



Pamantasan ng Lungsod ng Pasig

12 Alcalde St. Brgy. Kapasigan, Pasig City

College of Computer Studies



**ALUMBACK: EMPOWERING PLP ALUMNI TO COMEBACK, RECONNECT,
AND SHINE BRIGHTER – WHERE MEMORIES REIGNITE AND FUTURES**

TAKE FLIGHT

*In partial fulfillment of the requirement for the subject of CS403, dubbed as Open-Source
Programming with Database*

SUBMITTED BY:

DELA CRUZ, JHEIZON BRHYLLE P.

FRAGA, HARLENE P.

GODES, PAUL ADRIAN

TAJARROS, RITCHMOND JAMES S.

SUBMITTED TO:

PROF. MICHAEL MARASIGAN

TABLE OF CONTENTS:**Chapter I**

Background of the Study	3
--------------------------------------	----------

Chapter II

Project Objectives, Goal And Scope	6
---	----------

Chapter III

System Overview	8
------------------------------	----------

Chapter IV

Code Design & Structure	13
--	-----------

Chapter V

Database Structure	32
---------------------------------	-----------

Chapter VI

Outputs	35
----------------------	-----------

Chapter I

BACKGROUND OF THE STUDY

After thorough researching about Pamantasan ng Lungsod ng Pasig's (PLP's) background the researchers have collated their findings thus the section below is documented in a narrative format.



Figure 1: The logo of proponents targeted organizational entity.

PAMANTASAN NG LUNGSOD NG PASIG (PLP)

The PLP, also known as the Pasig City University, was established on March 15, 1999, through the Ordinance No. 11, Series of 1999, and was passed by the Sangguniang Panlungsod ng Pasig (City Council of Pasig). The authority for its establishment was granted by the Local Government Code of 1991, allowing Local Government Units (LGUs) to establish and operate educational institutions.

PLP was conceived as part of the Pasig City Development Plan of the year 2000, aiming to respond to the challenges of the new millennium. The entity is committed to providing accessible and affordable higher education to the residents of Pasig City and the surrounding areas.

The university's campus extends throughout Pasig City, utilizing modern facilities available within the city, including libraries, computer centers, sports complexes, and more, to offer quality education and improve the lives of Pasigueños.

Over the years, PLP has been led by various Presidents and Officers-in-Charge, with each respective contribution to the institution's growth and development. Notably, Dr. Amihan April M. Alcazar, a Ph.D., was appointed as the fourth University President in 2012, further solidifying PLP's commitment to educational excellence.

The establishment of PLP represents Pasig City's dedication to providing accessible and quality higher education, empowering its residents to pursue academic and career goals while contributing to the development of the local community.

The university's core values including both their sworn vision and mission can be interpreted below respectively.

PLP'S CORE VALUES:

- Academic Excellence
- Cultural Identity
- Peace Education
- Sustainable Development
- Social Equity and Responsibility
- Global Competitiveness

MISSION:

We, a community of service-oriented individuals, supported by the City Government of Pasig, are committed to lifelong learning and to produce graduates strong in their global outlook, cultural identity, and social responsibility through teaching strategies, methodologies, relevant research and dedicated public service.

VISION:

The Pamantasan ng Lungsod ng Pasig is a leading center for academic excellence among locally funded colleges and universities that produces responsible and productive individuals who are responsive to the changing demands of development locally and globally.

Chapter II

PROJECT OBJECTIVES, GOAL AND SCOPE

This chapter will tackle case study's scope, goal, and objectives which will serve as a foundation for their alumni system's development, which are about to unfold at Chapter #.

Project Goal: The main goal of the PLP Alumni System is to create an easy-to-use online platform/page under the PLP's main website. These interconnected webpages, was made to help graduates of Pamantasan ng Lungsod ng Pasig (PLP) stay connected with each other, find new job opportunities, and keep in touch with their old university even after their residency period expires. The researchers want to make it easier for former students to support, and be involved with the university community, helping both sides grow, and eventually, stay connected.

Scope: The project will involve constructing a comprehensive website equipped with an authentication system for alumni registration and login. This site will serve, not only as a networking hub, but also as a resource center, offering updated personal information management, university news, and event participation options. A key aspect of the system is the maintenance of data security and privacy through a localized database system.

The objectives of the PLP Alumni System are:

- To establish a secure and efficient login and registration process, ensuring alumni data privacy and proper access control.
- To provide a platform for alumni to effortlessly update their profiles, stay connected, and access university-related information.
- To facilitate easy submission and processing of alumni inquiries, ensuring timely and effective communication.
- To integrate real-time date and time displays for keeping alumni informed about important university events and deadlines.
- To enhance the user experience through the implementation of cookies and sessions, personalized preferences, and remembering login details.

Chapter III

SYSTEM OVERVIEW (DATA FLOW DIAGRAM or DFD)

This chapter will discuss how certain distinct modules interacts with each other, which will serve as a guide not only for the proponents, but also, stakeholders for them to easily comprehend system processes, and furtherly identify data sources and destinations, and serve as an effective communication tool among technical and non-technical individuals.

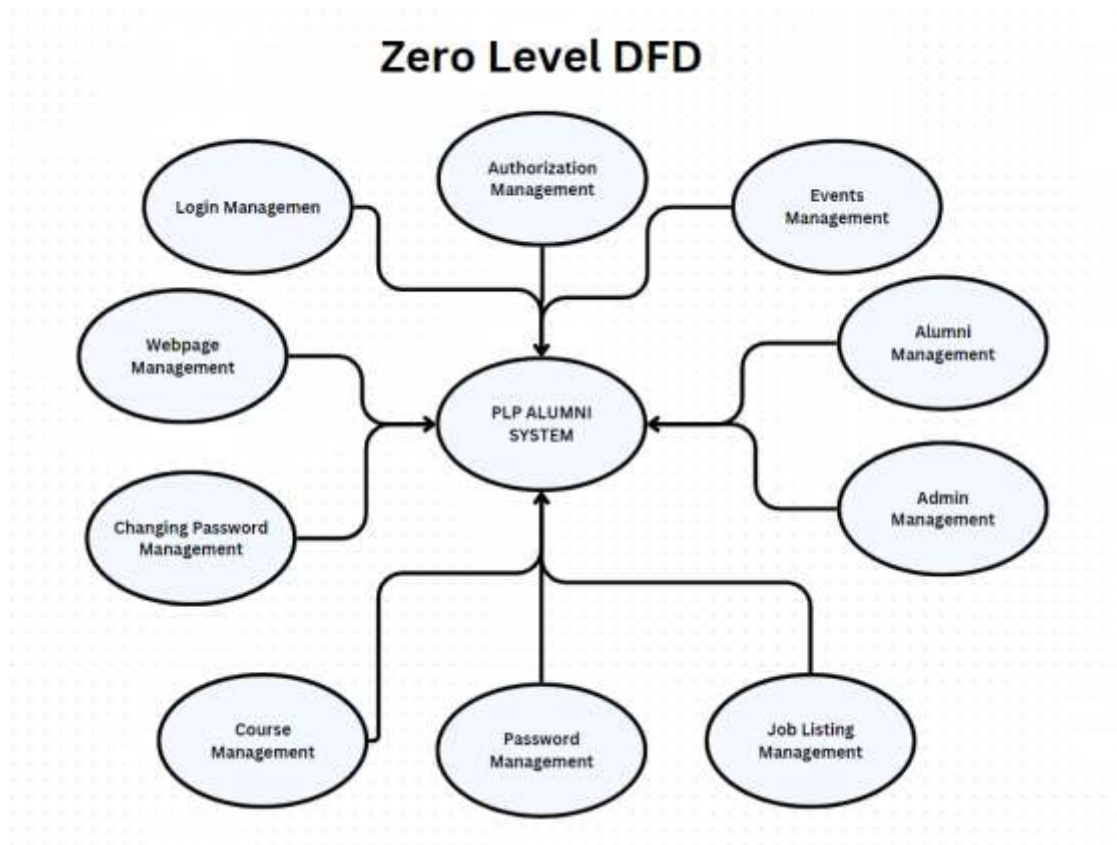


Figure 2: Default Level of Data Flow Diagram DFD

This is a high-level (level zero) overview of our "PLP ALUMNI SYSTEM." Around this central or main system, are several processes interconnected, which indicates the flow of data (or interaction) between these subprocesses (submodules) towards the main system. Each

process is represented as an oval, and the lines connecting them to the central system represent the interaction or data flow between the processes and the PLP Alumni System.

