1. INTRODUCTION

Resume is the first meeting between you and a prospective employer more often now than ever. So, how do you want to be remembered? Wrinkled and unorganized. Neat and structured. Long and boring. Precise and interesting. Companies do not have the time to interview every applicant that is interested in the job. If they did, there would not be a company to work for. They use an eliminating process. That's right - resumes.

When a job seeker wants to apply for a job online then generally he/she needs to attach his/her resume with the email. Online Resume Builder can be used in accordance with the requirements of the customers. Customers can customize their resumes with their choice of themes & details. The services are hard to be defeated by the competitors as the system is providing the customers exactly what they want.

Online Resume Building System provides the users the popular resume formats & a better way to show their resumes to the employers.

Purpose of Online Resume Builder is to provide a way to the customers to design their resumes according to their requirements.

- Creating resumes online.
- Customizing the look and details.

1.1 EXISTING SYSTEM

Present system is the manual system where every task is performed again & again. We need to manually change the format of the resume. Sending of resumes is done manually. All the activities in the system, which can be done by using automated methods, are being done manually. Since all tasks are being performed manually so it needs a lot of work force.

Disadvantages of Existing System

Since, our present system is a manual system it has several disadvantages described below

- Lack of immediate retrievals**: -**The information is very difficult to retrieve (download) and to find particular information. This results in inconvenience and wastage of time.
- Lack of immediate information storage: **-** The information takes time and efforts to be attached again & again with mails or to be sent by post.
- Lack of prompt updating: **-** Various changes to information are difficult to make if resume is sent.
- Error prone manual calculation: **-** Manual calculations are error prone and take a lot of time this may result in incorrect information.
- Preparation of accurate and prompt reports: **-** This becomes a difficult task as information is difficult to collect from various registers.
- Inability of modification of data
- Provides fixed format of resume
- Lack of immediate retrieval
- Inability of sharing the resume data

1.2 PROPOSED SYSTEM

Our proposed system provides automation to Resume Building. This system provides online storage/ updates and retrieval facility. This system promises very less or no paper work and also provides help to customers and viewers/employers. In this system everything is stored electronically so very less amount of paper work is required and information can be retrieved very easily without searching here and there into registers.

This system ensures certain features that are not available with present manual system. These are described below:

- Planned approach towards working: - The working in the system will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.
- Accuracy: - The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the server is accurate.
- Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
- No Redundancy: - In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.
- Immediate Retrieval of Information: - The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the viewer or customer requires.
- Immediate storage of information: - In manual system there are many problems to store & update the large amount of information.
- Easy to Operate: - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

2. LITERATURE SURVEY

2.1 RELATED WORK

Purpose of Online Resume Builder is to provide a way to the customers to design their resumes according to their requirements

Online Resume Builder is a system which allows the customers to design their resumes in accordance with their requirements.

System provides facilities like

- Customizing the resumes according to the user requirements.
- Editing the design.
- Providing an account to the user.
- Password recovery option.
- Choosing from latest professional designers.
- Viewed resume notification for the customers.
- Allowing users to give feedback.

Various operations done in the system are as follows

- Registering customers.
- Writing resumes.
- Editing in design.
- Keeping track of latest formats of resumes.
- Viewed resume notification.
- Browsing resume templates.
- Resetting password

2.2 SYSTEM STUDY

System study is a first stage according to the System development life cycle model.

This system study is a process that starts with the analyst.

To perform this study, it was necessary to understand how a basic online system works. Then, we apply the same procedure to this online resume builder. The next stages are to analyze the procedure and start coding process.

Analysis is a detailed study of the various operations performed by the system and the relationships within and outside of the system. One aspect of the study is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During studying data is collected on the available files, decision points and transactions handled by the present systems.

Logical system models and tools that are used in analysis. Training, experience and common sense are required for collection of the information needed to do the study.

In designing resume templates, we look at certain aspects that determine a good resume builder and tackle some problems that you might face. Some of them are

- The layout of the homepage.
- The layout of the individual pages.
- Login screen designs.
- Resume template gallery.

3. DESIGN

3.1 REQUIREMENT SPECIFICATION

The primary goals of the system analyst is to improve the efficiency the existing system. For that the study of specification of the requirement is very essential .For the development of the new system will be conducted.

Hardware Requirements:-

Server:-

- 128 MB RAM
- 2.1 GB Free Space
- Router
- 10 Base T Ethernet Card

Client:-

- 128 MB ram
- 1GB free space
- 10 Base T Ethernet Card

Software Requirements:-

- wamp
- mysql
- notepad++
- php

3.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representations of the entities that are to be used in the products being developed are needed to be designed.

There are various kinds of methods involved in software design. They are as follows:

- Class diagram
- Use case diagrams
- Sequence diagrams
- Collaboration diagrams
- State chart diagram

3.2.1 CLASS DIAGRAM

Class is nothing but a structure that contains both variables and methods. The class diagram shows a set of classes, interfaces, collaborations and their relationships. There is most common diagram in modeling the object oriented systems and used to give the static view of a system. It shows the dependency between the classes that can be used in our system.

Class:

A description of set of objects that shares the same set of attribute operations, relationships and semantics

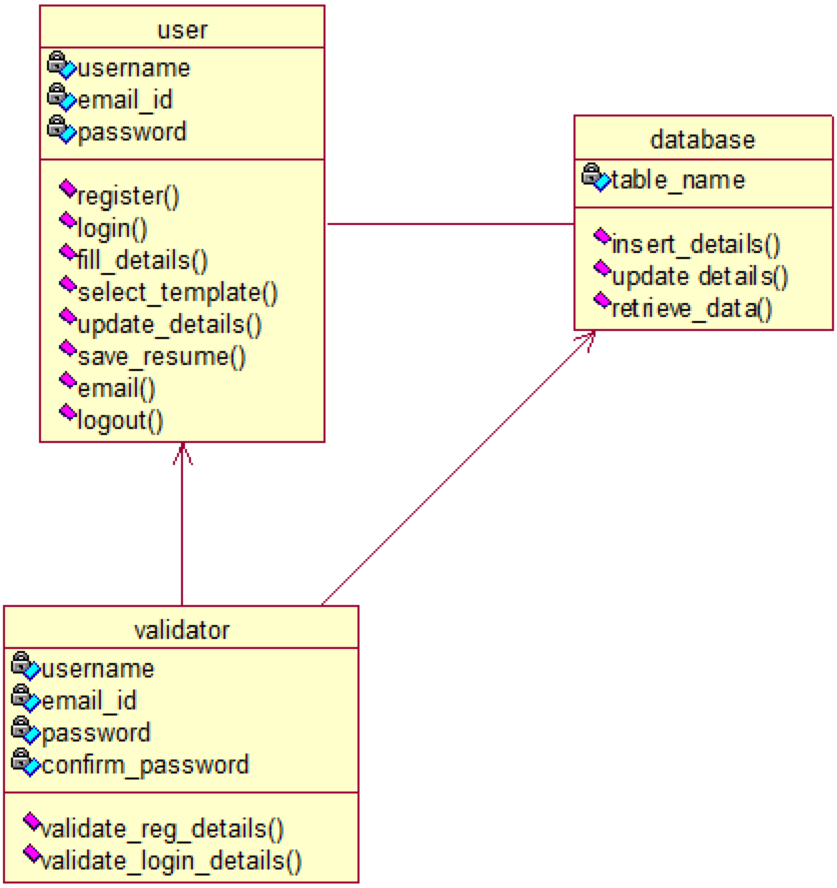


FIG:3.2.1:class diagram

3.2.2.USE CASE DIAGRAM

Use case model behavior within the system helps the developers understand of what the user require. The stick man represents what's called an actor.

Use case diagrams are useful for getting an overall view of the system and clarify that what it can do and more importantly what they can't do.

A use case diagram consists of use cases and actors and shows the interaction between use case and actors.

- The purpose is to show the interaction between use case and actors.
- To represent the system requirements from the user's perspective.
- An actor could be the end user of the system or an external system.

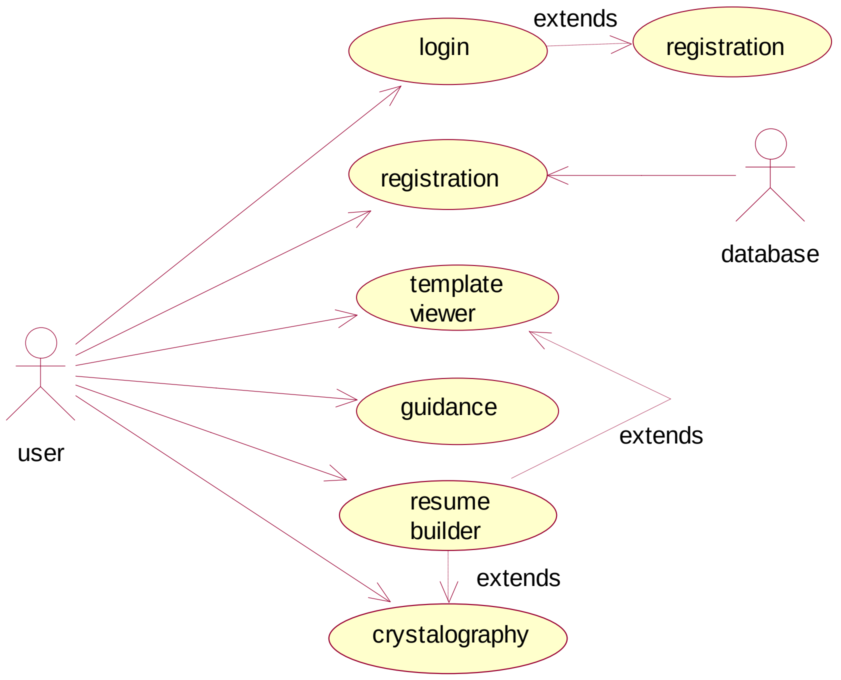


Fig:3.2.2:use case diagram

3.2.3 SEQUENCE DIAGRAM

Sequence diagrams and Collaboration diagrams are called INTERACTION diagrams. An interaction diagram shows an interaction, consisting of set of objects and their relationships including messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objet arranged along the x-axis and message order in increasing time along the y-axis.

