

A
Project Report
On
“Alumni Hub”

AT
“SGMS Infotech LLP, Pune”

SUBMITED BY
Mr. Shubham Sanjay Sahane

GUIDE BY
Prof. Kalyani Alisetty
Dept. M.C.A.

SUBMITTED TO
SAVITRIBAI PHULE PUNE UNIVERSITY

In partial fulfilment of the award of the degree Of
Masters In Computer Application

SINHGAD INSTITUTE OF BUSINESS ADMINISTRATION AND RESEARCH
KONDHWA, PUNE-411048

During Academic Year
2022-2023

CERTIFICATE OF APPROVAL

This is to certify that the project report entitled **Alumni Hub** submitted to the Department of Computer Application, Sinhgad Institute of Business Administration and Research in partial fulfillment of the requirement for the award of the Degree of MASTER OF COMPUTER APPLICATIONS (MCA Affiliated to Savitribai Phule Pune University) is an original work carried out by,

Name: Shubham Sanjay Sahane.

Seat No: 1632

The matter embodied in this project is a genuine work done by the student and has been certified by the following internal and external examiners deputed by Savitribai Phule Pune University.

Internal Examiner

External Examiner



SGMS INFOTECH LLP

Office No.4, SN 26, ABC Junction,
Front of Akurdi Railway Station,
Nigadi Pradhikaran, Pune - 411044
Email ID : contact@sgmsinfotech.com
Web : www.sgmsinfotech.com
Contact : +91 8275329929

Ref. No. : SGMS/2023/JI0031

Date : 01 / 02 / 2023

Internship Joining Letter

To,

Sinhgad Institute of Business Administration and Research,
Sinhgad, Pune 411048

Subject: Confirmation of the Project work and Accepting your Student as Project Interns.

Respected Sir/Madam,

It to bring your kind notice that Mr. Shubham Sanjay Sahane has been short listed in our organization as a Project Interns on Java Platform for a period of 01 February 2023 To 31 July 2023, within the period, intern is expected to follow the entire stipulated rule and regulation put forwarded by the Organization.

During the training process, the Project trainees are subjected to vital information, which is very crucial and personal to the organization. The secrecy of the information should be protected at every stage. The information provided by the trainee is consider to be the soul proprietary of the SGMS Infotech LLP, and if founded to be leaked under any circumstances, Project trainee is subjected to face the legal action taken by the legal cell of SGMS Infotech LLP.

If the trainee accepts the offer, within the specified legal guideline of SGMS Infotech LLP, Please attest the same with your initials along with the initials of the trainee and return the same in duplicate to organization at the time of him/her joining the project work. The trainee also should submit curriculum vitae with all necessary details for our official reference and attendance will be send to for the academic references.



SGMS Infotech LLP



Sinhgad Institutes

SINHGAD TECHNICAL EDUCATION SOCIETY'S
SINHGAD INSTITUTE OF BUSINESS ADMINISTRATION & RESEARCH

(Approved by AICTE, Recognized by Government of Maharashtra,
Affiliated to Savitribai Phule Pune University, Accredited by NAAC)
Near PMC Octroi Post, Kondhwa - Saswad Road, Kondhwa (Bk), Pune - 411048

Phone : +91 20 26934543/26933635/20 26933633 Email : directormca_sibar@sinhgad.edu Website : www.sinhgad.edu



Prof. M. N. Navale

M.E. (Elect.), MIE, MBA

FOUNDER PRESIDENT

Dr. (Mrs.) Sunanda M. Navale

B.A, M.P.M., Ph. D.

FOUNDER SECRETARY

Dr. Netra Patil

MCA, Ph. D. (Computer Mgmt.)

DIRECTOR

CERTIFICATE OF ORIGINALITY

This is to certify that the project report entitled

Submitted to the Department of Computer Applications, Sinhgad Institute of Business Administration and Research in partial fulfillment of the requirement for the award of the degree of MASTER OF COMPUTER APPLICATIONS (MCA Affiliated to Savitribai Phule Pune University), is an original work carried out by

Mr/Ms Exam No.....

under my guidance.

The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this Organization or to any other University/ Organization for the fulfillment of the requirement of any course of study.

Signature of the Student :

Name of the student :

Signature of the Guide :

Name and Designation of the Guide :

Signature of Director-MCA

Dr. Netra Patil :

Date :

INDEX

Chapter No	Details	Page
1	Introduction	
1.1	Company Profile	
1.2	Abstract	
1.3	Existing System and Need for System	
1.4	Scope of System	
1.5	Operating Environment-Hardware and Software	
1.6	Brief Description of Technology Used	
2	Proposed System	
2.1	Study of Similar Systems(If required research paper can be included)	
2.2	Feasibility Study	
2.3	Objectives of Proposed System	
2.4	Users of System	
3	Analysis and Design	
3.1	System Requirements (Functional and Non-Functional requirements)	
3.2	Entity Relationship Diagram(ERD)	
3.3	Object Diagram	
3.4	Class Diagram	
3.5	Use Case Diagrams	
3.6	Sequence Diagram	
3.7	Collaboration Diagram	
3.8	Activity Diagram	
3.9	State Chart Diagram	
3.10	Interface Diagram	
3.11	Deployment Diagram	
3.12	Module Hierarchy Diagram	
3.13	Website Map Diagram (In Case of Web Application)	
3.14	Module Specifications	
3.15	Table Structure	
3.16	Sample Input and Output Screens (Screens must have valid data. All Reports must have at-least 5 valid records.)	
4	Coding	
4.1	Code snippets	

5	Testing	
5.1	Test Strategy	
5.2	Unit Testing	
5.3	Acceptance Test Plan	
5.4	Test Case	
5.5	Defect report/Test Log	
6	Limitations of Proposed System	
7	Proposed Enhancements	
8	Conclusion	
9	Bibliography	
10	Appendix– Cost sheet	
11	User Manual (Purpose ,Overview, Roles,)	

CHAPTER 1

Introduction

Chapter 1: Introduction

1.1 Company Profile :

SGMS INFOTECH LLP works on Web Application Project, Academic Project & Mobile Application Development, also develop the Connection in Hardware and Software Project and Provide the Best Service to client, also Provide Internship to Computer Sciences and Information Technologies Branches Student like B.E, MCA, MCS and BCA.

Rather than business we develop relation. So for any software and web site development contact us and we provide you satisfactory services. We are motivating our distribution channel aggressively by serving your engineering need in sharp time, which protects your time management.

You give us the requirements we cater it promptly & punctually, quality & price is in our veins which is most competitive, you can contented. Just call us we are very near to your voice. Experience our services & then define us, whether we fit to your esteemed organization or not. We have excellent marketing network with technical support. We also deal with application development were we develop different application which help you to do work efficiently and effectively.

Services:

1. Website Design & development
2. Software Development
3. Digital Marketing
4. Search Engine Optimization
5. E-commerce Web Design
6. Android App Development

About Internship:

SGMS INFOTECH LLP offers internship to MCA, MCS students to give them an exposure to real time work environment by letting them work on live projects. Objective of SGMS INFOTECH LLP Internship Program is to assist MCA, MCS graduates in enhancing their overall skills set to improve their employability level to match industry expectations. SGMS INFOTECH LLP is offering internship programs to MCA, MCS and Engineering students, which will be totally a practical exposure to the students. Topics Covered Experience of

working on live projects. The internship program is designed to enable interns work on live projects to give them an opportunity to apply the theoretical knowledge they have gained in the classroom. Interns get to see the immediate impact of programming tools and techniques applied by them who should attend

Real time trainees :

Always the employer prefers to go for a person who has an experience, because, by hiring an inexperienced person the company has to incur an additional cost on training of the candidate, so always they try to avoid going for a fresher. To remove this barrier for the fresher's SGM SOURCE INFOTECH is giving them an opportunity to become a real time trainee in the best and top most companies.

Exposure to current technologies : In today's world, academic knowledge alone is not sufficient to grow and excel in life; you need something more which adds up to your qualification. As we already told, you will be treated as a real time trainee; you will be exposed to the latest technologies in the market, which will give you a better chance of placement.

Corporate culture : Along with the knowledge and skills, it is very important to know the corporate culture that you have to follow in the corporate world. Even though you are not an employee of the company, you will be working with the employees of the company through which you can learn about the corporate culture and the organization structure.

Selection Process:

- Aptitude Test
- Technical Round

Contact Person:

Mr. Ritesh Surange

Director, SGMS Infotech LLP

Mob No: +91 8275329929

Email ID: sgmsinfotech@gmail.com

1.2 Project Abstract :

Alumni Hub is a system designed to connect alumni of a particular educational institution, providing them with a platform to network, share information, and stay connected with each other. The system aims to create a community that fosters collaboration and support among alumni, as well as encourage engagement and participation in the institution's events and activities.

The problem that the Alumni Hub system aims to solve is the lack of a centralized platform that allows alumni to connect with each other and stay engaged with their Institute. Currently, alumni may have to rely on social media platforms or personal networks to keep in touch with each other, which can be inefficient and may not allow for easy access to relevant information about the institution. Additionally, alumni may not be aware of or have easy access to the institution's events and activities, which can limit their participation and engagement.

To address these challenges, the Alumni Hub system will provide alumni with a platform that enables them to connect with each other, share information and experiences, and stay informed about institution-related news and events. The system will also provide the institution with a way to engage with its alumni and create a sense of community among them.

The Alumni Hub system will be implemented using Java as the programming language. The system will utilize Java-based web technologies such as Java Server Pages (JSP) and Java Servlets to build the user interface and handle user requests. The system will also use Java Database Connectivity (JDBC) to interact with the database, which will store information about alumni and institution-related events and activities.

Along with all this system has other features like arranging sessions for current students, alumni also post job vacancies in the company they are working to help current students and students can also connect with them to get guidance. The system also provides features of gallery, events which are scheduled in the institute.

1.3 Existing System and Need of System :

The current system for alumni engagement at our educational institution lacks a centralized platform for alumni to connect and stay engaged with their institute. Alumni face several challenges in keeping in touch with each other and the institution's activities, leading to limited collaboration and interaction. The absence of a dedicated system hinders the establishment of a strong and active alumni community.

Currently, alumni rely on social media platforms and personal networks to connect with fellow alumni. However, this approach is inefficient, as it lacks organization and does not provide easy access to relevant information about the institution. Alumni may miss out on crucial updates, events, and opportunities due to the scattered nature of communication channels.

Furthermore, the current system does not offer any specific features for alumni to interact with current students or teachers. Alumni are unable to share their experiences, offer mentorship, or participate in academic discussions, depriving students of valuable insights and guidance.

Another limitation of the existing system is the lack of a centralized event and job posting mechanism. Alumni interested in attending events or seeking job opportunities within the institution's network have no dedicated platform to access such information.

Additionally, there is no central repository for sharing photo galleries or organizing sessions on various topics of interest. The absence of such features restricts the alumni community's ability to collaborate and engage in meaningful discussions and activities.

Furthermore, the current system lacks an efficient way to distribute newsletters to alumni, current students, and teachers. Consequently, important updates, institutional news, and achievements often go unnoticed.

In summary, the existing system faces the following challenges:

- Lack of a centralized platform for alumni to connect and stay engaged.
- Inefficient reliance on social media and personal networks for communication.
- Limited interaction between alumni and current students or teachers.
- Absence of a centralized event and job posting mechanism.
- No dedicated space for sharing photo galleries or arranging sessions.
- Inefficient distribution of newsletters.

These shortcomings highlight the urgent need for the "Alum Hub" system to address these challenges and create a robust and thriving alumni community. The proposed system aims to overcome the limitations of the existing system by providing a comprehensive platform for alumni networking, engagement, and collaboration with the institution and its current members.

1.4 Scope of system:

The scope of the "Alum Hub" system have a wide range of features and functionalities designed to create a comprehensive platform for alumni engagement and networking. The system aims to connect alumni of our educational institution, providing them with a user-friendly and centralized platform to interact, share information, and stay connected with each other and the institution. The following sections outline the key components and functionalities within the scope of the system:

1. Alumni:

- Alumni Profile Creation: The system will allow alumni to create personalized profiles with relevant information such as contact details, professional experience, company they are working and other personal details.

- Search: A searchable directory will enable alumni to find and connect with fellow alumni based on various criteria like location, industry, Skill, Profession.

2. Contact with Teachers and Current Students:

- Contact Teachers: Alumni will have the opportunity to connect with current and retired teachers to facilitate knowledge exchange, mentorship, and appreciation.

- Contact Current Students: The system will enable alumni to interact with current students, offering guidance, sharing experiences, and nurturing a sense of support and mentorship.

3. Event and Job Postings:

- Event Postings: Alumni can post and view upcoming events organized by the institution, alumni chapters, or other alumni..

- Job Postings: Alumni can share job opportunities within their professional network, promoting career growth and networking among members.

4. Gallery of Photos and Memories:

- Photo Sharing: The system will allow alumni to upload and share photos related to various events, gatherings, and memorable moments during their time at the institution.

5. Organize Sessions and Workshops:

- Session Creation: Alumni can organize sessions, workshops, or webinars on specific topics of interest and invite other alumni, students, or teachers to participate.

- Knowledge Sharing: These sessions will provide a platform for knowledge exchange and skill development within the alumni community.

6. Tracking Session Participation:

- Session Participation: Alumni can view the list of students who have joined the sessions, encouraging engagement and fostering a sense of community.

7. Newsletter Distribution:

- Newsletter Management: The system will facilitate the creation and distribution of newsletters containing updates on institutional activities, alumni achievements, and upcoming events to all registered users, including alumni, teachers, and current students.

8. Admin Panel:

- Administrator Control: An admin panel will be accessible to authorized personnel to manage user accounts, monitor system activities, and moderate content.

The "Alum Hub" system will be implemented using Java-based web technologies, including Java Server Pages (JSP), Java Servlets, and Java Database Connectivity (JDBC). It will utilize a database to store user information, event details, job postings, and other relevant data. The

system's interface will be intuitive, ensuring ease of use for all users, irrespective of their technical proficiency.

The scope of the "Alum Connect" system is focused on fostering a strong and active alumni community, enhancing collaboration between alumni, teachers, and current students, and encouraging engagement in the institution's events and activities. By providing a centralized platform for alumni networking and interaction, the system aims to strengthen the bond between alumni and the educational institution, fostering a lifelong connection and a sense of pride in being part of the institution's legacy.

1.5 Operating Environment-Hardware and Software:

The operating environment for deploying the "Alum Connect" system includes both hardware and software components. Here's an overview of the required hardware and software:

Hardware Requirements:

1. Server: A dedicated server is needed to host the web application and manage the database. The server should have sufficient processing power, memory, and storage to handle the expected user traffic and data storage requirements.
2. Storage: Adequate storage space is required to store user data, event details, job postings, photos, and other information.
3. Network Infrastructure: A stable and reliable network connection is essential to ensure smooth communication between the server and users.
4. Backup and Redundancy: Implementing a backup strategy and redundancy measures is crucial to safeguard data and ensure system availability in case of hardware failures.

Software Requirements:

1. Operating System: The server should run on a compatible and secure operating system. Common choices include Linux distributions like Ubuntu, CentOS, or Red Hat Enterprise Linux.
2. Web Server: An HTTP server software is needed to handle web requests and serve the web application to users. Apache HTTP Server or Nginx are popular options.
3. Java Development Kit (JDK): The JDK is required to compile and run the Java-based web application. Ensure that the server has the appropriate version of JDK installed.
4. Java Servlet Container: A Java Servlet container, like Apache Tomcat or Eclipse Jetty, is necessary to deploy and manage Java Servlets and Java Server Pages (JSP).
5. Database Management System: Choose a reliable and scalable database management system (DBMS) to store user data, event details, job postings, and other information. Common options include MySQL 5.7
6. Version Control System: Implement a version control system like Git to track changes to the source code and collaborate with a development team effectively.
7. Security Software: Install security software and firewalls to protect the server from potential threats and unauthorized access.
8. SSL Certificate: Implement an SSL certificate to enable secure communication over HTTPS, ensuring data encryption during transmission.

9. Email Server: Set up an email server or integrate with a third-party email service to facilitate email communication, including newsletter distribution and account verification.
e.g. Scala (SMTP Server).
10. Monitoring and Logging Tools: Employ monitoring and logging tools to track system performance, identify issues, and analyze user behavior.

Development Tools:

1. Integrated Development Environment (IDE): Use a Java IDE such as Eclipse (Kepler), IntelliJ IDEA, or NetBeans for developing and debugging the web application.
2. Build Tools: Utilize build tools like Apache Maven or Gradle to manage dependencies and build the project or one can manage dependencies manually.

Client:

Any Latest Browser

It is essential to ensure that the hardware and software components meet the system's performance, security, and scalability requirements. Regular maintenance and updates are also necessary to keep the system secure and up-to-date with the latest software patches and enhancements. Additionally, consider utilizing cloud-based services for scalability and redundancy, as they offer flexibility and ease of management.

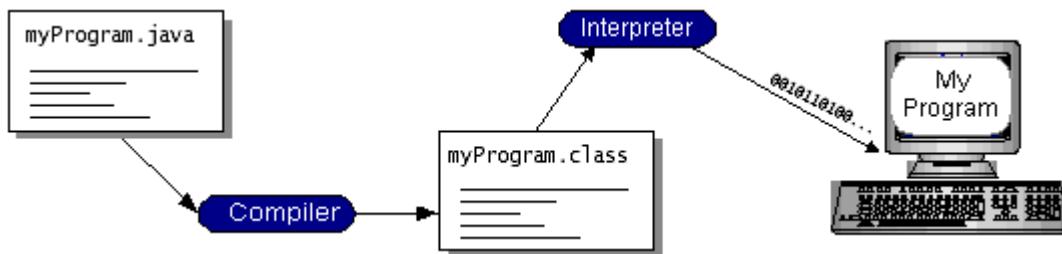
1.6 Brief Technology Used :

The "Alum Hub" system leverages an advanced technology stack to create a robust and user-friendly platform for connecting alumni of our educational institution. This brief document outlines the various technologies employed in the development of the system and their respective roles in achieving the project's objectives.

Back-End Technologies:

1. Java:

- Java serves as the primary programming language for the back-end development of the "Alum Hub" system.
- Its object-oriented nature and platform independence make it a suitable choice for building scalable and maintainable applications.
- Java provides robust libraries and frameworks, facilitating efficient development and seamless integration with other technologies.



2. Java Servlets:

- Java Servlets are used to handle user requests on the server-side.
- They process incoming HTTP requests and generate dynamic responses, ensuring data retrieval and manipulation based on user interactions.
- Servlets manage user authentication, event registration, job posting submissions, and other server-side functionalities.

3. Java Server Pages (JSP):

- Java Server Pages allow the dynamic generation of HTML content, enabling the presentation layer of the web application.
- Combined with Java Servlets, JSP pages create interactive and data-driven web pages,

enhancing the user experience.

4. Java Database Connectivity (JDBC):

- JDBC facilitates interaction with the chosen Database Management System (DBMS) to store and retrieve user data, event details, job postings, and other relevant information.
- It enables seamless integration between the Java application and the back-end database, ensuring efficient data handling.

5. Database Management System (DBMS):

- A relational database management system such as MySQL, PostgreSQL, or Oracle is used to store and manage structured data.
- The DBMS securely stores user profiles, event information, job postings, and other relevant data, allowing efficient retrieval for display and analysis.

6. HTML, CSS, and JavaScript:

- Standard web technologies such as HTML, CSS, and JavaScript are employed to create the front-end user interface.
- HTML defines the structure of the web pages, CSS handles presentation and styling, while JavaScript enables client-side interactivity and dynamic behaviors.

7. Bootstrap:

- Bootstrap, a popular front-end framework, is utilized to enhance the visual appearance and responsiveness of the "Alum Connect" web application.
- It provides a collection of pre-designed CSS styles, layout components, and JavaScript plugins, streamlining front-end development and ensuring consistent user experience across different devices.

Web Server and Deployment:

8. Apache Tomcat:

- Apache Tomcat is chosen as the Java Servlet container to deploy and manage the "Alum Connect" web application.
- It handles incoming HTTP requests and manages the execution of Java Servlets and JSP

pages, ensuring seamless communication between the server and users.

Security:

9. SSL Certificate:

- An SSL certificate is implemented to enable secure communication over HTTPS, encrypting data during transmission between the user's browser and the server.
- It ensures data privacy and protection against potential threats such as data interception and tampering.

10. Version Control (e.g., Git):

- A version control system, such as Git, will be used to track changes to the source code, enable collaboration among developers, and manage code revisions.

11. Security Measures:

- Security best practices and measures will be implemented to safeguard user data, prevent unauthorized access, and protect against potential vulnerabilities.

12. Email Integration:

- The system will integrate with an email service or email server to facilitate user account verification, password resets, and email communication.

The "Alum Hub" system leverages a comprehensive technology stack, incorporating Java, Java Servlets, JSP, JDBC, and Bootstrap, to deliver a seamless, secure, and user-friendly platform for alumni engagement and networking. By combining these cutting-edge technologies, the system aims to foster a strong and active alumni community, reinforcing the bond between alumni and their alma mater.

CHAPTER 2

Proposed System

Chapter 2: Proposed System

2.1 Proposed System:

To address these challenges, the Alumni Hub system will provide alumni with a platform that enables them to connect with each other, share information and experiences, and stay informed about institution-related news and events. The system will also provide the institution with a way to engage with its alumni and create a sense of community among them.

The Alumni Hub system will be implemented using Java as the programming language. The system will utilize Java-based web technologies such as Java Server Pages (JSP) and Java Servlets to build the user interface and handle user requests. The system will also use Java Database Connectivity (JDBC) to interact with the database, which will store information about alumni and institution-related events and activities.

With a wide range of features and functionalities, the system aims to create a vibrant alumni community, providing alumni with a centralized platform to interact, share information, and stay connected with each other and the institution. The following key components and functionalities outline the scope of the proposed "Alum Hub" system:

1. Alumni Profile Creation: The system allows alumni to create personalized profiles, providing relevant information such as contact details, professional experience, current company or organization, and other personal details. This profile creation process ensures that each user has a well-defined and searchable identity within the platform, fostering effective networking.
2. Searchable Alumni Directory: "Alumni Hub" features a searchable directory that allows alumni to find and connect with fellow alumni based on specific criteria. Alumni can conduct targeted searches based on location, industry, skill sets, and profession, facilitating meaningful collaborations and interactions.
3. Contact with Teachers and Current Students: The system facilitates seamless communication between alumni and current and retired teachers, fostering knowledge exchange, mentorship, and appreciation. Alumni can express gratitude and provide insights to teachers who have played a significant role in shaping their education and career paths.
4. Interaction with Current Students: "Alum Hub" enables alumni to engage with current students, offering guidance, sharing experiences, and nurturing a sense of support and mentorship. This feature contributes to the educational journey of current students, helping them gain valuable insights from experienced professionals.

5. Event and Job Postings: The platform allows alumni to post and view upcoming events organized by the institution, alumni chapters, or other alumni groups. Job postings shared within the alumni community promote career growth and networking opportunities among members.
6. Gallery of Photos and Memories: "Alum Hub" provides alumni with the ability to upload and share photos capturing various events, gatherings, and memorable moments during their time at the institution. This feature fosters nostalgia and strengthens the emotional bond between alumni and their alma mater.
7. Organize Sessions and Workshops: Alumni can take the initiative to organize sessions, workshops, or webinars on topics of interest. They can invite other alumni, students, or teachers to participate, creating a platform for knowledge exchange and skill development.
8. Tracking Session Participation: Alumni will have visibility into the list of students who have joined the sessions they organized. This feature encourages alumni to actively engage in knowledge sharing and fosters a sense of community and responsibility.
9. Newsletter Distribution: The system facilitates the creation and distribution of newsletters containing updates on institutional activities, alumni achievements, and upcoming events. All registered users, including alumni, teachers, and current students, receive these newsletters, ensuring everyone stays informed.
10. Admin Panel: An admin panel will be accessible to authorized personnel to manage user accounts, monitor system activities, and moderate content. This administrative control ensures the smooth functioning and security of the platform.

2.2 Feasibility Study:

While developing any computerized system it is necessary to examine whether the developed computerized system is feasible in the point of view operational, technical and financial. Requirement analysis take place can be combined into a feasibility analysis and a business proposal is put forth. With a very general plan for the project and some cost estimates. For feasibility analysis, some understanding of the major requirements of the system is essential.

The cases are categorized as follows:

1. Technical Feasibility
2. Economic Feasibility
3. Operational Feasibility

Technical Feasibility:

Technical feasibility is the study of cost benefit analysis of the system and study of the hardware and software requirements i.e. technical requirements of the system in order to inform management and user for particular system designing this much technical sources are required, in order to know user and management view, regarding the technical resources. By studying the technical study the actual cost of the system designing is known. Organization can provide the minimum required as specified in operational environment.

