A Stage-2 Project Report

on

Alumni Management Portal

Submitted By

1. Dhangar Yadnyesh K. (B400090102)

2. Gudgal Aniket N. (B400090112)

3. Kulkarni Shubham M. (B400090129)

In partial fulfillment for the award of

Bachelor of Engineering
(Electronics &Telecommunication Engineering)

Guided by

Prof. C. K. Kalawade

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE (M.S.)

Department of Electronics and Telecommunication Engineering

D.V.V.P. College of Engineering, Ahmednagar

2024-2025

CERTIFICATE

This is to certify that, the seminar titled, **"Alumni Management Portal"** submitted by **Dhangar Yadnyesh K. (B400090102), Gudgal Aniket N. (B400090112), Kulkarni Shubham M. (B400090129)** is a Bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Engineering (Electronics and Telecommunication) Degree from D.V.V.P. College of Engineering, Ahmednagar affiliated to Savitribai Phule Pune University, Pune (M.S, INDIA).

Place: Ahmednagar

Date:

Prof. C. K. Kalawade
Guide
Department of Electronics and
Telecommunication Engineering

Prof. J.P. Botkar
Coordinator
Department of Electronics and
Telecommunication Engineering

Dr. A. K. Patil
Head
Department of Electronics and
Telecommunication Engineering

Dr. U.P. Naik
Principal
D.V.V.P. College of Engineering
Ahmednagar

PROJECT APPROVAL SHEET

Dhangar Yadnyesh K. (B400090102), Gudgal Aniket N. (B400090112), Kulkarni Shubham M. (B400090129) has done the appropriate work related to "**Alumni Management Portal**" in partial fulfilment for the award of Bachelor of Engineering (Electronics & Telecommunication) of Savitribai Phule Pune University, Pune (M.S, INDIA) is being submitted to D.V.V.P. College of Engineering, Ahmednagar.

External Examiner:

Guide: Prof. C. K. Kalawade

Date:

Place: D.V.V.P. College of Engineering
Ahmednagar – 414111, M.S., INDIA

Acknowledgement

A Project work of such a great significance is not possible without the help of several people, directly or indirectly. First and foremost, I have immense happiness in expressing my sincere thanks to my guide, **Prof. C. K. Kalawade** for his/her valuable suggestions, co-operation and continuous guidance.

I feel a deep sense of gratitude to **Dr. Anita K. Patil**, Head of Electronics and Telecommunication Engineering Department for her continuous encouragement and for developing a keen interest in this field. It's my pleasure to thank **Dr. U.P. Naik** Principal, who is always a constant source of inspiration.

I am very much thankful to all my faculty members, whose presence always inspires me to do better. My happiness culminates, when I recall the co-operation extended by my friends during the completion of this seminar/project work.

A final and heartily thanks go to my parents.

Name of the Student

1. Dhangar Yadnynesh Kailas
2. Gudgal Aniket Navnath
3. Kulkarni Shubham Mahendra

ABSTRACT

Alumni portal is providing common platform for every institute. This project is aimed at developing an interactive system for the alumni of particular college. The proposed system will help alumni and the concern institution to create strong bounding through sharing their experiences, views, ideas, guidance, motivation inputs and strategies. A system that will be able to manage alumni data of an institution and provide easy access to secure information at both ends. This portal highlights the feature of communication, which will enable the current students to have interaction with the alumni of the college for getting various updates regarding job opportunities like industrial trends and industrial events etc. The proposed system will be developed with an open-source platform on web. This will help the user to access the portal from any location. Few of the required web design packages will be developed for better understanding. Various reporting formats can be generated based on filtering strategies. The admin panel will have all the rights to control the complete operation of system and implement add ones if required in future. The system will automatically list all alumni information their graduation and their status will be transferred from the student module to the alumni module.

Contents

List of Abbreviations i

List of Figures ii

List of Tables iii

Acknowledgement

1. INTRODUCTION 12

1.1 Introduction 12

1.2 Objective 12

1.3 Scope 12

1.4 Web Application 12

1.5 Significance 14

2. LITERATURE SURVEY 16

2.1 Introduction 16

2.2 Literature survey of methods/algorithms/published paper 16

2.3 Literature Survey Technical Specifications 18

2.4 Comparative Study of All the Methods 19

3. METHODOLOGIES 20

3.1 Requirements 20

3.2 Technologies 20

4. SYSTEM DEVELOPMENT 25

4.1 System Architecture 25

4.2 Block Diagram 26

4.3 Flow Diagram 27

5. TECHNICAL FLOW 30

5.1 User Authentication 30
5.2 Alumni Registration 30
5.3 Alumni Coordinator Dashboard 30
5.4 Event Creation 31
5.5 Batch Mentor Assignment 31
5.6 Alumni Gallery 31
5.7 Top Alumni Selection 32
5.8 Email & Bulk Communication System 32
6. PERFORMANCE ANALYSIS 33
6.1 Introduction 33
6.2 Functional Testing 33
6.3 Performance Testing 34
6.4 Usability Evaluation 35
6.5 Security Analysis 36
6.6 Test Execution Summary 37
7. RESULT DISCUSSION 38
7.1 Result 38
7.2 Homepage 38
7.3 About Us 39
7.4 Gallery 40
7.5 Manage Events 41
7.6 Manage Top Alumni 42
7.7 Alumni Registration 43
7.8 login page 44
7.9 Forgot Password 44

7.10 Alumni Profile 45
7.11 Batch Mentor Registration Form 46
7.12 login page for batch mentors 47
7.13 Batch Mentor Dashboard 48
7.14 Alumni Coordinator Dashboard 49
7.15 Send Email Notification 50
7.16 Batch Mentor Management interface 51
7.17 Adding Event 52
7.18 Email Notification Interface 53
7.19 Manage Top Alumni 54
8. CONCLUSION 55
8.1 Conclusions 55
8.2 Future Work 55
8.3 Future Scope 55
8.4 Advantages 55
8.5 Disadvantages 56
8.6 Application 56
REFRENCES

List of Abbreviations

Symbol	Illustration
BPMN	Business Process Modelling Notation
CSS	Cascading Stylesheets
CV	Curriculum Vitae
DBMS	Database Management System
DFD	Data Flow Diagram
E-R	Entity Relationship
HAA	Harvard Alumni Association
HTML5	Hypertext Markup Language (version 5)
NUS	National University of Singapore
PHP	Hypertext Pre-processor
RAD	Rapid Application Development
SQL	Structured Query Language
UML	Unified Modelling Language
API	Application Programming Interface

List of Figures

Figure No.	Name	Page
4.1	System Architecture	25
4.2	System Overview	26
4.3.1	Admin Login	27
4.3.2	Mentor Login	28
4.3.3	Alumni Login/Signup	29
6.1	Home Page	38
6.2	About Us	39
6.3	Gallery	40
6.4	Manage Events	41
6.5	Manage Top Alumni	42
6.6	Alumni Registration	43
6.7	Login Page	44
6.8	Forgot Password	44
6.9	Alumni Profile	45
6.10	Batch Mentor Registration Form	46
6.11	login page for batch mentors	47
6.12	Batch Mentor Dashboard	48
6.13	Alumni Coordinator Dashboard	49
6.14	Send Email Notification	50
6.15	Batch Mentor Management interface	51
6.16	Adding Event	52
6.17	Email Notification Interface	53
6.18	Manage Top Alumni	54

List of Tables

Table No.	Name	Page
6.1	Functional Testing	34
6.2	Performance Testing	35
6.3	Usability Evaluation	36
6.4	Security Analysis	37
6.5	Test Execution	37

1. INTRODUCTION

1.1 AIM OF THIS PROJECT:

Alumni associations play an essential role in higher education institutions, as they provide networking opportunities and contact between the university and the alumni or among the alumni. However, the lack of communication between the alumni, faculty and the department as a whole leading to the lack of updated data of alumni hinders the effective development of an alumni association. The lack of data update can be tied to disinterest or ignorance of the alumni about the potential of an alumni association to promote a professional network, academic and even personal relationships. Similarly, the need for an effective system to maintain these records and to update them frequently becomes increasingly pressing as the number of students enrolled in the campus continues to increase. One of the ways to consolidate the relationship between alumni and educational institutions is to create a virtual and interactive portal to provide useful information to the alumni to help extend the contact between the alumni and the university to beyond the period of course completion, regardless of geographic location or time.

1.2 OBJECTIVE:

- **Streamline Alumni Data Management** – Maintain structured records of alumni, easy retrieval and updates for networking and professional tracking.
- **Enhance Alumni Engagement & Networking** – Provide alumni with opportunities to connect with mentors, coordinators, and fellow graduates for career guidance and collaboration.
- **Improve Communication & Event Coordination** – Implement **bulk email notifications, event scheduling, and mentorship assignments** to keep alumni informed and involved.
- **Integrate Scalable & Efficient Technologies** – Utilize **Python, Django, HTML, CSS, JavaScript, and MySQL** to ensure a robust, efficient, and scalable system.

1.3 SCOPE:

To bring together all the old students and the faculty members to share their experiences with each other which utilize the rich experiences of old students of the college for the benefit and progress of the present students. It also helps to provide guidance to the

present students in their endeavor for better employment and higher studies. To get the valuable advice of the Alumni in the overall development of the college.

1.4 Web Application:

1.4.1 Preface:

A website application, which is also popularly known as a web app, is a software application program that uses web-based technology to perform specific tasks. Remote web servers host web applications and store relevant information from numerous connected computers. You can use a client program to run the web applications and access or enter the required data. That is why people often refer to web apps as client-server programs. Web applications have undergone many developments, and the current ones are far more sophisticated in their features and use than the earlier simplistic ones. They are also far more indispensable for personal and business use.

1.4.2 Frontend:

Front-end web development, also known as client-side development, is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site, they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

1.4.3 Backend:

Web development activities that are done at the back end of programs are referred to as back-end development. Back-end development covers server-side web application logic and integration and activities, like writing APIs, creating libraries, and working with system

components instead of frontend development, which focuses on customer-facing services and programs.

Backend developers build code that allows a database and an application to communicate with one another. Backend developers take care and maintain the back-end of a website, including databases, servers, and apps, and they control what you don't see. Backend is the server-side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

1.4.4 Database:

The Database is an essential part of our life. As we encounter several activities that involve our interaction with databases, for example in the bank, in the railway station, in school, in a grocery store, etc. These are the instances where we need to store a large amount of data in one place and fetch these data easily.

A database is a collection of data that is organized, which is also called structured data. It can be accessed or stored in a computer system. It can be managed through a Database Management System (DBMS), a software used to manage data. Database refers to related data in a structured form.

In a database, data is organized into tables consisting of rows and columns and it is indexed so data can be updated, expanded, and deleted easily. Computer databases typically contain file records data like transactions money in one bank account to another bank account, sales and customer details, fee details of students, and product details. There are different kinds of databases, ranging from the most prevalent approach, the relational database, to a distributed database, cloud database, and NoSQL databases.

1.4.5 Container:

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Podman container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Podman containers – images become containers when they run on Podman Engine. Available for both

Linux and Windows-based applications, containerized software will always run the same, regardless of the
infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

1.5 Significance:

The system will be of a high importance to the following:

- **Department:** The College will keep track of its alumni so as to leverage on the bene its of easier monetary nuancing wherever need be. The College will also gain publicity due to its healthy relationship with its alumni who will always passionately speak about it. They will also track the quality of their graduates out there in the job market and therefore ne tune the content delivered to the ongoing students.
- **Students**: The students will enhance their chances of getting internships and employment opportunities due to the connections with the alumni. They will also enhance their soft skills through mentorship and career guidance by the alumni.
- **Alumni:** The Alumni will keep connected with a batch of their old classmates so leveraging the opportunities created within their network. They will get updates of the latest information and news from the institution.

2. LITERATURE SURVEY

2.1 Introduction:

- Alumni Engagement and Communication: Strategies and techniques for engaging alumni through online platforms. Effective communication channels and tools for alumni outreach. Personalization and customization in alumni communication.
- Alumni Database and Directory Management: Data management techniques for maintaining an alumni database. Alumni directory design and functionality. Privacy and security considerations in managing alumni data.
- Event Management and Reunions.
- Planning and organizing alumni events and reunions through online platforms. Event promotion, registration, and ticketing systems for alumni gatherings. Alumni event success stories and best practices.
- Career Services and Job Placement: Integration of career services and job placement features in alumni portals. Alumni mentoring and networking opportunities for career advancement. Alumni involvement in internships, job fairs, and recruitment activities.
- Technology and Frameworks: Evaluation and comparison of different web frameworks for building alumni portals. Use of technologies like HTML, CSS, Python, Django, And Excel for alumni platforms. Scalability, performance, and security considerations in portal development.
- Case Studies and User Experience: Case studies of successful alumni portals and their features. User experience design and usability studies for alumni portals.