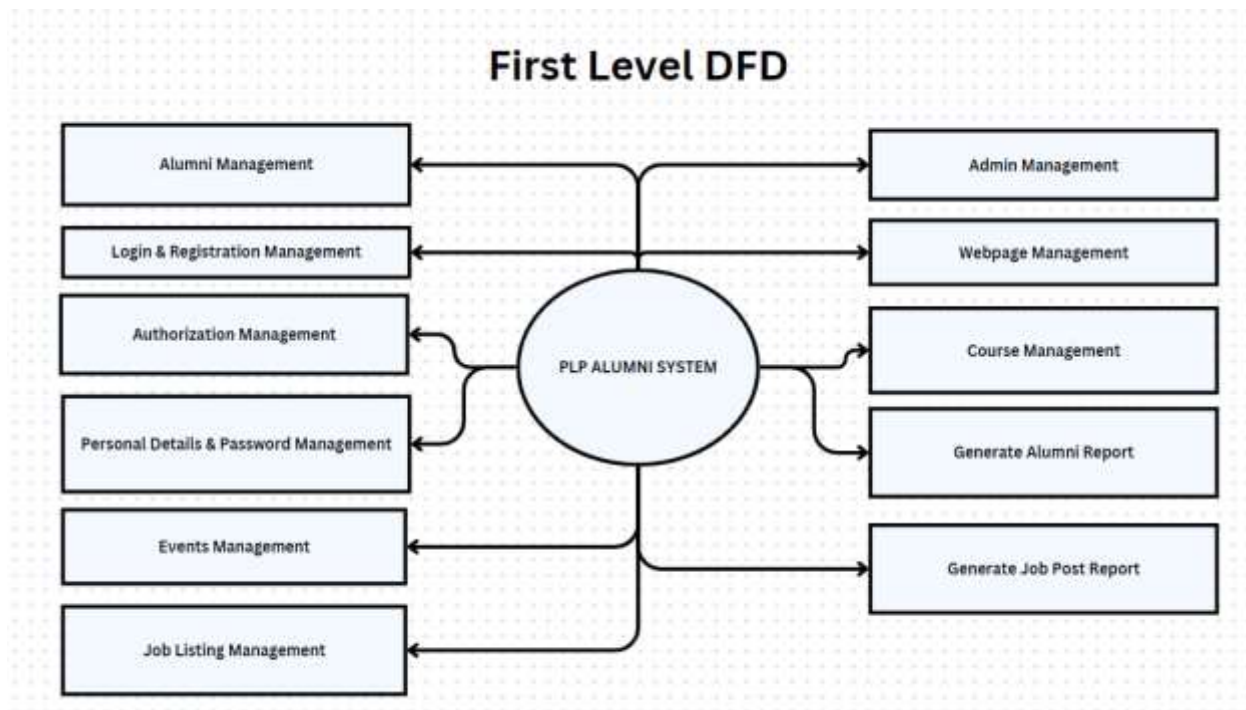


Figure 3: An Exploded Level of the Previous DFD

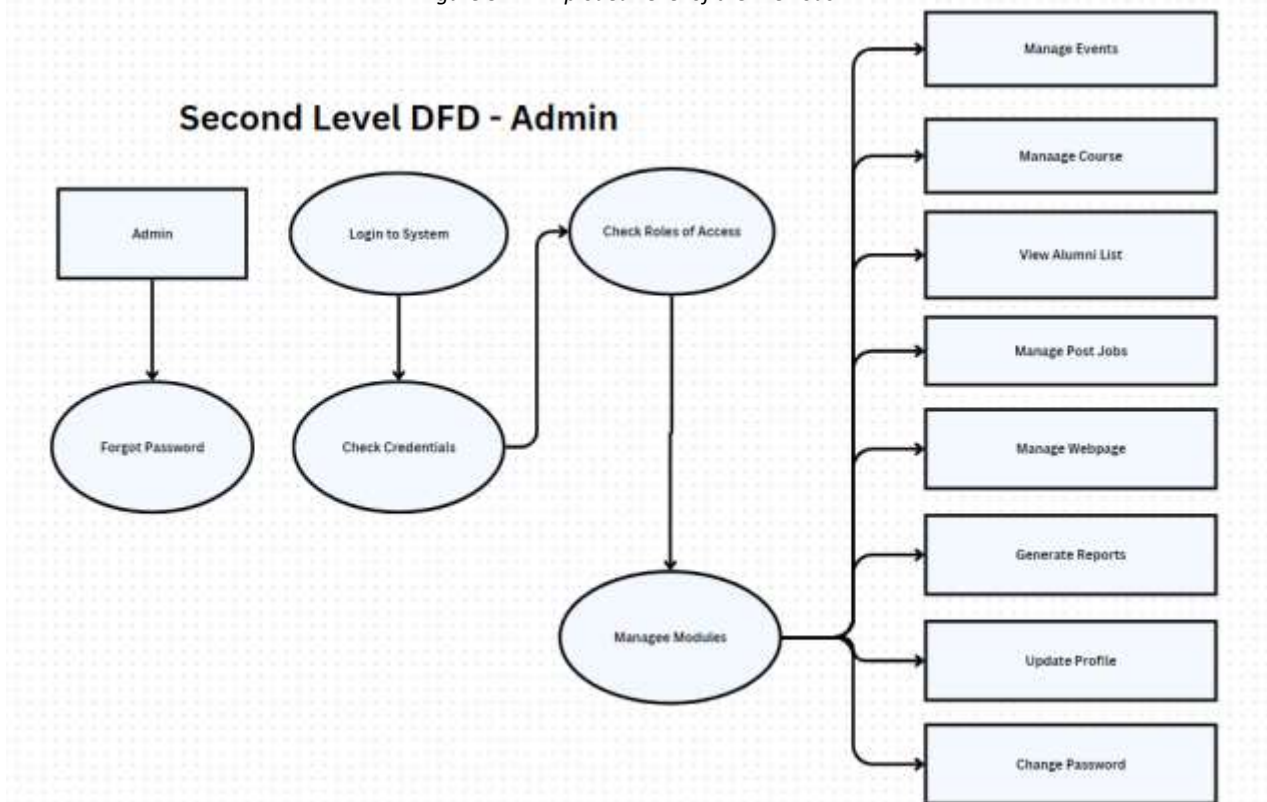
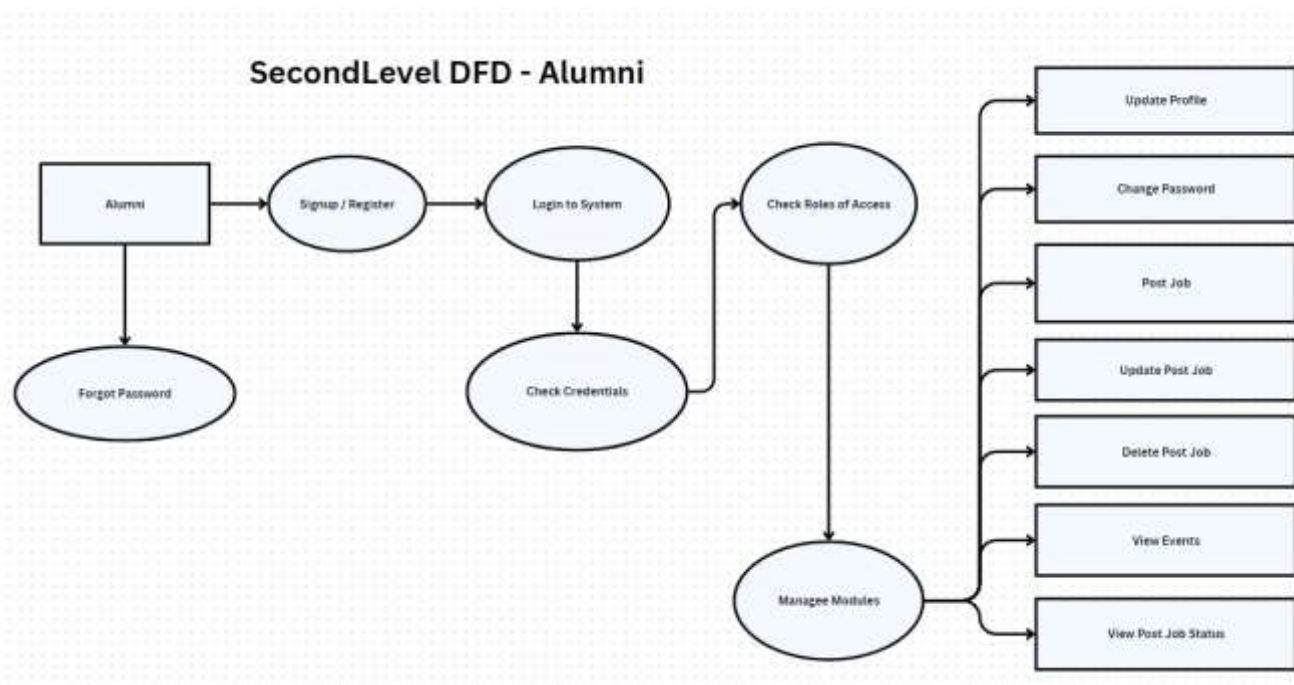


Figure 4: This is the Exploded Level of Admin Management

The provided diagram at Figure 4, depicts an expanded view of the first level within the admin module of a system, offering a comprehensive breakdown of processes and sub-processes. Positioned at the top left is the "Admin" rectangle, representing the user or role engaging with the system. The initial step involves the "Login to System" process, outlining the procedure for the admin to access the system. A linked subprocess, labeled "Forgot Password," hints at a password recovery feature associated with the login process. Following this, the "Check Credentials" process is depicted, indicating a verification step where the system validates the login details provided by the admin. Connected to this, the "Check Roles of Access" process suggests a subsequent step wherein the system examines the admin's permissions post-login. The central process oval, labeled "Manage Modules," is connected to "Check Roles of Access" and symbolizes the admin's capability to access and control various system modules based on their assigned role. Additionally, each rectangle connected to "Manage Modules" signifies the diverse modules or functions that the admin can manage within the system, demonstrating a streamlined flow from login authentication to the execution of specific administrative tasks.



Above is DFD for the alumni component of the system that depicts a user-centric process. The "Alumni" rectangle represents the alumni users, initiating the system through the "Signup/Register" process for account creation. An additional feature, "Forgot Password," allows alumni to recover their passwords if needed. Subsequently, the "Login to System" process enables access to accounts, followed by the verification of login details in the "Check Credentials" step. The system further evaluates permissions through the "Check Roles of Access" process. A central process, "Manage Modules", is connected to "Check Roles of Access," symbolizing the diverse actions alumni can perform within the system. On the right, various rectangles denote specific functions available to alumni, including updating profiles, changing passwords, posting, and managing job listings, and accessing event information. Each of these functions is intricately linked to the "Manage Modules" process, outlining a user journey from account creation to the management of alumni-specific tasks and functionalities within the system.

Chapter IV

CODE DESIGN & STRUCTURE

The researchers' code contains at least three (3) folders that will divide the system into distinct major interfaces. The philosophy of the halving was to better abstract the information that should only show within an authorized level, if an end-user is merely a passerby from an internet, a public page face should only show the public-level information. The first scripts listed inside each folder are the main modules that will become the driver for the submodules or fragments, inside the "includes" folder (which every major interface is expected to contain). This technique is known as modular programming and was done for better handling of modifications or debugging when the time calls for it.

