for building and deploying web applications.

b. The IDE provided robust features for coding and debugging the website using MERN technologies.

2. **MongoDB Server for Database Management:**

a. MongoDB was employed as the NoSQL database for storing and managing data related to grievances, user profiles, and system configurations.

b. MongoDB facilitated efficient data storage, retrieval, and manipulation

c. operations required by the “EduHub” college website, offering flexibility and scalability for handling diverse data types.

- **Development Approach:**

The development process followed a bottom-up strategy, starting from the foundational data access layer and progressively building the business logic and presentation layers. Each layer was designed to handle specific responsibilities, promoting modular development and code reusability. Frameworks and tools were leveraged to streamline development tasks and ensure adherence to best practices in software engineering.

2.3.4 Implementation Phase

The implementation process outlines the execution phase of the college website project, focusing on the chosen development methodology, key stages, configuring server environments, establishing database connections, and considerations specific to the MERN stack (MongoDB, Express.js, React.js, Node.js).

The development process will follow an **iterative approach**, promoting continuous improvement and feedback integration throughout the project lifecycle. Here's a breakdown of the anticipated stages:

- **Front-End Development:**

a. UI Design and Development:

i. Utilize React.js and relevant libraries (e.g., Material UI, Bootstrap) to design and build visually appealing and interactive user interface components.
ii. Prioritize clear navigation, intuitive design principles, and responsive layouts for optimal user experience across various devices (desktops, tablets, mobiles).

b. Interactive Features: Implement functionalities like search bars, form submissions, dynamic content updates, and user interactions using JavaScript and React functionalities.

c. **API Integration:** Establish seamless communication between the front-end and back-end by integrating with the APIs developed for data fetching and manipulation. This ensures dynamic and responsive content updates on the user interface.

- **Back-End Development:**

a. **API Development:** Employ the Express.js framework to develop robust APIs that efficiently handle user requests and responses.

b. **Data Modelling:** Design an efficient data model within MongoDB to store and organize website content (text, images, documents) in a structured and scalable manner. Consider data relationships, ease of querying, and future expansion needs.

c. **Authentication and Authorization:** If required, implement secure measures for user login and access control based on roles (e.g., admin, student, faculty) using secure authentication mechanisms (e.g., password hashing with salting).

d. **Third-Party Integration:** Integrate with external services like email notification tools or payment gateways to enhance functionalities if needed. Consider security implications and implement appropriate safeguards.

- **Dashboard Development:**

a. **User-Friendly Interface:** Design and develop a user-friendly interface within the website for authorized personnel to manage content (create, edit, publish).

b. **Role-Based Access Control:** Implement role-based access control to ensure appropriate user permissions for content management tasks (e.g., editors can edit content, admins can publish).

- **Testing:**

a. **API Testing:** Conduct rigorous testing of all APIs to ensure they function correctly and securely, handling various data requests and edge cases (e.g., invalid inputs, empty data sets).

b. **Integration Testing:** Meticulously test the integration between front-end and back-end components, verifying seamless data exchange, user interactions, and overall website functionality.

The MERN stack will be the foundation for website development, leveraging the strengths of each component:

- **Front-End (React.js):** Responsible for rendering the dynamic user interface, handling user interactions, and communicating with back-end APIs to retrieve and manipulate data.
- **Back-End (Node.js & Express.js):** Acts as the server-side core, handling user requests, routing them to appropriate functions, and interacting with the database.
- **Database (MongoDB):** Stores website data in a flexible and scalable NoSQL format, allowing for efficient storage of diverse content types (text, images, documents).
- **API Communication:** The front-end will interact with the back-end through well-defined APIs that facilitate data transfer between the user interface and the database.

Benefits of MERN Stack for College Website:

- **Dynamic and Interactive UI:** React.js enables building a modern and engaging user interface with dynamic features.
- **Scalability and Performance:** The MERN stack can efficiently handle growing amounts of data and user traffic.
- **Flexibility:** MongoDB's NoSQL structure allows for storing diverse content types (text, images, documents) with ease.

2.3.5 Testing Phase

Testing plays a critical role in the success of the college website project. By conducting rigorous testing throughout the development process, we can ensure:

- **High-Quality Website:** Testing helps identify and rectify bugs, ensuring a functional and reliable website.

- **Enhanced User Experience:** Testing identifies usability issues and ensures the website is user-friendly and caters to different user groups.
- **Improved Security:** Testing helps identify and mitigate security vulnerabilities, protecting user data and website integrity.
- **Scalability and Performance:** Testing helps ensure the website can handle future growth in traffic and maintain optimal performance.

By following a comprehensive testing strategy, we aim to deliver a high-quality, user-friendly, and secure college website that meets the needs of the college community.

The testing process will adhere to a multi-layered approach, encompassing various testing levels to identify and rectify any potential issues before deployment.

1. **Unit Testing:**
    a. Focuses on testing individual components of the code (front-end and back-end) in isolation.
    b. Ensures each component functions as intended and produces the expected output for various input scenarios.

2. **Integration Testing:**
    a. Tests the seamless integration and communication between front-end and back-end components.
    b. Verifies data exchange between APIs and the user interface, ensuring smooth user interactions and data manipulation.

3. **API Testing:**
    a. Conducted to ensure all APIs function correctly, handle diverse data requests, and respond appropriately.
    b. Tests include positive and negative test cases, covering valid user inputs, invalid inputs, and edge cases (empty data sets, unexpected errors).
    c. Utilize automated testing tools like Thunder client for efficient API testing.

2.4 Constraints

The constraints will influence the development process, technology choices, and project timelines.

1. **Project Deadlines:** Clearly defined project deadlines will influence the scope of features and functionalities that can be realistically implemented within the allotted timeframe.

2. **Development Team Availability:** The availability and capacity of the development team will determine the pace of development and the need for potential resource allocation adjustments.

3. **Target Audience:** The website should cater to a diverse range of users (students, faculty, staff, prospective students) with varying technical skills and access to technology. This may influence the design complexity and the need for alternative content access options.

4. **Internet Connectivity:** The website should be accessible to users with varying internet connection speeds. Techniques like image optimization and responsive design can help ensure website usability even on slower connections.

- **Strategies for Mitigating Constraints:**

1. **Prioritization:** Clearly define project priorities and focus on core functionalities within the given timeframe and budget constraints.

2. **Phased Development:** Consider a phased development approach where core features are implemented first, with additional functionalities added later based on available resources.

3. **Usability Testing:** Conduct user testing to ensure the website is accessible and user-friendly for the target audience.

By acknowledging and proactively managing these constraints, the development team can deliver a high-quality college website that meets project goals while operating within the defined limitations. Display of content varies from Browser to Browser it is recommended to use Google

Chrome or similar browsers to view the site perfectly. Testing process is carried out on a limited number of hosts.

2.5 Assumptions and Dependencies

This section outlines the key assumptions made during the planning phase of the college website project and the dependencies that need to be addressed for successful development.

1. Assumptions

- **Content Availability:** We assume that the college will provide all necessary content (text, images) in a digital format for website population.
- **User Roles and Permissions:** A clear definition of user roles (admin, student, faculty) and their corresponding access permissions for content management will be established.
- **Technical Expertise:** The development team possesses the necessary skills and experience to work with the chosen MERN stack (MongoDB, Express.js, React.js, Node.js).

2. Dependencies

- **Dashboard:** The successful functionality of the website depends on the development and implementation of Dashboard within the website. This allows authorized personnel to create and publish content.
- **Server Configuration:** The website's functionality depends on proper server configuration to ensure secure data storage, efficient database communication, and optimal website performance.
- **Ongoing Maintenance:** The website will require ongoing maintenance to address potential bugs, security updates, and content management needs.

2.6 User Characteristics

Role-based security on the “EduHub” college website designates several user roles with corresponding rights and privileges depending on their duties within the system. The system's defined features are used to identify the following user roles and the traits that go along with them. Here's a breakdown of the key user characteristics for the core user groups of a college website:

Students

- Access course materials (syllabi, assignments, presentations).
- View grades and academic progress.
- Communicate with faculty and advisors.
- Find campus resources (library, career services, student clubs).
- Access online learning platforms and tools.

Faculty

- Enter the system by logging in with their "Password" and "EmailID."
- Able to utilize the capabilities of the website.
- Has the ability to upload resources such as assignments, course syllabuses, and PDFs.
- Instructors are given a single platform (dashboard) to handle all of their course materials, giving students easy access to them.
- View course material and other content.

Administrators

- Manage website content (news, announcements, events)
- Create and delete user accounts for different user groups (students, faculty, staff)
- Manage security settings and user access permissions
- Integrate with external systems (e.g., admissions forms, alumni databases)
- Secure access to an administrative dashboard with comprehensive management tools
- User-friendly interface for content creation and editing within the CMS
- User management tools for adding, and deleting user accounts.

CHAPTER 3

FUNCTIONALITY

3.1 Logical Database Design

3.1.1 Table Structures

3.2 Input and Output Design

3.3 Use case Description

CHAPTER-3: FUNCTIONALITY

3.1 Local Database Design

3.1.1 Table Structures:

The structure of various database tables designed for “EduHub” college website is described in Table 3.1 to 3.7.

Table 3.1: Structure of User Database table

S. No.	Name of Column	Data Type	Description
1.	userId	ObjectId	Primary Key
2.	name	String	Name of the User
3.	email	String	Email of the User
4.	password	String	Password of the User
5.	isAdmin	Boolean	Specifies user is admin or not

Table 3.2: Structure of Contact Us Database table

S. No.	Name of Column	Data Type	Description
1.	feedbackId	ObjectId	Primary Key
2.	name	String	Name of the visitor
3.	email	String	Email of the visitor
4.	subject	String	Subject line for the feedback
5.	message	String	Feedback or message written by the visitor

Table 3.3: Structure of Course Database table

S. No.	Name of Column	Data Type	Description
1.	courseId	ObjectId	Primary Key
2.	semester	String	Academic session
3.	courseType	String	Category of courses, MCA or BAJMC
4.	subName	String	Name of the subject
5.	syllabus	String	Syllabus of the subject
6.	courseMaterial	String	Study Materials in the form of pdfs.
7.	intppr	String	Question Paper of internal exams
8.	extppr	String	Question Paper of external exams
9.	labFile	String	Lab Manual File
10.	assignment	String	Assignments
11.	teacherName	String	Name of the assigned faculty

Table 3.4: Structure of Images Database table

S. No.	Name of Column	Data Type	Description
1.	imgId	ObjectId	Primary Key
2.	name	String	Name of the Image
3.	description	String	Description of the Image
4.	imageType	String	Image Type like events, gallery and slider image
5.	photo	String	Url of the Image

Table 3.5: Structure of Notice Database table

S. No.	Name of Column	Data Type	Description
1.	noticeId	ObjectId	Primary Key
2.	Name	String	Name of the notice
3.	description	String	Description of the notice
4.	noticeType	String	Category of notices like admission, placements, examination etc.
5.	Photo	String	Url of the picture
6.	url	String	Link to external websites for related information.
7.	file	String	File related to notices
8.	filename	String	Name of the file

Table 3.6: Structure of Faculty Database table

S. No.	Name of Column	Data Type	Description
1.	facultyId	ObjectId	Primary Key
2.	name	String	Name of the faculty
3.	designation	String	Designation of the faculty
4.	email	String	Email of the faculty
5.	facultyType	String	Teaching or Support Staff
6.	qualification	String	Degrees of the faculty
7.	areaofexpertise	String	Specialty in any particular field of the faculty
8.	experience	String	Experience of the faculty
9.	achievement	String	Achievements of the faculty
10.	seminar	String	Participated in or hosted the seminars
11.	photo	String	Url of the picture

Table 3.7: Structure of Event Database table

S. No.	Name of Column	Data Type	Description
1.	eventid	ObjectId	Primary Key
2.	name	String	Name of the event
3.	description	String	Description of the event
4.	date	String	Date of the scheduled event
5.	photo	String	Url of the picture

3.2 Input Output Design

The Input Output Design of various Screens designed for "EduHub" college website is described in Table 3.8 to 3.16.