This system is technically feasible because the JAVA/J2EE is used in the system and hence it is a very secure system. The required operational environment is very cheap and it is open source. The development tools and libraries are available for free so it is technically more feasible.

Economic Feasibility:

Economic feasibility study is the actual cost and benefit of the system. Actual cost of the system is calculated in the economic study so as to inform the user and management that this much cost will be required to develop the system. Because system cost is the first cost before starting system designing. The organization is ready to invest on hardware and software required by the proposed system. So system is economically feasible.

This system is also economically feasible because all the development resources are open source and are available for free. So it requires the fewer amounts for development and implementation.

Operational Feasibility:

Training program will be required to the user. A group of user will be trained during the training period. It focuses on the willingness and the ability of the user. E-Nursery website does not require special efforts to educate the staff on new way on conducting ways. The system incorporates user friendly interface, default processing and required validations. These all are done keeping in view the individual with minimum computer knowledge.

This system is operationally feasible because the user interface provided is very user friendly and normal user can easily understand it and can perform all the operations easily.

2.3 Objectives of Proposed System:

1. Enhanced Alumni Engagement: The proposed system aims to enhance alumni engagement with the educational institution by providing a centralized platform for networking and interaction. Alumni can connect with each other and the institution, fostering a sense of community and belonging.
2. Facilitate Networking: The system seeks to facilitate seamless networking among alumni through a searchable directory. Alumni can find and connect with fellow alumni based on location, industry, skill sets, and profession, promoting meaningful collaborations.
3. Promote Knowledge Exchange: By providing a platform for alumni to interact with current and retired teachers, the system fosters knowledge exchange, mentorship, and appreciation for educators' contributions.
4. Support for Current Students: The system enables alumni to engage with current students, offering guidance, sharing experiences, and providing valuable insights to support their educational journey.
5. Event and Job Opportunities: "Alum Hub" allows alumni to post and view upcoming events organized by the institution, alumni chapters, or other alumni groups. It also facilitates job postings, promoting career growth and networking opportunities.
6. Memories and Nostalgia: Alumni can share photos and memories related to various events and experiences during their time at the institution. This feature fosters nostalgia and strengthens the emotional bond with their alma mater.
7. Knowledge Sharing Sessions: The system empowers alumni to organize sessions, workshops, or webinars on topics of interest, inviting other alumni, students, or teachers to participate. This promotes knowledge sharing and skill development within the community.
8. Community Participation: Alumni can track session attendance, allowing them to view the list of students who joined their sessions. This feature encourages active participation and fosters a sense of community and responsibility.
9. Timely Communication: The system facilitates the creation and distribution of newsletters containing updates on institutional activities, alumni achievements, and upcoming events. This ensures timely communication with all registered users.
10. Admin Management: "Alum Hub" includes an admin panel for authorized personnel to manage user accounts, monitor system activities, and moderate content, ensuring smooth functioning and security.
11. Alumni Recognition: The system showcases alumni achievements and contributions, celebrating their successes and fostering a sense of pride within the community.
12. Enhanced Alumni-Teacher Relationship: By enabling continued interaction and appreciation, the system strengthens the relationship between alumni and teachers, acknowledging the pivotal role of educators.
13. Professional Networking: "Alum Hub" fosters professional networking opportunities, enabling alumni to connect with peers in their respective industries and leverage their network for career growth.
14. Institutional Involvement: The system encourages greater alumni participation and engagement in the institution's events, activities, and initiatives, strengthening the ties between alumni and their alma mater.
15. User-Friendly Interface: "Alum Hub" is designed with an intuitive and user-friendly interface, ensuring ease of use for all users, regardless of their technical proficiency, enhancing their overall experience.

2.4 User of System:

1. Admin:

The admin, short for administrator, is a user of a system who holds the highest level of access and privileges. They are responsible for managing and overseeing the overall functioning of the system. The admin has the authority to configure system settings, create and manage user accounts, assign roles and permissions, and ensure system security. They play a crucial role in maintaining the integrity and smooth operation of the system.

Admins are typically employed by organizations and institutions to handle system administration tasks. They have in-depth knowledge of the system's architecture and functionality, allowing them to troubleshoot issues, perform software updates, and ensure data backup and recovery procedures are in place. They are also responsible for monitoring system performance, identifying potential vulnerabilities, and implementing security measures to safeguard sensitive information. Some key points of admin are as follow:

- User Management: The admin has the authority to manage user accounts, including creating, updating, and deactivating profiles.
- Content Moderation: The admin can moderate content shared by users to ensure it aligns with platform guidelines and maintains a positive user experience.
- Event Management: Admins can create, edit, and manage events, ensuring accurate information and proper organization.
- Analytics and Reports: The admin can access analytics and generate reports related to user engagement, event participation, and other system metrics.
- System Maintenance: The admin oversees system maintenance, ensuring smooth functioning, security, and regular updates.

2. Alumni:

Alumni are individuals who have graduated from an educational institution and have since left the system but maintain a connection with it. They are former students who have completed their studies and moved on to pursue careers or further education. Alumni play an important role in the system by serving as ambassadors, mentors, and potential donors.

As users of the system, alumni often have access to specific resources and benefits provided by the institution. They may have access to an alumni directory, job placement services, or special events. Alumni can stay connected with their institute, fellow graduates, and current students through various communication channels, such as alumni newsletters, social media groups, and dedicated alumni platforms.

Alumni can contribute to the system by sharing their experiences, providing career guidance, and mentoring current students. They can also give back to the institution through donations, sponsorships, or volunteering. Alumni often serve as a valuable resource for the institution, leveraging their professional networks and expertise to support the growth and development of the system and its current users. Summary of key points:

- Profile Creation: Alumni can create personalized profiles, providing contact details, professional experience, and personal information.
- Networking: Alumni can search for and connect with other alumni based on various criteria, fostering networking opportunities.

- Interaction with Teachers: Alumni can interact with current and retired teachers, expressing appreciation and seeking mentorship.
- Involvement in Events: Alumni can participate in events organized by the institution, alumni chapters, or fellow alumni.
- Knowledge Sharing: Alumni can organize knowledge-sharing sessions, workshops, or webinars and invite others to participate.
- Job Sharing: Alumni can share job opportunities within their professional network, promoting career growth among the community.

3. Student:

Students are the primary users of an educational system. They are individuals who are currently enrolled in an educational institution and actively participate in learning activities. Students have access to course sessions, event gallery, and communication tools within the system.

As users, students utilize the system to access, engage in online discussions, collaborate with peers, communicate with teachers, and access learning resources. They can view their job posts. Summary of key points:

- Profile Creation: Current students can create profiles with information about their academic interests and aspirations.
- Interaction with Alumni: Current students can interact with alumni, seeking guidance and mentorship for their academic and career paths.
- Participation in Sessions: Students can join knowledge-sharing sessions or workshops organized by alumni.
- Event Participation: Students can participate in events organized by alumni or the institution to expand their network and knowledge.
- The "Alum Hub" system ensures that each user group has distinct functionalities tailored to their roles, promoting a cohesive and engaging experience for all stakeholders within the alumni community.

4. Teachers:

Teachers are an integral part of any educational system. They are users who are responsible for instructing and guiding students in their learning journey. Teachers utilize the system to create and manage, deliver instructional materials, assess student profile, and provide feedback and support.

Within the system, teachers can create and organize session with alumni's. They can communicate with students, provide instructions, and facilitate discussions through discussion boards, chat features, or video conferencing tools. Summary of key points:

- Profile Creation: Teachers can create profiles with their contact information and details about their role and experience.
- Interaction with Alumni: Teachers can engage with alumni, offering mentorship and staying updated on their achievements.
- Participation in Sessions: Teachers can participate in knowledge-sharing sessions or workshops organized by alumni.
- Event Involvement: Teachers can be invited to participate in events and activities organized by the alumni community.

CHAPTER 3

Analysis & Design

Chapter 3: Analysis & Design

3.1 System Requirements (Functional and Non-Functional requirements)

Functional requirements: In software engineering, a functional requirement gives a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other functionality that define what a system is supposed to accomplish.

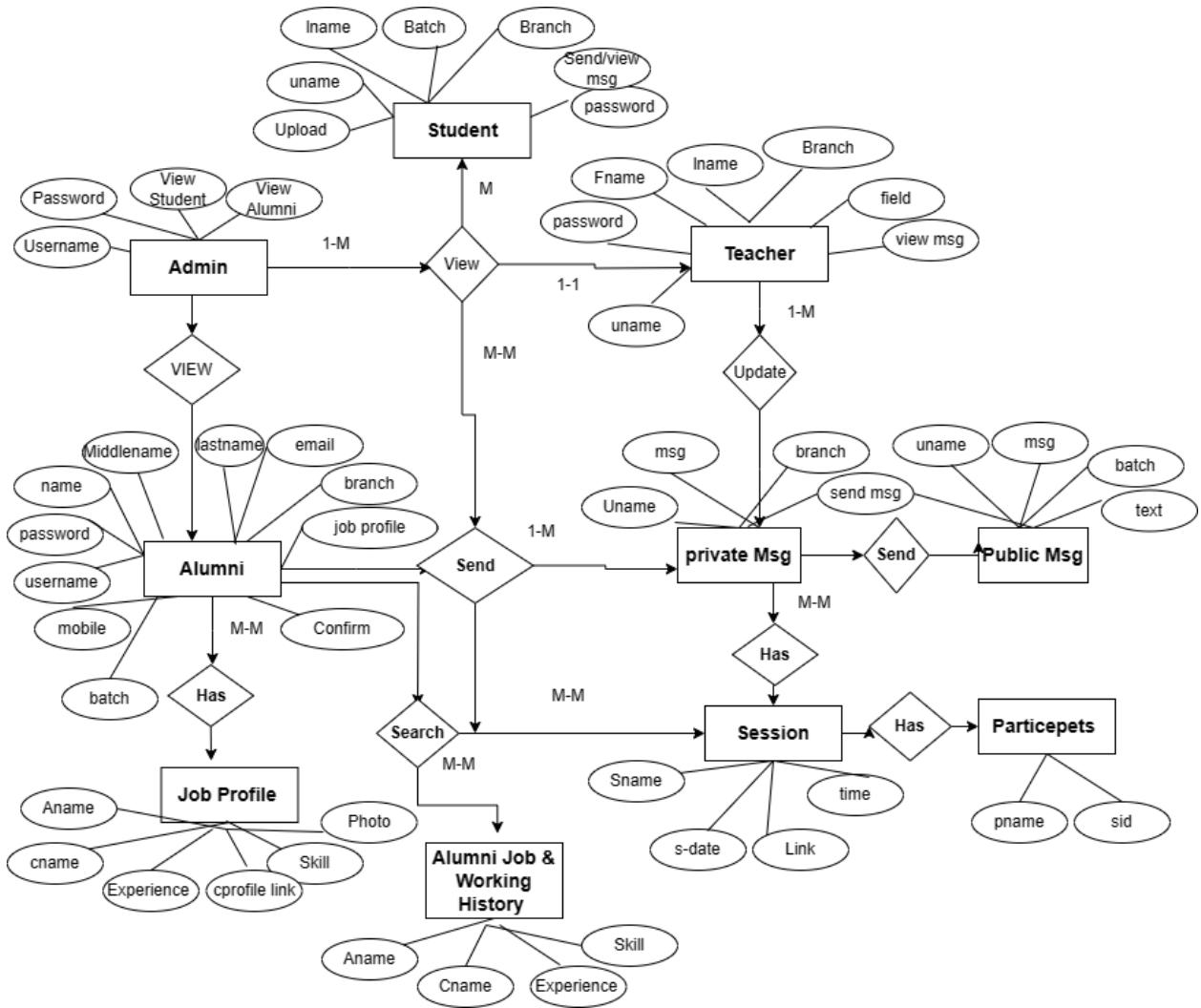
1. User Registration: Users should be able to register with the system by providing necessary details such as name, email, and password.
2. Profile Creation and Management: Users should be able to create and manage personalized profiles with information such as contact details, professional experience, and other relevant data.
3. Search and Networking: The system must provide a search functionality that allows users to find and connect with fellow alumni based on various criteria like location, industry, skill sets, and profession.
4. Interaction with Teachers and Students: Alumni should be able to initiate communication with current and retired teachers, and interact with current students through messages or comments.
5. Event Management: Alumni should be able to create and manage events, and other users should be able to view and register for these events.
6. Knowledge Sharing Sessions: Alumni should be able to organize sessions, workshops, or webinars on specific topics and invite other alumni, students, or teachers to participate.
7. Job Postings: Alumni should be able to post job opportunities within their professional network, and other users should be able to view and apply for these job postings.
8. Gallery of Photos and Memories: Alumni should be able to upload and share photos related to various events and memorable moments during their time at the institution.
9. Event Registration and Attendance: Users should be able to register for events and view the list of attendees to track session participation and encourage engagement.
10. Newsletter Distribution: The system should enable the creation and distribution of newsletters containing updates on institutional activities, alumni achievements, and upcoming events.
11. Admin Privileges: Authorized personnel should have administrative privileges to manage user accounts, moderate content, and perform system maintenance.
12. User Authentication and Authorization: The system should have robust authentication mechanisms to ensure secure access and authorization levels for different user roles.
13. Error Handling and Validation: The system should provide appropriate error messages and validation checks to ensure data accuracy and prevent incorrect inputs.
14. Data Storage and Retrieval: The system should efficiently store and retrieve user information, event details, job postings, and other relevant data.
15. Responsive Design: The system's user interface should be responsive and adapt to different screen sizes and devices for an optimal user experience.

These functional requirements define the core functionalities and behavior that the "Alum Hub" system is expected to accomplish, providing an engaging and user-friendly platform for alumni engagement and networking.

Non-functional requirement

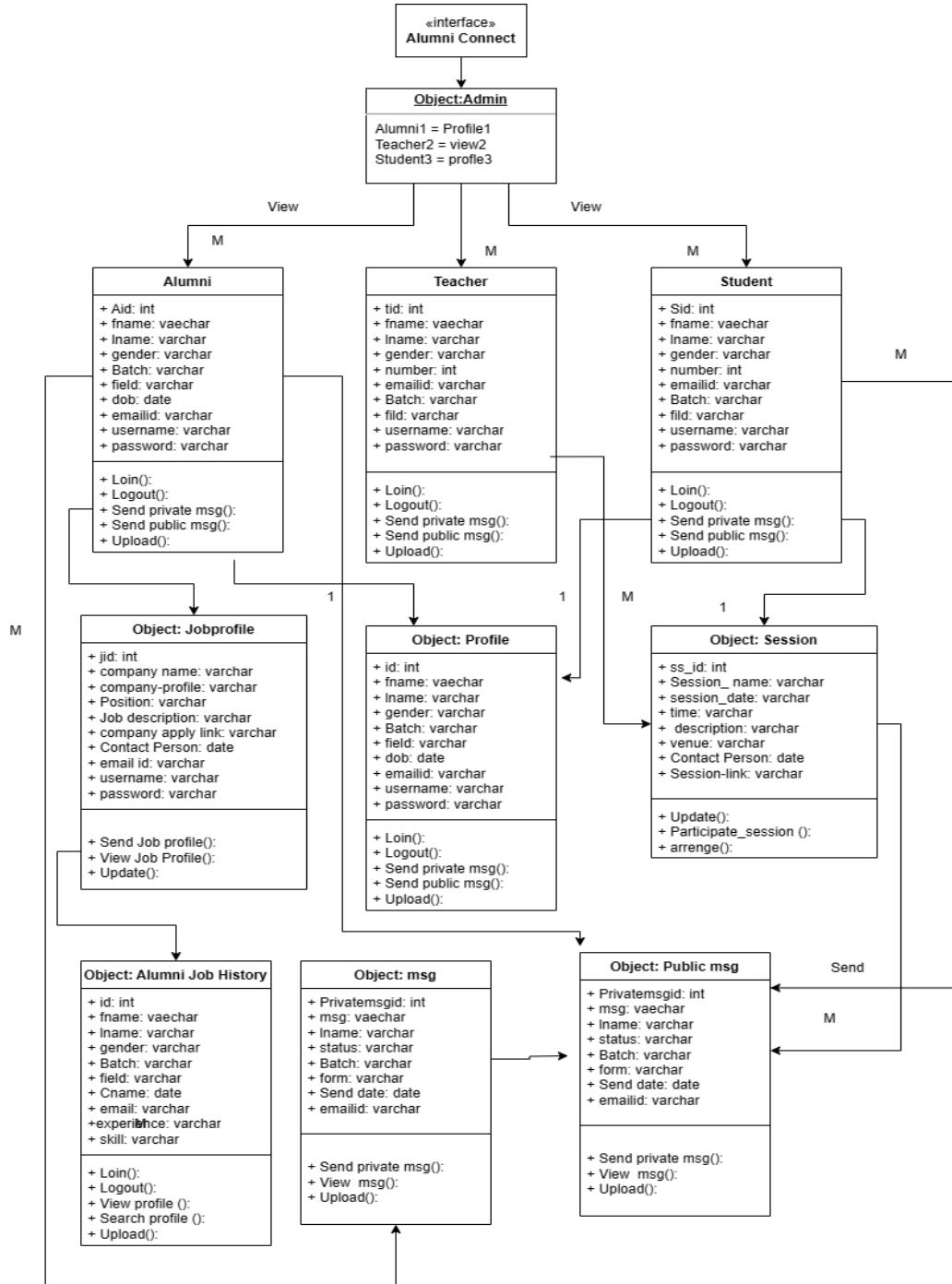
- Availability : The system will be available 24*7*365
- Capacity: Any time you can use the system within a day and the system will able to generate same results each time.
- Throughput: The system performs the speedy operations and has very less throughput time.
- Security: Only users of the system can have the specified access to the system.

3.2 Entity Relationship Diagram(ERD)



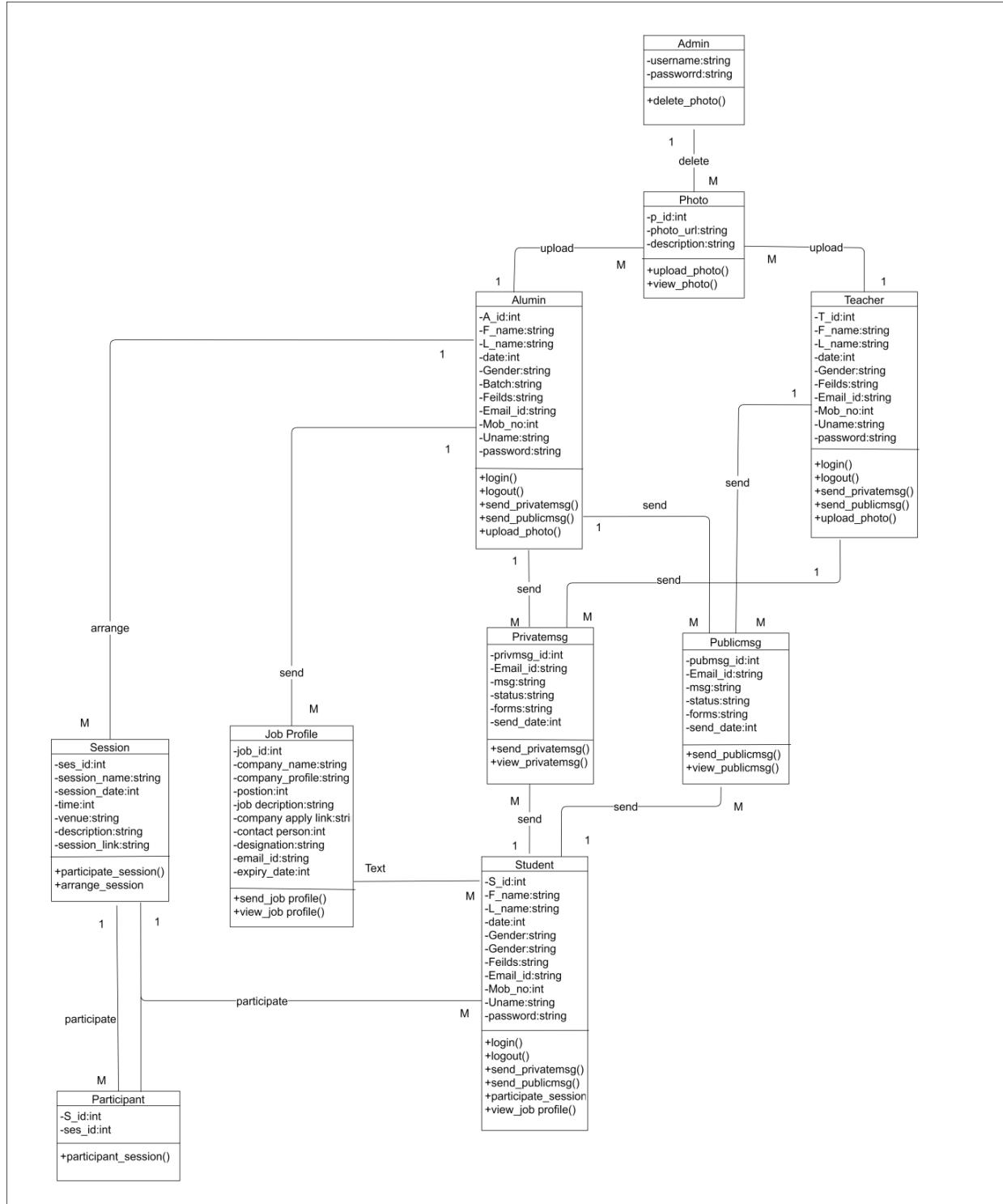
3.3 Object Diagram

An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time. The use of object diagrams is fairly limited, namely to show examples of data structure.

