R.V. COLLEGE OF ENGINEERING

BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belgavi)

RV ALUMNI PORTAL

PROJECT REPORT

Submitted by:

Utkarsh Jain (1RV13IS059)

Under the Guidance of

Dr. Cauvery N K
HOD,
ISE, RVCE

Prof. Priya
Assistant Professor,
ISE, RVCE

in partial fulfillment for completion

of

Database Management Systems Laboratory

Bachelor of Engineering

In

Department of Information Science & Engineering

R.V. COLLEGE OF ENGINEERING
BANGALORE - 560059
(Autonomous Institution Affiliated to VTU, Belgaum)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the project work titled **'RV Alumni Portal'** is carried out by **Utkarsh Jain(1RV13IS059),** who is a bonafide student of R.V College of Engineering, Bangalore, in partial fulfillment for the completion of **Database Management Systems (12IS64),** the requirement for the award of degree of **Bachelor of Engineering** in **Department of Information Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2015-16**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed by the institution for the said degree.

Signature of
Prof. Priya D.
Assistant Professor,
ISE, RVCE

Signature of
Dr. Cauvery N K
HoD,
ISE, RVCE

Semester End Examination

Name of Examiners	**Signature with date**
1	
2	

R.V. COLLEGE OF ENGINEERING
BANGALORE - 560059
(Autonomous Institution, Affiliated to VTU, Belgaum)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

DECLARATION

I, **Utkarsh Jain**, student of seventh semester B.E., **Information Science & Engineering,** hereby declare that the project titled **"RV Alumni Portal"** has been carried out and submitted in partial fulfillment for the completion **Database Management Systems (12IS64),** the requirement for the award of degree of **Bachelor of Engineering** in **Department of Information Science & Engineering.**

Place: Bengaluru

Date:

Utkarsh Jain
1RV13IS059

ACKNOWLEDGMENT

This project was the result of the encouragement of many people who helped in shaping it and provided feedback, direction and valuable support. It is with hearty gratitude that the contributions to this project are acknowledged.

Would like to thank the internal guide and lab-in-charge **Prof. Priya D,** Assistant Professor, Department of Information Science & Engineering, RVCE, for the guidance, suggestions in the area of improvement, and implementation of the project.

Also grateful to the HOD, **Dr. Cauvery N K**, Department of Information Science & Engineering, RVCE, for permitting to take up this project and for encouragement. Thank the Principal, **Dr. K N Subramanya**, who has always been a great source of inspiration. Thank RSST, for the infrastructure and facilities provided that helped in the completion of the project successfully.

Last, but not the least, would like to thank family and friends who provided with valuable suggestions to improve the project.

Utkarsh Jain
1RV13IS059

ABSTRACT

R.V. Alumni Portal is a project that intends to help the institute strategically build and manage its alumni network, by facilitating engagement, community-building, networking, communications and giving back. The data pertaining to the alumni is collected and made use of to generate statistics which could be of immense importance for the institute. This data is also continuously updated. The data is centralized and combined with a host of exciting front-end member modules and time-saving, back-end administration tools. The purpose of the project is to start creating value for the alumni community of the college and for the college itself. The portal facilitates the creation of a list of events to be conducted in college, which can be displayed to the alumni. The alumni could also be notified via mail in case of an upcoming event or even for some other relevant purpose.

The system can be used to generate statistics of alumni precisely the comparison or determination of the number of students opting for higher studies vs the number of students going for a job immediately after graduation. The portal mainly aims to serve as a communication channel between the institute and the alumni.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Purpose:

The purpose of the proposed system is to help the the institute strategically build and manage its alumni network, by facilitating engagement, community-building, networking, communications and giving back. The data pertaining to the alumni is collected and made use of to generate statistics which could be of immense importance for the institute, the alumni are also notified via mail in case of an event. The proposed system will minimize the burden on the institute to manage its vast alumni network.

1.2 Scope:

This system has its scope with every department in the college as every one of them needs to make a database of its alumni and manage the alumni network. They have to keep track of all the students who graduate from the college. As the time passes the network becomes larger and larger. Also each alumnus either goes for higher studies or some profession after graduating from the college. The portal would help keep an account of the alumnus's progress. The details pertaining to his profession or qualification would be updated automatically. The system will help in a way that would break boundaries in gathering and maintaining alumni details and also help the alumni keep in touch with each other.

1.3 Motivation:

The motivation for the project came from understanding all the efforts that have to be put in by the members of the staff to accumulate all essential details of alumni. Apart from the tight schedule they have the extended pressure of keeping an account of all the alumni. The Alumni Portal reduces the efforts of the staff by automating the process of acquisition and updating of alumni details. The portal also eases the process of keeping-in-touch for the alumni as they would be able to know the details of the other alumni and also may be able to contact each other.

1.4 Literature Survey:

Having staff members to manually collect the individual details of the graduated students and record it into registers/excel sheets is the traditional practice in our institute. This practice of manually gathering the data of alumni from various sources such as social media websites like Facebook or by directly contacting the alumni is a tedious process. Also the staff needs to make sure that details are up-to-date. This requires a lot of effort on behalf of the staff and consumes a lot of time as well. At the same time, it does not help the alumni in any way to know about their fellow batch-mates. The Alumni won't be able to contact their batch-mates as and when required. The manual storage of alumni information would also result in wastage or inefficient use of the storage space.

To overcome all these issues one can go for an automated software that can gather the details of the alumni and constantly update it. The proposed system aims to provide a new service, so that the staff can store as many alumni details as required.

SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 Overall Description

The proposed system has been designed to help the members of the staff to strategically build and manage its alumni network, by facilitating engagement, community-building, networking, communications and giving back. The data pertaining to the alumni is managed and used for the purpose of statistics generation and sending notification(s). The portal also eases the process of keeping-in-touch for the alumni as they would be able to know the details of the other alumni and also may be able to contact each other.

2.2 Specific Requirements

The requirements of the database management information system are to develop:

- A web based front end for administrator to verify alumni account, update alumni details and update his/her own password.
- A web based front end for the alumni to register on the database, update password and view list of events in college and other alumni details.
- A web based front end for the admin to send notification to alumni and generate necessary stats.
- A graphical representation of the various attributes of alumni such ae year of graduation, branch, current profession and qualifications.

2.2.1 Functionality

The two main user groups and access levels are:

- Alumni (Normal access level)
- Admin (Privileged access level)

Therefore, the requirements could be efficiently analyzed depending on the user group and the functionalities they should be allowed to perform.

2.2.1.1 Functional Requirement –Alumni

- The Alumnus needs to furnish some basic details about him such as his branch, USN and year of passing that will be verified.

- The Alumnus once added to the database can view the details of the other alumni and could also get in touch with them.
- The Alumnus can view the list of events to be conducted in the college.
- The Alumnus can also send his details for updating.

2.2.1.2 Security Requirements

Ensuring that only the alumni and admin can access the database is of paramount importance. Alumni can use the software using their username and password. Each Alumnus has the privileges to access, modify and update only the content of certain tables in the database. However, the admin being the one with highest privilege has all the access, insert and update privilege. The details of all the alumni are stored safely and securely in the database. The details of the alumni can't be altered or modified anyhow by anyone other than the admin or the alumni himself.

2.2.2 Performance Requirement

The PCs used must be at least Pentium 4 machines so that they can give optimum performance of the product.

In addition to these requirements, the system should also embrace the following requirements:

Reliability: The system should have little or no downtime and be able to handle multiple concurrent users.

Ease of Use: The general and administrative views should be easy to use and intuitive. Quick access to documentation and a help page should be provided.

System and Browser Compatibility Testing: The system should be accessible on the following browsers - Microsoft Internet Explorer 9.0+, Google Chrome and Mozilla 3.6+.

2.2.3 Design Constraints

The designers must design the database is such a way that any change in the information should be updated and saved effectively in the database despite the fact that multiple faculties are accessing the database.