Table 3.8: Input Output Design of Login Screen

Purpose	Enables Faculty or Admin to log into the Dashboard
Description of field(s)	• **Email Id:** User enters his/her login name to log into the System. • **Password:** User enters Password. • **Login Button:** Logs the user in or displays an error on mis-match of email id and password.
Validation Checks	• Email Id should be a registered Email Id. • Password should match the entered Email Id in the database.

Table 3.9: Input Output Design of Home Screen

Purpose	Enables visitor to view content of the website.
Description of field(s)	• All the tabs are displayed here. • By clicking on each tab, we can navigate further. • The tabs present are Notice Board, Events, Gallery, Faculty Information, Contact Us etc.
Validation Checks	• This is the website's home page. • Student cannot edit but view the course material and other college related things.

Table 3.10: Input Output Design of Dashboard Screen

Purpose	Displays interactive modules for content updates and additions.
Description of field(s)	• Provide a space for adding and updating the faculty information. • Management of images, events, notices and feedback form. • Upload and organize the course materials. • Grant access to users by providing emailID and password.
Validation Checks	• All fields are mandatory with an asterisk symbol. • Password should not be less than 3 characters. • Name length should not exceed 100 characters. • Course material should only be in pdf format.

Table 3.11: Input Output Design of Contact Us Screen

Purpose	Enables user to give the feedback.
Description of field(s)	• **Name:** User enters his/her name. • **Email:** User enters his/her emailId. • **Subject:** User enters the heading of the feedback. • **Message:** User writes his views. • **Submit Button:** After providing the details user clicks on submit button.
Validation Checks	• All fields are mandatory.

Table 3.12: Input Output Design of Faculty Information Screen

Purpose	Enables visitor to view all the faculty member details.
Description of field(s)	• Displays the faculty member's name, photo, email, awards, their designation and qualification. • Provides working experience and area of expertise of the faculty member.
Validation Checks	This is non editable, profile viewing page.

Table 3.13: Input Output Design of Notice Screen

Purpose	Used to display all types of notices.
Description of field(s)	• Acts as a notice board providing important details. • Notices are displayed specific to their categories.
Validation Checks	Notice category, name and description are mandatory.

Table 3.14: Input Output Design Of Course Material Screen

Purpose	Provides semester wise material to the users.
Description of field(s)	• Course page clearly displays the subject name, syllabus (downloadable PDF), assignments, examination papers and learning objectives. • Consider options like dropdown menu of semesters.
Validation Checks	From this page user can download and view the material.

Table 3.15: Input Output Design of Gallery Screen

Purpose	To show the images of all the college related activities.
Description of field(s)	It acts like gallery page displaying the name and image.
Validation Checks	Image size should lie between 10Kb and 3MB.

Table 3.16: Input Output Design of Static Pages

Purpose	Get to Know about the college, these pages make the website as a resource for prospective students.
Description of field(s)	The pages like About Us, Director's Message, Training & Placement, Campus, Hostel, Library etc. are present to establish the college's image and reputation.
Validation Checks	These pages act as comprehensive guide for all things related to the college.

3.3 Use Case Description

The Use Case Description for “EduHub” is described in Table 3.17 to 3.24.

Table 3.17: Use Case Description of Login Process

Purpose	• This use case defines Login process of Admin and user(faculty).
Actors	• Faculty • Admin
Preconditions	• User should have emailID and password given by the admin so as to change its password further. • In case of Admin Login, the User should be registered under Admin Role.
Post Conditions	• Dashboard is displayed depending on the successful login depending on the role of the user.
Basic Flow	• On Failure Error Message is displayed. • The system requests the actor to enter his/her emailId and Password. The role can be any one of the admin or the faculty. • The system validates the entered emailId and password and Logs in the actor into the dashboard portal.
Alternate Flows	• User is shown an error message on invalid emailId or Password.

Table 3.18: Use Case Description of Home Screen Process

Purpose	• Enables actors to access the content.
Actors	• Student • Faculty • Admin
Preconditions	• Website content should be displayed properly without any inconsistency or ambiguity.
Post Conditions	• Content is displayed to all the visitors.
Basic Flow	• All the tabs and major event resides on the home screen. • Actor clicks on tabs according to his/her requirement.
Alternate Flows	• Content is available for every visitor at their own ease of access.

Table 3.19: Use Case Description of Dashboard Process

Purpose	• Enables admin and faculty to view and add interactive modules for content.
Actors	• Admin • Faculty
Preconditions	• User should be logged into the system
Post Conditions	• All the dashboard features are displayed.
Basic Flow	• Actor clicks on notices, events, faculty information for addition and updating of the content. • All the forms/fields specific to each feature are displayed. • Content could be updated as per the changes done.
Alternate Flows	• Invalid details are entered and an appropriate error message is displayed.

Table 3.20: Use Case Description of Contact Us Process

Purpose	• Enables to give feedback/responses.
Actors	• Student • Faculty
Preconditions	• Contact Us form should be displayed with all necessary fields.
Post Conditions	• Feedback is successfully submitted after filling all the fields.
Basic Flow	• User clicks on the Contact us tab. • All the details are filled and thank you note is displayed after submission.
Alternate Flows	• Error message if any important field is missing.

Table 3.21: Use Case Description of Faculty Information Process

Purpose	• Enables visitor to view all the faculty member details.
Actors	• Admin
Preconditions	• User should be logged into the dashboard.
Post Conditions	• Faculty is created if all the details are filled.
Basic Flow	• Actor will click on Faculty Information and add the relevant field like member's name, photo, email, awards, their designation, qualification etc. It will create the faculty's profile.
Alternate Flows	• Error message if any important field is missing.

Table 3.22: Use Case Description of Notice Process

Purpose	• Used to create all the notices related to the college.
Actors	• Student • Faculty
Preconditions	• Notices should be uploaded properly so that user can view them.
Post Conditions	• The notices should be downloadable in pdf format.
Basic Flow	• The actor clicks on Notices tab. • All the notices are displayed according to the category like examinations, placements, etc.
Alternate Flows	• None

Table 3.23: Use Case Description Of Course Material Process

Purpose	• Gives semester wise material to the users.
Actors	• Student
Preconditions	• Semester and course (MCA or BAJMC) should be present for selection.
Post Conditions	• Material is displayed according to the requirement.
Basic Flow	• User clicks on the Course Material Tab. • User selects its semester and the course. • Course page clearly displays the subject name, syllabus (downloadable PDF), assignments, examination papers and learning objectives.
Alternate Flows	• None

Table 3.24: Use Case Description of Gallery Process

Purpose	• To show the images related to the college campus.
Actors	• Student • Faculty
Preconditions	• Image size should follow the range.
Post Conditions	• Picture Gallery is displayed.
Basic Flow	• User clicks on the Gallery tab and view the image and its name.
Alternate Flows	• None

CHAPTER 4

TESTING

4.1 Test Activities

4.2 Unit Testing

4.3 System Testing

4.3.1 Functional Testing

4.4 Test Reports and Debugging

CHAPTER-4: TESTING

4.1 Testing Activities

Testing is a process rather than a single activity. Testing must be planned, and it requires discipline to act upon it. The quality and effectiveness of software testing are primarily determined by the quality of the test processes used.

The activities of testing can be divided into the following basic steps:

- Planning and Control
- Analysis and Design
- Implementation and Execution
- Evaluating exit criteria and Reporting
- Test Closure activities

Test planning involves producing a document that describes an overall approach and test objectives. It involves reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases and designing the test environment. Completion or exit criteria must be specified so that we know when testing (at any stage) is complete.

Purpose:

- To determine the scope and risks and identify the objectives of testing.
- To determine the required test resources like people, test environments etc.
- To schedule test analysis and design tasks, test implementation, execution.

Control is the activity of comparing actual progress against the plan, and reporting the status, including deviations from the plan. It involves taking actions necessary to meet the mission and objectives of the project.

Test analysis and Test Design has the following major tasks:

- To review the test basis. The test basis is the information on which test cases are based, such as requirements, design specifications, product risk analysis, architecture and interfaces.
- To identify test conditions.
- To design the tests.
- To design the test environment set-up and identify the required infrastructure and tools.

Test execution involves running the specified test on a computer system either manually or by using an automated test tool. It is a Fundamental Test Process in which actual work is done. Test implementation has the following major task:

- To develop and prioritize test cases by using techniques and create test data for those tests.
- To create test suites from the test cases for efficient test execution.
- Test suite is a collection of test cases that are used to test a software program
- To re-execute the tests that previously failed in order to confirm a fix.
- To log the outcome of the test execution. A test log is the status of the test case (pass/fail).
- To compare actual results with expected result.

Evaluating exit criteria is a process defining when to stop testing. It depends on coverage of code, functionality or risk. Basically, it also depends on business risk, cost and time and vary from project to project. Exit criteria come into picture, when:

- Maximum test cases are executed with certain pass percentage
- Bug rate falls below certain level
- When we achieve the deadlines

Test closure activities are done when software is ready to be delivered. The testing can be closed for the other reasons also like:

- When a project is cancelled
- When some target is achieved
- When a maintenance release or update is done

4.2 Unit Testing

It is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class.

- **Methodology Used**

  Unit testing was carried out at the developer environment only. Manual testing was done. The developers review their code to check whether their respective units under tests behave as expected.

- **Tools Used**

  No tools were used to do the Testing as manual testing was carried out for Unit Testing of all the modules. As Manual Testing doesn't require any tools, so Tools are not applicable.

4.3 System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system, the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective. System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and non-functional testing.

4.3.1 Functional Testing

- **Methodology Used**
  Under this the whole system was tested under the development team. Basically, all functionalities as per requirements are tested here.

- **Tools Used**
  No tools were used to do the Testing as manual testing was carried out for Functional Testing. As Manual Testing doesn't require any tools, so Tools are not applicable.

4.4 Test Reports and Debugging

Table 4.1: Functional Testing Report

Id	Test Case Description	Test Case Input	Test Results		Status	Corrective Measure
			Expected	Actual		
1.	Admin Login	Email Id and password	Successfully logged in	Successfully logged in	Pass	None
2.	Notice Details	All the fields are required	Data inserted successfully	Data inserted successfully	Pass	None
3.	Staff Details	All the fields are required	Data inserted successfully	Data inserted successfully	Pass	None
4.	Images	All the fields are required	Data inserted successfully	Data inserted successfully	Pass	None
5.	Feedback	Non empty string of message	Feedback is successfully submitted	Feedback is successfully submitted	Pass	None

Id	Test Case Description	Test Case Input	Test Results		Status	Corrective Measure
			Expected	Actual		
6.	Events	All the fields are required	Events added	Events added	Pass	None
7.	Notice Category	All the fields are required	Organize notices into distinct categories	Organize notices into distinct categories	Pass	None
8.	Course Details	All the fields are required	Data inserted successfully	Data inserted successfully	Pass	None
9.	Faculty Login	Usernam e and password	Successfully logged in	Successfully logged in	Pass	None
10.	Faculty Details	All the fields are required	Data inserted successfully	Data inserted successfully	Pass	None

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

5.2 Limitations of the System

5.3 Future Scope for Modification

5.1 Conclusion

In conclusion, the development of this college website project has been a comprehensive endeavour aimed at enhancing the digital presence and functionality of our institution. Through planning, design, and implementation, we are on the way to create a user-friendly platform that serves as a centralized hub for students, faculty, staff, and prospective members of our community.
Through collaborative efforts and meticulous attention to detail, we are shaping a platform that will serve as a dynamic gateway to our college's offerings, resources, and vibrant community.