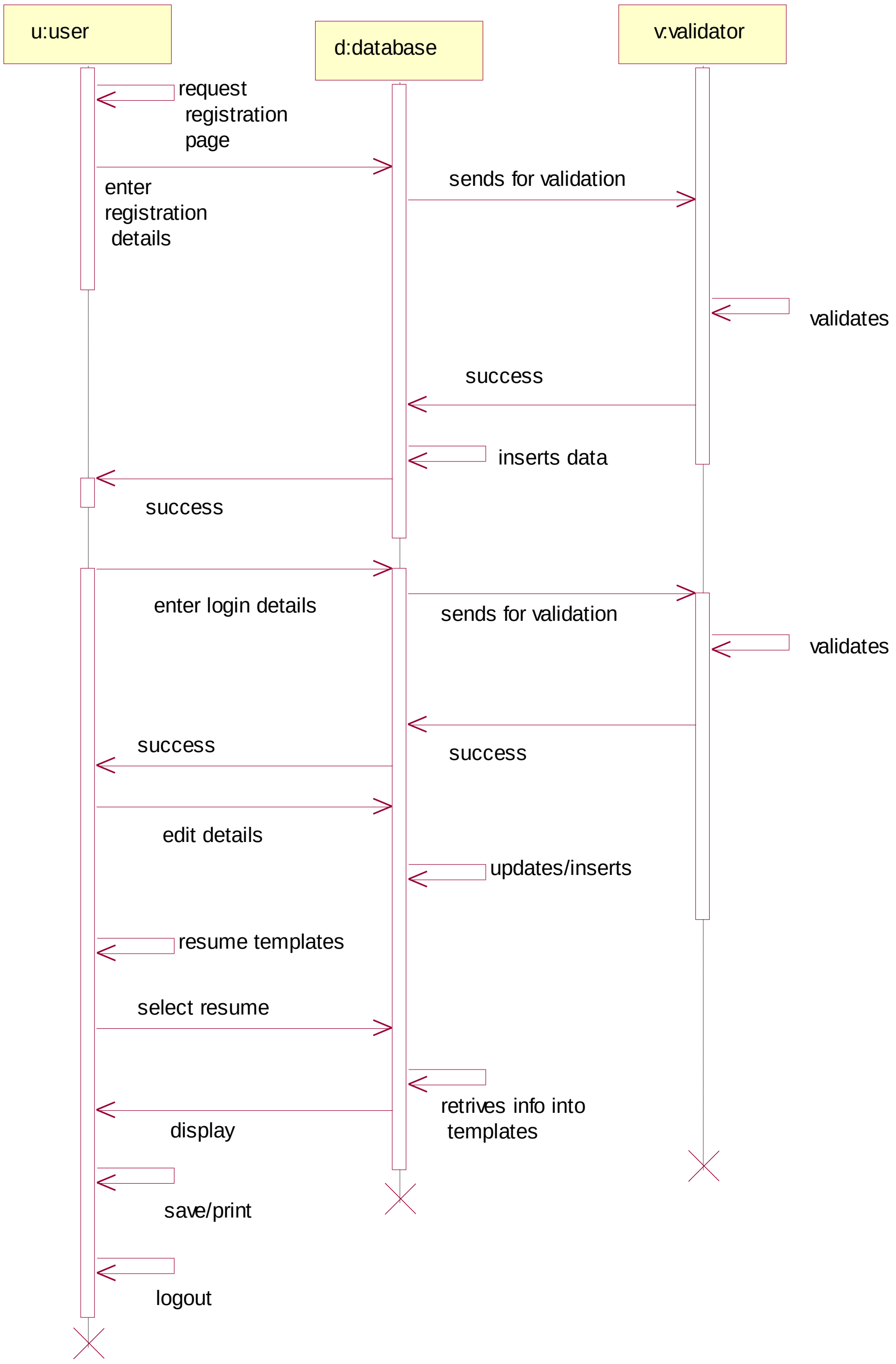


Fig:3.2.3: Sequence diagram

3.2.4 COLLABORATION DIAGRAM

Collaboration diagram emphasizes on the structural organization of the objects that send and receive messages. Collaboration diagrams aim at showing the communications that happen between objects, by defining messages that flow between each other. They basically consist of superimposing the communication actions upon an object diagram. The temporal aspect can be shown here too, by numbering the interactions with sequential labels. A collaboration diagram shows the interactions between objects or classes in terms of links and messages that flow through the links. This describes at the same time some kind of static structure (links and nodes) and dynamic behavior (messages) of the system.

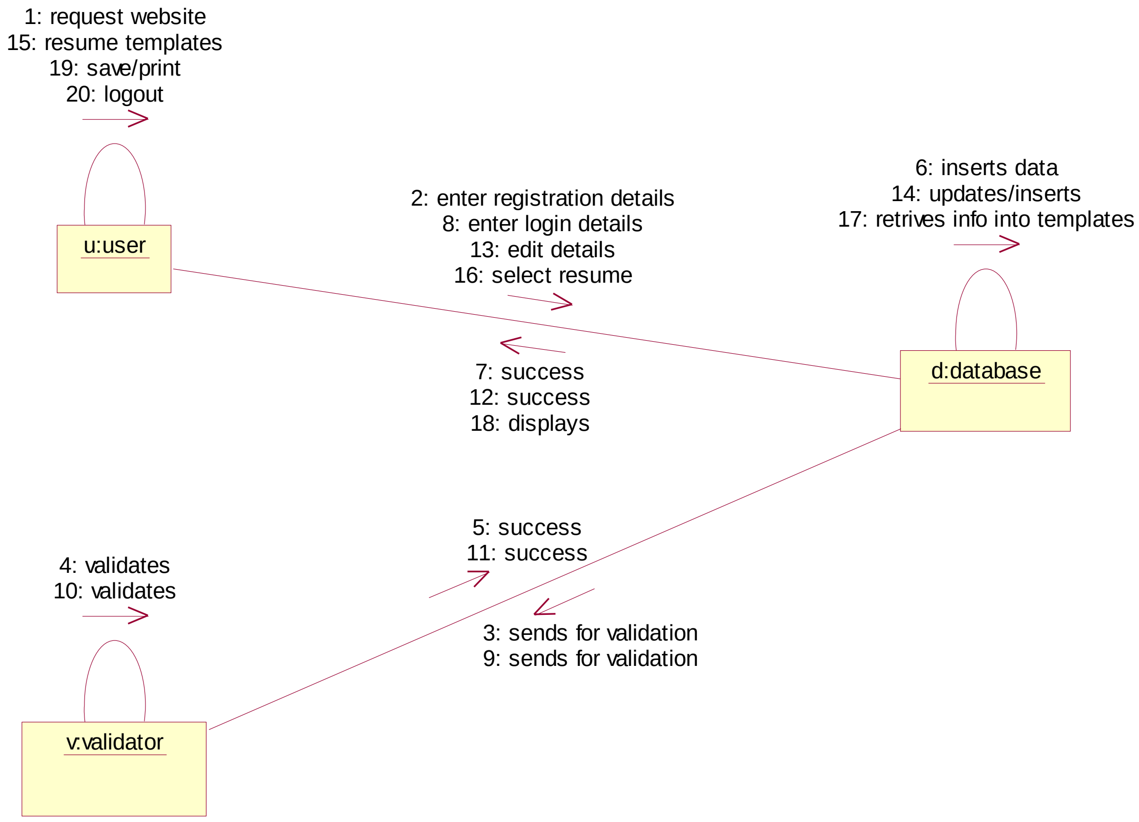


Fig:3.2.4: Collaboration diagram

4. IMPLEMENTATION

4.1 MODULES

- Registration Module:

Registration module provides the new users accessing the website to create their account for their resume building process. They need to provide information on fields like

- username,
- email-id ,
- password and
- confirm password.

Once their registration is successful they are directed to the further related links.

- Login Module

Login module is for the users operating the website .In this module the user needs to provide his unique username and his password to login and access the further links.

- Edit Details Module:

Details module consists of all the details to be filled by the user accessing the profile which are to be displayed on the resume. It has the facility to update the details in all the sections.

- Resume templates Module:

Resume templates module provides the user with the resume templates once he is done with filling the details needed for the contents of resume. This module has the various templates proven under the expert's guidance.

- Database

A database is a repository of data, designed to support efficient data storage, retrieval and maintenance. Multiple types of databases exist to suit various industry requirements. A database may be specialized to store binary files, documents, images, videos, relational data, multidimensional data, transactional data, analytic data, or geographic data to name a few.

Data can be stored in various forms, namely tabular, hierarchical and graphical forms. If data is stored in a tabular form then it is called a relational database. When data is organized in a tree structure form, it is called a hierarchical database. Data stored as graphs representing relationships between objects is referred to as network database. In this book, we focus on relational databases.

Database management system:

A database management system, or simply DBMS, is a set of software tools that control access, organize, store, manage, retrieve and maintain data in a database. Why do we need database software or a DBMS? Can we not just store data in simple text file for example? The answer lies in the way users access the data and the handle of corresponding challenges. First, we need the ability to have multiple users insert, update and delete data to the same data file without "stepping on each other's toes". This means that different users will not cause the data to become inconsistent, and no data should be inadvertently lost through these operations. We also need to have a standard interface for data access, tools for data backup, data restore and recovery, and a way to handle other challenges such as the capability to work with huge volumes of data and users. Database software has been designed to handle all of these challenges.

The most mature database systems in production are relational database management systems (RDBMS's). RDBMS's serve as the backbone of applications in many industries including banking, transportation, health, and so on.

