# PROJECT REPORT

# ON

**“JOB PORTAL”**

SUBMITTED TO

Dr. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE ENGINEERING

BY

MOHD AMIR (1533010025)

POOJA TOMAR (1533010017)

MOHD SHAHNAWAZ (1533010026)

# Under the supervision of

Mr. ATUL BHARDWAJ

Computer Science Engineering Department

SIET



## DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

**SARASWATI INSTITUTE OF ENGINEERING & TECHNOLOGY, GHAZIABAD**

# DECLARATION

We, “MOHD AMIR, POOJA TOMAR, MOHD SHAHNAWAZ, students of B. Tech.(CSE) hereby declare that the project titled “Job Portal” which is submitted by us to the Department of Computer Science Engineering, SARASWATI INSTITUTE OF ENGINEERING & TECHNOLOGY, GHAZIABAD(Affiliated to Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW) in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology has not previously formed the basis for the award of any Degree, Diploma other similar title or recognition.

Place: MOHD AMIR (1533010025)

Date: POOJA TOMAR (1533010017)

MOHD SHAHNAWAZ (1533010026)

# ACKNOWLEDGEMENT

Nothing concrete can be achieved without an optimum combination of inspection and perspiration. The idea of presenting this material without adequate thanks to those who give it to us or pointed in the right direction seems simply indefensible. Generating this piece has been time consuming and an arduous task and has involved various contributions.

It is my pleasure to acknowledge the help I have received from different individuals, Project Guide and all the Staff of the institute during the project.

My first sincere appreciation and gratitude goes to respected Mr. ATUL BHARDWAJ, Project guide, Net max Technologies for his guidance, constructive comments, valuable suggestions and inspirations. During the entire training session, i have received endless help from him.

Also, it gives me immense pleasure to express my sincere and whole hearted thank to Mr. ATUL BHARDWAJ (H.O.D. Computer Science engineering) for giving us the required guidance.

**OBJECTIVE OF THE PROJECT**

This project is aimed at developing a web-based and central job portal for the HR Group for a company. Some features of this system will be creating vacancies, storing Applicants data, Interview process initiation, Scheduling Interviews, Storing Interview results for the applicant and finally Hiring of the applicant. Reports may be required to be generated for the use of HR group.

This system automates the manual recruitment process. We believe that once the organization chooses to use this system, it will eventually recognize the value and necessity of this system and understand the problems involved in the manual process.

This document provides details about the entire software requirements specification for the job portal. The project Online Job portal is aimed at developing a web-based and central Recruitment Process System for the HR Group of a company.

**TABLE OF CONTENTS**

**PARTICULARS PAGE NO**

**2. INTRODUCTION 6-10**

* + 1. About the Project 6
  1. Problem Formulation 7

1.4 PROJECT CATEGORY 8

1.5 PURPOSE 9

1.6 FUTURE SCOPE OF Job Portal SITES 10

**2. System Development Life Cycle 11-12**

2.1 Feasibility Study 11

2.1.1 Economic Feasibility 12

2.1.3 Operational Feasibility 13

**3. SYSTEM Analysis 13-29**

3.1 Requirement Specification 13

3.2 System Requirements 14

3.2.1 Hardware & Software Requirements 14-19

3.3 Lifecycle Model 20

3.4 Task Dependency Diagram 23

**4. System Design and Coding 29-79**

4.1 Goals and activities 29

4.2 Database Structures 31

4.3 GENERIC CODES 35

4.4 Snapshots 73

**5. System Testing and Implementation 80-91**

5.1 Objective of Testing 80

5.2 Test Case 86

**1.2About the Project**

Now-a-days the job market is so extensive that a variety of industries and companies are searching for right candidates and the prospective candidates are searching for right companies for growth opportunities. This purpose is served by most of the job portals on line. This is another job portal with an open environment for the job seekers and recruiters to meet on the same dais and know about each other so that the right candidate is placed in a right company.

The aim of Job Portal is facilitate both the candidates seeking jobs as well as the employers looking for employees for their companies. In this online application, any job seeker can search for the available jobs at any moment with updated information. When he finds a job, he can post his application to the job on line. Employers can advertise the vacancies by taking the membership, logging in and posting the job information with the eligibility criteria for the jobs. This software establishes a direct connection between the employer and the job seeker.