We embraced emerging technologies and best practices to create a user-centric website that reflects our college's ethics and values. From designing intuitive navigation to integrating interactive features, every aspect of the website has been thoughtfully crafted to enhance the user experience and streamline access to information.

Throughout the project, we adhered to best practices in web development, including responsive design, ensuring the website is accessible across various devices. We incorporated feedback from multiple users to refine the content and usability, aiming to meet the diverse needs of our audience in future also. Key features such as a dynamic content updates, an interactive campus map, and comprehensive academic and admin information were prioritized to enhance user experience and engagement.

Moving forward, continuous evaluation and updates will be essential to maintain the website's effectiveness and relevance. We recommend regular feedback collection, ongoing maintenance, and timely updates to content and features to adapt to the changing needs of users and technological advancements.

5.2 Limitations of the System

Here's a breakdown of some key limitations to consider when building a college website with MERN:

1. Scalability:

- **Large Datasets:** MERN can handle moderate website traffic and data volumes well. However, for colleges with a vast number of students, faculty, and complex data needs (e.g., extensive course catalogues, research databases), scalability limitations might arise. Scaling a MERN stack application can become complex and resource-intensive.

2. Security:

- **Shared Responsibility Model:** Cloud providers offering MERN-based solutions often follow a shared responsibility model. While the provider secures the underlying infrastructure, developers are responsible for securing the application itself. This requires a strong understanding of security best practices and diligent maintenance to mitigate potential vulnerabilities.

3. Performance Optimization:

- **Performance Tuning:** Optimizing website performance for fast loading times can require careful code optimization and infrastructure configuration within the MERN stack. This might involve additional development effort compared to using pre-optimized website templates or frameworks.

5.3 Future Scope

The future scope presents exciting possibilities for expansion and adaptation of latest technologies to our college website that can significantly enhance user experience and engagement. Key areas of potential development include:

Alumni Engagement Platform: Create an interactive platform for alumni to reconnect with the college community, share career experiences, mentor current students, and contribute to institutional initiatives. This platform could feature alumni directories, event invitations, and collaborative projects to foster a strong alumni network and promote lifelong connections.

Virtual Campus Tours: Develop virtual campus tours using 360-degree panoramic images, videos, and interactive maps to provide prospective students and parents with a comprehensive view of the college facilities, infrastructure, and amenities.

Integration with Emerging Technologies: Explore the integration of emerging technologies such as artificial intelligence (AI), machine learning (ML), augmented reality (AR), and virtual reality (VR) to enhance various aspects of the website, including adaptive learning experiences, virtual labs, and interactive simulations.

AI-Driven Chatbots: Implement AI chatbots to provide instant support and guidance to users. This can improve response times and provide personalized assistance based on user queries.

Multilingual Support: Expand multilingual support to cater to a diverse user base, making the website more inclusive and welcoming to international students and visitors.

User Behaviour Analytics: Implement advanced analytics tools to gain deeper insights into user behaviour and preferences. This data can inform ongoing improvements and content strategies.

BIBLIOGRAPHY

Books

- Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node by Vasan Subramanian.
- Getting Started with Node.js, Third Edition by Azat Mardan.

Research Paper

- A university Website design project: the design process, the prototype and some design issues. - IEEE
- College Website Using MERN Stack. - International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Online resources

- MongoDB Documentation: https://docs.mongodb.com/
- React.js Official Documentation: https://reactjs.org/docs/getting-started.html
- Node.js Documentation: https://nodejs.org/en/docs/
- https://reactrouter.com/en/main
- https://stackoverflow.com/

ANNEXURES

A-1 Use Case Diagram

A-2 Data Flow Diagram

A-3 Entity-Relationship Diagrams

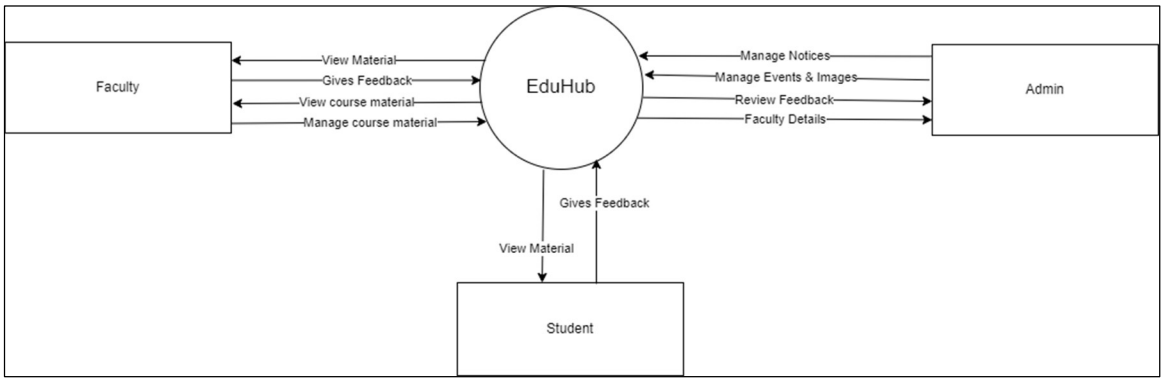
A-4 Activity Diagram

A-5 Screenshots

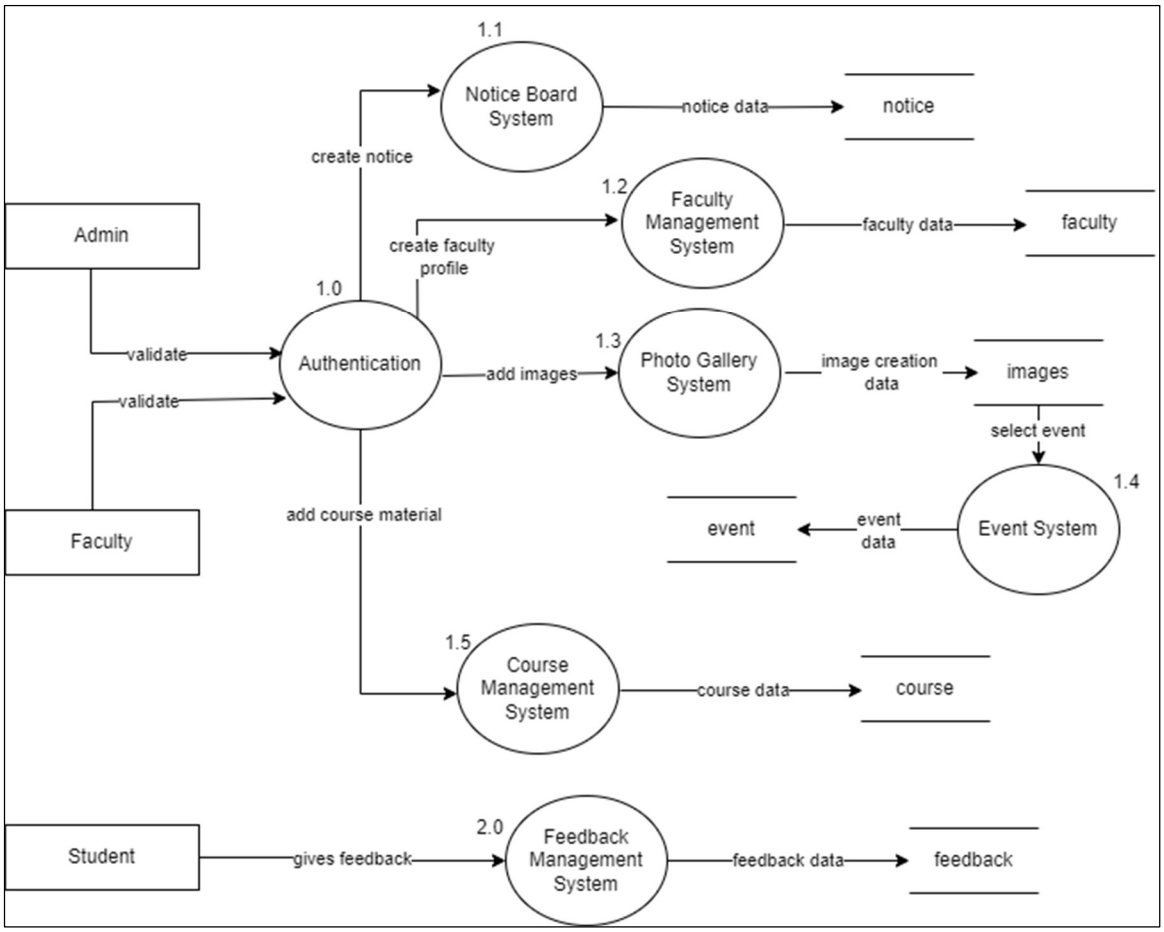
A-1 USE CASE DIAGRAM

A-2 DATA FLOW DIAGRAM

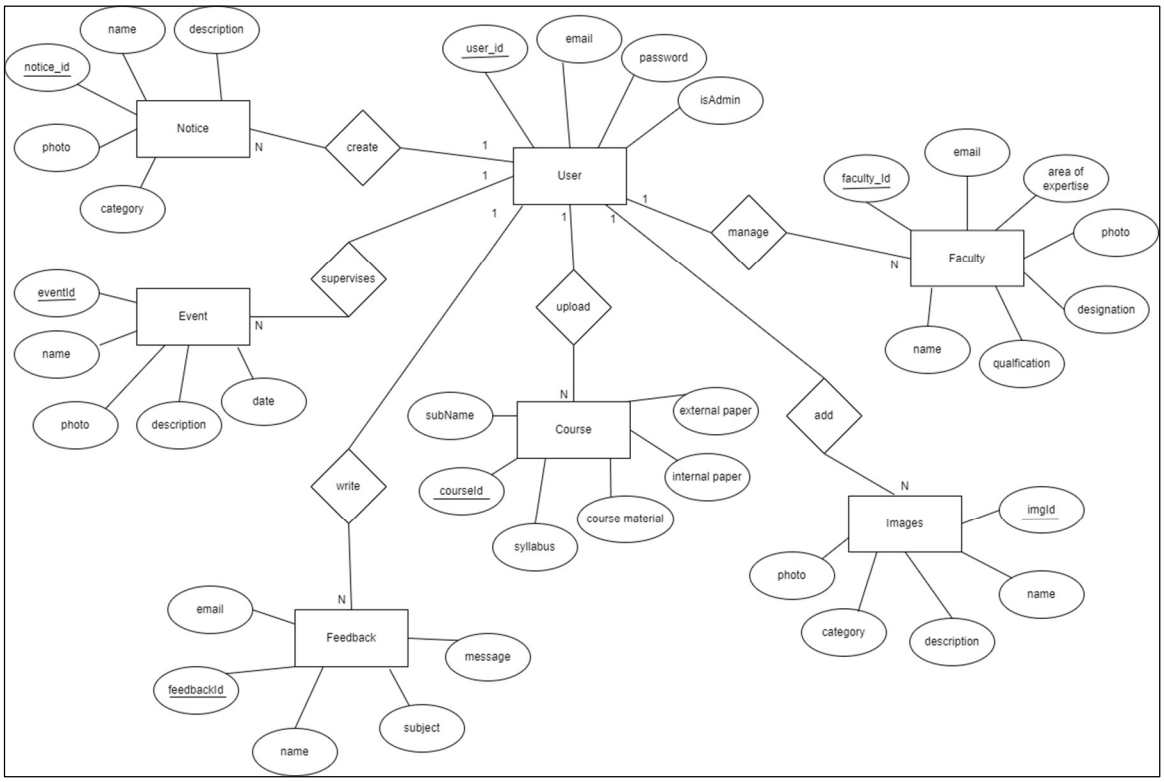
DFD-0 (Level 0)



