A PROJECT REPORT ON

"Alumni Management System"

Submitted to:

CENTRE FOR COMPUTER SCIENCE AND APPLICATIONS,

DIBRUGARH UNIVERSITY

In the partial fulfilment of the requirement for the

6th Semester of Master of Computer Applications (MCA)

Session: 2019-2022

Under the Supervision of Dr. Rizwan Rehman, Assistant Professor, Centre for Computer Science and Applications, Dibrugarh University

Submitted by:

BHASKARJYOTI GOGOI

ROLL NO. 19992011

CENTRE FOR COMPUTER SCIENCE AND APPLICATIONS
DIBRUGARH UNIVERSITY,
DIBRUGARH – 786004,
ASSAM, INDIA

CERTIFICATE

This is to certify that the project report entitled **"Alumni Management System"** by **Bhaskarjyoti Gogoi (19992011)** to the Dibrugarh University, Dibrugarh, Assam, in partial fulfillment of the requirement for the award of degree of **MASTER OF COMPUTER APPLICATIONS.** It is a bona fide record of the project work carried out by them under my supervision during the year of 2019-2022.

.....

Guide: Dr Rizwan Rehman

Assistant Professor

Centre for Computer Science and Applications

Dibrugarh University

Date:

CERTIFICATE

This is to certify that the project reports entitled **"Alumni Management System"** by **Bhaskarjyoti Gogoi (19992011)** to the Dibrugarh University, Dibrugarh, Assam, in partial fulfillment of the requirement for the award of degree of **MASTER OF COMPUTER APPLICATIONS.** It is a bona fide record of the project work carried out by them under my supervision during the year of 2019-2022.

.....

Dr. Tazid Ali

Chairperson

Centre for Computer Science and Applications

Dibrugarh University

Date:

EXAMINATION CERTIFICATE

This is to certify that **Bhaskarjyoti Gogoi** bearing roll number **19992011** of the Centre for Computer Science and Applications, Dibrugarh University has carried out the project work in a manner satisfactory to warrant its acceptance and also defended it successfully. I wish him all the success in his future endeavors.

Examiners:

1. Name……. Signature……
2. Name……. Signature……
3. Name……. Signature……
4. Name……. Signature……
5. Name……. Signature……

DECLARATION

I hereby declare that the project work entitled **"Alumni Management System"** submitted to the Centre for Computer Science and Applications, Dibrugarh University, Assam for the partial fulfilment of the requirement for the award of degree of **MASTER OF COMPUTER APPLICATIONS**. These are original work done by me under the guidance of Dr. Rizwan Rehman and has not been submitted for the award of any degree.

BHASKARJYOTI GOGOI

Roll No: 19992011

ACKNOWLEDGEMENT

In performing my project, I had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives me much pleasure. I would like to show my gratitude to all the faculty members of CCSA, Dibrugarh University for giving me good guideline for the project throughout numerous consultations. I would also like to expand my deepest gratitude to all those who have directly and indirectly guide me in completion of this project.

In addition, a thank you to my entire faculty members, who introduced us to the Art of Computer Programming and their passion for the "Underlying Structures", had lasting effect. I also thank the University of Dibrugarh for consent to include copyright pictures as a part of our paper.

I express my heart-felt gratitude to our Chairperson, Dr Tazid Ali, Centre for Computer Science and Applications, for giving me the opportunity to complete the project successfully.

I wish to extend my sincere thanks to Dr. Rizwan Rehman, Assistant Professor, Centre for Computer Science and Applications, for his valuable guidance and support. I thank him for the timely suggestions and constant encouragement that led to the accomplishment of this project. I am very much thankful for the proper guidance and immense help provided in the process of our project work.

Lastly, I take this opportunity to express my sincere word of gratitude and thanks to all the staffs of the centre, friends and all my well-wishers, who had supported me directly or indirectly, to complete my project

THE AUTHOR

BHASKARJYOTI GOGOI

Abstract

The main aim of the project is to develop a web-based application that is going to make it possible for the former students of university to keep in touch with each other and that will substitute the manual system of the alumni office with an automated one. First of all, the tasks that need to be completed are stated. There is a separate module where each task that was identified as necessary in order to successfully develop the alumni web site of university is presented and described in detail. After each phase of the software development life cycle will be explained, the created system will be evaluated together with the software development process that was applied in order to implement the web portal.

Chapter 1: Introduction

1.1. Title of the Project

Alumni Management System

1.2. Introduction

Many universities around the world offer services and benefits to the alumni students. In order to make the services and benefits available for the graduated students, many university/colleges have alumni websites. An alumni website is a web-based application where the former students can take advantage of the benefits and services that a university offers after they graduate. Being a member of an alumni web site of a university/college offers many services and benefits for the graduated students. For example, the alumni web site of a university/college keeps a person in track with the events that are organized by the university and informs the members when some important events will occur that have not been organized by the university. Furthermore, an alumni website contains information about some important news which is related to the university/college or some important news that is related to a person's field of study. Another advantage of being a member of an alumni web site, which is considered to be the main reasons why many universities/colleges have an alumni web site, is that a person can easily find some information concerning a former student and the person can easily contact any other member of the alumni community. College alumni web site is of great importance to the university at the moment. Because of the fact that the number of people that have graduated at college is growing fast, maintaining contact with the graduates has become a very important issue. Furthermore, it is also very important for university to make it possible for a graduate student to be able to maintain contact with another graduated student. At this moment, Centre for Computer Science and Applications, Dibrugarh University does not has a alumni web portal.

1.3 Purpose

This system can be used as an web application to manage the university information and student's information. The system is an online application that can be accessed throughout the organization and outside which will give better experience to the users. The graduated

students can easily find some information such as contact details for another graduate of university using the portal, which means that they can easily contact any person that is a member of the alumni web site. Making it possible for the former students to keep in touch is not the only functionality that the application will offer. The former students can easily contact the alumni officer if some help is required. Furthermore, the former students will have access to some news that are related to college.

1.4 Achievements:

By successfully implementing the project, a substantial knowledge can be acquired on the implementation of a full stack application using various programming languages. This knowledge will be useful in the future in creating other application.

Chapter 2: Objective & Scope

2.1. Objectives

Main objective of this project is as follows:

- Details of the alumni can be accessed through web 24x7.
- Provide various information to the alumni using the web portal
- Collect donations for the university.

2.2. Scope

The system will provide registration of alumni of the organization. Admin can send notifications to the alumni also alumni can donate for the university. The system will provide constituent information to requestors only under certain conditions such as the alumni should be a registered user.

Chapter 3: Problem Definition

Since Alumni is an important and integral part of any institution, it is very much necessary for an institution to have a proper alumni management system that is digitalised and easy to access and maintain. The institution must opt for a centralized database system for this purpose. 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 the organization 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. Further more if the system is digitalised it will be easy to access, manage, utilise and accessible.

Chapter 4: System Analysis

4.1 Existing System:

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 the university 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.

4.1.1 Limitations in Existing System:

- The current system is not completely complete computerized and manual system in entering students and handling it.
- There is no centralized database maintenance.
- There is no easy access to the particular students record.
- It is not possible to students to access any information.

4.2 Proposed 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 the organization want any record, they can easily search all the records.

4.2.1 Advantages over Existing System:

- It is completely automated system in handling the database.
- This system provides centralized database maintenance.
- This system provides easy access to the particular students account or his/her complete details.
- This system provides student to easily navigate through the application for more information in a most secure manner.

Chapter 5: System Requirements and Specifications

5.1 Software Requirements:

Operating System : Windows 7 or higher, Mac OS X 10.7 or higher, Linux

Browser : Google Chrome, Safari, Opera, Firefox, Edge

5.2 Hardware Requirements:

RAM : 512 MB or higher

Processor : Intel Pentium IV or higher

Hard Disk : 20 GB or higher

Chapter 6: Technologies, Tools and Services Used

6.1 Technologies Used

6.1.1: Frontend Technologies

ReactJs: ReactJS is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

HTML: HTML 5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard, and subsumes XHTML. It currently exists in two standardized forms: HTML 5.2 Recommendation by the World Wide Web Consortium (W3C, a broad coalition of organizations), intended primarily for Web content developers; and HTML Living Standard by WHATWG (a small consortium of four browser vendors), intended primarily for browser developers, though it also exists in an abridged Web developer version.

CSS: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Bootstrap: Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front-end development only.

6.1.2: Backend Technologies

Laravel: This is a free and open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern and based on Symfony. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar.

MySQL: MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founders Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

6.2 Tools Used

Visual Studio: Visual Studio Code (also known as VS Code) is a free open-source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring. And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes.

XAMPP: XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming

languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

phpmyadmin: phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while we still have the ability to directly execute any SQL statement.

Postman: Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better API faster. The Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle—from design, testing, documentation, and mocking to the sharing and discoverability of our APIs.

Composer: Composer is an application-level dependency manager for the PHP programming language that provides a standard format for managing dependencies of PHP software and required libraries. It was developed by Nils Adermann and Jordi Boggiano, who continue to manage the project. They began development in April 2011 and first released it on March 1, 2012.

React Developer Tools: It is a Chrome DevTools extension for the open-source React JavaScript library. It allows us to inspect the React component hierarchies in the Chrome Developer Tools.

We will get two new tabs in our Chrome DevTools: "⚛ Components" and "⚛ Profiler". The Components tab shows us the root React components that were rendered on the page, as well as the subcomponents that they ended up rendering.

6.3 Third party services used:

Fast2Sms: Fast2SMS is a bulk SMS service provider. We have used it in our project to send OTPs.

Razorpay: Razorpay is a payment solution in India that allows businesses to accept, process and disburse payments with its product suite. It gives you access to all payment modes including credit card, debit card, net banking, UPI and popular wallets including Jio Money, Mobikwik, Airtel Money, Free Charge, Ola Money and PayZapp. We have used Razorpay to accept payments form the alumni.

OneSignal: We have used OneSignal to send push notification to the registered alumni.

Chapter 7: Feasibility Study

A feasibility study is undertaken to determine the possibility of developing and adopt the proposed solution for managing "Alumni Management System". The purpose of feasibility study is to determine the existence of scope of developing a new computerized information system to manage alumni details, payment details etc.

In feasibility analysis section we find out the feasibility of our system from three points of view:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility
- Legal Feasibility

7.1 Technical Feasibility

Our project is technically feasible because the software is prepared with Laravel 8 and MySql 7.0 as back end and React 18.0, HTML, CSS, Bootstrap 5 as front end, which are easily freely available. Algorithm developed to meet the purpose is also mathematically feasible for real time calculation without making any delay in processing. Hardware support required for the project is also easily available in the market. So, the final system is very much suitable in terms of accuracy, reliability, ease of access and data security.

7.2 Economic Feasibility

Our system is economically feasible to the organization as the proposed system has many cost advantages over the existing system. The system will minimize the human efforts required to manage the system. Also, the information will be processed very quickly. The benefit derived from the proposed system is much more than cost efficient and so we can accept it as economically feasible.

