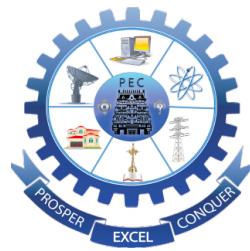


## ALUMNI VIRTUAL OFFICE



### A PROJECT REPORT

*Submitted by*

**VARUN KRISHNAN V (19104117)**

**SURIYASANKAR P (19104108)**

**NANDHAKUMAR M (19104069)**

*in partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**PAAVAI ENGINEERING COLLEGE**

**(AUTONOMOUS)**

**DECEMBER 2022**

## **BONA FIDE CERTIFICATE**

Certified that this project report “**ALUMNI VIRTUAL OFFICE**” is the bona fide work of **VARUN KRISHNAN V (19104117)**, **SURIYASANKAR P (19104108)**, **NANDHAKUMAR M (19104069)** who have carried out the project work under my supervision.

### **SIGNATURE**

**Dr.D.BANUMATHY, M.E., Ph.D.,**

### **HEAD OF THE DEPARTMENT**

Department of Computer Science and  
Engineering

Paavai Engineering College,

Namakkal – 637018.

### **SIGNATURE**

**Mrs.P.RENUKADEVI, M.E., (Ph.D.),**

### **SUPERVISOR – INTERNAL**

Department of Computer Science and  
Engineering

Paavai Engineering College,

Namakkal – 637018.

Submitted for End Semester Examinations held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **DECLARATION**

We **VARUN KRISHNAN V (19104117)**, **SURIYASANKAR P (19104108)**, **NANDHAKUMAR M (19104069)** hereby declare that the project report titled **ALUMNI VIRTUAL OFFICE** done by me under the guidance of **Mrs.P.RENUKADEVI,M.E.,(Ph.D.)**, (Internal Supervisor) at **PAAVAI ENGINEERING COLLEGE** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering. Certified further that, to the best of my knowledge, the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## **ACKNOWLEDGEMENTS**

A great deal of arduous work and effort has been spent in implementing this project work. Several special people have guided us and have contributed significantly to this work and so this becomes obligatory to record our thanks to them.

We express my profound gratitude to our honourable Chairman, **Shri. CA.N.V.NATARAJAN, B.Com., F.C.A.,** and also to our Correspondent, **Smt. MANGAI NATARAJAN, M.Sc.,** for providing all necessary facilities for the successful completion of my project.

We wish to express my sincere thanks to our respected **Director-Administration, Dr.K.K.RAMASAMY, M.E., Ph.D.,** for all the blessing and help provided during the period of project work.

We would like to thank our respected **Principal, Dr.M.PREM KUMAR, M.E., Ph.D.,** for allowing us to do this project and providing required time to complete the same.

We extend our sincere thanks to **Dr.D.BANUMATHY, M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering for being the Project Coordinator and for the able guidance and useful suggestions, which helped us for completing project work in time.

We would like to extend my sincere thanks to **Mrs.P.RENUKADEVI,M.E.,(Ph.D.),** Associate Professor, Department of Computer Science and Engineering for being a Supervisor and for giving me an opportunity to do this project and also for her inspiring guidance, generous help and support.

We would like to extend our sincere thanks to **all our department staff members and our parents** for their advice and encouragement to do the project work with full interest and enthusiasm.

## **ABSTRACT**

Alumni Virtual Office is an online virtual space for passed out Students in our College. The aim of this project is to create a platform where alumni can discuss on topics, and can also able to solve their problems via discussing with other alumni of our college. Besides this one can also able to let the college management know that they are interested to contribute to college through conducting webinars, workshops, guest lectures on the topics they are familiar with and can also help the students in placements. We planned to make this forum a centralized one through admin approval from registering to commenting on discussions. This approval system helps to keep the virtual space disciplined through which we can able to bring value to the alumni questions. The alumni management system which exists already are not centralized as there is no approval system on other modules except the registration part. All these things helps alumni to be in contact with each other and help out each other through which we can able to keep the Paavaians connected with each other as a Paavai Family. We have planned to design and develop this application as we seen many students from other colleges and our colleges felt that there is a detachment from their batchmates and alumni after some time after the completion of college. We all also look for someone who is trustable to get our doubts cleared. So we think Alumni Virtual Office going to be a perfect solution for this.

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	i
	<b>LIST OF FIGURES</b>	ii
	<b>LIST OF ABBREVIATIONS</b>	iii
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Project Overview	1
	1.2 Project Objectives	2
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>3</b>
<b>3</b>	<b>SYSTEM ANALYSIS</b>	<b>7</b>
	3.1 Existing System	7
	3.2 Disadvantages	7
	3.3 Proposed System	8
	3.4 Advantages	8
<b>4</b>	<b>SYSTEM ENVIRONMENT</b>	<b>9</b>
	4.1 Hardware Specification	9
	4.2 Software Specification	9
	4.3 Project Description	9
	4.3.1 System Model	10
	4.3.2 NodeJS and MongoDB	12

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>5</b>	<b>SYSTEM DESIGN</b>	<b>14</b>
	5.1 System Architecture	14
	5.2 Use Case Diagram	16
	5.3 E-R Diagram	17
	5.4 Class Diagram	18
<b>6</b>	<b>MODULES DESCRIPTION</b>	<b>19</b>
	6.1 User Authentication	19
	6.2 All users	19
	6.3 Discussions forum	19
	6.4 Contributions	19
<b>7</b>	<b>SYSTEM TESTING</b>	<b>20</b>
	7.1 System Testing	20
	7.2 Types of Testing	21
	7.2.1 Functional Testing	21
	7.2.2 Integration Testing	21
	7.2.3 White Box Testing	21
<b>8</b>	<b>CONCLUSION</b>	<b>22</b>
<b>9</b>	<b>APPENDIX</b>	<b>23</b>
	9.1 Source Code	23
	9.2 Screenshots	39
<b>10</b>	<b>REFERENCES</b>	<b>46</b>

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
5.1	Authentication architecture	14
5.2	Dashboard architecture	15
5.3	Use Case Diagram	16
5.4	E-R Diagram	17
5.5	Class Diagram	18
9.1	Alumni registration form	39
9.2	Alumni login form	40
9.3	My profile (In-approval)	40
9.4	My profile (Active)	41
9.5	Alumni dashboard	42
9.6	Creating discussion	42
9.7	Discussion forum	43
9.8	Comments section before approval	43
9.9	Comments section after approval	44
9.10	All users	44
9.11	Alumni profile	45
9.12	Notifications page	45

## **LIST OF ABBREVIATIONS**

<b>ACRONYMS</b>	<b>ABBREVIATION</b>
AVO	Alumni Virtual Office
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JS	Javascript
MERN	MongoDB, ExpressJs, ReactJs, NodeJs
URL	Uniform Resource Locator
DB	Database
HTTPS	Hypertext Transform Protocol Secure
API	Application Programming Interface
JWT	JSON Web Tokens
MVC	Model View Controller
JSON	Javascript Object Notation

# CHAPTER 1

## INTRODUCTION

### 1.1 PROJECT OVERVIEW

The greatest asset any institution can have is the Alumni Virtual Office. Alumni are the people who represent the institution in the real world. Alumni Virtual Office is created for the students that have graduated from the institution. This is an online website that allows former students to take advantage of the benefits and services that an institution offers after graduation. The alumni network is becoming important in the development of the institution because of their vast potential that benefits both the institution and the students. There are many benefits for being an alumni member of a college or institution . Some of these benefits are keeping a person informed on the events that are organized by the institution, and when some important events are going to be held in the institution. Another benefit is that the information concerning a former student can easily be received and other members of the alumni community can be located without much stress. The student and alumni can communicate with each other.

HTML, CSS, JS, ReactJS, ExpressJS, NodeJS, MongoDB, TailwindCSS, Material UI were used to create the Alumni Virtual Office. This project contains an alumni database, making it easy to keep track of alumni records. There are numerous records, including alumni details such as name, mobile number, email, address, passing year, married status, and social media accounts (LinkedIn, Twitter, Facebook).Because this is a fully functional internet website, there are two technologies involved: frontend and backend. The routes were protected using JSON Web token to avoid the unauthorized user accessing the portal. For more security the passwords are also hashed while storing the database and Redux is used to make the state management for the entire application. The Git and Github also used to keep track on the project versions.

## **1.2 PROJECT OBJECTIVES**

The main objective of Alumni Virtual Office is to help the management manage their alumni and to bridge the gap between the management and alumni. Besides this the discussion forum in Alumni Virtual Office help the passed-out student to clear their doubts with other alumni. It also helps the alumni to let the management know that they are interested to contribute to college through being a part in alumni guest lectures, conducting workshops and helping out the juniors in placements.