4.1.1 Database tables:

Table	Action	Record	Type	Collation	Size	Overhead
Aid		8	InnoDB	Latin1_swedish_ci	32.0KiB	-
Det		19	InnoDB	Latin1_swedish_ci	32.0KiB	-
Educ		12	InnoDB	Latin1_swedish_ci	32.0KiB	-
objective		9	InnoDB	Latin1_swedish_ci	32.0KiB	-
Regis		26	InnoDB	Latin1_swedish_ci	48.0KiB	-
Work		9	InnoDB	Latin1_swedish_ci	32.0KiB	-
6 table(s)	Sum	83	InnoDB	Latin1_swedish_ci	208.0KiB	0B

Table:4.1.1: Database tables

4.1.2 Registration table:

Field	Type	Collation	Attributes	Null	Default	Extra
Id	int(30)	Latin1_swedish_ci		No	none	Auto_increment
uname	varchar(30)	Latin1_swedish_ci		No	none	
email	varchar(30)	Latin1_swedish_ci		No	none	
pwd	varchar(30)	Latin1_swedish_ci		No	none	

Table:4.1.2: registration tables

4.1.3 Personal details table:

Field	Type	Collation	Attributes	Null	Default	Extra
Did	int(30)			No	none	Auto_increment
Id	int(30)			No	none	
fname	varchar(30)	Latin1_swedish_ci		No	none	
lname	varchar(30)	Latin1_swedish_ci		No	none	
faname	varchar(30)	Latin1_swedish_ci		No	none	
Dob	Date			No	none	
country	varchar(30)	Latin1_swedish_ci		No	none	
Ad	Text	Latin1_swedish_ci		No	none	
City	varchar(30)	Latin1_swedish_ci		No	none	
state	varchar(30)	Latin1_swedish_ci		No	none	
Zip	int(30)			No	none	
phone	int(30)			No	none	
email	varchar(30)	Latin1_swedish_ci		No	none	

Table:4.1.3:personal details table

4.1.4 Education details table:

Field	Type	Collation	Attributes	Null	Default	Extra
Eid	int(30)			No	None	Auto_increment
Id	int(30)			No	None	
school	varchar(30)	Latin1_swedish_ci		No	None	
sagg	int(30)			No	None	
syear	int(30)			No	None	
cname	varchar(30)	Latin1_swedish_ci		No	None	
iagg	int(30)			No	None	
cyear	int(30)			No	None	
graduate	varchar(30)	Latin1_swedish_ci		No	None	
gcname	varchar(30)	Latin1_swedish_ci		No	None	
gagg	int(30)			No	None	
gyear	int(30)			No	None	
Pg	varchar(30)	Latin1_swedish_ci		No	None	
pgname	varchar(30)	Latin1_swedish_ci		No	None	
Pgagg	Int(30)			No	None	
pgyear	Int(30)			No	None	

Table:4.1.4:education details table

4.1.5 Objective details table:

Field	Type	Collation	Attributes	Null	Default	Extra
Oid	int(30)			No	none	Auto_increment
Id	int(30)			No	none	
Obj	Text	Latin1_swedish_ci		No	none	

Table:4.1.5: objective details

4.1.6 Work experience table:

Field	Type	Collation	Attributes	Null	Default	Extra
Wid	int(30)			No	none	Auto_increment
Id	int(30)			No	none	
company	varchar(30)	Latin1_swedish_ci		No	none	
position	varchar(30)	Latin1_swedish_ci		No	none	
sdate	Date			No	none	
edate	Date			No	none	
City	varchar(30)	Latin1_swedish_ci		No	none	
state	varchar(30)	Latin1_swedish_ci		No	none	
country	varchar(30)	Latin1_swedish_ci		No	none	

Table:4.1.6:working details

4.1.7 Additional details table:

Field	Type	Collation	Attributes	Null	Default	Extra
Aid	int(30)			No	none	Auto_increment
Id	int(30)			No	none	
Com	Text	Latin1_swedish_ci		No	none	
Intt	Text	Latin1_swedish_ci		No	none	
Ref	Text	Latin1_swedish_ci		No	none	
Head	Text	Latin1_swedish_ci		No	none	
Headdd	Text	Latin1_swedish_ci		No	none	

Table:4.1.7:additional details

4.2 OVERVIEW TECHNOLOGY

PHP is a popular computer scripting language that is optimized for web development. It generally runs on a web server and can be deployed on most web servers and on almost every operating system and platform free of charge, under an open source license. Its free availability, power, and ease of use account in large part for its widespread adoption: PHP is installed on more than 20 million websites and 1 million web servers.

PHP is very flexible: Simple PHP functions can be embedded within web pages, or it can work the other way: HTML code can be embedded within PHP. We'll cover that in more detail in a bit.

PHP is very versatile: it can talk to a variety of external applications using a wide variety of protocols, and it can parse (understand) a wide range of data interchange formats.

PHP has modules that can parse XML, various forms of delimited content, and of course, our old friend MARC.

PHP is now a recursive acronym, that stands for "PHP Hypertext Pre-processor" ... but it didn't always.

- relatively easy to use, compared to traditional languages like C
- run inside another program
- interpreted rather than compiled

PHP is just one element in a complex ecology of technologies and protocols that make up the web environment, and it works the way it does in large part because of the niche that it occupies and the other technologies that it has to interact with.

The web wasn't always so complex, of course. Back in the very early days, you could model the typical web transaction something like this:

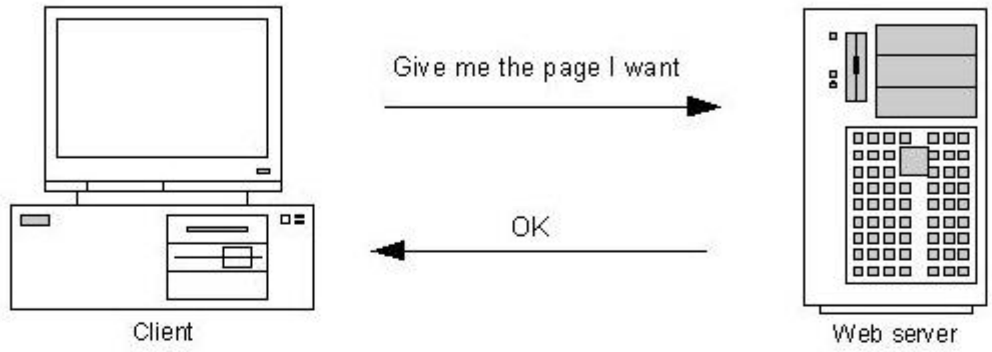


Fig:4.2.1: Typical web transaction

A client (typically a web browser running on a desktop computer) sends a request for a particular file to a server. The request is formatted in a particular way that allows the server to identify where the file lives in its file system. The server returns the file, along with a brief header indicating the status of the request (successful, not successful) and a few other things that might help the client software process the response.

You can see that the URL can be chunked into a number of definable parts. The first part specifies the protocol that is to be used to handle the request (the hypertext transfer protocol, or http, the fundamental protocol of the web); the second is the address of the server to which the request is being sent; the third is where the file lives on the server; and the fourth is the name of the file that the client wants to retrieve.

An important thing to note here is that the file being sent back to the client (typically an html document, but possibly a binary file like an image) is a static entity. That works reasonably well for certain kinds of information (reports, essays, articles, pictures of cats) but falls down when it comes to other kinds of data. For example, what if you had a list of people and a list of organizations, and you wanted to show the people sorted alphabetically, and also sorted by organization? Well, you could maintain the data in two separate files. And then someone comes along and asks for the same thing, but this time sorted by email address. Now we're up to 3 separate files containing the same data, and we have the beginnings of a maintenance problem.

So early on it was seen that simply serving up static files was pretty limiting, and some way was needed to produce a more dynamic response to user input. Of course, this first requires that the user be able to send a wider range of input than is available just be clicking on a hyperlink.

One very common way to gather user input is to have the user fill in some fields on a web form, which in turn affixes that input to the end of the http request, in the form of name/value pairs called parameters.

So we can now visualize the transaction as being more like this:

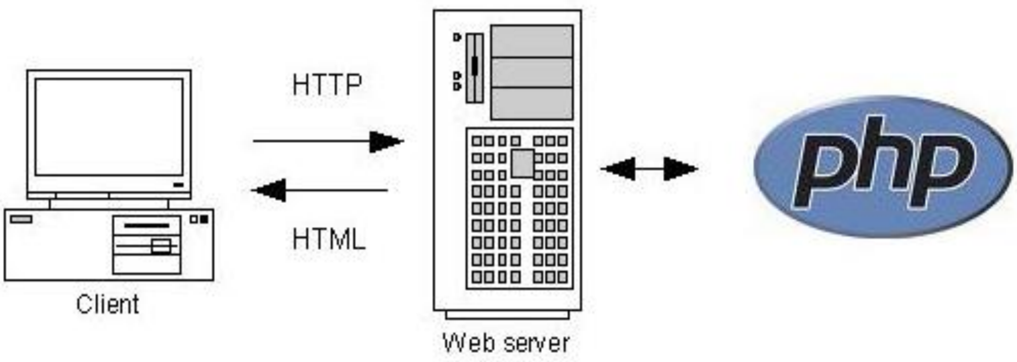


Fig:4.2.2: Transaction

The web server still receives the request, but can no longer process the whole request by itself. Instead, it sees that the request is for a php script, so it passes off the processing of the script to the PHP interpreter, which is just another piece of software. The PHP interpreter processes the script (which in turn processes any parameters that have accompanied the request), and returns some output (generally html) to the web server, which in turn passes it back to the client.

Of course, this diagram might be a bit confusing because PHP (the interpreter) typically lives on the same hardware as the web server software. What you need to keep in mind here is that all of these components are really just software, and software can live anywhere, even though I've followed a diagrammatic convention of using hardware icons to represent two of the 3 components. As you'll soon see, all three components (browser, web server, and php interpreter) can even live together on the same piece of hardware, if you want them to.

That's about as far as we're going to take you in this course, but we'd be remiss if we didn't at least mention that there's yet another level of complexity, wherein PHP in turn interacts with other services and applications before returning output to the web server. That's where things really get interesting.