2.2 Literature survey of methods/algorithms/published paper:

2.2.1 Parth P. Sawai1, Prajyot V. Chambhare2

This research paper presents the development and implementation of the "Alumni Connect Hub," a web-based alumni management system for [Sipna College of Engineering and Technology Amravati]. Designed to facilitate seamless interaction and information exchange among alumni, students, and staff, the system aims to cultivate a strong and engaged alumni community within the institution. The paper delves into the system's design, development, objectives, scope, functionalities, and technical architecture, and highlights its potential benefits for fostering improved alumni relations and community engagement within the college. Additionally, it discusses the broader applicability of the Alumni Connect Hub as a web-based alumni management system for educational institutions, emphasizing

its role in addressing the common challenge of maintaining strong connections between alumni and their alma mater, thereby contributing to the development of robust alumni communities across diverse educational settings

2.2.2 Alumni.harvard.edu/

The Harvard Alumni Association (HAA) is an alumni association that maintains and enhances a highly engaged, vibrant community of the University of Harvard alumni and friends worldwide.

The HAA helps alumni connect with the University and with fellow graduates. This is achieved through a wealth of Club activities, continuing education programs, online services, and alumni events around the world.

This association has an online platform on which it carries out a section of its activities. These activities include bringing together alumni and associate members. Associate members are those persons that enrolled for a certificate, degree or diploma but didn't complete their education. These members can contribute to the various projects at Harvard University. The alumni are updated about the events that are happening around the university.

However, the alumni cannot view the current progress of the projects in real time i.e. there's no real time accountability. The alumni cannot engage with one another or even the students.

2.2.3 Babu M, Sandhiya K, Preetha V, Sankara Eshwari S, Ramya Chitra M

An alumni association consists of graduate and former students. Colleges depend on alumni to give mentoring to the students which will help the college students to have contact with the alumni of the college to get various information regarding job opportunities like industrial trends, industrial events, internships, scholarships, etc. This project manages the freshers and the old graduate students with their respective information, to actively participate in registering, searching, and managing the alumni information to get further updates. Hence, the preferred Dynamic design of the Alumni portal is to provide communication between all alumni and students. This Alumni Portal is used to give details regarding the data about colleges, Special events, Careers, and Post Campus Placements. SMS4-BSK cryptosystem is used to protect the data.

2.2.4 Siddiqui Maryam, Momin Alimuddin, Prof. P.S Lokhande

Alumni portals is providing common platform for every institute. Owing to the need to have all the Alumni (already passed out students) must be connected to the Institute, resulted in sharing their experiences, views, ideas, guidance motivations and strategies. The

objective of Alumni portal application is to allow old and new students of the college to have communication. This allows students to know about each other and their current activities. This portal highlights the feature of communication, which will enable the current students to have interaction with the alumni of the college for getting various updates on current industry trends, Internship opportunity, sponsored projects and various referrals opening in the corporate world. This portal will serve the cause of integrating all the stakeholders of Institute such as, Alumni, College students, Faculties to avail the guidance and knowledge sharing on various domains. In this project we proposed the Dynamic Architectural design of the Alumni portal, which enable the two-way communication between all stakeholders.

2.2.5 IIT Kanpur

IIT has Gallery with many images, Lectures details with specific day and time; contacts which show whom all are connected with this portal, woman alumni convention as separate for the woman only on the Home page. And Features are Gallery present on the home page; visit for guide allocations is given so that the junior student can get the solution or the advice for their difficulties. And Notice with highlighted form is shown for better visualization. Also gives the Donor Initiative Chance. And IIT KANPUR shows information events about present on every page where the current events are highlighted also there is a separate women alumni convention and that is the unique feature.

However, the system lacks an inbuilt feature to support alumni groups as currently the groups are supported through an integration with a private vendor called Gradual that provide software solutions to associations.

2.3 Literature Survey Technical Specifications Used in the Project

In the development of the Online Alumni Management System, several key technologies were employed to ensure a robust and user-friendly platform. Python was utilized for its versatility and efficiency in handling backend processes, including data manipulation and server-side scripting. Python's extensive libraries and frameworks, such as Django and Flask, facilitated rapid development and integration of complex functionalities.

Excel was employed for data analysis and management, leveraging its powerful features for organizing, visualizing, and interpreting large datasets. This tool was particularly useful in the initial stages of data collection and analysis, providing a straightforward interface for handling alumni information. HTML and CSS were fundamental in crafting the front-end of the system, ensuring a responsive and aesthetically pleasing user interface.

HTML provided the structural framework of the web pages, while CSS was used to style and layout the content, enhancing the overall user experience. The combination of these technologies allowed for the creation of a comprehensive system that effectively bridges the communication gap between the university, its alumni, and current students.

2.4 Comparative Study of All the Methods

A comparative study of the various methods employed in the development of the Online Alumni Management System reveals distinct advantages and limitations associated with each approach. Python, with its robust libraries and frameworks, offered significant advantages in terms of backend development, enabling efficient data processing and server-side scripting. Its integration capabilities with other technologies made it a versatile choice for this project.

Excel provided a reliable tool for data management and preliminary analysis, though its limitations in handling dynamic web-based interactions necessitated the use of more advanced technologies for the final implementation. HTML and CSS were indispensable for front-end development, ensuring that the system was both functional and visually appealing.

However, these technologies alone were insufficient for creating interactive and dynamic web applications, highlighting the need for complementary scripting languages such as JavaScript. The comparative analysis underscores the importance of selecting appropriate technologies based on their strengths and limitations, ensuring a balanced and effective approach to system development. This holistic integration of various methods and tools was crucial in achieving the project's objectives, providing a seamless and efficient platform for alumni management.

3. METHODOLOGIES

3.1 REQUIREMENTS:

3.1.1HARDWARE REQUIREMENTS:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

01. CPU Processor: Min Pentium IV
02. RAM: Min 4 GB (Main Memory)
03. Processor Clock Speed: 2.4 GHz
04. Storage Memory: Min 30 GB
05. Mouse: At least Trackball Mouse
06. Keyboard: QWERTY Keys
07. Monitor Display: IPS display
08. Network Speed: Min 2 Gbps

3.1.2 SOFTWARE REQUIREMENTS:

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

01. Operating System: Windows / Mac OS / Linux distros
02. IDE: Visual Studio Code
03. Browser: Chrome / Firefox / Safari / Opera
04. API Validator: Thunder Client/postman

3.2 TECHNOLOGIES:

3.1.2 FRONTEND:

- **HTML:**

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are

generally used to describe a web page's appearance/presentation or functionality. "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as **Error! Filename not specified.** and directly introduce content into the page. Other tags such as surround and provide information about document text and may include other tags as sub elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

- **CSS:**

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, which reduces complexity and repetition in the structural content; and enable the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

- **JavaScript:**

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more about JavaScript. This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about API specifics to Web pages, please see Web APIs and DOM. The standards for JavaScript are the ECMAScript Language Specification (ECMA-262) and the ECMAScript Internationalization API specification (ECMA-402). The JavaScript documentation throughout MDN is based on the latest draft versions of ECMA-262 and ECMA-402. And in cases where some proposals for new ECMAScript features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features.

3.2.2 BACKEND:

3.2.2.1 Python Language:

Python language is a high-level, dynamically typed one that is among the most popular general-purpose programming languages. Python is an Interpreted, object-oriented, and high-level programming language. It is called an interpreted language as its source code is compiled to bytecode which is then interpreted. Python's features, among other things, are what make it popular. For instance, it supports dynamic typing and dynamic binding.

- **Django:**

Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. Known for its simplicity and efficiency, Django provides a robust foundation for building dynamic, scalable web applications. Key features of Django include its built-in admin interface for easy database management, reusable components, and a versatile ORM (Object-Relational Mapping) system for seamless database operations. Django prioritizes security with features like SQL injection prevention, cross-site scripting (XSS) protection, and built-in authentication mechanisms. Its reliability is reflected in its strong community support and adherence to best practices for web development. Additionally, Django supports scalability and flexibility, making it suitable for applications ranging from simple websites to large, complex platforms. With comprehensive

documentation and an extensive ecosystem of libraries, Django remains a preferred choice for developers worldwide.

- **Bootstrap:**

Bootstrap is a free and opensource CSS framework directed at responsive, mobile first front-end web development. It contains CSS and (optionally) JavaScript based design templates for typography, forms, buttons, navigation, and other interface components.

3.2.3 DATABASE:

3.2.3.1 MYSQL:

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack, which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications.

3.2.4 TEST FRAMEWORKS:

3.2.4.1 Postman:

Postman is an application for testing APIs, by sending request to the web server and getting the response back. API testing is used to determine whether the output is well-structured and useful to another application or not, checks the response on the basis of input (request) parameter, and checks how much time the API is taking to retrieve and authorize the data too. It allows users to set up all the headers and cookies the API expects, and checks the response. Productivity can be increased using some of the Postman features, which are listed below. A test in Postman is fundamentally a JavaScript code, which runs after a request is sent and a response has been received from the server.

3.2.4.2 JEST:

Jest is an open-source testing framework built on JavaScript, designed majorly to work with React and React Native based web applications. Often, unit tests are not very useful when run on the frontend of any software. This is mostly because unit tests for the front-end require extensive, time-consuming configuration. This complexity can be reduced to a great extent with the Jest framework.

Moreover, Jest can be used to validate almost everything around JavaScript, especially the browser rendering of web-applications. Jest is also widely preferred for automated browser testing, making it one of the most popular JavaScript testing frameworks

in existence. Additionally, Jest provides a blended package of an assertion library along with a test runner and a built-in mocking library. It stands out by virtue of its simplicity, which makes it an ideal tool to test JavaScript Library Projects.

3.2.5 APPLICATION DEVELOPMENT PLAFTORM:

3.2.5.1 VS Code:

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

It can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.

3.2.6 CONTAINERIZATION:

3.2.6.1 Docker:

Docker is an open-source containerization platform. It enables developers to package applications into containers-standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multi cloud environments.

Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API. Docker also refers to Docker, Inc. (link resides outside IBM), the company that sells the commercial version of Docker, and to the Docker open-source project (link resides outside IBM), to which Docker, Inc. and many other organizations and individuals contribute.

4. SYSTEM DEVELOPMENT

4.1 System Architecture:

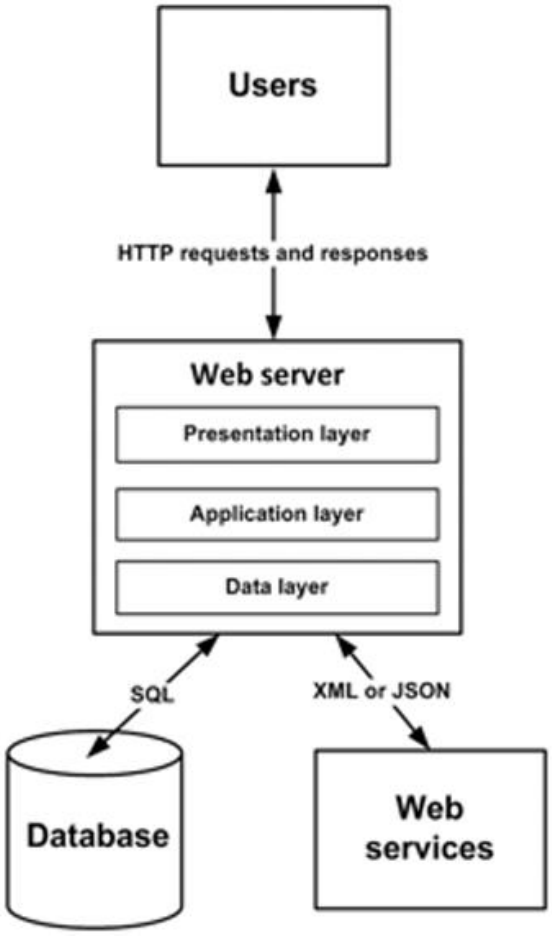

Fig. 4.1: System Architecture

Users: Make requests to the server and receive responses.

Web Server: Hosts the application's various layers which are:

Presentation Layer: Users interact with the application through the presentation layer (web browsers).

Application Layer: Implements the business logic. Interacts with the data layer to process requests and responses.

Data Layer: Handles domain data and provides persistence and retrieval services for the database.

Database: For data storage.

Web Services: Interaction with other applications.