## **CHAPTER 2**

### **LITERATURE SURVEY**

**Title: CAs Based Student-Alumni Management System**

**Authors: D.V. Nishanth; Satish; Niteesh S. Narasimha**

Students and alumni management system of any university is a collaborative framework which integrates the knowledge between the students who are pursuing education and the students who have already successfully established themselves after graduation. There are various profession choices that are available wherein a person could choose from and this proposed framework is a predictive system which tries to predict the career option based on the success rate of the student in terms of his credentials CGPA, number of projects, number of publication, and so on and the credentials of the alumni like number of years of experience, proprietorship, number of recognitions, designation reached, and many more. The proposed system tries to collaborate students with alumni or the ones who have graduated out of the university and together they are provided an option to choose their career based on all the experiences they have had so far and the necessary credentials they own . Cognitive-Agents (CAs) which exhibits rational thinking which is applied in the paper to collaborate the students and alumni where in the beliefs over the students and alumni are formulated using the Behaviour-Observation-Belief (BOB) model. The performance achieved by the proposed CAs based framework towards alumni management in university is found to be good with respect to parameters like employment rate, student alumni association, and so on.

**Title: Centralized Alumni Management System (CAMS) - A Prototype Proposal**

**Authors: Aritra Mukherjee, Adrita Roy, Manish Kumar**

This paper proposes a centralized system for alumni management which is institution independent and concentrates on alumni network pan organizations. The fundamental incentive of the system lies in the mentorship process in the institution as well as organization verticals. The system recognizes as an alumni not only graduates but also those individuals who are presently in the institution/organization, thereby facilitating

network among the professionals and the students of the institutions from where the professional had graduated. Apart from the alumni, the institutes/organizations also enjoy a comprehensive list of benefits. The minimum age for registration for alumni has been mandated to be 15 years. Hence, the system takes into account those institutions also from where a certain individual has passed the 10<sup>th</sup> examinations. Hence, the system defines the list of almamater those institution which serve degrees for 10<sup>th</sup>, 12<sup>th</sup>, graduation, post-graduation, doctoral and working organizations. Literature comparison has been done with an existing proposal and our system exhibits better features.

**Title: Alumni Info-Com Management with Distinct Classification of Data**

**Authors: Sasikumar R, Haritha B, Borshiya Vincy T, Kamali M**

The Alumni Info-Com Management System is able to manage alumni data of a college and provide easy access to the same. Alumni of college stay in touch with their immediate friends and it is hard to stay connected with college mates. Contact between alumni develops business connections and to gain insight in a new field. Current students will be initially given a student login id. Access to the system can help them to seek help in their projects or for placements. This single system can satisfy almost every requirement of the alumni. Usually, alumni associations are organized in colleges, but may also be organized in a place where the alumni can meet each other. Despite the fact that there are many existing systems in colleges to maintain the alumni information, they are manual and more time consuming to current students to reach out their alumni and maintaining the privacy of the alumni. To overcome these issues, we proposed a web application which allows alumni to update their information and students can connect with them and can view the filtered events posted by alumni and admin through support vector machine algorithm.

**Title: Alumni Management System Using Web Technologies**

**Authors: K. Lavanya, Y. Supriya, K. Bhargavi, N. Aswini, G. SwarajyaLakshm**

Alumni portal is providing a common platform for an institute. All advanced education establishments need to make a relationship with their graduated class.

So, the main aim to develop this web-based application is that it going to possible for the former students of the college to keep in touch with each other and that will substitute the manual system of the alumni with an automated one. This allows students to know about each other and their current status. Graduated class and understudy can impart just through the administrator consent. This application will have the option to oversee graduated class information of a school and give simple access to the equivalent. The main aim of the project is to build an interaction between alumni, admin and the students; a system that will be able to manage alumni data of a college and provide easy access to the same. The alumni will also be interested to maintain relations with their institutions. Alumni can communicate to the students regarding job opportunities and the students can share the department technology activities to the alumni. The alumni and the student can communicate only through the admin permission. A system that will be able to manage alumni data of a college and provide easy access to the system. Final year students will be initially given a student login ID. Access to the system can help them in building connections to their projects or for placements. The system will automatically list all Alumni information (name, passing year, company currently working in,) and their status will be transferred from the student module to the alumni module.

**Title: An Innovation System that Can Quickly Responses to the Needs of Students and Alumni**

**Authors: Vittavas Rattanamethawong, Sukree Sinthupinyo, Emeritus Achara Chandrachai**

Interaction between the University and students starts from the student enrols into University as a freshman over the period in which the student lives on campus. Cooperation between University and students has occurred since admission until graduation. When students graduate and then begin their work, the only thing that can make an engagement to alumni is the Alumni Association. Alumni Association will serve as a bridge between the University and alumni. Especially in Thailand, the alumni associations are not much active with enablement of technology as a communication tool for engagement. The problem is lack of communication between alumni and University,

this causes consequence problems. For example, the University gets less sponsored by alumni; alumni do not get what they want from University; and so on. The review of literatures in this study concluded that to motivate the participations among alumni members eight factors are needed, e.g. 1) Alumni Demographic, 2) Communications, 3) Value Creation, 4) Awareness, 5) Motivation, 6) Collaboration, 7) Engagement, and 8) Alumni Satisfaction. This study aims to explore an innovative framework for responding to the needs of students and alumni and the alumni relationship management system to support the framework. Evidence from the survey found that University needs cooperation from students and alumni. In contrast, Students and alumni need information from the University as well. The results of the survey findings will lead to further research to identify the factors that contribute to build the innovation of relationship management system that can quickly responses to the Needs of students and alumni.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

Many colleges maintain present and old students manually. Re collecting this data in the manual system is very difficult. If student needs any data about old students they have to approach college management. Overall collecting the information is very tedious task in this system.

The existing system is built with innumerable excel sheets that are created by each user. These sheets may be collated by an alumni organization and shared with all the alumni but this activity may not be frequent. E – Mail ID's may be freely shared and can lead to excessive spam mails. This apparent lack of privacy will force many alumni to avoid sharing their details with fellow alumni. The system is difficult to maintain on a regular basis by a small group of students.

The Existing system is a computerized system but which is maintained at individual databases i.e in excels sheets, it's a time delay process. And maintaining all the records in Excel sheets is difficult. If they want any record , they have to search all the records. It doesn't provide multiple user accessibility and also doesn't have different user privileges. So the system is not accessible for all the employees of the organization.

The current system is not completely computerized and manual system in entering students and staff data and handling it. There is no centralized database maintenance. There is no easy access to the particular students record. The student cannot easily navigate through the database

#### **3.2 DISADVANTAGES**

- No approval systems
- Lack of discussion forum
- Decentralized system

### **3.3 PROPOSED SYSTEM**

The proposed system will be on-line so it can be accessed by alumni anywhere. It will enable quick and easy communication. Each user will be responsible for the updating their own information. Each user will also have the option to maintain their privacy. It does not require the constant attention of a group of students for its maintenance. Alumni will be able to organize meetings and find out about job opportunities on themselves using this system.

The Proposed system is a computerized system, but which is maintained at centralized databases i.e. in automated forms it's a very fast process. And maintaining all the records in online systems database which makes it very easy to access and retrieve data from the database. If they want any record they can easily search all the records. It provides multiple user accessibility and has different user privileges. So the system is accessible for all the employees of the organization.

- This project allows students to register through online.
- The students can easily access old students data.
- Students can directly communicate with old students regarding Job, or any enquiry.
- And this project has many additional features.

### **3.4 ADVANTAGES:**

- It is completely automated system in handling the college database. This system provides centralized database maintenance. This system provides easy access to the alumni account or his/her complete details. This system provides student to easily navigate through the application for more information in a most secure manner.
- Fast data retrieval.
- Centralized management for making the Virtual Office safe and secure.

## **CHAPTER 4**

### **SYSTEM ENVIRONMENT**

#### **4.1 HARDWARE SPECIFICATION**

- Processor : Intel Core Processor 2.71GHz
- RAM : 2 GB
- Hard disk : 1000 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

#### **4.2 SOFTWARE SPECIFICATION**

- Operating system : Windows OS
- Front End : ReactJS
- Back End : MongoDB, ExpressJS
- Application : Web Application
- IDE : Visual Studio Code

#### **4.3 PROJECT DESCRIPTION**

##### **PROBLEM STATEMENT**

To create a platform for Alumni of our college to interact with their batchmates as well as with other alumni. Through “Alumni Virtual Office” they should be able to clear their query, share some information or they can start a discussion on some topic. Besides this they can also able to help budding Paavaians through sharing their knowledge on specific domain through guest lectures or workshops.

### **4.3.1 SYSTEM MODEL**

#### **FRONT END:**

##### **ReactJS**

The ReactJS is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the ReactJS Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime , an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the ReactJS.

The ReactJS Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the ReactJS Framework and other libraries. The ReactJS Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for ReactJS software called Visual Studio.