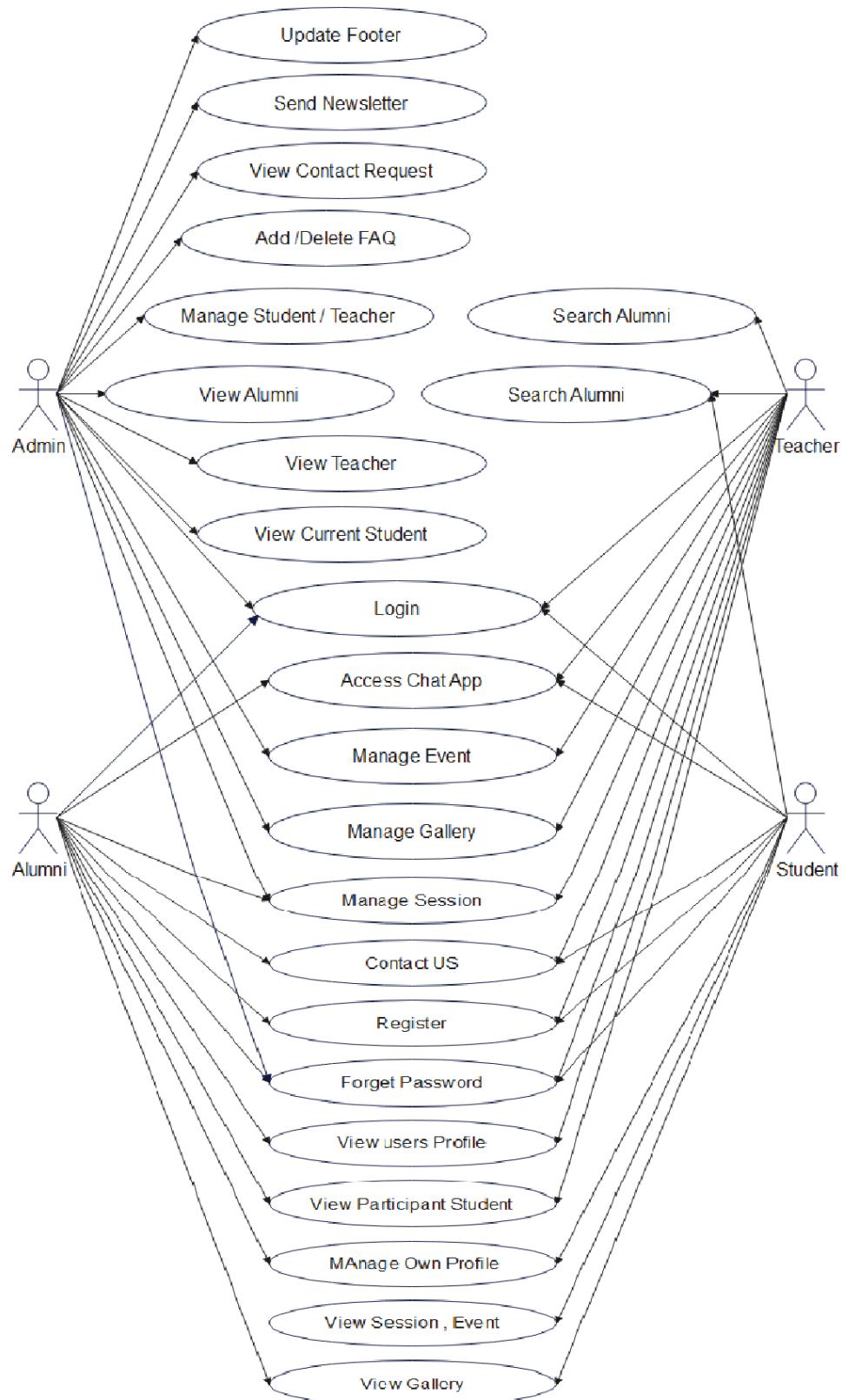


3.4 Class Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.



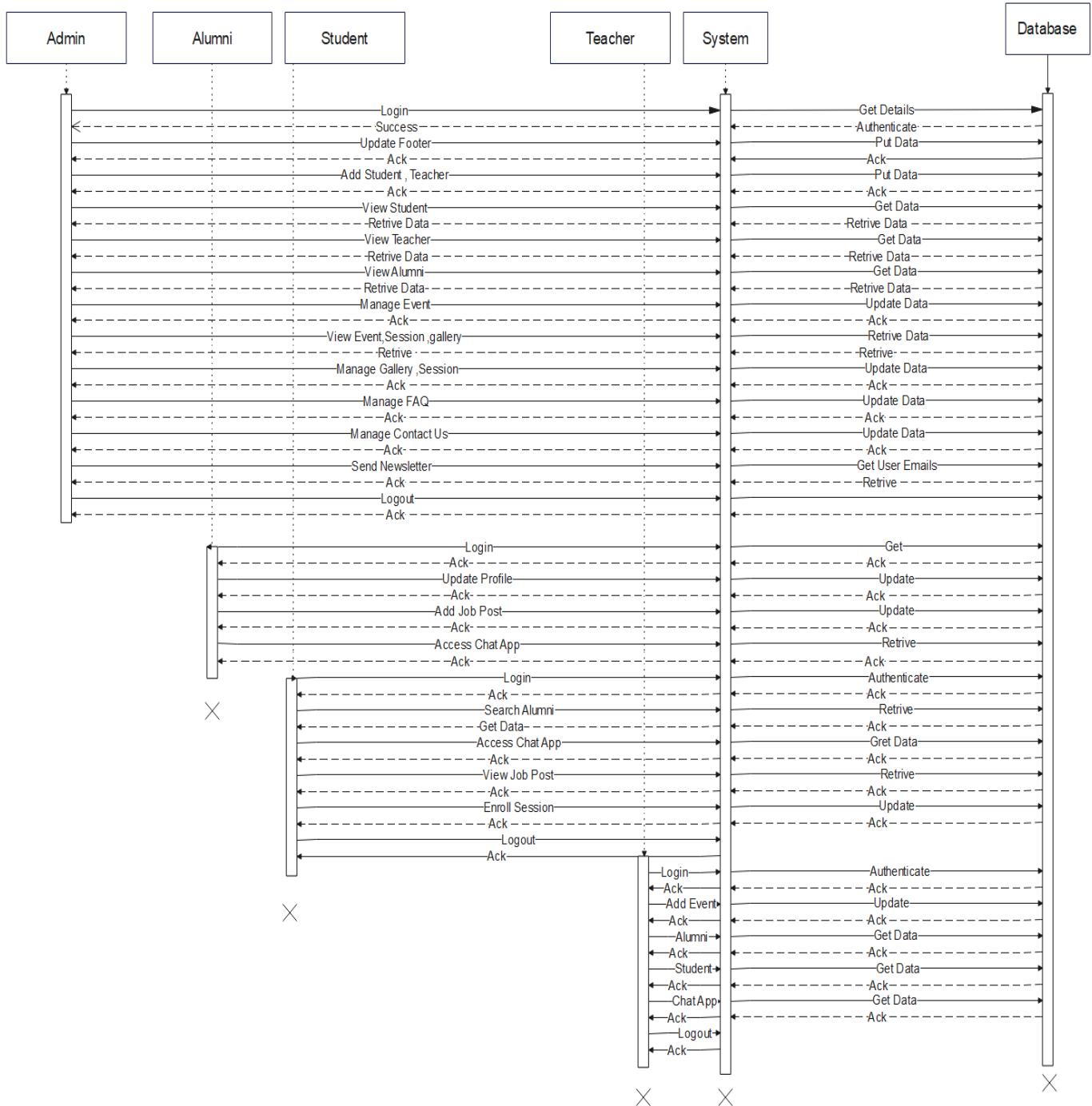
3.5 Use Case Diagram:



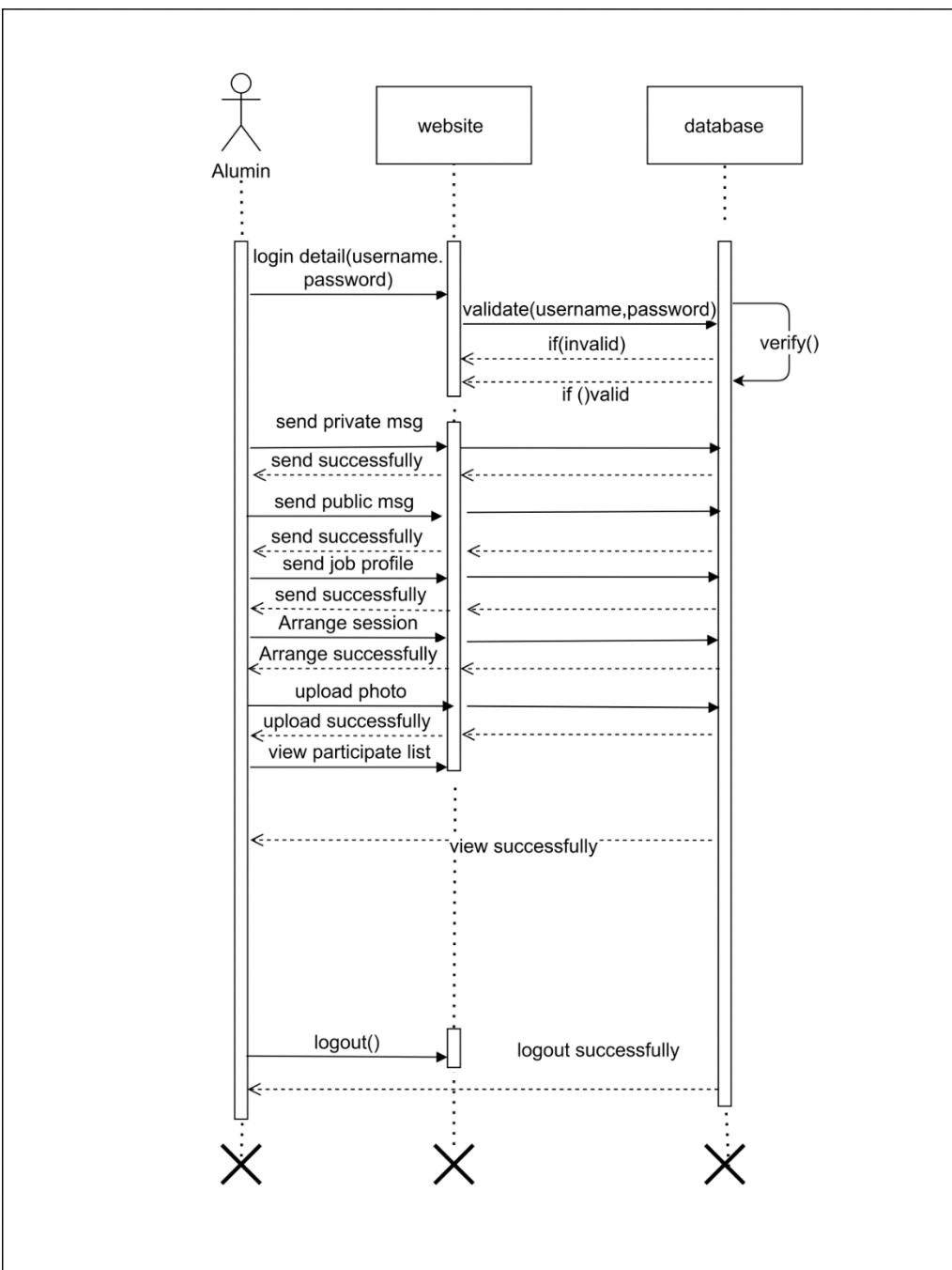
3.6 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

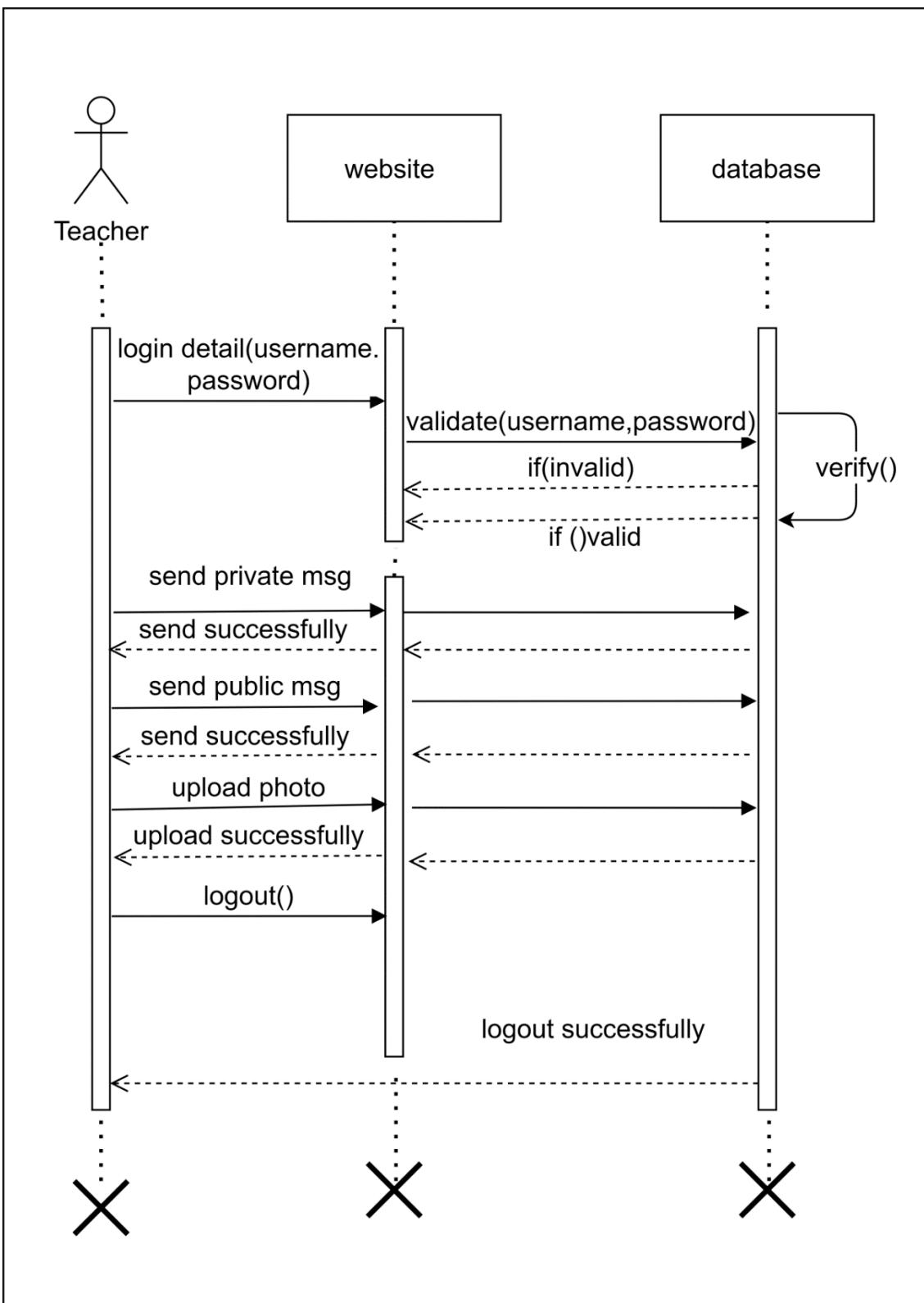
OverAll



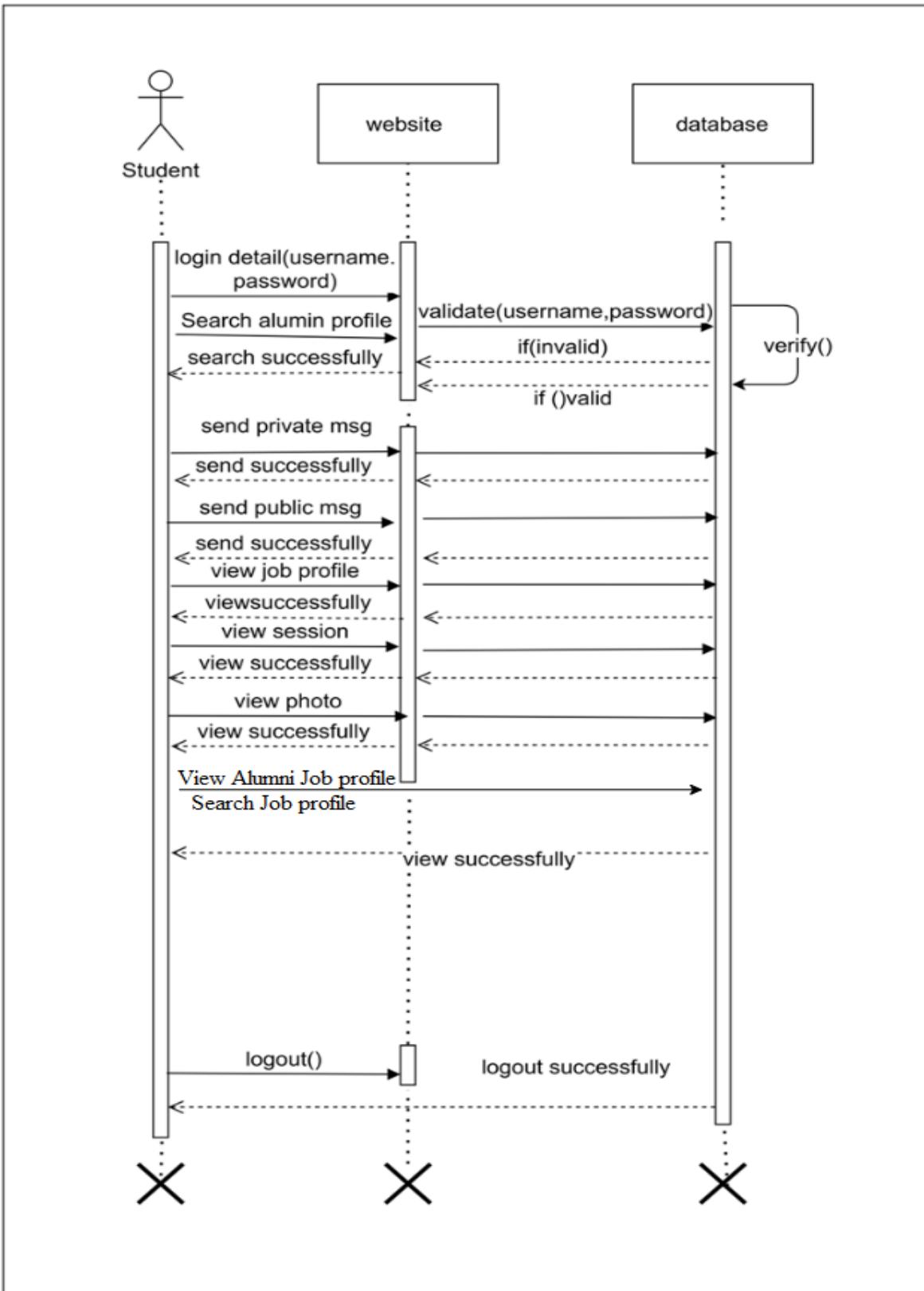
Alumni sequence diagram:



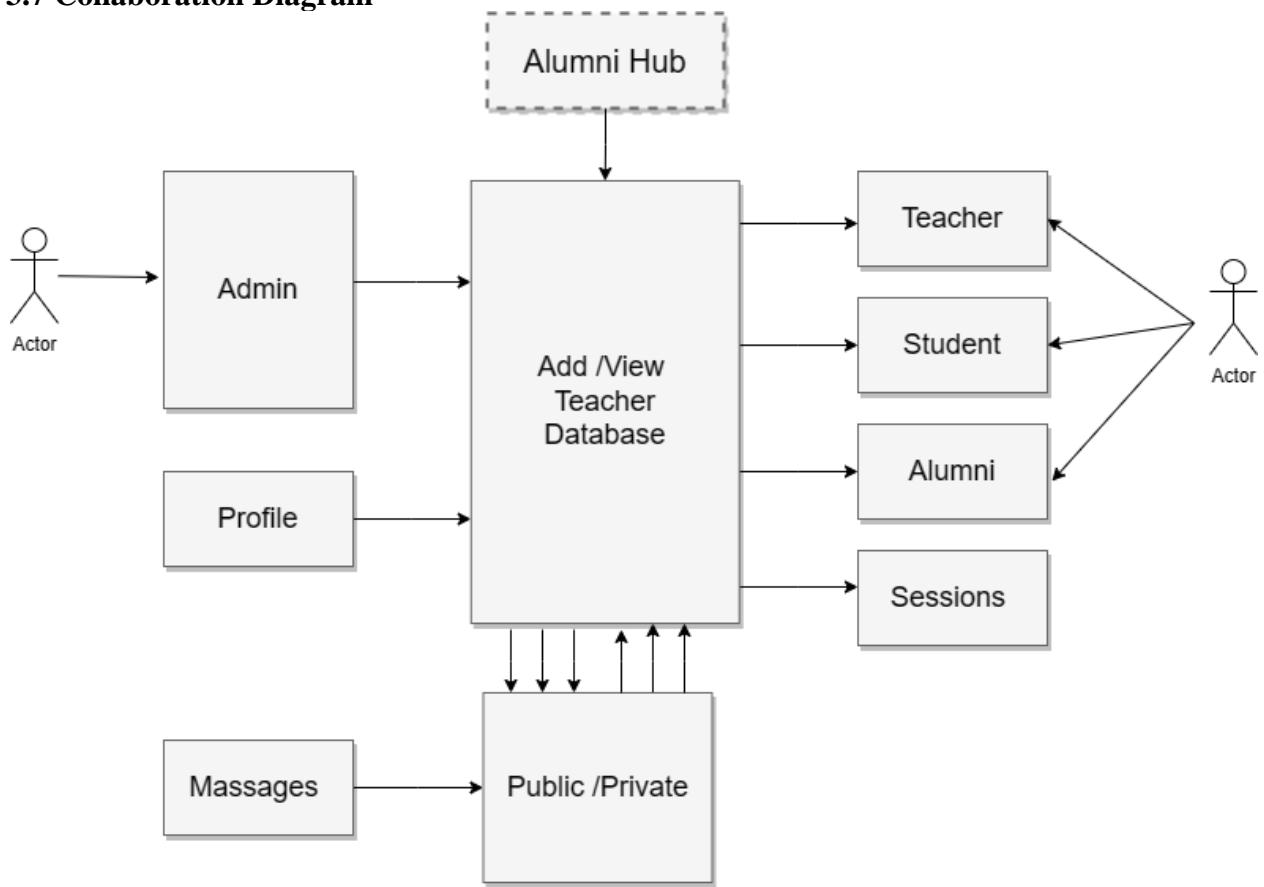
Teacher sequence diagram:



Student sequence diagram:

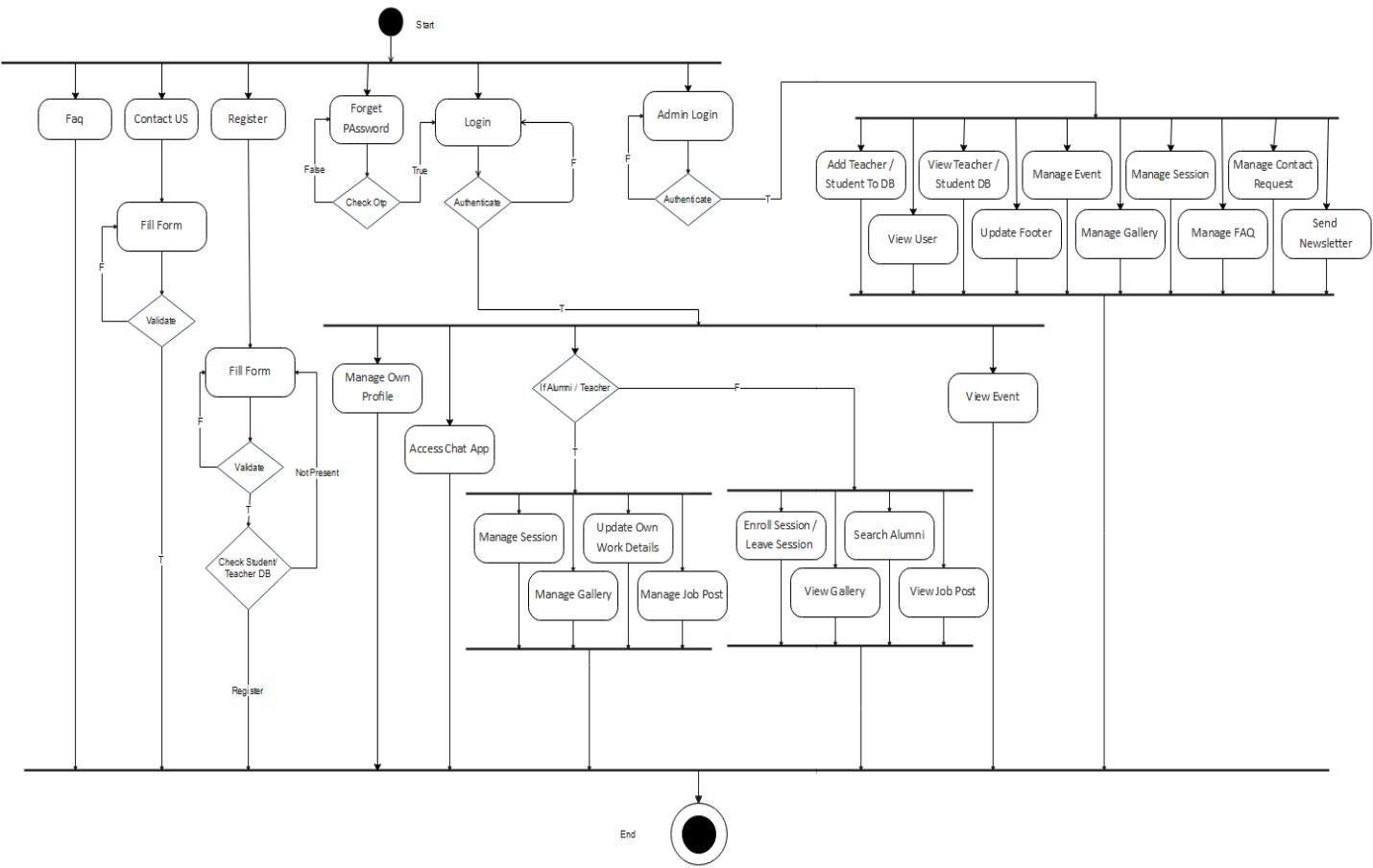


3.7 Collaboration Diagram

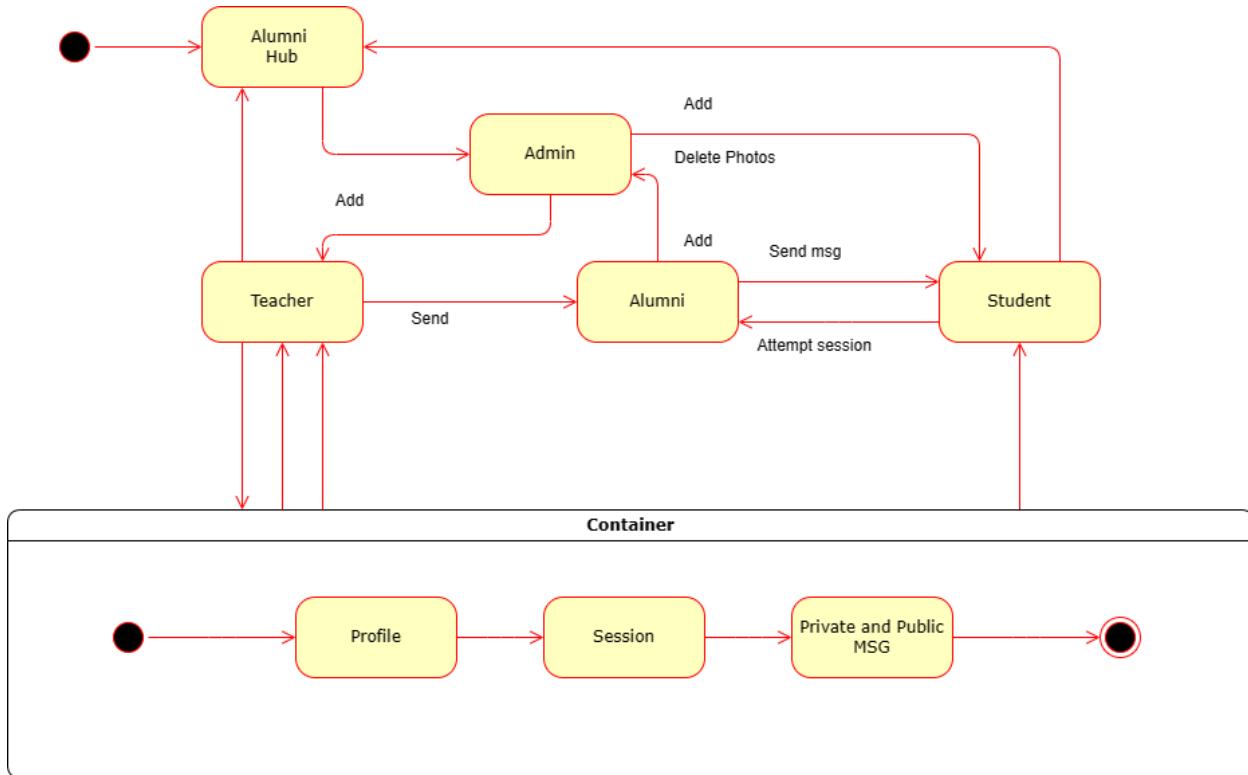


3.8Activity Diagram

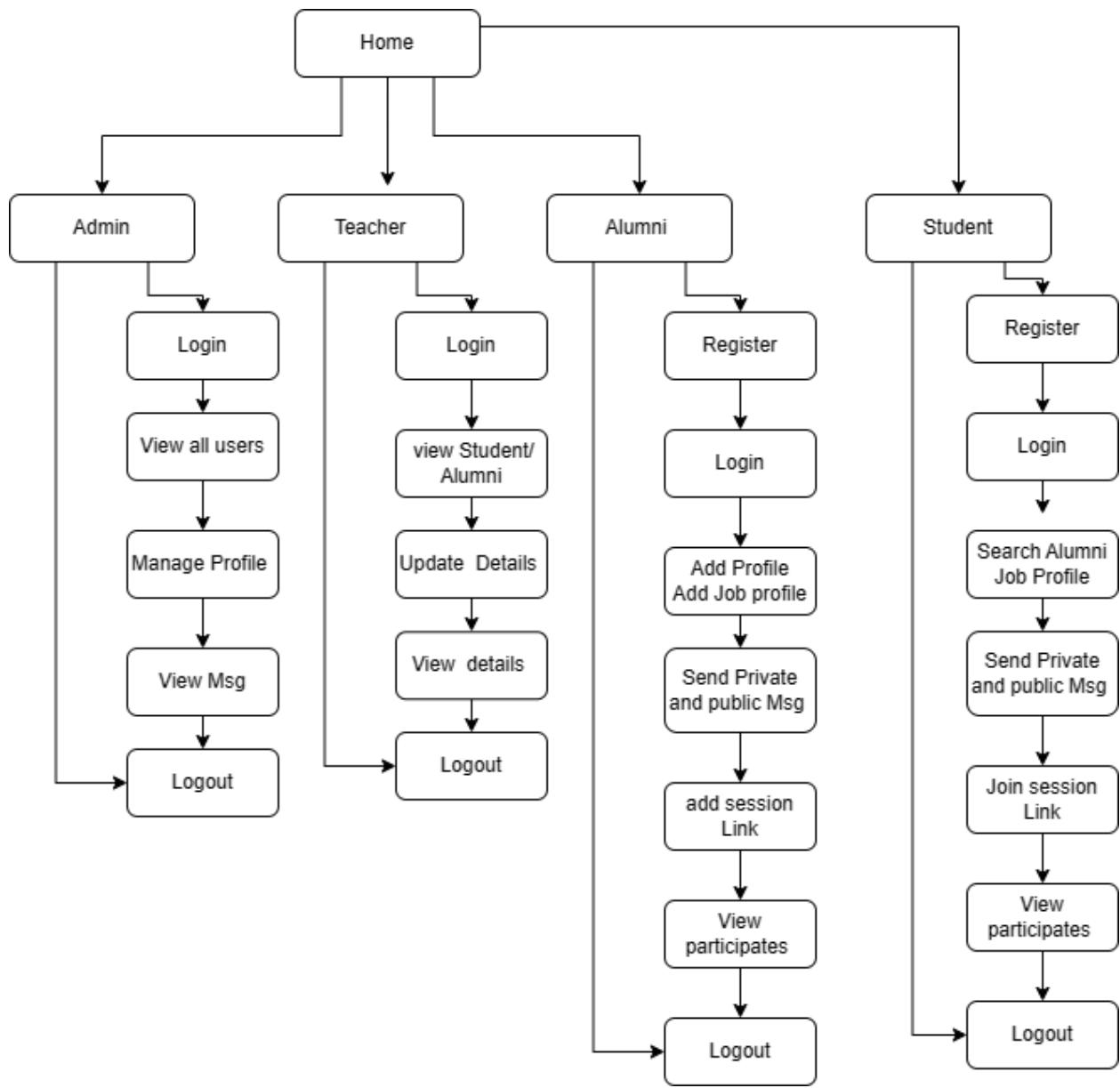
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.



3.9 State Diagram:

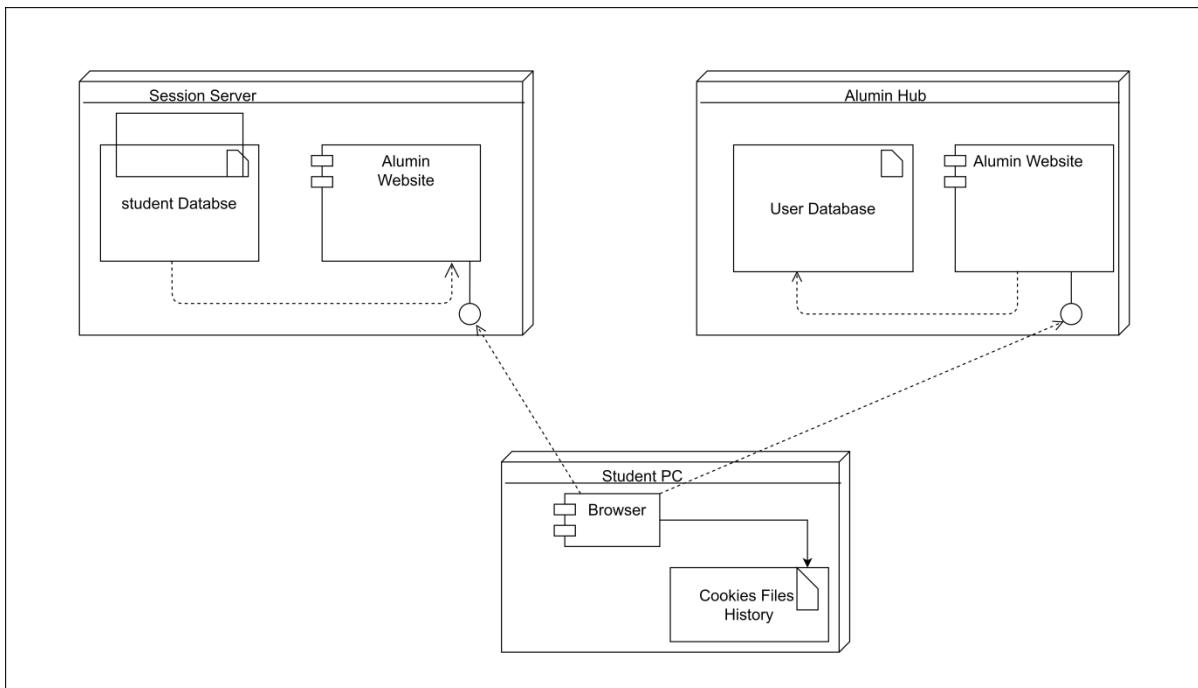


3.10 User Interface Screens

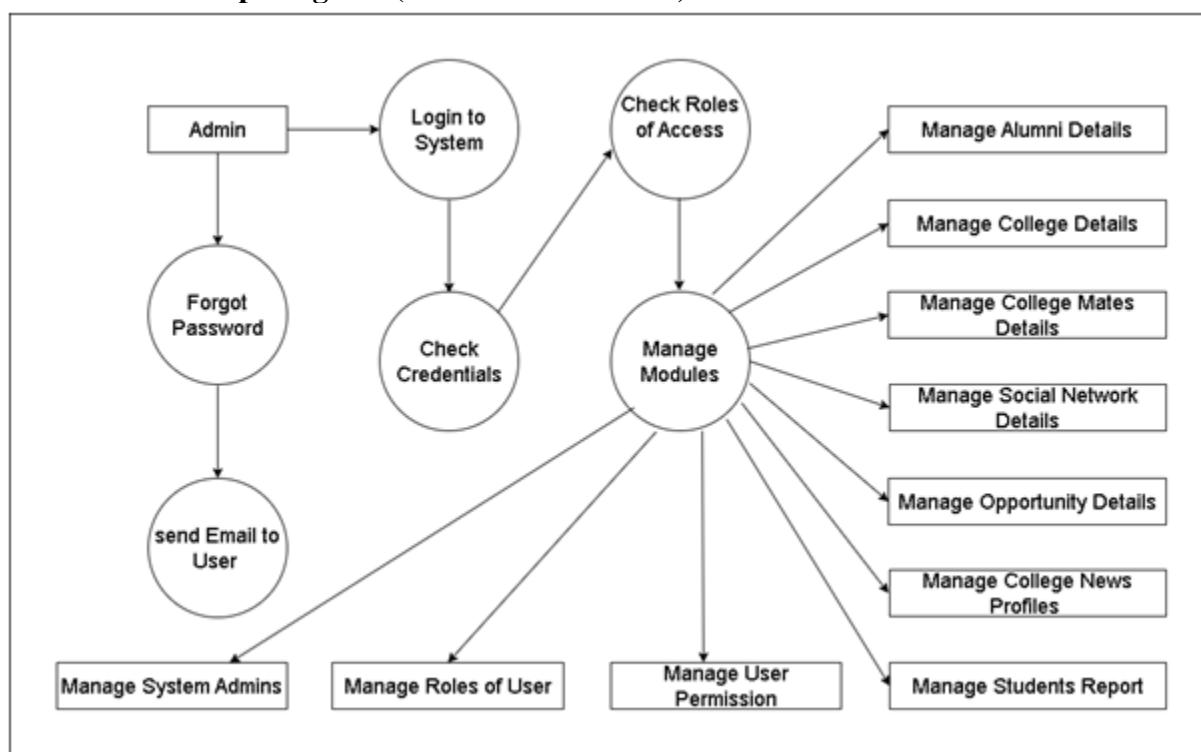


3.11 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.



3.13 Web Site Map Diagram (in case of Web Site)



3.14 Module Specification/DFD:

Alumni :

- Update Profile: Access and edit your personal and professional information.
- Alumni Directory: Explore the directory to find and connect with other alumni based on location, industry, or profession.
- Message Alumni: Send direct messages to fellow alumni through the system.
- View Events: Access and view a list of upcoming events organized by the institution.
- View Job Posts: Explore job postings shared by other alumni.
- View Sessions: Enroll for sessions or webinars and leave joined sessions.
- Chat Box: Interact with teachers, current students, and public chat groups.
- Gallery: Access and view photos in the system's gallery.

Teacher :

- Update Profile: Access and edit your personal and professional details.
- Message Alumni: Send direct messages to alumni through the system.
- Message Students: Send direct messages to students through the system.
- View Events: Access and view a list of upcoming events organized by the institution.
- View Job Posts: Explore job postings shared by alumni.
- View Sessions: View scheduled sessions and arrange sessions for students.
- Edit Session: Modify session details such as date, time, and description.
- Delete Session: Remove scheduled sessions from the system.
- View Participants: Access the list of participants in a session.
- Chat Box: Interact with alumni, current students, and public chat groups.
- Gallery: Access and view photos in the system's gallery.

Current Student :

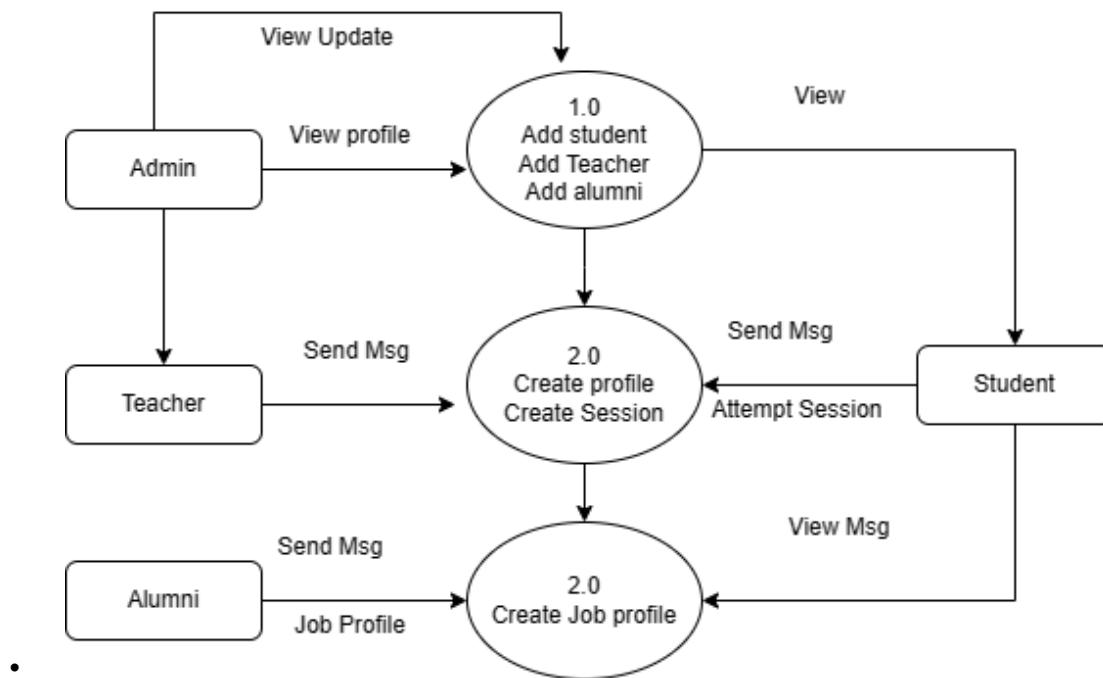
- Update Profile: Access and edit your personal and academic details.
- Message Alumni: Send direct messages to alumni through the system.
- View Events: Access and view a list of upcoming events organized by the institution.
- View Job Posts: Explore job postings shared by alumni.
- View Sessions: Enroll for sessions or webinars and leave joined sessions.
- Chat Box: Interact with alumni, teachers, and public chat groups.
- Gallery: Access and view photos in the system's gallery.

Admin :

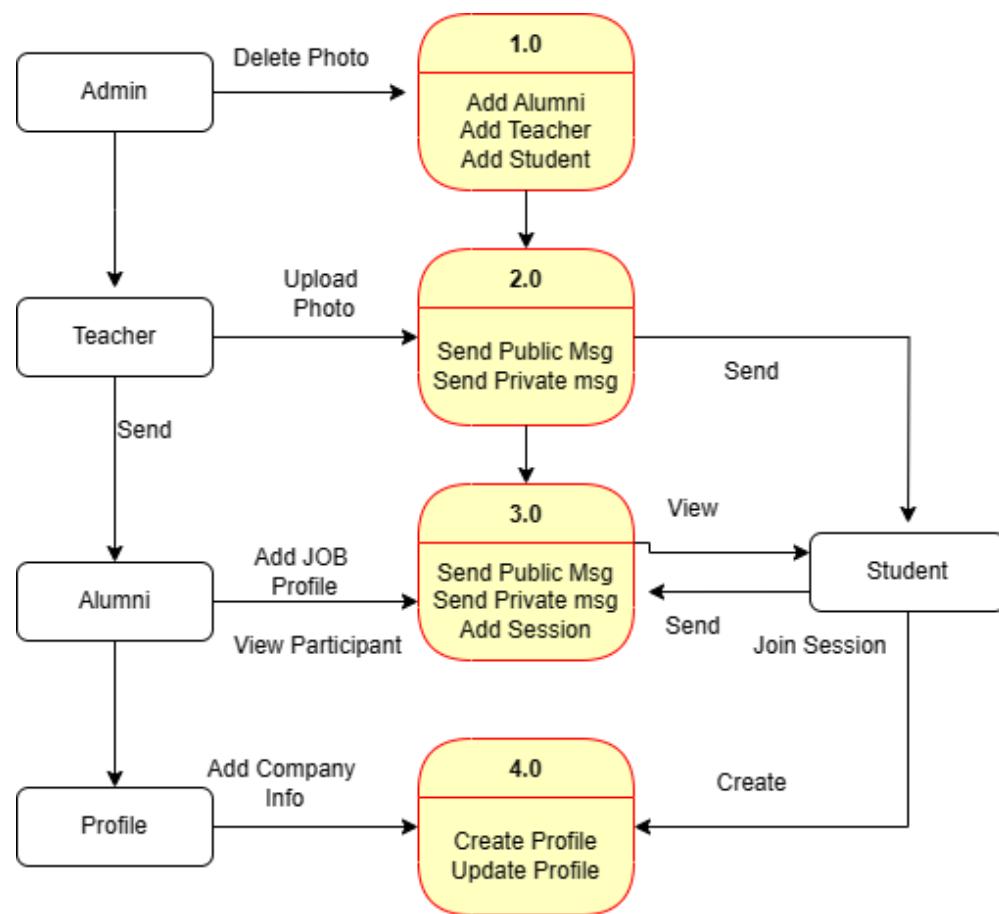
- Add Student: Add new students to the student database.
- Add Teacher: Add new teachers to the teacher database.
- View Teacher Profile: Access and view teacher profiles and their details.
- View Student Profile: Access and view student profiles and their details.
- View Alumni: View a list of all alumni registered in the system.
- Update Footer Content: Modify and update footer content on the system's pages.
- Add Event: Create and add new events to the platform.

- View Event: Access and view a list of all events organized by the institution.
- Add Photos: Upload and add photos to the system's gallery.
- View Photos: Access and view all the photos available in the system's gallery.
- Edit Event: Edit event details such as date, time, and description.
- Add Session: Create and schedule sessions or webinars for users.
- View Session: Access and view all scheduled sessions and webinars.
- Delete Session: Remove scheduled sessions from the system.
- Delete Event: Delete events from the system.
- Delete Photos: Remove photos from the gallery.
- Delete Student: Delete student profiles from the system.
- Delete Teacher: Delete teacher profiles from the system.
- Add/Delete FAQs: Add or delete frequently asked questions from the system.
- View Contact Us Requests: Access and view messages received via the "Contact Us" form.
- Send Newsletter: Create and send newsletters to all registered users, including alumni, teachers, and students.
- Generate Reports: Generate reports in various formats like CSV, Excel, PDF, or direct print.

DFD ZERO :



DFD ONE :



3.15 Table Structure

Table Structure For adminrecord :

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
username	varchar(30)	NULL	YES	NO	The username of the admin user.
password	varchar(30)	NULL	NO	NO	The password of the admin user.

Table structure for table communication_details:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the communication record.
from_id	varchar(30)	NULL	NO	NO	The ID of the sender of the communication.
to_id	varchar(30)	NULL	NO	NO	The ID of the recipient of the communication.
msg_content	text	NULL	NO	NO	The content of the communication.
msg_date_time	varchar(30)	NULL	NO	NO	The date and time of the communication.
msg_type	varchar(30)	NULL	NO	NO	The type of the communication.

Table structure for table contactus:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the contact record.
name	varchar(200)	NULL	NO	NO	The name of the person who contacted us.
email	varchar(200)	NULL	NO	NO	The email address of the person who contacted us.
subject	varchar(200)	NULL	NO	NO	The subject of the contact message.
message	varchar(300)	NULL	NO	NO	The message of the contact message.

Table structure for table event_details:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the event record.
title	varchar(100)	NULL	NO	NO	The title of the event.
event_details	text	NULL	NO	NO	The details of the event.

pic_name	varchar(30)	NULL	NO	NO	The name of the image associated with the event.
upload_by_id	varchar(30)	NULL	NO	NO	The ID of the user who uploaded the event.
upload_by	varchar(45)	DEFAULT 'NA'	NO	NO	The name of the user who uploaded the event.
archive	tinyint(4)	DEFAULT '0'	NO	NO	Indicates whether the event is archived.

Table structure for table faq:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the FAQ record.
question	varchar(100)	NULL	NO	NO	The question in the FAQ.
answer	varchar(400)	NULL	NO	NO	The answer to the FAQ question.
date	varchar(45)	DEFAULT NULL	NO	NO	The date the FAQ was created.

Table structure for table footer:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the footer record.
contact	varchar(45)	DEFAULT NULL	NO	NO	The contact information for the website.
email	varchar(45)	DEFAULT NULL	NO	NO	The email address for the website.
address	varchar(45)	DEFAULT NULL	NO	NO	The address of the website.
map_link	varchar(45)	DEFAULT NULL	NO	NO	The link to the website's map.
twitter	varchar(45)	DEFAULT NULL	NO	NO	The Twitter handle for the website.
facebook	varchar(45)	DEFAULT NULL	NO	NO	The Facebook page for the website.
instagram	varchar(45)	DEFAULT NULL	NO	NO	The Instagram handle for the website.
github	varchar(45)	DEFAULT NULL	NO	NO	The GitHub handle for the website.
website	varchar(45)	DEFAULT NULL	NO	NO	The website's URL.

Table structure for table gallery:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes

id	int(11)	NOT NULL	YES	NO	The ID of the gallery record.
title	varchar(45)	DEFAULT NULL	NO	NO	The title of the gallery.
details	varchar(100)	DEFAULT NULL	NO	NO	The details of the gallery.
pic_name	varchar(45)	DEFAULT NULL	NO	NO	The name of the image in the gallery.
upload_by_id	varchar(45)	DEFAULT '0'	NO	NO	The ID of the user who uploaded the image.
upload_by	varchar(45)	DEFAULT 'NA'	NO	NO	The name of the user who uploaded the image.
archive	varchar(45)	DEFAULT '0'	NO	NO	Indicates whether the gallery is archived.
date	varchar(45)	DEFAULT NULL	NO	NO	The date the gallery was created.

Table structure for table job_details (job post's details):

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the job details record.
designation	varchar(100)	DEFAULT NULL	NO	NO	The designation of the job.
requirements	varchar(1000)	DEFAULT NULL	NO	NO	The requirements for the job.
company_name	varchar(100)	DEFAULT NULL	NO	NO	The name of the company.
salary	varchar(30)	DEFAULT NULL	NO	NO	The salary for the job.
email	varchar(30)	DEFAULT NULL	NO	NO	The email address for the job.
add_by_id	varchar(30)	DEFAULT NULL	NO	NO	The ID of the user who added the job.
date	varchar(45)	DEFAULT NULL	NO	NO	The date the job was added.
jd	varchar(150)	DEFAULT NULL	NO	NO	The job description.

Table structure for table profile_details (work detail's):

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the profile details record.
u_id	varchar(30)	DEFAULT	NO	NO	The ID of the user.

		NULL			
org_name	varchar(100)	DEFAULT NULL	NO	NO	The name of the organization.
org_city	varchar(100)	DEFAULT NULL	NO	NO	The city of the organization.
designation	varchar(100)	DEFAULT NULL	NO	NO	The designation of the user in the organization.
org_j_date	varchar(30)	DEFAULT NULL	NO	NO	The start date of the user's employment in the organization.
org_e_date	varchar(30)	DEFAULT NULL	NO	NO	The end date of the user's employment in the organization.
current_org	varchar(30)	DEFAULT NULL	NO	NO	Whether the user is currently employed in the organization.
job_detail	varchar(100)	DEFAULT 'NA'	NO	NO	A brief description of the user's job.

Table structure for table sessions_details :

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the session details record.
session_types	varchar(100)	DEFAULT NULL	NO	NO	The type of session.
session_details	text	NULL	NO	NO	The details of the session.
session_d_t	varchar(30)	DEFAULT NULL	NO	NO	The date and time of the session.
session_arrange_by	varchar(30)	DEFAULT NULL	NO	NO	The name of the person who arranged the session.
session_arrange_id	int(11)	DEFAULT '0'	NO	NO	The ID of the person who arranged the session.
session_link	varchar(255)	DEFAULT 'NA'	NO	NO	The link to the session.