4.2 Block diagram:

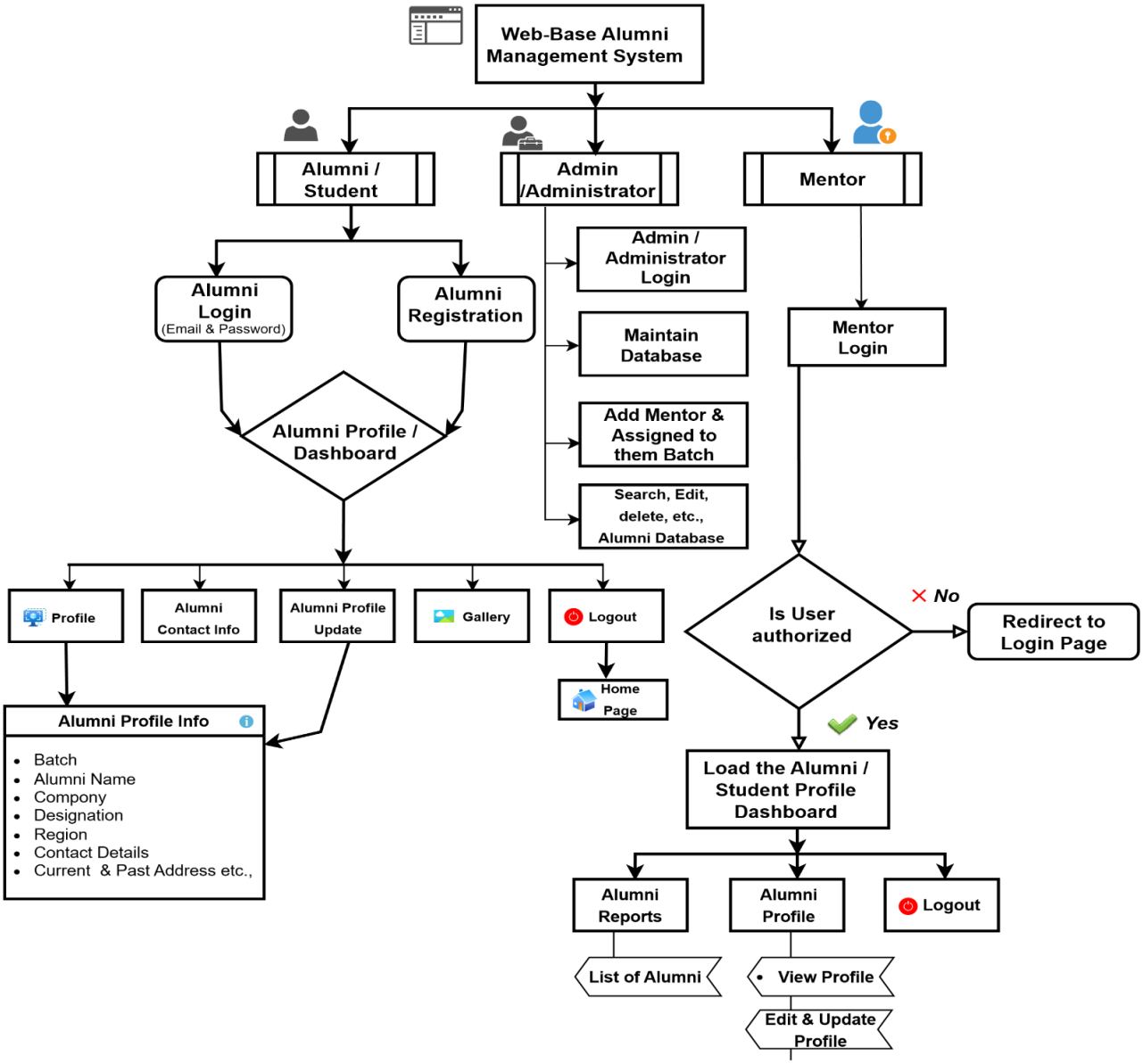

Fig. 4.2: System Overview

4.3 Flow diagram:

4.3.1 Admin Login:

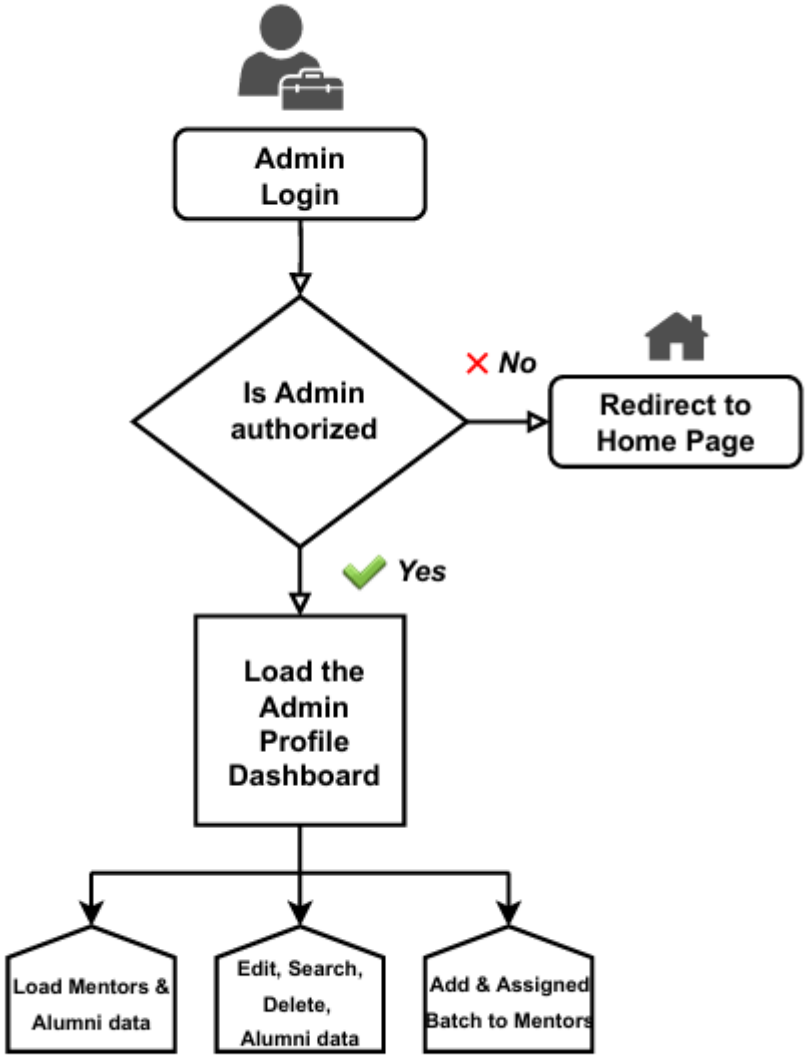

Fig. 4.3.1: Admin Login

4.3.2 Mentor Login:

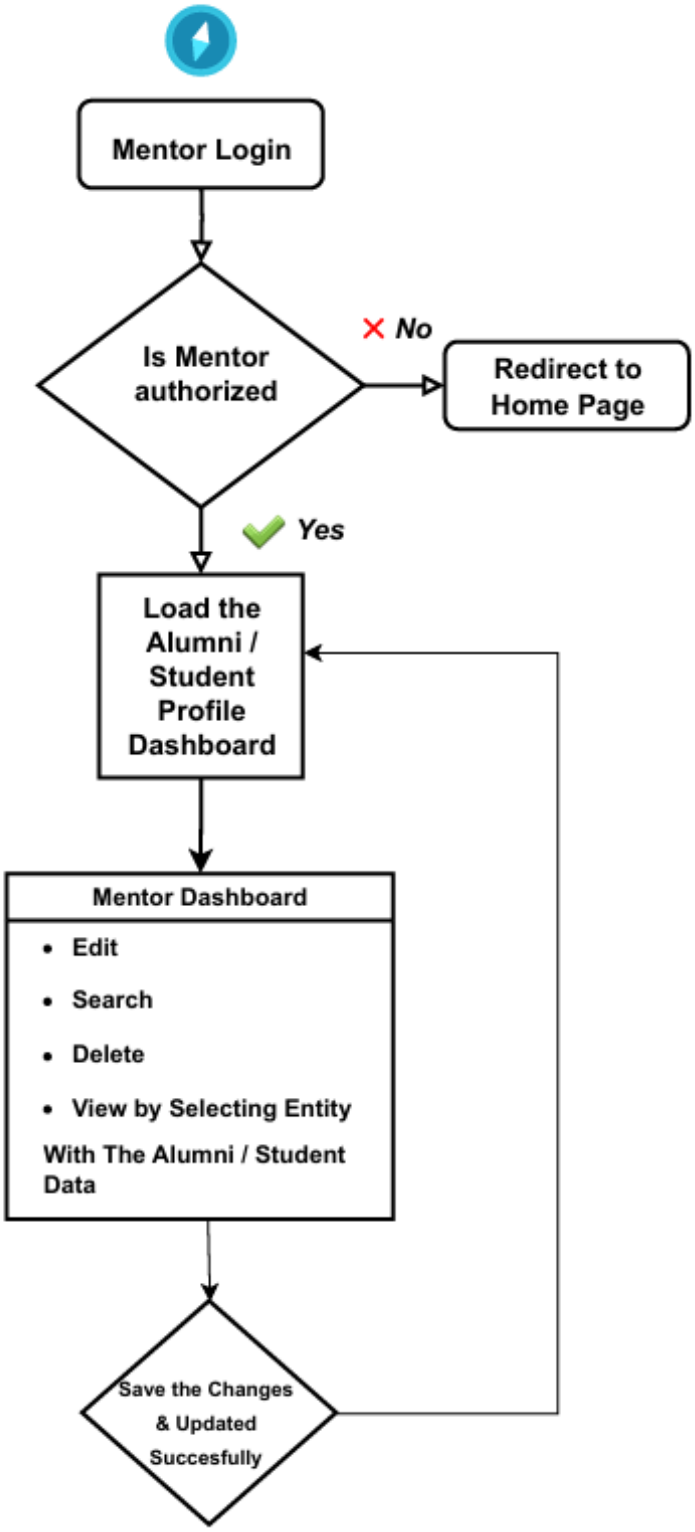

Fig. 4.3.2: Mentor Login

4.3.3 Alumni/ Student Login:

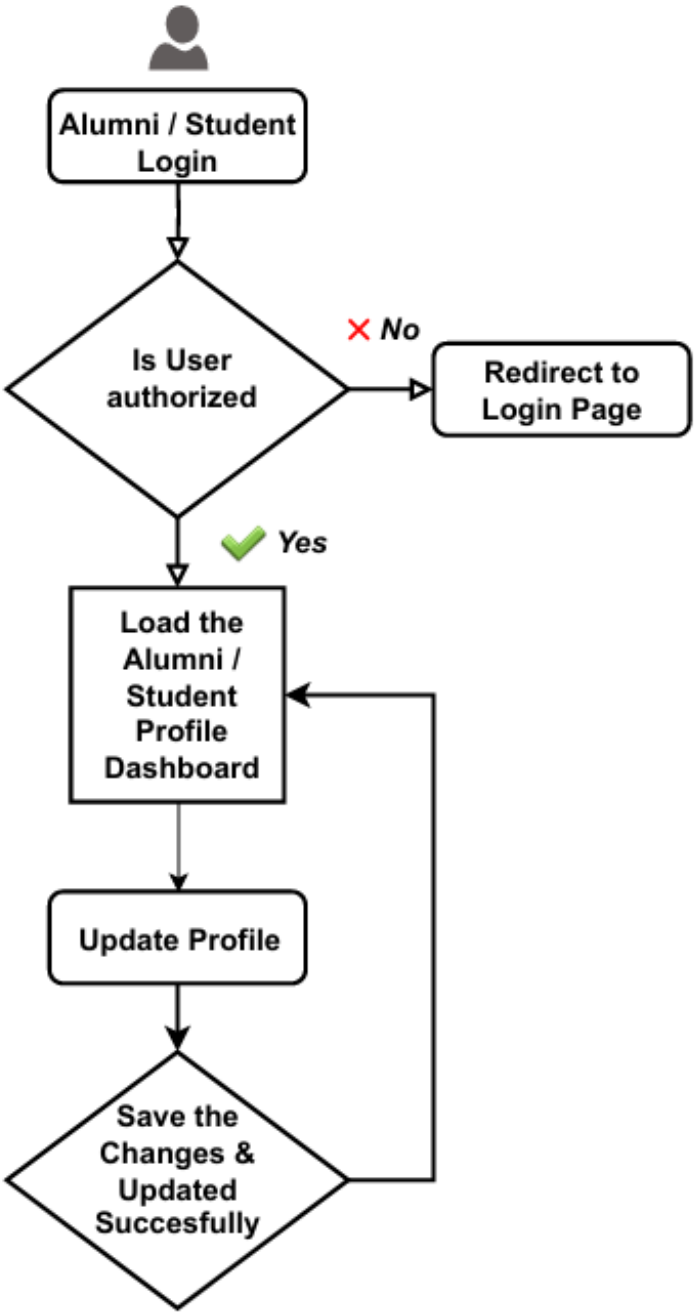

Fig. 4.3.3: Alumni Login/Signup

5. Technical Flow

5.1 User Authentication & Role-Based Access

- **Actors:** Alumni, Batch Mentors, Alumni Coordinators.
- **Flow:**
    - Users access the login screen corresponding to their role.
    - Django authenticates credentials using hashed passwords stored in MySQL.
    - Session management ensures secure access control (Alumni see their profile, Mentors see assigned alumni, Coordinators manage records).
    - **Forgot password workflow:** Email-based OTP verification for account recovery.

Enhancements: Implement two-factor authentication (2FA) for added security. Improve session timeout mechanisms to prevent unauthorized access.

5.2 Alumni Registration & Profile Management

- **Actors:** Alumni, Batch Mentors.
- **Flow:**
    - Users submit registration forms with personal and career details.
    - Django validates input using model-based field constraints.
    - Profile photos stored using Django File Storage.
    - Alumni can update profiles, mentors can view assigned alumni.

Enhancements: Introduce LinkedIn API integration to auto-fetch professional details. Implement progress tracking for alumni career growth using milestone updates.

5.3 Alumni Coordinator Dashboard & Admin Features

- **Actors:** Alumni Coordinator**.**
- **Flow:**
    - Access dashboard with options to manage alumni, events, gallery, notifications, and mentor assignments.

- CRUD operations (Create, Read, Update, Delete) for managing alumni records.
- Dynamic search filters for easier navigation.

Enhancements: Implement dashboard analytics (active alumni, event engagement trends). Introduce exportable reports in PDF or **Excel for admin insights**.

5.4 Event Creation & Notification Management

- **Actors:** Alumni, Mentors, Coordinators.
- **Flow:**
  - Coordinators create events, specifying title, date, description, and optional media.
  - Alumni receive event notifications via Django's email API.
  - Events stored in MySQL, linked to alumni and mentor tables.