7.3 Operational Feasibility

Under the consideration it is examined that, our proposed system does not affect the staff related with information processing, as people are familiar with computer system. In fact,

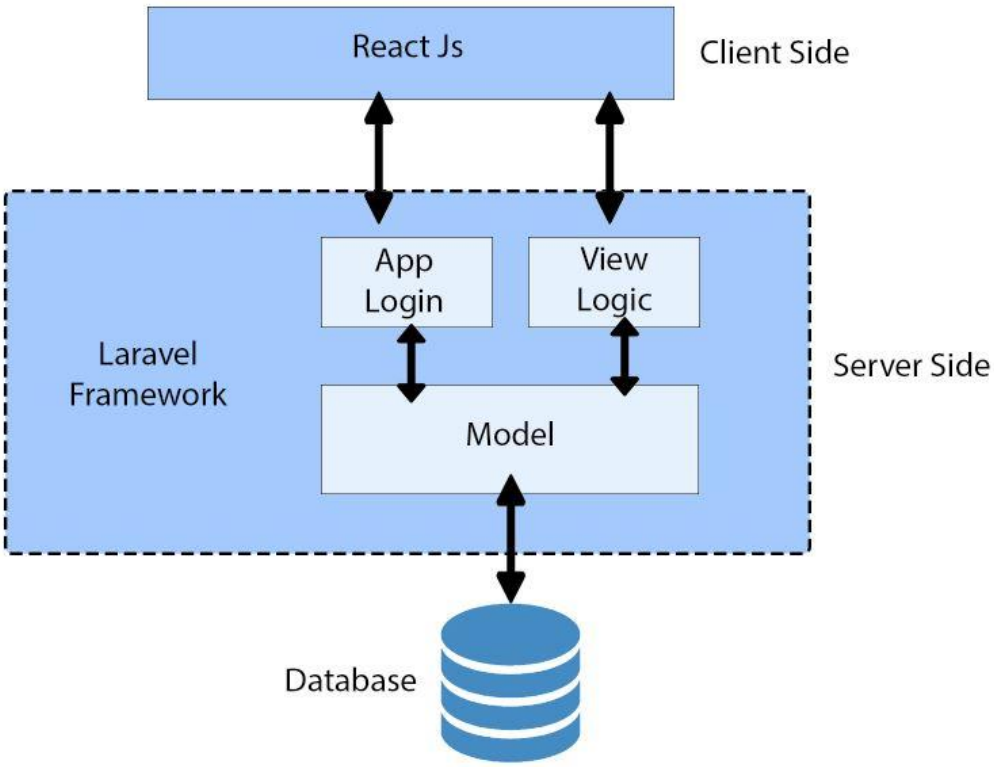
only it reduces fatal man-hour of work service and necessary calculation over existing system. So, the proposed system is considered operationally feasible.

7.4 Legal Feasibility

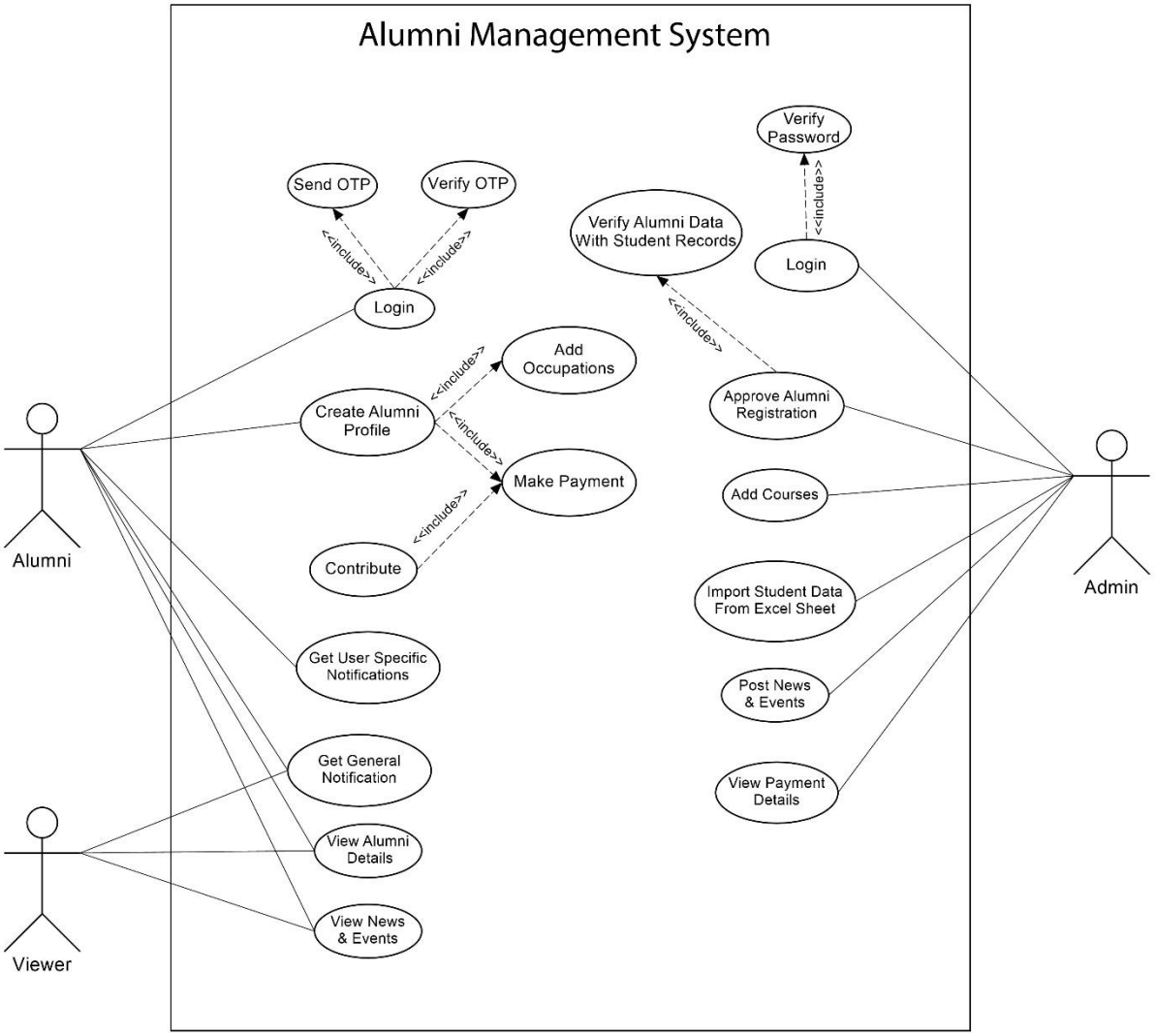
It is examined that our project does not break any laws or making any threat to society. So our project is legally feasible.

