

Project Title: PrimaAlumni

Group Task Submitted By:

ramdeobaba - webdev - team - PrimaAlumni

Team Members:

1. Radha Sachin Malichkar
2. Rachita Bharambe
3. Snehal Nirajkumar Mopkar
4. Tanmay Pawan Bisen
5. Sameer Shekhar Mishra

College Name:

Shri Ramdeobaba College of Engineering and Management, Nagpur-440013

Date of commencement : 9 Nov 2021

Date of Submission: 9 Jan 2022

Project Details

Technology used:

- Frontend: HTML, CSS, Bootstrap4
- Backend: PHP
- Database: MySQL (phpmyadmin) wamp/xampp server

Output:

- Please find attached OUTPUT.pdf file, pdf contains collection of screenshots representing UI and basic workflow of website.

Login\ Register\ Edit basic information:

- **Pages created:** login.php, signup.php, welcome.php
Logout.php, partials folder has _dbconnect.php file for database connections, along with these there are style.css files used for external styling.
- **Data Storage:**
 1. *Database:* users (import users.sql)
 2. *Table:* alumni (import alumni.sql)
 3. *Attributes* : username, password, fullname, email, qualification, dob, yop.
- First page of the project is **signup.php** for new users, signup is necessary for access to key features.

- If the user has already registered then the login modal can directly be filled on **login.php**. All the database operations are going to be done with database Users and table Alumni.
- For login system authentication one way function is used, that is using the **hash function** to generate a hashed version of the password.
- After successful verification of password, the user is given access to the **welcome.php** file, inturn we have granted access to all alumni functionalities to the authentic user.
- On welcome.php a basic feature of the edit profile is provided, that is related to primitive user details and password changing.
- For security purposes, if anyone tries to trespass any core files, or web pages without login, they will be forcefully redirected to the login.php page.
- Finally, the user has the option of Logout, through **logout.php** file user session can be ended. This brings back users to welcome.php again.

Events :

- Alumni can view the upcoming events by logging in to his alumni account.
- Events can be added or pre existing event details can be updated but the ***Event name cannot be updated once the event is created.***
- **Data Storage:**
 1. *Database* : users (import users.sql)
 2. *Table* : events (import events.sql)
 3. *Attributes* : Event name,Venue,Time and Organiser.
- Event name is the primary key for the table.
- If you wish to update details for an event you need to provide the original event name whose details are to be updated, The details to be updated can then be provided through the form.
- Users can view the latest and upcoming events,create events and update any event.
- **Pages Created:**
 1. *EventDetails.php* -: Display the created or updated events and has the gateway to ADD and UPDATE events page.
 2. *Eventupdate.php* -:Update the existing event by means of a form.
 3. *Eventadd.php* -:Add the existing event by means of a form.

Jobs:

- Users can view various job opportunities and apply in any of them.
- In the details column alumni can add the link for job opportunities and interested users can use it.
- Alumni who are registered users can update their details whenever they feel like doing so.
- **Data Storage:**
 1. *Database* : users
 2. *Table Name* : jobdata
 3. *Attributes* : Company Name, Description, Starting_year, Ending_year , Details
- **Pages Created:**
 1. *jobsDetails.php* -: user can view data which is fetched from database, this page consists of add and update button which will link to add and update forms respectively.
 2. *jobsadd.php* -: It consists of a form where the user can add his/her job details.
 3. *jobsUpdate.php* -: It consists of a form where users can update their details which will lead to changes in database.

User Details:

- **Page Name:** *personal.php* :

This file will first of all search for the user data wrt username and print the data in the respective column.

Users can change the data whenever they wish to. They just have to update the respective field and click on the save button which is placed at the bottom of the file.
- **Data Storage:**
 1. *Database* : users
 2. *Table name* : perdata
 3. *Attributes* : fname, lname, email, dob, pno, country, state, city, username
- **Page Name:** *edu.php*

This file will first of all search for the user data wrt username and print the data in the same page one after the other in the form of a card.

Users can add the data whenever they wish to. They just have to fill the respective field and click on the add button.

- **Data Storage:**
 1. *Database* : users
 2. *Table name* : edudata
 3. *Attributes* : Course, School, Grade, Year, Desc, username
- **Page Name:** *work.php*
 This file will first of all search for the user data wrt username and print the data in the same page one after the other in the form of a card.
 Users can add the data whenever they wish to. They just have to fill the respective field and click on the add button.
- **Data Storage:**
 1. *Database* : users
 2. *Table name* : workdata
 3. *Attributes* : title, comname, desc, syear, eyer, username

Contact Us Page :

Files Created: index.php , sendemail.php , style.css

Folder Created : phpmailer

- **Page Name :** *index.php*
 This file contains contact information of PrimaAlumni and contains a form that will allow the user to send a message/query/feedback etc in the form of email to the Alumni. It also contains a google map to locate the headquarters.
- **Page Name:** *sendemail.php*
 This file contains the main php code for sending email. It also contains the code that will display an alert message after a message has been sent.
- **Page Name :** *style.css*
 This file contains css code for styling the frontend of contact us page.
- **Folder Name :** *phpmailer*
 Files within folder : *Exception.php* , *PHPMailer.php* , *SMTP.php*

PHPMailer is the classic email sending library for PHP. It supports several ways of sending email messages such as mail(), Sendmail, qmail, and direct dispatch to SMTP servers. In addition, it provides a list of advanced features:

- SMTP authentication
- secure/MIME encryption
- support of TLS and SSL protocols
- HTML content along with plain text
- multiple fs, string, and binary attachments