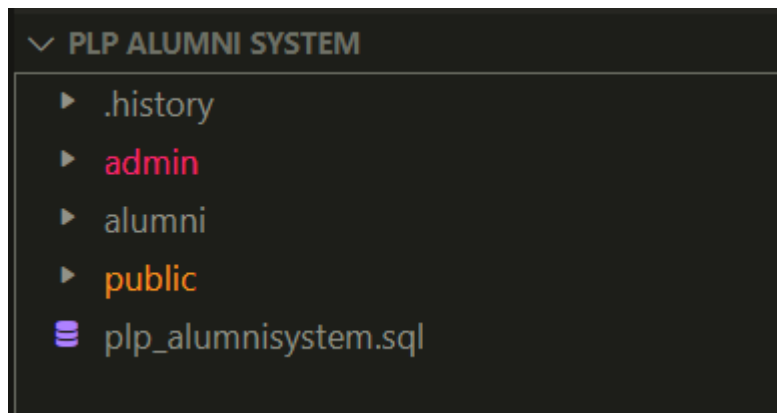


Figure 5: The Illustration of File Package Structuring.

A. Public Folder (this is the forefront of our website): Everyone on the internet can see this.

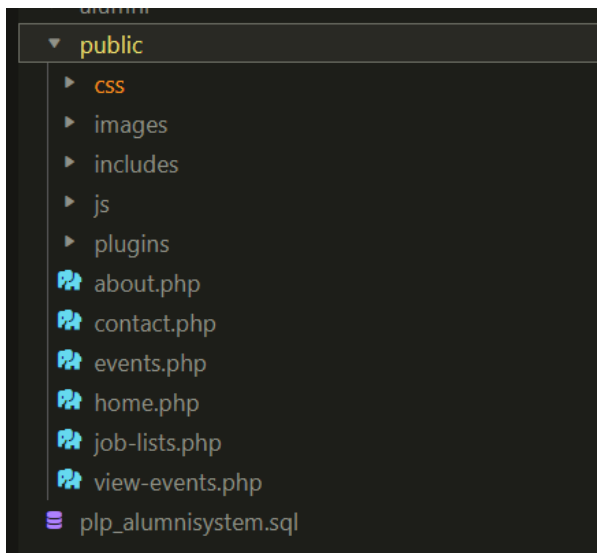


Figure 6: The Illustration of the Insides of Public Folder.

1. **home.php** (In a nutshell, this should be the entry-point of our website): This PHP code generates a dynamic web page for the PLP Alumni, featuring a responsive layout with a home slider introducing the institution and upcoming events. The page fetches event details from the database, presenting them with titles, schedules, and brief descriptions in a visually appealing format. Additionally, it includes header and

footer sections for consistency, incorporates JavaScript functionalities for interactivity, and connects to external stylesheets for enhanced styling.

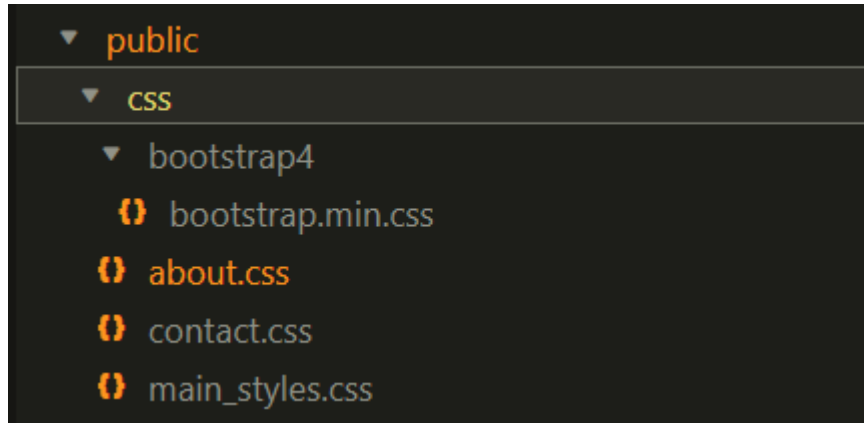
2. **about.php**: While this PHP script generates a dynamic web page for the Pamantasan ng Lungsod ng Pasig Alumni, focusing on the "About Us" section. The page includes information about the alumni association, highlighting the institution's pride in its diverse community and the achievements of its graduates. It introduces the alumni association officers with their respective roles, accompanied by profile images. The page also features a video section showcasing a heartwarming alumni story and an "Achievements" section, with placeholder content for future updates. The layout is structured, incorporating external stylesheets and JavaScript for enhanced styling and interactivity. The footer section maintains consistency throughout the website.

3. **contact.php:** Now, this script superficially generates a web page for the PLP Alumni Contact Information. The page includes a header, contact information section, and a footer. The contact information is dynamically retrieved from the database using PHP and includes the address, contact number, and email. The Google Maps API frame displays the location of the institution. While the page is styled using Bootstrap and custom CSS and includes JavaScript for handling contact-related functionalities. The layout follows a responsive design, ensuring a user-friendly experience on different devices.
4. **events.php** (TLDR: generates a web page for listing upcoming events): The PHP script will generate a page that includes a header, breadcrumbs, a section for upcoming events with pagination, and a footer. The events are fetched from a database using PDO (PHP Data Objects) for interaction with the database. The events are displayed in a responsive layout, each with an image, title, schedule, and a brief description. Pagination is implemented to display a limited number of events per page. The design is styled using Bootstrap and custom CSS, with JavaScript for handling scroll events. The page is part of a larger web application for managing and displaying events related to the Pamantasan ng Lungsod ng Pasig.
5. **job-lists.php** (TLDR: this script is responsible for the generation of a webpage for displaying approved job listings from alumnus): While this page includes a header, a container for job details, and a footer. The job details are presented in a table, showing information such as job title, company name, location, vacancy, designation, job description, last date, posting date, contact number, and contact person. Pagination is implemented to display a limited number of job postings per page, with navigation links to move between pages. The design

is styled using Bootstrap and custom CSS, and JavaScript is included for handling specific page functionalities. The page is part of a larger web application for managing and viewing job postings related to the PLP.

6. **view-events.php** (TLDR: A script that generates a web page for viewing details of a specific event): Last among the “public folder”, is the view-events.php that includes a header, breadcrumbs, and a container for displaying event details. The event details consist of the event banner image, event title, description, date, and time. The design is styled using Bootstrap and custom CSS. The page is part of a college alumni system for managing and displaying information related to events at the PLP.

While these are the Cascading Style Sheets (CSS) used within the interface of public content (folder), the CSS files are encapsulated within a folder for organization or better hierarchical structure of file packaging.



Without these three (3) CSS, the landing page for the Alumni Webpage is merely a clean, and white space. Below are the JS Files.

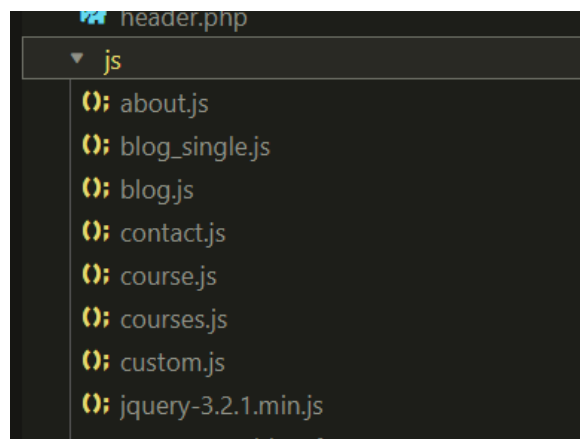


Figure 7: JS Files Under Admin Folder

B. Alumni Interface (Folder)

1. **add-job.php** – The PHP code responsible for an "Add Jobs" page in a College Alumni System. It begins with session handling and database connection inclusion. The code checks whether a user is logged in; if not, it redirects them to the logout page. If the user is logged in, the code

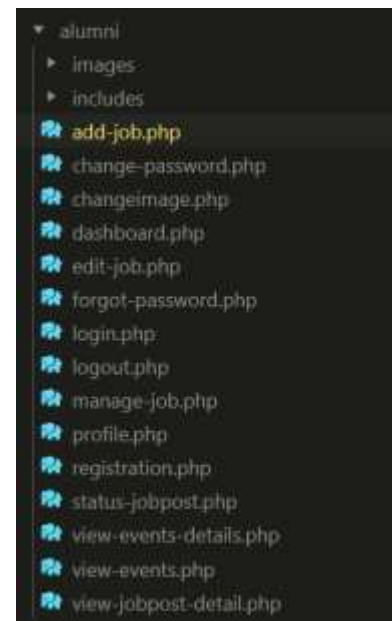


Figure 8: PHP Scripts Under Alumni

processes form submissions for adding job postings. It captures job details such as title, company name, location, vacancy, designation, description, contact person, contact number, and the last date for application. This information is then inserted into the database, and the user receives corresponding success or error messages. The HTML part of the code includes form elements for entering job details, and it follows a standard structure with header, sidebar, and footer sections typical of a web application. The page includes various CSS and JavaScript dependencies for styling and functionality.

2. **change-password.php** – is responsible for a "Change Password" page in the College Alumni System. It includes session handling, error reporting, and database connection. The code checks whether a user is logged in, and if not, redirects them to the logout page. If the user is logged in, the code processes form submissions for changing the password. It validates the current password, and if correct, updates the password in the database. The HTML part of the code includes form elements for entering the current password, new password, and confirming the new password. It also includes CSS and JavaScript dependencies for styling and functionality. The page structure follows a standard format with a header, sidebar, and footer sections typical of a web application.
3. **changeimage.php** – is responsible for updating the profile picture of an alumni in a College Alumni System. It includes session handling, error reporting, database connection, and file upload functionality. The code first checks if the alumni is logged in; if not, it redirects to the logout page. If logged in, it processes the form submission for updating the profile picture. The HTML part of the code includes a form that allows the user to upload a new profile picture. It displays the current profile picture and the full name of the alumni. Upon submission, the new picture is uploaded, and the database is updated. The page

structure follows a standard format with a header, sidebar, and footer sections typical of a web application.