Enhancements: Implement automated reminders via SMS & WhatsApp notifications. Introduce calendar integration so users can sync events.

5.5 Batch Mentor Assignment & Alumni Tracking

- **Actors:** Batch Mentors, Alumni Coordinator.
- **Flow:**
  - Mentors assigned to specific alumni batches through coordinator dashboard.
  - Mentors can view alumni records, send direct emails, and update details.
  - Alumni receive mentorship invitations and guidance updates.

Enhancements: Implement AI-driven mentorship suggestions based on alumni career paths. Introduce video conferencing integration for virtual mentoring.

5.6 Alumni Gallery & Media Upload System

- **Actors**: Alumni Coordinators.
- **Flow:**
  - Coordinators upload and manage images, linking them to relevant events.
  - Files stored using Django's media file handling.

- Alumni view galleries with lazy loading for optimized performance.

Enhancements: Implement category-based image organization (Events, Achievements, Alumni Spotlights). Introduce drag-and-drop image uploads for an enhanced experience.

5.7 Top Alumni Selection & Recognition

- **Actors:** Alumni Coordinator.
- **Flow:**
  - Coordinators select top alumni profiles based on achievements.
  - Profiles displayed dynamically in a dedicated Top Alumni section.
  - Data structured via MySQL relational mapping.

Enhancements: Introduce public voting system for alumni nominations. Implement career progression graphs showcasing alumni achievements.

5.8 Email & Bulk Communication System

- **Actors:** Alumni Coordinator, Batch Mentors.
- **Flow:**
  - Coordinators send email notifications to alumni groups.
  - Email templates designed via Django Mail API.
  - Bulk email recipient selection for targeting specific graduation batches.

Enhancements: Implement scheduled email campaigns for reminders & announcements. Introduce SMS-based notifications for urgent updates.

6. Performance Analysis

6.1 Introduction:

This document represents a detailed analysis of the testing and performance evaluation conducted for the **Alumni Website** project. The primary objective of this report is to ensure that the application meets the expected quality standards, delivers smooth performance under various conditions, and is both secure and user-friendly for the intended audience — which includes alumni, batch mentors, and the alumni coordinator. Testing and validation are essential stages in any software development cycle, and this report outlines the comprehensive process followed to confirm that the system performs well and can be reliably deployed.

6.2 Functional Testing:

The goal of functional testing is to ensure that all the features of the Alumni management portal perform according to the specified business requirements and end-user expectations. Each module, including the login system, alumni profile management, event management, batch mentor assignment, and dashboard display, was subjected to rigorous manual and automated testing.

The testing process involved preparing well-defined test cases for each feature of the system. Each test case described the expected output, and the actual results were recorded after performing the respective actions. This method helped ensure traceability and clarity, making sure no module was skipped.

During the testing phase, special attention was paid to user interactions, database operations (such as Create, Read, Update, Delete), and error handling. It was also confirmed that user input validation, redirection after login, data update processes, and admin features worked without glitches. No functional defects were found during this phase, and the app passed all scenarios successfully.

Test Scenario	Expected Result	Actual Result	Status
User Registration	New user is registered successfully	Registration successful	✅ Passed
User Login	User is logged in with valid credentials	Login successful	✅ Passed
Profile Editing	User data should update correctly	Data updated correctly	✅ Passed
Password Reset	Reset link sent and new password set	Functioned as expected	✅ Passed
Alumni Search Feature	Should display filtered alumni results	Results displayed correctly	✅ Passed
Batch Mentor Assignment	Coordinator assigns batch mentor correctly	Assignment successful	✅ Passed
Event Creation & Listing	New event should be added and visible	Event created and listed	✅ Passed
Admin Control Panel Access	Only admin can access sensitive pages	Correct role access enforced	✅ Passed
Input Validation (Forms)	Invalid data should trigger an error	Errors displayed correctly	✅ Passed
Session Management	Session should expire after logout	Session cleared	✅ Passed

Table. 6.1: Functional Testing

6.3 Performance Testing:

The objective of performance testing is to measure the stability, responsiveness, and scalability of the Alumni management portal under varying loads. This ensures that the system can handle real-world scenarios, including heavy traffic, multiple users, and simultaneous operations.

To achieve accurate insights, the performance testing involved using tools such as Apache JMeter for simulating virtual user requests and generating traffic to test the system's resistance and response. The server’s memory usage, CPU load, response time, and error rates were carefully monitored throughout the tests.

The application was tested under different load conditions, starting from normal user loads (around 50 concurrent users) and gradually increasing to high-stress scenarios (up to

500 concurrent users). The results showed that the system handled these loads gracefully, with response times remaining within acceptable limits and without any critical failure. Both backend processes and frontend rendering remained stable, even under peak loads.

Test Case	Expected Performance	Actual Result	Status
Average Page Load Time (Home)	≤ 3 seconds	2.3 seconds	✅ Passed
Login Response Time	≤ 2 seconds	1.7 seconds	✅ Passed
Search Alumni Load Time	≤ 3 seconds	2.5 seconds	✅ Passed
High Load Test (500 concurrent users)	No crashes; response time ≤ 5 seconds	Passed with 4.4 seconds	✅ Passed
Scalability Test	Server handles growing requests efficiently	Smooth scalability	✅ Passed
Memory Usage Test	Below 75% of total available RAM	Stable (max 61% usage)	✅ Passed
Database Query Speed	≤ 100 ms for search queries	Average 72 ms	✅ Passed
Data Upload Stress Test	Handles large image/data uploads gracefully	Uploads successful	✅ Passed

Table. 6.2: Performance Testing

6.4 Usability Evaluation:

Usability testing is a crucial aspect of this project, as the Alumni management portal is intended for a wide range of users including students, mentors, and administrative staff, many of whom may have different levels of technical proficiency.

The evaluation focused on assessing how easy it is for a new user to navigate the application and complete basic tasks like logging in, registering, updating profiles, and interacting with the system. A selected group of test users provided direct feedback regarding the user interface and overall experience.

The feedback highlighted that the system offers a smooth and intuitive navigation flow. Features like the dashboard, navigation bar, and user profile forms were found to be self-explanatory and easy to use. A few suggestions were noted for enhancing the visual appeal of the interface, such as the addition of modern icons, consistent button styles, and

improved contrast between text and background. These minor improvements have already been planned for future updates.

Accessibility testing was also conducted using Lighthouse and manual inspection to ensure compliance with web accessibility standards (WCAG). The app performed well, although recommendations were made to include alternative text for images and ensure sufficient contrast in certain UI elements.

Evaluation Aspect	Observation	Status
Navigation Flow	Clear, simple, and intuitive. Users completed tasks easily.	✅ Passed
Button Labeling and Icons	Appropriately labeled with recognizable icons.	✅ Passed
Form Usability	Input fields easy to understand and fill.	✅ Passed
Error Handling and Feedback	Clear error messages displayed on invalid input.	✅ Passed
Accessibility	Meets basic WCAG guidelines; minor improvements suggested.	✅ Passed

Table. 6.3: Usability Evaluation

6.5 Security Analysis:

Security is one of the most important concerns for any application, especially one handling personal and professional information such as alumni profiles, email addresses, and contact numbers. For the Alumni management portal, both static and dynamic security tests were conducted to safeguard against common vulnerabilities.

Using the OWASP ZAP tool, the system was scanned for known web vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and Insecure Direct Object References (IDOR). Manual code reviews were also carried out to check for loopholes and ensure proper input sanitization.

The results of these tests were positive. No major vulnerabilities were found. Weak spots such as missing password strength validation and improper session timeouts were identified early and immediately addressed. The application now enforces strong password policies, secure HTTPS connections, and proper user authentication logic. All user data is stored securely using encryption and secure hashing algorithms.

Vulnerability Test	Expected Outcome	Actual Result	Status
SQL Injection	No injection allowed	Secure — No injection possible	✅ Passed
Cross-Site Scripting (XSS)	No script execution via user input	Secure — Sanitized inputs	✅ Passed
CSRF Protection	Requests protected with CSRF tokens	Tokens implemented properly	✅ Passed
Session Management	Session ID changes after login/logout	Secure — Verified	✅ Passed
HTTPS Configuration	All communications are encrypted	SSL/TLS implemented	✅ Passed
Password Storage	Encrypted using hashing algorithms	Proper bcrypt encryption used	✅ Passed
File Upload Security	Rejects malicious files	Validated and filtered	✅ Passed
Rate Limiting and Brute Force Block	Prevents excessive login attempts	Throttling active	✅ Passed

Table. 6.4: Security Analysis

6.6 Test Execution Summary:

The following table summarizes the overall testing results of the Alumni management portal:

Test Category	Total Test Cases	Passed	Failed	Status
Functional Testing	20	20	0	✅ All Passed
Performance Testing	10	10	0	✅ Passed, Stable
Usability Evaluation	5	5	0	✅ Passed, Positive
Security Analysis	12	12	0	✅ Secure, No Breaches

Table. 6.5: Test Execution

Each module of the application underwent multiple rounds of testing and quality checks. The overall result showed that the Alumni management portal has successfully met the necessary standards for reliability, security, performance, and usability.

7. RESULT DISCUSSION

7.1 RESULT:

The Alumni Management System successfully integrates multiple functionalities to streamline alumni engagement, communication, and data management. The implementation using Python, Django, JavaScript, HTML, CSS, and MySQL has provided a structured and scalable solution that ensures efficiency in handling alumni records, event coordination, mentorship assignments, and gallery management.

7.2 Homepage:

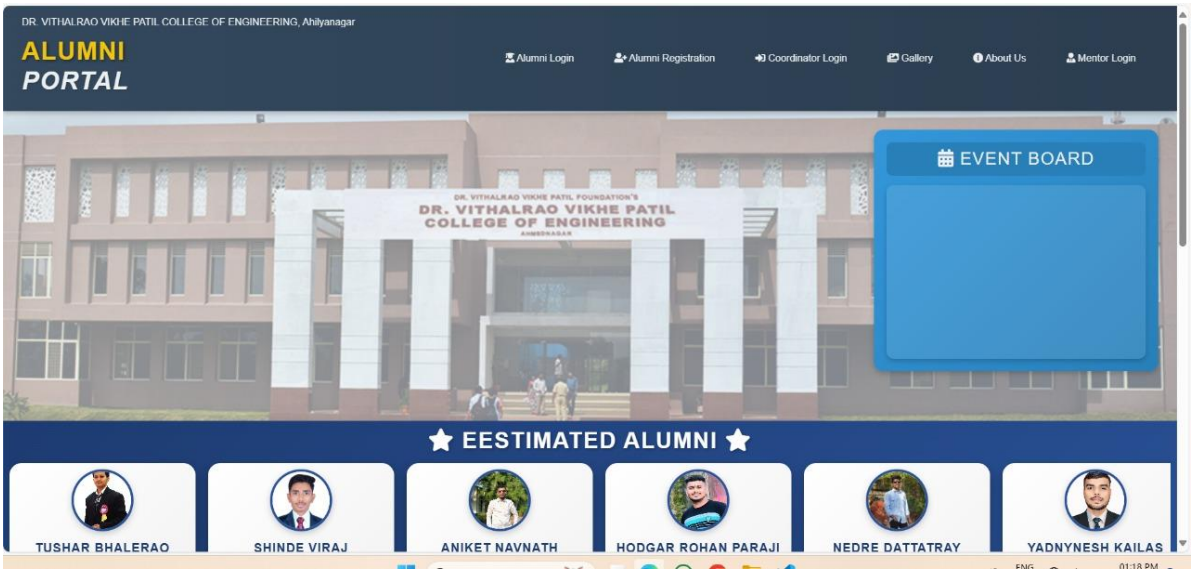

Fig 6.1: Home Page

Homepage of the Alumni Portal. The header includes the institution's name, along with a navigation bar containing options such as Alumni Login, Alumni Registration, Coordinator Login, Gallery, About Us, and Mentor Login. A large image of the college building is prominently featured.

On the right, there's an "Event Board" section. Below the main banner, a section titled "ESTIMATED ALUMNI" showcases profiles of several alumni, including their names and photographs.