2.2.4 Hardware Requirement

Users Side:

- Operating System: Ubuntu 12.04+, Windows XP, Windows 7(or higher), Mac OS X.
- Processor: Pentium 4.0 or higher
- RAM: 256 MB or more

- Hard Drive: 10 GB or more

2.2.5 Software Requirement

- Front End: HTML, CSS, JavaScript, Ajax and PHP (Personal Home Page)
- Back End: PHP, MySQL
- Operating System: Ubuntu 15.04/Windows(7 or higher)
- Web Server: XAMPP

2.2.6 Interfaces

Two types of interfaces are provided in this Project. Alumni are authorized to create their account, change their password and view other alumni details. The admin has to verify the alumnus's account, update certain details and can delete their account also.

2.2.6.1 User Interfaces

The database designed should be very easy to use and user friendly.

- Only the admin can approve the approve. The admin needs to verify an alumni account so that an alumnus is added to the database system.
- The Admin first needs to verify that the person is an alumnus of the college after which he/she can be added.
- The Admin adds or updates the details of the alumnus from time to time and he could do it either manually or this process may be automated.
- The Admin has the authority to send mail notifications to the alumni informing them about the upcoming events to be conducted in the college.

2.2.6.2 Communication Interfaces

Communication between database and front end pages is through PHP, which is auto configured application for virtual server in a computer which is automatically configured for XAMPP Server. XAMPP server is an open-source software powered with database usage which is connected with frontend with a connection string used in PHP pages. There are two types of authentication that are administrator and SQL server. It has inbuilt API for connectivity.

HIGH LEVEL DESIGN

3.1 Design Considerations

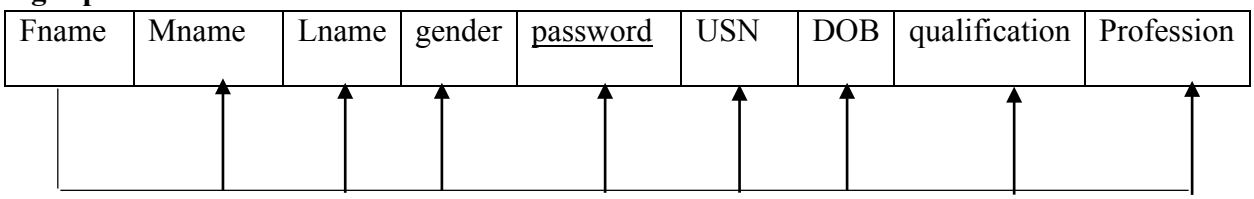
3.1.1 Assumptions and Dependencies

3.1.1.1 Assumptions

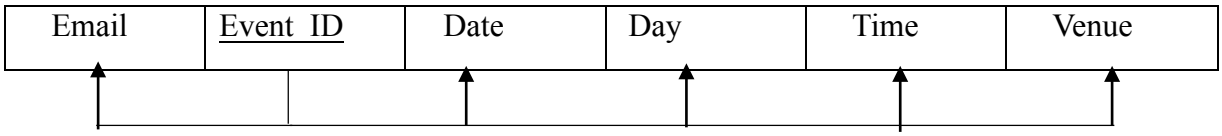
- It is assumed that admin can only add, update and delete from the database.
- It is assumed that only admin can verify a candidate.
- The Alumni is added to the portal only after the verification by admin is complete.

3.1.1.2 Functional Dependencies

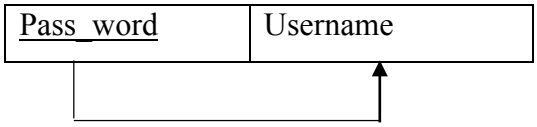
Signup



Events



Adminlogin



Update

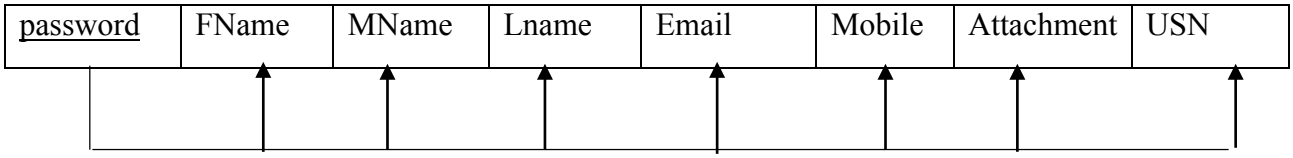
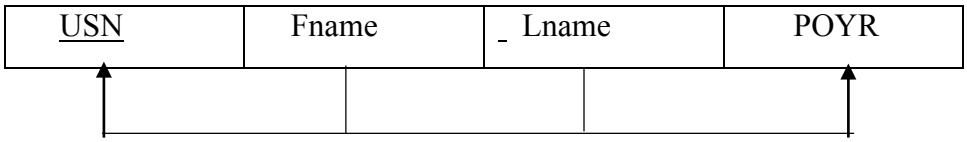
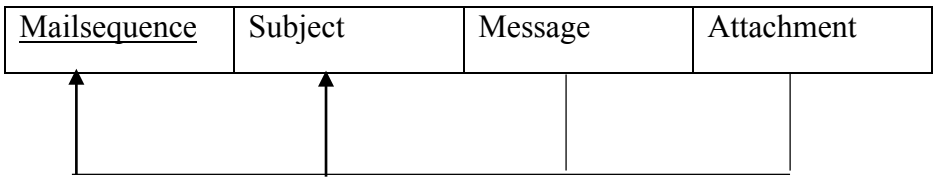


Figure 3.1 Functional Dependency

Mastertable



Tempmail



Gallery

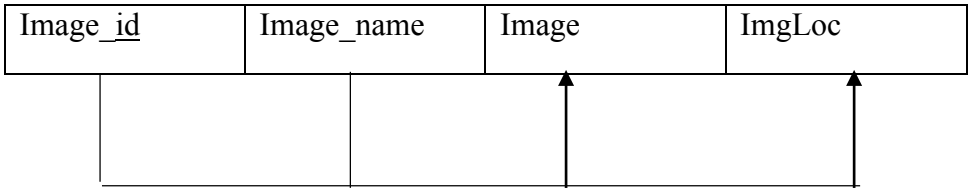


Figure 3.1 Functional Dependency (contd.)

3.1.2 General Constraints

General Constraints

The design involves the production of technical and visual prototypes. This stage has some non-technical aspects such as gathering of web content; content can be one of the biggest problems in web projects. For the server side programming and other technical aspects of the design emphasis will be laid on such design concepts. The goal is to make the system easier to adapt, enhance, test and use. Some of the general constraints are:

1. Clarity of the information

Each information inserted in the project shall be clear, without ambiguity. Each Alumnus has unique information and Id that should be stored and retrieved as and when required.

Also only the admin has the right to modify or edit any information pertaining to the alumnus and the authority to make use of the alumnus's information for the purpose of statistical analysis. The admin also has unique login credentials for the purpose of logging into the portal and the uniqueness should be preserved at any cost.

2. Structuring

For guaranteeing reusability of data and its information for different views and layouts the structuring of data and separation of content, layout, and structure should be supported in future.

3. Verifiability of the information

Each item of information inserted in the project shall be verifiable.