- 4. dashboard.php** – This script represents a dashboard page for the College Alumni System. It starts by checking if the alumni user is logged in; if not, it redirects to the logout page. The dashboard displays various statistics related to events and job postings. It includes the total number of events and job postings, along with links to view details for each category. The layout is structured with a sidebar, header, and footer, following a typical web application structure. The dashboard provides a quick overview of essential information for the logged-in alumni user.
- 5. edit-job.php** – This code is responsible for updating job details in the College Alumni System. It first checks if the alumni user is logged in; otherwise, it redirects to the logout page. The page allows alumni users to edit and update job postings. The form is pre-populated with existing job details fetched from the database, and users can modify the information. Upon submission, the updated job details are stored in the database. The layout includes a sidebar, header, and footer, following a typical web application structure. The page provides a user-friendly interface for alumni to manage and edit their posted job listings.
- 6. forgot-password.php** – A script that is primarily responsible for the "Forgot Password" feature in the College Alumni System. When a user submits their email address through the form, it checks if the email exists in the alumni database. If the email is valid, it updates the password for that email address with the new password provided in the form. The page includes form validation using JavaScript to ensure that the new password matches the confirmation password. The design is part of a login system, displaying a form with fields for email, new password, and confirmation password. If the password change is

successful, a success message is displayed; otherwise, an error message is shown. The page also provides a link to the login page.

7. **login.php** – While this script is responsible for the login functionality of the PLP Alumni System. As it checks the submitted email and password against the database, and if the credentials are valid, it sets up a session for the user and redirects to the dashboard. Additionally, it provides an option for users to remember their login credentials using cookies. The page includes a form with fields for email and password, a "Remember Me" checkbox, and links for forgotten passwords, the home page, and registration. The design is part of a login system, displaying a form with relevant fields. If the login fails, it shows an "Invalid Details" alert. If successful, it sets the session, remembers the user with cookies, and redirects to the dashboard.
8. **logout.php** – Inversely, this script is responsible for the logout functionality of the Alumni System, as it, in background, starts the session, unsets all session variables, destroys the session, and then redirects the user to the login page. This ensures that the user is logged out and their session is terminated.
9. **manage-job.php** – This script is a part of a web application for managing job posts. Because it includes functionality for deleting job posts, displaying a paginated list of job posts, and providing options to edit or delete each post. The code retrieves job posts from the database based on the logged-in alumni's ID and displays them in a table. It also implements pagination to manage many records. The code checks the alumni's session, ensuring that they are logged in; otherwise, it redirects them to the logout page. If the user attempts to delete a job post, it handles the deletion by accepting the post's ID through the GET method. Additionally, the code implements pagination for a better user experience when viewing a large number of job posts.

10. profile.php – While this script is responsible for representing a web page for viewing and updating the profile details of a college alumni. It begins by checking the alumni's session, ensuring they are logged in; otherwise, it redirects to the logout page. The code handles form submissions for updating profile information, including fields like full name, gender, batch, course graduated, and current connection details. It retrieves the alumni's information from the database and populates the form fields with the existing data. The alumni can update their profile details by submitting the form, and the changes are reflected in the database. The code also includes HTML and CSS for the page layout and styling, with Bootstrap for responsiveness and various JavaScript libraries for additional functionality. Additionally, the page provides options for editing the alumni's profile picture and includes a link to change the image.

11. registration.php – represents the script that handles the registration form which collects various details, including full name, college ID, gender, batch, course graduated, current connection details, email, and password. The user is also prompted to upload a profile picture. The code performs validation checks, ensuring that the provided email and college ID do not already exist in the database. If the input is valid, the information is inserted into the database after encrypting the password and storing the profile picture in a designated folder. Success or failure messages are displayed accordingly. The HTML structure, along with Bootstrap styling, facilitates a user-friendly interface. Overall, the code handles alumni registration securely and includes necessary form validations.

12. status-jobpost.php – While this page represents a page for managing job posts in a college alumni system. The page includes functionalities such as listing job posts, displaying details, and providing pagination. Key features include the ability to view job details, delete job posts, and navigate through paginated results. The researchers uses

Bootstrap for styling and responsiveness. It interacts with a database to retrieve and display job post information. It also incorporates a pagination system to manage a large number of job posts efficiently. A session check is performed at the beginning to ensure that the user is logged in. If not, the user is redirected to the logout page. Additionally, the code allows for the deletion of job posts when a specific condition is met, and a confirmation alert is displayed upon successful deletion.

13. view-events-details.php – is the script responsible for viewing details of college events in a college alumni system. It fetches event details from the database based on the event ID, displaying the title, banner image, description, and schedule. The page layout is responsive and uses Bootstrap. It includes session checks for user authentication and incorporates necessary libraries for styling and functionality. Ensure security measures, like input validation and prevention of SQL injection, are implemented for robustness, and consider adding proper error handling mechanisms for user feedback.

14. view-events.php – The script responsible for viewing college events in a college alumni system. It includes features like session checking for user authentication, Bootstrap for styling, and pagination for better user experience. The page displays a table with event details, such as event title, schedule, and creation date. Users can click on "View" to see more details about a specific event.

15. view-jobpost-detail.php – The final script in line under the folder of “alumni” folder which is responsible for viewing details of a specific job post along with the associated alumni details in a college alumni system. It includes features like session checking for user authentication, Bootstrap for styling, and displays comprehensive information about the job post and the alumni who posted it. The page also shows the status and remarks, if any, provided by the admin.

C. Admin Interface (Folder)

1. **about-us.php** – a PHP code represents a page for updating the "About Us" section in a College Alumni System. It includes form fields to modify the page title and description. The code also retrieves the current content of the "About Us" page from the database for pre-filling the form. A session check is performed to ensure that only authenticated users with valid sessions can access this page. The form data is processed when the user submits the form, and the changes are updated in the database. Additionally, a success alert is displayed after the update is completed.
2. **add-course.php** – It is script responsible for adding a new course to the College Alumni System. It includes a form to input the course name, and upon submission, the entered data is inserted into the database table named "tblcourse". After successful insertion, an alert is displayed, and the user is redirected to the "Add Course" page. The script also includes basic HTML and CSS for styling the form and the page layout. It checks for an active session to ensure that only authenticated users can access this page.
3. **add-events.php** – A script that is designed to handle the addition of events in a College Alumni System. It begins by checking the user's session and redirects to the logout page if no session is active. The code then processes the form submission, validating and handling the input data for the event title, schedule, description, and banner image. It ensures that the image format is valid (jpg, jpeg, png, gif) before uploading it to the server. The event details are then inserted into the database table tblevents, and upon successful insertion, an alert is displayed, and the user is redirected to the "Add Events" page. The accompanying HTML and CSS provide a well-structured and visually appealing form for adding events. Each form field, including the event title, schedule, description, and banner

image, is presented with corresponding labels and input fields. The form also includes necessary elements for file upload. The overall layout is designed to fit into the College Alumni System's user interface, with a responsive and user-friendly design.

4. **all-job-posts.php** – The PHP script's only one purpose is to implement a web page for displaying all job posts within a College Alumni System. The page features a well-designed table presenting job details such as title, company name, location, vacancy, designation, last date, posting date, and status. Pagination is incorporated for efficient navigation through the list of job posts. The system dynamically fetches and displays job records from a database, ensuring a user-friendly interface. Additionally, the code includes responsive design elements and integrates pagination functionality for a smoother user experience.
5. **alumni-jobs.php** – The provided PHP code constitutes a web page for displaying job posts specific to a particular alumnus in a College Alumni System. The page incorporates a well-designed table presenting details such as job title, company name, location, vacancy, designation, last date, posting date, and status. Pagination functionality is implemented for efficient navigation through the job posts. The system dynamically fetches and displays job records from a database based on the alumni ID. The code demonstrates responsiveness and includes pagination controls for enhanced user experience.
6. **alumni-list.php** – The provided PHP code represents a page for managing alumni lists in the Alumni System. It incorporates functionality to delete alumni records and associated job posts. The page includes a well-designed table displaying alumni details such as full name, college ID, batch, email, and registration date. Actions for viewing details, deleting records, and checking posted jobs are provided. Pagination is implemented for efficient navigation through the alumni list.

7. **approved-job-post.php** – This script generates and represents a web page for displaying approved job posts in a College Alumni System. The page includes a table presenting details such as job title, company name, location, vacancy, designation, last date, posting date, and status. The records are paginated for efficient navigation. The code ensures that only job posts with an "Approved" status are retrieved from the database. Each job entry includes a "View Details" button linking to a detailed view page. The page adheres to a clean and responsive design, with a sidebar and top bar for navigation. Additionally, there are provisions for user authentication and error reporting.
8. **betweendates-alumni.php** – The stated script is responsible for managing and displaying alumni records based on a specified date range. The page allows administrators to delete alumni entries and filter records by registration date. It includes a form for entering the date range, and upon submission, the page retrieves and displays alumni details within the specified period. The records are presented in a paginated table, showing details such as full name, college ID, batch, email, and registration date. The code ensures user authentication and includes a clean and responsive design with a sidebar and top bar for navigation. Additionally, there are provisions for error reporting and handling.
9. **betweendates-jobpost.php** – this script is responsible for managing and displaying job posts based on a specified date range. The page includes a form to input the date range, and upon submission, it retrieves and displays job posts within the specified period. The records are presented in a paginated table, showing details such as job title, company name, location, vacancy, designation, last date, posting date, and status. The code ensures user authentication, handles date filtering, and features a clean and responsive design with a sidebar and top bar for navigation. Also has error reporting and handling.

10. **cancel-job-post.php** – This script is responsible for managing and displaying canceled job posts. The page includes a table showing details such as job title, company name, location, vacancy, designation, last date, posting date, and status of canceled job posts. The records are paginated, and the page features a clean and responsive design with a sidebar and top bar for navigation. The code ensures user authentication and handles the retrieval and display of canceled job posts from the database.
11. **change-password.php** – This script represents a change password functionality for an admin user in a College Alumni System. The page includes a form where the admin can enter the current password, new password, and confirm the new password. Upon form submission, the code checks if the entered current password is correct. If correct, it updates the password in the database with the new one. Plus ensures that the new password and confirm password fields match using client-side validation with JavaScript. The page has a clean and responsive design with a sidebar and top bar for navigation. Additionally, user authentication is enforced to ensure that only logged-in admins can access the change password functionality.
12. **changeimage.php** – This script manages the update of a banner image for a specific event in a College Alumni System. Admin authentication is enforced, and the system redirects to the logout page if the admin is not logged in. The code handles the submission of a form that allows the admin to update the banner image for a selected event. It validates the image format and, if valid, uploads the new image, updating the corresponding database record. The page features a clean and responsive design with a sidebar and top bar for navigation.