Chapter 8: System Design

8.1 Architecture Diagram:

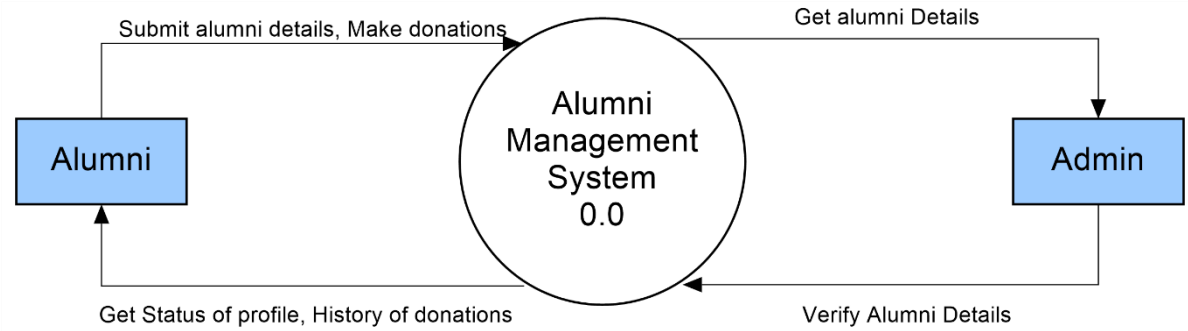


8.2 Use Case Diagram:

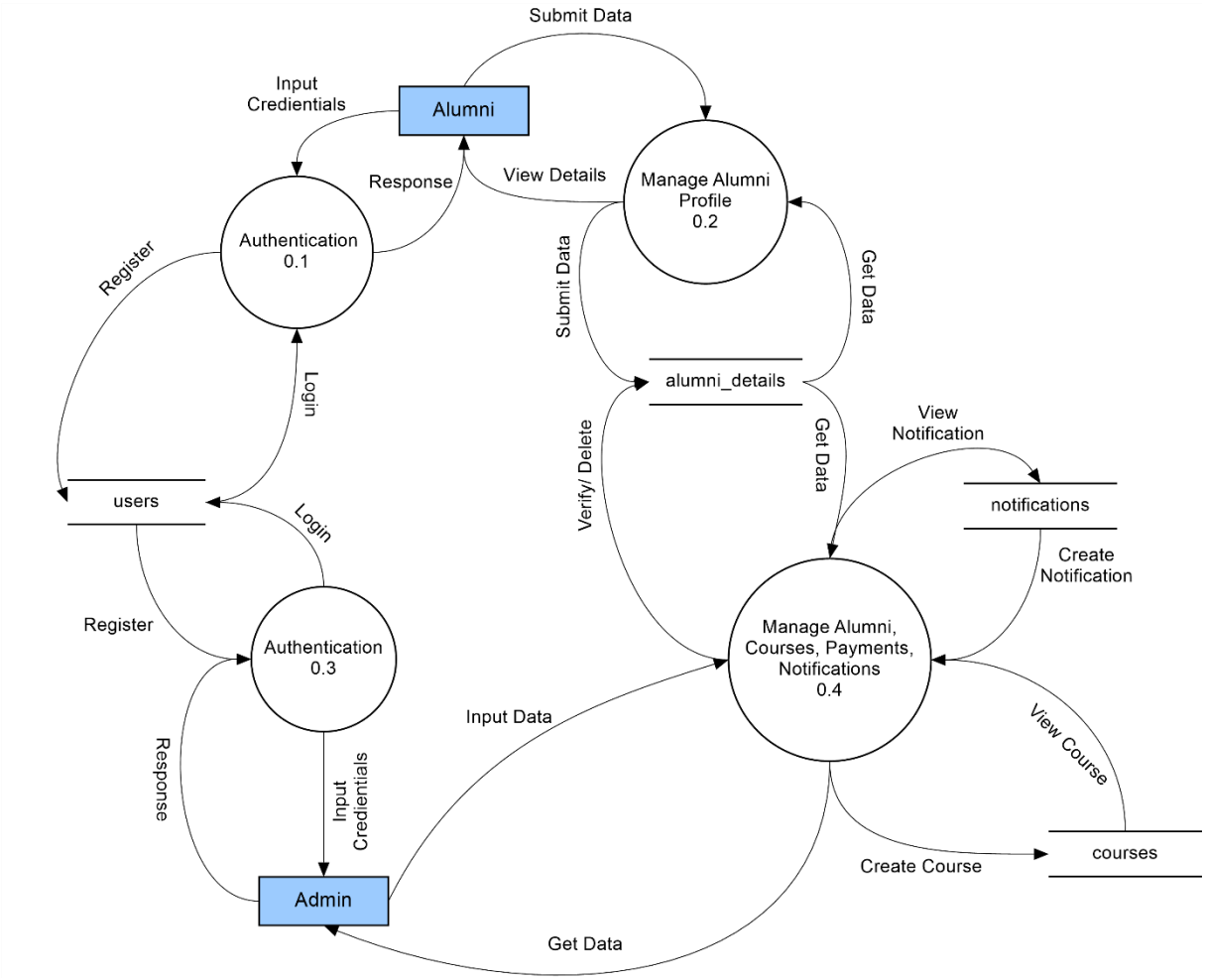


8.3 Data Flow Diagram:

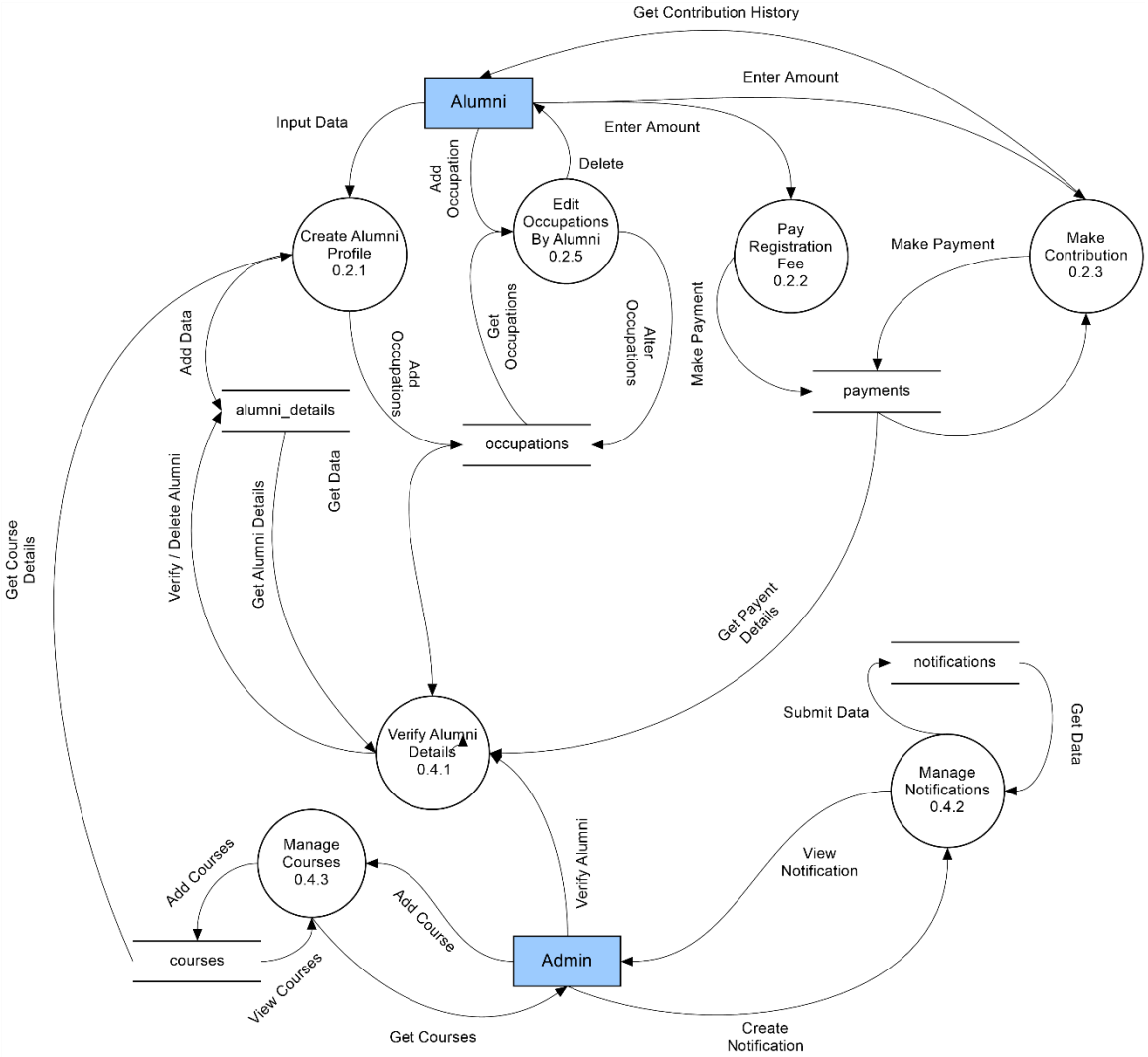
Level 0:



Level 1:



Level 2:



8.4 Module Description:

Our project mainly consists of 2 modules they are:

- General module
- Alumni module
- Admin module

8.4.1 General Module:

The general module is the starting webpage of our project which gives information about the centre (CCSA) and provides a menu to navigate through the different webpages of the portal. The menu present in the general webpage are:

- Home
- Alumni
- News & Events
- Login

Home: The home is linked to the first page or index page of the portal.

Alumni: The alumni is linked to the alumni details page where viewers can see the details of the alumni.

News & Events: News & Events is linked to the news page where viewers can read about various news and events organized in the organization.

Login: Login is linked to the login page where alumni can login and register themselves on the portal.

Also, a notification section is available in the home page, where general notifications posted by the admin will be displayed.

8.4.2 Alumni Module:

The alumni module is used by the alumni of the organization. This module displayed in the webpage when the alumni logged in the system using login page. This module is provided only for the registered members of the organization. In this module students can register by providing his/her phone number. After providing an opt will be sent to the provided number and after successful otp verification the student can register on the portal by providing his/her name and email. If the student is already registered, the student will be redirected to the dashboard. Students can create their alumni profile by filling up a form on the website and pay a registration fee. After successfully fill up the form, admin will verify the details provided by the student and if admin approves the students, students can view the different webpages of the module. In this module there is a notification section where an alumni can view the notifications specifically sent to the alumni.

The alumni can edit various information eg. occupation using the alumni module. Also they make monetary donations to the organization.

- Service used to send OTP: **Fast2SMS**
- Service used to collect money: **Razorpay**

Menu present in the Admin Module:

- Home
- Dashboard
- Profile
- Notifications
- Contribute
- Settings
- Logout

Home: Home is linked to the index page of the general module. By clicking on home users can go to the index page of the portal.

Dashboard: Dashboard is linked to the dashboard page of the Alumni Module.

Profile: Profile is linked to the alumni profile page where alumni can their details they have provided on creating the alumni profile. Also they can edit some information on this page.

Notifications: Notifications is linked to the notification page where notifications sent to the alumni can be seen.

Contribute: Contribute is linked to the contribution page where alumni can make monetary contributions. Also they can see the history of their contributions. They can see the top contributors also.

Settings: Settings is linked to the settings page where alumni can see the phone number and email they provides. If they wish to change the phone number and email, it is also possible.

Logout: By clicking on logout alumni can logout from the system.

8.4.3 Admin Module:

The admin is the one who looks after all the changes and developments that takes places. There may be one or more admin. This module is strictly dedicated to the administrator of the system. In this module the following menus come into view:

- Home
- Dashboard
- Alumni

- New Alumni
- Courses
- Notification
- User Notifications
- News & Events
- Payments

Home: By clicking on home the admin will be redirected to the index page of the portal.

Dashboard: In dashboard admin can see various details such as total alumni registered, total new alumni registered, total payments collected and top contributors etc.

Alumni: In alumni they can see the registered alumni. Also there is a filter option using which admin can retrieve alumni data by course and year of passing. By click on a particular alumnus the alumni details page will be displayed where admin can see full details of a particular alumnus. There is also a button to delete alumni details.

New Alumni: Here admin can see the newly registered alumni which are not approved yet. Alumni can view the details of an alumnus and can verify his/her profile.

Courses: In courses admin can add/remove courses and also can see the courses that are previously added.

Notification: Admin can send/delete general notifications from here and also can see all the notifications.

User Notifications: Admin can send/delete user specific notifications from here and also see all the notification previously sent.

News & Events: Add can add/remove news from here.

Payments: Here all the payments details will be viewed.

Logout: Admin logout from the system by clicking on logout.

8.5 Database

To move with the times, we need to keep track of information. The success of business enterprise today depends much on how quickly it can retrieve the desired information needed for decision making and routine operation. Databases are store-houses of information. For retrieving any information, we fall back on databases. Herein comes the role of database management system which helps in systematic organization of information. Amongst the many database management system available in the market. MySQL is quite useful and popular because of its unique feature. A database has many advantages. Many of the useful databases are those which can provide latest information's. The information in a database should be organized in such a way that it is easy to update. And only then, it can quickly provide the information.

8.5.1 Objects of a relational database:

Tables: A Table is a collection of information on specific topic. In a table, the information is stored in rows and columns. We can store different types of data in different tables.

Components that need to make up a table are:

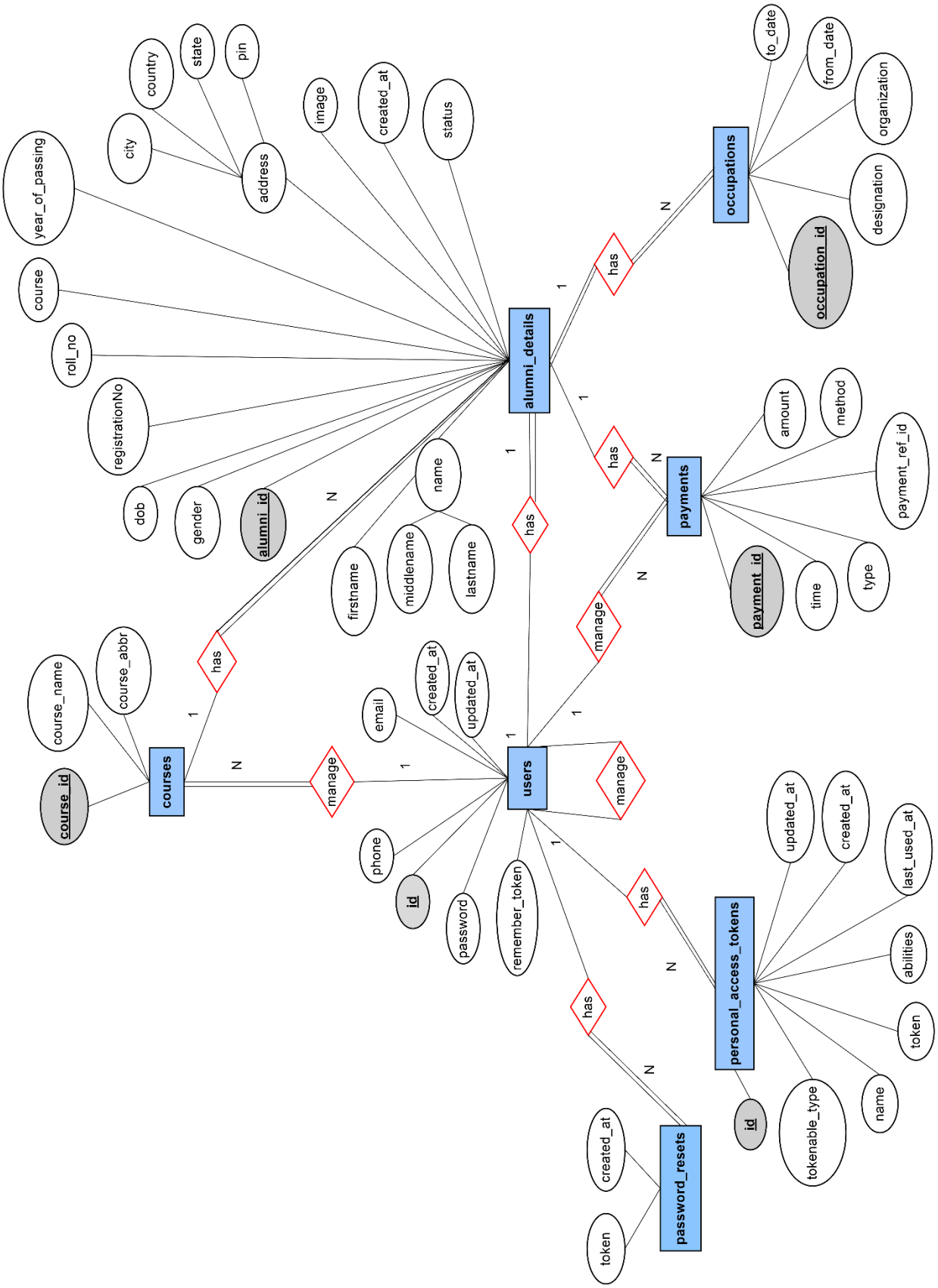
Columns: Each column represents a field. A field stores only a specific category of information.

Rows: Each row in a table is called a record and it consists of a number of related fields. Each field contains some bits of data about the record.

Domain: The maximum and minimum values a field can have, is called its domain.
Primary key: A primary key refers to one or more fields in a table that uniquely identify each record in the table i.e. it gives a distinct identity to a record.

Foreign key: A foreign key refers to primary key of another table.