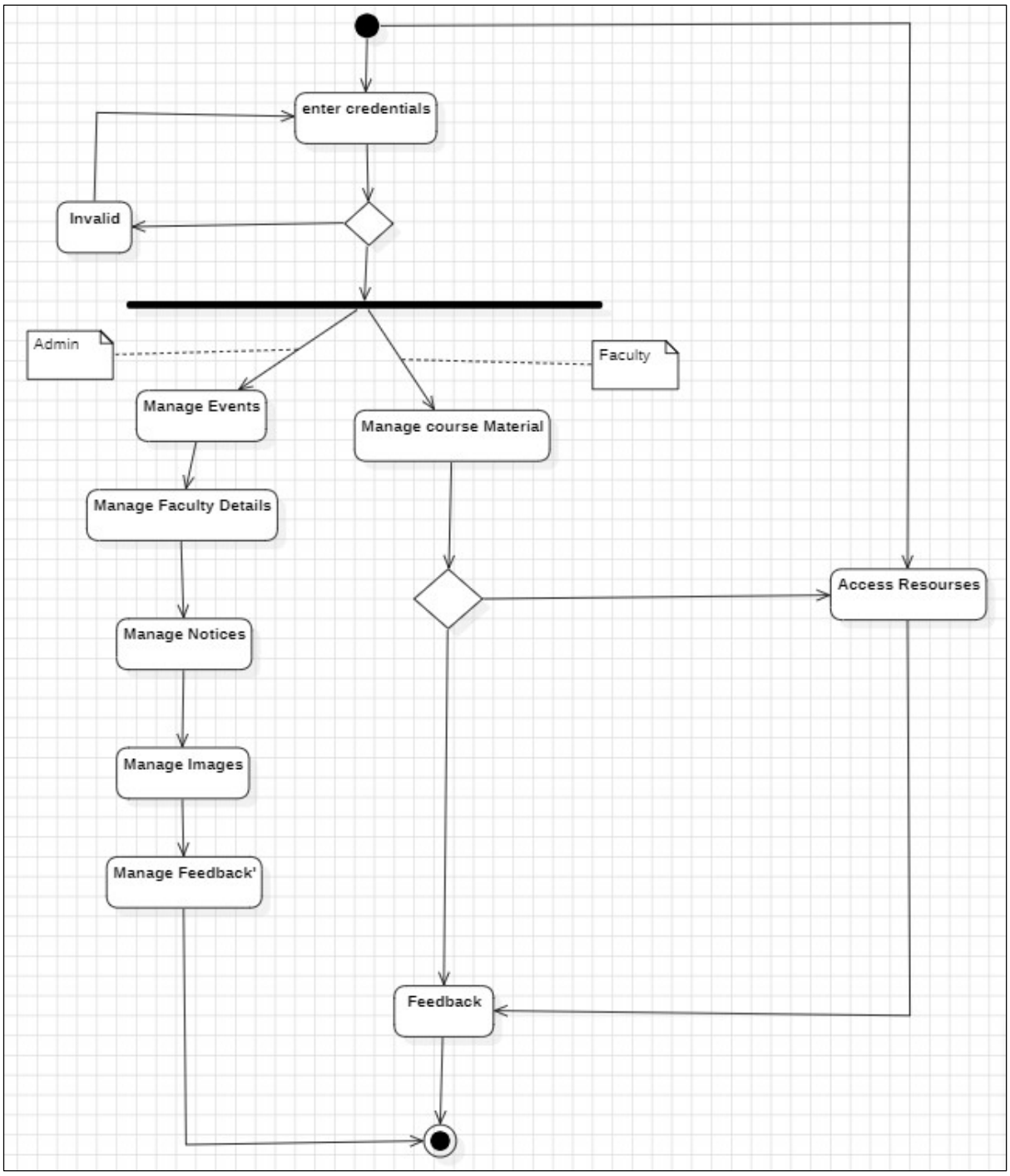
DFD-1 (Level 1)