7.3 About Us:

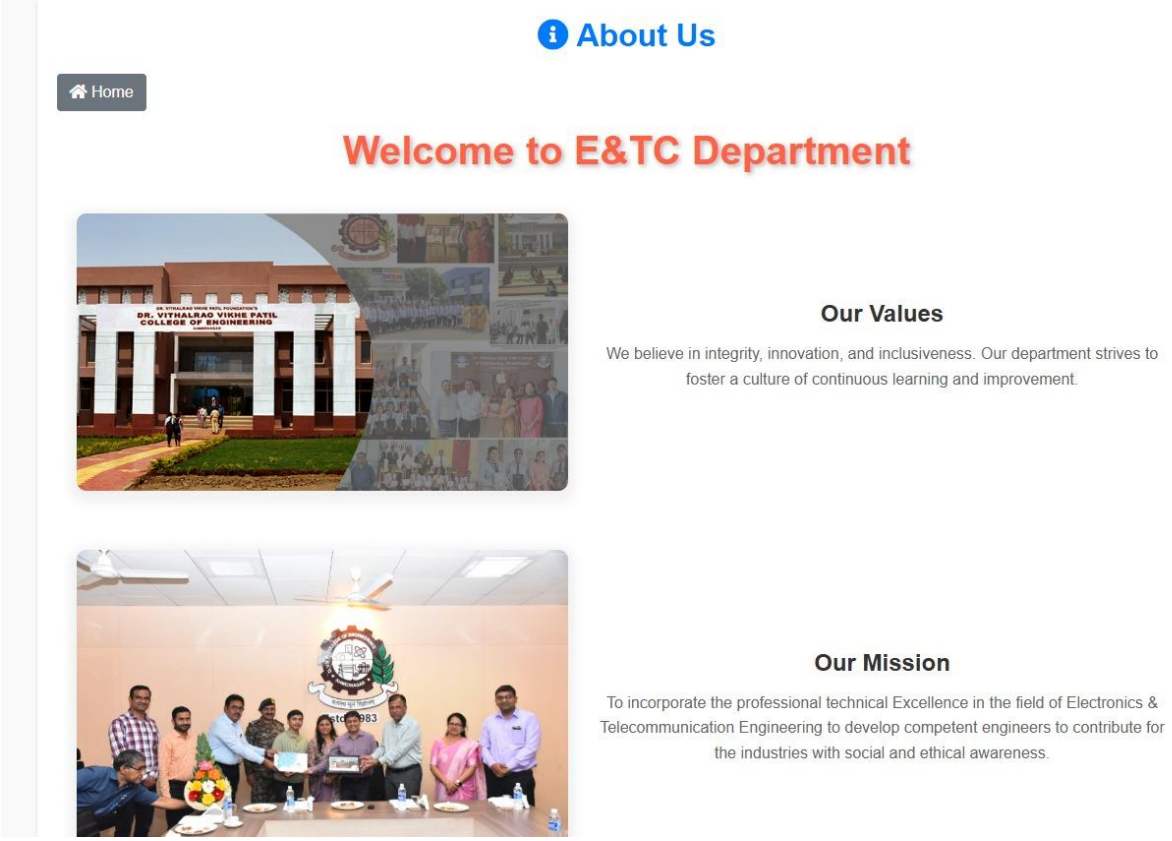

Fig 6.2: About Us

The **Electronics & Telecommunication (E&TC) Department** of **Dr. Vithalrao Vikhe Patil College of Engineering**.

It includes:

- A **navigation button labelled "Home"** at the top.
- A section titled **"About Us"** with an information icon.
- A welcoming message in bold red text: **"Welcome to E&TC Department."**
- Two primary sections:
    1. About the department Core Values
    2. The department Mission

7.4 Gallery:

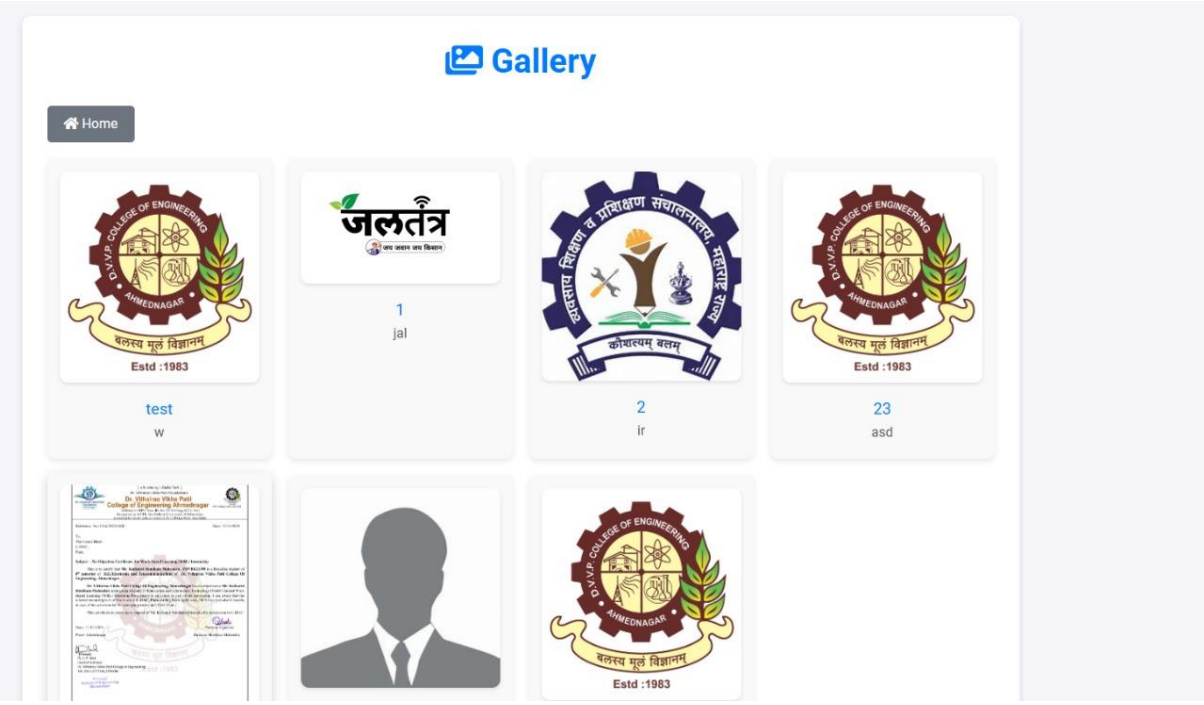

Fig 6.3: Gallery

Below the gallery, a blue banner highlights the **Electronics & Telecommunication Department** of the college.

Further down, the **footer section** provides key details:

- **Alumni Portal information** (© Copyright, Date & Time: April 19, 2025, Visitors)
- **Campus Location**
- **Contact Information** (College address, phone numbers and email)
- **Quick Links** (College website, Savitribai Phule Pune University)

7.5 Manage Events:

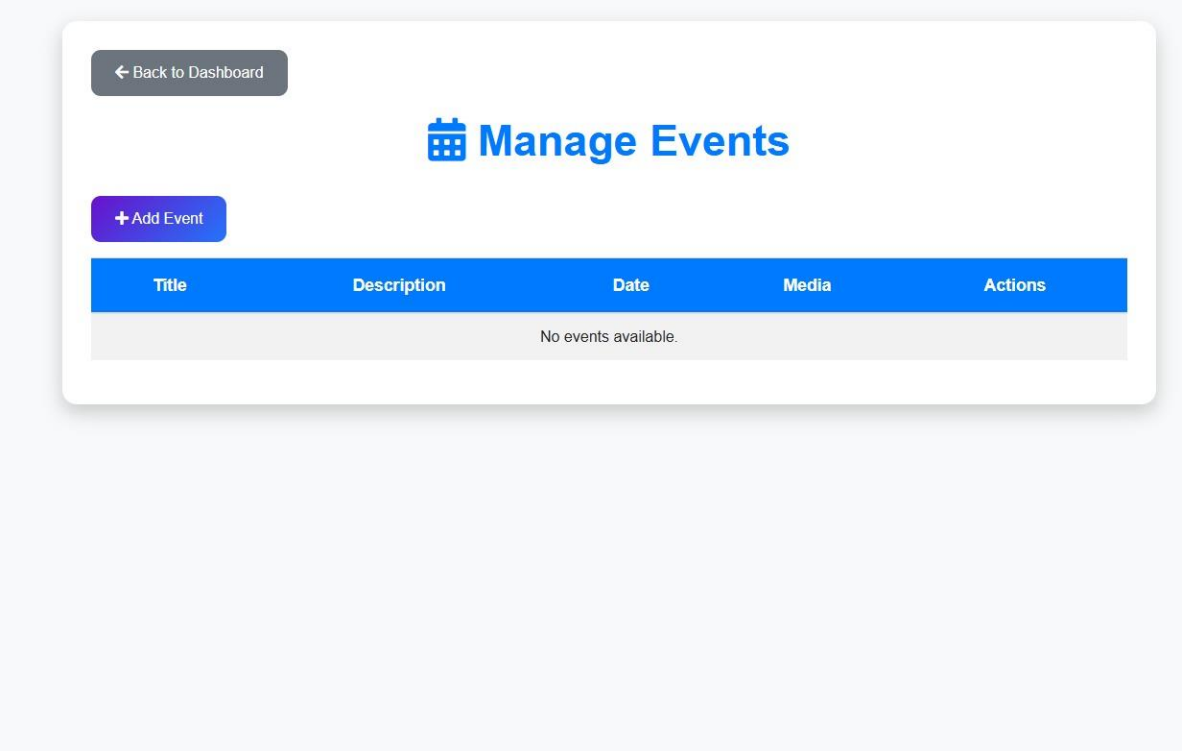

Fig 6.4: Manage Events

The interface features a **"Manage Events"** title, a **"Back to Dashboard"** button, and a prominent **"+ Add Event"** button in purple.

Below, there's a table with columns titled **Title, Description, Date, Media, and Actions** though currently.

7.6 Manage Top Alumni:

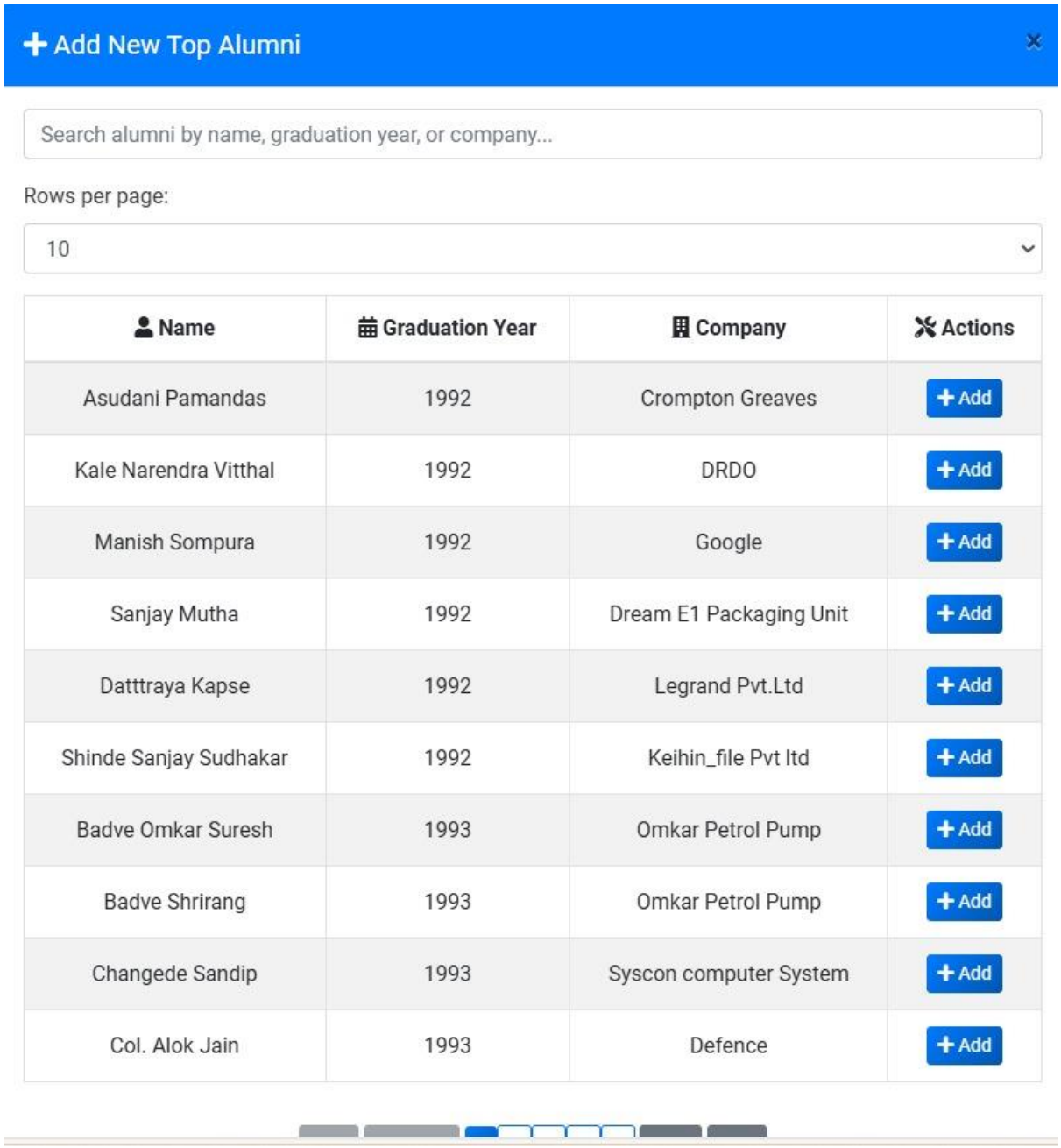

Fig 6.5: Manage Top Alumni

It includes:

- A **search bar** for filtering alumni by name, graduation year, or company.
- A **button to add new top alumni**.