3.2 System Block Diagram

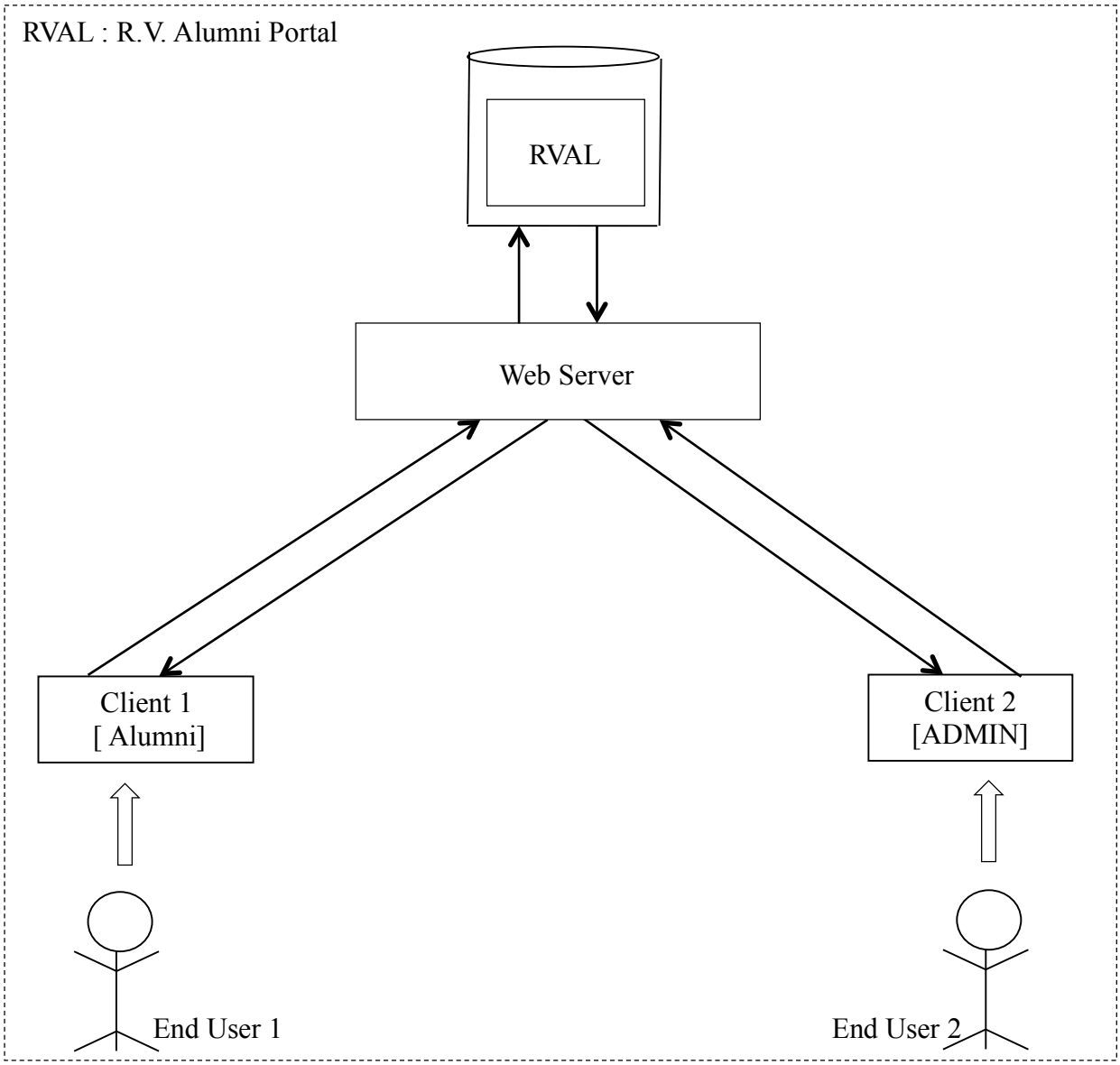


Figure 3.2 System Block Diagram

This diagram shows the principal parts of the R.V. Alumni Portal and the interactions between the different modules.

3.3 Entity Relationship Diagram

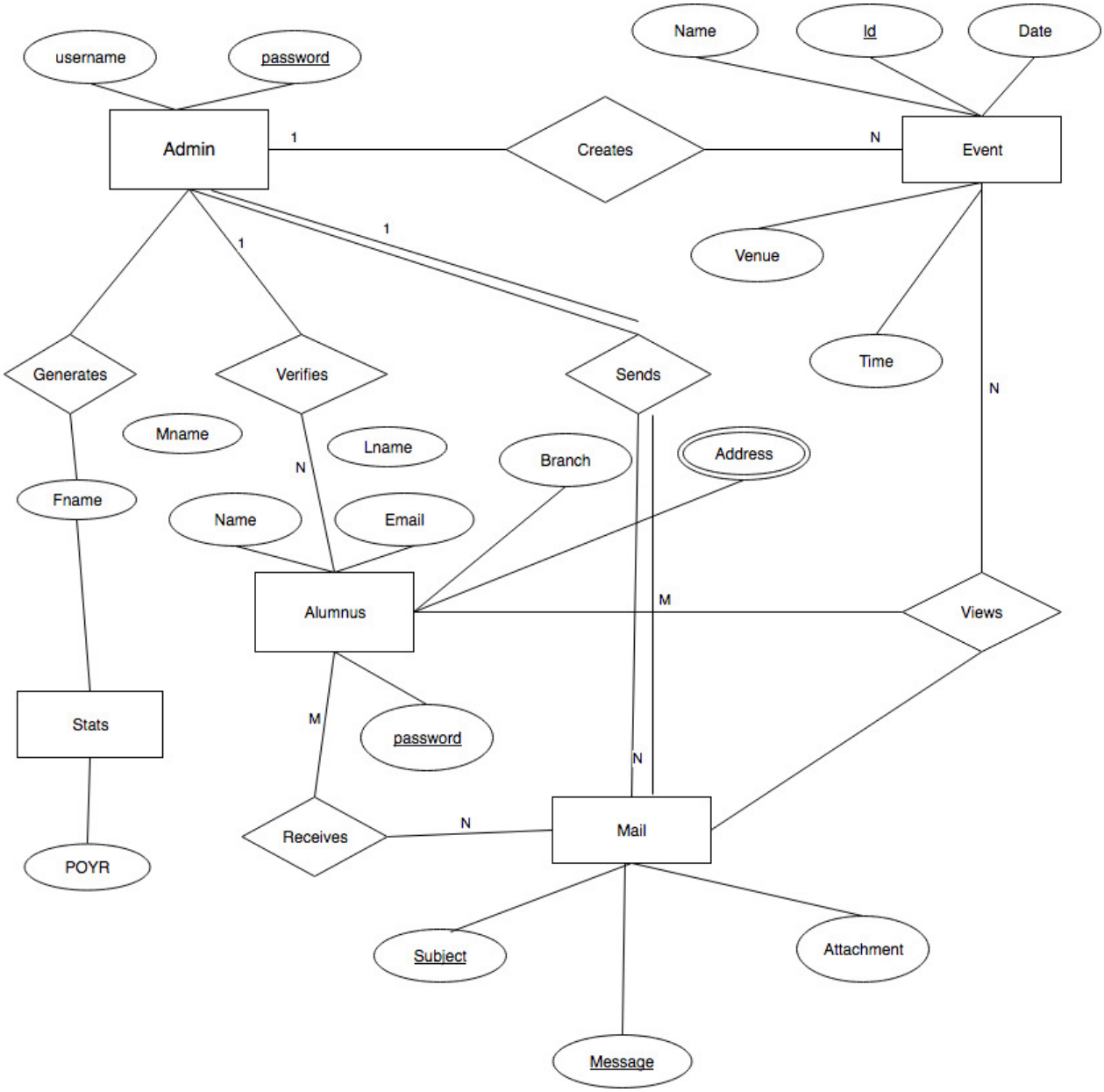


Figure 3.3 E-R Diagram

3.4 Data Flow Diagram

3.4.1 Level 0 Diagram

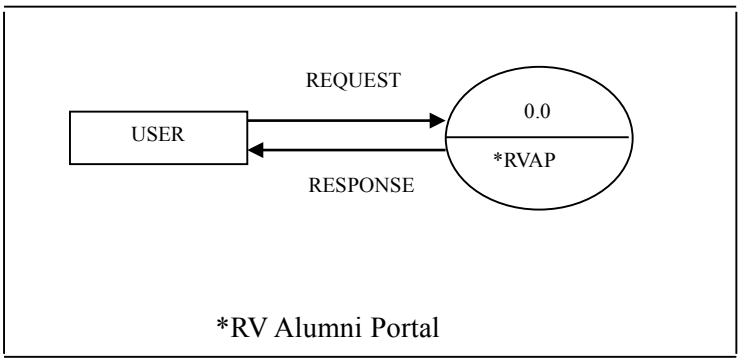


Figure 3.4 Dataflow diagram Level-0

- The User sends a request which is to be authenticated, which if allowed for that type of user and if that kind of request exist is allowed and converted into a query.
- If that requests is not authenticated a response to the GUI or user end is generated for e.g. validations on fields.
- Query hence is executed on the server side and an output is generated and displayed to the user.