A-3 ENTITY RELATIONSHIP (E-R) DIAGRAM



A-4 ACTIVITY DIAGRAM



A-5 SCREENSHOTS

Fig 5.1 Home Page

Fig 5.1.1 Home Page-Notices

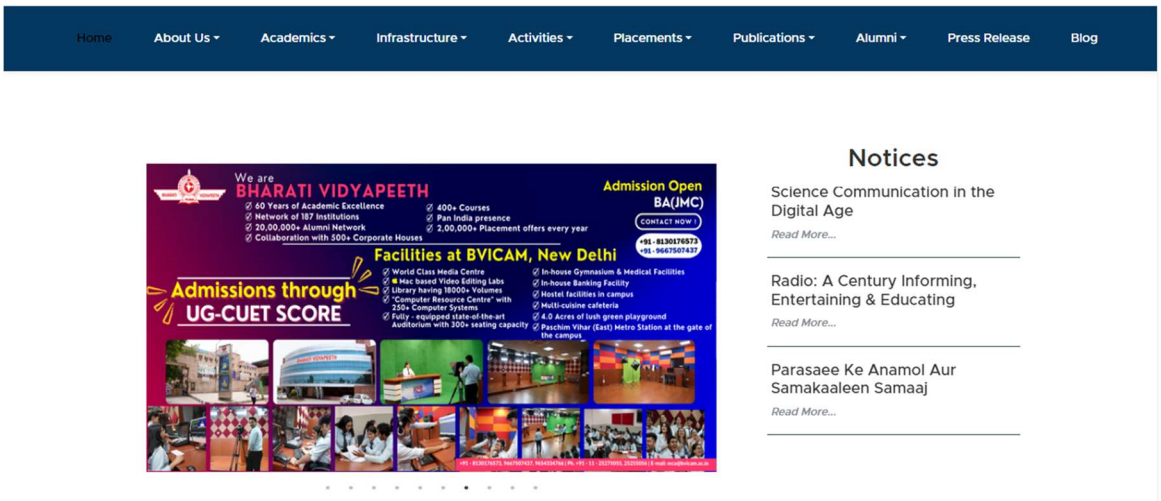


Fig 5.1.2 Home Page-About Us

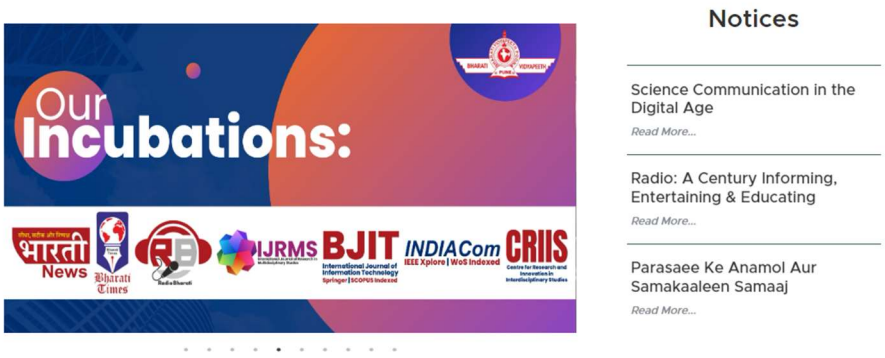


Fig 5.1.3 Home Page-Events

Fig 5.1.4 Home Page-Footer

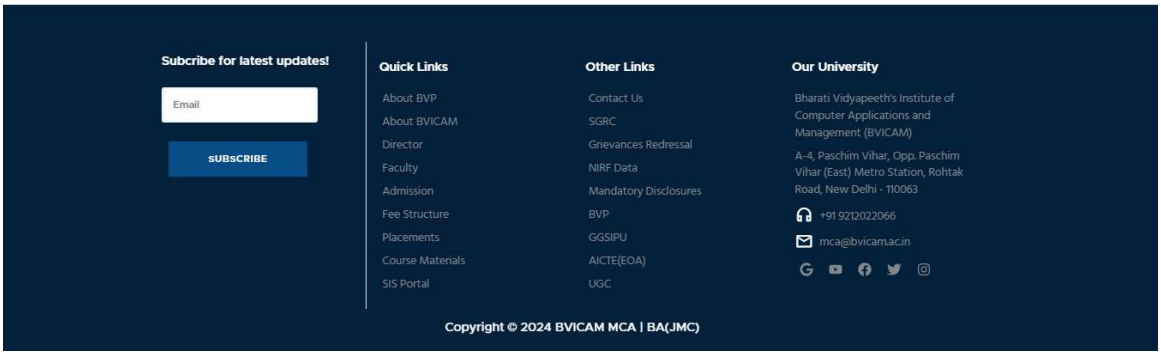


Fig 5.2 Notices Page

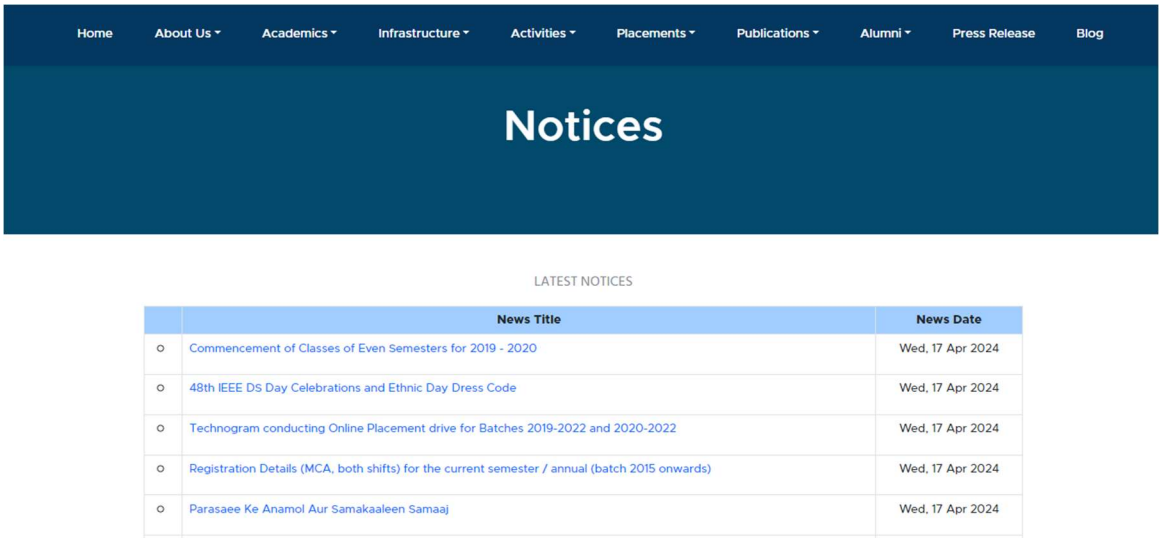


Fig 5.2.1 Notice Details Page

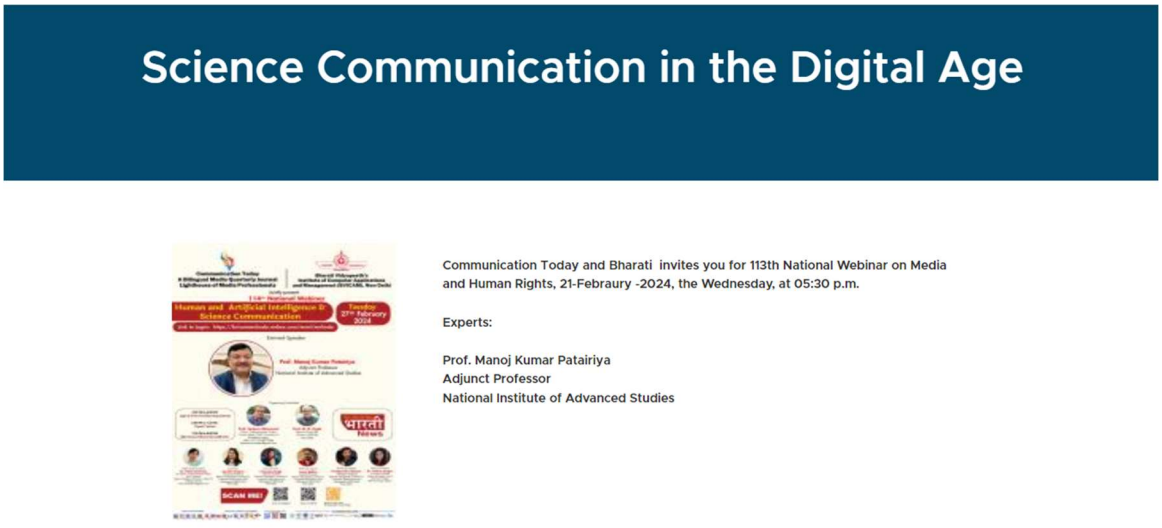


Fig 5.2.2 Types of Notices Page