13. **contact.php** – Is a PHP script with basic email form handler that validates and processes user input from an HTML form. It checks for the presence and validity of the name, email, and message fields, displaying error messages if any are missing or incorrect. The script then constructs an email with the user's input and attempts to send it using the mail() function. If the email is sent successfully, it displays a success message; otherwise, it shows an error message.
14. **contactus.php** – This script specializes in the updating of contact information on a website's "Contact Us" page. It retrieves existing data from the database, including page title, description, email, and mobile number, and presents it in a form for editing. Upon submission, the script updates the database with the modified information. The form includes fields for page title, description, email, and mobile number, pre-filled with the current data. The script also incorporates basic HTML and CSS for page layout and styling.
15. **dashboard.php** – The script constitutes a dashboard for a college alumni system. It utilizes sessions for user authentication and includes various statistical counters such as the total number of events, courses, new job post requests, approved job posts, canceled/rejected job posts, total job requests, and total alumni registrations. The counters are dynamically fetched from the database using PDO queries. The script incorporates HTML, CSS, and Bootstrap for layout and styling, presenting a visually organized dashboard with clickable buttons for detailed views of each category. Additionally, it includes JavaScript libraries for animations and charting.
16. **edit-course.php** – A PHP script that manages the updating of a college course name in the database. It incorporates session handling for user authentication, and if the session is not active, it redirects to a logout page. The script includes an HTML form for updating

the course name, which is retrieved from the database based on the provided course ID. Upon submission of the form, the course name is updated in the database. The layout is styled using Bootstrap and custom CSS, and JavaScript libraries are included for various functionalities. However, it is important to note that the script lacks proper input validation, potentially exposing it to security vulnerabilities.

17. **edit-events.php** – This PHP script facilitates the updating of event details in a college alumni system by checking the user's session and redirects to a logout page if not active. Upon form submission, it updates the event title, schedule, and description in the database. The form is pre-filled with existing event details fetched based on the provided event ID. The layout is styled using Bootstrap, and JavaScript libraries are included for various functionalities.

18. **forgot-password.php** – A PHP script manages a password reset functionality for a College Alumni System. It starts by verifying the provided email and mobile number against the database. If a match is found, it allows the user to reset their password by entering a new one. The form includes validation to ensure the new password and confirm password fields match. The layout is styled using Bootstrap, and the user is informed of the password change status through alert messages.

19. **login.php** – This PHP script handles user authentication for a College Alumni System. It checks user credentials against the database and, if valid, grants access to the dashboard. The login form allows users to enter their username and password, with an option to remember the login details using cookies. The layout is styled using Bootstrap, and the script lacks input sanitation for potential security vulnerabilities. The user is informed of any login errors through alert messages. Additionally, there is a "Forgotten Password?" link for password recovery and a "Home Page" link for navigating to the public home page.

20. **manage-courses.php** – This script manages courses in a College Alumni System. It includes session handling for user authentication and redirects to the logout page if the user is not logged in. The script allows for the deletion of courses, displays a table of courses with options for editing and deleting each entry, and implements pagination for better user experience. The design is styled using Bootstrap, and the pagination logic involves calculating total pages based on the number of records per page.
21. **manage-events.php** – The PHP script that manages events in a College Alumni System, featuring session handling for user authentication and redirection to the logout page if the user is not logged in. It allows for the deletion of events, displays a table with options for editing and deleting each entry, and incorporates pagination for improved user experience. The design is styled using Bootstrap, and the pagination logic calculates total pages based on the number of records per page.
22. **new-job-post.php** – This PHP script facilitates the management of new job posts, incorporating session handling for user authentication and redirection to the logout page if the user is not logged in. The design is styled using Bootstrap, with modular components such as a sidebar, header, and footer. The script displays a table of job posts with details like job title, company name, location, vacancy, designation, last date, posting date, and status. It includes pagination for enhanced user experience, allowing users to view details of each job post and navigate through the pages. The pagination logic calculates total pages based on the number of records per page.
23. **profile.php** – The script responsible for managing the profile of an admin user in a College Alumni System. It begins by checking the user's session and redirects to the logout page if not logged in. The script handles form submission for updating the admin's profile information, such as name, contact number, and email. The design utilizes Bootstrap for

styling, and the page includes a form with pre-filled values retrieved from the database. Upon submission, the script updates the profile and displays an alert, redirecting the user back to the profile page. The page layout consists of a sidebar, header, and various sections for displaying and updating profile details.

24. **view-alumni-details.php** – The PHP script responsible for viewing details of a specific alumni by checking the user's session and redirects to the logout page if not logged in. The script retrieves alumni details, including full name, gender, batch, course graduated, working organization, email, and profile picture, from the database based on the provided alumni ID. The page layout uses Bootstrap for styling and displays the details in a table format. The alumni details are presented in a structured manner with labels and corresponding values. The script also includes various CSS and JS files for styling and functionality. Additionally, there's a script to conditionally show/hide a section based on the selected value in a dropdown menu, allowing for dynamic behavior in response to user actions.

25. **view-jobpost-detail.php** – This PHP script is designed to view job details and take actions on job posts. It first checks the user's session and redirects to the logout page if the user is not logged in. The script retrieves job details and alumni information from the database based on the provided job post ID (\$vid). The details include job title, company name, location, vacancy, designation, job description, contact person, contact number, last date, posting date, status, and admin remarks. The script also displays this information in a table format. If the job post status is not updated, it provides a button to take action, and a modal (#myModal) is displayed for updating the status and providing remarks. The available status options are "Approved" and "Cancelled." The admin can enter remarks in a text area, select a status, and submit the form to update the job post status.

D. PHP's Code Modularization

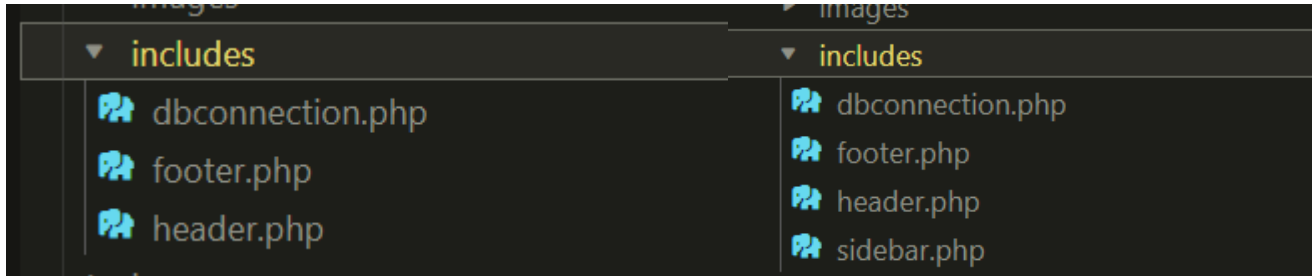


Figure 9: An example of Modularization or Functional Decomposition for reusability purposes.

Almost all major folders (or interface) of the researchers' created website contains a subfolder named "includes" which will become the fragments that, instead of rewriting the code all over again, is loaded via the syntax,

"`<?php include_once('includes/whatever_php_code_to_include'); ?>`", for a one-change rule, that will help the researchers debug or modify markups easier since it is on a modular/object type architecture.

E. Disclaimer:

As for the codes, due to the sheer scale amount of Line of Code (LOC) that the proponents have accumulated over the span of development, the decision was collectively made to not list it down here to make the budget for printing less. Instead, it will be presented publicly on the internet at GitHub platform, an online collaborative tool.

The link on the remote repository is here: <https://github.com/GMBrhylle/alumback>

Chapter V

DATABASE STRUCTURE

This chapter will reveal the database and its table at the back-end operation.



Figure 10: Overview of Database Structure in phpMyAdmin

The following snapshots are the table structures of each tables inside the MySQL database named “plp_alumnisystem”.


a.) tbladmin or the Table of Administrators

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	ID 🔑	int(10)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	AdminName	varchar(120)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 3	UserName	varchar(120)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 4	MobileNumber	bigint(10)			Yes	NULL		
<input type="checkbox"/> 5	Email	varchar(200)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 6	Password	varchar(200)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 7	AdminRegdate	timestamp			Yes	current_timestamp()		


b.) tblalumni or the Table of Alumnus

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	ID 	int(10)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	FullName	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 3	CollegeID	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 4	Gender	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 5	Batch	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 6	CourseGraduated	int(10)			Yes	NULL		
<input type="checkbox"/> 7	CurrentlyConnected	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 8	Image	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 9	EmailId	varchar(255)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 10	Password	varchar(100)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 11	RegDate	timestamp			Yes	current_timestamp()		


c.) tblcourse or the Table of Courses (currently available at PLP)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	ID 	int(5)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	CourseName	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 3	CreationDate	timestamp			Yes	current_timestamp()		

d.) tblevents or the Table of Events (events currently anticipated at PLP)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	ID 	int(5)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	EventTitle	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 3	Schedule	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 4	Description	mediumtext	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 5	BannerImage	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 6	CreationDate	timestamp			Yes	current_timestamp()		

e.) tbljobpost or the Table of Job Listings (made available by Alumnus or Admin)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 ID 	int(10)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 AlumniID	int(10)			Yes	NULL		
<input type="checkbox"/>	3 JobTitle	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	4 CompanyName	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	5 Location	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	6 Vacancy	int(10)			Yes	NULL		
<input type="checkbox"/>	7 Designation	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	8 JobDescription	mediumtext	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	9 ContactPerson	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	10 ContactNumber	bigint(11)			Yes	NULL		
<input type="checkbox"/>	11 LastDate	date			Yes	NULL		
<input type="checkbox"/>	12 PostingDate	timestamp			Yes	current_timestamp()		
<input type="checkbox"/>	13 Remark	varchar(250)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	14 Status	varchar(100)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	15 UpdationDate	timestamp			Yes	NULL		ON UPDATE CURRENT_TIMESTAMP()

f.) tblpage or the Table of Pages (for the sticky strip of header located on top of the navigation bar)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 ID 	int(10)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 PageType	varchar(200)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/>	3 PageTitle	mediumtext	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/>	4 PageDescription	mediumtext	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/>	5 Email	varchar(200)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/>	6 MobileNumber	bigint(10)			Yes	NULL		
<input type="checkbox"/>	7 UpdationDate	date			Yes	NULL		
<input type="checkbox"/>	8 Timing	varchar(200)	latin1_swedish_ci		No	None		