8.5.2 ER Diagram:



8.5.3 Database design:

users table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		No	*None*		
3	phone	bigint(20)			No	*None*		
4	email	varchar(255)	utf8mb4_unicode_ci		No	*None*		
5	password	varchar(255)	utf8mb4_unicode_ci		Yes	*NULL*		
6	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	*NULL*		
7	created_at	timestamp			Yes	*NULL*		
8	updated_at	timestamp			Yes	*NULL*		

The users table is used to storing the details of the student in the database. It consists of 8 attributes. The user registration is done in user and admin module. Further the entries can be modified in alumni modules.

personal_access_tokens table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	tokenable_type	varchar(255)	utf8mb4_unicode_ci		No	*None*		
3	tokenable_id	bigint(20)		UNSIGNED	No	*None*		
4	name	varchar(255)	utf8mb4_unicode_ci		No	*None*		
5	token	varchar(64)	utf8mb4_unicode_ci		No	*None*		
6	abilities	text	utf8mb4_unicode_ci		Yes	*NULL*		
7	last_used_at	timestamp			Yes	*NULL*		
8	created_at	timestamp			Yes	*NULL*		
9	updated_at	timestamp			Yes	*NULL*		

The personal_access_tokens table is used to store the authentication tokens of alumni. This table is consisting of 9 attributes.

***password_resets* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	email	varchar(255)	utf8mb4_unicode_ci		No	*None*		
2	token	varchar(255)	utf8mb4_unicode_ci		No	*None*		
3	created_at	timestamp			Yes	*NULL*		

password_resets table contains the details of password resets by the admin. This table is used in admin module. This table is consisting of 3 attributes.

***otp* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	otp	int(8)			No	*None*		
3	ref_no	varchar(50)	utf8mb4_general_ci		No	*None*		

otp table contains the otps sent to alumni to login / register to the alumni module. This table is consisting of 3 attributes.

***payments* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	payment_id	int(10)			No	*None*		AUTO_INCREMENT
2	payment_ref_id	varchar(100)	utf8mb4_general_ci		No	*None*		
3	alumni_id	int(9)			No	*None*		
4	amount	varchar(10)	utf8mb4_general_ci		No	*None*		
5	method	varchar(50)	utf8mb4_general_ci		No	*None*		
6	time	varchar(100)	utf8mb4_general_ci		No	*None*		
7	type	varchar(100)	utf8mb4_general_ci		No	*None*		

payments table keeps records of the payments made by the alumni for registration and donations. This table is consisting of 7 attributes.

***alumni_details* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	alumni_id	int(11)			No	*None*		AUTO_INCREMENT
2	user_id	int(8)			No	*None*		
3	firstname	varchar(100)	utf8mb4_general_ci		No	*None*		
4	middlename	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
5	lastname	varchar(100)	utf8mb4_general_ci		No	*None*		
6	gender	varchar(8)	utf8mb4_general_ci		No	*None*		
7	dob	varchar(20)	utf8mb4_general_ci		No	*None*		
8	registrationNo	varchar(255)	utf8mb4_general_ci		No	*None*		
9	roll_no	bigint(10)			No	*None*		
10	course	varchar(50)	utf8mb4_general_ci		No	*None*		
11	year_of_passing	varchar(4)	utf8mb4_general_ci		No	*None*		
12	address	text	utf8mb4_general_ci		No	*None*		
13	city	varchar(255)	utf8mb4_general_ci		No	*None*		
14	country	varchar(255)	utf8mb4_general_ci		No	*None*		
15	state	varchar(255)	utf8mb4_general_ci		No	*None*		
16	pin	int(8)			No	*None*		
17	image	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
18	created_date	date			No	current_timestamp()		
19	status	varchar(20)	utf8mb4_general_ci		No	submitted		

alumni_details table keeps the records of alumni profile registrations. This table is consisting of 19 attributes.

***occupations* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	occupation_id	int(8)			No	*None*		AUTO_INCREMENT
2	alumni_id	int(8)			No	*None*		
3	designation	varchar(255)	utf8mb4_general_ci		No	*None*		
4	organization	varchar(255)	utf8mb4_general_ci		No	*None*		
5	from_date	varchar(30)	utf8mb4_general_ci		No	*None*		
6	to_date	varchar(20)	utf8mb4_general_ci		No	*None*		

occupations table keeps the record of occupations of alumni. This table is consisting of 6 attributes.

***courses* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	course_id	int(10)			No	*None*		AUTO_INCREMENT
2	course_name	varchar(255)	utf8mb4_general_ci		No	*None*		
3	course_abbr	varchar(50)	utf8mb4_general_ci		No	*None*		

courses table keeps the records of courses available on the institute. This table is consisting of 3 attributes.

***notifications* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	notification_id	int(9)			No	*None*		AUTO_INCREMENT
2	notification	text	utf8mb4_general_ci		No	*None*		
3	type	varchar(10)	utf8mb4_general_ci		No	*None*		
4	date	date			No	current_timestamp()		

notification table contains the notifications which are sent to the users and general viewers. This table is consisting of 4 attributes.

***events* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	event_id	int(9)			No	*None*		AUTO_INCREMENT
2	title	varchar(255)	utf8mb4_general_ci		No	*None*		
3	content	text	utf8mb4_general_ci		No	*None*		
4	image	varchar(255)	utf8mb4_general_ci		No	*None*		
5	date	date			No	current_timestamp()		

events table contains the news or articles published on the portal by the admin. This table is consisting of 5 attributes.

***jobs* table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	queue	varchar(255)	utf8mb4_unicode_ci		No	*None*		
3	payload	longtext	utf8mb4_unicode_ci		No	*None*		
4	attempts	tinyint(3)		UNSIGNED	No	*None*		
5	reserved_at	int(10)		UNSIGNED	Yes	*NULL*		
6	available_at	int(10)		UNSIGNED	No	*None*		
7	created_at	int(10)		UNSIGNED	No	*None*		

This table is used to keep the records of the background tasks like sending email, notifications etc. The system will do all the jobs present here one by one in the background.

student_data table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_general_ci		No	*None*		
3	course	varchar(50)	utf8mb4_general_ci		No	*None*		
4	year_of_passing	varchar(6)	utf8mb4_general_ci		No	*None*		
5	reg_no	varchar(50)	utf8mb4_general_ci		No	*None*		
6	roll_no	int(20)			No	*None*		

This table contains student data of passed out students. Data can be imported from an excel sheet. These data help to verify the alumni profiles. If the data submitted by alumni is matched with the student data, then the alumni will be verified.

Chapter 9: Testing

‘Software testing is the process used to help identify the correctness, completeness, Security, and quality of developed computer software. Testing is a process of executing a program or application with the intent of finding errors with that in mind, testing can never completely establish the correctness of arbitrary computer software. Type of tests done in our project are as follows.

9.1 Acceptance testing:

User acceptance of a system is the key factor for the success of any system. It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not.

9.2 Compatibility Testing:

Compatibility testing is software testing is performed on an application to check its compatibility (running capability) on different platform/environments. This testing is done only when the application becomes stable. Means simply this compatibility test aims to check the developed software application functionality on various software, hardware platforms, network and browser etc. This compatibility testing is very important in product production and implementation point of view as it is performed to avoid future issues regarding compatibility.

9.3 System Testing:

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different type of tests with the purpose to exercise and examine the full working of an integrated software computer system against requirements.

9.4 Unit Testing:

Unit testing focuses on the verification of the smallest unit of software design using the unit plans prepared during the design phase of the system, error within the boundary of the module is uncovered in this testing phase, each and sub module were found to be working satisfactorily. In unit testing the entire system is divided into small part or units according to roughly. So many errors in a unit if any are debugged right there, thus saving the time. In this project all the views are tested to ensure that they operate correctly.

- In our project forms, views and reports are readable and abbreviations are avoided as much as possible.
- List boxes, radio buttons, check boxes are used so that users can easily select the available options.
- Proper validations are done so as to avoid any errors related to database.

9.5 Integration testing:

All modules were combined in this testing step. Then the program was tested as a whole. Using integrated test plans prepared in the design phase of the system development as a guide, the integration was carried out. All the errors found in the system were corrected. In our project

- All the modules are checked for correctness and made sure that is works in the way intended after integrating with one another.
- All the links are made sure to point at the right location.

9.6 Validation testing:

This test succeeds when the software function in a manner that can be reasonable expected by the user. After the validation test has been conducted one of the following two conditions exists.

- The function or the performance characteristics confirmed to the specifications and are accepted.
- A deviation from specification is uncovered and appropriate messages are given. This testing is used in our project to test each page, form, view and report is checked against the requirement document so as to ensure that it’s desired.

Chapter 10: Security Measures

Application security is the process of protecting websites, apps or online services against different security threats that exploit vulnerabilities in an application's code. Web application vulnerabilities are typically the result of a lack of input/output sanitization, which are often exploited to either manipulate source code or gain unauthorized access.

We have taken care of some common attack methods in this system.

- **SQL Injection:** It occurs when a perpetrator uses malicious SQL code to manipulate a backend database so it reveals information. Consequences include the unauthorized viewing of lists, deletion of tables and unauthorized administrative access. We have prevented the SQL injection in our project
- **Cross-site Request Forgery (CSRF):** It's caused when a malicious web application makes a user's browser perform an unwanted action in a site to which a user is logged on. CSRF token has been used to protect the system from CSRF attack.
- **Token based authentication:** Token-based authentication is a protocol which allows users to verify their identity, and in return receive a unique access token. During the life of the token, users then access the website that the token has been issued for, rather than having to re-enter credentials each time they go back to the same webpage or any resource protected with that same token. Users will get the token once the credentials are verified along with otp verification.

Chapter 11: Future Scope of The Project

The system can be modified or upgraded to any extent adding some extra features according to the organization's need at any time. The "Alumni Management System" can be further enhanced into an advanced system with adding some of the following features:

- We can add functionality to make the system available for the whole university.
- We can develop the frontend using Next.js which is framework build on top of react. By using Next.js the SEO will be better.
- It is also possible to develop a chat system for the alumni to stay connected with other alumni of the organization without sharing their personal credentials.