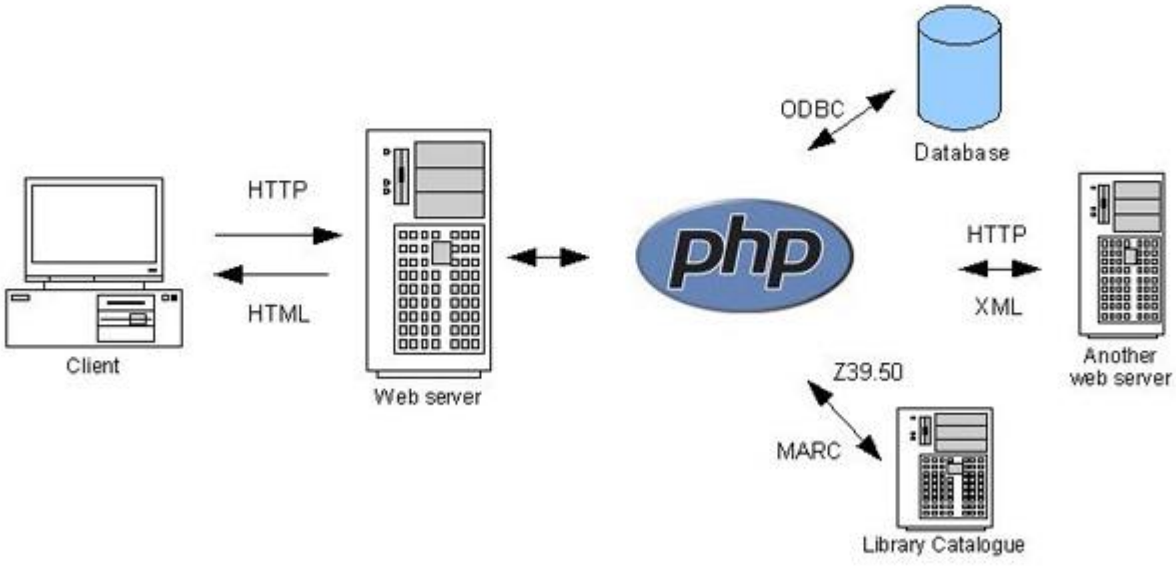


Fig:4.2.3: Php interacting with other services

WAMP SERVER

Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). It is often used for web and internal testing, but may also be used to serve live websites.

The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet.

WAMP also includes MySQL and PHP ,which are two of the most common technologies used for creating dynamic websites. MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.

While Apache, MySQL, and PHP are open source components that can be installed individually, they are usually installed together. One popular package is called "WampServer," which provides a user-friendly way to install and configure the "AMP" components on Windows.

5. TESTING

Introduction

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible.

Testing methods:

The box approach software testing methods are traditionally divided into white and black box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases

White box testing:

White box testing is when tester has access the internal data structures and algorithms including the code that implement these

Types of white box testing:

The following are the types of white box testing

- API testing(application programming interface):-testing of the application for the public and private API's
- Code coverage:-creating testes to satisfy some criteria of code coverage (eg.,The test designer can create to causes all statements in program to be executed atleast once).
- Fault injection method:-improving the coverage of test by introducing faults to test the code paths

Black box testing:

Black-box test design treats the system as a “black-box” so it doesn’t explicitly use the knowledge of the internal structure. Black box test design is usually described as focusing on testing functional requirements

A simple black box specification:

Black box testing is testing technique having no knowledge of internal functionalities/structure of the system. This testing technique treats the system as black box or closed box. Tester will only know the formal inputs and projected results .tester doesn’t no how the program actually arrives at those results.

Hence tester tests the system based on functional specifications given by him. That is the reason black box testing is also consider as the functional testing.

Unit testing:

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system ,the datas are stored in database and retrieved. If you want required module to access input or get the output from the End user. any error will accrued the time will provide handler to show what type of error will accrued .

System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on input basis .if you want change any values or inputs will change all information. so specified input is must.

5.1 TEST CASES

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to assure their validity.

The test case speciation for system testing has to be submitted for review before system testing commences.

TEST CASE 1:

Object**:** verification of login fields

Test Case ID**:** TC resume1

Test Case Name**:** Test Cases of login fields while entering into database by user

Test Procedure**:** Look and feel of the application is tested with its behavior of the application

Data Matrix:

Input attributes	ECP		BVA	
	Valid	Invalid	Min	Max
Username	0-9,a-z,A_Z	special symbols and all blank fields	7	20
Password	0-9,a-z,A-Z	special symbols and all blank fields	5	10

Table 5.1.1: Test case for Login Field

TEST CASE 2:

Object: verification of Registration fields

Test Case ID: TC resume2

Test Case Name**:** Test Cases of registration fields while entering into database by user

Test Procedure**:** Look and feel of the application is tested with its behavior of the application

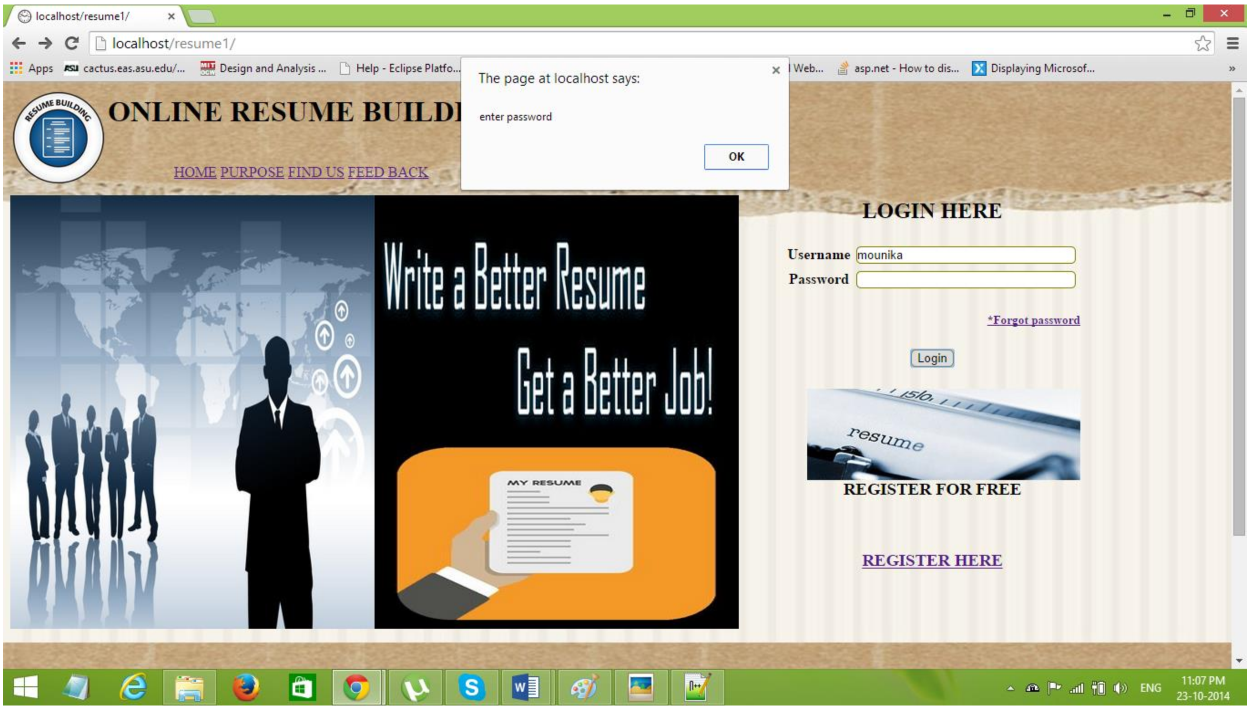
Data Matrix:

Input attributes	ECP		BVA	
	Valid	Invalid	Min	Max
User name	0-9,a-z,A-Z	special symbols and all blank fields	7	20
Password	0-9,a-z,A-Z	special symbols and all blank fields	5	10
Email	0-9,a-z,A-Z,@, . ,	special symbols and all blank fields	-	30

Table 5.1.2: Test case for registration Field

5.2 TEST RESULTS

- As user has not mentioned any password in the password field, since it is important while login, it will be show alert in the login page



The page at localhost says:

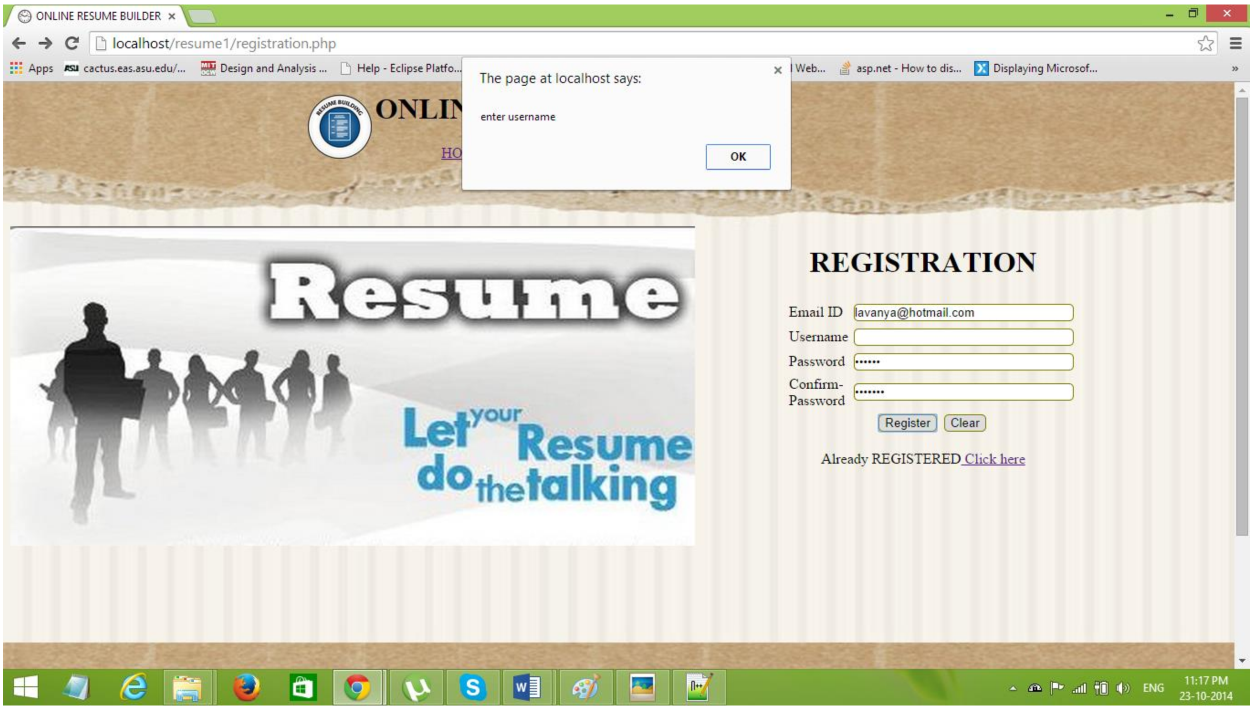
enter password

Username mounika

Password

Fig 5.2.1: password field empty

- As user has not mentioned any username in the username field, since it is important while registration, it will be show alert in the registration page.