##### **Material UI**

Material-UI (MUI) is a CSS framework that provides React components out-of-the-box and follows Google's Material Design launched in 2014. MUI makes it possible to use different components to create a UI for a company's web and mobile apps. Google uses Material Design to guarantee that no matter how users interact with the products they use, they will have a consistent experience. Material Design includes guidelines for typography, grid, space, scale, color, images, etc. And it also allows designers to build deliberate designs with hierarchy, meaning, and a focus on results.

The MUI library for React has over 76k stars on GitHub and is one of the most improved UI libraries. You can build an incredibly stylish application in a short

amount of time with pre-styled components, as well as tune and expand these components according to your needs. It is based on Leaner Style Sheets (LESS), a CSS development extension.

## **Redux**

Redux is simply a store to store the state of the variables in your app. Redux creates a process and procedures to interact with the store so that components will not just update or read the store randomly. Similar to the bank. It does not mean because you have money in the bank that you can go anytime, open the vault, and take money. You have to go through certain steps to withdrawal money.

In the rest of the article, I will show how to create a Redux Hello World to explain how Redux works before adding it to React.

In short, Redux is a way to manage the “state” or we can say it is a cache or storage that can be accessed by all components with a structured way. It has to be accessed through a “Reducer” and “Actions”

## **JWT**

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA. Although JWTs can be encrypted to also provide secrecy between parties, we will focus on signed tokens. Signed tokens can verify the integrity of the claims contained within it, while encrypted tokens hide those claims from other parties. When tokens are signed using public/private key pairs, the signature also certifies that only the party holding the private key is the one that signed it.

### **4.3.2. NodeJS and MongoDB**

#### **NodeJS**

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). It was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official documentation is as follows:

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

#### **MongoDB**

- MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time
- The document model maps to the objects in your application code, making data easy to work with
- Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyse your data
- MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use optionally across more than one file system. Log files are identified with the extension.

#### **Concurrency and locking:**

MongoDB allows multiple clients to use the same database concurrently. As such, it needs to control concurrent access to shared data, to ensure data integrity—when multiple clients update the same data, or clients attempt to read data that is in the process of being changed by another client. MongoDB provides two modes of concurrency control: pessimistic concurrency and optimistic concurrency. When pessimistic concurrency control is being used, MongoDB controls concurrent access by

using locks. Locks can be either shared or exclusive. MongoDB uses them for DMV and other resources that are usually not busy. MongoDB also monitors all worker threads that acquire locks to ensure that they do not end up in deadlocks—in case they do, MongoDB takes remedial measures, which in many cases are to kill one of the threads entangled in a deadlock and roll back the transaction it started. To implement locking, MongoDB contains the Lock Manager. The Lock Manager maintains an in-memory table that manages the database objects and locks, if any, on them along with other metadata about the lock. Access to any shared object is mediated by the lock manager, which either grants access to the resource or blocks it. MongoDB also provides the optimistic concurrency control mechanism, which is similar to the multiversion concurrency control used in other databases. Social database frameworks are the most critical database frameworks utilized as a part of the product business today.

The most critical parts of MongoDB 8 are:

- MongoDB is anything but difficult to utilize.
- MongoDB scales from a portable tablet to symmetric multiprocessor frameworks.
- MongoDB gives information warehousing elements that as of recently have just been accessible in Oracle and other more costly DBMS.

A database framework is a general gathering of distinctive database programming segments and databases containing the parts viz. Database application projects, Front - End segments, Database administration frameworks, and Databases.

A database framework must give the accompanying elements:

- A mixture of client interfaces
- Physical and Logical information autonomy
- Query advancement
- Data honesty
- Concurrency control

# CHAPTER 5

## SYSTEM DESIGN

### 5.1 SYSTEM ARCHITECTURE

The entire system is centralized under the Admin. Initially the user will singup in our portal. The admin will see the list of alumni signup and approve them to let them access the dashboard. Once the alumni get into the dashboard they can able to acces the dashboard and can access all other modules of Alumni Virtual Office.