Chapter 12: Code

Alumni Authentication:

```
public function sendOTP(Request $request){
        $validator = Validator::make($request->all(), [
            "phone"     =>  "required | digits:10"
        ]);
        if ($validator->fails()) {
            return response()->json(['error'=>$validator->errors()], 400);
        }
        $phone = $request->phone;
        //generate OTP
        $otp_digit = rand(1231,9875);
        $ref_no = Str::random(30);
        $otp = new Otp;
        $otp->otp = $otp_digit;
        $otp->ref_no = $ref_no;
        $saved = $otp->save();
        $otp->id;

        $otp_tbl = DB::table('otp')->where('id', $otp->id)->first();
        $ref_no = $otp_tbl->ref_no;

        //check if user exists
        $user = User::where('phone', '=', $request->phone)->first();
        if ($user === null) {
            $isUserExists = 'false';
        } else {
            $isUserExists = 'true';
        }
        if($saved){
            //send otp
            $fields = array(
```

```
        "variables_values" => $otp_digit,
        "route" => "otp",
        "numbers" => $phone,
    );
    $curl = curl_init();
    curl_setopt_array($curl, array(
      CURLOPT_URL => "https://www.fast2sms.com/dev/bulkV2",
      CURLOPT_RETURNTRANSFER => true,
      CURLOPT_ENCODING => "",
      CURLOPT_MAXREDIRS => 10,
      CURLOPT_TIMEOUT => 30,
      CURLOPT_SSL_VERIFYHOST => 0,
      CURLOPT_SSL_VERIFYPEER => 0,
      CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
      CURLOPT_CUSTOMREQUEST => "POST",
      CURLOPT_POSTFIELDS => json_encode($fields),
      CURLOPT_HTTPHEADER => array(
        "authorization:
YOT8Bn1zaKDEchSPJsmAfZ6ij3qQNCe2ypoVlrFu9dGw50UbLRwcsqoF9HMNDgAB2leOTpnQKm50PE1v",
        "accept: */*",
        "cache-control: no-cache",
        "content-type: application/json"
      ),
    ));
    $response = curl_exec($curl);
    $err = curl_error($curl);
    curl_close($curl);
    //if otp saved into database, send the otp to user
    // SendLoginOtp::dispatch($phone, $otp_digit)
    //     ->delay(now()->addSeconds(1));
    return response()->json([
        'message'       => 'OTP Sent Successfully!',
        'phone'         => $request->phone,
        'isUserExists'  => $isUserExists,
```

```
                'ref_no'        => $ref_no
            ], 200);
        } else {
            return response()->json([
                'message'       => 'Error sending the OTP!',
                'phone'         => $request->phone
            ], 500);
        }
    }

    public function verifyOTP(Request $request){
        $validator = Validator::make($request->all(), [
            "otp"           =>  "required | digits:4",
            "ref_no"        =>  "required | string"
        ]);
        if ($validator->fails()) {
            return response()->json(['error'=>$validator->errors()], 400);
        }
        $isMatched = Otp::where('otp', $request->otp)->where('ref_no', $request->ref_no)-
>first();
        if($isMatched){
            return response()->json([
                'isVarified'    => true,
                'message' => 'OTP Verified'
            ], 200);
        } else {
            return response()->json([
                'isVarified'    => false,
                'message' => 'OTP Verification Failed!'
            ], 500);
        }
    }

    public function login(Request $request){
```

```
        $validator = Validator::make($request->all(), [
            "otp"           =>  "required | digits:4",
            "ref_no"        =>  "required | string",
            "phone"         =>  "required | digits:10",
        ]);

        $user = User::where('phone', $request->phone) -> first();
        $isMatched = Otp::where('otp', $request->otp)->where('ref_no', $request->ref_no)-
>first();

        if($isMatched && $user){
            $token = $user->createToken('myapptoken')->plainTextToken;
            return response()->json([
                'user'  => $user,
                'token' => $token
            ], 200);
        } else {
            return response()->json([], 401);
        }
    }

    public function register(Request $request){
        $validator = Validator::make($request->all(), [
            // "firstname"     =>  "required | string",
            "name"          =>  "required | string",
            "email"         =>  "required | email | unique:users",
            "phone"         =>  "required | digits:10 | unique:users",
        ]);
        if ($validator->fails()) {
            return response()->json(['error'=>$validator->errors()], 400);
        }
        $user = User::create([
            'name' => $request->name,
            // 'lastname' => $request->lastname,
```

```
            'email' => $request->email,
            'phone' => $request->phone,
        ]);
        $token = $user->createToken('myapptoken')->plainTextToken;
        return response()->json([
            'user' => $user,
            'token' => $token
        ], 201);
    }
```

Verify Payment:

```
public function verifyPayment(Request $request){
        $webhookSecret = "12345";
        $webhookSignature = $request->header('X-Razorpay-Signature');
        $webhookBody = $request->getContent();
        $json_body = json_decode($webhookBody, false);
        $expected_signature = hash_hmac('sha256', $webhookBody, $webhookSecret);

        if($expected_signature == $webhookSignature){
            $payment_ref_id = $json_body->payload->payment->entity->id;
            $email = $json_body->payload->payment->entity->email;
            $contact = $json_body->payload->payment->entity->contact;
            $amount = $json_body->payload->payment->entity->amount;
            $amount = $amount/100;
            $method = $json_body->payload->payment->entity->method;
            $type = $json_body->payload->payment->entity->description;
            $created_at = $json_body->payload->payment->entity->created_at;

            $user = User::where("email", $email)->first();
            $update = AlumniDetails::where("user_id", $user->id)->update(["status" =>
"Paid"]);
            $alumni_id = AlumniDetails::where("user_id", $user->id)->first();
            $payment = Payment::create([
                'alumni_id'         => $alumni_id->alumni_id,
```

```
                'payment_ref_id'    => $payment_ref_id,
                'amount'            => $amount,
                'method'            => $method,
                'time'              => $created_at,
                'type'              => $type
            ]);
        }
        return response()->json([
            'status' => "ok"
        ], 200);
    }
```

Send Push Notification:

```
protected $notification;
    public function __construct($notification)
    {
        $this->notification = $notification;
    }
    public function handle()
    {
        $client = new \GuzzleHttp\Client();
        $response = $client->request('POST', 'https://onesignal.com/api/v1/notifications',
[
            'body' => '{"included_segments":["Subscribed Users"],"contents":{"en":"'.$this-
>notification.'"},"name":"INTERNAL_CAMPAIGN_NAME","app_id":"349ed92b-68ca-47ec-97f7-
9759a8a3e937"}',
            'headers' => [
              'Accept' => 'application/json',
              'Authorization' => 'Bearer OTA1YmI5ODYtMWQ3NC00YzFhLTg0MTEtZGQ4NjE2N2U3NjA5',
              'Content-Type' => 'application/json',
            ],
          ]);
    }
```

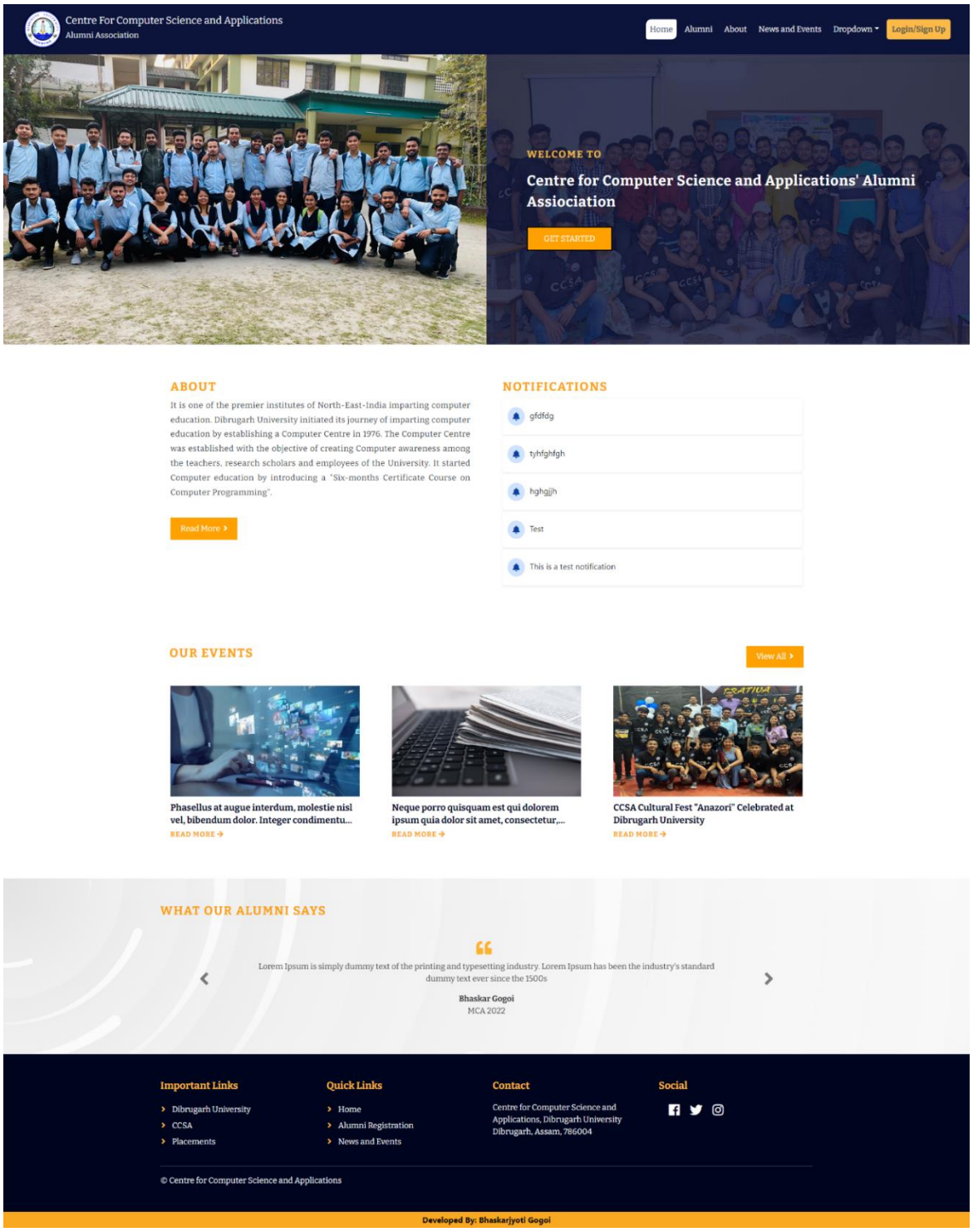
Import Excel Sheet:

```
<?php
namespace App\Imports;
use App\Models\Student;
use Maatwebsite\Excel\Concerns\ToModel;
use Maatwebsite\Excel\Concerns\WithHeadingRow;
class StudentData implements ToModel, WithHeadingRow
{
    public function model(array $row)
    {
        return new Student([
            'name' => $row['name'],
            'course' => $row['course'],
            'year_of_passing' => $row['year_of_passing'],
            'reg_no' => $row['reg_no'],
            'roll_no' => $row['roll_no'],
        ]);
    }
}
class ImportController extends Controller
{
    public function viewImportForm(){
        return view('import-student-data');
    }
    public function import(Request $request)
    {
        Excel::import(new StudentData, $request->file);
        return redirect('import-student-data')->with('status', 'Data Imported
        Successfully!');
    }
}
```