3.4.2 Level 1 Diagram

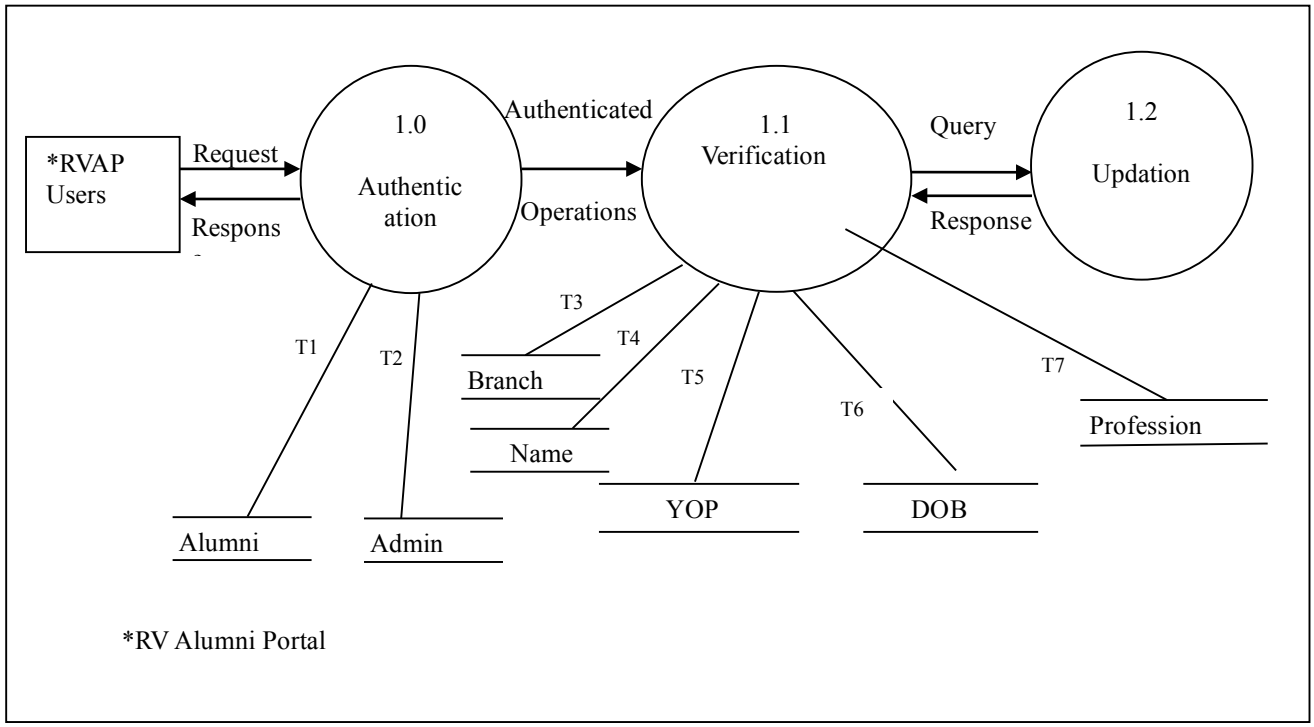


Figure 3.5 Dataflow diagram level 1

T1: [Alumni Id]-It contains Alumni id and password for login.

T2: [Admin]-It contains login for administrator.

T3: [Branch]-Contains various Branch details.

T4: [Name]- Contains various names of the persons interested to sign-up.

T5:[YOP]-Contains the passing out year for all the alumni.

T6:[DOB]-Contains date of birth of the alumni.

T7:[Profession]- Contains the details of profession of the alumni.

3.5 Schema Diagram

Signup

Fname	Mname	Lname	gender	password	USN	DOB	qualification	Profession

Events

Name	Event_ID	Date	Day	Time	Venue

Adminlogin

Pass_word	Username

Update

password	FName	MName	LName	USN	Email	Mobile	Achievement

Mastertable

Fname	Lname	USN	POYR

Tempmail

Mail Sequence	Subject	Message	Attachment

Gallery

Image_Id	Image_name	Image	ImgLoc

Figure 3.6 Schema Diagram

3.6 Definition of tables

ADMINLOGIN TABLE

```
CREATE TABLE  ADMINLOGIN
(
        Uname  varchar (20) NOT NULL,
        Password varchar(20) PRIMARY KEY
);
```

MASTERTABLE TABLE

```
CREATE TABLE MASTERTABLE
(
        USN varchar (20) PRIMARY KEY,
        Fname varchar (20) NOT NULL,
        Mname varchar (20),
        Lname varchar (20) NOT NULL,
        POYR varchar(20) NOT NULL
);
```

SIGNUP TABLE

```
CREATE TABLE SIGNUP
(
        Password varchar (10) PRIMARY KRY,
        Fname varchar (20) NOT NULL,
        Mname varchar (20),
        Lname varchar(10)NOT
        NULL,
        DOB date NOT NULL,
        Gender varchar(6) NOT NULL,
        Branch varchar (20) NOT NULL,
        Address varchar (50) NOT NULL,
        Email varchar (20) NOT NULL,
        Mobile int NOT NULL,
        USN int(20) REFERENCES MASTERTABLE(USN) ON DELETE SET NULL,
 );
```

EVENTS TABLE

```
CREATE TABLE EVENTS
(
   EventId varchar(10) PRIMARY KEY,
   Date date NOT NULL,
   Day varchar(10) NOT NULL,
   Venue varchar(20) NOT NULL,
    Name VARCHAR(20) NOT NULL
);
```

UPDATE TABLE

```
CREATE TABLE UPATE
(Password varchar(10) PRIMARY KEY,
 Email varchar(20) ,
 Mobile int,
 Achievement varchar(50)
 );
```

-

TEMPMAIL TABLE

```
CREATE TABLE TEMPMAIL
(
 Mailsequence varchar(10) PRIMARY KEY,
 Subject varchar(30) NOT NULL,
 Message varchar(100) NOT NULL,
 Attachment blob
);
```

-

GALLERY TABLE

```
CREATE TABLE GALLERY
(
 Image_Id varchar(10) PRIMARY KEY,
 Image_name varchar(10) NOT NULL,
 ImageLoc varchar(20) NOT NULL,
 Image blob
 );
```

Chapter 4

IMPLEMENTATION

4.1. Selection of the platform

The platform selected for the project is Ubuntu which supports XAMPP package as it is compatible with PHP in the front end, MYSQL in the back end and is supported by the Apache and MYSQL Server.

4.2 Selection of the programming language

Web applications are an extension of a web server PHP (Personal Home Page). Web applications are either service oriented or presentation oriented. A presentation oriented web application produces interactive web pages containing markup languages like (XML and HTML) and dynamic content in response to requests (PHP). PHP is used in programming between back end and front end.

4.2.1 Front End: PHP

The programming language used for the development work is PHP. The reason for selection of this language includes among many others the following few: -

- Open Source, PHP is completely free.
- PHP can be easily embedded directly into HTML.
- Platform independent can run on Windows, Linux or Mac servers.
- Run faster on the internet and easily integrate AJAX, Callback etc.
- Interfaces very easily with Apache/MySQL
- Lots of good books and on-line help.
- It's available with documentation in many languages.
- Easy to learn as compared to many other scripting languages. It has a syntax that is easy to parse and is actually rather human-friendly.
- Lots of hosting services have it ready to use, no special configuration.
- Pretty easy to access other web-based tools through PHP.
- Lots of good source code out there to use and/or learn from, as well as many useful libraries for working with PDFs, graphics, etc.