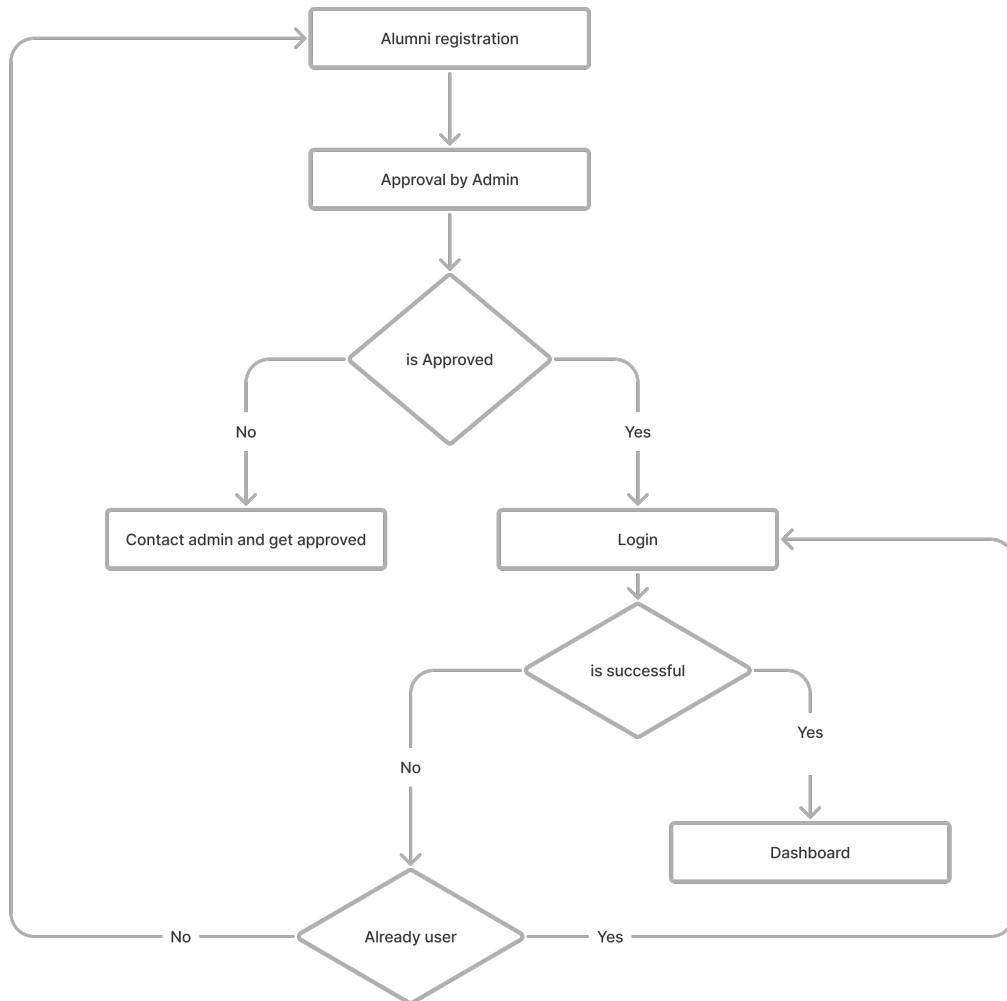


Figure.5.1 Authentication Architecture

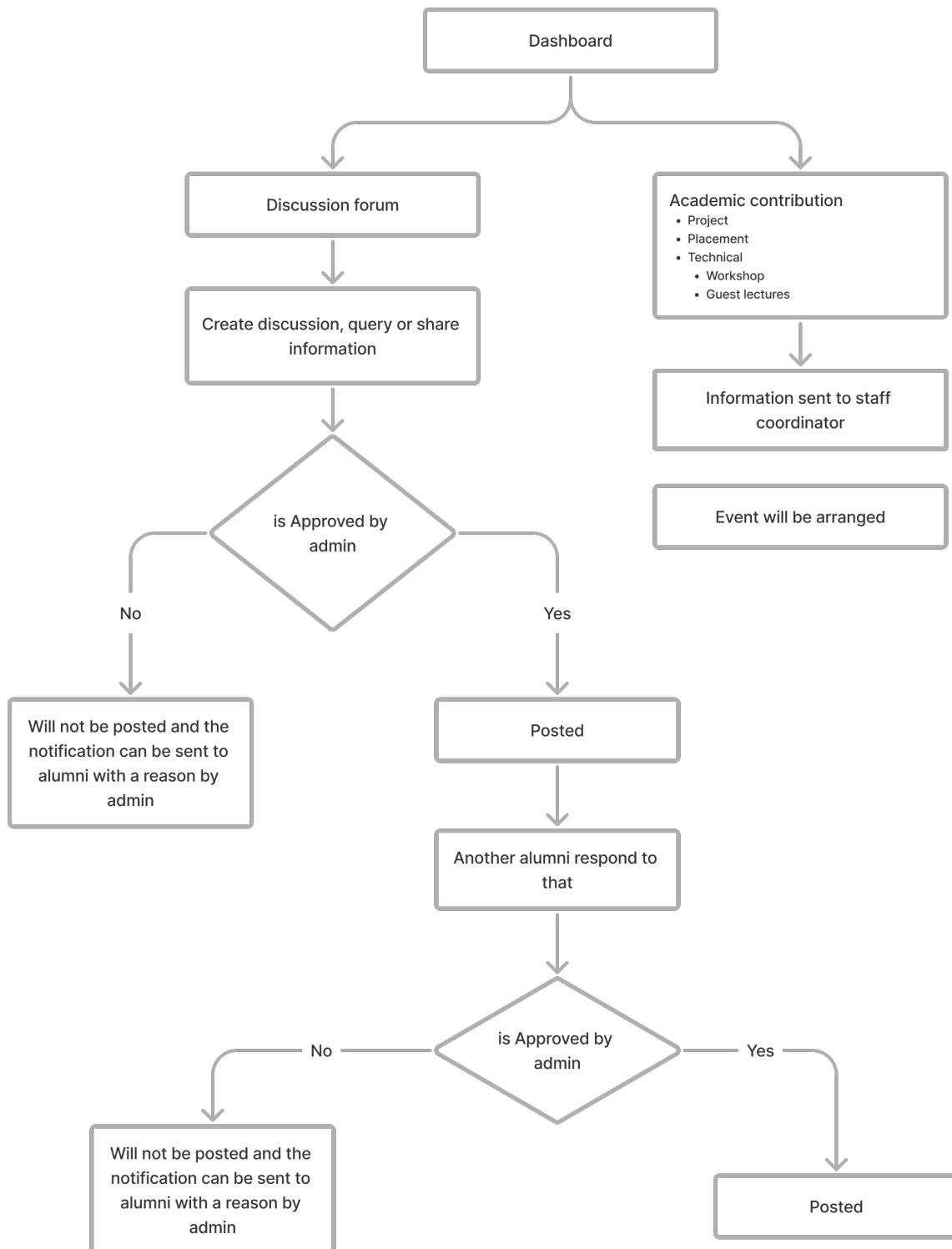


Figure.5.2 Dashboard Architecture

## 5.2 USE CASE DIAGRAM

Use cases share different kinds of relationships. Defining the relationship between two use cases is the decision of the software analysts of the use case diagram. A relationship between two use cases is basically modeling the dependency between the two use cases. The reuse of an existing use case by using different types of relationships reduces the overall effort required in developing a system.

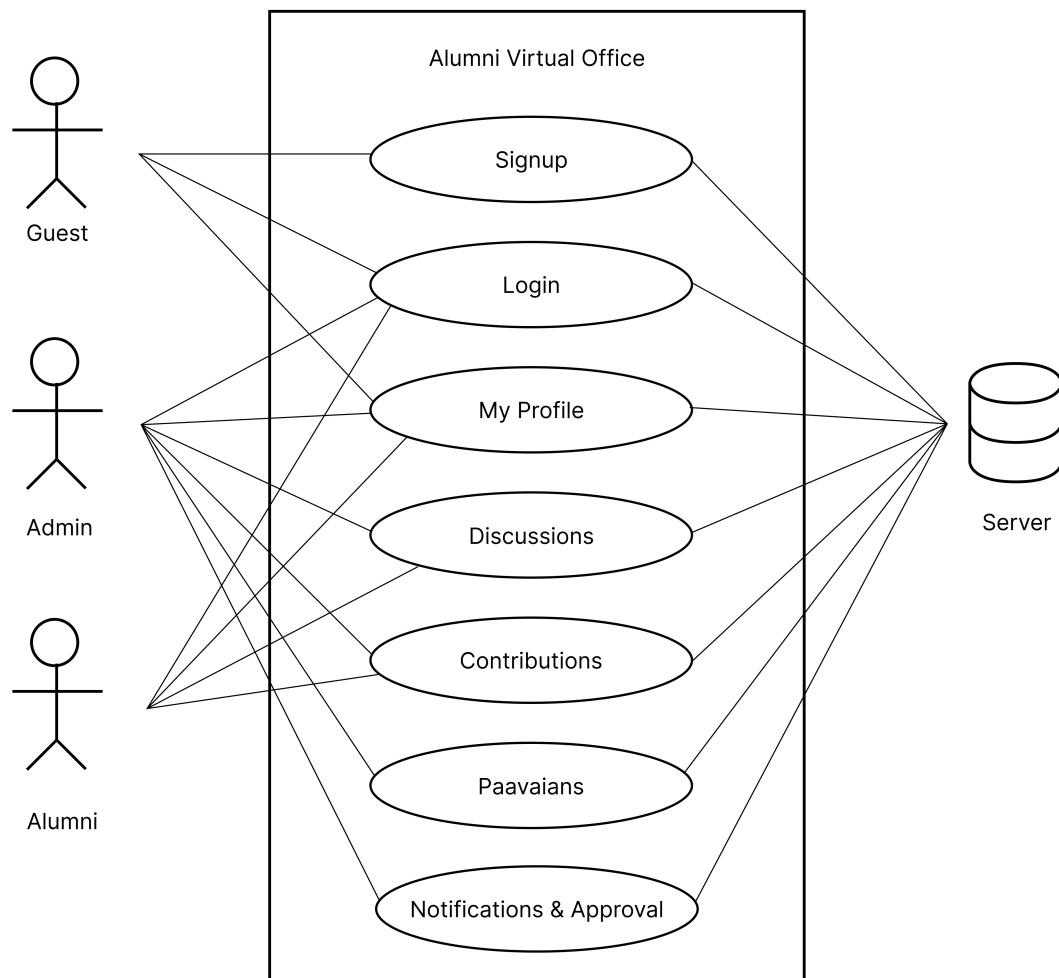


Figure.5.3 Use Case Diagram

### 5.3 E-R DIAGRAM

An Entity Relationship Diagram (ER Diagram) pictorially explains the relationship between entities to be stored in a database. Fundamentally, the ER Diagram is a structural design of the database. It acts as a framework created with specialized symbols for the purpose of defining the relationship between the database entities. ER diagram is created based on three principal components: entities, attributes, and relationships.