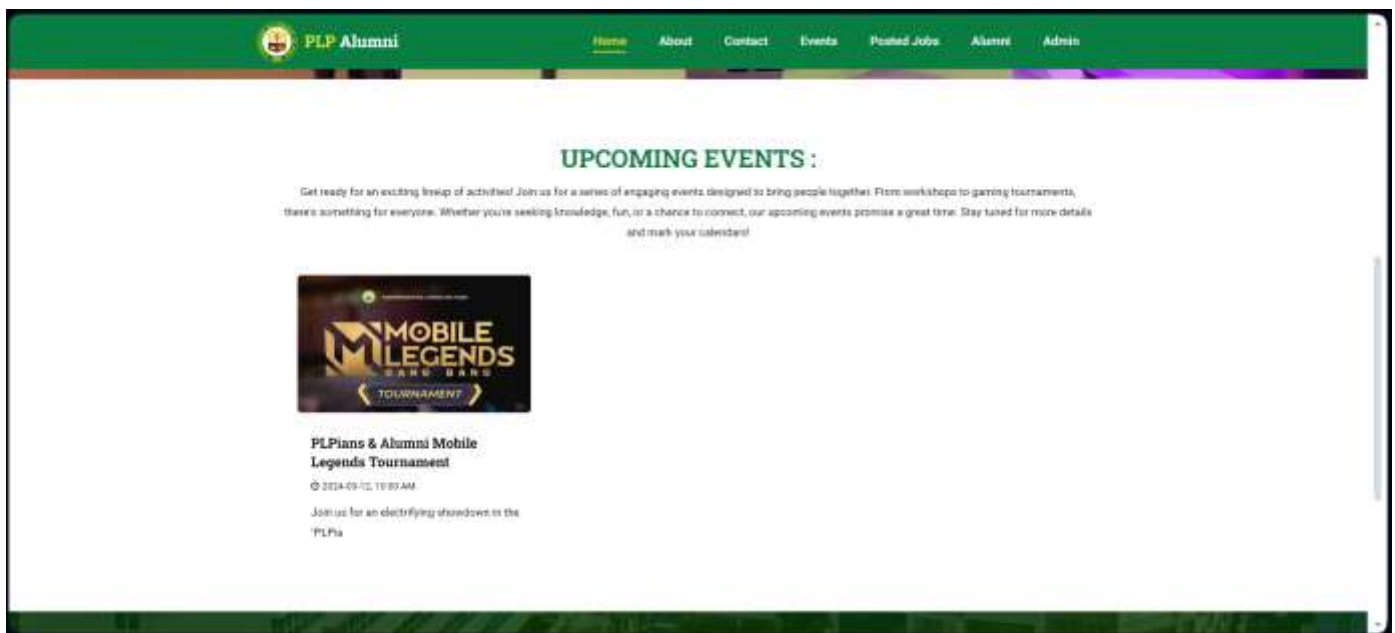
Chapter VI

OUTPUTS

This section will illustratively discuss the pictures through a single caption method.



Figure 11: Illustration of HomePage Output (Part 1)



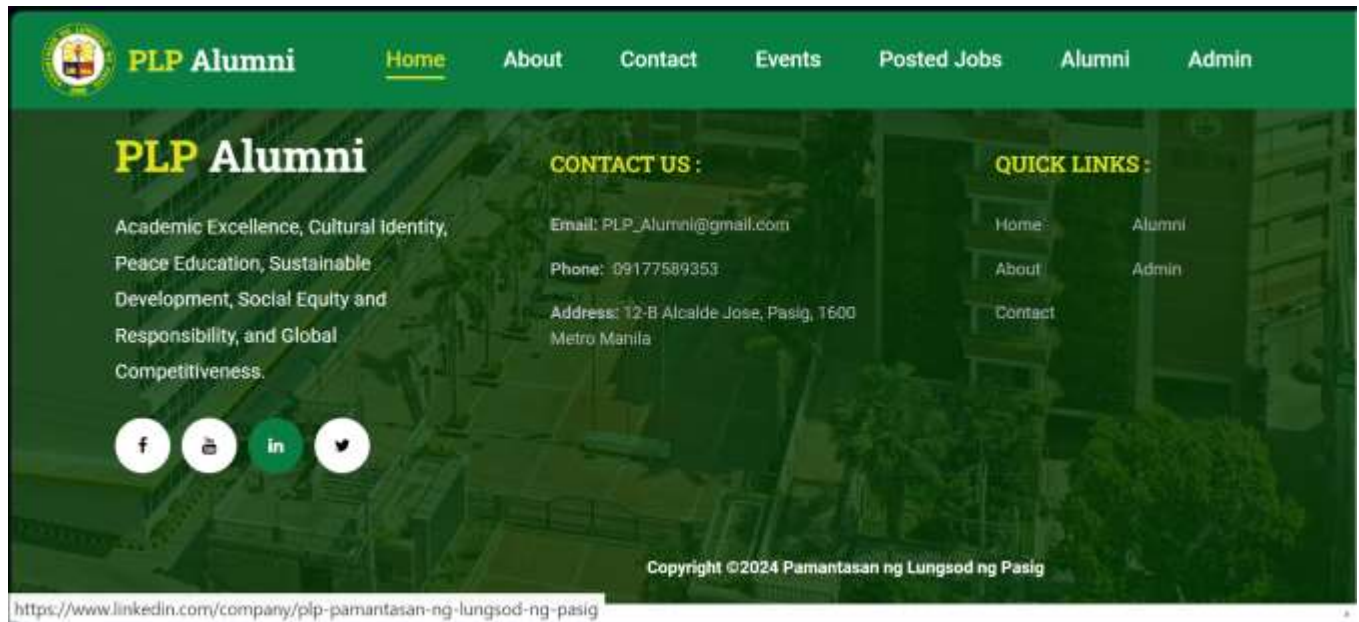


Figure 12: Illustration of the footer at the Home Page, in parallax motion. (Part 2)



Figure 13: The parallax hiding motion of header, gives way to the peeking motion of footer below.



Figure 14: About Us - Section PLP's Background (Part 1)



Figure 15: About Us - Section: PLP's Alumni Assoc. Officers (Part 2)

Journey Through Alma Mater: A Heartwarming Alumni Story



Figure 16: Featured Uplifting Video for Alumnus (Part 3), Below is the Milestones Achieved by the University.

Achievements

Our alumni have achieved milestones in various fields SOON TO BE UPLOADED...

PLP Alumni

Academic Excellence, Cultural Identity, Peace Education, Sustainable Development, Social Equity and Responsibility, and Global Competitiveness.

CONTACT US :

Email: PLP_Alumni@gmail.com

Phone: 09177589353

Address: 12-B Alcade Jose, Pasig, 1600 Metro Manila

QUICK LINKS :

Home	Alumni
About	Admin
Contact	

PLP Alumni

[Home](#)
[About](#)
[Contact](#)
[Events](#)
[Posted Jobs](#)
[Alumni](#)
[Admin](#)

CONTACT INFORMATION

Address
12-B Alcade Jose, Pasig, 1600 Metro Manila

Contact Number
+9177589353

Email
PLP_Alumni@gmail.com

Figure 17: Contact Section with Map Pointing at the Coordinates of PLP

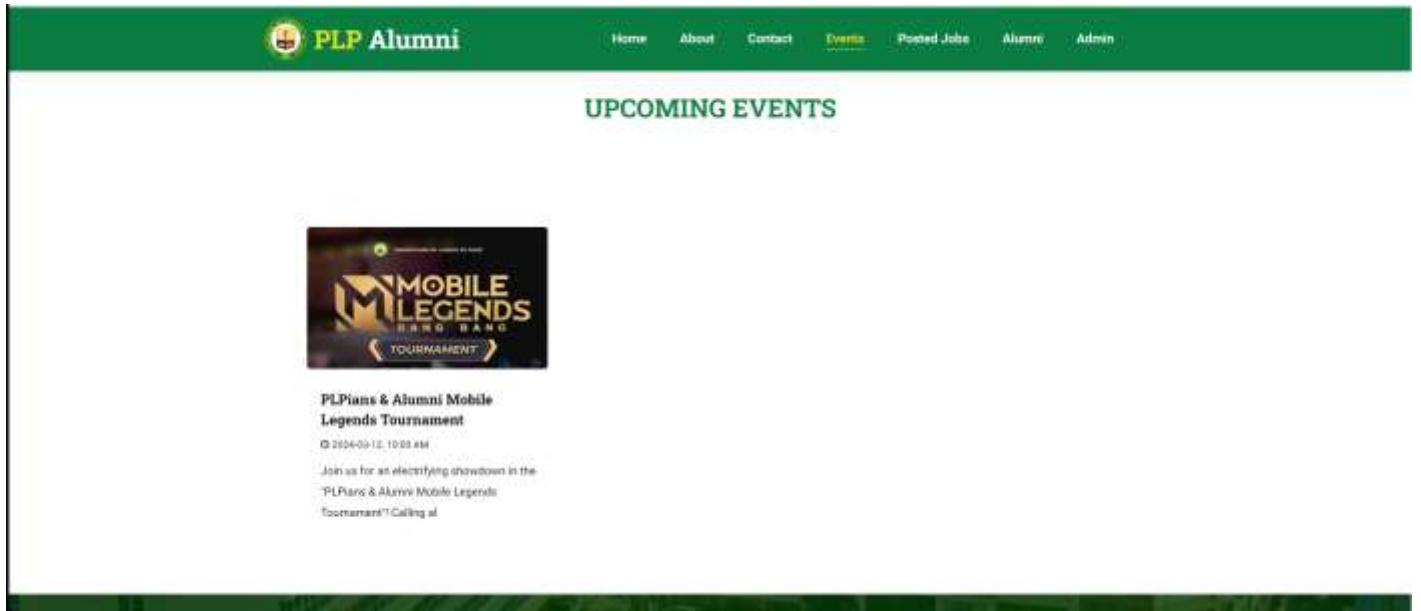


Figure 18: Highlights the Events Tab and its Details.