4.2.2 BACK END: MYSQL

MySQL is a relational data base management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

- MySQL is an open source tool.
- MySQL is a popular choice of database for use in web applications, and is a central component of the widely-used XAMPP web application software stack — XAMPP is an acronym for "Cross-Platform, Apache, MySQL/MariaDB, Perl and PHP".
- MySQL is primarily an RDBMS and therefore ships with no GUI tools to administer MySQL databases or manage data contained within.

MySQL implements the following features, which some other RDBMS systems may not:

- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

4.2.3 Server: XAMPP server

- It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purpose.
- Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.
- XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.
- It also comes with a number of other modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more.
- Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another.

4.2.4 JavaScript: Ajax, JQuery

- Ajax is used to synchronously send requests to the server, which is coded using PHP, and getting and displaying to the users the results from the server.

- Major advantage achieved using Ajax library is the avoidance of reloading the page on submission of the form. This brings in a smoother and quicker experience to the user.
- JQuery is used to implement animations corresponding to various user actions. JQuery is also used in the validation of the user data entered in the form.
- Also many user actions can be detected and appropriate actions can be taken using on Click and other listeners in the library.

4.3 Module Description

The two modules are identified on the basis of various users of the portal.

1. Administrator
2. Alumni

Administrator module contains various sub-modules. Each of these modules corresponds to the various functionality provided by them.

1. Add Alumnus- Add an alumnus to the database.
2. Add Details- To add details about the alumnus.
3. Update details- Generate various reports pertaining to alumnus details.
4. Remove Alumnus- To remove the alumnus from the database.

Alumni module contains various sub-modules. A brief functionality of each sub-module is given below.

1. Register – The Alumnus needs to furnish some basic details to the admin so that he cab be verified.
2. Check details- To check for the alumni details.
3. Update- The Alumnus can send the details of his current status to the admin via mail or an excel sheet.
4. Guidance- Allows the Alumnus to offer students with respect to their academics or educate them about the changing trends in the industry.

4.3.1 Login and Password Details

The login and password details of the modules are given as follows:

1. **Module: Administrator**

   Login: admin

   Password: 12345

2. Module: Alumni

Login: Amar

Password: amar123

Once the administrator login is done, any number of additional administrator logins or alumni logins can be added using the options available in the dashboard.

4.3.2 Variables Used

1. Module: Administrator

i. $con - To connect with the MYSQL database.

ii. $op – Text-field to hold the old password received from the user.

iii. $uname – To hold the username entered by the user.

iv. $pwd – To hold the password entered by the user.

v. $_SESSION['start'] – Session variable to hold the start time of the session

vi. $_SESSION[‘username’] – Session variable to hold username logged in.

vii. $sql and $query– To hold the query to be executed on the database.

viii. $result and $res - To hold the result of the MySQL query execution.

ix. $dob – To hold the date of birth of the alumnus.

x. $address – To hold the current address of the alumnus.

xi. $poyr – To hold the year of passing of the alumnus.

xii. $photo – To hold the photo of the alumnus.

xiii. $email – To hold the email id of the alumnus.

xiv. $br – To hold the branch of the alumnus.

xv. $gender – To hold the gender of the alumnus.

xvi. $contact – To hold the contact details of the alumnus.

xvii. $session_life – To measure the session time in milli- seconds.

xviii. $_SESSION['subject'] – To hold the subject selected. (session variable)

xix. $fname- To hold the first name of the alumnus.

2. Module – Alumni

i. $f – Text-field for the first name.

ii. $m – Text-field for the middle name.

iii. $l – Text-field for the last name.

iv. $username – To hold the username of the alumnus.

v. $password – To hold the password of the alumnus.

vi. $found – To hold a counter variable.

vii. $result – To hold the result of query execution.

viii. $q – To hold the qualification details.

ix. $table – Hold the Alumni details.

x. $p – To hold the profession details.

4.3.3 Function Description

1. Update Qualification and Profession details:

```
foreach ($all_result as $row) {
        $oq=$row[12];
        $op=$row[13];
      foreach ($all_result1 as $row1) {
            if($row[6]==$row1[0]){
              echo "Updating Details of ".$row[6]."<br>";
              echo "Updating  Qualification =".$oq."     by ".$row1[1]."
   Profession=".$op." by ".$row1[2]."<br><br><br><br>";
       $em1=$row[6];
       $em2=$row1[0];
        echo "success";
        $q=$row1[1];
        $p=$row1[2];
```

```
            $con = mysqli_connect("localhost", "root", "");
             mysqli_select_db($con,"alumni");
             $query="update signup set qualify='{$q}',
             profession='{$p}'  where email='{$em2}'";
             //echo $query;
             mysqli_query($con,$query);
             echo("  Updation success" );
            }
     }
}
```

2. Edit Admin password:

```
$uname=$_COOKIE["user"];


$op=$_POST["textfield"];
$np=$_POST["textfield3"];
$cp=$_POST["textfield2"];

//////////////////
$mysqli= new mysqli("localhost", "root", "", "alumni");

if ($mysqli->connect_errno)
{
      echo "Failed to connect to MySQL: (". $db->connect_errno . ") " . $db->connect_error;
}

$sql="select * from  signup where email ='{$uname}' and  password= '{$op}'";
//echo $sql;
$found=0;
$result=$mysqli->query($sql);
```

```
if($result ->num_rows == 1)
{
$found=1;
}
if($np==$cp  && $found==1)
{
$con = mysqli_connect("localhost", "root", "");
mysqli_select_db($con,"alumni");
$query="update signup set password='{$np}'  where email='{$uname}'";
//echo $query;
mysqli_query($con,$query);
//echo("Password  Updated Successfully" ."<h1>");
header('Location:sucssupdt.php');
//echo("<h4 align=center><a href=sucssupdt.php><br></a></h4>");
}
else
{
//                  echo "<h1 align=center> Some thing Wrong Please try again </h1>";

                                        header('Location:unsucssupdt.php');
}
```

Chapter 5

TESTING

Software Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer software. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors.

5.1 Unit Testing

Unit testing is a software verification and validation method in which a programmer tests if individual units of source code are fit for use. Some of the tests performed in the project are insert, delete, retrieve and modify.

5.1.1 Unit Test Case1

Serial No. of test case:	1
Name of test:	Check or login test
Item / Feature being tested:	Submit button
Sample Input:	Admin ID=’admin’, PASSWORD=’12345’ Upon mouse click on button.
Expected output:	Admin dashboard page appears if login information is correct if not, the login page reappears.
Actual output:	Admin Dashboard.
Remarks:	Test succeeded

Table 5.1 Unit test case for Login Information Check operation

5.1.2 Unit Test Case2

Serial No. of test case:	2
Name of test:	Verifying Alumnus account by admin
Item / Feature being tested:	Verifying Alumnus
Sample Input:	Table of candidates appear whose verification is pending, verify Upon mouse click on button.
Expected output:	Alert showing Alumni Verification successful..
Actual output:	Alert that shows Alumni Verification successful.
Remarks:	Test succeeded

Table 5.2 Unit test case for verifying an Alumni

5.1.3 Unit Test Case 3

Serial No. of test case:	3
Name of test:	Editing password for Alumni
Item / Feature being tested:	Change password button
Sample Input:	Old password of the Alumni, New password and confirm password
Expected output:	Alert showing password has been successfully updated
Actual output:	Alert that shows password updated successfully
Remarks:	Test succeeded

Table 5.3 Unit test case for Edit password of Alumni

5.1.4 Unit Test Case 4

Serial No. of test case:	4
Name of test:	Searching Alumni through Alumni account
Item / Feature being tested:	Search button
Sample Input:	Select Year or First name
Expected output:	List of Alumni sorted according to the selected option
Actual output:	List of Alumni
Remarks:	Test succeeded