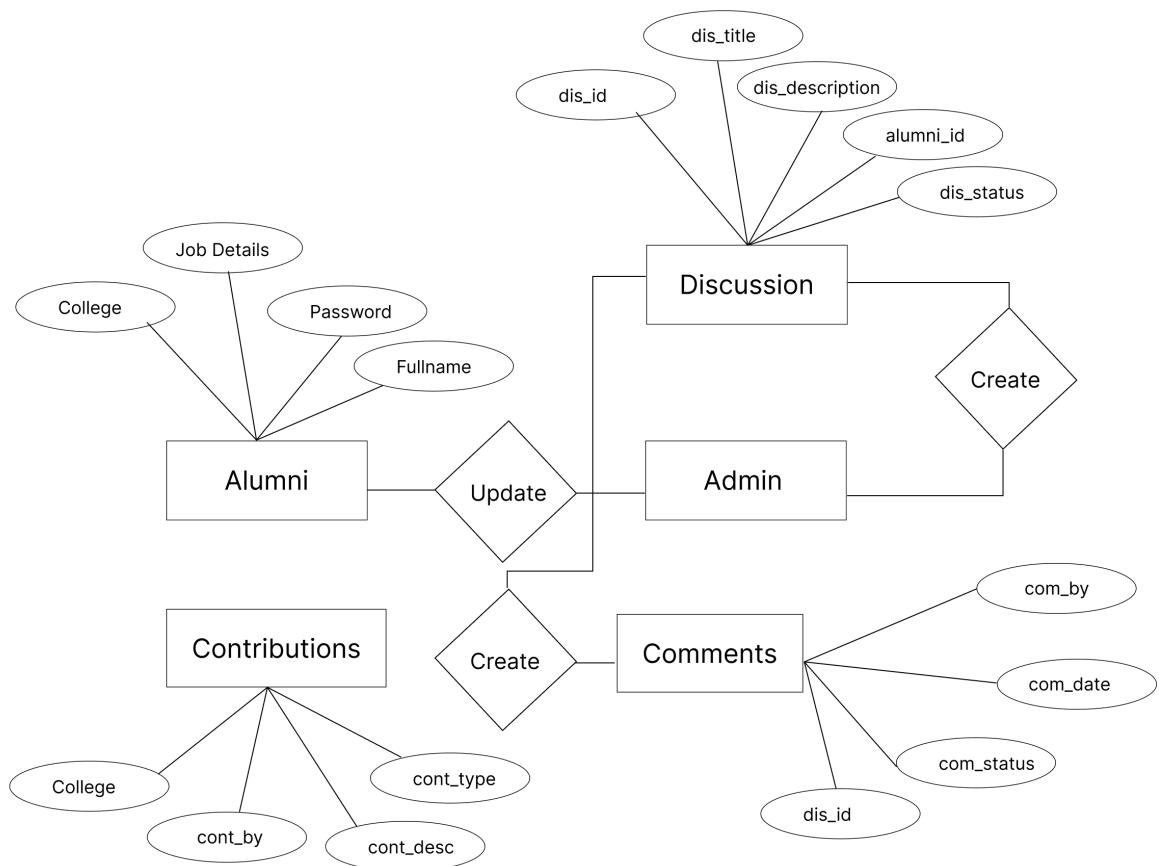


Figure 5.4 E-R Diagram

## 5.4 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

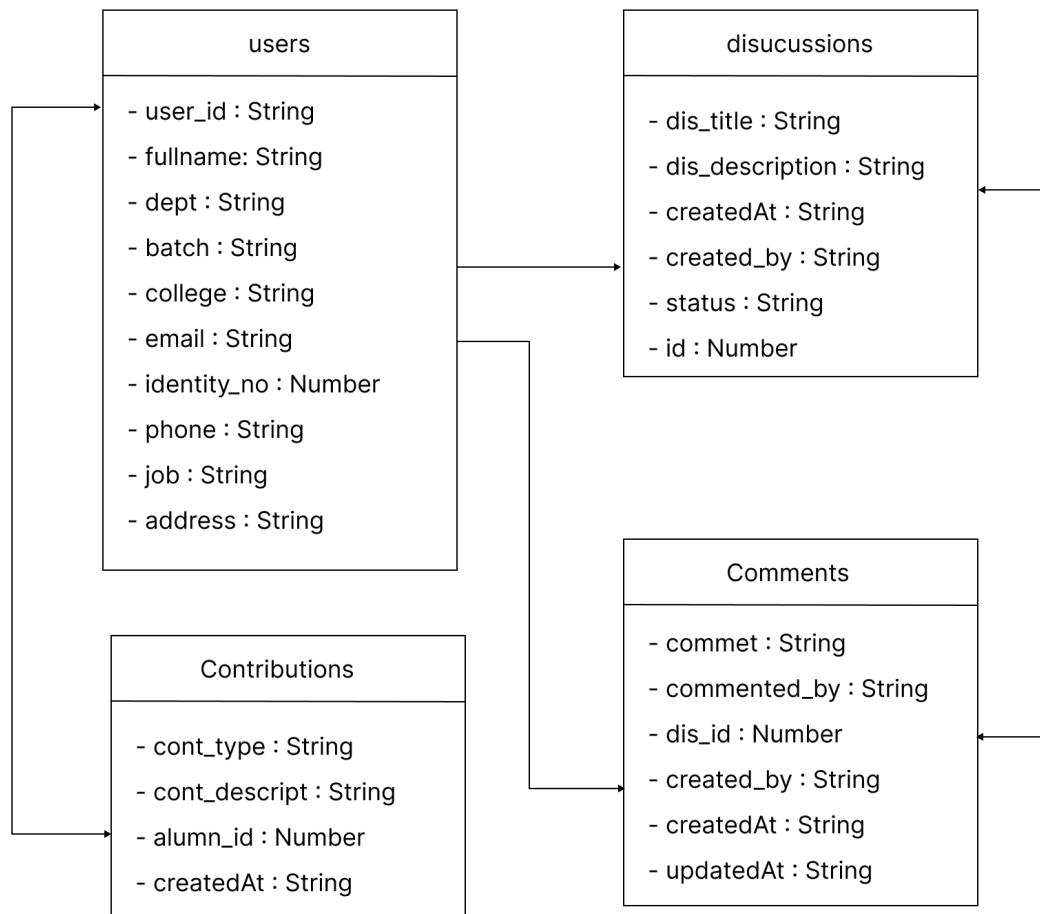


Figure.5.5 Class Diagram

## **CHAPTER 6**

### **MODULES DESCRIPTION**

#### **6.1 USER AUTHENTICATION**

In order to access the Alumni Virtual Office it's necessary for an alumnus to sign up in our portal. Once they signup they will get that message of wait for the approval from admin. Once admin receive the registration and he verify his details and approve him. He can be able to access the Virtual office just by logging in to the application.

#### **6.2 PAAVAIANS**

We will be listing down all the alumni details in the Paavaians module including the place they are working in and their job description. So, it would be easier for our management to connect the students to alumni if they need help on placements in specific company or it can also help alumni to get referrals in some companies where our alumni were working.

#### **6.3 DISCUSSIONS**

It is the core module of the entire AVO. In this module an alumni can create a discussion. Once they create the discussion the admin will look into the discussion and will either publish or reject the discussion. If the discussion is published other alumni can be able to comment down their thoughts on that discussion. But in the comments section also there will be an approval system. Whenever a someone comment down the admin on the other side will receive the notification about that and on approval of that comment it will be posted for that post.

#### **6.4 CONTRIBUTIONS**

If some alumni are interested to conduct webinar, workshop or any events in Paavai or willing to help juniors in getting placed they can just fill the form in the portal and the staff coordinator in our college will connect with them directly and schedule the events according to them and college.

# **CHAPTER 7**

## **SYSTEM TESTING**

### **7.1 SYSTEM TESTING**

Software testing is a method of assessing the functionality of a software program. There are many different types of software testing but the two main categories are dynamic testing and static testing. Dynamic testing is an assessment that is conducted while the program is executed; static testing, on the other hand, is an examination of the program's code and associated documentation. Dynamic and static methods are often used together. There are several rules that can serve as testing objectives, they are

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has high probability of finding an undiscovered error.
3. A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to be working according to the specification, that performance requirements appear to have been met. There are three ways to test a program

1. For Correctness
2. For Implementation efficiency
3. For Computational Complexity.

Tests for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

Tests for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code-refining process, which reexamines the implementation phase of algorithm development. Tests for computational complexity

amount to an experimental analysis of the complexity of an algorithm or an experimental comparison of two or more algorithms, which solve the same problem.

A quality team deputed by the management verified all the necessary documents and tested the Software while entering the data at all levels. The development process involves various types of testing. Each test type addresses a specific testing requirement. The most common types of testing involved in the development process are:

- Unit Test.
- System Test
- Integration Test
- Functional Test

## **7.2 TYPES OF TESTING**

### **7.2.1 FUNCTIONAL TESTING:**

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

### **7.2.2 INTEGRATION TESTING:**

Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is released.

### **7.2.3 WHITE BOX TESTING:**

Testing based on an analysis of internal workings and structure of a piece of software. This testing can be done sing the percentage value of load and energy. The tester should know what exactly is done in the internal program. Includes techniques such as Branch Testing and Path Testing.

## **CHAPTER 8**

### **CONCLUSION**

This system will be available for public use through the web interface. A non-registered visitor can look at the list of graduates according to year of graduation or a field of study. He can also look at graduates' profiles. The level of profile details shown to the public is limited. By default, a public visitor can only see name and surname of a graduate, year of graduation and a field of study. The faculty endeavors to propagate its graduates. Therefore, graduates can also add some information about themselves into the system during the study such as working experience, knowledge. Graduates can enable to display this information in their profiles for public visitors. Inserted information can be used as an input for generating graduate's curriculum vitae in pdf format, which is provided automatically. It is in a graduate's competence, which information will be searching pages with their crawlers. A graduate can use it for the building of his virtual web identity on the internet. Our Alumni system solves the problem concerned with graduate's feedback to the faculty with an inquiry module. In this module the faculty can define questions with answers which active graduates can respond. This module should be used for collecting data which are not included in graduate's profiles and have high information value for the faculty.

So, the Alumni Virtual Office is mainly used to share the views between the users of the application which is very useful to upgrade the knowledge of everyone. The application is also served as a useful site to know what is going on in our college and can also know about the various opportunities of the outer world. The application can be further expanded by following the future Enhancements mentioned above.

# CHAPTER 9

## APPENDIX

### 9.1 SOURCE CODE

#### index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<link rel="icon" type="image/svg+xml" href="/vite.svg" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<!-- google fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
  href="https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"
  rel="stylesheet">
<title>Alumni Virtual Office</title>
</head>
<body>
<div id="root"></div>
<script type="module" src="/src/main.jsx"></script>
</body>
</html>
```

### main.jsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import {
  createBrowserRouter,
  RouterProvider,
  Navigate,
} from "react-router-dom";
import store from './store/index.js'
import DiscussionSingle from './Pages/DiscussionSingle';
const token = Cookies.get("token");
const router = createBrowserRouter([
  {
    element: <App />,
    children: [
      {
        path: "/",
        element: (
          <CheckAuth>
          <Home />
          </CheckAuth>
        ),),
        {
          path: "/profile",
          element: (
            <CheckAuth>
            <Profile />
            </CheckAuth>),
        },
    ],
  },
]);
```

```
{  
  path: "/paavaians",  
  element: (  
    <CheckAuth>  
    <Paavaians />  
    </CheckAuth>),  
  ),  
  {  
    path: "/paavaians/:id",  
    element: (  
      <CheckAuth>  
      <PaavaianDetails />  
      </CheckAuth>),  
    ),  
    {  
      path: "/discussions",  
      element: (  
        <CheckAuth>  
        <Discussions />  
        </CheckAuth>),  
      ),  
      {  
        path: "/discussions/:id",  
        element: (  
          <CheckAuth>  
          <DiscussionSingle />  
          </CheckAuth>),  
        ),  
        {
```

```
path: "/contributions",
element: (
<CheckAuth>
<Contributions />
</CheckAuth>),
},],},
{
path: "/login",
element: (
<GuestAuth>
<Login />
</GuestAuth>),
},
{
path: "/signup",
element: (
<GuestAuth>
<Signup />
</GuestAuth>),
},
]);
}

ReactDOM.createRoot(document.getElementById("root")).render(
<Provider store={store}>
<RouterProvider router={router} />
</Provider>
);
```