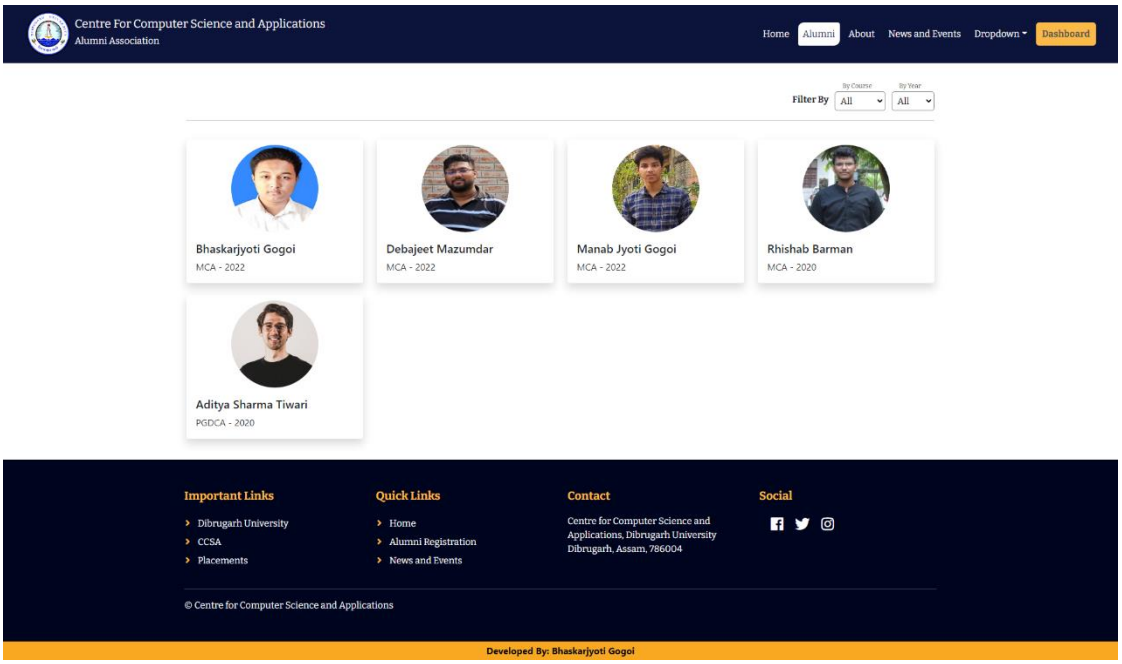
Chapter 13: Output

13.1 General Module Screenshots:

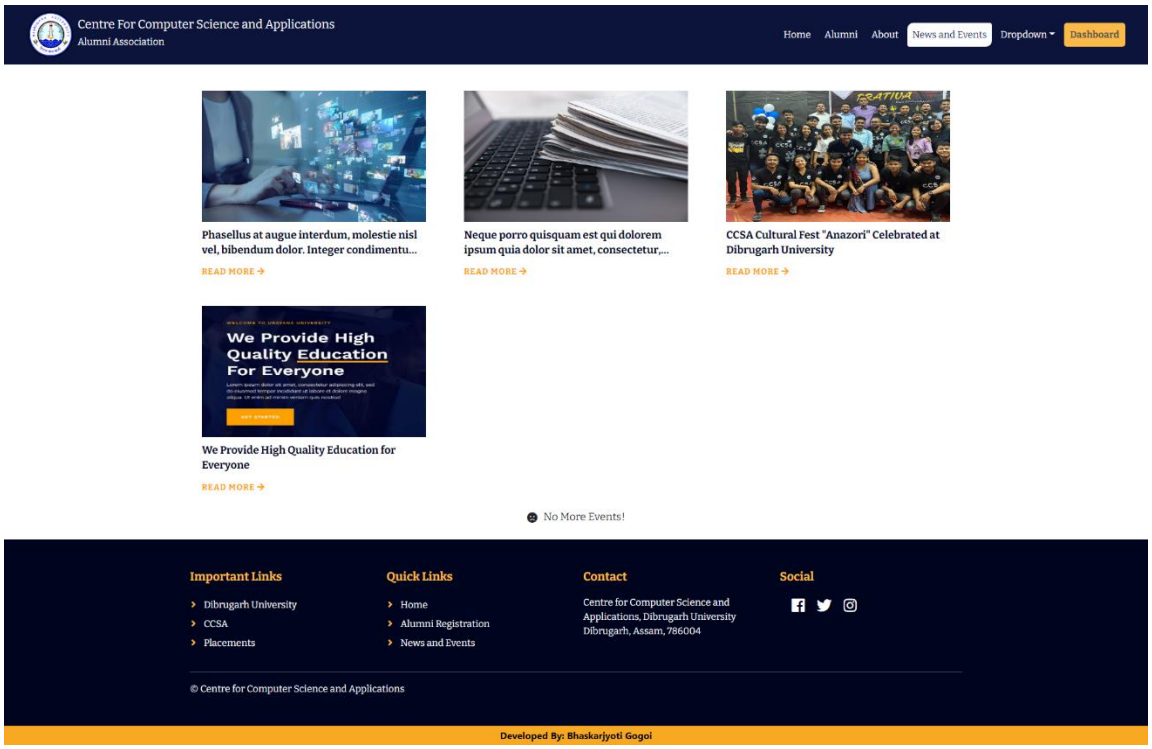
Home Page:



Alumni Page:

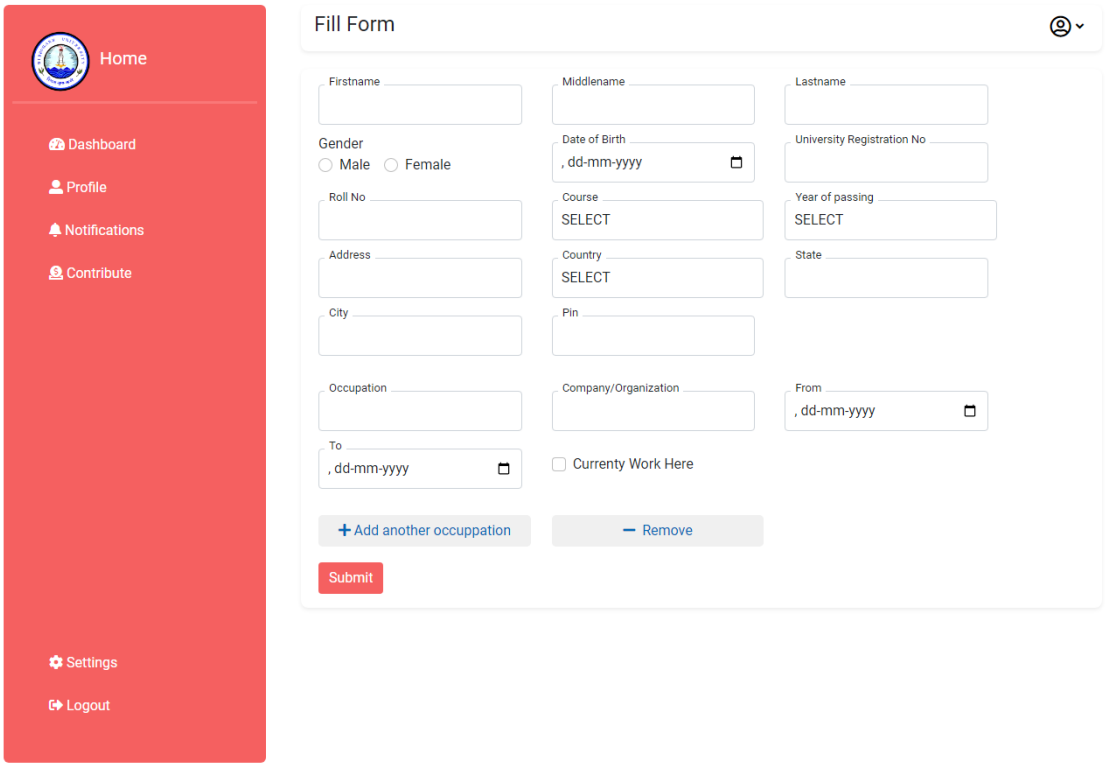


News & Events Page:

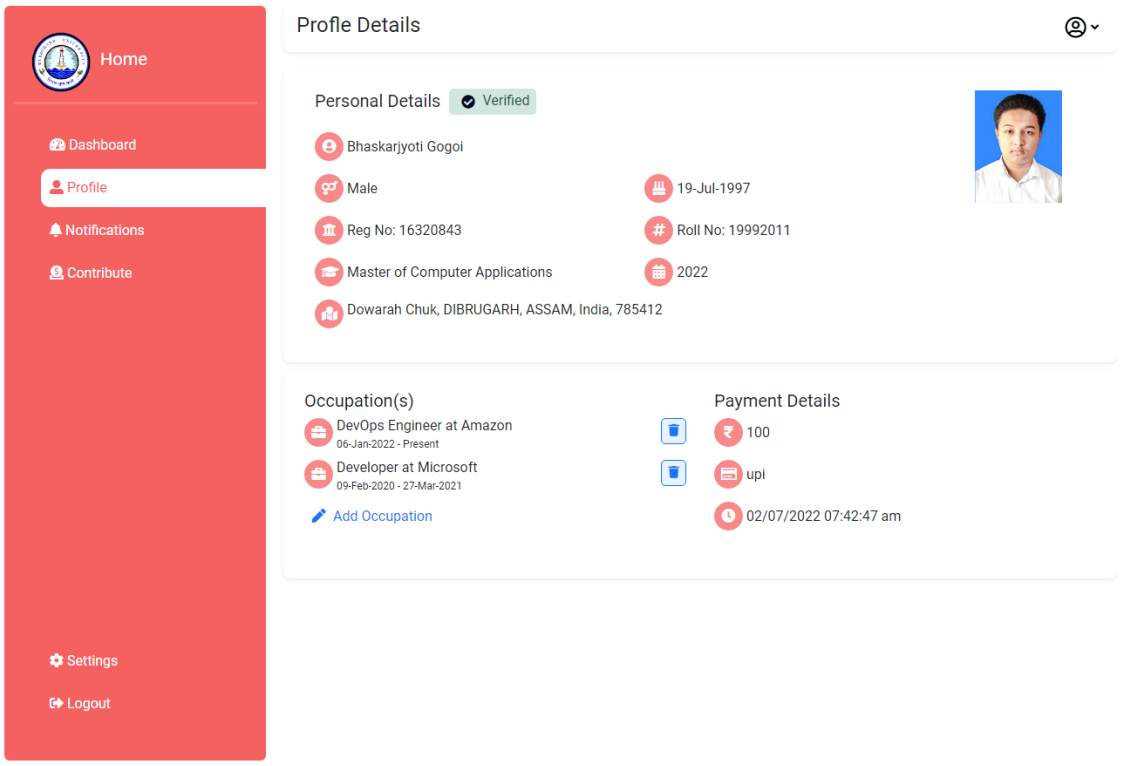


13.2 User Module Screenshots

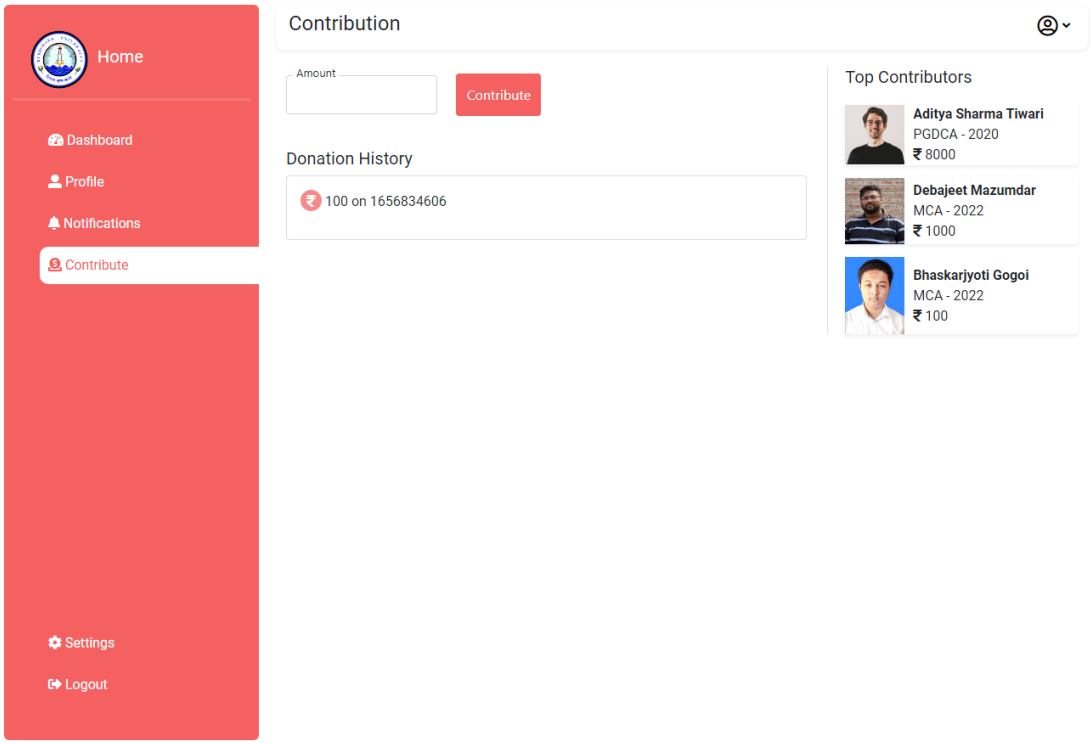
Create Profile Page:



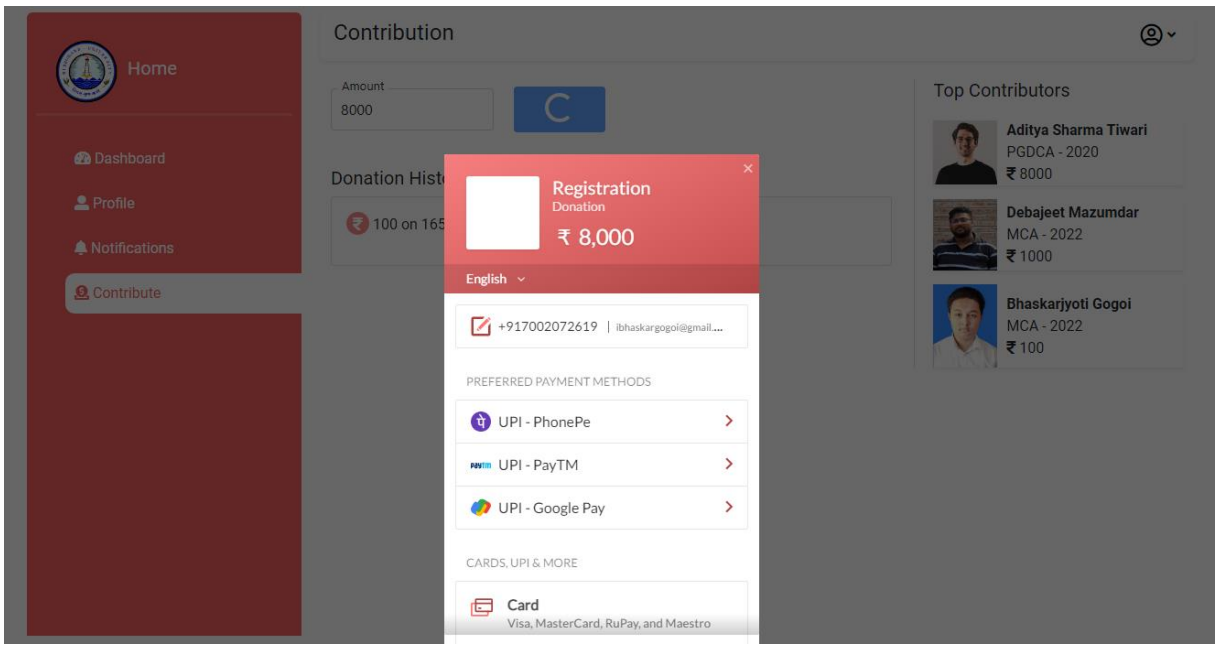
Profile Page:



Contributions Page:

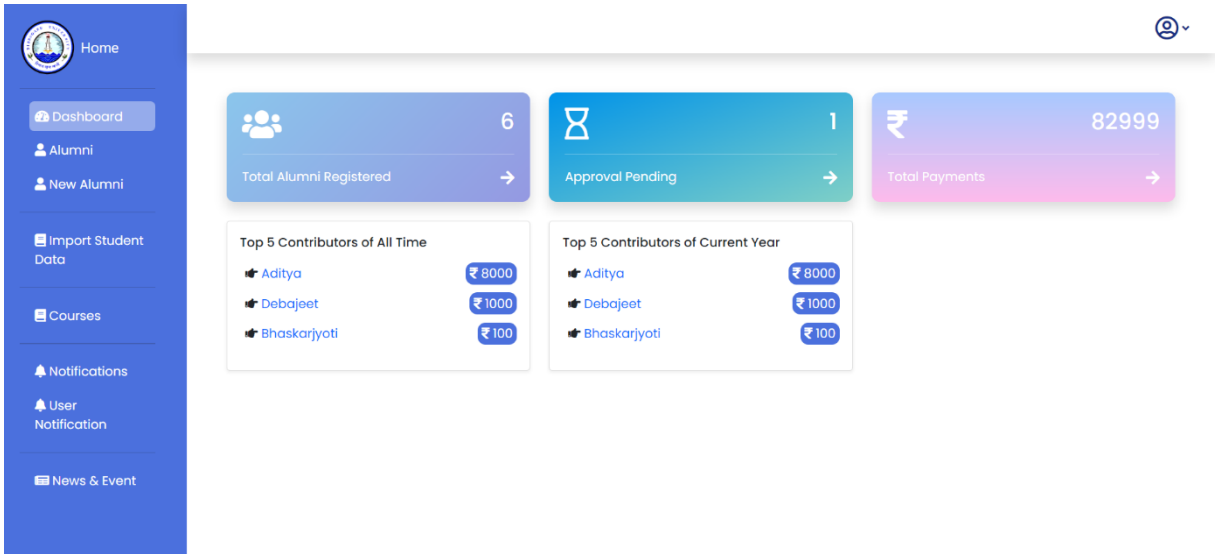


Payment Page:

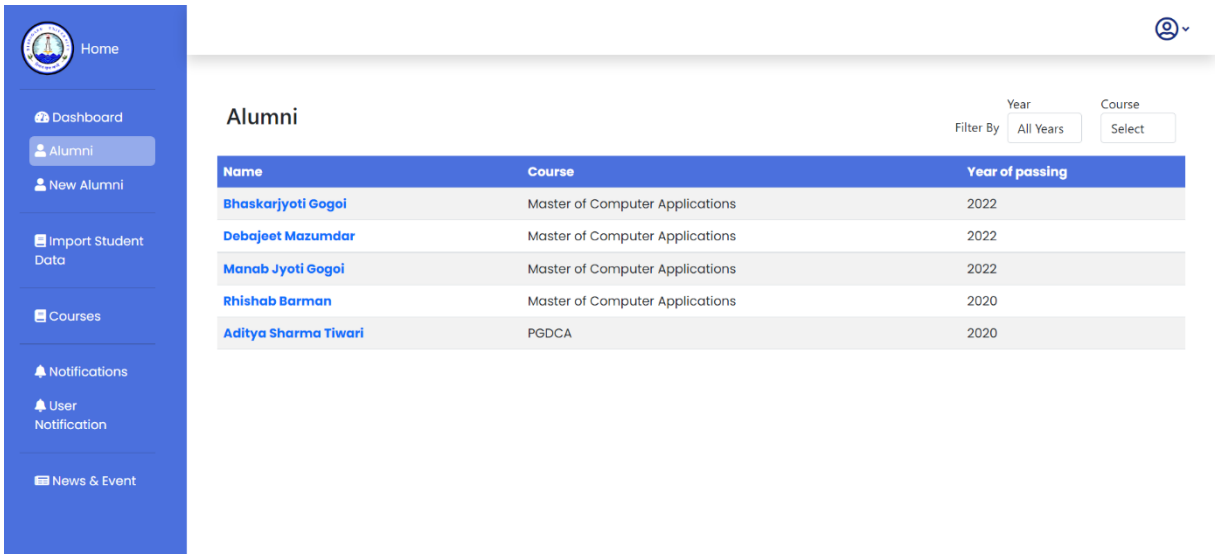


13.3 Admin Module Screenshots:

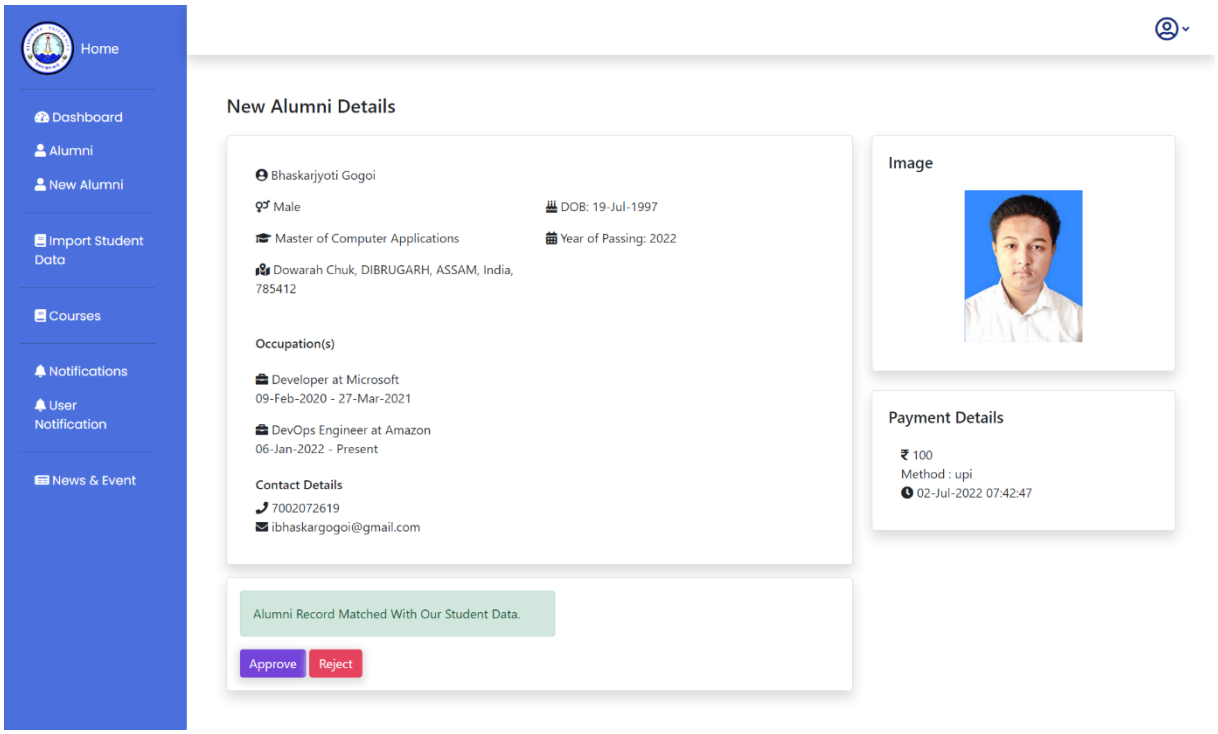
Admin Dashboard Page:



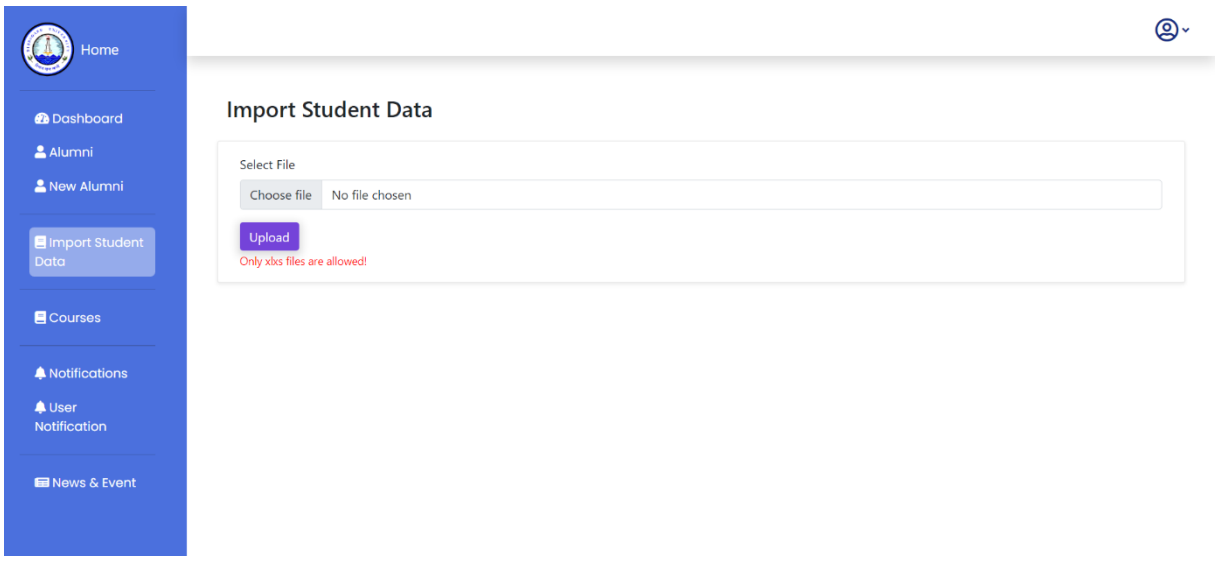
Alumni Page:



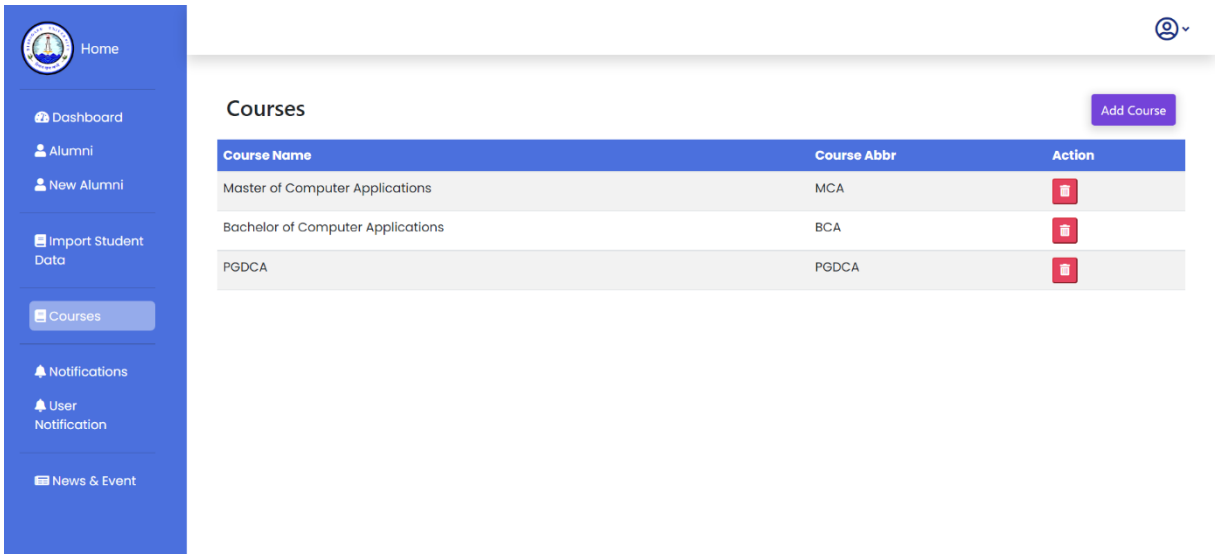
Alumni Details Page:



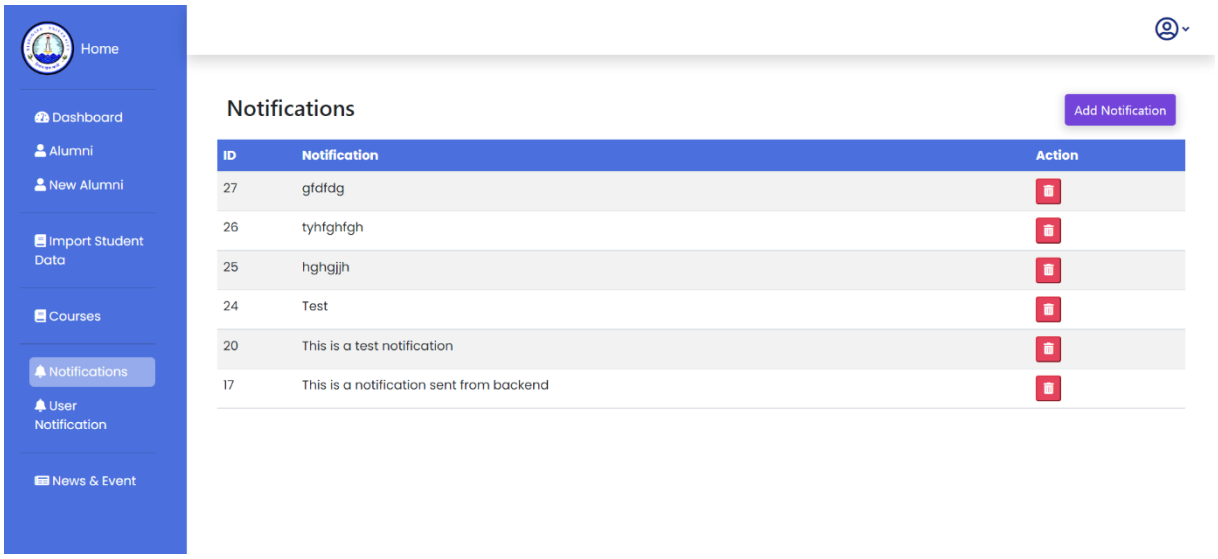
Import Student Data Page:



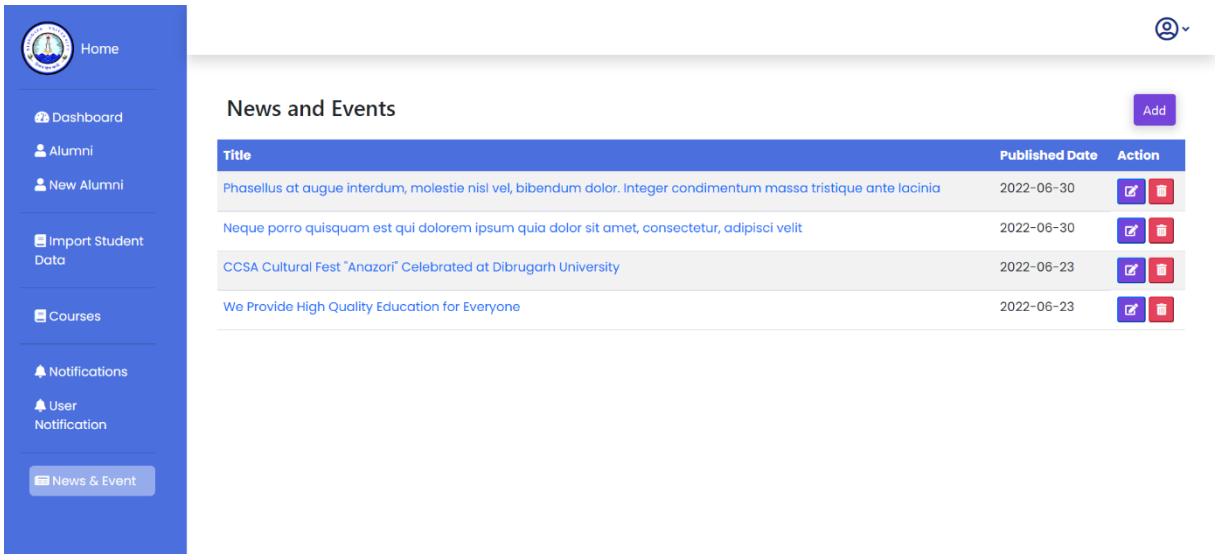
Courses Page:



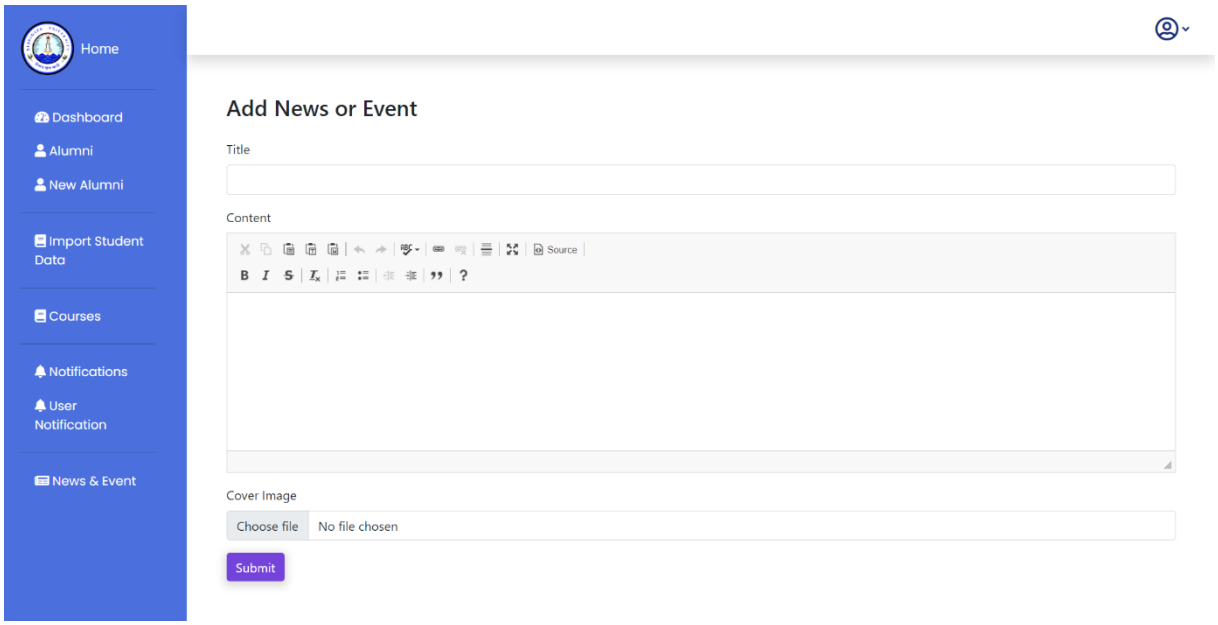
Notifications Page:



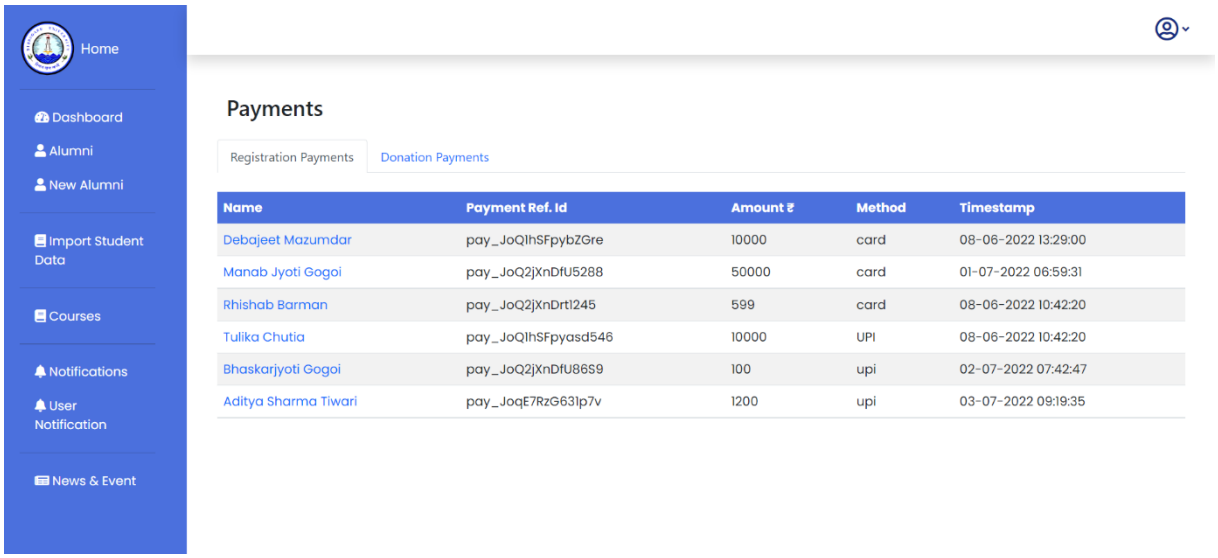
News and Events Page:



Add News Page:



Payment Details Page:



Chapter 14: Conclusion

The application for alumni project is used to store the former student's detail of the college and students can get the information from institute through the web page easily at any time. So, this project is very useful to both university and students. During this project we learnt more about Laravel, react, HTML, JavaScript, and CSS, PHP programming language and we got technical knowledge a lot. This project done by me for academic purpose only. From the beginning of this project, we planned about usage of the system, gradually we implemented on that idea.