Table structure for table session_join_list:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the session join list record.
session_id	int(11)	DEFAULT NULL	NO	YES	The ID of the session.
student_id	int(11)	DEFAULT NULL	NO	YES	The ID of the student who joined the session.

Table structure for table student_db:

Column	Data Type	Nullability	Primary Key	Notes
s_id	int(11)	NOT NULL	YES	The ID of the student record.
s_fname	varchar(70)	DEFAULT NULL	NO	The first name of the student.
s_lname	varchar(70)	DEFAULT NULL	NO	The last name of the student.
s_email	varchar(45)	DEFAULT NULL	NO	The email address of the student.
s_batch	varchar(45)	DEFAULT NULL	NO	The batch of the student.
s_div	varchar(45)	DEFAULT 'NA'	NO	The division of the student.
s_mobile	varchar(45)	DEFAULT NULL	NO	The mobile number of the student.

Table structure for table teacher_db:

Column	Data Type	Nullability	Primary Key	Notes
t_id	int(11)	NOT NULL	YES	The ID of the teacher record.
t_fname	varchar(45)	DEFAULT NULL	NO	The first name of the teacher.
t_lname	varchar(45)	DEFAULT NULL	NO	The last name of the teacher.
t_email	varchar(45)	DEFAULT NULL	NO	The email address of the teacher.
t_mobile	varchar(45)	DEFAULT NULL	NO	The mobile number of the teacher.

Table structure for table subscribe:

Column	Data Type	Nullability	Primary Key	Foreign Key	Notes
id	int(11)	NOT NULL	YES	NO	The ID of the subscription record.
email	varchar(200)	DEFAULT NULL	YES	NO	The email address of the subscriber.

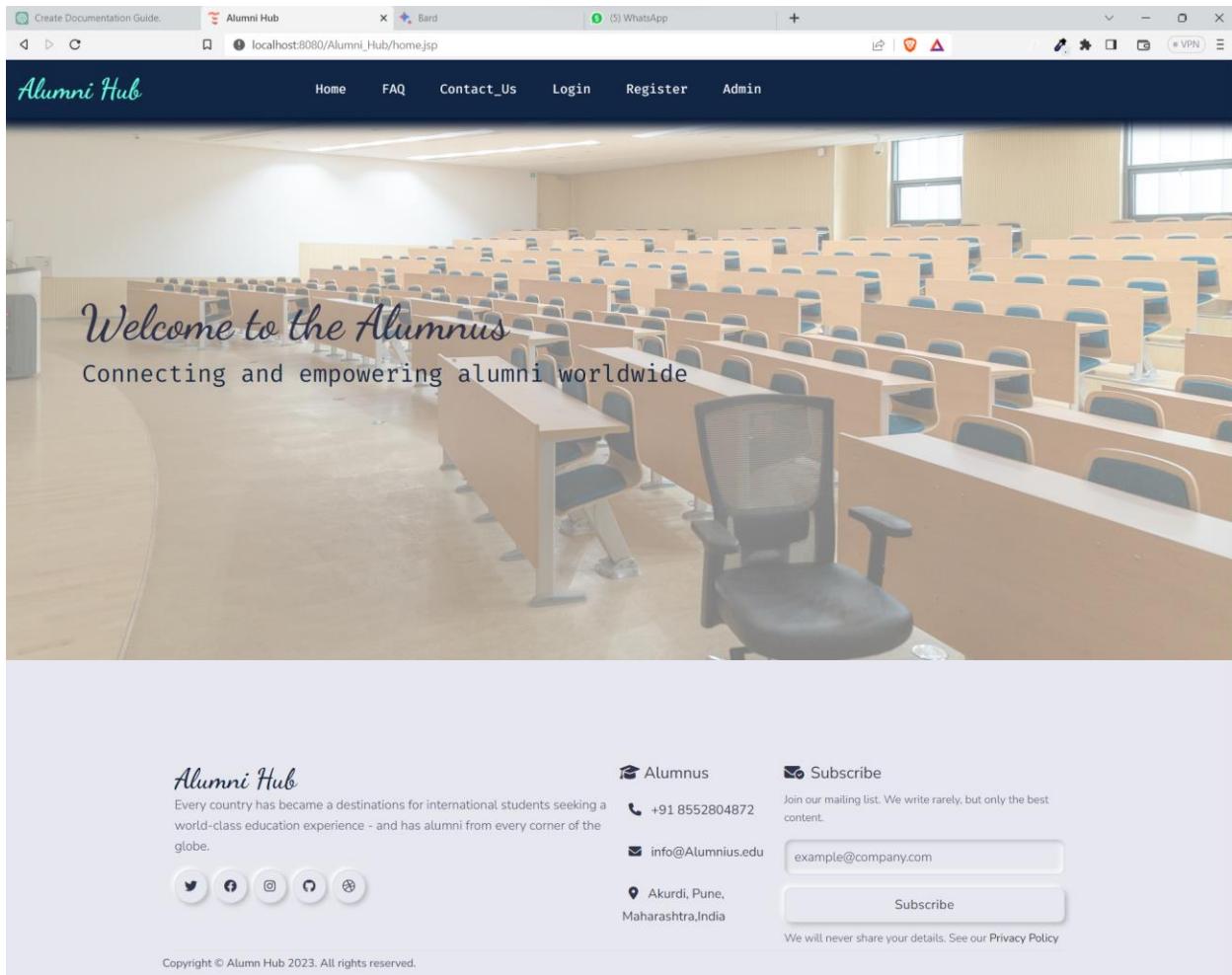
Table structure for table user_details:

Column	Data Type	Nullability	Primary Key	Foreign Key
id	int(11)	NOT NULL	YES	NO
user_type	varchar(30)	DEFAULT NULL	NO	The type of user (student, teacher, admin, etc.).
fname	varchar(30)	DEFAULT NULL	NO	The first name of the user.
lname	varchar(30)	DEFAULT NULL	NO	The last name of the user.
mobile	varchar(30)	DEFAULT NULL	NO	The mobile number of the user.
email	varchar(30)	DEFAULT NULL	NO	The email address of the user.
dob	varchar(30)	DEFAULT NULL	NO	The date of birth of the user.
gender	varchar(30)	DEFAULT NULL	NO	The gender of the user.

pwd	varchar(50)	DEFAULT NULL	NO	The password of the user.
instagram	varchar(100)	DEFAULT 'NA'	NO	The Instagram handle of the user.
facebook	varchar(100)	DEFAULT 'NA'	NO	The Facebook handle of the user.
twitter	varchar(100)	DEFAULT 'NA'	NO	The Twitter handle of the user.
git	varchar(100)	DEFAULT 'NA'	NO	The GitHub handle of the user.
address	varchar(100)	DEFAULT 'NA'	NO	The address of the user.
profession	varchar(50)	DEFAULT 'NA'	NO	The profession of the user.
skill	varchar(100)	DEFAULT 'NA'	NO	The skills of the user.
website	varchar(100)	DEFAULT 'NA'	NO	The website of the user.
photo	varchar(100)	DEFAULT 'NA'	NO	The photo of the user.
bio	varchar(200)	DEFAULT 'Add bio'	NO	The bio of the user.

3.16 Sample Input Output Screen:

Home Page:



Create Documentation Guide. Alumni Hub WhatsApp

localhost:8080/Alumni_Hub/sign-up.jsp

Create Account

Your First | Last Name

Fname Lname

Choose Birth Date

Select date

Mobile Number

Mobile number

Gender

Male Female

Your email

example@company.com

User Type

Choose...

Password

Password

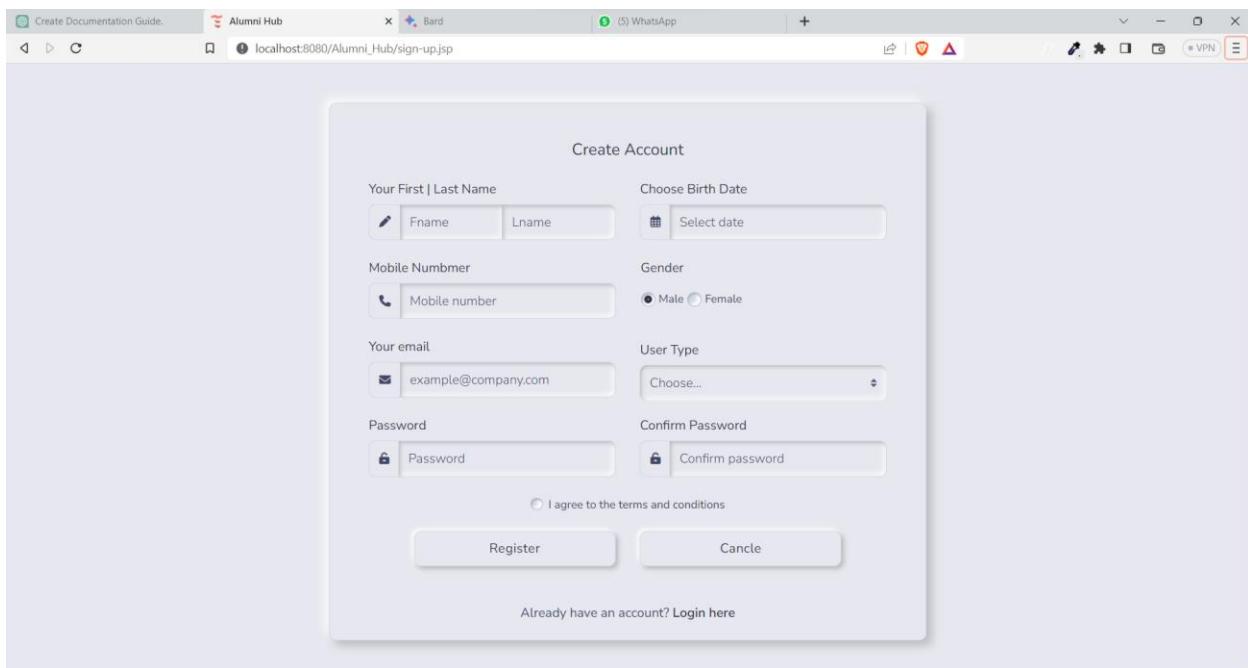
Confirm Password

Confirm password

I agree to the terms and conditions

Register Cancel

Already have an account? [Login here](#)



Create Documentation Guide. Alumni Hub WhatsApp

localhost:8080/Alumni_Hub/sign-in.jsp

Sign in to Alumnus

Choose Login

Choose...

Your email

example@company.com

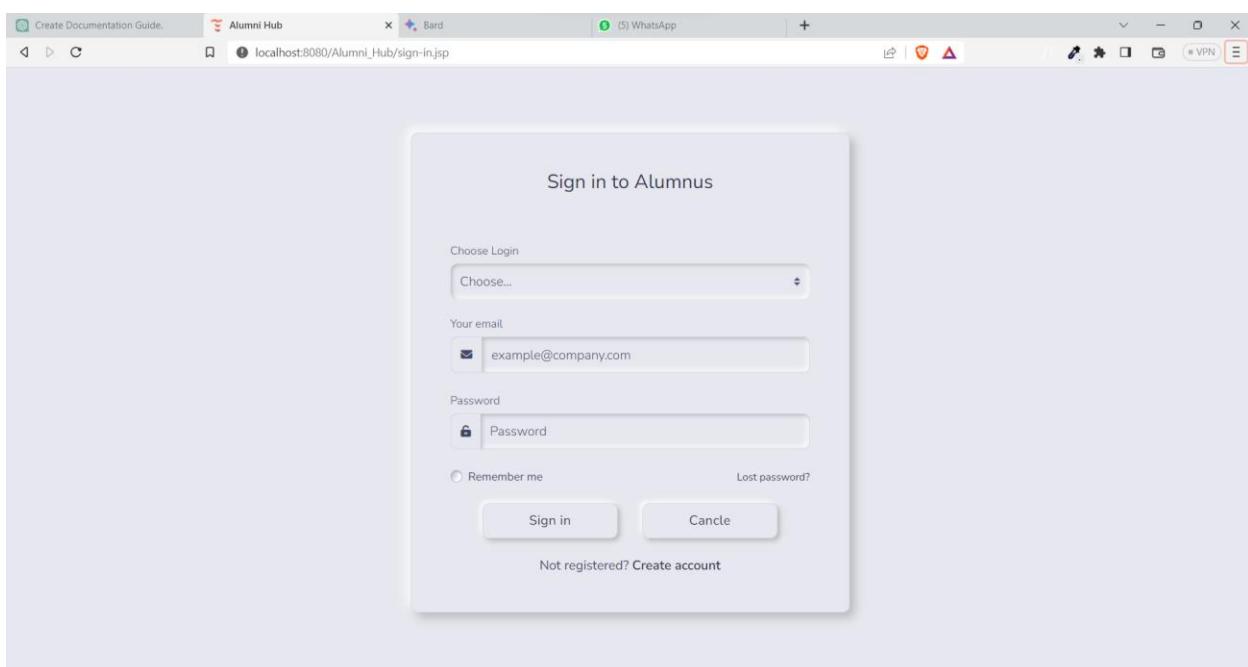
Password

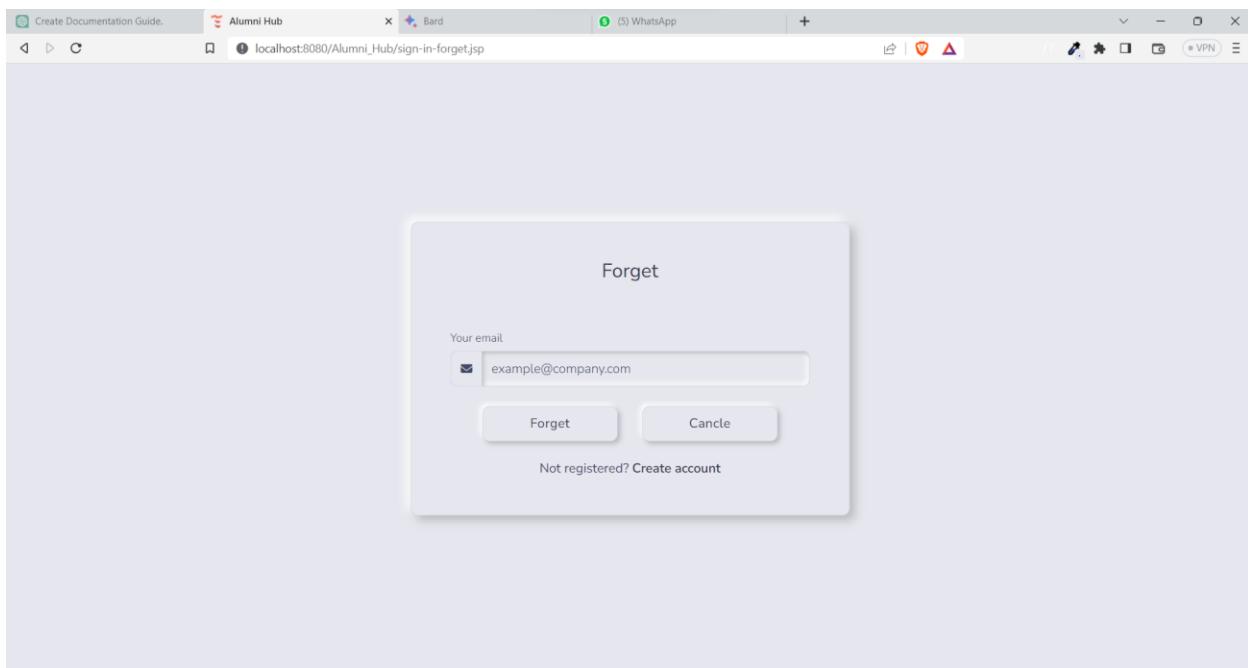
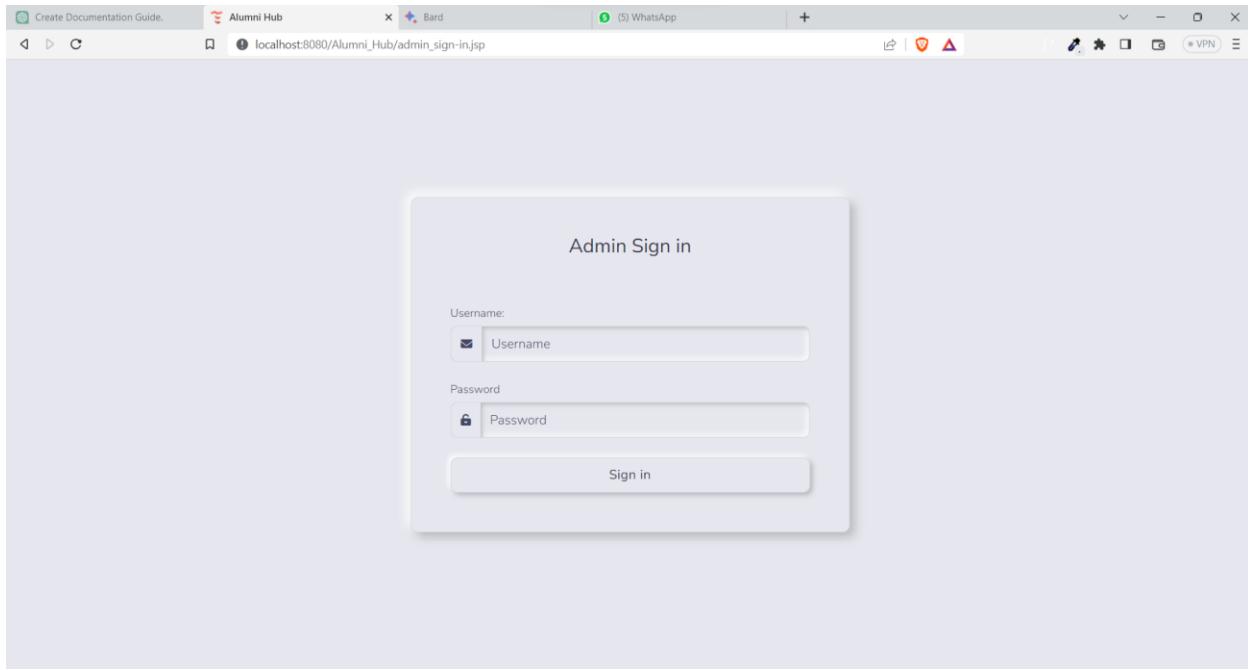
Password

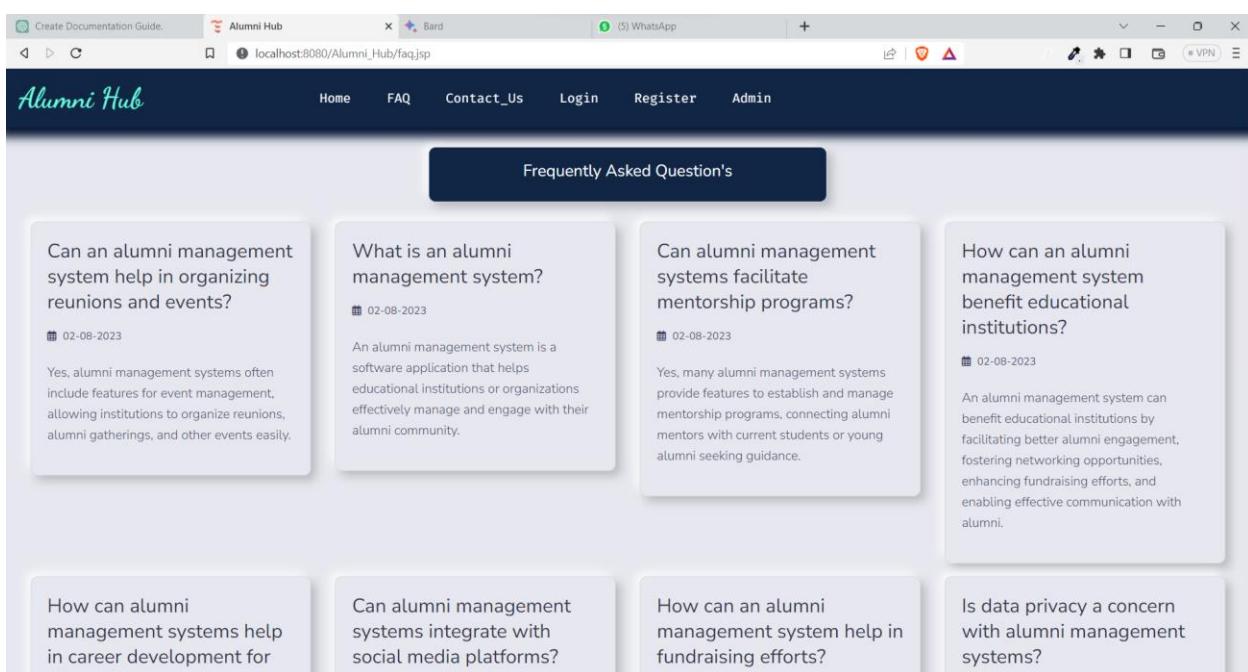
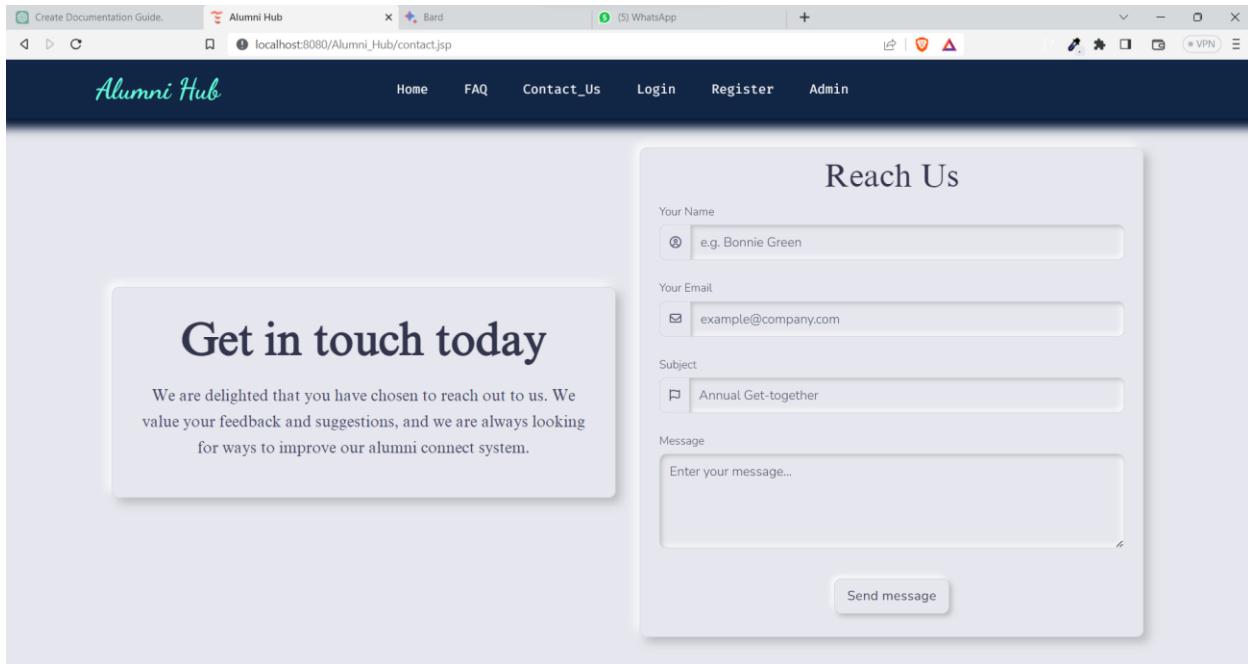
Remember me [Lost password?](#)

Sign in Cancel

Not registered? [Create account](#)







Admin Screen's :

The screenshot shows the 'Update Footer' page within the 'Alumni Hub' application. The page has a light gray background with a white central form area. The title 'Update Footer' is at the top. The form contains eight input fields arranged in two columns of four. Each field includes an icon and a placeholder text. At the bottom right of the form is a 'Apply Changes' button.

Contact Number	Twitter ID
+91 8552804872	shubhamsahane27
Email address	Facebook ID
info@Alumniius.edu	shubhamsahane27
Address	Instagram ID
Akurdi, Pune, Maharashtra, India	mr.silent_99V2.0
Address Map Link	Github ID
https://goo.gl/maps/Px9wK6MDeB2AePtU	SilentVirus27
Website	Website
	https://www.example.com

Apply Changes

The screenshot shows the 'Student Database' page within the 'Alumni Hub' application. The title 'Student Database' is at the top. Below it is a message: 'After adding only student can register into system !!'. The page features four input fields for 'Your First | Last Name', 'Your email', 'Batch', and 'Mobile Number', each with a placeholder and a small icon. At the bottom of these fields is a green 'Apply Changes' button. Below this is a data table with columns: Sr.No, Student Name, Email, Contact No., Batch, and Action. Two student entries are listed: Shubham Sahane and Roshan Muneshwar. Each entry has a red trash can icon in the 'Action' column. At the bottom left is a pagination message 'Showing 1 to 2 of 2 entries'. At the bottom right are buttons for 'Copy', 'CSV', 'Print', 'PDF', 'Excel', 'Previous', '1', and 'Next'.