Events Details



PLPians & Alumni Mobile Legends Tournament

Join us for an electrifying showdown in the "PLPians & Alumni Mobile Legends Tournament"! Calling all gaming enthusiasts from PLP and alumni to team up and clash in thrilling battles. Compete, strategize, and showcase your skills in this epic gaming event. It's a chance to reconnect, have fun, and embrace the spirit of friendly competition. Mark your calendars and get ready to dive into the action-packed world of Mobile Legends with your fellow PLPians and alumni! [CLICK ME TO REGISTER](#)

Date and Time

2024-03-12, 10:00 AM

Posted Job Details

S.No	Job Title	Company Name	Location	Vacancy	Designation	Job Description	Last Date	Posting Date	Contact Number	Contact Person
1	Vacancy For Telecaller	Kamila Pvt Ltd	Cutugram	350	Telecaller	As a Telecaller, you are responsible for handling sales over the phone entirely. You are also responsible for solving questions in regard to the product or service provided by the company. In addition to this, you should collect desired information from the clients and maintain healthy relations with them.	2023-12-13	2023-11-30 21:46:39	8395464648	Hemanshu

First

← Prev

Next →

Last

Figure 21: Listed/Posted Job Details with 1 Entry.

**PLP ALUMNI**

WELCOME ALUMNI 😊

Email ID

enter your email id

Password

enter your password

☐ Remember Me

Forgotten Password?

Login

Home Page

Registration Page

Figure 19: PLP Alumni Login

FORGOT PASSWORD

Email

Email

New Password

New Password

Repeat Password

Confirm Password

RESET

Login

Figure 20: PLP Alumni Login - Forgotten Password

REGISTRATION PAGE

Fullname

enter your fullname

College ID

enter your college id number

Gender

Select Gender

Batch

enter your Batch

Graduated

Select Course

Currently Conncted To

enter your companyname

Pic

Choose File

No file chosen

Email

enter your emailid

Password

enter your password

Register

Signin

Figure 22: Registration Page

AFTER ALUMNI LOGIN

View College Events

Job Details Posted by (JHEIZON BRHYLLE DELA CRUZ)

Job Title	CV Engineer	Company Name	2570
Location	New York, USA	Vacancy	3
Designation	Me	Job Description	The applicant should know the fundamentals to masters of Python, and must know about CNN and different variants. He/She should show a great interest on Software Engineering
Contact Person	GM Brhyll	Contact Number	892827057
Last Date	2024-01-13	Posting Date	2024-01-03 03:38:42
Status	Not Response Yet	Admin Remark	Not Updated Yet

Alumni Details

Full Name	JHEIZON BRHYLLE DELA CRUZ	College ID	25-0432
Gender	Male	Batch	2023-2024
Course Graduated	BS in Computer Science (BSCS)	Working Organization	2570
Email	delacruz_jheizonbrhyll@gmail.com	Photo	

Figure 23: An Example of Posted For Approval Jobs

Manage Job Post

Manage Job Post

	Job Title	Company Name	Location	Vacancy	Designation	Last Date	Posting Date	Status	Action
1	CV Engineer	2570	New York, USA	3	Me	2024-01-13	2024-01-03 03:38:42	Not updated Yet	View Details

First Prev Next Last

Figure 24: An Example of Management of Posted Jobs

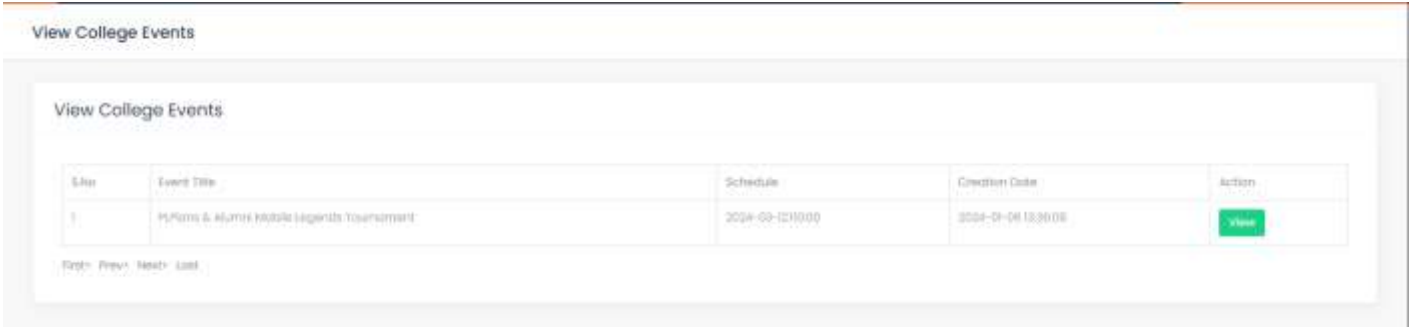


Figure 28: View Events Tab at Verified Alumni

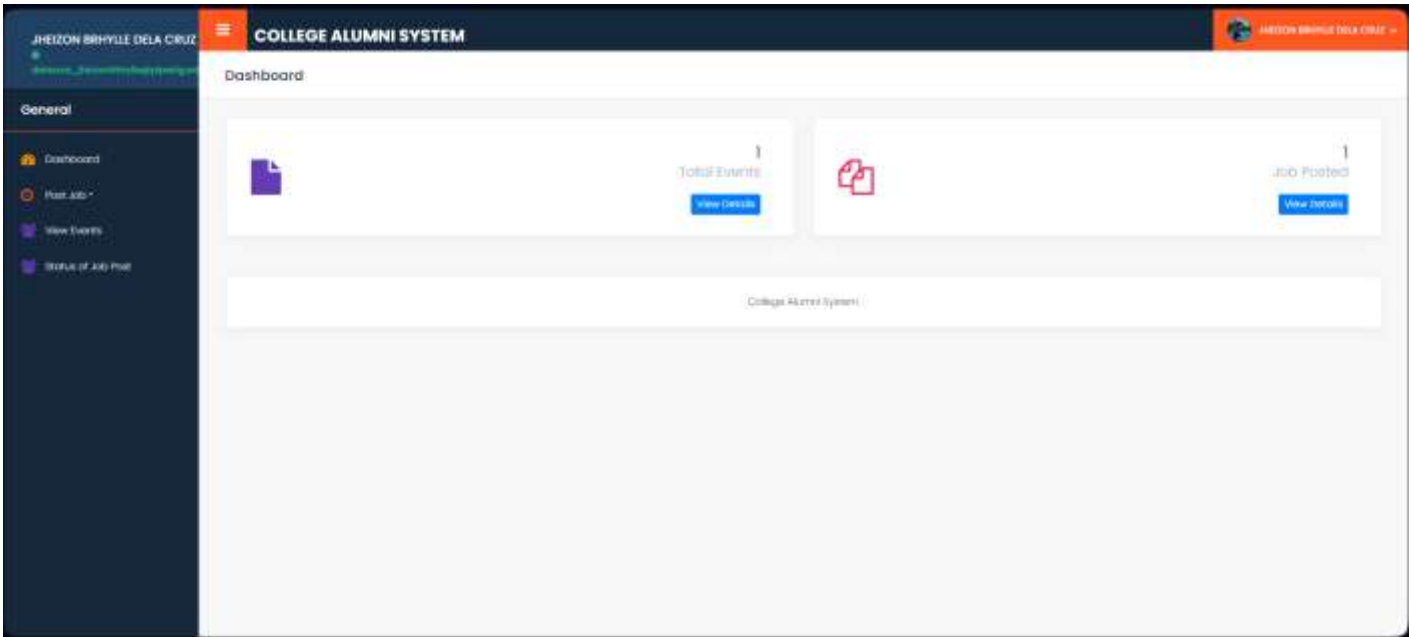


Figure 27: Dashboard Overview

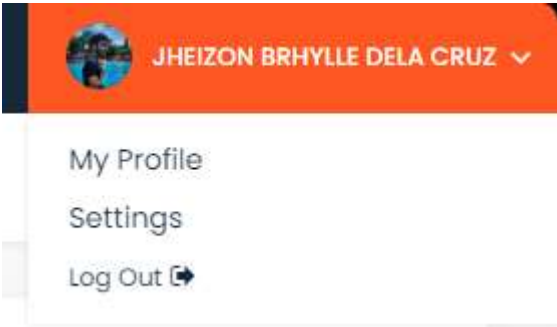


Figure 25: A dropdown menu at individual’s profile.

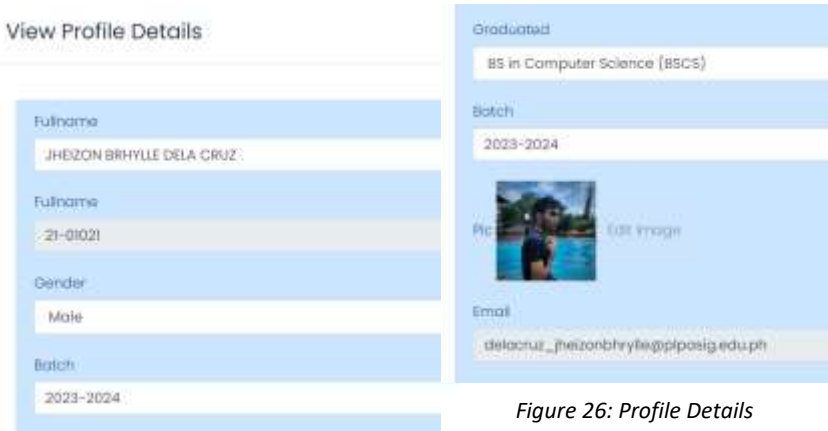


Figure 26: Profile Details

Email

delacruz_jheizonbhrylle@plpasig.edu.ph

Currently Connected To

2STO

Registration Date

2024-01-13 03:33:29

Update

Figure 31: An option to UPDATE.