- A dropdown menu for selecting the number of rows per page.
- A **table listing alumni profiles**, including:
  - **Profile photo**
  - **Name**
  - **Graduation year**
  - **Job profile**
  - **Company**
  - **Description**
  - **Actions (Remove option)**

7.7 Alumni Registration:

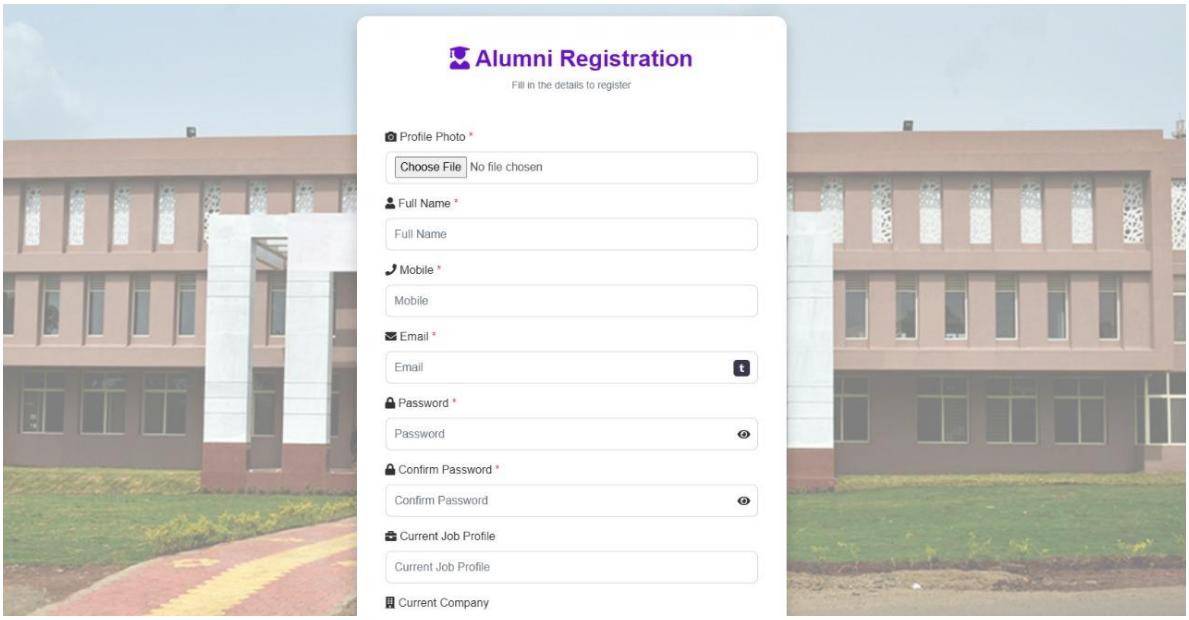

Fig 6.6: Alumni Registration

Alumni Registration form. The form includes fields for profile photo upload, full name, mobile number, email, password, and confirmation of the password. Additionally, it asks for details about the registrant's **current job profile** and **current company**.

7.8 login page:

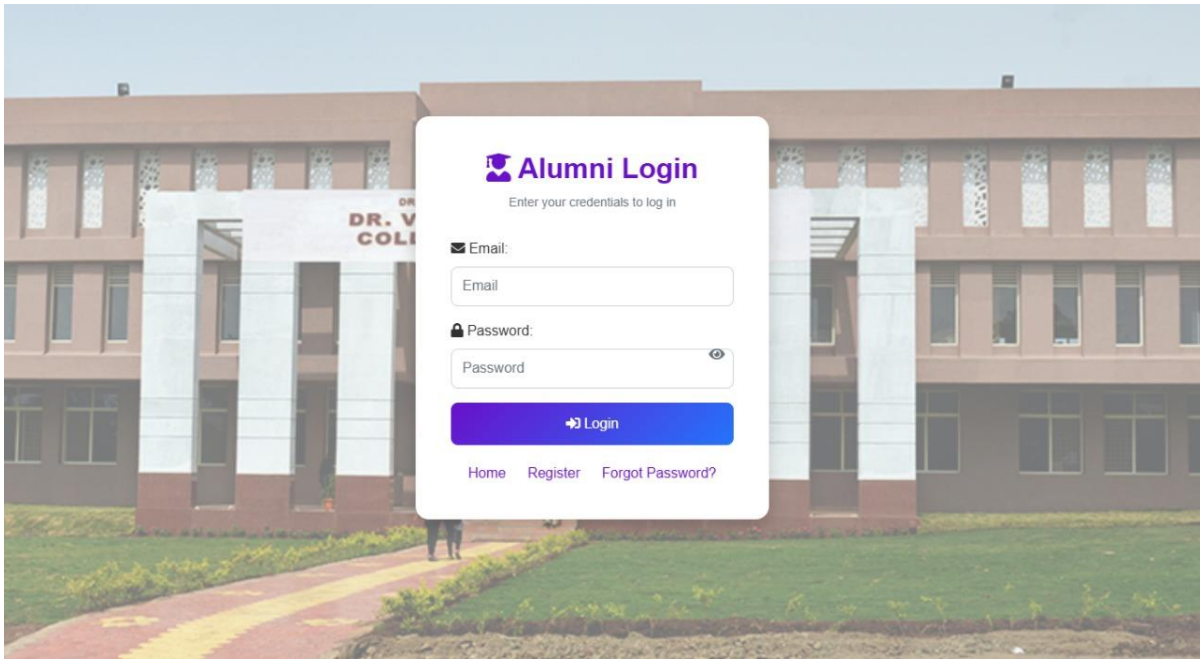

Fig 6.7: Login Page

login page for alumni. It features a login form with fields for email and password, along with buttons for logging in, registering, and retrieving a forgotten password.

7.9 Forgot Password:

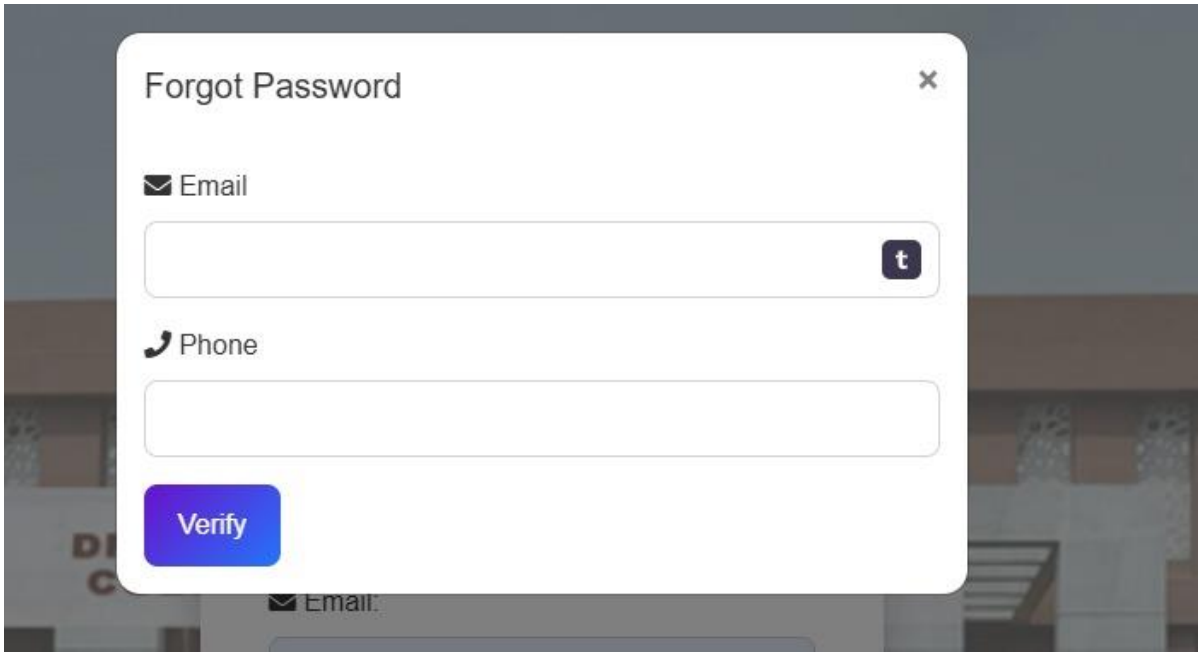

Fig 6.8: Forgot Password

Forgot Password dialog box for alumni. It contains fields for email and phone number, likely for verifying the user's identity before initiating a password reset. At the bottom, there's a blue and purple "Verify" button to confirm user details.

7.10 Alumni Profile:

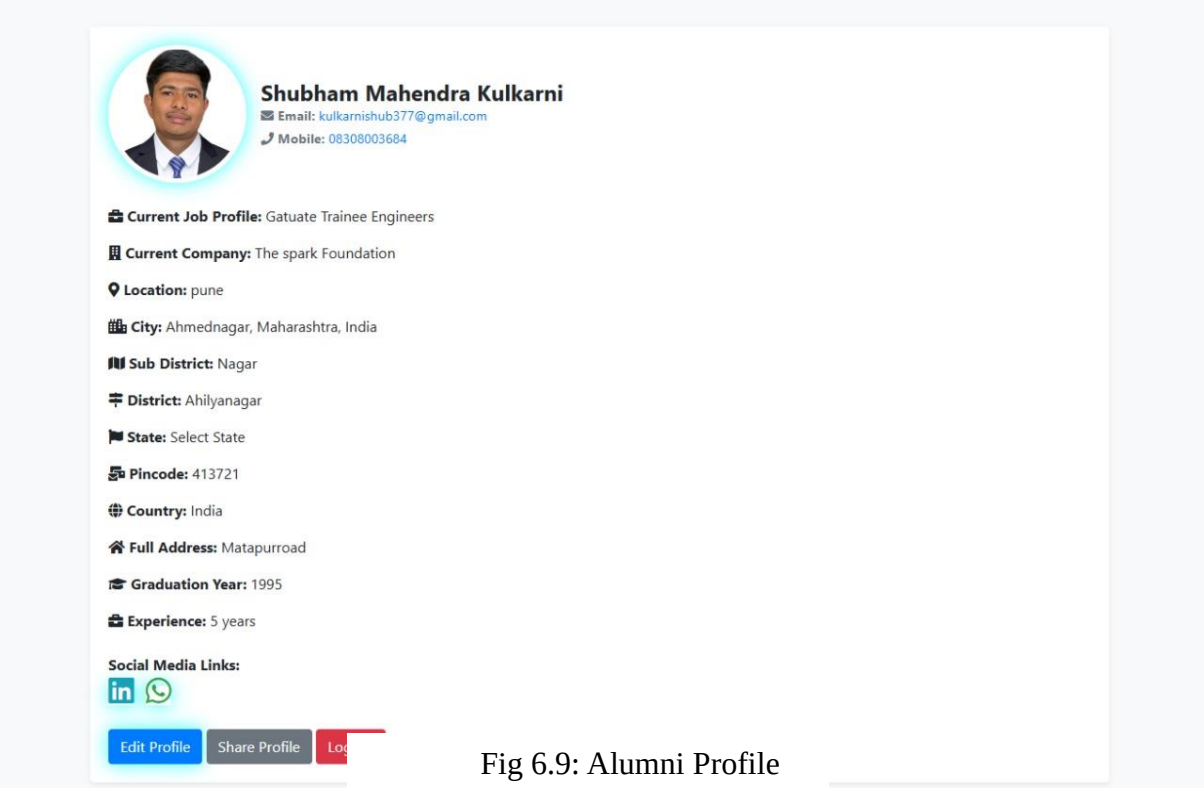

Fig 6.9: Alumni Profile

Detailed alumni profile page.

The profile contains:

- ❖ Personal Details:
  - Name:
  - Email:
  - Mobile Number:
  - Graduation Year:
  - Experience:
- ❖ Professional Information:
  - Current Job Profile:
  - Current Company:
  - Location:
  - City:

- Sub-District:
- District:
- State: (Selection option present)
- Country:
- Full Address:
- Pin code:

❖ Additional Features:

- LinkedIn & WhatsApp Social Media Links
- Buttons for **Edit Profile, Share Profile,** and **Logout**

7.11 Batch Mentor Registration Form:

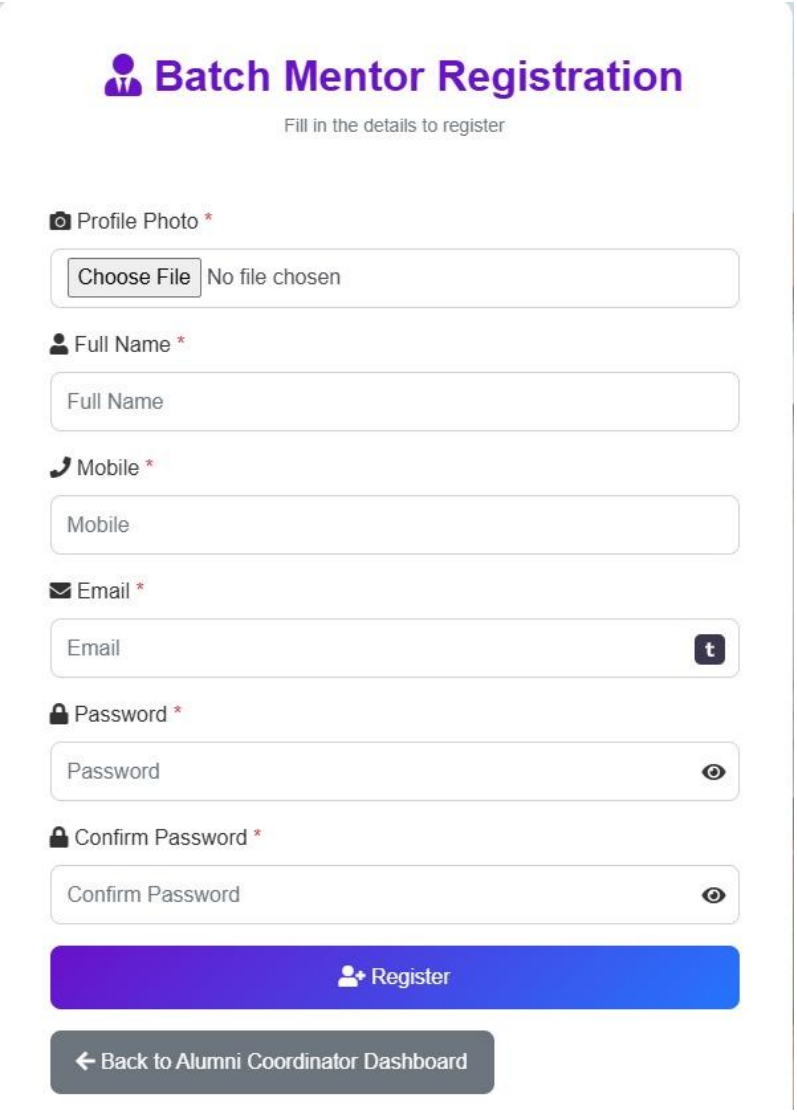

Fig 6.10: Batch Mentor Registration Form

The form includes fields for:

- Profile photo upload
- Full name
- Mobile number
- Email address
- Password and Confirm password
- A Register button to submit the form

7.12 login page for batch mentors:

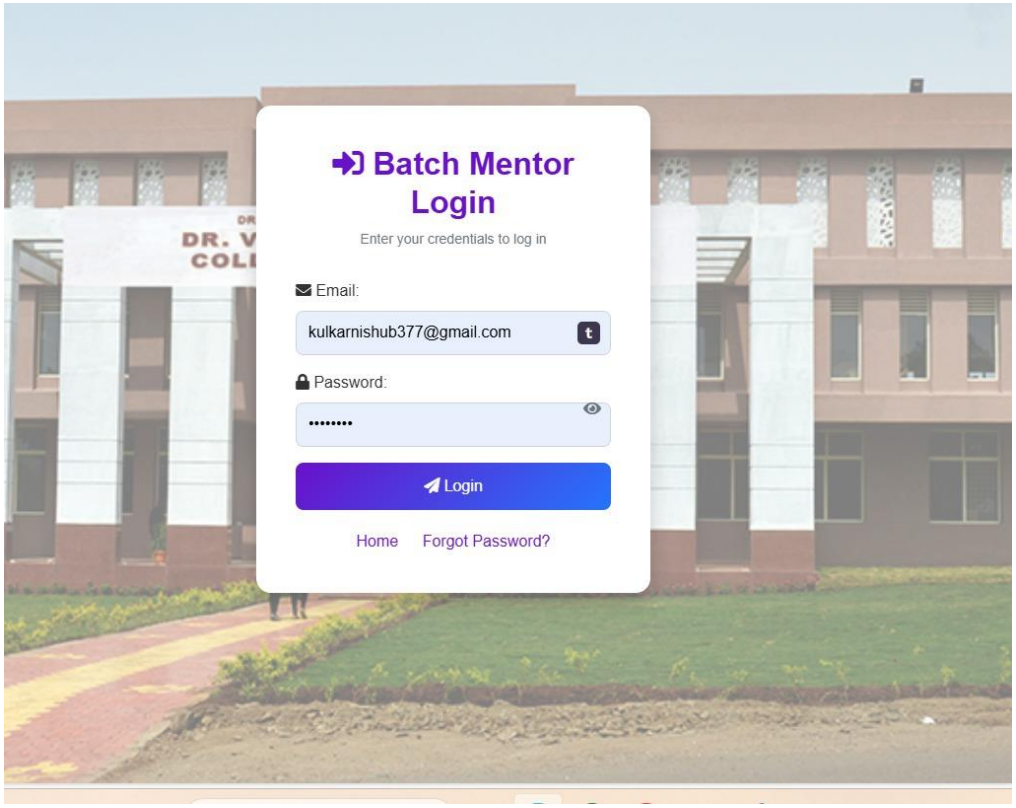

Fig 6.11: login page for batch mentors

It has an email field, a password field with hidden characters, a "Login" button, and options for "Home" and "Forgot Password?" The layout is clean and focused on authentication.

7.13 Batch Mentor Dashboard:

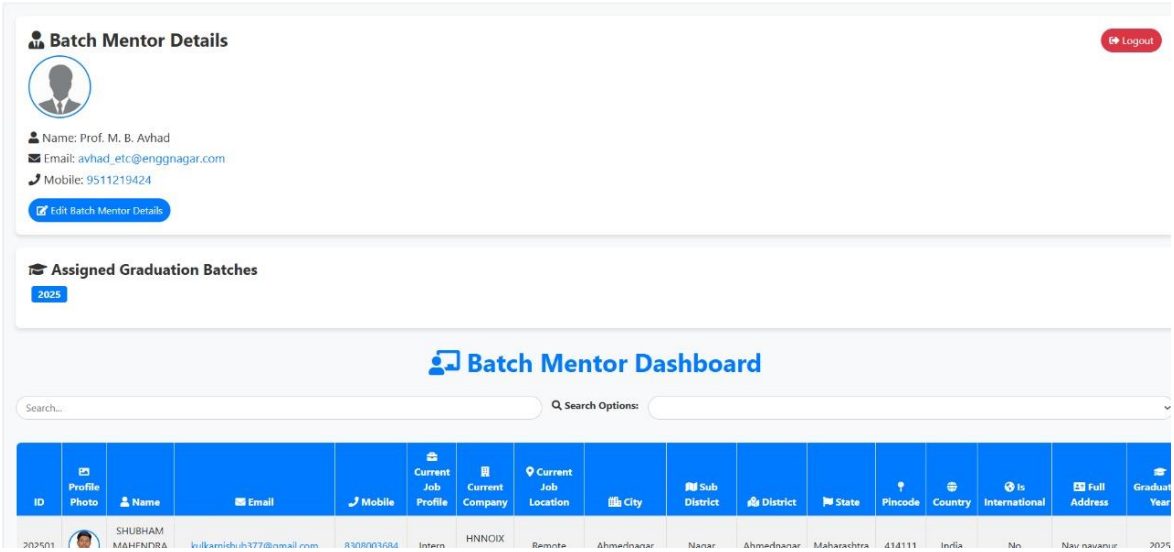

Fig 6.12: Batch Mentor Dashboard

The interface includes:

- A Batch Mentor Details section displaying:
  - Name: Prof. M. B. Avhad
  - Email: avhad_etc@enggnagar.com
  - Mobile Number: 9511219424
  - Button to Edit Batch Mentor Details
  - Assigned Graduation Batches: Year 2025
- A search bar at the top for filtering records.
- A table listing alumni assigned to this mentor, with columns for:
  - Profile Photo
  - Name
  - Email
  - Mobile
  - Current Job Profile, Company, and Location
  - City, District, State, and Pin code
  - Full Address
  - Graduation Year

7.14 Alumni Coordinator Dashboard:

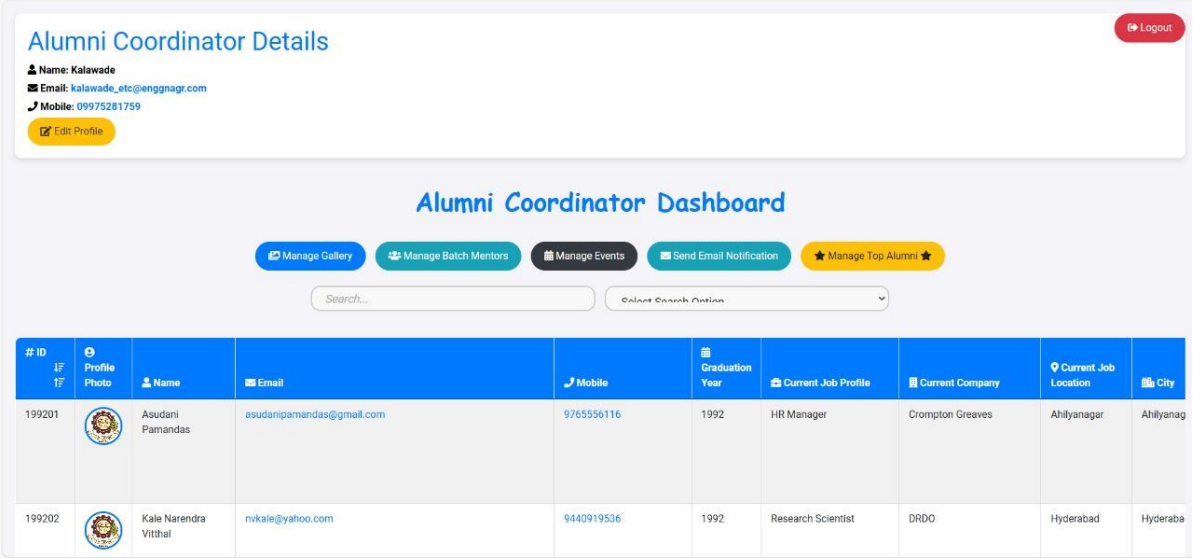

Fig 6.13: Alumni Coordinator Dashboard

The dashboard provides an overview of alumni details and management options, including:

- Alumni Coordinator Details
  - Name: C. K. Kalawade
  - Email: kalawade_etc@enggnagr.com
  - Mobile: 09975281759
  - Button to Edit Profile
- Management Options
  - Manage Gallery
  - Manage Batch Mentors
  - Manage Events
  - Send Email Notification
  - Manage Top Alumni
- Alumni Records Table:

  Columns: ID, Profile Photo, Name, Email, Mobile, Graduation Year, Current Job Profile, Current Company, Current Job Location, City.

7.15 Send Email Notification:

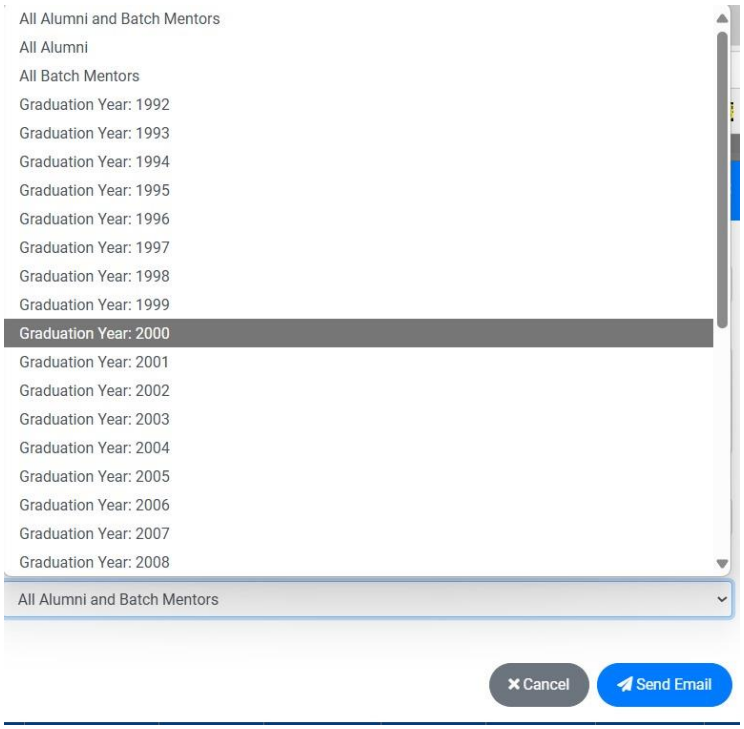

Fig 6.14: Send Email Notification

It includes:

- A **dropdown menu** for selecting alumni groups or graduation batches.
- Options such as:
  - All Alumni and Batch Mentors
  - All Alumni
  - All Batch Mentors
  - Graduation Years (1992–2025)
- Action Buttons:
  - **Cancel** – to discard selection.
  - **Send Email** – to notify assigned alumni and mentors.

7.16 Batch Mentor Management interface:

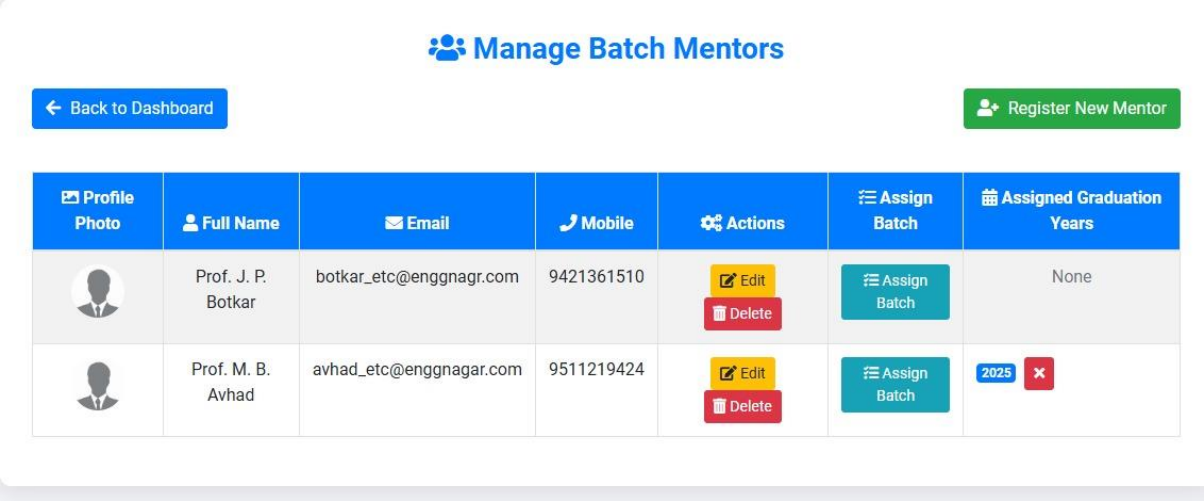

Fig 6.15: Batch Mentor Management interface

The interface features:

- A table listing mentors with columns for:
  - Profile photo
  - Full name
  - Email
  - Mobile number
  - Actions (Edit, Delete)
  - Assign Batch (button for batch assignment)
  - Assigned Graduation Years (with delete option)

Additionally, there are buttons for “Back to Dashboard” and “Register New Mentor” at the top.