Table 5.4 Unit test case for Search Alumni

5.2 System Testing

5.2.1 System Test Case 1

Sl No. of test case:	1
Name of test:	System overall test
Item / Feature being tested:	Entering and retrieving data
Sample Input:	Data entered through forms and retrieved in the form of Alumni table
Expected output:	Updated data displayed in proper format
Actual output:	Updated data displayed in proper format
Remarks:	Test succeeded

Table 5.5 System test case for overall system check

Chapter 6

RESULTS

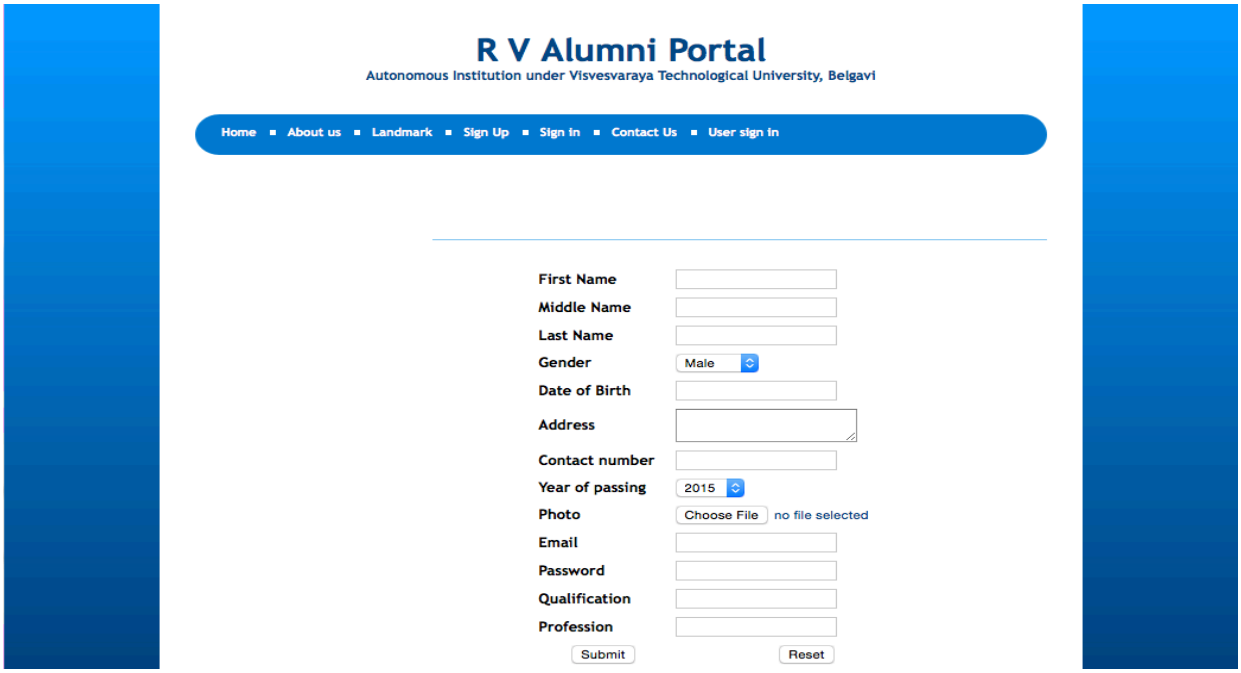


Figure 6.1 Signup page for Alumni

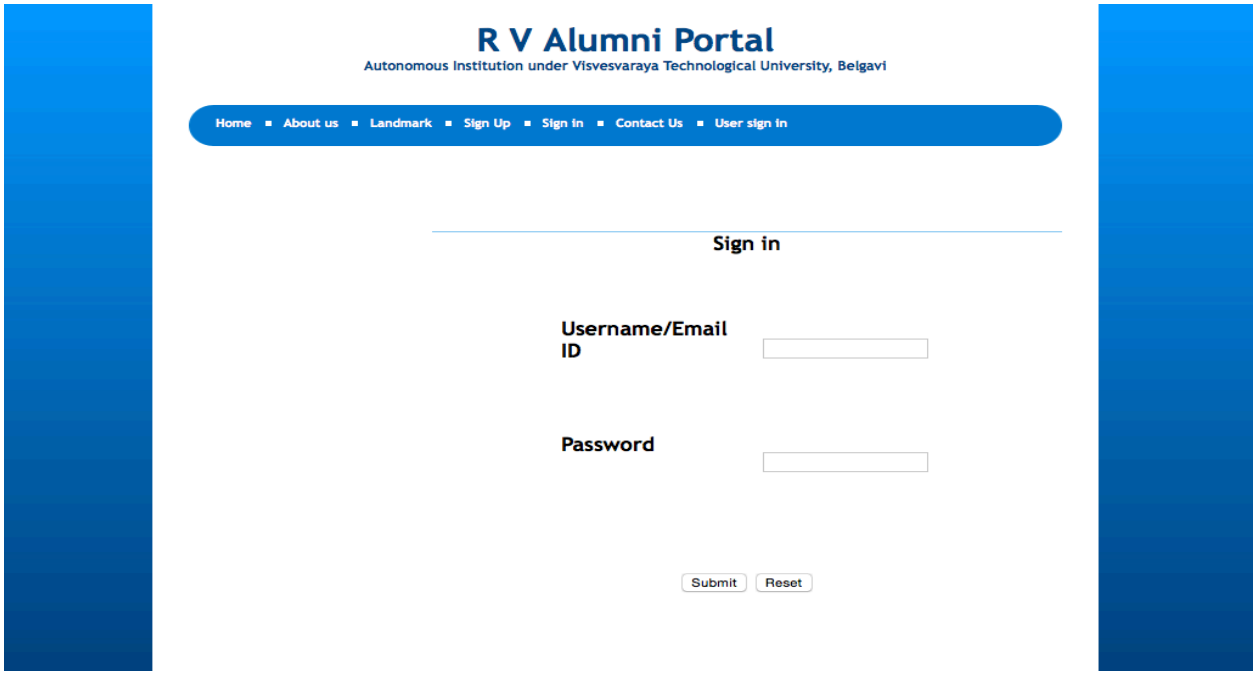


Figure 6.2 Sign-in page for Admin

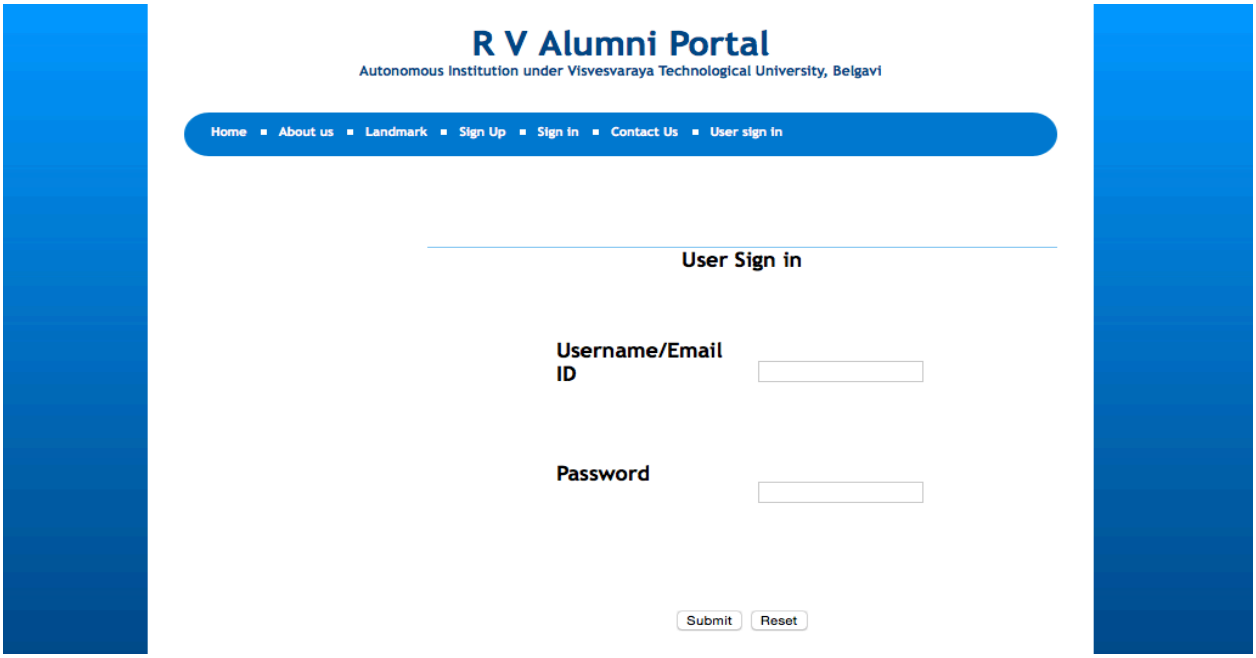


Figure 6.3 Sign-in page for Alumni

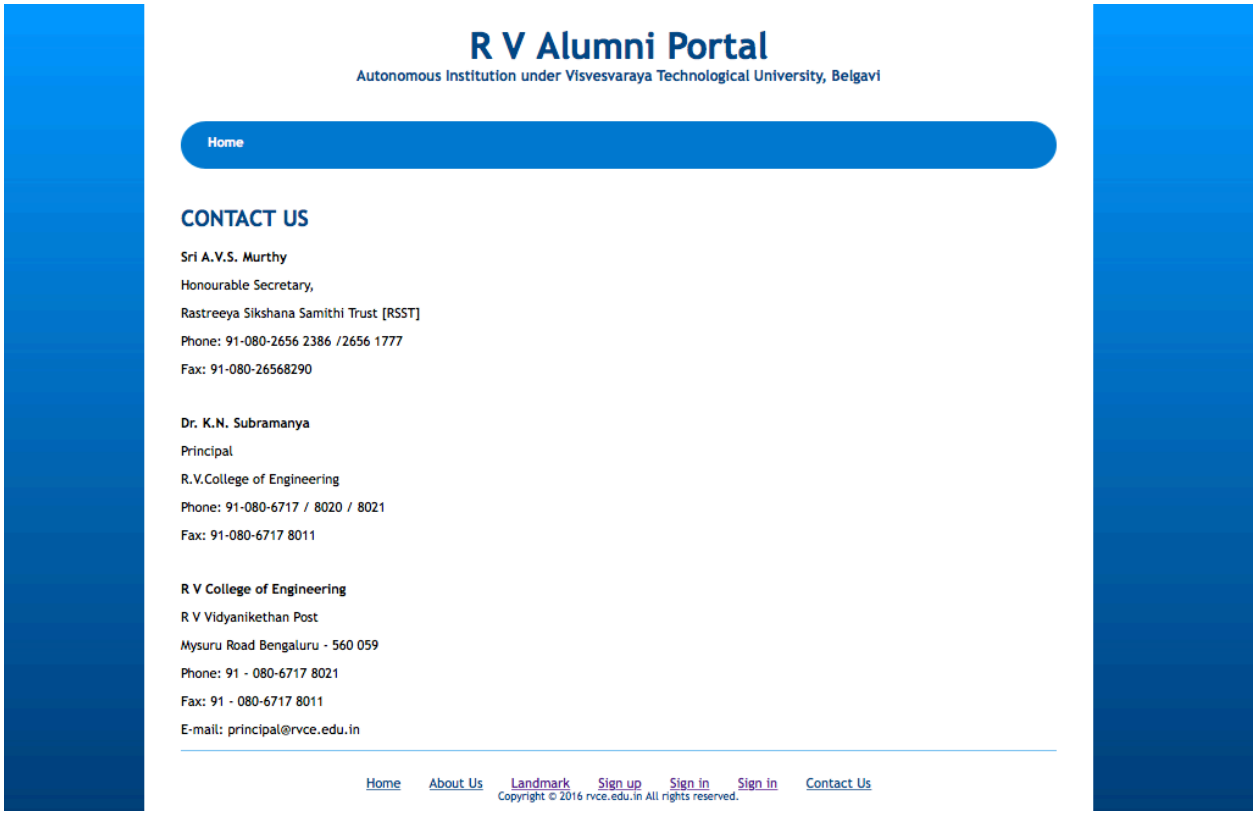


Figure 6.4 Contact Us page

6.1 Description of Menus and Options

Administrator module contains various menus and options. Each of these options correspond to the various functionality provided by them: -

1. Add Alumnus- Add an alumnus to the database.
2. Add Details- To add details about the alumnus.
3. Update details- Generate various reports pertaining to alumnus details.
4. Remove Alumnus- To remove the alumnus from the database.

Alumni module contains various sub-modules. A brief functionality of each sub-module is given below: -

1. Register – The Alumnus needs to furnish some basic details to the admin so that he cab be verified.
2. Check details- To check for the alumni details.
3. Update- The Alumnus can send the details of his current status to the admin via mail or an excel sheet.
4. Guidance- Allows the Alumnus to offer students with respect to their academics or educate them about the changing trends in the industry.

6.2 Sequence of Screens and Actions

1. **Signing up** – The Alumni needs to signup first, he/she needs to create an account by filling in his/her basic details.

2. **Admin login** – The Admin then logs into his/her account and then a list of candidates is seen by him/her if he/she selects the verify option. The verification for the candidates is done and is successful if the candidate was truly a student of the college.