The page at localhost says:

enter username

REGISTRATION

Email ID lavanya@hotmail.com

Username

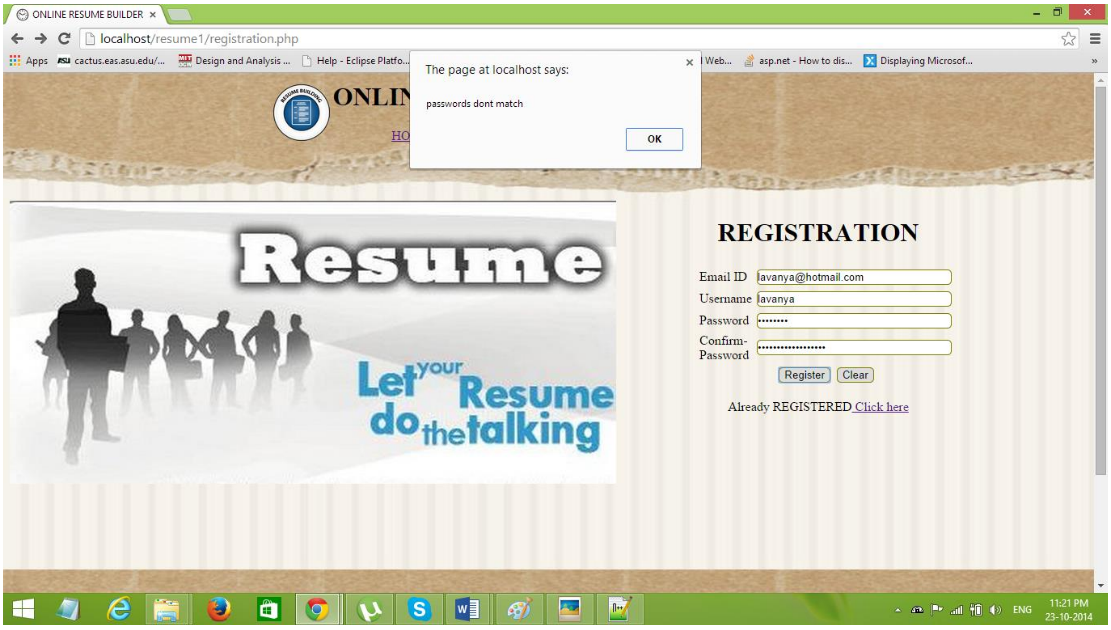
Password ••••••

Confirm-Password •••••••

Already REGISTERED Click here

Fig:5.2.2: username field empty

- User has not mentioned same passwords in the password field and confirm password fields,since it is important while registration, it will be show alert in the registration page.



The page at localhost says:

passwords dont match

OK

REGISTRATION

Email ID lavanya@hotmail.com

Username lavanya

Password ••••••••

Confirm-Password ••••••••••••••••••

Register Clear

Already REGISTERED Click here

ONLINE RESUME BUILDER
localhost/resume1/registration.php
Resume
Let your Resume do the talking
11:21 PM
23-10-2014

Fig 5.2.3: passwords did not match

6. RESULTS

Login page:

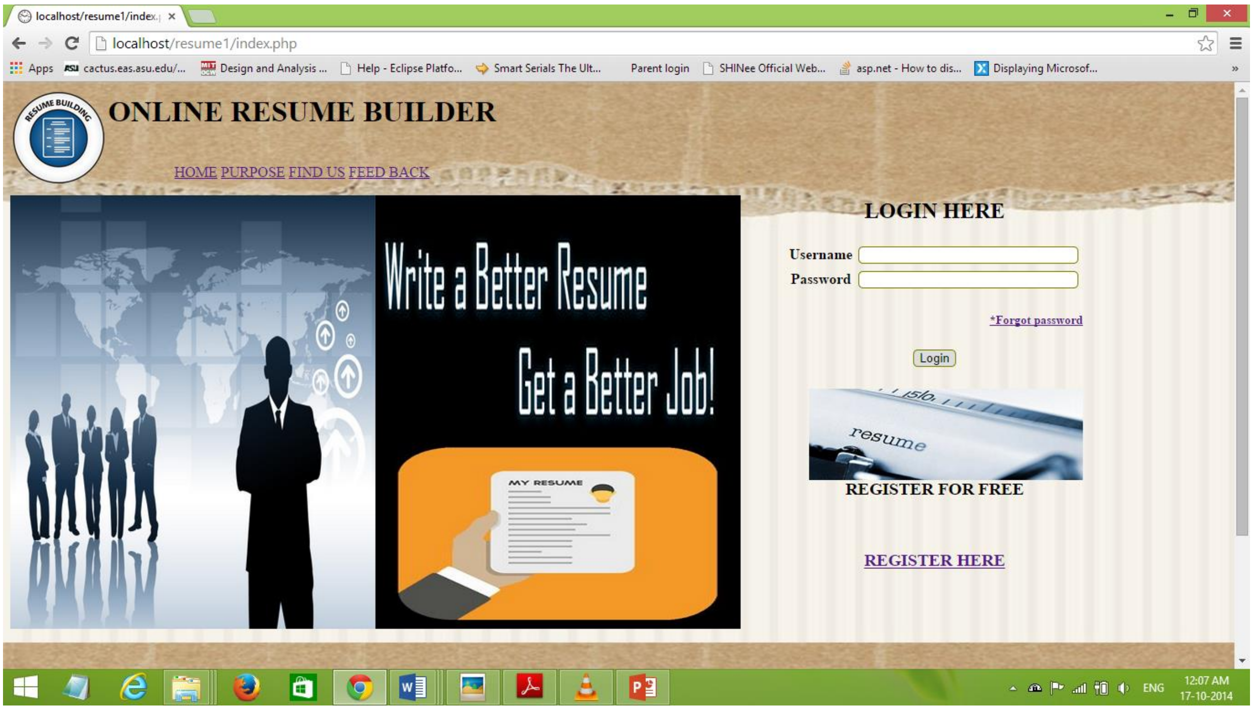


Fig 6.1.1 Login Page

Registration page:

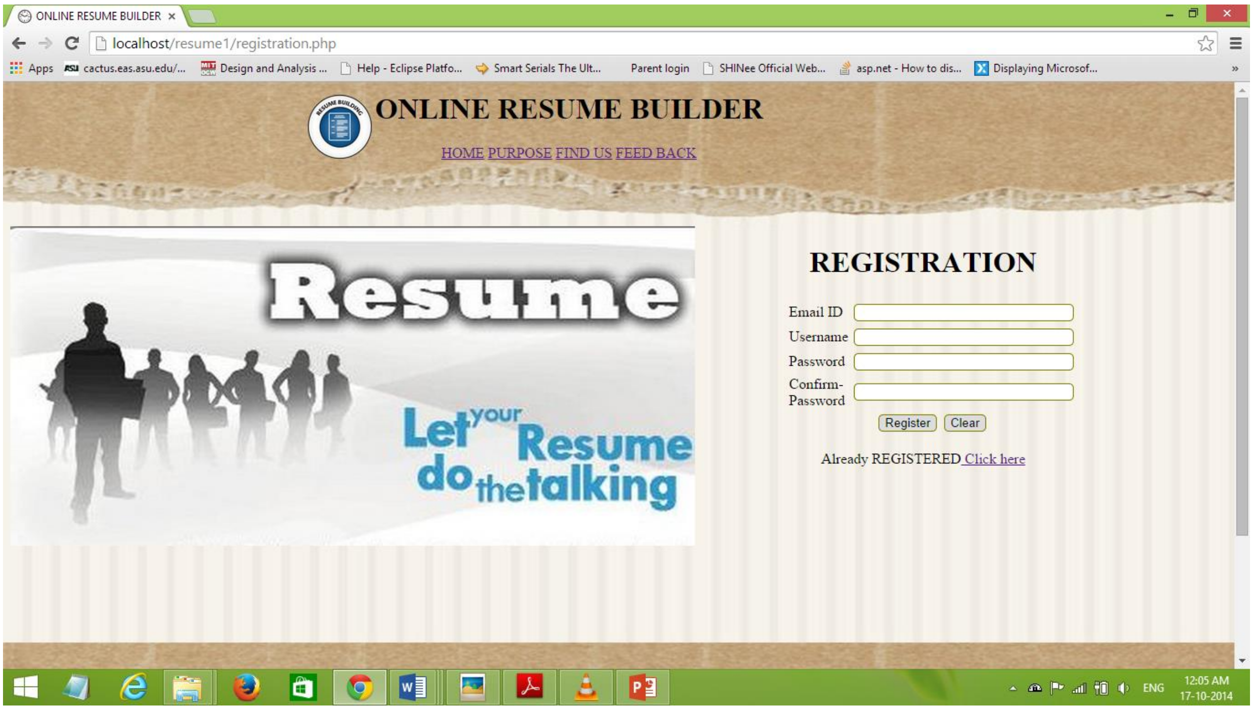


Fig 6.1.2 Registration Page

Contact details:

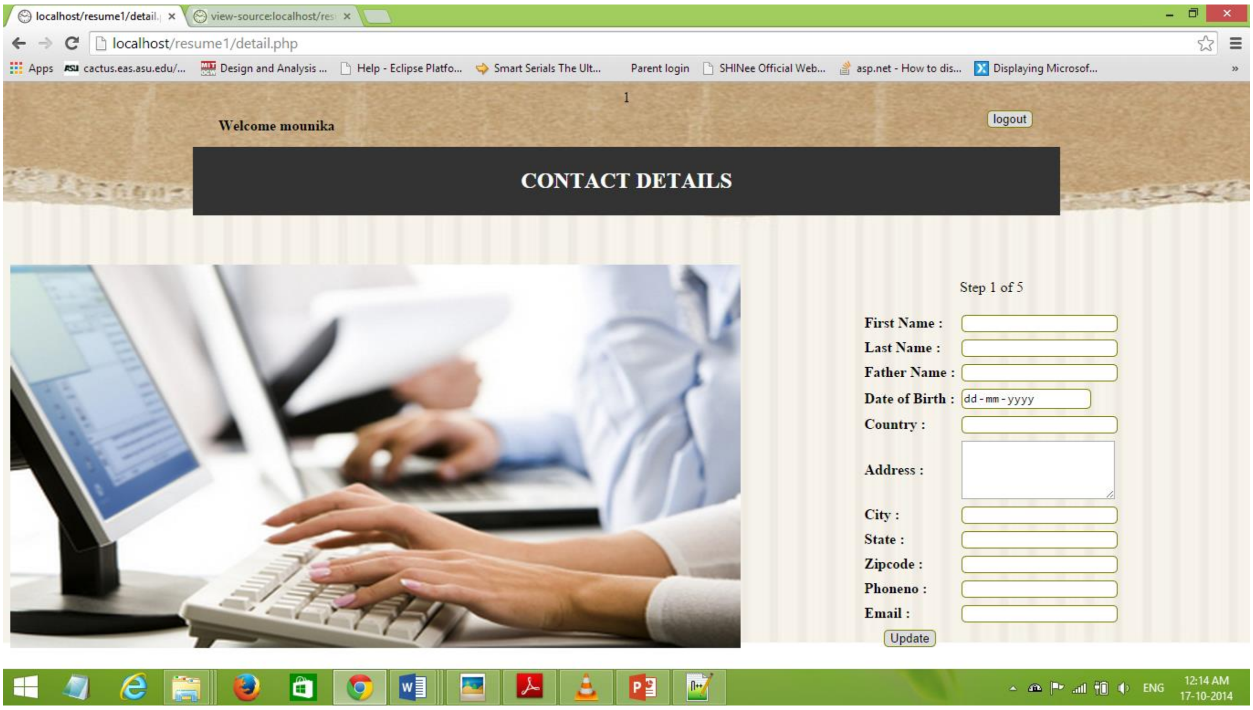
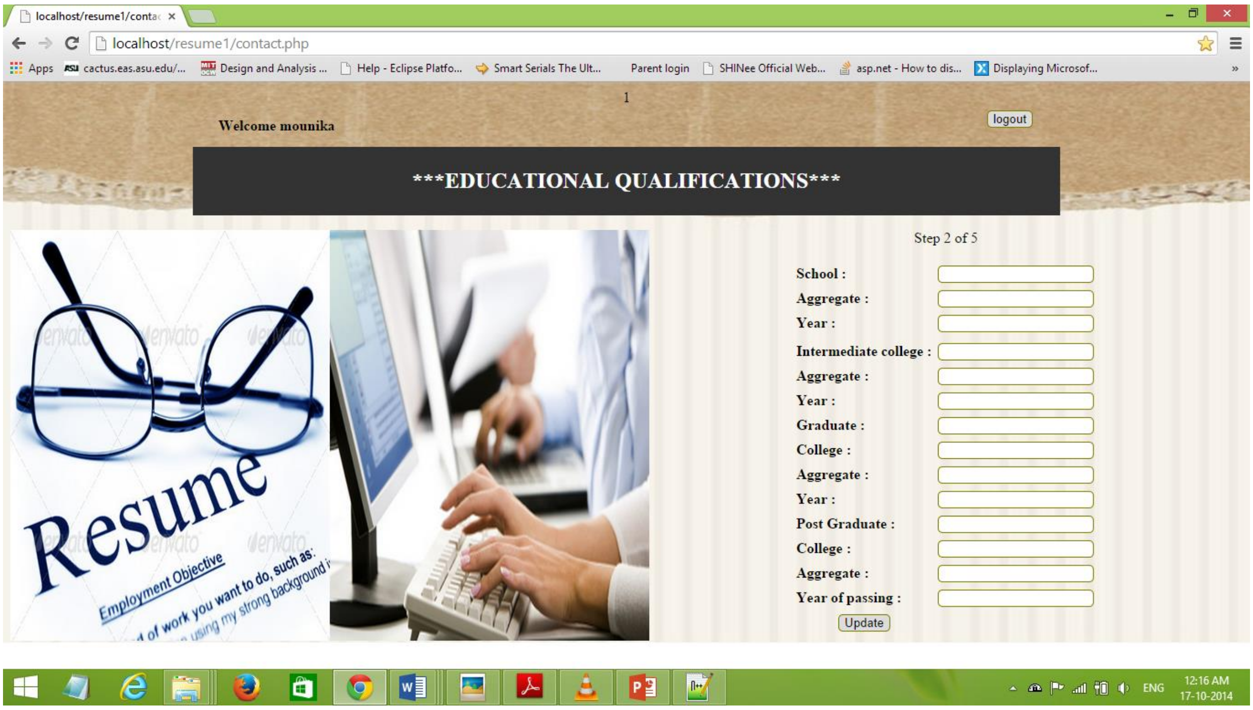


Fig 6.1.3 Contact details

Educational details:



1

Welcome mounika

EDUCATIONAL QUALIFICATIONS

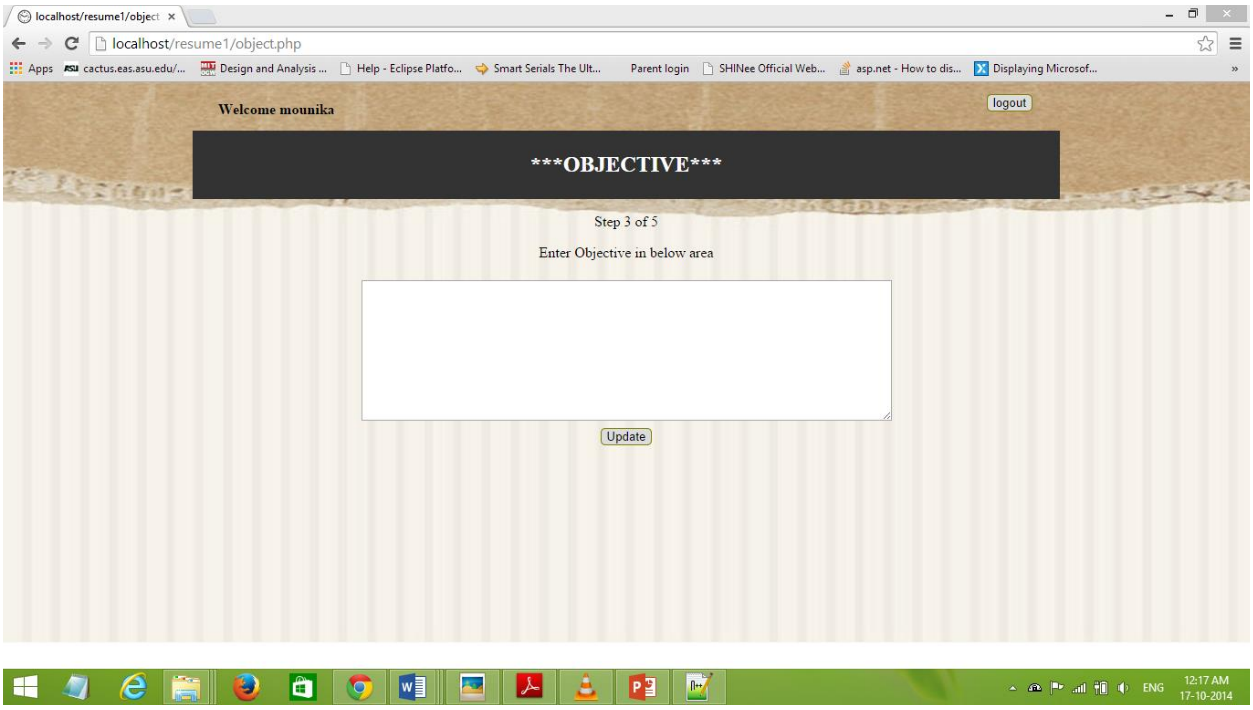
Step 2 of 5

School :
Aggregate :
Year :
Intermediate college :
Aggregate :
Year :
Graduate :
College :
Aggregate :
Year :
Post Graduate :
College :
Aggregate :
Year of passing :

Update

Fig 6.1.4 Educational details

Objective details:



Welcome mounika

OBJECTIVE

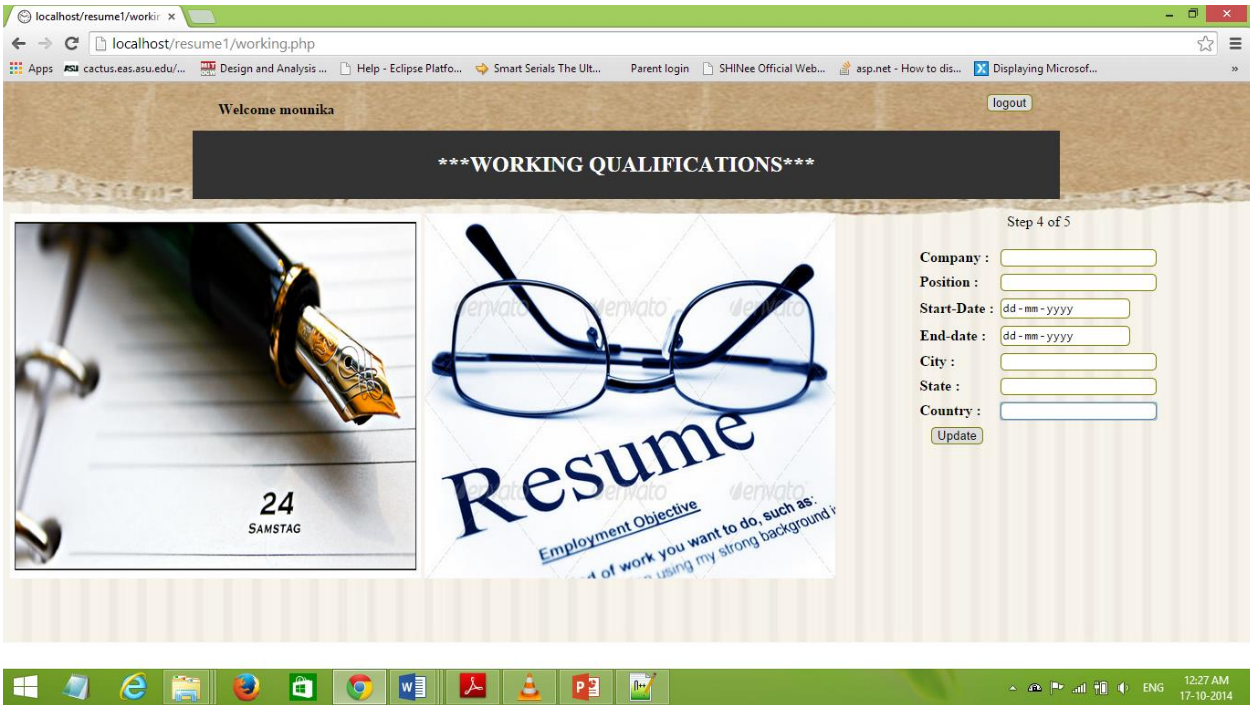
Step 3 of 5

Enter Objective in below area

Update

Fig 6.1.5 Objective details

Working skills:


localhost/resume1/working.php

Welcome mounika

WORKING QUALIFICATIONS

Step 4 of 5

Company :