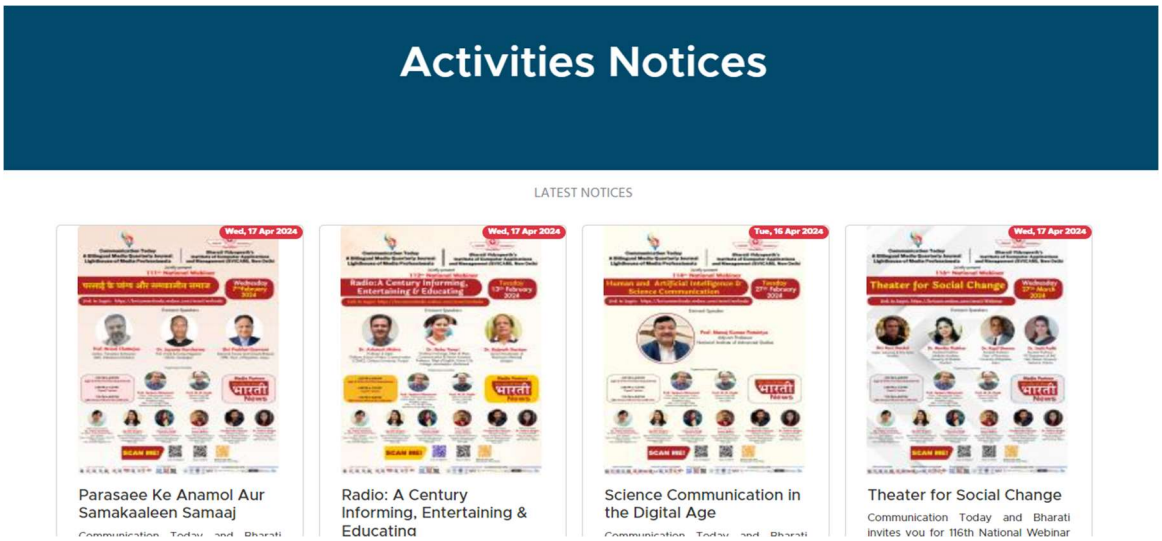


Fig 5.3 Picture Gallery Page

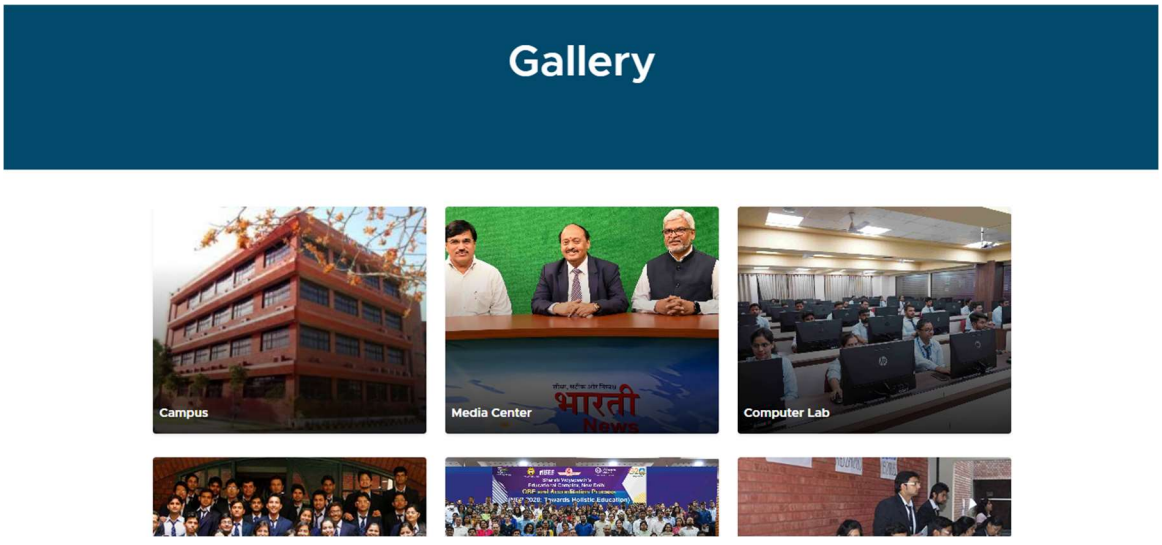


Fig 5.4 Contact Us Page

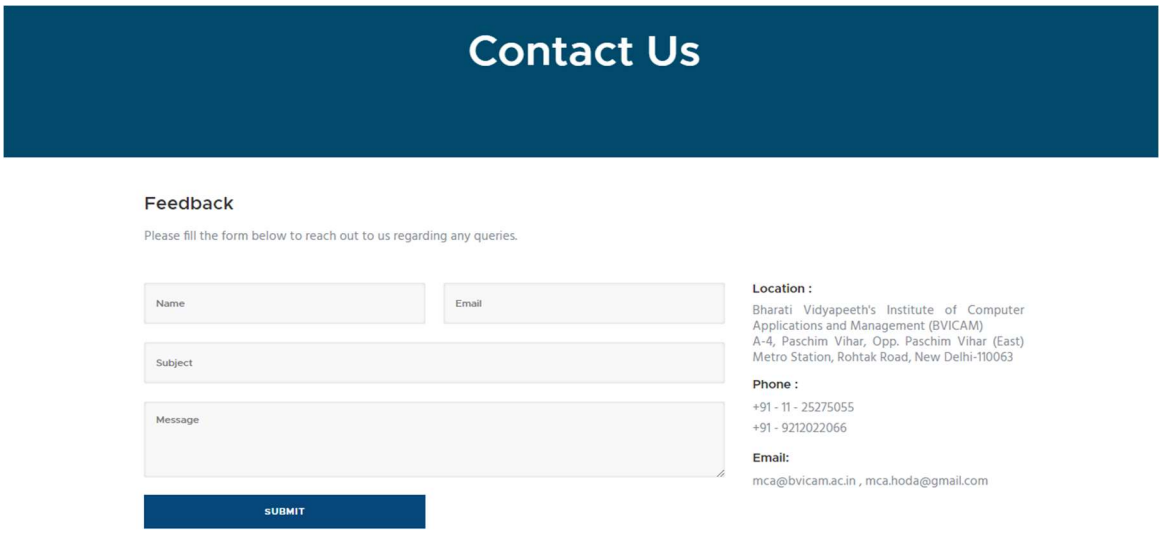


Fig 5.5 Faculty Page

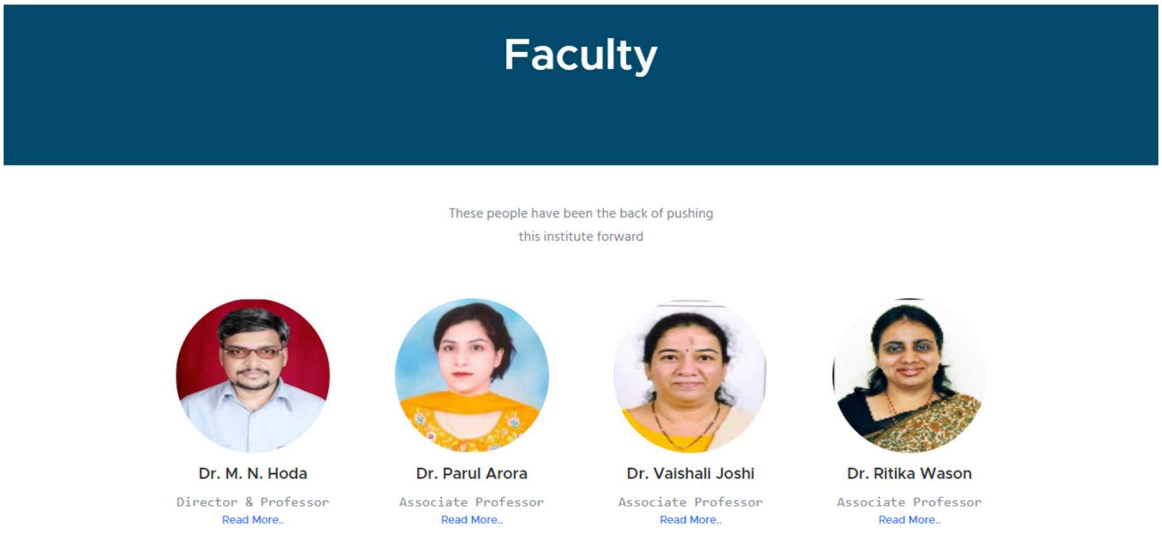


Fig 5.5.1 Faculty Information Page

Fig 5.6 Support Staff Page

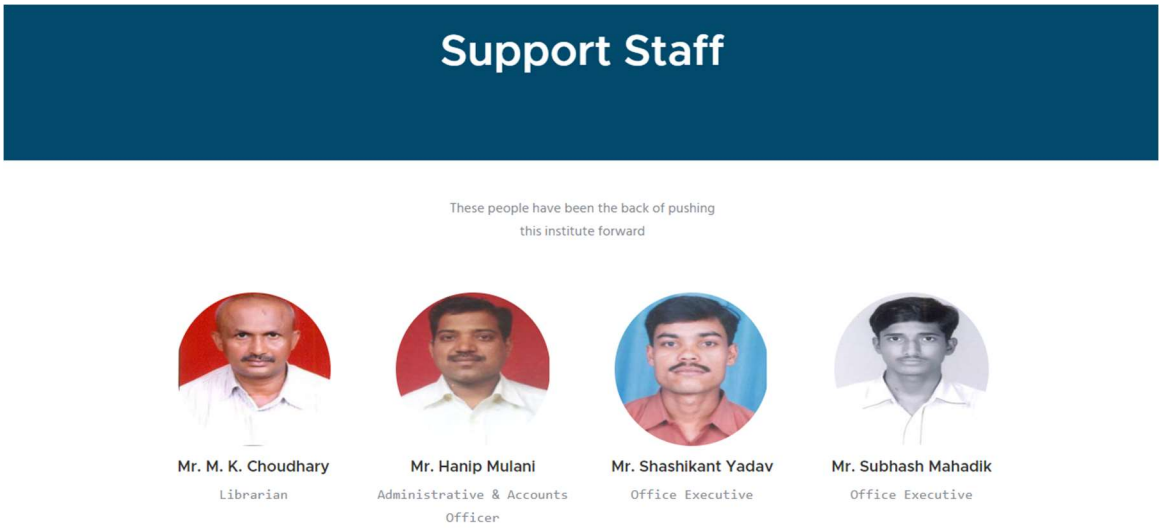


Fig 5.7 Subjects List

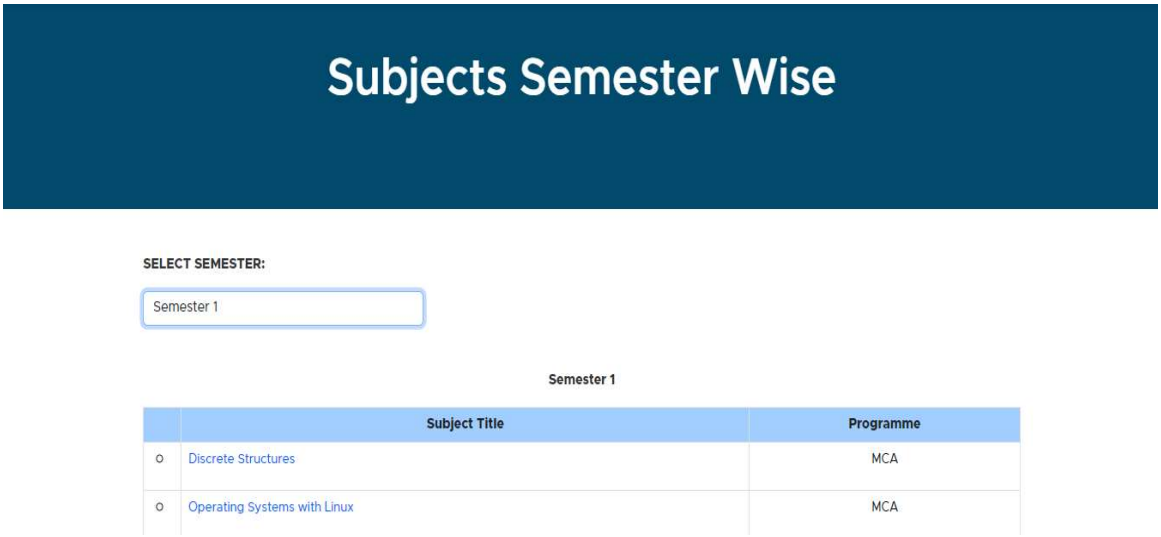


Fig 5.8 Dashboard Login Page

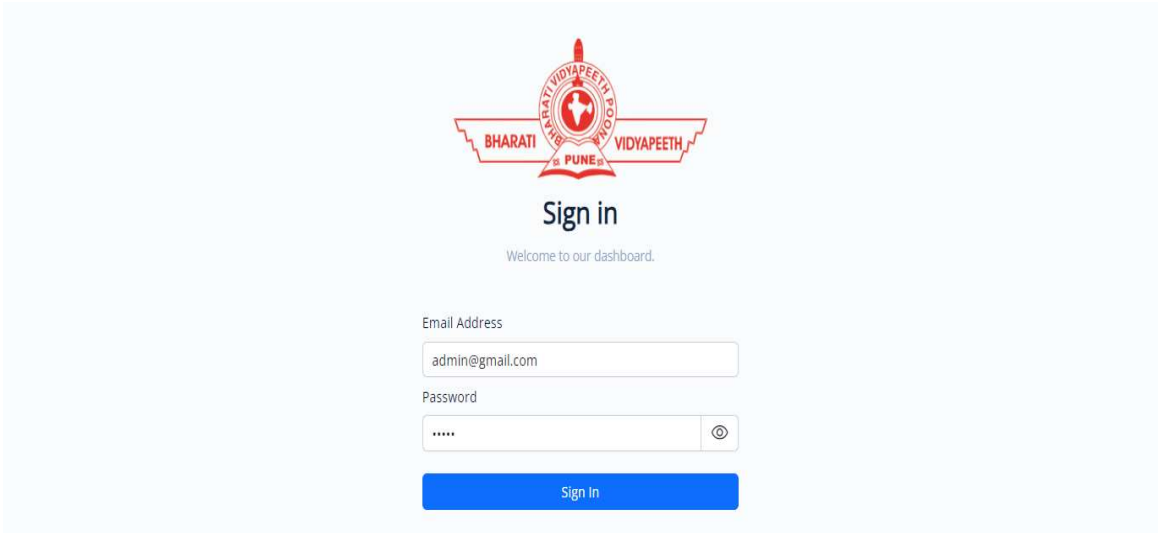


Fig 5.9 Course Material

Fig 5.10 Admin Dashboard