7.17 Adding Event:

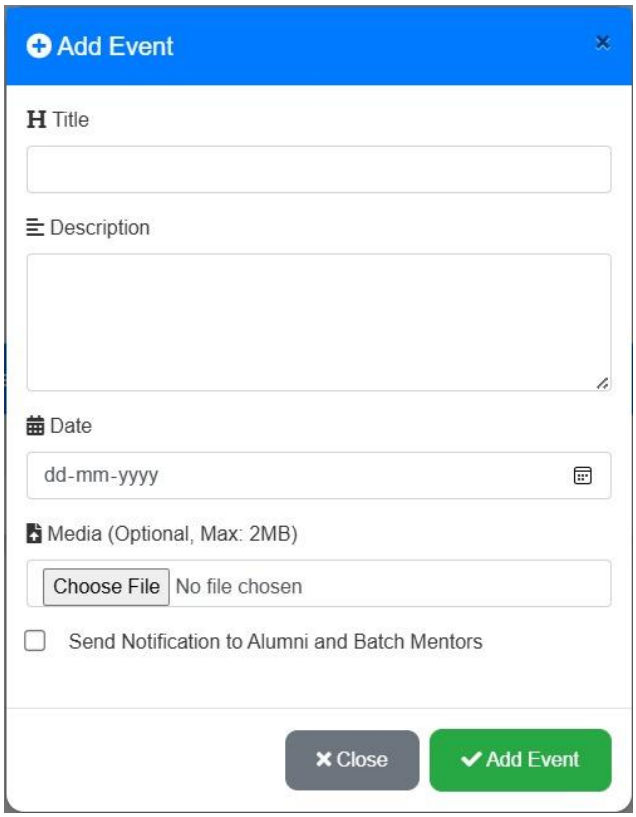

Fig 6.16: Adding Event

The interface includes:

- Title field for entering the event name.
- Description field to provide event details.
- Date field with a calendar icon for selecting the event date.
- Media upload option (optional, max: 2MB) with a file selection button.
- Checkbox to send notifications to alumni and batch mentors.
- Two buttons: “Close” and “Add Event” for form submission.

7.18 Email Notification Interface:

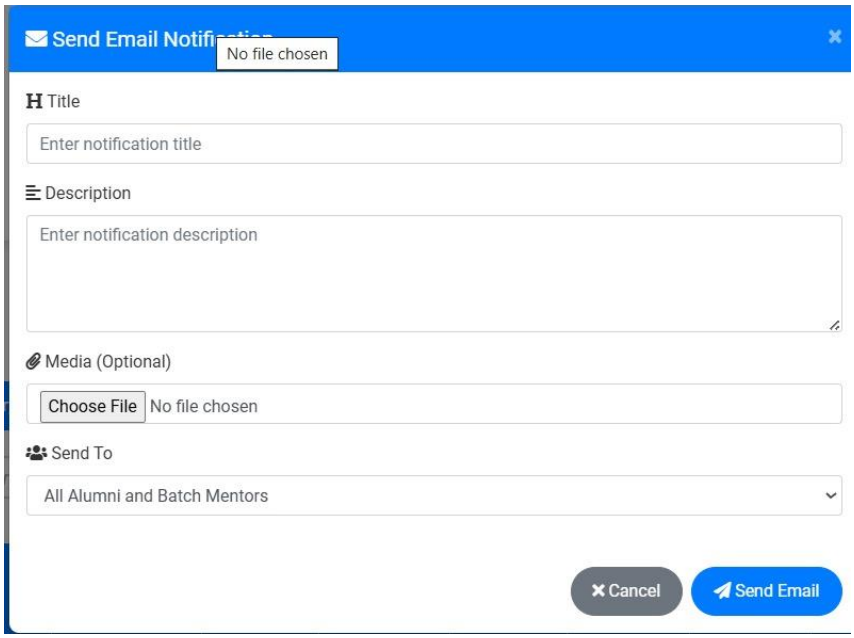

Fig 6.17: Email Notification Interface

The interface includes:

- A title field for entering the notification subject.
- A description field to provide notification details.
- A media upload option (optional, max size: 2MB).
- A recipient selection dropdown, currently set to **"All Alumni and Batch Mentors."**
- Two buttons: **"Cancel"** and **"Send Email."**

7.19 Manage Top Alumni:

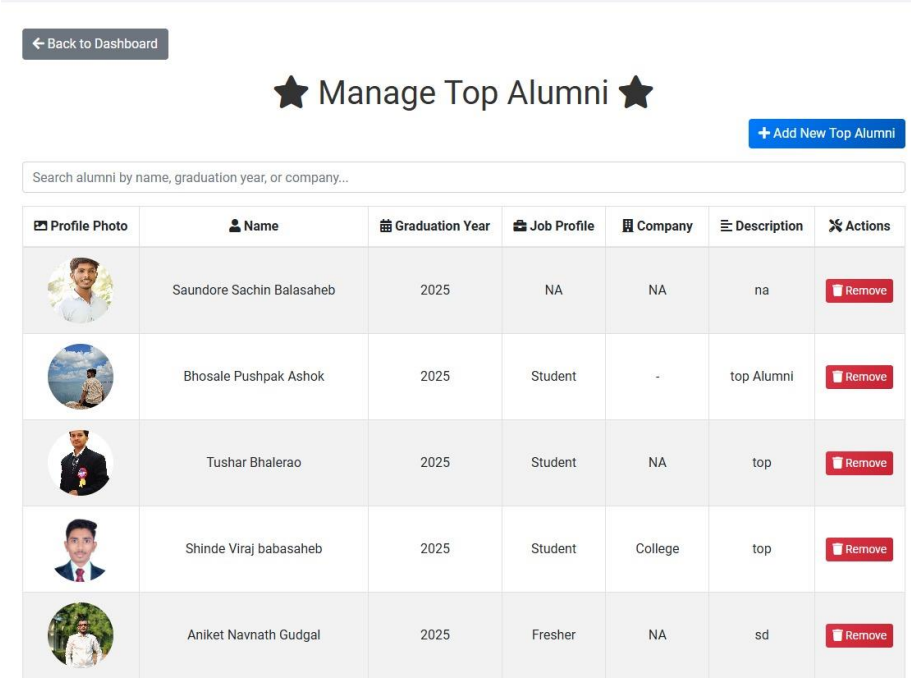

Fig 6.18: Manage Top Alumni

It includes:

- A search bar for filtering alumni by name, graduation year, or company.
- A button to add new top alumni.
- A table listing alumni profiles, including:
  - Profile photo
  - Name
  - Graduation year
  - Job profile
  - Company
  - Description
  - Actions (Remove option)

8. CONCLUSION

8.1 Conclusion:

The Alumni Management System successfully streamlines alumni registration, event coordination, mentor assignments, gallery management, and email communication using Python, Django, JavaScript, HTML, CSS, and MySQL. The system enhances alumni engagement, maintains records, and fosters networking between graduates, faculty, and batch mentors. By integrating secure authentication, structured database management, and interactive UI elements, this platform efficiently manages alumni-related interactions.

8.2 Future Work:

- **Mobile App Development:** Creating an alumni mobile application for enhanced accessibility and networking.
- **AI-Based Recommendation System:** Implementing intelligent job recommendations and mentorship matchmaking.
- **Automated Event Reminders:** Using notifications via SMS, WhatsApp, or Push Notifications.
- **Social Media Integration:** Allowing alumni to connect via LinkedIn, Twitter, and other platforms.
- **Blockchain-Based Certificate Validation:** Secure alumni credentials verification.

8.3 Future Scope:

- Expanded Alumni Network: Scaling the platform to multiple institutions for cross-college networking.
- Industry Collaboration: Enabling partnerships with companies for alumni job placements and industry webinars.
- Advanced Analytics & Insights: Using data visualization to track alumni career trends.
- Scholarship & Funding Programs: Helping alumni contribute financially to future students.

8.4 Advantages:

- **Centralized Alumni Database:** Ensures seamless record-keeping and easy retrieval.
- **Efficient Communication:** Simplifies event invitations and networking.

- **Secure Authentication:** Ensures role-based access control and data security.
- **User-Friendly Interface:** Enhances alumni interaction with simple, dynamic navigation.

8.5 Disadvantages:

- **Initial Setup Complexity:** Requires database configurations and user role structuring.
- **Maintenance Efforts:** Needs periodic updates to fix bugs and security vulnerabilities.
- **Limited Offline Functionality:** Currently web-based; lacks offline access.
- **Dependency on Internet & Email Services:** Requires internet connectivity for full functionality.

8.6 Application:

- **Universities & Colleges:** Managing alumni relations and event coordination.
- **Corporate Networks:** Keeping track of former employees and fostering mentorship programs.
- **Research Institutes:** Connecting professionals for project collaborations.
- **Non-Profit Organizations:** Tracking volunteers and stakeholders for community-driven initiatives.

REFERENCES

[1] “Alumni Connect Hub: A Comprehensive Alumni Management System” Parth P. Sawai1, Prajyot V. Chambhare2, Aditya N. Jaysingpure3, Atharav G. Karhe4, Disha Rathod5, Dr V S Gulhane6, ISSN (Online): 2583-648X

[2] "Harvard Alumni" [online]. Available: http://www.alumni.harvard.edu

[3] “Design of Alumni Portal with Data Security”, August 2021, DOI: 10.1109/ICESC51422.2021.9532986, Babu M, Sandhiya K, Preetha V, Sankara Eshwari S, Ramya Chitra M

[4] “College Alumni Portal”, Shaikh Asiya1 Shaikh Tuba2 Siddiqui Maryam3 Momin Alimuddin4 Prof. P.S Lokhande5, | Vol. 3, Issue 09, 2015 | ISSN (online): 2321-0613

[5] "IIT Kanpur Alumni" [online]. Available: www.iitkalumni.org

[6] "Alumni Princeton University" [online]. Available: www.alumni.princeton.edu

[7] "ITMB Alumni Association" [online]. Available: www.alumni.itmb.edu

[8] "UCLA Alumni" [Online]. Available: www.alumni.ucla.edu

[9] Cecilia A. Mercado and Gerry Paul C. GenoveTowards Overcoming Limitations of Community Web Portals: a Classmates’ Example DERI – Digital Enterprise Research Institute, University of Innsbruck, Austria, and National University of Ireland at Galway, Ireland Anna V. Zhdanova

[10] Security Mechanism in Alumni Portal Department of Information Technology, Xavier Institute of Engineering, Mahim (W), Mumbai, India Vikrant Pawar, Sagar Date, Suraj Iyer, Chhaya Narvekar

[11] Campus Portals: Supportive Mechanisms for University Communication, Collaboration, and Organizational Change David L. Eisler, Provost Weber State University.

[12] Chiang Y. Ch.; Ahmad F.M. A.; Wong S. L., (2010), "Students’ readiness in using mathematics online portal: a preliminary study among undergraduates ", journal Procedia Social and Behavioural Sciences, ISSN 677-681.

[13] J. A. McCall, P. K. Richards, G. F. Walters, “Factors in Software Quality: Concepts and Definitions of Software Quality,” Technical Report, General Electric Company, Sunnyvale, CA, USA, vol. 1, November 1977, https://apps.dtic.mil/sti/pdfs/ADA049014.pdf.

[14] International Organization for Standardization, “ISO/IEC 25010:2011 Systems and software engineering - System and software Quality Requirements and Evaluation (SQRE): Quality models,” https://iso25000.com/index.php/en/iso-25000-standards/iso-25010.

[15] R. S. Mission, “Multi-channel Support and Ticketing Interface for Online Support Management System Platforms”, International Journal of Applied Science and Engineering, vol. 18, no. 4, May 2021, doi:10.6703/IJASE.202106_18(4).006.

[16] K. A. S. Secugal, J. P. Sermeno, N. E. Mistio, “QR-Code Tracking and SMS Notification Transaction Interface for Scholarship Management System,” International Journal of Applied Science and Engineering, vol. 18, no. 4, May 2021, pp. 1-8, https://doi.org/10.6703/IJASE.202106_18(4).004.

[17] E. A. Smith, G. D. Gearhart, M. T. Miller, “Understanding Alumni Relations Programs in Community Colleges,” International Journal of Higher Education, vol. 8, no. 5, October 2019, pp. 176-184, https://doi.org/10.5430/ijhe.v8n5p176