Sr.No	Student Name	Email	Contact No.	Batch	Action
1	Shubham Sahane	chaitanya@gmail.com	+918552804872	2023-2024	
2	Roshan Muneshwar	roshanmuneshwar143@gmail.com	+91 9309887572	2022-2023	

Show 10 entries

Search:

Showing 1 to 2 of 2 entries

Copy CSV Print PDF Excel Previous 1 Next

localhost:8080/Admin_Hub/admin_student.jsp?typ=cs

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

Current Student

Show 10 entries Search:

Sr.No	Student Name	Contact No.	Email	Gender	Batch	Action
1	Shubham Sahane	8552804872	student@gmail.com	Male	2023-2024	
2	Roshan Muneshwar	973207886	roshanmuneshwar143@gmail.com	Male	2022-2023	

Showing 1 to 2 of 2 entries

Copy CSV Print PDF Excel Previous 1 Next

localhost:8080/Admin_Hub/view_profile.jsp?id=%2713%27

localhost:8080/Admin_Hub/admin_student.jsp?typ=a

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

Search Alumni's

Show 10 entries Search:

Sr.No	Alumni Name	Batch	Company	Designation	Location	Action
1	Shubham Sahane	2023-2024	TCS	Full Stack Developer	Hinjewadi,Pune	

Showing 1 to 1 of 1 entries

Copy CSV Print PDF Excel Previous 1 Next

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

Teacher Database

After adding only teacher can register into system !!

Your First	Last Name	Your email	Mobile Number
<input type="text" value="Fname"/>	<input type="text" value="Lname"/>	<input type="text" value="example@company.com"/>	<input type="text" value="e.g. +91 xxxxxxxxx"/>

Apply Changes

Show 10 entries Search:

Sr.No	Teacher Name	Email	Contact No.	Action
1	Shubham Sahane	teacher@gmail.com	+918552804872	

Showing 1 to 1 of 1 entries

Copy CSV Print PDF Excel

Previous 1 Next

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

Add Teacher

All Teacher

Search Teacher

Show 10 entries Search:

Sr.No	Teacher Name	Email	Contact No	Address	Gender	Action
1	Shubham Sahane	teacher@gmail.com	8552804872	Pune, India	Male	

Showing 1 to 1 of 1 entries

Copy CSV Print PDF Excel

Previous 1 Next

localhost:8080/Alumni_Hub/admin_teacher.jsp?typ=at#

The screenshot shows a web browser window with the URL `localhost:8080/Alumni_Hub/admin_event.jsp?typ=ad`. The page title is "Alumni Hub". A central modal dialog is open with the heading "Add Event". The form contains fields for "Title" (with placeholder "Title...") and "Details" (with placeholder "Put Some Details About Event"). There is also a file upload section with "Choose file" and "Browse" buttons, and a final "Add" button at the bottom.

The screenshot shows a web browser window with the URL `localhost:8080/Alumni_Hub/admin_event.jsp?typ=ae`. The page title is "Alumni Hub". A table titled "All Event's" displays two entries. The columns are "Sr.No", "Title", "Details", "Added By", "Image", and "Action". The first entry has Sr.No 1, Title "Came to celebrate", Details "Hello everyone From MCA 2023 come to college to celebrate college last day", Added By "Shubham Sahane", and Action buttons (pink square with trash icon and red square with trash icon). The second entry has Sr.No 2, Title "Laptop available for", Details "abcd", Added By "Shubham Sahane", and Action buttons (pink square with trash icon and red square with trash icon). At the bottom, there are buttons for "Copy", "CSV", "Print", "PDF", "Excel", and navigation links "Previous", "1", and "Next".

Sr.No	Title	Details	Added By	Image	Action
1	Came to celebrate	Hello everyone From MCA 2023 come to college to celebrate college last day	Shubham Sahane		
2	Laptop available for	abcd	Shubham Sahane		

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

Add Photo

Title

Photo Details

Waiting for mstripe.com...

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

All Photo

Sr.No	Title	Details	Added By	Date	Image	Action
1	Test 1	Testing Gallery photo	Shubham Sahane	31-07-2023		
2	Test 2	Testing Gallery photo	Shubham Sahane	02-08-2023		
3	Test 3	Testing Gallery photo	Shubham Sahane	02-08-2023		
4	Test 4	Testing Gallery photo	Shubham Sahane	02-08-2023		
5	Test 5	Testing Gallery photo	Shubham Sahane	02-08-2023		
6	Pratic Bro	abcdssss	Shubham Sahane	05-08-2023		

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

Session

Session Name

Choose a date Time

16	18
17	19
18	20
19	21
20	22
21	23
22	24

Session Details

Session Join Link

Add

Alumni Hub

Home Footer Student Teacher Event Gallery Session FAQ ContactUs Newsletter Logout

All Session

Show 10 entries Search:

Sr.No	Title	Details	Added By	Date & Time	Join Link	Joined	Action
1	Webinar on Python	We Will Check Some CRUD Operations With python Django.	Shubham Sahane	08/01/2023 : 17:00	https://www.example.com	1	
2	Webinar on Css	hello everyone we are arranging webinar on CSS	Shubham Sahane	08/23/2023 : 01:50	https://www.example.com	1	

Showing 1 to 2 of 2 entries

[Copy](#) [CSV](#) [Print](#) [PDF](#) [Excel](#)

Previous 1 Next

localhost:8080/Alumni_Hub/view_participant.jsp?id=9

The screenshot shows the 'FAQ' section of the Alumni Hub application. At the top, there's a header bar with tabs for Home, Footer, Student, Teacher, Event, Gallery, Session, FAQ, ContactUs, Newsletter, and Logout. Below the header, a title 'Frequently Asked Question's' is centered. There are two input fields: 'Question' and 'Answer', each with a placeholder 'Question...'. A green 'Add' button is located to the right of the answer field. Below these fields, there are buttons for 'Show 10 entries' and a search bar. The main content area displays a table with three rows of data. The columns are labeled Sr.No, Question, Answer, Date, and Action. Each row contains a delete icon (red bin) in the Action column.

Sr.No	Question	Answer	Date	Action
1	Can an alumni management system help in organizing reunions and events?	Yes, alumni management systems often include features for event management, allowing institutions to organize reunions, alumni gatherings, and other events easily.	02-08-2023	
2	What is an alumni management system?	An alumni management system is a software application that helps educational institutions or organizations effectively manage and engage with their alumni community.	02-08-2023	
3	Can alumni management systems facilitate mentorship programs?	Yes, many alumni management systems provide features to establish and manage mentorship programs, connecting alumni mentors with current students or young alumni seeking	02-08-2023	

The screenshot shows the 'Contact' section of the Alumni Hub application. At the top, there's a header bar with tabs for Home, Footer, Student, Teacher, Event, Gallery, Session, FAQ, ContactUs, Newsletter, and Logout. A modal dialog box titled 'Open Mail?' is displayed in the center, asking if the user wants to open the application. It includes a checkbox for 'Always allow http://localhost:8080 to open links of this type in the associated app' and two buttons: 'Open Mail' (blue) and 'Cancel' (orange). Below the modal, the main content area shows a table of contact entries. The columns are labeled Sr.No, Name, Email, Subject, Message, and Action. Each row contains a blue mail icon and a red bin icon in the Action column. At the bottom of the table, there are buttons for Copy, CSV, Print, PDF, and Excel, along with navigation links for Previous, Next, and a page number (1).

Sr.No	Name	Email	Subject	Message	Action
1	Shubham Sahane	shubhamsahane2710@gmail.com	Have a serious talk	I have Some Issue With My Login	
2	Shubham Sahane	shubhamsahane2710@gmail.com	Have a serious talk	qdwrdw	
3	Shubham Sahane	alumni@gmail.com	Have a serious talk	qww	

Alumni Hub

Send Newsletter

Subject:

Content:

Create Newsletter Content

Send

Alumni:

Alumni Hub

Profile

Personal Details

Add Job Details

View Job Details

Shubham Sahane

Full Stack Developer

Hellow guys i am shubham and this is the bio text for sampleHellow guys i am shubham and this is the bio text for sample

Email:	alumni@gmail.com	Gender:	Male
Birth Date:	10/27/1999	Mobile:	9879879876
Profession:	Full Stack Java	Skill:	Java,HTML,CSS
Address:	Hinjewadi,Pune	Edit	

[Twitter](#) [Facebook](#) [Instagram](#) [LinkedIn](#) [YouTube](#)

localhost:8080/Alumni_Hub/alumni_profile.jsp?typ=own_profile

Alumni Hub

Home Profile Message Session Gallery Job Post Event's Logout

Your First | Last Name: Shubham Sahane

Choose Birth Date: 10/27/1999

Mobile Number: 9879879876

Gender: Male

Your email: alumni@gmail.com

User Type: Alumni

Password: Password

Confirm Password: Confirm password

Instagram: @ Mr.silent_99v2.0

Twitter: shubhamsahane27

Facebook: shubhamsahane27

GitHub: SilentVirus27

Profession: Full Stack Java

Skill: Java,HTML,CSS

Website: www.example.com

Address: Hinjewadi ,Pune

About: Hello guys i am shubham and this is the bio text for sampleHello guys.i am shubham and this is the bio text for sample

Update

Alumni Hub

Alumnus

Subscribe

Join our mailing list. We write rarely, but only the best.

Create Documentation Guide. Alumni Hub Bard (5) WhatsApp

localhost:8080/Alumni_Hub/alumni_add_jobprofile.jsp

Alumni Hub Home Profile Message Session Gallery Job Post Event's Logout

Add Job Details

Company Name:

Company City:

Designation:

Joining Date

Leaving Date

Working Environment:

Is this your Current Job

Create Documentation Guide. Alumni Hub Bard (5) WhatsApp

localhost:8080/Alumni_Hub/alumni_profile.jsp?typ=job_profile

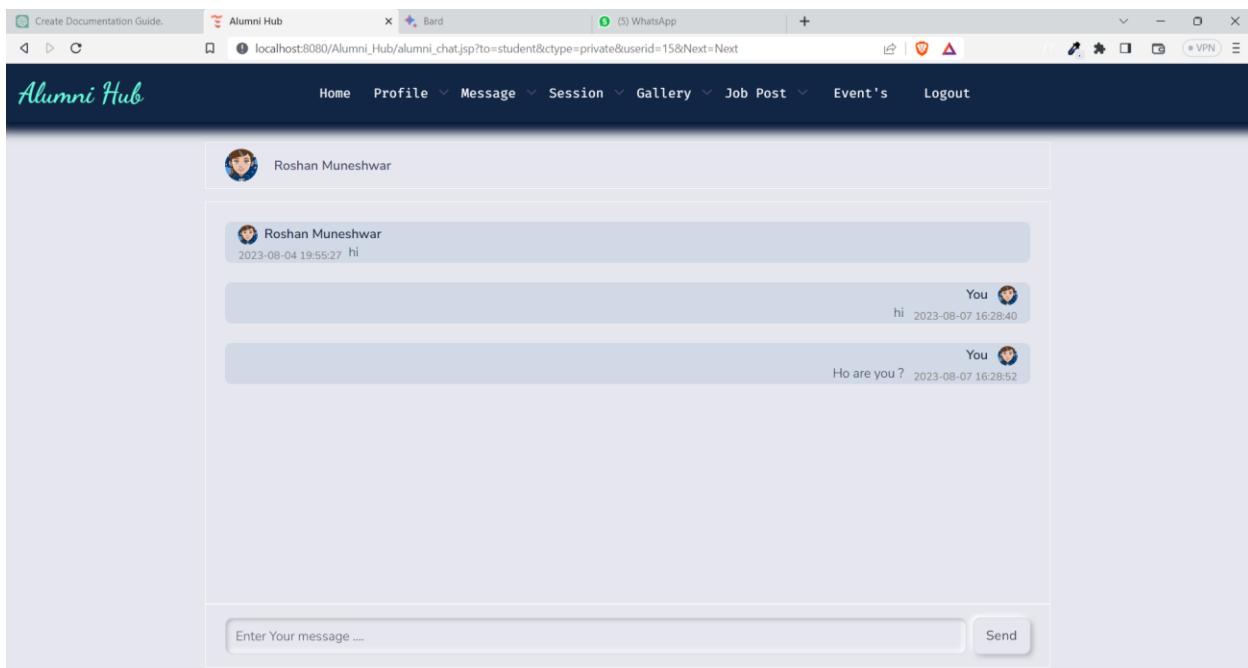
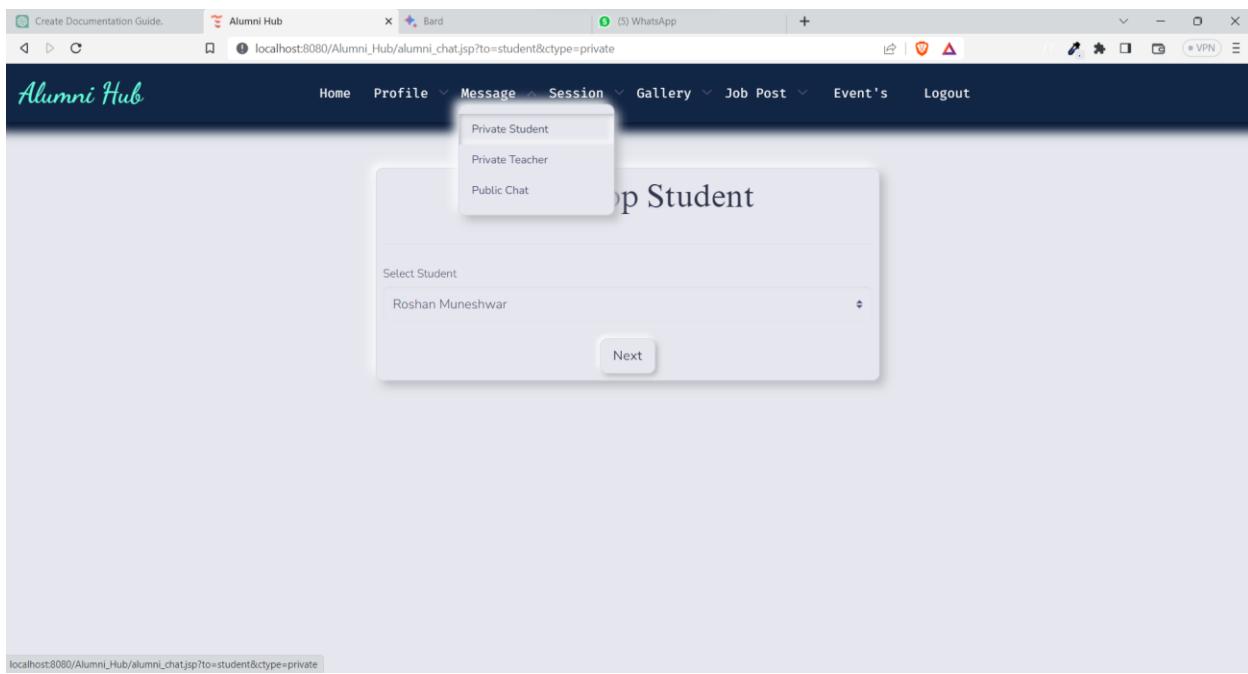
Alumni Hub Home Profile Message Session Gallery Job Post Event's Logout

Personal Details Add Job Details View Job Details

Alumni Job Profile

07/31/2023 :: 08/01/2023 Hinjewadi,Pune TCS Full Stack Developer Good Environment To Work 	07/31/2023 :: Till Date Hinjewadi,Pune TCS Full Stack Developer Good and healthy Environment to work 
---	--

localhost:8080/Alumni_Hub/alumni_profile.jsp?typ=job_profile



Create Documentation Guide. Alumni Hub Bard (5) WhatsApp

localhost:8080/Alumni_Hub/alumni_arrange_session.jsp?typ=ad

Alumni Hub Home Profile Message Session Gallery Job Post Event's Logout

Session

Session Name

Choose a date Time

Session Details

Session Join Link

Create Documentation Guide. Alumni Hub Bard (5) WhatsApp

localhost:8080/Alumni_Hub/alumni_arrange_session.jsp?typ=view

Alumni Hub Home Profile Message Session Gallery Job Post Event's Logout

All Session

Session Name	Arranged By	Date And Time	Action
Webinar on Python	Shubham Sahane	08/01/2023 : 17:00	<input type="button" value="View Participant"/> <input checked="" type="button" value="Edit Session"/> <input type="button" value="Delete Session"/>
Webinar on Css	Shubham Sahane	08/23/2023 : 00:00	<input type="button" value="View Participant"/> <input checked="" type="button" value="Edit Session"/> <input type="button" value="Delete Session"/>

The screenshot shows a participant list for a session named "Webinar on Css" on "08/23/2023 : 01:50". There is one entry:

Sr. No.	Student Name	Email	Skill	Action
1	Shubham Sahane	student@gmail.com	Java,HTML,CSS,JavaScript,ReactJS	

Buttons at the bottom include Copy, CSV, Print, PDF, Excel, Previous, 1, and Next.

The screenshot shows a gallery view with several placeholder cards labeled "TEST". A modal window titled "Upload Photo" is open, showing tabs for "All" and "2023" with "Gallery" selected. A snipping tool interface is overlaid on the bottom right, showing a screenshot of the gallery and a message: "Screenshot copied to clipboard and saved Select here to mark up and share the image".

The screenshot shows a web browser window with the URL `localhost:8080/Alumni_Hub/alumni_add_job.jsp`. The page title is "Alumni Hub". The main content is a form titled "Add Job Post" with the following fields:

- Designation: A text input field with a placeholder "Title..." and a pen icon.
- Company: A text input field with a placeholder "e.g. TCS" and a company icon.
- Salary: A text input field with a placeholder "₹ 450000 -550000" and a rupee symbol icon.
- Skill: A text input field with a placeholder "Java,HTML,CSS,MySQL" and a list icon.
- Job Details: A text input field with a placeholder "Put Some Details".

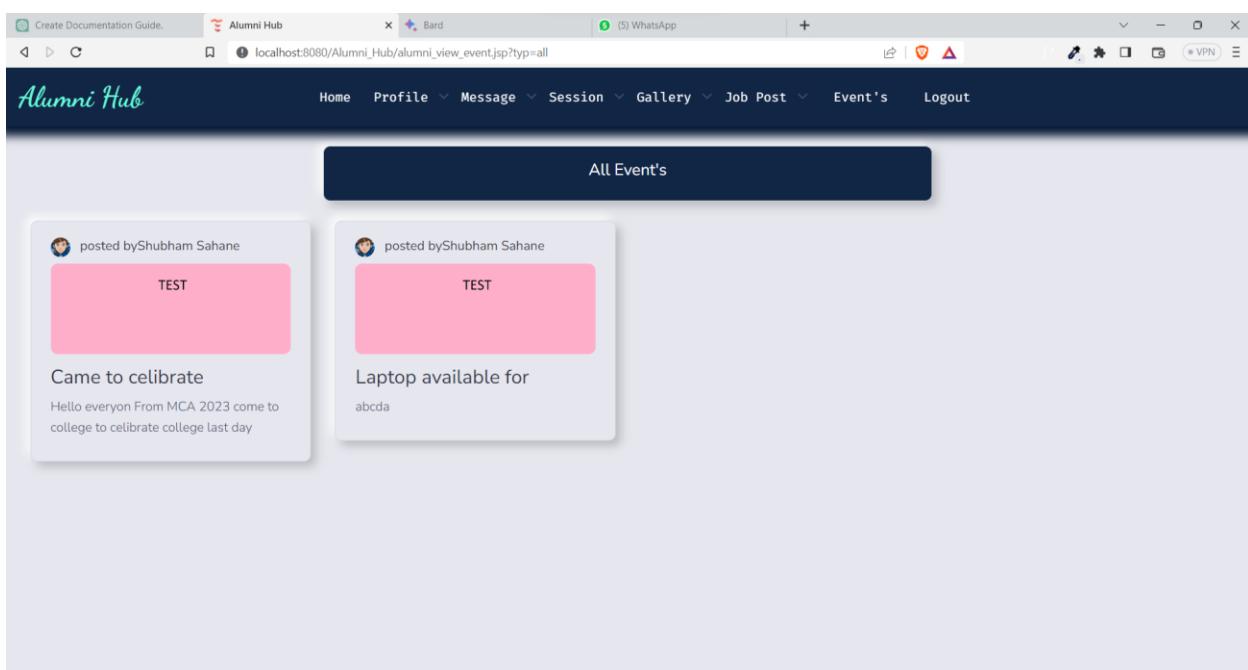
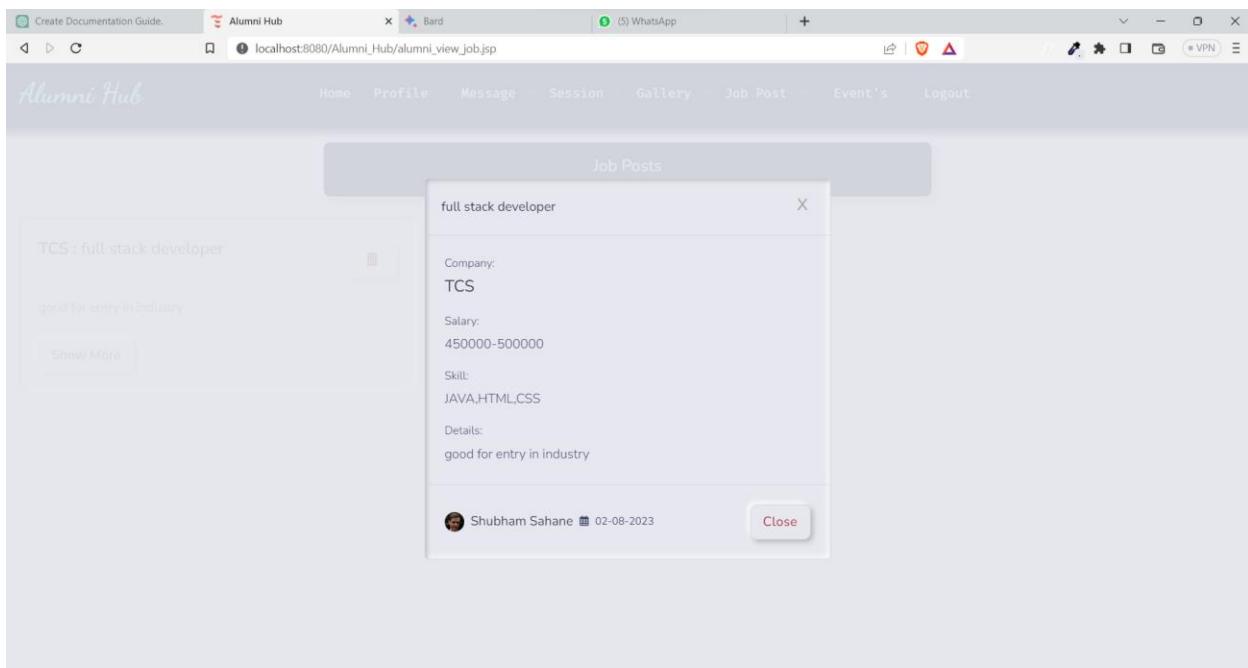
At the bottom right of the form is a blue "Add" button.

The screenshot shows a web browser window with the URL `localhost:8080/Alumni_Hub/alumni_view_job.jsp`. The page title is "Alumni Hub". The main content is a list of job posts under a header "Job Posts". The first post is displayed in a card format:

TCS : full stack developer Delete

good for entry in industry

[Show More](#)



Student:

The screenshot shows a web browser window titled "Alumni Hub" with the URL "localhost:8080/Alumni_Hub/student_search.jsp". The page displays a table titled "Search Alumni's" with one entry. The columns are Sr.No, Alumni Name, Batch, Company, Designation, Location, and Action. The data row is: 1, Shubham Sahane, 2023-2024, TCS, Full Stack Developer, Hinjewadi,Pune, with a refresh and email icon in the Action column. Below the table, it says "Showing 1 to 1 of 1 entries" and has buttons for Copy, CSV, Print, PDF, and Excel. At the bottom right are Previous, Next, and page number 1 buttons.

The screenshot shows a web browser window titled "Alumni Hub" with the URL "localhost:8080/Alumni_Hub/student_session.jsp?typ=joined". The page displays a "Joined Session" header. Below it are two cards: "Webinar on Python" and "Webinar on Css". Each card has details like Arranged By (Shubham Sahane), Date And Time, and a session link (<https://www.example.com>). Each card also has "Joining Student: 1" and "Leave" buttons.