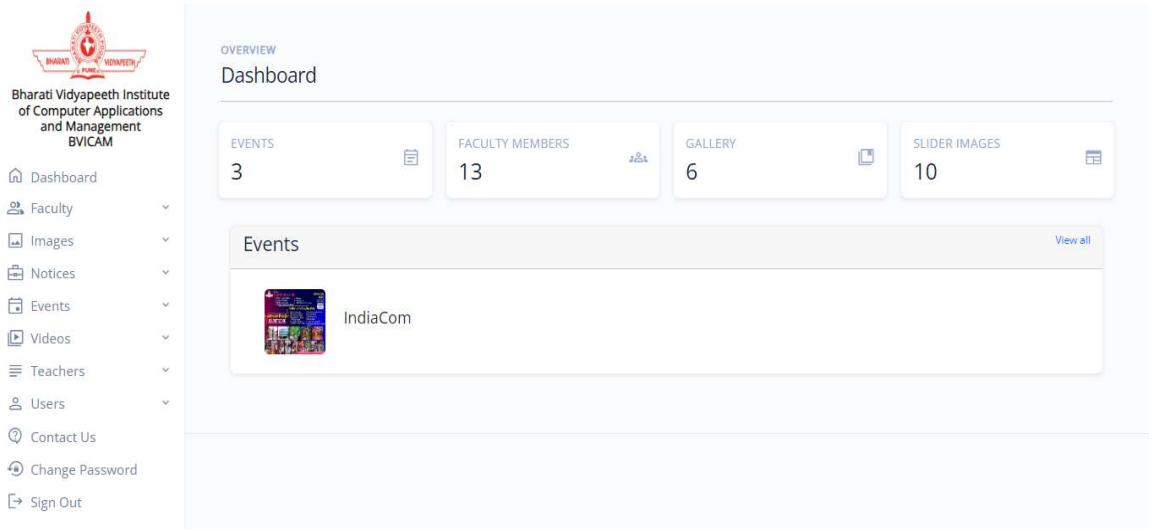


Fig 5.11 Faculty Dashboard

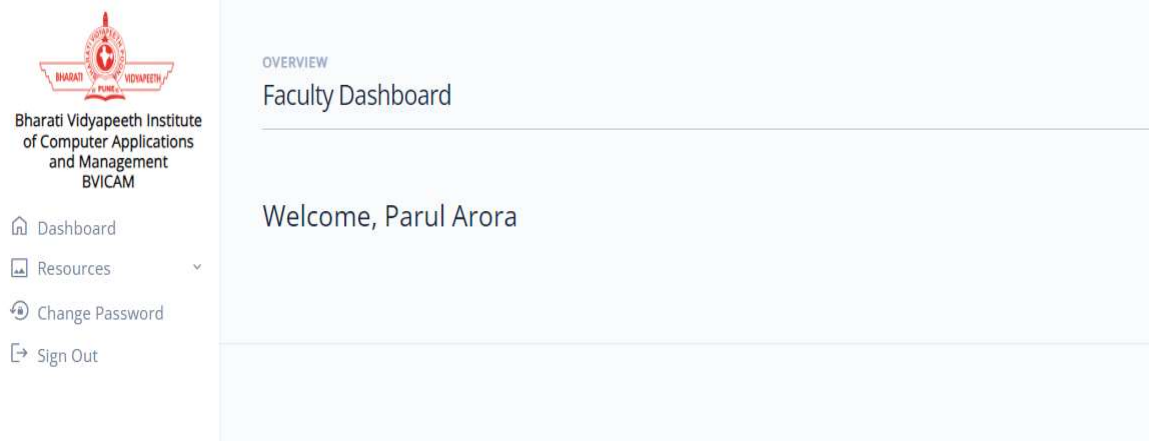


Fig 5.12 Add Faculty Page

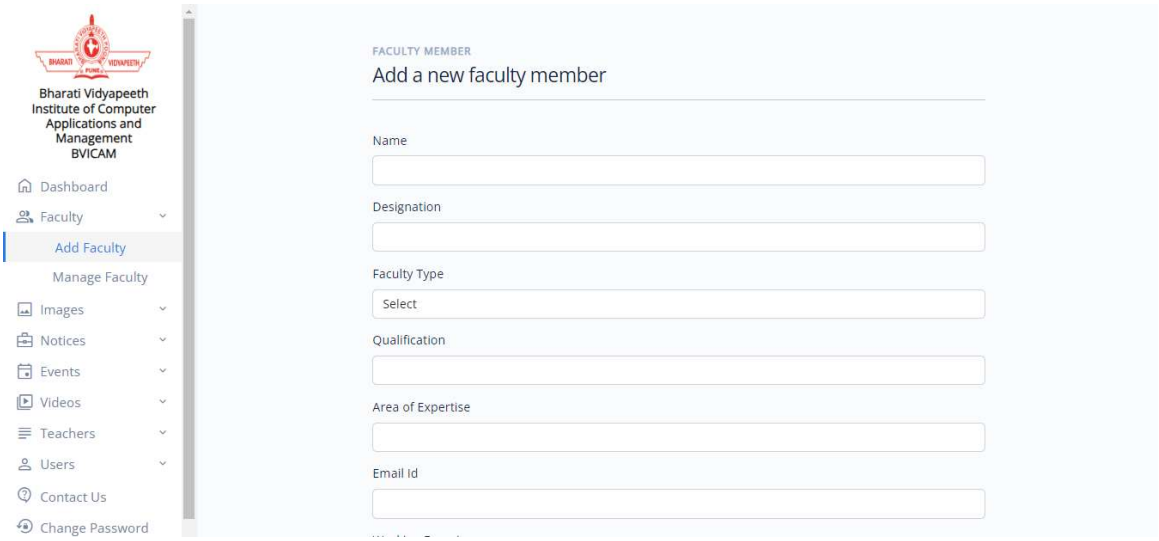


Fig 5.13 Manage Faculty Page

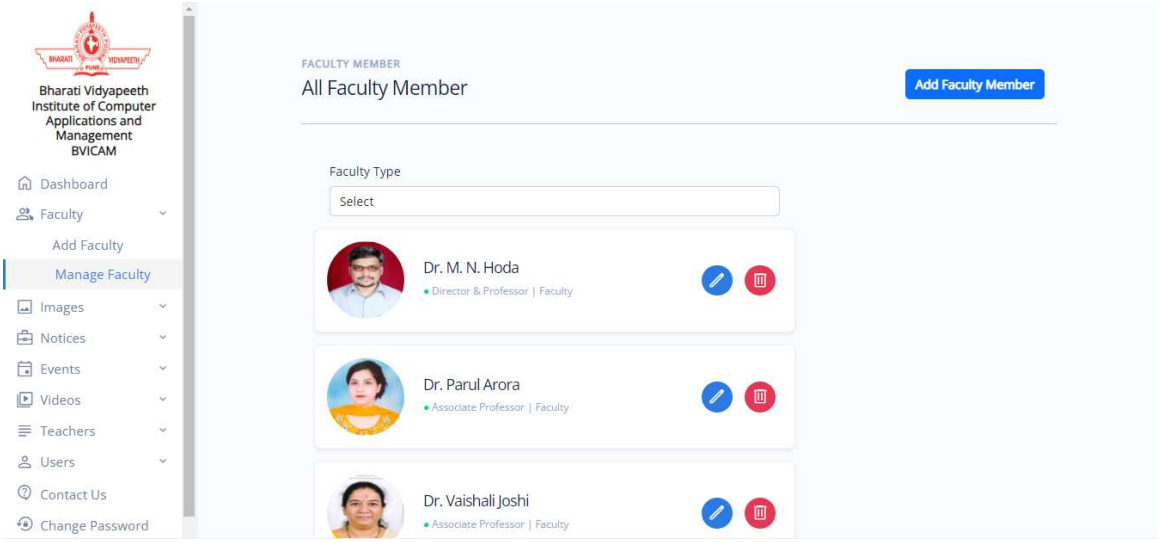


Fig 5.14 Add Images Page

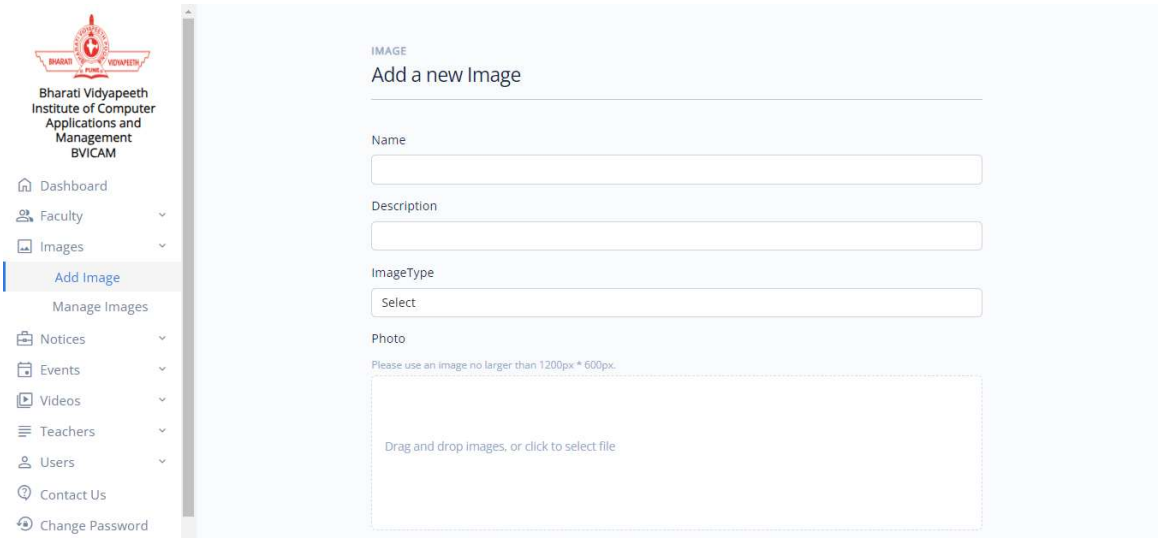


Fig 5.15 Images Page

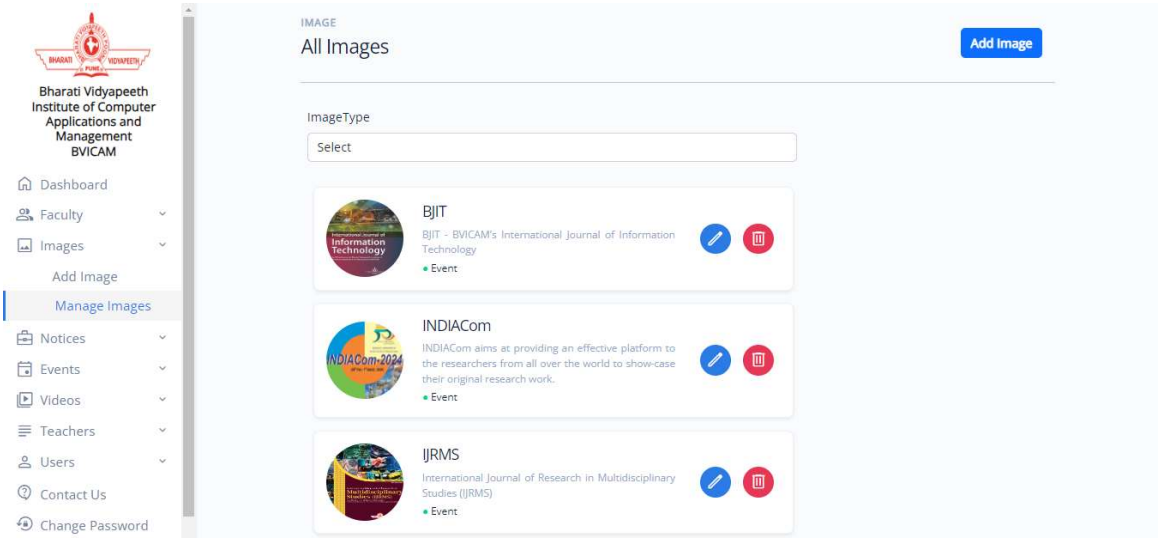


Fig 5.16 Add Notices Page

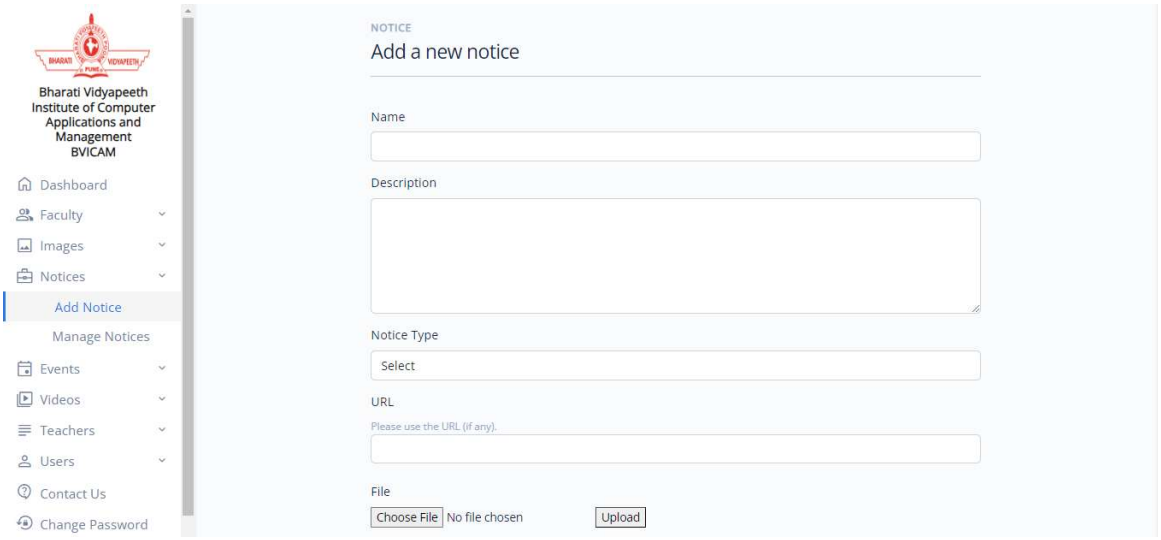


Fig 5.17 Manage Notices Page

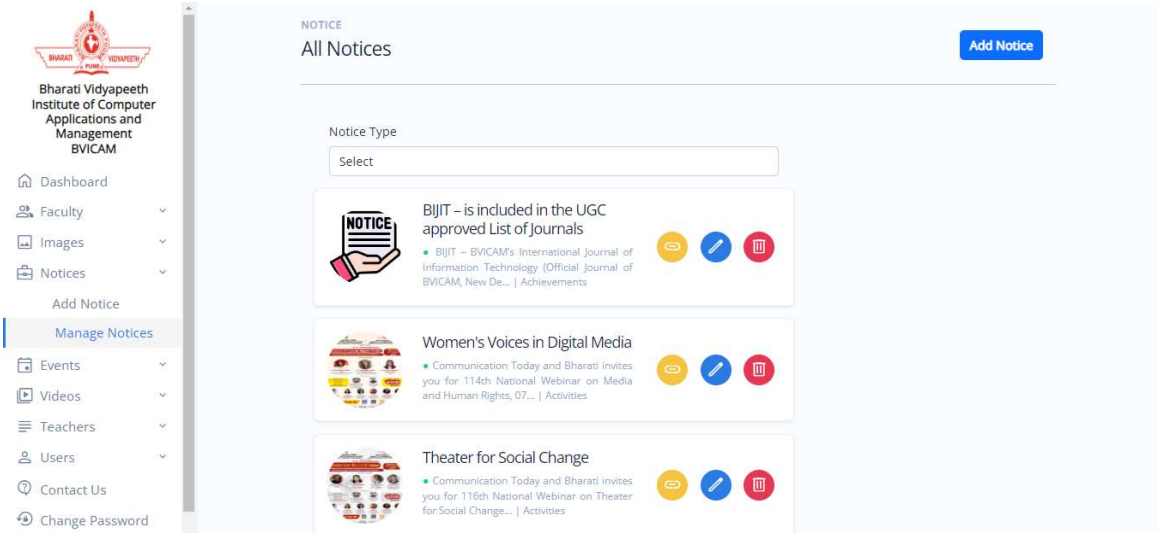


Fig 5.18 Create User Page