## Dashboard.jsx

```
import "./App.css";
import Cookies from "js-cookie"
import { useDispatch } from "react-redux"
import { logout } from "./store/auth.js"
import { Outlet } from 'react-router-dom'
const drawerWidth = 240;
function Dashboard(props) {
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const { window } = props;
  const [mobileOpen, setMobileOpen] = React.useState(false);
  const handleDrawerToggle = () => {
    setMobileOpen(!mobileOpen);
  };
  function _logout() {
    Cookies.remove("token")
    dispatch(logout())
    navigate("/login")
  }
  const navItems = [
    { key: "1", label: "Home", route: "/", icon: iconHome },
    { key: "2", label: "Profile", route: "/profile", icon: iconUser },
    { key: "3", label: "Paavaians", route: "/paavaians", icon: iconAllUsers },
    { key: "4", label: "Discussions", route: "/discussions", icon: iconDiscussions },
    { key: "5", label: "Contributions", route: "/contributions", icon: iconContributions },
  ]
  const drawer = (
    <>
```

```

<div style={{ height: '65px', display: 'flex', alignItems: 'center', justifyContent: 'center' }}>
  <img src={appLogo} />
</div>
<Divider />
<List>
  {
    navItems.map((hello) => {
      return (
        <Link to={hello.route} key={hello.key}>
          <ListItemIcon key={hello.key} disablePadding>
            <ListItemIconButton>
              <img style={{ marginRight: "20px" }} src={hello.icon} />
              <ListItemText primary={hello.label} />
            </ListItemIconButton>
          </ListItemIcon>
        </Link>
      )));
    <ListItem key="logout" disablePadding>
      <ListItemIconButton onClick={_logout}>
        <img style={{ marginRight: "20px" }} src={iconLogout} />
        <ListItemText primary="Logout" />
      </ListItemIconButton>
    </ListItem>
  </List>
</ >
);
const container = window !== undefined ? () => window().document.body : undefined;
return (

```

```

<Box sx={{ display: 'flex' }}>
  <CssBaseline />
  <AppBar position="fixed" sx={{ width: { sm: `calc(100% - ${drawerWidth}px)` }, ml: { sm: `${drawerWidth}px` }, } }>
    <Toolbar>
      <IconButton color="inherit" aria-label="open drawer" edge="start" onClick={handleDrawerToggle} sx={{ mr: 2, display: { sm: 'none' } }}>
        <MenuIcon />
      </IconButton>
      <Typography variant="h6" noWrap component="div">
        Alumni Virutal Office
      </Typography>
    </Toolbar>
  </AppBar>
  <Box component="nav" sx={{ width: { sm: drawerWidth }, flexShrink: { sm: 0 } }} aria-label="mailbox folders">
    <Drawer
      container={container}
      variant="temporary"
      open={mobileOpen}
      onClose={handleDrawerToggle}
      ModalProps={{ keepMounted: true, // Better open performance on mobile. }}
      sx={{ display: { xs: 'block', sm: 'none' },
        '& .MuiDrawer-paper': { boxSizing: 'border-box', width: drawerWidth },
      }}>

```

```

{drawer}
</Drawer>
<Drawer variant="permanent" sx={{ display: { xs: 'none', sm: 'block' }, '& .MuiDrawer-paper': { boxSizing: 'border-box', width: drawerWidth }, }} open>
{drawer}
</Drawer>
</Box>
<Box style={{ marginTop: "70px" }} component="main" sx={{ marginTop: "60px", flexGrow: 1, p: 3, width: { sm: `calc(100% - ${drawerWidth}px)` } }}>
<Outlet />
</Box>
</Box>
);
export default Dashboard;

```

### **server.js**

```

import { Router } from "express";
import DiscussionRoutes from "./discussionRoutes.js";
import AuthRoutes from "./authRoutes.js";
import UserRoutes from "./userRoutes.js";
import passport from "passport";
import AllUserRoutes from "./allUserRoutes.js";
import CommentsRoutes from "./CommentsRoutes.js";
const router = Router();
router.use(
  "/discussion",
  passport.authenticate("jwt", { session: false }),
  DiscussionRoutes
);

```

```

router.use(
  "/allusers",
  passport.authenticate("jwt", { session: false }),
  AllUserRoutes
);
router.use("/auth", AuthRoutes);
router.use("/user", UserRoutes);
router.use("/comments",passport.authenticate("jwt", { session: false }));
CommentsRoutes
);
export default router;

```

### **discussionController.js**

```

import DiscussionSchema from "../models/DiscussionSchema.js";
export const get = async (req, res) => {
  const retrieveDiscussions = await DiscussionSchema.find({
    alumni_id: req.user._id,
  }).sort({
    createdAt: -1,
  });
  res.json({ data: retrieveDiscussions });
};

export const getLatest = async (req, res) => {
  const retrieveDiscussions = await DiscussionSchema.find().limit(4).sort({
    createdAt: -1,
  });
  res.json({ data: retrieveDiscussions });
};

export const getOne = async (req, res) => {

```

```

const retrieveDiscussions = await DiscussionSchema.find({
  _id: req.params.id,
});

res.json({ data: retrieveDiscussions });

};

export const create = async (req, res) => {
  const { dis_title, dis_description, dis_likes } = req.body;
  const createDiscussion = new DiscussionSchema({
    alumni_id: req.user._id,
    dis_title,
    dis_description,
    dis_likes,
  });
  await createDiscussion.save();
  res.json({ data: req.body });
};

export const destroy = async (req, res) => {
  const deleteDiscussion = await DiscussionSchema.findOneAndDelete({
    _id: req.params.id,
  });
  res.json(`Deleted Successfully. ${deleteDiscussion}`);
};

```

### **userSchema.js**

```

import mongoose, { Schema } from "mongoose";
const UserSchema = new Schema(
  {
    firstName: { type: String, required: ["First name is required"] },
    lastName: { type: String, required: ["Last name is required"] },
  }
);

```

```

email: { type: String, required: ["Email is required"] },
phone: { type: String, required: ["Contact number required"] },
college: { type: String },
identity_number: { type: String },
dept: { type: String },
batch: { type: String },
password: { type: String, required: ["Password is required"] },
alumni_status: { type: String, default: "in-approval" },
user_type: { type: String, default: "alumni" },
},
{ timestamps: true }
);
export default new mongoose.model("UserSchema", UserSchema);

```

### **comments.js**

```

import CommentsSchema from "../models/CommentsSchema.js";
export const create = async (req, res) => {
const { commented_by, discussion_id, comment, status } = req.body;
const createComment = new CommentsSchema({
commented_by: req.user._id,
discussion_id,
comment,
status,
});
await createComment.save();
res.json({ data: req.body });
};

export const get = async (req, res) => {
const retrieveComments = await CommentsSchema.find().sort({

```

```

createdAt: -1,
});

res.json({ data: retriveComments });

};

export const getSpecific = async (req, res) => {
console.log(req.params.id);
const retriveComments = await CommentsSchema.find({
discussion_id: req.params.id,
}).sort({
createdAt: -1,
});

console.log(retriveComments);
res.json({ data: retriveComments });
};

```

### **home.jsx**

```

import React from 'react'

import { useState, useEffect } from 'react';
import { styled } from '@mui/material/styles';
import Grid from '@mui/material/Grid';
import Paper from '@mui/material/Paper';
import { useSelector } from 'react-redux';
import "./styles/home.css"

import DiscussionsLatest from '../components/Discussions/DiscussionsLatest';
import Cookies from 'js-cookie';
import List from '@mui/material/List';
import ListItem from '@mui/material/ListItem';
import Avatar from '@mui/material/Avatar';
import FiberManualRecordIcon from '@mui/icons-material/FiberManualRecord';

```

```

const Item = styled(Paper)(({ theme }) => ({
  backgroundColor: theme.palette.mode === 'dark' ? '#1A2027' : '#fff',
  ...theme.typography.body2,
  color: theme.palette.text.secondary,
  padding: theme.spacing(1),
}));

function Home() {
  function stringAvatar(name) {
    return {
      children: `${name.split(' ')[0][0]}${name.split(' ')[1][0]}`,
    };
  }

  const [latestDiscussions, setLatestDiscussions] = useState([])
  const auth = useSelector((state) => state.auth)
  const { firstName } = auth.user;
  const token = Cookies.get("token")
  async function loadLatestDiscussions() {
    const discussions = await fetch(` ${import.meta.env.VITE_API_URL}/discussion/latest`,
    {
      headers: {
        Authorization: `Bearer ${token}`
      }
    });
    const { data } = await discussions.json();
    setLatestDiscussions(data);
  }
  useEffect(() => {
    loadLatestDiscussions()
  }, [])
}