Position :

Start-Date : dd-mm-yyyy

End-date : dd-mm-yyyy

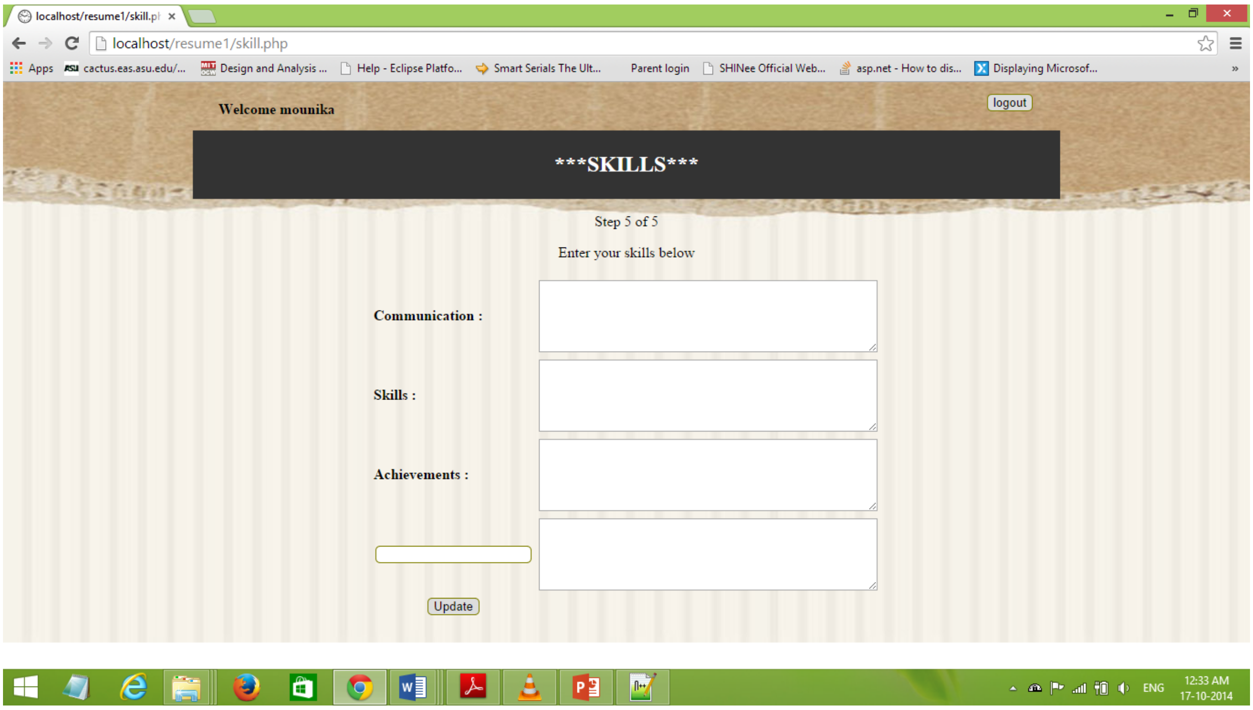
City :

State :

Country :

Fig 6.1.6 Working skills

Additional skills:


localhost/resume1/skill.php

Welcome mounika

SKILLS

Step 5 of 5

Enter your skills below

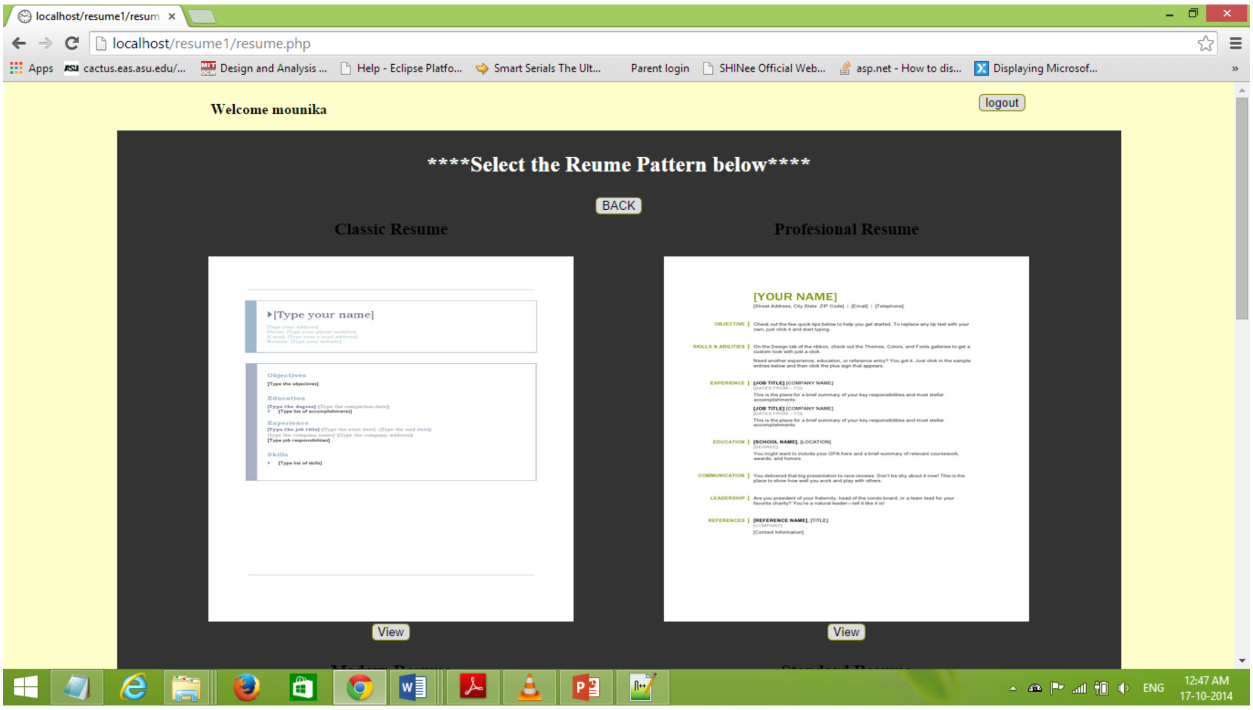
Communication :

Skills :

Achievements :

Fig 6.1.7 Additional skills

Resume Templates:



Welcome mounika

****Select the Reume Pattern below****

Classic Resume

[Type your name]
[Type your address]
Phone: [Type your phone number]
E-mail: [Type your e-mail address]
Website: [Type your website]

Objectives
[Type the objectives]

Education
[Type the degree] ([Type the completion date])
- [Type list of accomplishments]

Experience
[Type the job title] ([Type the start date] –[Type the end date])
[Type the company name] ([Type the company address])
[Type job responsibilities]

Skills
- [Type list of skills]

Profesional Resume

[YOUR NAME]
[Street Address, City State ZIP Code] | [Email] | [Telephone]

OBJECTIVE | Check out the few quick tips below to help you get started. To replace any tip text with your own, just click it and start typing.

SKILLS & ABILITIES | On the Design tab of the ribbon, check out the Themes, Colors, and Fonts galleries to get a custom look with just a click.

Need another experience, education, or reference entry? You got it. Just click in the sample entries below and then click the plus sign that appears.

EXPERIENCE | [JOB TITLE] [COMPANY NAME]
[DATES FROM – TO]
This is the place for a brief summary of your key responsibilities and most stellar accomplishments.
[JOB TITLE] [COMPANY NAME]
[DATES FROM – TO]
This is the place for a brief summary of your key responsibilities and most stellar accomplishments.

EDUCATION | [SCHOOL NAME], [LOCATION]
[DEGREE]
You might want to include your GPA here and a brief summary of relevant coursework, awards, and honors.

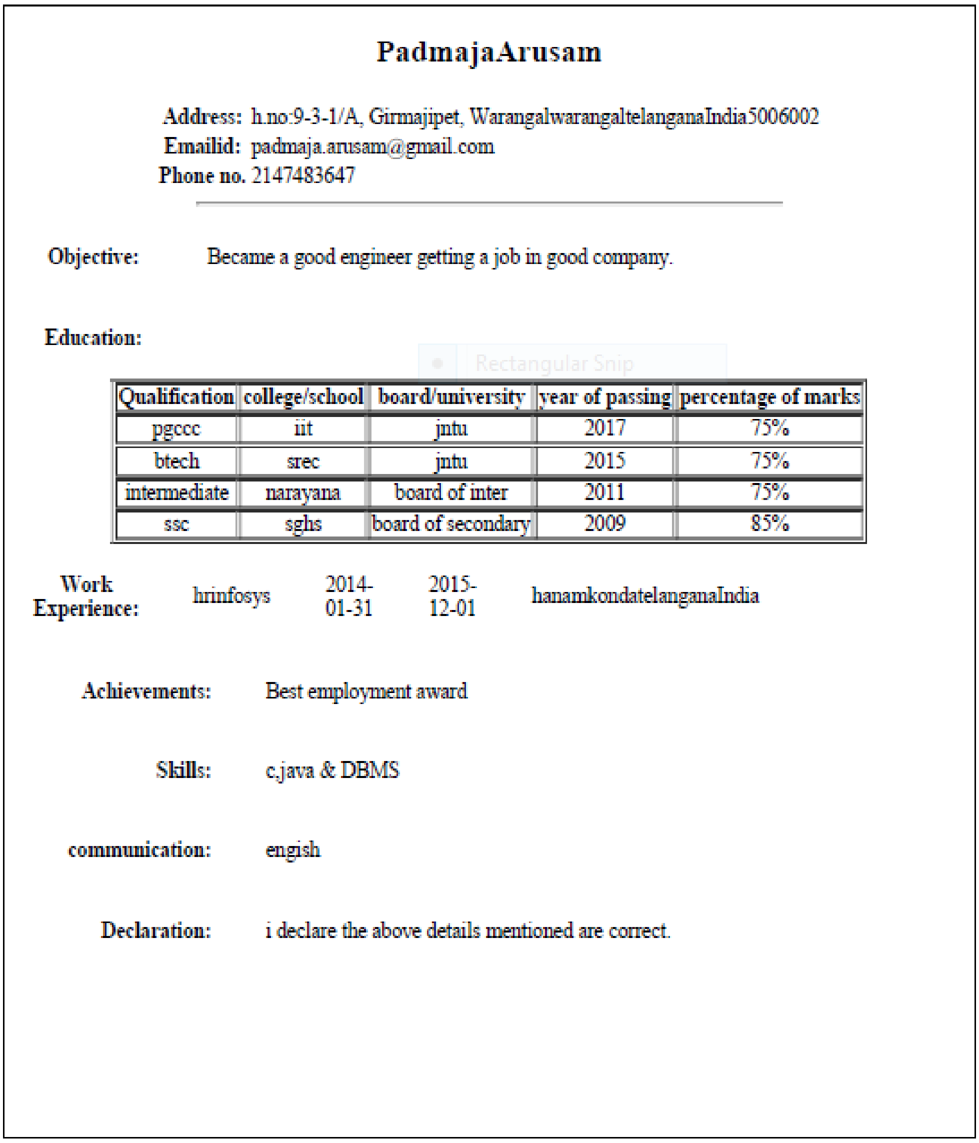
COMMUNICATION | You delivered that big presentation to rave reviews. Don't be shy about it now! This is the place to show how well you work and play with others.