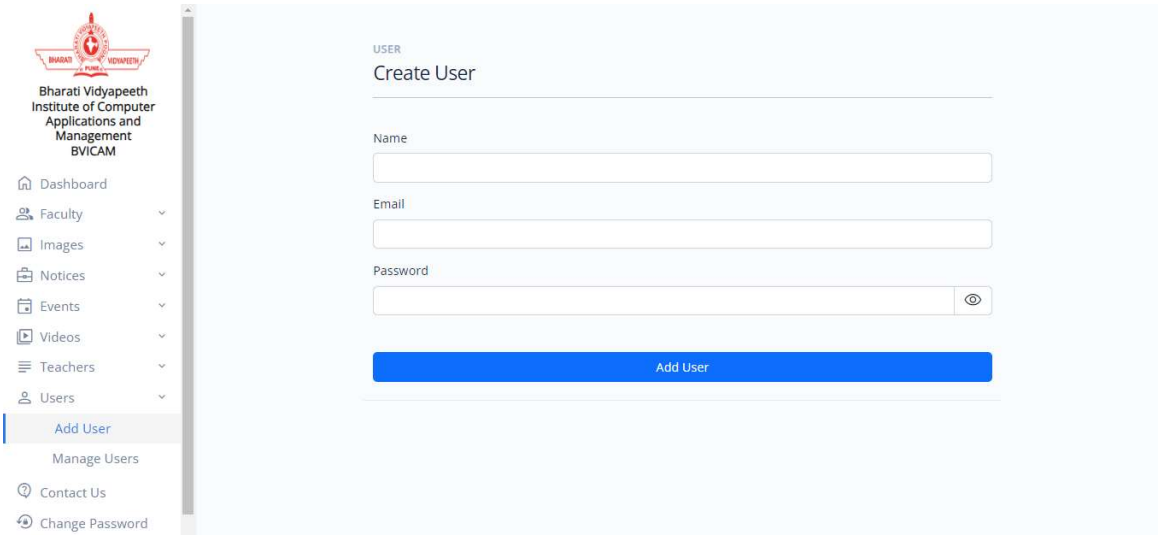


Fig 5.19 Manage Users Page

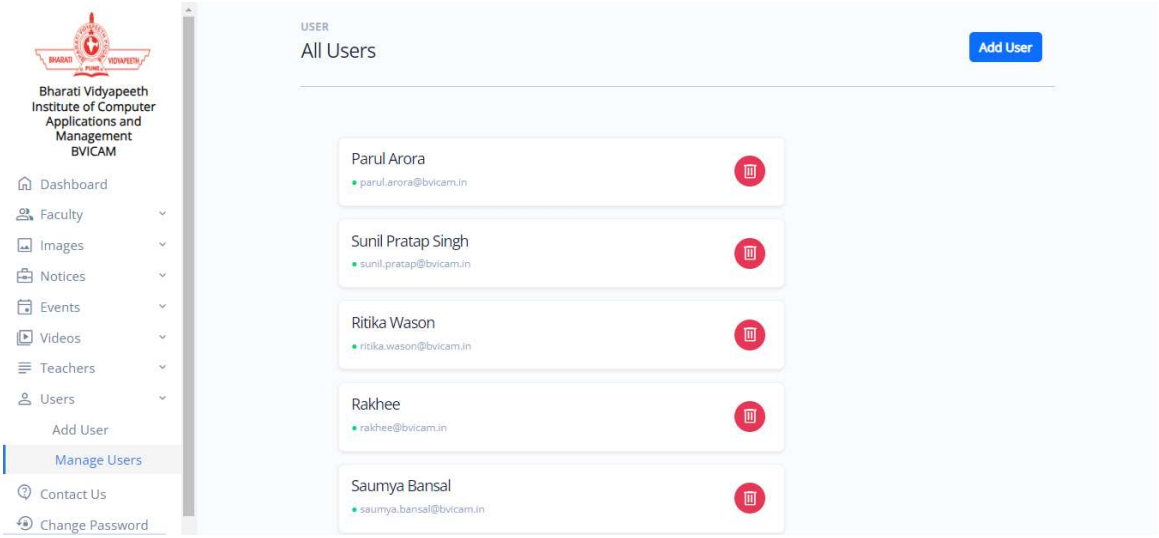


Fig 5.20 Teacher Assigned Page

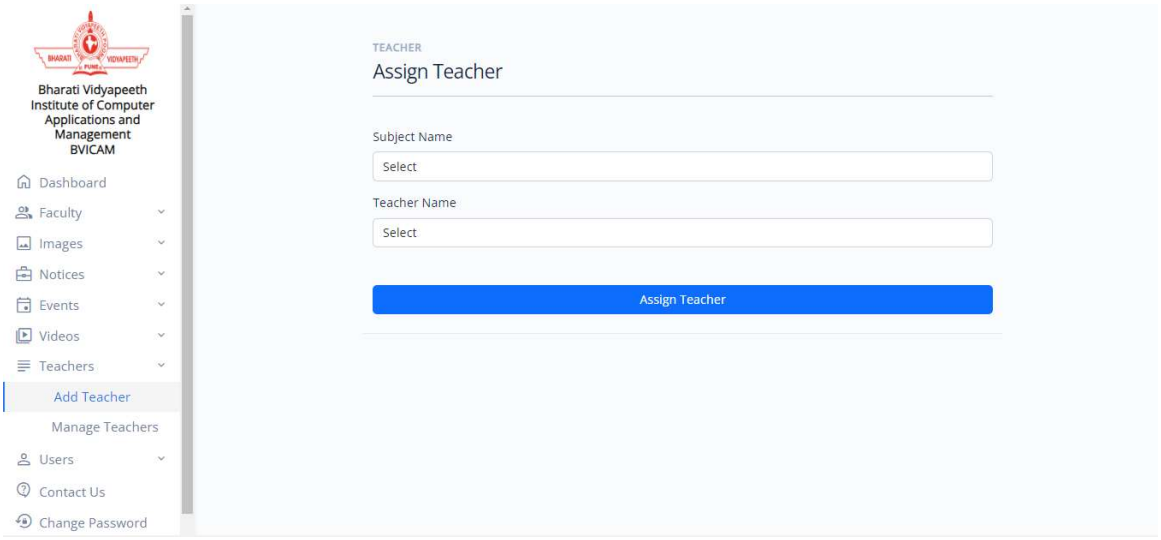


Fig 5.21 Manage Teachers Page

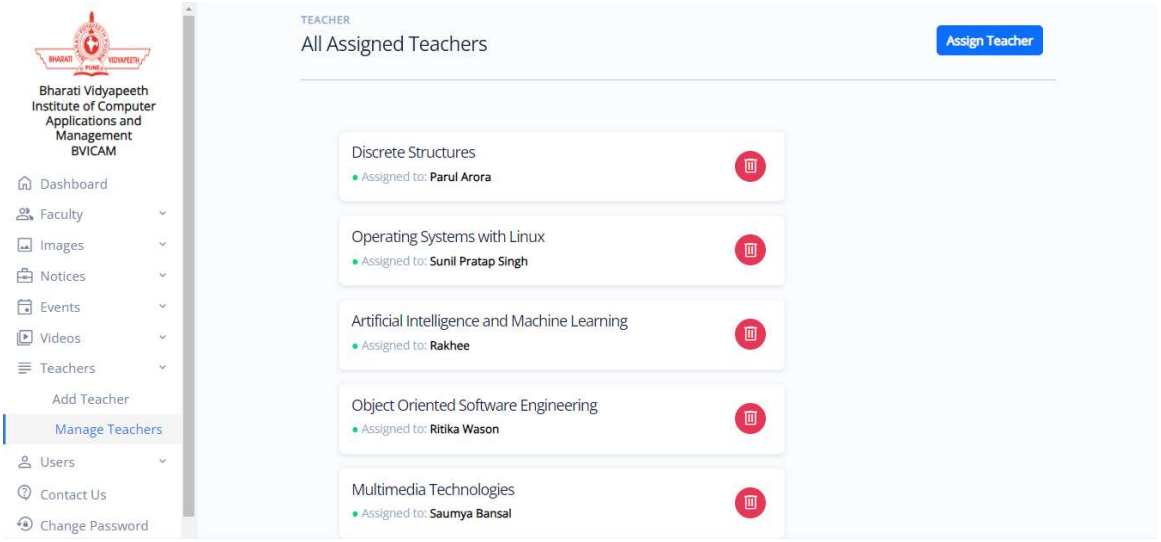


Fig 5.22 Contact Us Details Page

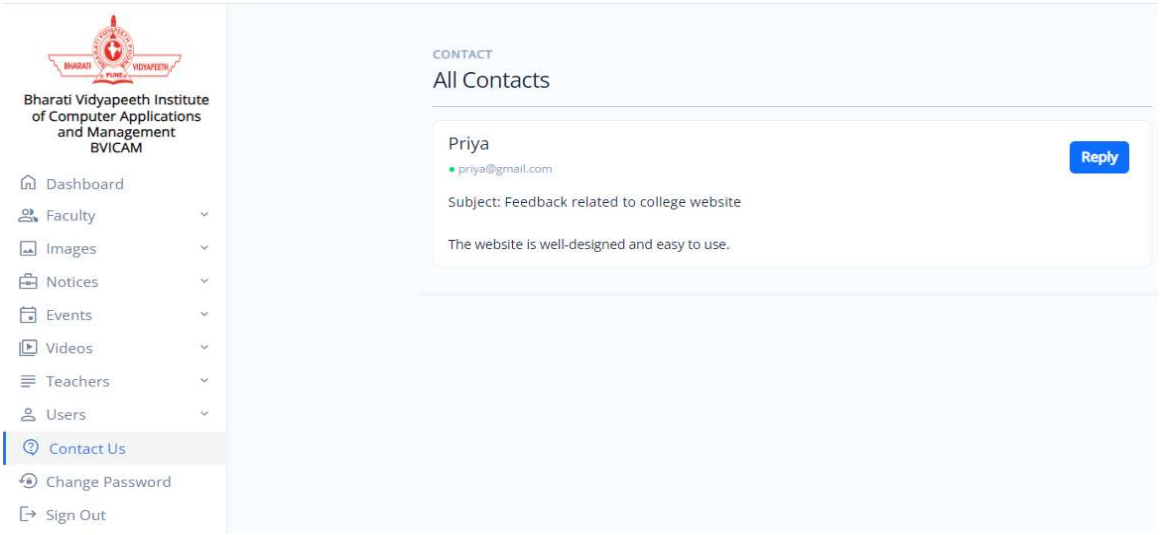


Fig 5.23 Change Password Page

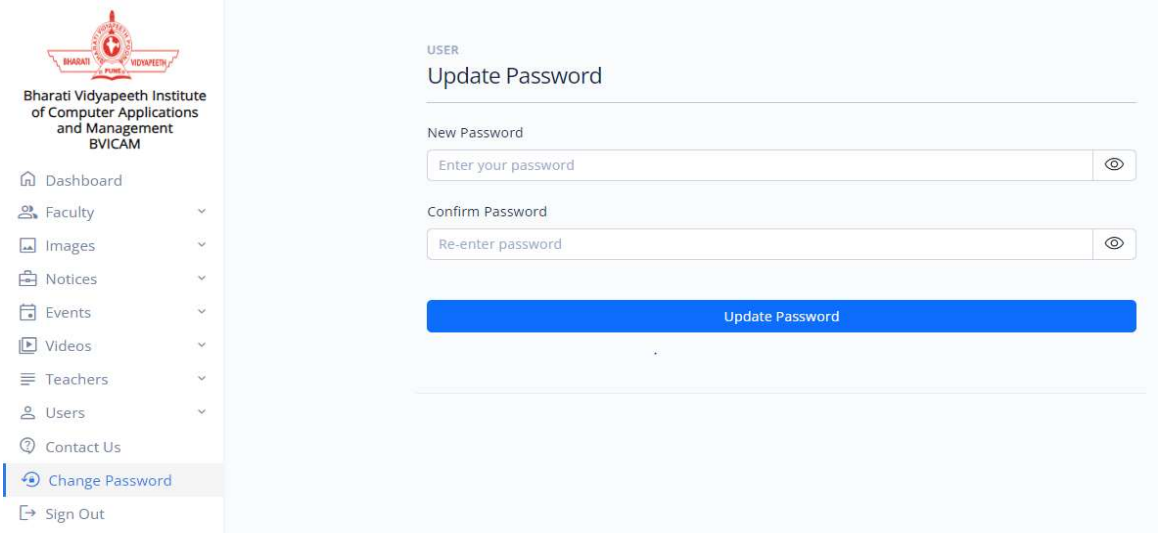


Fig 5.24 Add Subject Material Page

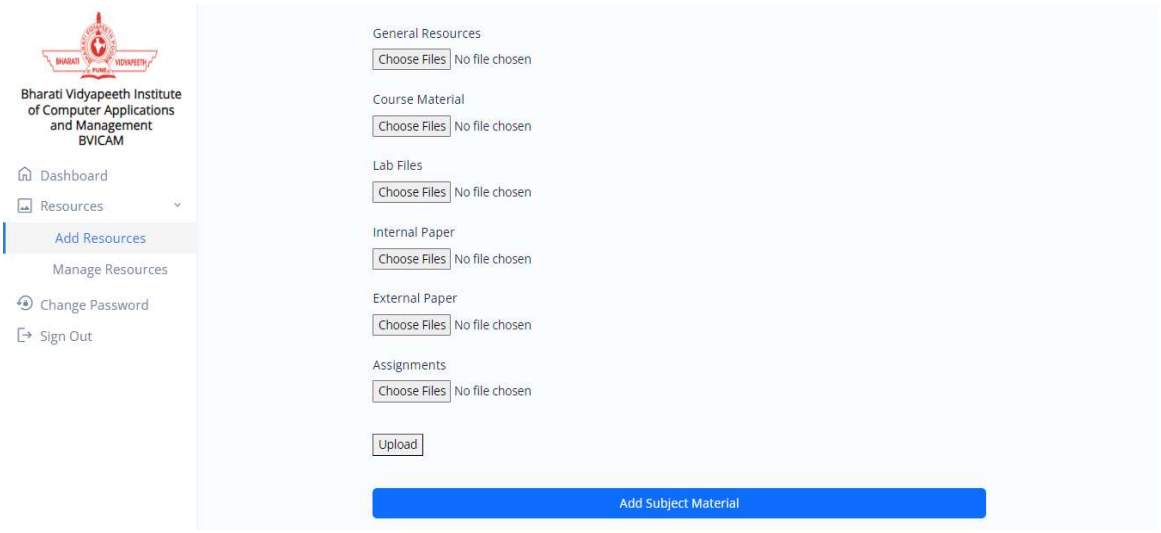


Fig 5.25 Manage Subject Material Page