3. **Sign-in by Alumnus –** Once an alumnus's account has been successfully created, the Alumnus may login by using his credentials. The Alumnus then sees a host of options including updating his/her profile, viewing/searching the other Alumni.

4. **Search Alumni-** The Alumnus then can search other Alumni by either their year of passing or by their First Name. A detailed list of Alumni appears before the Alumnus sorted either by Year of graduation or First Name.

Chapter 7

CONCLUSION

The RV Alumni Portal mainly mainly reduces the efforts required by the members of the staff to gather and store the Alumni details. It also helps them to be able to easily update important details of the Alumni such as qualification or profession. It provides an opportunity to the Alumni to get to know about their batch-mates and also stay in touch with them. Thus this portal helps the institute strategically build and manage its alumni network, by facilitating engagement, community-building, networking, communications and giving back.

7.1 Future Enhancement

The RV Alumni Portal is built so as to minimize the efforts to manage the alumni network of the institute. The constraints of the current implementation do not allow it to automatically gather data from social media platform such as Facebook or WhatsApp. This can be overcome with the help of a web-crawler that crawls through Facebook data and makes our job easier. The updating would be extremely easy and could be automated.

REFERENCES

[1] Elmasri and Navathe, *Fundamentals of Database Systems,* 5th edition, Addison-Wesley, 2007.

[2] Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, 3rd edition, McGraw-Hill, 2003.

[3] Silberschatz, Korth and Sudharshan, *Database System Concepts*, 5th Edition, McGraw-Hill 2006.

[4] Mike Shema *, Web Application Security for Dummies*, John Wiley & Sons, Ltd (2011)

[5] Carles Mateu, *Introduction to Web Applications Development,* Free Technology Academy - Fundació per a la Universitat Oberta de Catalunya (February, 2010)

[6] C.J.Date, A.Kannan, S.Swamynatham, *A Introduction to Database Systems*, 8th edition, Pearson Education, 2006.