```

```

const [latestUsers, setLatestUsers] = useState([])
async function getLatestUsers() {
  const res = await fetch(
    `${import.meta.env.VITE_API_URL}/allusers/latest`, {
      headers: {
        Authorization: `Bearer ${token}`,
      },
    });
  const users = await res.json();
  setLatestUsers(users.latestUsers);
}
useEffect(() => {
  getLatestUsers();
}, [])
return (
<div>
<Grid container item rowSpacing={{ sm: 1 }} columnSpacing={{ xs: 1, sm: 2, md: 3 }}>
<Grid item xs={12}>
<Item className="banner">
<h1 className='bannerHeading'>👋 Hello {firstName} </h1>
<Grid container item xs={12} sm={12} spacing={{ sm: 3, md: 3 }}>
<Item className="bannerStats">
<Grid item xs={12} md={3} lg={3}>
<Item className='bannerStats_item'>
<p>123</p>
<span>Discussions</span>
</Item>
</Grid>
<Grid item xs={12} md={3} lg={3}>
<Item className='bannerStats_item'>

```

```

<p>12</p>
<span>Comments</span>
</Item>
</Grid>
<Grid item xs={12} md={3} lg={3}>
<Item className='bannerStats_item'>
<p>86</p>
<span>Contributions</span>
</Item>
</Grid>
</Grid>
</Item>
</Grid>
<Grid container item xs={12} style={{ marginTop: "30px" }} columnSpacing={{ xs: 10, sm: 4, md: 12 }} rowSpacing={{ sm: 4 }}>
<Grid item xs={12} lg={8}>
<div>
<h1 style={{ fontSize: "24px", fontWeight: 600, color: "#a02136" }}>Recent Discussions</h1>
<DiscussionsLatest latestDiscussions={latestDiscussions}>
  <loadLatestDiscussions={loadLatestDiscussions}>
    <setLatestDiscussions={setLatestDiscussions} />
  </loadLatestDiscussions>
</DiscussionsLatest>
</div>
</Grid>
<Grid item xs={12} lg={4}>
<h1 style={{ fontSize: "24px", fontWeight: 600, color: "#a02136" }}>Alumni Recently Joined</h1>
<List item sx={{ width: '100%', gap: "30px" }}>
  {

```

```

latestUsers.map((user) => {return (
  <ListItem key={user._id} style={{ gap: "10px", padding: "10px 0", borderBottom: "1px solid #EFEFEF" }}>
    <Avatar style={{ background: "#a02136", fontSize: "14px" }}>
      {...stringAvatar(` ${user.firstName} ${user.lastName}`)} />
    <div className="user__detials">
      <p className="user__name" style={{ fontWeight: 600, fontSize: "16px" }}>
        `${user.firstName} ${user.lastName}`
      </p>
      <div className='user__meta' style={{ textTransform: "uppercase", fontSize: "14px", display: "flex", alignItems: "center", gap: "4px", fontWeight: 600, color: "#A6A6A6" }}>
        {user.dept}
        <FiberManualRecordIcon className="content__separater" />
        {user.college}
      </div>
    </div>
  </ListItem>
);})
</List>
</Grid>
</Grid>
</Grid>
</div >
)}
export default Home

```

## 9.2 SCREENSHOTS

The image shows a registration form for the Alumni Virtual Office. The form is overlaid on a photograph of a university campus with several large, modern buildings and surrounding greenery. The registration fields include:

- Full Name \*
- Email Address \*
- Phone number \*
- Roll / Register number
- College (dropdown menu)
- Batch (dropdown menu)
- Department (dropdown menu)
- Gender (dropdown menu)
- Date of Birth (with calendar icon)
- Highest Qualification \*
- Present organization \*
- Designation \*
- Current City \*
- State \*
- Country \*
- Password \*

At the bottom right of the form area, there is a red "SIGN UP" button and a link "Already have an account? Sign in".

Figure 9.1: Alumni registration form –  
A user needs to fill this form if he is new to portal

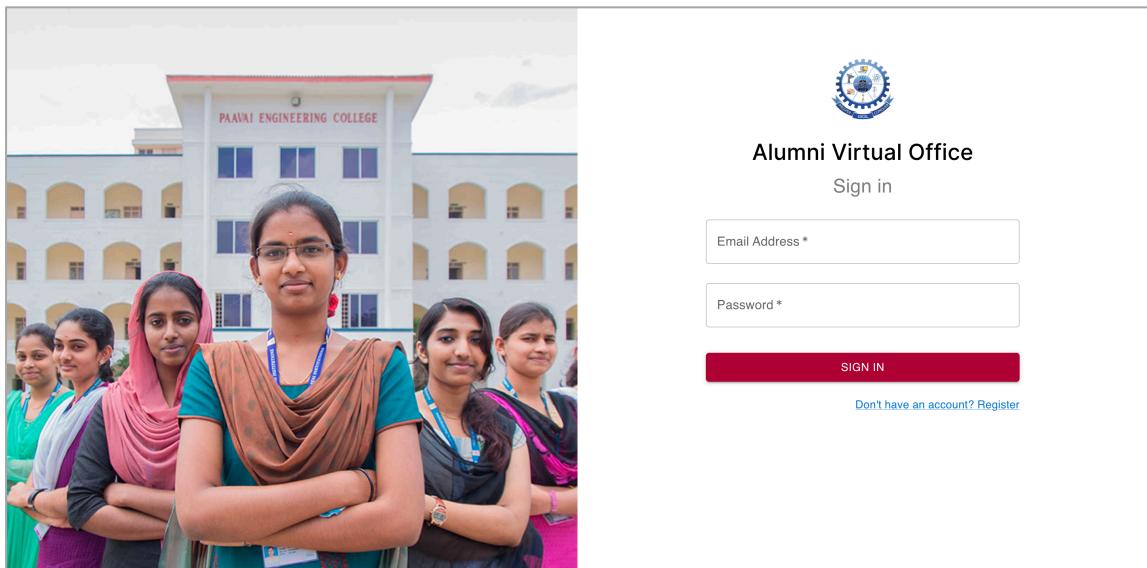


Figure 9.2: Alumni login form – Registered users can enter the login into the form and keep on check the status of the profile.

Account Details	
Profile Status	<b>in-approval</b>
ID	638987bc7e88c6f75f12567e
Joined on	Dec 02, 2022 - 10:12:AM
Email	guest@gmail.com
Phone	9487060472
Gender	male
DOB	12/10/1997
Highest Qualification	B.E.,

Figure 9.3: My profile (In-approval) – User profile will be in-approval by default

The screenshot shows a user profile page titled "Your Profile" for "Vignesh". The page has a dark header with the text "Alumni Virtual Office". On the left is a sidebar with navigation links: Home, Profile, Paavaians, Discussions, Contributions, Notifications, and Logout. The main content area features a large circular placeholder for a profile picture with the letters "VG" in white. To the right of the placeholder, the name "Vignesh" is displayed in a large, bold, black font. Below the name are three sections: "Account Details", "College Details", and "Job Details", each containing several key-value pairs.

**Account Details**

Profile Status	active
ID	638988cd7e88c6f75f125687
Joined on	Dec 02, 2022 - 10:12:AM
Email	vikki.tech@gmail.com
Phone	9994711163
Gender	male
DOB	09/08/1992
Highest Qualification	Ph.D.,

**College Details**

College	pec
Department	CSE
Batch	2023

**Job Details**

Present Organization	Paavai Engineering College
Designation	Admin
Current City	Namakkal
State	Tamilnadu
Current City	Namakkal
Country	India