The screenshot shows a web browser window with the URL localhost:8080/Alumni_Hub/student_session.jsp?typ=all. The page title is "Alumni Hub". The main content area displays two session cards:

- Webinar on Python**
 - Arranged By: Shubham Sahane
 - Date And Time: 08/01/2023 : 17:00
 - Description: We Will Check Some CRUD Operations With python Django.
 - Session Link: <https://www.example.com>
 - Joining Student: 1
 - [Enroll](#)
- Webinar on Css**
 - Arranged By: Shubham Sahane
 - Date And Time: 08/23/2023 : 01:50
 - Description: hello everyone we are arranging webinar on CSS
 - Session Link: <https://www.example.com>
 - Joining Student: 1
 - [Enroll](#)

Teacher :

The screenshot shows a web browser window with the URL localhost:8080/Alumni_Hub/teacher_search.jsp. The page title is "Alumni Hub". The main content area displays a search results table titled "Search Alumni's":

Sr.No	Alumni Name	Batch	Company	Designation	Location	Action
1	Shubham Sahane	2023-2024	TCS	Full Stack Developer	Hinjewadi,Pune	

Showing 1 to 1 of 1 entries

Buttons at the bottom: Copy, CSV, Print, PDF, Excel

localhost:8080/Alumni_Hub/teacher_c_search.jsp

Search Current Student's

Sr.No	Student Name	Contact	Email	Gender	Location	Action
1	Shubham Sahane	8552804872	student@gmail.com	Male	At Hinjewadi,Pune 411048 ,Maharashtra	 
2	Roshan Muneshwar	973207886	roshanmuneshwar143@gmail.com	Male	pune ,India	 

Showing 1 to 2 of 2 entries

Copy CSV Print PDF Excel

Previous 1 Next

localhost:8080/Alumni_Hub/teacher_profile.jsp

Shubham Sahane

Teacher

Hello i am teacher and im teaching student at sinhgad institute

Email:	teacher@gmail.com	Gender:	Male
Birth Date:	10/27/1999	Mobile:	8552804872
Profession:	Teacher	Skill:	RDBMS Java
Address:	Pune ,India	Edit	

CHAPTER 4

Chapter:4 Code

4.1 Code Snippets:

Login :

```
@WebServlet("/SignInUser")
public class SignInUser extends HttpServlet {
    private static final long serialVersionUID = 1L;
    String utype,email,password,query;
    DbConnection dbConnection;
    static Connection con;

    public void init(ServletConfig config) throws ServletException {
        try {
            dbConnection=new DbConnection();
            con=dbConnection.getConnection();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session=request.getSession();

        utype=request.getParameter("utype");
        email=request.getParameter("email");
        password=request.getParameter("password");
        MD5_Password md5=new MD5_Password();
        String enc_pwd=md5.encryptPassword(password);

        query="SELECT * FROM user_details WHERE email='"+email+"' And
pwd='"+enc_pwd+"'";
        try
        {
            PreparedStatement ins=con.prepareStatement(query);
            System.out.println("PS :"+ins);
            ResultSet Rs= ins.executeQuery();

            /*ResultSet Rs=dbConnection.selectOperation(query);*/
            if(Rs.next())
            {

                System.out.println(Rs.getString("user_type").trim()+":"+utype.trim());
                if(Rs.getString("user_type").trim().equals(utype.trim())){
                    session.setAttribute("utype",
Rs.getString("user_type").trim());
                    session.setAttribute("fname",
Rs.getString("fname").trim());
                    session.setAttribute("lname",
Rs.getString("lname").trim());
                }
            }
        }
    }
}
```

```

        session.setAttribute("mobile",
Rs.getString("mobile").trim());
        session.setAttribute("email",
Rs.getString("email").trim());
        session.setAttribute("dob",
Rs.getString("dob").trim());
        session.setAttribute("gender",
Rs.getString("gender").trim());
String c_un=Rs.getString("fname")+" "+Rs.getString("lname");
        session.setAttribute("c_un",c_un);
        session.setAttribute("id", Rs.getInt("id"));
        System.out.println(c_un + " :" +password+
:enc_pwd:+enc_pwd);

    if(Rs.getString("user_type").trim().equals("Student")){
        System.out.println("Student Login Success");
        response.sendRedirect("student_home.jsp");
    }else
    if(Rs.getString("user_type").trim().equals("Teacher")){
        System.out.println("Teacher Login Success");
        response.sendRedirect("teacher_home.jsp");
    }else
    if(Rs.getString("user_type").trim().equals("Alumni")){
        System.out.println("Alumini Login Success");
        response.sendRedirect("alumni_home.jsp");
    }else {
        System.out.println("Login Error Occur For
User"+Rs.getString("user_type").trim());
        response.sendRedirect("sign-
in.jsp?message=Failed!!! Please Try Again");
    }
    }else {System.out.println("Login Error Occur For
User"+Rs.getString("user_type").trim()+" Select correct user type");
        response.sendRedirect("sign-in.jsp?message=Failed!!!
Select correct user type");
    }
} else {
    System.out.println("Login Failed!!! Please Try Again");
    response.sendRedirect("sign-in.jsp?message=Failed!!!
Please Try Again");
}
} catch (SQLException e) {
    System.out.println("Login Exception"+e);
}
}
}

}

```

Logout :

```
@WebServlet("/SignOut")
```

```

public class SignOut extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        HttpSession session = request.getSession(false); // Retrieve the session
object if it exists

        if (session != null) {
            session.invalidate(); // Invalidate the session to remove all session
attributes
        }
        System.out.println("Sign out successfully!!!");
        response.sendRedirect("home.jsp"); // Redirect the user to the sign-in page
    }
}

```

Forget Pass:

```

@WebServlet("/SignInUserForget")
public class SignInUserForget extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String SMTP_HOST = "smtp.elasticemail.com";
    private static final int SMTP_PORT = 2525;
    private static final String SMTP_USERNAME = "alumni.connect.hub@gmail.com";
    private static final String SMTP_PASSWORD =
"C0213F9F7EB70041A0E5F7A50DC32CFFD018";

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    // Get the email entered by the user
    String email = request.getParameter("email");

    // Check if the email exists in the database
    boolean emailExists = checkEmailExists(email);

    if (emailExists) {
        // Check if OTP has already been sent for this email
        boolean otpSent = hasOTPSent(request);

        if (!otpSent) {
            // Generate a random OTP
            String otp = generateOTP();

            // Send the OTP to the user's email
            boolean otpSentSuccess = sendOTP(email, otp);

            if (otpSentSuccess) {
                // Store the OTP and email in session for verification
                HttpSession session = request.getSession();
                session.setAttribute("email", email);
                session.setAttribute("otp", otp);
                session.setAttribute("otpSent", true);
            }
        }
    }
}

```

```

        // Redirect to enter new password and OTP page
        response.sendRedirect("sign-in-forget-new.jsp");
        return;
    } else {
        // Handle the case when OTP sending fails
        System.out.println("Otp send Fail");
        response.sendRedirect("sign-in-
forget.jsp?message=otp_sending_failed");
        return;
    }
} else {
    // Redirect to enter new password and OTP page if OTP has already
been sent
    System.out.println("redirect to new password");
    response.sendRedirect("sign-in-forget-new.jsp");
    return;
}
} else {
    // Handle the case when email does not exist in the database
    System.out.println("email_not_found");
    response.sendRedirect("sign-in-forget.jsp?message=email_not_found");
    return;
}
}

private boolean checkEmailExists(String email) {
    // Implement your database logic here to check if the email exists
    // Return true if email exists, false otherwise
    // Example:
    // return userDao.checkEmailExists(email);
    // Replace 'userDao' with your actual DAO or database access code
    return true; // Placeholder implementation
}

private boolean hasOTPSent(HttpServletRequest request) {
    HttpSession session = request.getSession();
    return session.getAttribute("otpSent") != null;
}

private String generateOTP() {
    // Generate a random 6-digit OTP
    Random random = new Random();
    int otp = 100000 + random.nextInt(900000);
    return String.valueOf(otp);
}

private boolean sendOTP(String email, String otp) {
    // SMTP server properties
    Properties props = new Properties();
    props.put("mail.smtp.host", "SMTP_HOST");
    props.put("mail.smtp.port", "SMTP_PORT");
    props.put("mail.smtp.auth", "true");

    // Create a session with SMTP server

```

```
Session session = Session.getInstance(props, new Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(SMTP_USERNAME, SMTP_PASSWORD);
    }
});

try {
    // Create a new email message
    Message message = new MimeMessage(session);
    message.setFrom(new InternetAddress(SMTP_USERNAME));
    message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(email));
    message.setSubject("Password Reset OTP");
    message.setText("Your OTP for password reset is: " + otp);

    // Send the email
    Transport.send(message);
    return true;
} catch (MessagingException e) {
    e.printStackTrace();
    return false;
}
}
```

Check Otp :

```
@WebServlet("/CheckOTP")
public class CheckOTP extends HttpServlet {
    private static final long serialVersionUID = 1L;
    DbConnection dbConnection;
    public void init(ServletConfig config) throws ServletException {
        try {
            dbConnection=new DbConnection();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Connection Exception"+e);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //Get the email, OTP, and new password from the user
        String email = request.getParameter("email");
        String otp = request.getParameter("otp");
        String newPassword = request.getParameter("pass");
        String confirm = request.getParameter("confirmpass");

        // Check if the entered OTP is correct
    }
}
```

```

boolean isOtpCorrect = verifyOTP(request,email, otp);

if(newPassword.equals(confirm)){

    MD5_Password md5=new MD5_Password();
    String enc_password=md5.encryptPassword(newPassword);

    if (isOtpCorrect) {
        // Update the password in the database

        boolean passwordUpdated = updatePassword(email, enc_password);

        if (passwordUpdated) {
            // Password updated successfully, redirect to sign in page
            response.sendRedirect("sign-in.jsp?message=Password Updated
!!");

        } else {
            // Handle the case when password update fails
            response.sendRedirect("sign-in-
forget.jsp?message=password_update_failed");
        }
    } else {
        // Handle the case when OTP verification fails
        response.sendRedirect("sign-in-forget-
new.jsp?message=incorrect_otp");
    }
} else {response.sendRedirect("sign-in-forget-new.jsp?message=password not
match:"+newPassword+"|"+confirm);}
}

private boolean verifyOTP(HttpServletRequest request, String email, String
otp) {
    // Retrieve the stored OTP and email from session
    HttpSession session = request.getSession();
    String storedEmail = (String) session.getAttribute("email");
    String storedOTP = (String) session.getAttribute("otp");
    return email.equals(storedEmail) && otp.equals(storedOTP); //Placeholder implementation
}

private boolean updatePassword(String email, String enc_password) {
    String query = "UPDATE `alumini_hub`.`user_details` SET `pwd` = '" +
enc_password + "' WHERE `email` = '" + email + "'";
    boolean flag=false;
    try {
        int status=dbConnection.inupdelOperation(query);

        if(status>0){
            System.out.println("Row Affected"+status);
            flag= true;
        }else{
            System.out.println("Registration Fail");
            System.out.println(status);
            flag= false;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        }
    }
    catch (SQLException e)
    {
        System.out.println("Registration Exception "+e);
    }

    return flag;
}
}

```

Encrypt Password :

```

public class MD5_Password
{
    public String encryptPassword(String pwd)
    {
        String enc_pwd="";
        try
        {
            MessageDigest md = MessageDigest.getInstance("MD5");
            byte[] inputBytes = pwd.getBytes();
            byte[] hashBytes = md.digest(inputBytes);
            StringBuilder sb = new StringBuilder();
            for (byte b : hashBytes)
            {
                sb.append(String.format("%02x", b));
            }
            enc_pwd= sb.toString();
            // System.out.println("MD5 hash: " + enc_pwd);
        } catch (NoSuchAlgorithmException e)
        {
            e.printStackTrace();
        }
        return enc_pwd;
    }
}

```

Send Newsletter :

```

@.WebServlet("/Newsletter")
public class Newsletter extends HttpServlet{
    static Connection con;
    boolean success;
    private static final long serialVersionUID = 1L;
    private static final String SMTP_HOST = "smtp.elasticemail.com";
    private static final int SMTP_PORT = 2525;
    private static final String SMTP_USERNAME = "alumni.connect.hub@gmail.com";
    private static final String SMTP_PASSWORD = "C0213F9F7EB70041A0E5F7A50DC32CFFD018";

    public void init(ServletConfig config) throws ServletException
    {
        try {

```

```

        con=DbConnection.getConnection();
    } catch (Exception e) {
        System.out.println("Connection Error Exception "+e);
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
{
    try
    {
        String subject = request.getParameter("subject");
        String content = request.getParameter("content");

        PreparedStatement ps1=con.prepareStatement("SELECT * FROM
alumini_hub.subscribe");
        System.out.println("PS :"+ps1);
        ResultSet subscribers=ps1.executeQuery();
        int i=0;
        while(subscribers.next()){
            success = sendMail(subscribers.getString("email"),subject,
content);
            System.out.println("Successfully Send
to:"+subscribers.getString("email"));
            i++;
        }

        response.sendRedirect("admin_newsletter.jsp?message=
Successfully");
        System.out.println("Newsletter Send To:"+i+" Subscriber");
    }
    catch (Exception e)
    {
        System.out.println("Error Exception "+e);
        e.printStackTrace();
    }
}

private boolean sendMail(String email,String subject, String content) {
    // SMTP server properties
    Properties props = new Properties();
    props.put("mail.smtp.host", SMTP_HOST);
    props.put("mail.smtp.port", SMTP_PORT);
    props.put("mail.smtp.auth", "true");

    // Create a session with SMTP server
    Session session = Session.getInstance(props, new Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(SMTP_USERNAME, SMTP_PASSWORD);
        }
    });

    try {
        // Create a new email message

```

```
        Message message = new MimeMessage(session);
        message.setFrom(new InternetAddress(SMTP_USERNAME));
        message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(email));
        message.setSubject("Alumni Hub :" + subject);
        message.setText(content);
        // Send the email

        Transport.send(message);
        return true;
    } catch (MessagingException e) {
        e.printStackTrace();
        return false;
    }
}
```

Connection :

```
public class DbConnection {  
    PreparedStatement pst=null;  
    ResultSet rs=null;  
    private static Connection con;  
    String query=null;  
    public DbConnection() throws ClassNotFoundException, SQLException {  
        Class.forName("com.mysql.jdbc.Driver");  
        con=DriverManager.getConnection("jdbc:mysql://localhost:3308/alumini_hub", "root", "root");  
    }  
  
    public static Connection getConnection()  
    {  
        Connection con = null;  
        try  
        {  
            Class.forName("com.mysql.jdbc.Driver");  
            con =  
DriverManager.getConnection("jdbc:mysql://localhost:3308/alumini_hub", "root",  
"root");  
            return con;  
        } catch (Exception e) {  
            // TODO: handle exception  
            System.out.println("Exception is " + e);  
        }  
        return con;  
    }  
}
```

Add Event .Gallery :

```
@WebServlet("/ImageDB")
public class ImageDB extends HttpServlet{
    Connection con;
```

```

String plimage = "";
final String UPLOAD_DIRECTORY =
"C:/Users/technOrbit/Desktop/MCA_Project/Project Sem - IV/Alumni
Hub/WebContent/uploads";
public void init(ServletConfig config) throws ServletException
{
    try {
        con=DbConnection.getConnection();
    } catch (Exception e) {
        System.out.println("Connection Error Exception "+e);
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
{
    int i = 0;
    String[] data = new String[7];
    if (ServletFileUpload.isMultipartContent(request)) {

        try {

            List<FileItem> multipart = new ServletFileUpload(new
DiskFileItemFactory()).parseRequest(request);
            String FileExtention = "";
            long fileSize = 0;
            for (FileItem item1 : multipart) {
                if (!item1.isFormField()) {
                    plimage = new
File(item1.getName()).getName();
                    item1.write(new File(UPLOAD_DIRECTORY +
File.separator+ plimage));
                    plimage = item1.getName();
                    FileExtention = item1.getContentType();
                    System.out.println("file type is:" +
FileExtention);
                    fileSize = item1.getSize();
                    System.out.println("image :" + plimage);
                }
                if (item1.isFormField()) {
                    data[i] = item1.getString();
                    i++;
                }
            }
        try
        {
            if(data[0].equals("AddEvent")){
                PreparedStatement ps1 =
con.prepareStatement("INSERT INTO `alumini_hub`.`event_details` (`title`,
`event_details`, `pic_name`, `upload_by_id`, `upload_by`) VALUES ('"+data[1]+",
 '"+data[2]+", '"+plimage+"', '"+data[3]+"', '"+data[4]+"')");
                System.out.println("PS :" +ps1);
                int ii = ps1.executeUpdate();
                if(data[4]=="Admin"){

```

```

        if (ii > 0)
        {

response.sendRedirect("admin_event.jsp?typ=ad&message=Added Successfully");
    } else {

response.sendRedirect("admin_event.jsp?typ=ad&message=fail");
    }
} else{
    if (ii > 0)
    {

        response.sendRedirect(request.getHeader("Referer")+"?typ=ad&message=Added
Successfully");
    } else {

response.sendRedirect(request.getHeader("Referer")+"?typ=ad&message=fail");
    }
} else if(data[0].equals("AddGallery")){
    PreparedStatement ps2 =
con.prepareStatement("INSERT INTO `alumini_hub`.`gallery` (`title`, `details`,
`pic_name`, `upload_by_id`, `upload_by`, `date`) VALUES ('"+data[1]+', '"+data[2]+",
"+plimage+", '"+data[3]+", '"+data[4]+",DATE_FORMAT(NOW(), '%d-%m-%Y'))");
    System.out.println("PS :" +ps2);
    int ii = ps2.executeUpdate();
    if(data[4]=="Admin"){
        if (ii > 0)
        {

response.sendRedirect("admin_gallery.jsp?typ=ad&message=Added Successfully");
    } else {

response.sendRedirect("admin_gallery.jsp?typ=ad&message=fail");
    }
} else{
    if (ii > 0)
    {

        response.sendRedirect(request.getHeader("Referer")+"?typ=ad&message=Added
Successfully");
    } else {

response.sendRedirect(request.getHeader("Referer")+"?typ=ad&message=fail");
    }
}

} else {System.out.println("Error");}

} catch (Exception e)
{
    System.out.println("Exe:" + e);
    e.printStackTrace();
}

```

```

        } catch (Exception e) {
            System.out.println("Exe:" + e);
            e.printStackTrace();
        }
    }
}



## Sign Up:



```

@WebServlet("/SignUpUser")
public class SignUpUser extends HttpServlet {
 private static final long serialVersionUID = 1L;
 String fname, lname, dob, gender, email, utype, pass, confirmpass, phone, query, message, password, flag = "false";
 DbConnection dbConnection;
 static Connection con;
 public void init(ServletConfig config) throws ServletException {
 try {
 dbConnection=new DbConnection();
 con=dbConnection.getConnection();
 } catch (ClassNotFoundException | SQLException e) {
 System.out.println("Connection Exception"+e);
 }
 }

 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
 HttpSession session=request.getSession();

 utype=request.getParameter("utype");
 fname=request.getParameter("fname");
 lname=request.getParameter("lname");
 phone=request.getParameter("phone");
 email=request.getParameter("email");
 dob=request.getParameter("dob");
 gender=request.getParameter("gender");
 pass=request.getParameter("pass");
 confirmpass=request.getParameter("confirmpass");
 if(pass.equals(confirmpass)){
 flag="true";
 }else{response.sendRedirect("sign-in.jsp?message=Both Password Not Match");}
 if(flag.equals("true")){
 MD5_Password md5=new MD5_Password();
 password=md5.encryptPassword(pass);
 try {
 query="SELECT * FROM alumini_hub.student_db WHERE s_email='"+email+"'";
 PreparedStatement ins=con.prepareStatement(query);
 System.out.println("PS :" +ins);
 ResultSet Checkrs1= ins.executeQuery();
 if(Checkrs1.next()){

```


```

```

query="SELECT * FROM alumini_hub.student_db WHERE
s_email='"+email+"'";
PreparedStatement ins1=con.prepareStatement(query);
System.out.println("PS :"+ins1);
ResultSet Checkrs2= ins1.executeQuery();
if(Checkrs2.next()){
    query="INSERT INTO
`alumini_hub`.`user_details`(`user_type`, `fname`, `lname`, `mobile`, `email`, `dob`,
`gender`, `pwd`) VALUES
('"+utype+"','"+fname+"','"+lname+"','"+phone+"','"+email+"','"+dob+"','"+gender+"',
"+password+"')";
```

PreparedStatement

```

insusr=con.prepareStatement(query);
System.out.println("PS :"+insusr);
int status= insusr.executeUpdate();
```

/*int

```

status=dbConnection.inupdel0peration(query);*/
```

```

if(status>0){
    System.out.println("Row
Affected"+status);
    response.sendRedirect("sign-
in.jsp?message=User Registration Success!!");
}else{
    System.out.println("Registration
Fail");
    System.out.println(status);
    response.sendRedirect("sign-
up.jsp?message=Registration Failed Try Again!!");
}
}else{response.sendRedirect("sign-
up.jsp?message=User Already Exist Please Login !!");}
```

}else{response.sendRedirect("sign-up.jsp?message=Please
Ask your Administration to Add your Email into Student Database .");}

```

}
catch (SQLException e)
{
    System.out.println("Registration Exception "+e);
}
}
```

}

CHAPTER 5

Testing

Chapter: 5 Testing

5.1 Test Strategy:

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software testing can also be stated as the process of validating and verifying that a software program or application or product:

- **Test Objective:** The primary objective of testing is to ensure that the "Alum Hub" system meets the specified business and technical requirements. The system should work as expected, providing a user-friendly platform for alumni engagement and networking.
- **Testing Types:** Functional Testing: To verify that all functional requirements are implemented correctly and meet the intended behavior. Non-Functional Testing: To assess performance, security, usability, compatibility, and other non-functional aspects of the system. Integration Testing: To ensure seamless integration of various system components and modules. User Acceptance Testing (UAT): To involve stakeholders in testing and validate that the system meets their expectations.
- **Test Environment:** A dedicated testing environment will be set up to mimic the production environment without affecting real user data. Test data will be prepared, including sample alumni profiles, events, job postings, and other relevant information.
- **Test Scope:** The scope of testing will cover all key functionalities, including user registration, profile management, networking, event management, knowledge sharing sessions, job postings, and interactions with teachers and students.
- **Test Cases and Scenarios:** Test cases and test scenarios will be designed to cover both positive and negative test scenarios. Test cases will be created to validate each functional requirement and ensure proper system behavior.
- **Test Data Management:** Test data will be carefully prepared to represent different use cases and scenarios. Test data will include valid inputs, invalid inputs, and boundary conditions.

- Test Execution: Test cases will be executed systematically to identify defects and deviations from expected behavior. Test results will be recorded and tracked for proper defect management.
- Defect Tracking and Management: Defects will be logged and tracked throughout the testing process. Defect severity and priority will be assigned to facilitate timely resolution.
- Regression Testing: Whenever changes are made to the system, regression testing will be conducted to ensure that existing functionalities are not affected.
- Performance Testing: Performance testing will be conducted to assess system response times, throughput, and resource utilization under varying load conditions.
- Security Testing: Security testing will be performed to identify vulnerabilities and ensure that user data is protected.
- User Acceptance Testing (UAT): UAT will be conducted with the involvement of stakeholders to validate that the system meets their requirements and expectations.
- Documentation: Comprehensive test documentation will be prepared, including test plans, test cases, test results, and defect reports.
- Test Completion Criteria: Testing will be considered complete when all planned test cases have been executed, and critical defects have been resolved.
- Testing Sign-Off: After successful completion of testing and resolution of critical defects, the system will be ready for deployment.

By following this test strategy, the "Alum Hub" system will undergo thorough testing to ensure that it meets the business and technical requirements, functions as expected, and provides a reliable and user-friendly platform for alumni engagement and networking.

5.1.1 Scope of Testing :

The scope of testing for the "Alum Hub" system encompasses a comprehensive examination of all its functionalities and components to ensure that it meets the specified requirements and performs as expected. The testing scope includes various levels of testing and targets both functional and non-functional aspects of the system. The primary areas covered in the scope of testing are as follows:

1. Functional Testing:

- Verification of User Registration: Testing the user registration process to ensure that users can create accounts successfully and that their profiles are created accurately.
- Profile Management: Verifying that users can create, edit, and update their profiles with the required information.
- Networking and Search: Ensuring that the search functionality allows users to find and connect with other alumni based on different criteria.
- Interaction with Teachers and Students: Testing the interaction features to validate seamless communication between alumni, teachers, and current students.
- Event Management: Verifying the event creation, registration, and notification functionalities.
- Knowledge Sharing Sessions: Testing the organization and participation in knowledge-sharing sessions and workshops.
- Job Postings: Validating the job posting and application process.