LEADERSHIP | Are you president of your fraternity, head of the condo board, or a team lead for your favorite charity? You're a natural leader—tell it like it is!

REFERENCES | [REFERENCE NAME], [TITLE]
[COMPANY]
[Contact Information]

Fig 6.1.8 Resume Templates

Final resume:



PadmajaArusam

Address: h.no:9-3-1/A, Girmajipet, WarangalwarangaltelanganaIndia5006002
Emailid: padmaja.arusam@gmail.com
Phone no. 2147483647

Objective: Became a good engineer getting a job in good company.

Education:

Qualification	college/school	board/university	year of passing	percentage of marks
pgccc	iit	jntu	2017	75%
btech	srec	jntu	2015	75%
intermediate	narayana	board of inter	2011	75%
ssc	sghs	board of secondary	2009	85%

Work Experience: hrinfosys 2014-01-31 2015-12-01 hanamkondatelanganaIndia

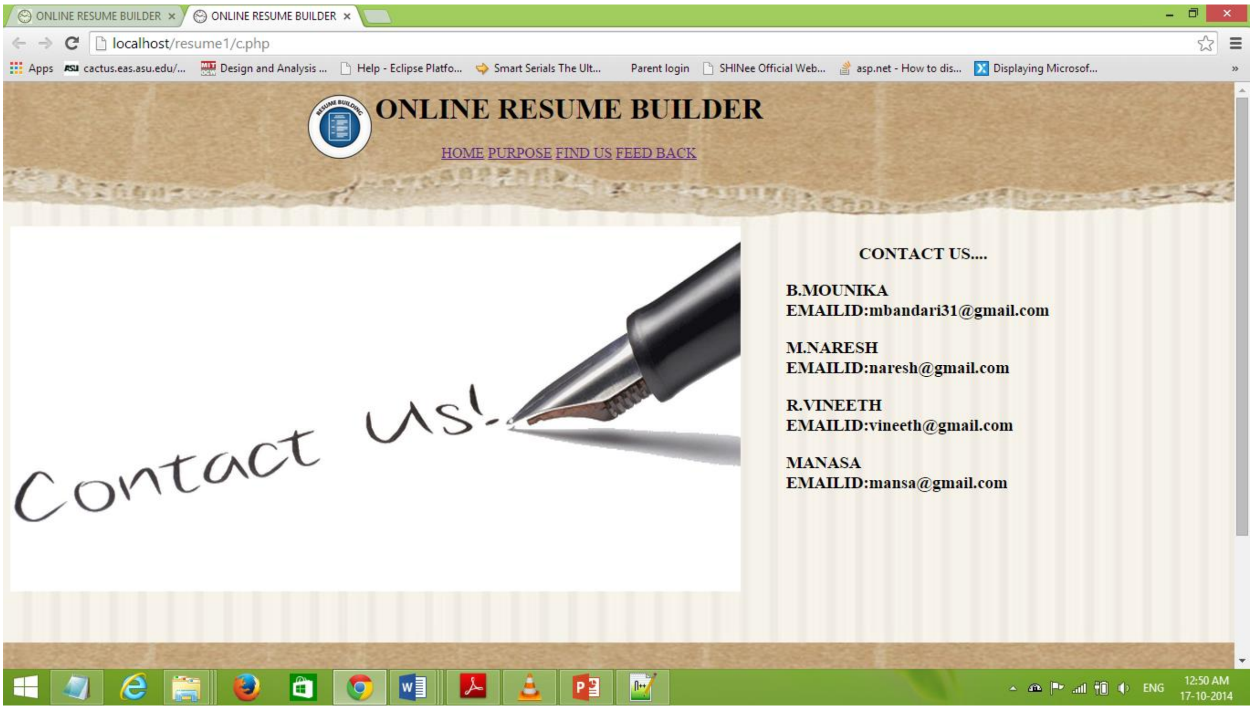
Achievements: Best employment award

Skills: c,java & DBMS

communication: engish

Declaration: i declare the above details mentioned are correct.

Contact Us:



ONLINE RESUME BUILDER

HOME PURPOSE FIND US FEED BACK

CONTACT US....

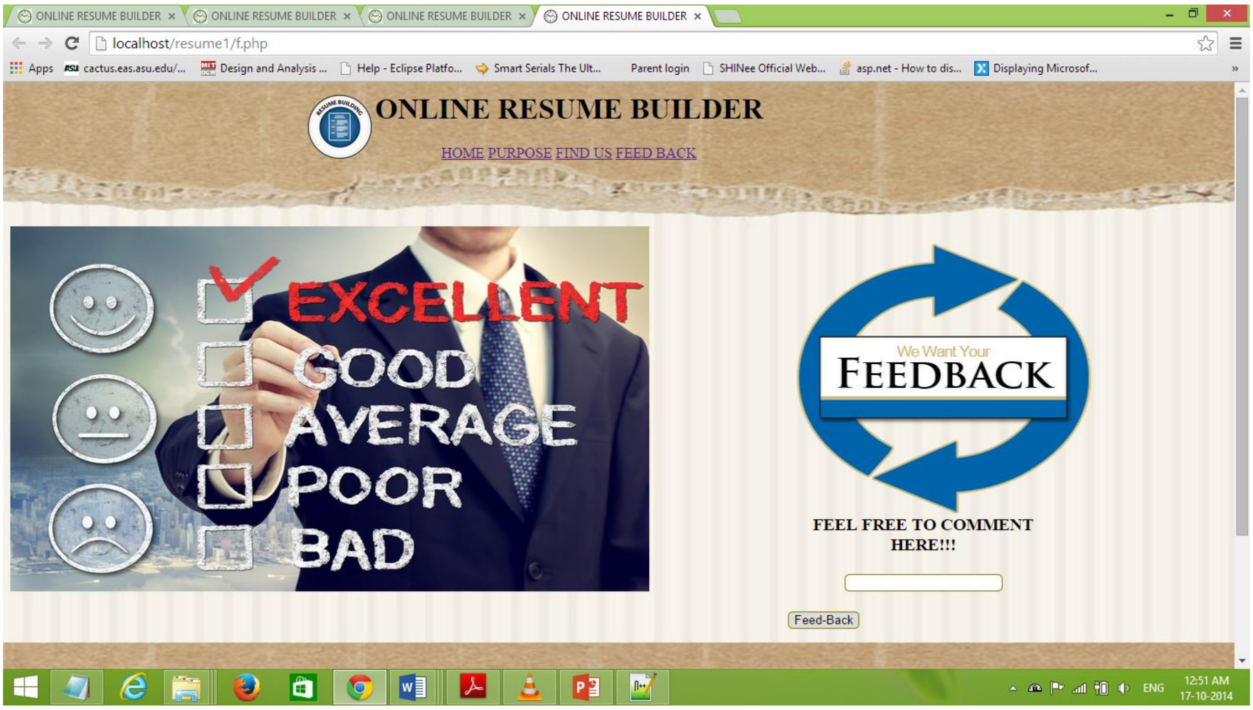
B.MOUNIKA
EMAILID:mbandari31@gmail.com

M.NARESH
EMAILID:naresh@gmail.com

R.VINEETH
EMAILID:vineeth@gmail.com

MANASA
EMAILID:mansa@gmail.com

Feedback:



ONLINE RESUME BUILDER

HOME PURPOSE FIND US FEED BACK

FEEL FREE TO COMMENT HERE!!!

7.CONCLUSION

7. CONCLUSION

The project **Online Resume Builder** is for computerizing the working of building resumes. The software takes care of all the requirements of the process and is capable to provide easy and effective storage of information related to customers and resumes that come up to the system.

It generates reports for customers & administrators. Provides easy designing tools and other interesting features. The system also provides the facility to contact the customer.

This system provides online storage/ updates and retrieval facility. This system promises very less or no paper work and also provides help to customers and viewers. In this system everything is stored electronically so very less amount of paper work is required and information can be retrieved very easily without searching here and there into registers.

8. FUTURE SCOPE

8. FUTURE SCOPE

Online resume builder is the website for creating the resume without any effort and provides the users the popular resume formats & a better way to show their resumes to the employers.

In near future other modules can be implemented such as

- Provide users with more than 20 resume templates.
- Option for saving the resume even in the doc format.
- Options for uploading the resume from the computer and making changes.
- Providing optional case for the fields which have been not entered in the forms, i.e if the fields are not entered in the form ,then the resume with data regarding that field should not be shown
- User choosing the font styles.
- User can prepare his own resume template and save that template.
- Showing the updates in the home page about the companies recruiting the jobs in the present and in the future

BIBLIOGRAPHY

- HTML: Complete Reference
- Database management systems
- www.w3schools.com
- www.stackoverflow.com
- http://php.resourceindex.com/
- http://www.developphp.com/
- PHP and MySQL Web Development by Luke Welling
- Programming PHP by Rasmus Lerdorf
- http://php.net/manual/en/function.list.php
- http://www.wampserver.com/en/