Figure 9.4: My profile (Active) – User profile will be in-approval by default

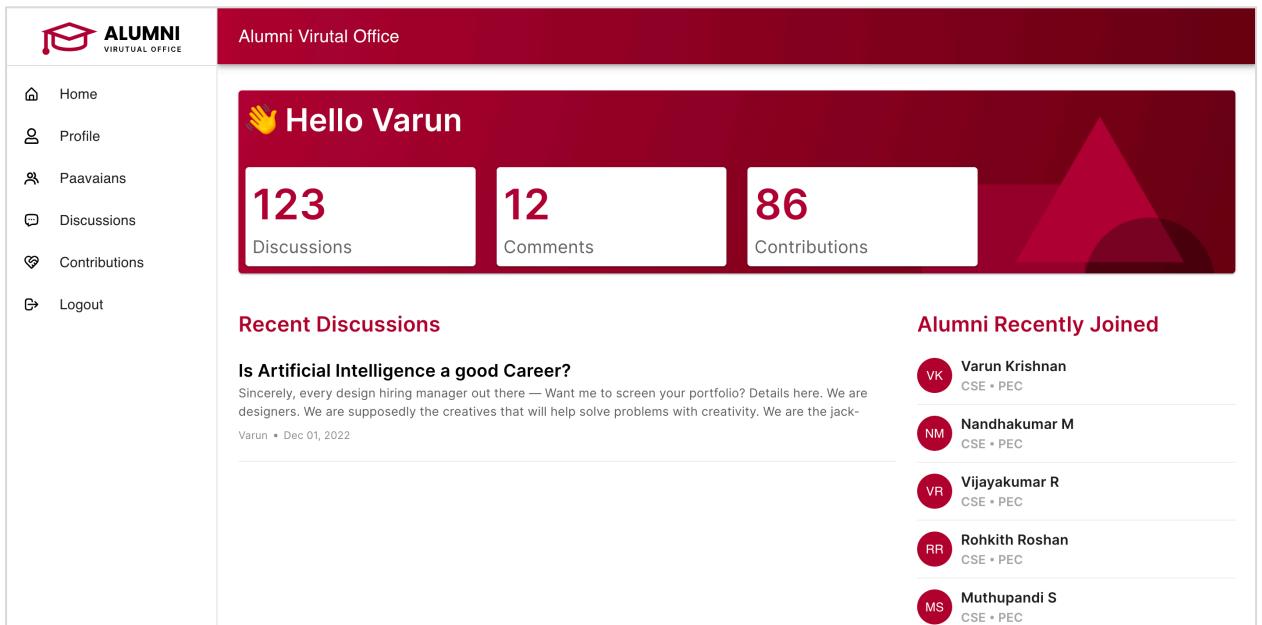


Figure 9.5: Alumni dashboard – Can view the latest discussions and the alumni who recently joined the portal

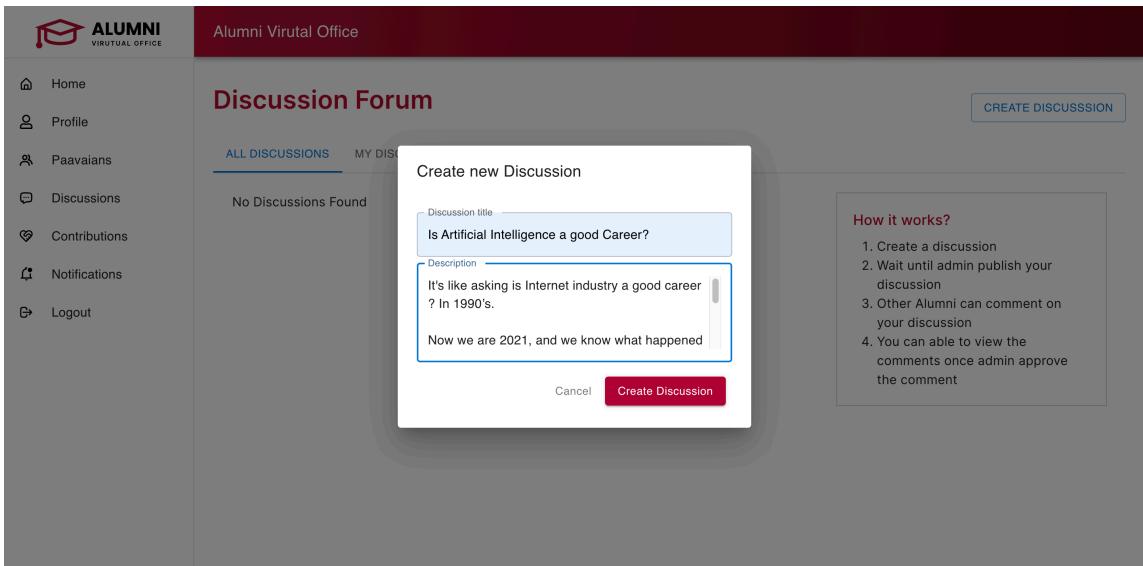


Figure 9.6: Creating discussion – If you have any queries regarding job or other alumni can create a discussion.

**How it works?**

1. Create a discussion
2. Wait until admin publish your discussion
3. Other Alumni can comment on your discussion
4. You can able to view the comments once admin approve the comment

Figure 9.7: Discussion forum – The created discussion will show under my discussions.

Figure 9.8: Comments section before approval - Alumni must wait for admin approval to get the discussion get published and other can be able to comment on your discussion.

The screenshot shows a discussion post titled "Is Artificial Intelligence a good Career?" by "Varun Krishnan" on December 01, 2022. The post content is: "Sincerely, every design hiring manager out there — Want me to screen your portfolio? Details here. We are designers. We are supposedly the creatives that will help solve problems with creativity. We are the jack-". Below the post, there are five approved comments from "Varun Krishnan" dated "18 July, 2022", each consisting of the placeholder text "Lorem ipsum dolor sit amet consectetur. Dignissim quam sed libero volutpat aenean odio.". A "Comment" button is visible at the bottom right of the comment section.

Figure 9.9: Comments section after approval – The comment section will be turned on once the admin approve the discussion

The screenshot shows a table titled "Paavians" displaying user details. The columns are: Alumni Name, Phone, Email, College, Batch, Department, Status, Company, and View Profile. The data rows are:

Alumni Name	Phone	Email	College	Batch	Department	Status	Company	View Profile
Alumni	9876321823	alumni@gmail.com	pec	2023	CSE	active	Tata Consultancy...	<a href="#">View Profile</a>
Guest	9487060472	guest@gmail.com	pec	2022	CSE	active	Google	<a href="#">View Profile</a>
Admin	9994711163	admin@gmail.com	pec	2023	CSE	active	Paavai Engineeri...	<a href="#">View Profile</a>

At the bottom, there are pagination controls: "Rows per page: 10", "1–3 of 3", and navigation arrows.

Figure 9.10: All users – Admin can be able to see all the users and their details.

The screenshot shows a user profile for "Paavaian Profile". The profile belongs to "Nandhakumar M". The user's ID is 6381e56bdb7babfa1363efd6, and their profile status is "in-approval". Their email is nandhakumarm26052002@gmail.com, and their phone number is 9345211050. They graduated in 2023 from PEC with a CSE degree. The profile was created at 2022-11-26T10:07:39.785Z.

Figure 9.11: Alumni profile – To view all the details of the alumni

The screenshot shows the "Notifications" page. It lists one user, "Paavians", with the following details:

Alumni Name	Phone	Email	College	Batch	Department	Status	View Profile
Guest	9487060472	guest@gmail.com	pec	2022	CSE	in-approval	<a href="#">View Profile</a>

Figure 9.12: Notifications page – It will show all the list of users, discussions, comments and contributions in approval.

## **CHAPTER 10**

### **REFERENCES**

- [1] H. Lei, Y. Cui and W. Zhou, "Relationships between student engagement and academic achievement: a meta-analysis", *Social Behavior and Personality: an international journal*, vol. 46, no. 3, pp. 517-528, 2018.
- [2] M. Dollinger, S. Arkoudis and S. Marangell, "University alumni mentoring programs: a win-win?", *Journal of Higher Education Policy and Management*, vol. 41, no. 4, pp. 375-389, 2019.
- [3] B. S. Babu, K. Bhargavi and P. Venkatarm, "CAs-based QoS scheme for remote health monitoring", *International Journal of Agent Technologies and Systems (IJATS)*, vol. 5, no. 4, pp. 44-67, 2013.
- [4] H. Lathashree, N. J. M. Katte, K. P. Pooja, B. S. Babu and K. Bhargavi, "Multiple CAs based Framework with ANN for Load Balancing in Cloud Computing Environment", 2018 4th International Conference for Convergence in Technology (I2CT), pp. 1-6, 2018, October.
- [5] H. Lathashree, N. J. M. Katte, K. P. Pooja, K. Bhargavi and B. S. Babu, "Multiple CAs Based Framework to Provide Remote Palliative Care for Patients Undergoing Chemotherapy", *International Conference on Advances in Computing and Data Sciences*, pp. 177-186, 2018, April.
- [6] M. S. Sengar, M. Gharewal, N. Patidar, P. Bhanodia and K. Tiwari, ALUMNI HUB.
- [7] B. Anthony, "Information Flow Analysis of a Knowledge Mapping-Based System for University Alumni Collaboration: a Practical Approach", *Journal of the Knowledge Econom*, pp. 1-32, 2020.
- [8] N. Rattanamethawong, S. Sinthupinyo and Chandrachai, "An innovation model of alumni relationship management: Alumni segmentation analysis", *Kasetsart Journal of Social Sciences*, vol. 39, no. 1, pp. 150-160, 2018.
- [9] V. Rattanamethawong, S. Sinthupinyo and Chandrachai, "An innovation system that can quickly responses to the needs of students and alumni", *Procedia- Social and Behavioral Sciences*, vol. 182, pp. 645-652, 2015.