2. Non-Functional Testing:

- Performance Testing: Evaluating the system's response times, load handling capacity, and resource utilization under different user loads.
- Usability Testing: Assessing the user interface and user experience to ensure ease of use and intuitive navigation.
- Security Testing: Identifying and addressing vulnerabilities to safeguard user data and system integrity.
- Compatibility Testing: Testing the system's compatibility with different web browsers, devices, and operating systems.
- Accessibility Testing: Verifying that the system is accessible to users with disabilities,

adhering to accessibility standards.

- Reliability Testing: Ensuring the system operates reliably with minimal downtime and system failures.
- 3. Integration Testing: Testing the seamless integration of different system components and modules to ensure smooth functioning as a whole.
- 4. User Acceptance Testing (UAT): Involving stakeholders to validate the system against their requirements and expectations.
- 5. Regression Testing: Conducting regression tests whenever changes are made to the system to ensure that existing functionalities are not affected.
- 6. End-to-End Testing: Testing the entire flow of user interactions, from registration to various features usage.
- 7. Boundary Testing: Verifying the system's behavior at the boundaries of input values and conditions.
- 8. Data Validation: Ensuring that the system handles data accurately and correctly validates user inputs.

The scope of testing is to ensure that the "Alum Hub" system is thoroughly assessed for functionality, performance, security, usability, and reliability. By conducting a comprehensive testing effort, the system's quality, accuracy, and adherence to requirements will be ensured, providing a stable and user-friendly platform for alumni networking and engagement.

5.1.2 Test Plan :

1. Introduction:

Purpose: The purpose of this test plan is to define the testing approach, scope, and objectives for the "Alum Hub" system.

Document Overview: This document outlines the testing strategy, test objectives, test scope, test deliverables, and the schedule for testing.

2. Test Objectives:

Validate the "Alum Hub" system meets the specified business and technical requirements.

Verify all functional and non-functional requirements are implemented correctly.

Ensure the system's performance, security, usability, and reliability meet the expected standards.

Identify and report software defects to facilitate their timely resolution.

3. Test Scope:

Functional Testing: The scope includes testing all major functionalities of the system, including user registration, profile management, networking, event management, knowledge sharing sessions, job postings, and interactions with teachers and students.

Non-Functional Testing: The scope includes testing performance, usability, security, compatibility, accessibility, and reliability aspects of the system.

Integration Testing: The scope includes testing the integration of different system components and modules.

User Acceptance Testing (UAT): The scope includes involving stakeholders to validate the system against their requirements and expectations.

4. Test Approach:

Testing Levels: The testing will include unit testing, integration testing, system testing, and user acceptance testing.

Testing Types: The testing will cover functional, non-functional, and regression testing.

Test Data: Realistic and representative test data will be used to cover various use cases and scenarios.

5. Test Environment: Testing Environment: A dedicated testing environment will be set up to mimic the production environment without affecting real user data. Devices and Browsers: The system will be tested on different devices, browsers, and operating systems.

6. Test Cases and Scenarios: Test cases will be designed to validate each functional and non-functional requirement. Test scenarios will cover positive and negative test cases to ensure comprehensive testing.
7. Defect Management: Defect Reporting: Defects will be logged with appropriate details, including steps to reproduce and severity. Defect Tracking: The status and resolution progress of defects will be tracked throughout the testing process.
8. Test Schedule: Test Execution: The testing activities and schedule will be outlined, considering the development timeline and milestones. Test Completion: Testing will be considered complete when all planned test cases have been executed, and critical defects have been resolved.
9. Risks and Mitigation: Potential risks to the testing process and their mitigation strategies will be identified and documented.
10. Test Deliverables: Test Plan: This document outlining the testing approach and strategy. Test Cases: Detailed test cases and test scenarios. Test Reports: Reports summarizing the test results, including defects and progress.
11. Approval: The test plan will be reviewed and approved by relevant stakeholders before commencing testing.

This test plan outlines the approach and strategy to ensure comprehensive testing of the "Alum Hub" system. It aims to validate its quality, reliability, and functionality, providing a stable and user-friendly platform for alumni networking and engagement.

5.2 Unit Testing

It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2.1 Purpose of Unit Testing:

The main purposes of unit testing are as follows:

1. Defect Identification: Unit testing helps identify defects and errors at an early stage of development when they are easier and less costly to fix. By isolating individual units, developers can quickly identify and rectify issues within specific code segments.
2. Isolation of Units: Unit testing allows developers to test each unit in isolation, regardless of the rest of the system's complexity. This isolation helps in identifying the cause of defects and reduces the complexity of debugging.
3. Validation of Requirements: Unit testing validates that each unit of code functions as intended and fulfills the specified requirements. It ensures that the units perform their tasks correctly before they are integrated into the system.
4. Regression Testing: Unit tests act as a form of regression testing, ensuring that existing functionalities are not affected by new changes or enhancements. If a new change introduces a regression, the unit test will catch it early in the development process.
5. Reduction of Integration Issues: When individual units have been thoroughly tested, integration issues are less likely to arise during the integration testing phase. This contributes to smoother integration and reduces the effort needed for debugging and defect resolution.
6. Enhancing Code Quality: Unit testing encourages developers to write modular, self-contained, and testable code. This leads to the creation of high-quality, maintainable, and robust software.

5.2.2 Integration Testing:

Integration testing is the next level of software testing where multiple units/modules of a system are combined and tested as a group to validate their interactions and interfaces. The purpose of integration testing is to identify any defects that may arise due to the integration of different components and to ensure that the system components work together seamlessly. Integration testing helps identify issues related to data flow, communication, and compatibility between units, enabling the identification and resolution of integration-related defects before moving on to system testing.

5.2.3 Purpose of Integration Testing:

The main purposes of integration testing are as follows:

1. Verification of Component Interaction: Integration testing verifies the interaction and communication between individual units/modules of the system. It ensures that data and control flow between components work as expected.
2. Detection of Integration Defects: Integration testing helps identify defects and issues that may arise due to the interaction of different units. It helps detect problems such as data inconsistency, interface mismatches, and communication failures.
3. Testing of Interfaces: Integration testing validates the interfaces between units to ensure that they correctly exchange information and function as intended.
4. Validation of System Architecture: Integration testing validates the overall system architecture by examining how different components interact and work together.
5. Risk Mitigation: Integration testing mitigates the risk of integration-related defects, which can be challenging to identify and fix during later stages of development.
6. Confirmation of Interoperability: Integration testing confirms that the integrated units work together cohesively and that the system functions as a whole.

Unit testing and integration testing together form the foundation of a robust testing process, ensuring that individual components work correctly and seamlessly integrate to deliver a reliable and fully functional software system.

5.2.4 Purpose:

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

Software Integration Testing is performed according to the Software Development Life Cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatic complexity of the software and software architecture, reusability of modules and life-cycle / versioning management.

Some different types of integration testing are big bang, top-down, and bottom-up, mixed (sandwich) and risky - hardest. Other Integration Patterns are: Collaboration Integration, Backbone Integration, Layer Integration, Client/Server Integration, Distributed Services Integration and High-frequency Integration.

5.3 Acceptance Test Plan:

Test Strategy: Describe the overall approach to acceptance testing. Discuss the types of testing to be performed, such as functional testing, usability testing, performance testing, and security testing. Specify the test techniques, tools, and methodologies that will be utilized.

Test Environment: Define the hardware, software, and network configurations required for executing the acceptance tests. Specify the versions of Java, operating systems, browsers, and any other dependencies necessary for running the system under test.

Test Scenarios: Identify the specific scenarios or use cases that will be tested. These scenarios should cover the main functionality and key features of the software system. Each scenario should be described concisely and should include the expected outcomes.

Test Data: Specify the test data that will be used during the acceptance testing phase. This includes sample inputs, expected outputs, and any necessary data configurations or setups. If there are any specific data generation or manipulation procedures, they should be documented as well.

Test Execution: Explain how the acceptance tests will be executed. This includes the steps to be followed, the expected results, and any specific testing techniques to be employed. Provide guidance on how to log defects or issues encountered during testing.

Test Schedule: Define the timeline for the acceptance testing phase, including start and end dates, as well as any milestones or checkpoints. If there are any dependencies on other testing activities or deliverables, they should be mentioned.

Test Reporting: Describe the format and contents of the test reports that will be generated during acceptance testing. Specify the metrics, such as test coverage or defect density, that will be tracked and reported. Outline the process for communicating test results and any associated risks to the relevant stakeholders.

5.4 Test Cases:

Test cases are detailed sets of conditions and steps used to verify the correctness and functionality of a software application or system. They serve as documented instructions for testers to execute during the testing process. Test cases are designed to cover various scenarios, including both positive and negative inputs, to ensure that the software behaves as expected under different conditions. They form an essential part of the software testing process and aid in identifying defects and ensuring the quality of the software product.

Test Cases :

Test case_ID	Scenario	Test summary	test step	Expected Result	Result
Tc_1	Admin Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter valid password 4.click on login button	Successfully login	Pass
Tc_2	Admin Login validation	To verify login functionality	1.enter application url 2.enter invalid username 3.enter valid password 4.click on login button	Popup message shows an error “invalid credentials”	Pass
Tc_3	Admin Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter invalid password 4.click on login button	Popup message shows an error “invalid credentials”	Pass
Tc_4	Alumni Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter valid password 4.click on login button	Successfully login	Pass
Tc_5	Alumni Login validation	To verify login functionality	1.enter application url 2.enter invalid username 3.enter valid password 4.click on login button	Popup message shows an error “invalid credentials”	Pass
Tc_6	Alumni Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter invalid password 4.click on login button	Popup message shows an error “invalid credentials”	Pass

Tc_7	Student Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter valid password 4.click on login button	Successfully login	Pass
Tc_8	Alumni Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter valid password 4.click on login button	Successfully login	Pass
Tc_9	Teacher Login validation	To verify login functionality	1.enter application url 2.enter valid username 3.enter valid password 4.click on login button	Successfully login	Pass
Tc_10	Update Profile	To verify user Authentication	1.enter the user id 2.enter the Job profile 3.click on submit	Successfully entered	Pass
Tc_11	Reset	To verify reset password functionality	1.enter the new password 2.re-enter the password 3.click on submit	Successfully entered	Pass
Tc_12	Add Session	To verify Link	1.enter the new link 3.click on submit	Successfully submit	Pass
Tc_13	New user registration	Enter the valid register detail	1. Enter data in all field 2.click on register	Successfully submit	Pass
Tc_14	New user registration	Enter the valid email id detail	1.Enter data in all filled. 2.click on register	Successfully submit	Pass
Tc_15	User Logout	To verify logout functionality	1.enter the application URL 2.click on logout	Logout successfully	Pass

Chapter: 6 Limitations of Proposed System

The limitations of the proposed "Alum Hub" system based on different factors and functionality:

1. User Adoption and Engagement:

- Limitation: The success of the system heavily depends on user adoption and active engagement. If a significant number of alumni do not register or participate, the platform's effectiveness may be limited.
- Mitigation: Implement user-friendly interfaces, provide incentives for participation, and conduct outreach campaigns to encourage user adoption.

2. Data Accuracy and Completeness:

- Limitation: The accuracy and completeness of user-provided data can impact search results and networking effectiveness.
- Mitigation: Implement data validation mechanisms during registration and profile updates, and encourage users to keep their information up-to-date.

3. Security and Privacy Risks:

- Limitation: The system may be vulnerable to security breaches or unauthorized access, compromising user data and trust.
- Mitigation: Implement robust security measures, encryption, and access controls to safeguard user information. Regular security audits and updates are essential.

4. Performance and Scalability:

- Limitation: As the user base grows, the system's performance may degrade, leading to slow response times and reduced user experience.
- Mitigation: Conduct load testing, optimize database queries, and use caching techniques to improve performance and scalability.

5. Compatibility Issues:

- Limitation: The system may face compatibility issues with different devices, browsers, or operating systems.
- Mitigation: Conduct thorough cross-browser and cross-device testing to ensure the platform works seamlessly across various environments.

6. Functionality Limitations:

- Limitation: The system may lack certain advanced features or integrations that users expect from modern alumni platforms.
- Mitigation: Prioritize feature development based on user feedback and focus on core functionalities initially, gradually expanding the system's capabilities.

7. Maintenance and Support:

- Limitation: The system will require ongoing maintenance, updates, and technical support, which could strain available resources.
- Mitigation: Allocate resources for continuous maintenance, establish a support system for user inquiries, and plan for future system enhancements.

8. Integration Challenges:

- Limitation: Integrating the "Alum Hub" system with existing databases or external systems might be complex and time-consuming.
- Mitigation: Plan integration efforts in advance, involve skilled developers, and conduct thorough testing during integration phases.

9. Limited User Engagement:

- Limitation: Encouraging alumni to actively participate in knowledge sharing sessions and job postings might be challenging.
- Mitigation: Organize engaging events, promote alumni achievements, and provide opportunities for meaningful networking to boost user engagement.

10. Budget and Resource Constraints:

- Limitation: Limited financial resources and development expertise could impact the system's scope and functionality.
- Mitigation: Prioritize essential functionalities, explore open-source solutions, and consider collaborations to overcome resource limitations.

By recognizing these limitations and implementing appropriate mitigation strategies, the "Alum Hub" system can be designed and managed more effectively, enhancing its value and usability for alumni networking and engagement.

Chapter:7. Proposed Enhancement

- Enhanced User Profiles: Allow alumni to showcase their achievements, projects, and skills in their profiles, making them more appealing to potential employers or collaborators.
- Social Media Integration: Integrate the system with popular social media platforms, enabling seamless sharing of alumni events and activities on their social media profiles.
- Alumni Chapters and Groups: Create dedicated groups or chapters based on location, industry, or interest, facilitating focused networking and knowledge exchange among alumni.
- Alumni Mentorship Program: Implement a mentorship program connecting experienced alumni with current students or recent graduates for career guidance and support.
- Personalized Newsfeed: Develop a personalized newsfeed for each user, displaying relevant updates, event recommendations, and alumni achievements.
- Alumni Achievements Showcase: Create a section to showcase notable achievements of alumni, inspiring others and fostering a sense of pride in the alma mater.
- Virtual Events and Webinars: Organize virtual events, webinars, and workshops, enabling broader participation and reaching alumni across the globe.
- Alumni Surveys and Feedback: Conduct periodic surveys to gather feedback from alumni, helping improve the platform's features and user experience.
- Alumni Fundraising Initiatives: Enable alumni to support their alma mater through fundraising campaigns for scholarships, infrastructure, or special projects.
- Enhanced Job Board: Partner with companies to offer exclusive job opportunities to alumni, enhancing their career prospects.
- Alumni Expert Directory: Create a searchable directory of alumni with expertise in various fields, encouraging collaboration on professional projects.
- Alumni E-commerce Store: Develop an e-commerce store to sell branded merchandise and memorabilia, generating revenue for alumni events and activities.
- Alumni Recognition and Awards: Establish an awards program to recognize outstanding alumni achievements and contributions to society.
- Alumni Hall of Fame: Implement a virtual hall of fame to honor distinguished alumni, celebrating their accomplishments.
- Machine Learning-Based Recommendations: Utilize machine learning algorithms to provide personalized recommendations for events, connections, and opportunities.
- Alumni Podcasts and Web Series: Launch podcasts and web series featuring alumni success stories, inspiring and engaging the alumni community.
- Virtual Alumni Reunions: Organize virtual reunions, allowing alumni to connect and reminisce, regardless of geographical limitations.
- Alumni Research Collaboration: Encourage research collaborations among alumni and faculty, promoting knowledge sharing and innovation.
- Multi-language Support: Provide support for multiple languages, ensuring inclusivity

- and accessibility for alumni from diverse backgrounds.
- Alumni Impact Reports: Share annual impact reports showcasing how alumni contributions have made a difference in various areas, fostering a sense of pride and involvement.

These proposed enhancements aim to enrich the Alumni Hub system, elevating the alumni experience, and creating a thriving, engaged, and supportive alumni community. By incorporating these features, the platform can effectively address the challenges of connecting alumni and strengthening their bond with the institution.

Chapter: 8.Conclusion

The "Alumni Hub" project aims to address the challenges faced by alumni in staying connected with their alma mater and each other. It is a Java-based web application that provides a valuable platform for former students to network, share information, and stay engaged with their educational institution. The system offers a range of features, including modules for Admin, Alumni, Teacher, and Student, ensuring personalized experiences for different user roles.

Throughout the project's development, we have emphasized scalability and customizability, allowing educational institutions to tailor the platform according to their specific needs. By leveraging Java Server Pages (JSP) and Java Servlets, the system offers a robust and reliable foundation, ensuring seamless user experiences.

The primary focus of the "Alumni Hub" System is to foster collaboration and support among alumni, creating a vibrant community that contributes to the growth of the institution. Through personalized user profiles, social media integration, and dedicated groups or chapters, the platform encourages active engagement and networking.

In addition to connecting alumni, the platform also supports the career growth of current students through mentorship programs and job postings shared by alumni. By providing opportunities for knowledge exchange and collaboration between students, teachers, and former students, the system contributes to a thriving educational ecosystem.

While the "Alumni Hub" System presents a valuable solution for alumni engagement, it does come with some limitations. Challenges related to user adoption, data accuracy, security, and budget constraints need to be addressed through proposed enhancements, such as machine learning-based recommendations, personalized newsfeeds, and virtual alumni events. These enhancements aim to enrich the platform and improve user engagement.

In conclusion, the "Alumni Hub" System represents a powerful tool for educational institutions to maintain a strong connection with their alumni community. By continuously refining and expanding the platform based on user feedback and evolving needs, the system can create a lasting impact on the career growth of current students and foster a sense of pride and involvement among alumni. As we move forward with the implementation and deployment of the "Alumni Hub" System, we remain committed to delivering a high-quality, user-centric platform that celebrates alumni achievements and nurtures a legacy of excellence within the educational institution's community.

Chapter:9. Bibliography

- [1] <https://www.tutorialspoint.com>
- [2] <https://www.guru99.com/>
- [3] <https://www.w3schools.com/java/>
- [4] <https://www.javatpoint.com/java-tutorial>
- [5] <https://beginnersbook.com/java-tutorial-for-beginners-with-examples/>
- [6] <https://www.mysqltutorial.org/>
- [7] <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html>
- [8] <https://www.w3resource.com/mysql/mysql-tutorials.php>
- [9] <https://help.shopify.com/en/manual/products/searching-filtering>
- [10] <https://neil Patel.com/blog/ecommerce-website-search/>

Chapter:10. Appendix

Appendix A: Cost Sheet

The cost sheet provides an overview of the estimated expenses for developing and maintaining the Alumni Hub System. Please note that the costs mentioned below are approximate and may vary depending on specific project requirements and market rates.

Item	Description	Cost
Development Cost	Software development and coding	₹ 15,00,000
Design and UI/UX	User interface and experience design	₹ 6,00,000
Database Setup and Hosting	Database configuration and hosting	₹ 2,25,000
Testing and Quality Assurance	Testing the system for bugs	₹ 3,75,000
Documentation and User Manual	Creating system documentation	₹ 1,50,000
Project Management	Coordination and management	₹ 3,00,000
Support and Maintenance	Post-development support	₹ 4,50,000/yr
Miscellaneous and Contingency	Unforeseen expenses	₹ 1,50,000
Total		₹ 37,50,000

Data Sheet:

A Data Sheet typically contains structured information about a particular entity or object. In the context of "Alumn Connect," it could include details about alumni, such as their names, contact information, graduation year, and other relevant data.

Chapter 11: User Manual

1.1 Purpose : Welcome to the Alumni Hub System! This user manual provides guidance on using the platform to stay connected with your alma mater, interact with other alumni, and engage with the institution's activities. The system aims to foster collaboration and support among alumni while offering career growth opportunities for current students.

1.2 System Overview: The Alumni Hub System is a Java-based web application that provides a platform for alumni to stay connected with their alma mater and contribute to its growth. The system offers a range of features for different user roles, including alumni, teachers, current students, and administrators. It enables seamless networking, knowledge exchange, event organization, and job posting opportunities.

2. Getting Started:

2.1 Accessing the System : To use the Alumni Hub, open your web browser and enter the system's URL provided by the institution. You will be directed to the login page.

2.2 User Registration : If you are a new user, click on the "Register" button to create your account. Fill in the required details, such as name, email, and role, and submit the registration form.

3. User Roles

3.1 Alumni :

- Update Profile: Access and edit your personal and professional information.
- Alumni Directory: Explore the directory to find and connect with other alumni based on location, industry, or profession.
- Message Alumni: Send direct messages to fellow alumni through the system.
- View Events: Access and view a list of upcoming events organized by the institution.
- View Job Posts: Explore job postings shared by other alumni.
- View Sessions: Enroll for sessions or webinars and leave joined sessions.
- Chat Box: Interact with teachers, current students, and public chat groups.
- Gallery: Access and view photos in the system's gallery.

3.2 Teacher :

- Update Profile: Access and edit your personal and professional details.
- Message Alumni: Send direct messages to alumni through the system.
- Message Students: Send direct messages to students through the system.
- View Events: Access and view a list of upcoming events organized by the institution.
- View Job Posts: Explore job postings shared by alumni.
- View Sessions: View scheduled sessions and arrange sessions for students.
- Edit Session: Modify session details such as date, time, and description.
- Delete Session: Remove scheduled sessions from the system.
- View Participants: Access the list of participants in a session.
- Chat Box: Interact with alumni, current students, and public chat groups.

- Gallery: Access and view photos in the system's gallery.

3.3 Current Student :

- Update Profile: Access and edit your personal and academic details.
- Message Alumni: Send direct messages to alumni through the system.
- View Events: Access and view a list of upcoming events organized by the institution.
- View Job Posts: Explore job postings shared by alumni.
- View Sessions: Enroll for sessions or webinars and leave joined sessions.
- Chat Box: Interact with alumni, teachers, and public chat groups.
- Gallery: Access and view photos in the system's gallery.

3.4 Admin :

- Add Student: Add new students to the student database.
- Add Teacher: Add new teachers to the teacher database.
- View Teacher Profile: Access and view teacher profiles and their details.
- View Student Profile: Access and view student profiles and their details.
- View Alumni: View a list of all alumni registered in the system.
- Update Footer Content: Modify and update footer content on the system's pages.
- Add Event: Create and add new events to the platform.
- View Event: Access and view a list of all events organized by the institution.
- Add Photos: Upload and add photos to the system's gallery.
- View Photos: Access and view all the photos available in the system's gallery.
- Edit Event: Edit event details such as date, time, and description.
- Add Session: Create and schedule sessions or webinars for users.
- View Session: Access and view all scheduled sessions and webinars.
- Delete Session: Remove scheduled sessions from the system.
- Delete Event: Delete events from the system.
- Delete Photos: Remove photos from the gallery.
- Delete Student: Delete student profiles from the system.
- Delete Teacher: Delete teacher profiles from the system.
- Add/Delete FAQs: Add or delete frequently asked questions from the system.
- View Contact Us Requests: Access and view messages received via the "Contact Us" form.
- Send Newsletter: Create and send newsletters to all registered users, including alumni, teachers, and students.
- Generate Reports: Generate reports in various formats like CSV, Excel, PDF, or direct print.

4. Dashboard

4.1 Navigation and Features :After logging in, you will be directed to your dashboard. Here, you can access various features like profile, events, job postings, sessions, newsletters, and more.

5. Profile Management

5.1 Update Profile :Click on the "Edit Profile" option to update your personal and professional information.

5.2 Alumni Directory :Explore the alumni directory to find and connect with other alumni based on location, industry, or profession.

6. Events and Job Postings

6.1 Post Event/Job :Share upcoming events or job opportunities within your professional network.

6.2 View Events/Job :Access a list of all events and job postings shared by other alumni and the institution.

7. Organize Sessions

7.1 Create Session :Organize sessions, workshops, or webinars on specific topics of interest and invite other alumni, students, or teachers to participate.

7.2 Track Participation: Keep track of the students who have joined your sessions.

8. Newsletters

8.1 Send Newsletter :Administrators can create and distribute newsletters containing updates on institutional activities, alumni achievements, and upcoming events to all registered users.

9. Networking

9.1 Contact Teachers/Students :Connect with teachers or current students for mentorship, knowledge exchange, or appreciation.

10. Support and Help

10.1 FAQs :Access frequently asked questions to find answers to common queries.

10.2 Contact Support :If you encounter any issues or have questions, click on the "Help" section to access FAQs and contact support.

11. Data Privacy and Security : The Alumni Hub system prioritizes data privacy and security.Your personal information is securely stored and used only for system-related purposes.

12. Contactus : We value your feedback! Help us improve the system by providing your valuable suggestions and comments.

Thank you for using the Alumni Hub System! We hope this user manual assists you in making the most of the platform and staying connected with your alma mater and fellow alumni.