User Name

enter your username

Password

enter your password

☐ Remember Me

Forgotten Password?

Login

Home Page

Figure 30: An Admin Portal

Email

Email

Mobile Number

Mobile Number

New Password

New Password

Confirm Password

Confirm Password

RESET

Login

Figure 29: An Admin Forgotten Password

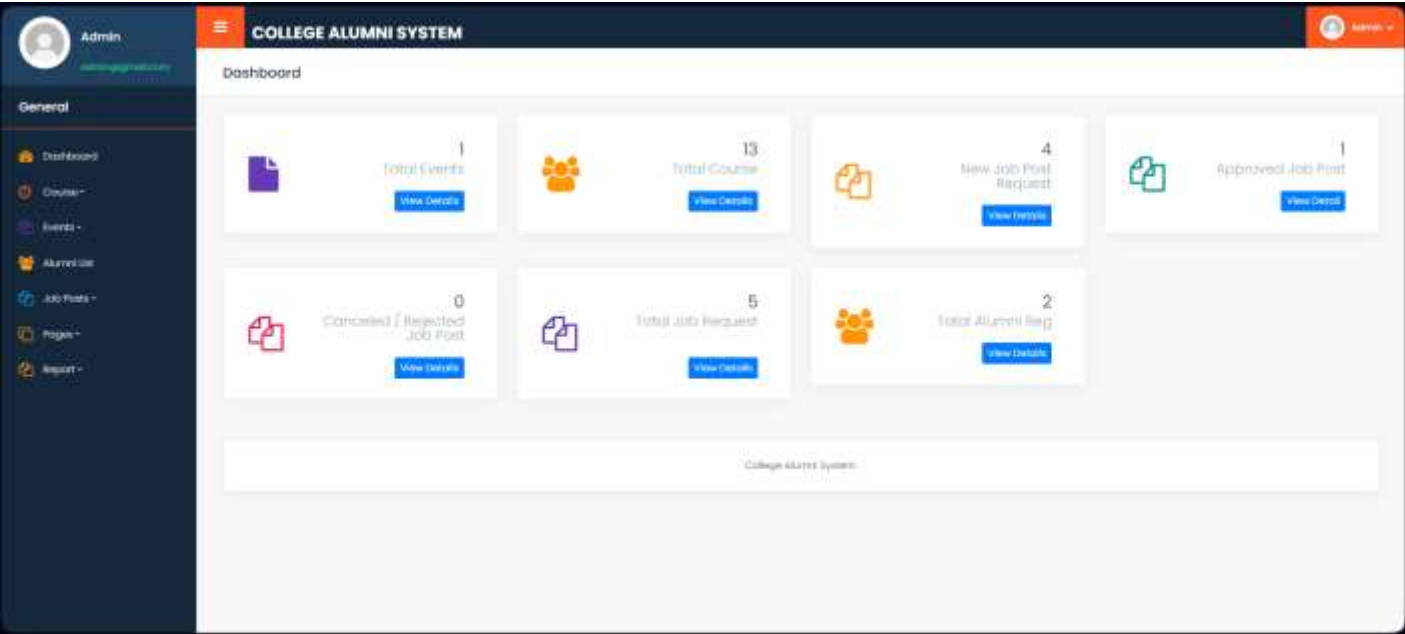


Figure 32: Dashboard for the Admin

Change Password

Change Password

Current Password

New Password

Confirm Password

Update

Figure 33: Change Password for Admin

Manage Events

S.No	Event Title	Schedule	Creation Date	Action
1	PLPlans & Alumni Mobile Legends Tournament	2024-03-12, 10:00 AM	2024-01-06, 01:36 PM	Edit Delete

First> Prev> Next> Last

Figure 34: Manage Events (Admin Authority)

Add Events

Event Titles

Schedule

dd/mm/yyyy --:-- --

Description

Banner Image

Choose File

No file chosen

Add

Events ▾

> Add Events

> Manage Events

Figure 36: Add Events (Admin Authority)

Add Course

Course Name

Add

Course ▾

> Add

> Manage

Figure 35: Add and Manage Course

Manage Course

S.No	Course Name	Creation Date	Action
1	BS in Information Technology (BSIT)	2024-01-03 20:51:09	<button>Edit</button> <button>Delete</button>
2	BS in Computer Science (BSCS)	2024-01-03 20:52:19	<button>Edit</button> <button>Delete</button>
3	Bachelor in Elementary Education (BEEd)	2024-01-03 20:52:35	<button>Edit</button> <button>Delete</button>
4	Bachelor in Secondary Education Major in English (BSSEd- English)	2024-01-03 20:52:45	<button>Edit</button> <button>Delete</button>
5	Bachelor in Secondary Education Major in Filipino (BSSEd- Filipino)	2024-01-03 20:53:05	<button>Edit</button> <button>Delete</button>
6	Bachelor in Secondary Education Major in Mathematics (BSSEd- Math)	2024-01-03 20:53:16	<button>Edit</button> <button>Delete</button>
7	Bachelor of Science in Nursing (BSN)	2024-01-03 20:53:23	<button>Edit</button> <button>Delete</button>
8	Bachelor of Science in Accountancy (BSA)	2024-01-03 20:53:40	<button>Edit</button> <button>Delete</button>
9	Bachelor of Science in Business Administration Major in Marketing Management (BSBA)	2024-01-03 22:02:03	<button>Edit</button> <button>Delete</button>
10	Bachelor of Science in Entrepreneurship (BS Entrep)	2024-01-03 22:05:30	<button>Edit</button> <button>Delete</button>

Figure 40: Lists all in a table with Edit and Delete Function

Manage Alumni List

S.No	Full Name	College ID	Batch	Email	Registration Date	Action
1	WILFRYD JAMES TUPAS	21-00878	2025	Wuparasdfermas2002@gmail.com	2024-01-07 21:34:23	<button>View</button> <button>Delete</button> <button>Posted Job</button>
2	JESSICA BEYVILL DE LA CRUZ	21-00121	2023-2024	delacruz_jhazordhryk@pilasig.edu.ph	2024-01-18 03:03:29	<button>View</button> <button>Delete</button> <button>Posted Job</button>

First Prev Next Last

Figure 39: Management of Alumni List

 Job Posts


> New Job Post

> Approved Job

> Unapproved Job


> All Job Post

Figure 38: Listed Jobs with Different Filters

 Report

> Job Post

> Alumni Registration

 Pages

> About Us

> Contact Us

Figure 37: Manage Pages

Contact Us

Page Title

Contact Us

Page Description

12-B Alcala Jose, Pasig, 1600 Metro Manila

Email

inquiry@pilasig.edu.ph

Mobile Number

9177589353

Update

Figure 41: Update the Organization Entity's Contact Details

Figure 42: Side-Menu of Reports Query

All Job Post

#	Job Title	Company Name	Location	Vacancy	Designation	Last Date	Posting Date	Status	Action
1	Vacancy For Telecaller	Kamini Pvt Ltd	Bangalore	250	Telecaller	2023-12-8	2023-11-08 11:48:48	Approved	View Details
2	Vacancy For Telecaller	Kamini Pvt Ltd	Bangalore	250	Telecaller	2023-12-8	2023-11-08 11:48:48	Not Updated Yet	View Details
3	Vacancy For Junior HR	HR Company	India	5	Jr. HR	2023-12-21	2023-11-04 11:54:31	Not Updated Yet	View Details
4	Full Stack Developer	HCL	India Off	1	Full Stack Developer	2023-12-30	2023-11-10 08:51:11	Not Updated Yet	View Details
5	CV Engineer	2510	New York, USA	3	Mr	2024-01-13	2024-01-13 03:38:41	Approved	View Details

Figure 43: All Job Posts Filter

Job Post from 13-01-2024 To 13-01-2024

#	Job Title	Company Name	Location	Vacancy	Designation	Last Date	Posting Date	Status	Action
1	CV Engineer	2510	New York, USA	3	Mr	2024-01-13	2024-01-13 03:38:41	Approved	View Details

First Prev Next Last

Figure 44: Query Reports (by specifying the range of date, below is the results of each query) See Figure 41.

Alumni List from 01-01-2024 To 13-01-2024

S.No	Full Name	College ID	Batch	Email	Registration Date	Action
1	Richmond James Tajaron	21-00879	2023	sojanorichyfranc2063@gmail.com	2024-01-07 21:34:29	View Delete
1	JACSON ORVILLE DELA CRUZ	21-01021	2023-2024	delacruz_jhelonbrylle@gmail.com	2024-01-13 03:33:19	View Delete

Approved Job Post

#	Job Title	Company Name	Location	Vacancy	Designation	Last Date	Posting Date	Status	Action
1	Vacancy For Telecaller	Kamini Pvt Ltd	Bangalore	250	Telecaller	2023-12-8	2023-11-08 11:48:48	Approved	View Details
2	CV Engineer	2510	New York, USA	3	Mr	2024-01-13	2024-01-13 03:38:41	Approved	View Details

First Prev Next Last

New Job Post

#	Job Title	Company Name	Location	Vacancy	Designation	Last Date	Posting Date	Status	Action
1	Vacancy For Telecaller	Kamini Pvt Ltd	Bangalore	250	Telecaller	2023-12-8	2023-11-08 11:48:48	Not Updated Yet	View Details
2	Vacancy For Junior HR	HR Company	India	5	Jr. HR	2023-12-21	2023-11-04 11:54:31	Not Updated Yet	View Details
3	Full Stack Developer	HCL	India Off	1	Full Stack Developer	2023-12-30	2023-11-10 08:51:11	Not Updated Yet	View Details

First Prev Next Last

Figure 45: Review On Process

Cancel Job Post

#	Job Title	Company Name	Location	Vacancy	Designation	Last Date	Posting Date	Status	Action
---	-----------	--------------	----------	---------	-------------	-----------	--------------	--------	--------