Appendix-A

LIST OF ACRONYMS

1. XAMPP: Cross-Platform, Apache, MySQL/MariaDB, PHP and Perl
2. SRS: Software Requirement Specifications
3. ER: Entity Relationship
4. OS: Operating System
5. RAM: Random Access Memory
6. SQL: Structured Query Language
7. PHP: Hypertext Preprocessor (Personal Home Page)
8. RDBMS: Relational Database management system

Appendix-B

SOURCE CODE

Code for sign-up page:

signup.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Free website template from 4Templates.com</title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="default.css" rel="stylesheet" type="text/css" />
<style type="text/css">
<!--
        @import url("layout.css");
.style10 {font-size: small; color: #000000; font-weight: bold; }
.style4 {       font-size: 18px;
        color: #000000;
}
-->
</style>
</head>
<script language="javascript">
function validateForm()
{

  var x=document.forms["form1"]["textfield5"].value;
  var phoneNum = new String();
  var regExpObj = /\d\d\d\d\d\d\d\d\d\d/;
  //alert(areaCode);
  if(regExpObj.exec(x) == null)
  {
   alert("Invalid Phone No !");
        //alert("not ok");
        return false;
  }

  var alphaExp = /^[a-zA-Z]+$/;
//      var alphaExp = /^[a-zA-Z]+$/;

/*  var y=document.forms["form1"]["textfield"].value;
alert(y)
        if(y.value.match(alphaExp))
        {
                return true;
        }
        else
        {
                alert("Sorry, only text");
```

```
                return false;
        }
*/

re = /^[A-Za-z]+$/;
if(re.test(document.forms["form1"]["textfield"].value))
{
//alert('Valid Name.');
return true;
}
else
{
alert('Invalid Name.');
return false;
}

 /* var y=document.forms["form1"]["textfield3"].value;
        var atpos=y.indexOf("@");
        var dotpos=y.lastIndexOf(".");
        if (atpos<1 || dotpos<atpos+2 || dotpos+2>=y.length)
  {
                alert("Not a valid e-mail address");
        return false;
  }*/



}
</script>
<body>
<div id="wrapper">
        <div id="header">
                <h1>mesalumni.com</h1>
                <h2>&amp;quot;arsie, awake stop not until the goal is reached&amp;quot;</h2>
  </div>
        <div id="main-menu">
                <ul>
                        <li class="first"><a href="#" id="main-menu1" accesskey="1"
title="">Home</a></li>
                        <li><a href="aboutus.html" id="main-menu2" accesskey="2"
title="">About us</a></li>
                        <li><a href="adlandmrk.html" id="main-menu3" accesskey="3"
title="">Landmark</a></li>
                        <li><a href="signup.html" id="main-menu4" accesskey="4" title="">Sign
Up</a></li>
                        <li><a href="signin.html" id="main-menu5" accesskey="5" title="">Sign
in</a></li>
                        <li><a href="contactus.html" id="main-menu6" accesskey="4"
title="">Contact Us</a></li>
                        <li><a href="usignin.html" id="main-menu7" accesskey="4"
title="">User sign in</a></li>
                        <li class="first"></li></ul>
```

```
 </div>
<div id="content">
                <div id="left">
                        <h2> </h2>
        </div>
                <div id="right">
                        <h2 class="bigger"> </h2>
        <div class="hr1">
                                <hr />
                        </div>
                        <form id="form1" name="form1" method="post" action="signup.php"
onsubmit="return validateForm()">
        <p align="center"> </p>
        <table width="411" height="363" border="0" align="center">
          <tr>
            <td width="157"><span class="style10">First Name</span></td>
            <td width="178"><input type="text" name="textfield" id="textfield" required/></td>
          </tr>
          <tr>
            <td><span class="style10">Middle Name</span></td>
            <td><input type="text" name="textfield2" id="textfield2" /></td>
          </tr>
          <tr>
            <td><span class="style10">Last Name</span></td>
            <td><input type="text" name="textfield3" id="textfield3" required/></td>
          </tr>
          <tr>
            <td><span class="style10">Gender</span></td>
            <td><select name="select2" id="select2">
              <option value="Male">Male</option>
              <option value="Female">Female</option>
            </select>                </td>
          </tr>
          <tr>
            <td><span class="style10">Date of Birth</span></td>
            <td><input type="date" name="textfield6" id="textfield6" required/></td>
          </tr>
          <tr>
            <td><span class="style10">Address</span></td>
            <td><textarea name="textfield4" id="textfield4"></textarea></td>
          </tr>
          <tr>
            <td><span class="style10">Contact number</span></td>
            <td><input name="textfield5" type="text" id="textfield5" maxlength="10"
required/></td>
          </tr>

          <tr>
            <td><span class="style10">Year of passing</span></td>
            <td><select name="select" id="select">
              <option value="2015">2015</option>
              <option value="2014">2014</option>
```

```
    <option value="2013">2013</option>
    <option value="2012">2012</option>
    <option value="2011">2011</option>
   </select>                </td>
  </tr>
  <tr>
   <td><span class="style10">Branch</span></td>
   <td><select name="select3" id="select3">
    <option value="CS">CS</option>
    <option value="EC">EC</option>
    <option value="CIVIL">CIVIL</option>
    <option value="MECH">MECH</option>
                </select></td>
  </tr>
  <tr>
   <td><span class="style10">Photo</span></td>
   <td><input type="file" name="textfield11" id="textfield11" required/></td>
  </tr>
  <tr>
   <td><span class="style10">Email </span></td>
   <td><input type="email" name="textfield9" id="textfield9" required/></td>
  </tr>

  <tr>
   <td><span class="style10"> Password</span></td>
   <td><input type="password" name="textfield10" id="textfield10" required/></td>
  </tr>
  <tr>
   <td><span class="style10">Qualification</span></td>
   <td><input type="text" name="textfield8" id="textfield8" required/></td>
  </tr>
  <tr>
   <td><span class="style10">Profession</span></td>
   <td><input type="text" name="textfield12" id="textfield12" required/></td>
  </tr>

  <tr>
   <td><div align="center">
    <input type="submit" name="button" id="button" value="Submit" />
   </div></td>
   <td><div align="center">
    <input type="reset" name="button2" id="button2" value="Reset" />
   </div></td>
  </tr>
 </table>
 <p> </p>
 <p align="center"> </p>
 <p> </p>
 <p> </p>
 <p> </p>
 <p> </p>
                </form>
```

```
                        <h2> </h2>
          </div>
        </div>
        <div class="hr1">
                <hr />
        </div>
        <div id="footer">
                <ul>
                        <li><a href="#" title="">Home</a></li>
                        <li><a href="#" title="">About us</a></li>
           <li><a href="#" title="">Landmark</a></li>
                        <li><a href="#" title="">Sign up</a></li>
       <li><a href="#" title="">Sign in</a></li>
                        <li><a href="#" title="">Contact Us</a></li>
                </ul>
                <p>Copyright &copy; 2016 mesalumni.com. All rights reserved. </p>
         </div>
</div>
</body>
</html>
```

signup.php

```
<?php


$mysqli= new mysqli("localhost", "root", "", "alumni");

if ($mysqli->connect_errno)
{
        echo "Failed to connect to MySQL: (". $db->connect_errno . ") " . $db->connect_error;
}




$f=$_POST["textfield"];
$m=$_POST["textfield2"];
$l=$_POST["textfield3"];
$g=$_POST["select2"];
$dob=$_POST["textfield6"];
$a=$_POST["textfield4"];
$c=$_POST["textfield5"];
$yop=$_POST["select"];
$photo=$_POST["textfield11"];
$e=$_POST["textfield9"];
$p=$_POST["textfield10"];
$p1=$_POST["textfield8"];
$p2=$_POST["textfield12"];
$p3=$_POST["select3"];

$sql="select email from  signup where email='{$e}' ";
```

```
//echo $sql;
$found=0;

$result=$mysqli->query($sql);

if($result ->num_rows == 1)
{
        $found=1;

}

if($found==0)
{

$mysqli= new mysqli("localhost", "root", "", "alumni");
if ($mysqli->connect_errno)
{
        echo "Failed to connect to MySQL: (". $db->connect_errno . ") " . $db->connect_error;
}

$con= mysqli_connect("localhost","root","");
mysqli_select_db($con,"alumni");
$query= "insert into signup values('$f','$m','$l', '$g', '$dob', '$a' , '$e' , '$yop', '$c', '$photo',
'$p','','$p1','$p2','$p3')";
//echo $query;
mysqli_query($con, $query);

//echo"<h3 align= center>Submitted Successfully </h3>";

header('Location:signupsubmit.php');
}
else
{
header('Location:unsucsssignup.php');


}
?>
```