A job seeker can directly visit this portal and view the jobs availability information along with downloading the required information. When he logs into the system, he would be able to upload his application and post walk-in details which he knows would be held by a company. This information helps other users very much to attend the same. Further, the user will be able to view the list of companies for which ha has already applied. This enables him to take a decision when he gets a call from a company and how much time elapsed since he has applied for the company.

A registered user will be able to get useful information regarding the placement papers and sample resumes which help him to create his own resume according to industry standards. Common interview questions and sample covering letters are also available online along with FAQ’s which aid the candidate to pave his route into the job world.

A recruiter or employer can view some part of the information of job seekers initially. When the recruiter logs into the system, he would be able to view the user profiles separately along with uploading the information of newly created jobs and walk-ins. He can also see all the applications received for a particular job in response to his advertisement. Thus Job Miller is a common platform where corporate recruiters and job seekers come under the same roof.

**1.3 Problem Formulation**

Formulation of web based system and applications represents a sequence of web engineering action that begins with the identification of business needs, moves into description of web app objectives, defines major features and functions and perform requirement gathering that leads to the development of an analysis model. Formulation focuses on the “big picture” on business needs and objectives and related information.

**Problem Definition**

Before doing any step of software development as Analysis, Design, testing, and implementation, there is a need of first defining a problem. One must know what the problem is before it can be solved. The basis for a candidate system is recognition of need for improving an information system or procedure. This is the first step in an initial investigation to define the problem that led user request. The problem must be clearly understood and agreed upon by the user and the analyst. It must state the objectives the user is trying to achieve and the result the user wants to see. Emphasis should be on logical requirements of the problem rather than the physical requirements.

Among the most fundamental principles of software engineering is: Understand the problem before you begin to solve it, and be sure that the solution you conceive is one that people really want. Another fundamental software engineering principle is: plan the work before you begin performing it. This problem definition is helpful in Requirement Analysis.

The goal of this activity is to understand the requirement of new system to be developed. Understanding the properties of new system that does not exist is more difficult and requires creative thinking. The problem is more complex because an automated system offers possibilities that do not exist otherwise. Consequently, even the client may not really know the needs of the system. The analyst has to make the client aware of the new possibilities, thus helping both client and analyst determines the requirements for the new system. Solving a problem without a clear knowledge of its components may turn out to be a futile exercise. So a problem statement has to be prepared which explains every minute detail of the problem beyond doubt.

**General Description**

The General Description describes all the information related to this project. It describes the main motivation for the web app, the objective that web app must fulfill and who will use the web app.

**User Problem Statement**

* This statement describes the problem definition of the proposed system. In problem statement customer define the following things:
* The Overall objective of the web app.
* The user’s background and sophistication relative to the content and functionality of the web app.
* It define how will the user arrive at the web app.
* It defines the user category that will use the application.
* That will be user friendly.
* That will restrict the user from accessing other user’s data.

**1.4PROJECT CATEGORY**

This is **Web-based project**. It uses some aspect on distributed programming to implement scalable management software.

One of the newest media available over the Internet is www. The www is a huge collection of interconnected hypertext with a web browser and we can jump from one link to the next following the links in any direction that interest us. Now a day everything on the internet is available by a web.

The internet is the most well known component of the information superhighway network infrastructure. Today the internet is an information distribution system spanning several continents. It is very general infrastructure target not only one to e-commerce but wide range of computer based services.

**1.5 PURPOSE:-**

The purpose of this software is to completely automate the following processes:

* This system provides users to register their various types of profile like social, personal, general, professional.
* This system provides users to send a scrap message, images, and data files to their friends. User can maintain the scrap book whatever scraps he has send to users.
* The system provides user to upload the photos so that user can maintain own album.
* This system provides user to join the communities according to their scenario.
* This system provides the user to maintain their friend list and user can update their friend list.
* This system provides user to send invitation to another friend and can add to their friend list for future.

1.6 FUTURE SCOPE OF Job Portal SITES:-

**Designing and Implementing Job Seekers & Job Providers:**

The aim of this module is to collect data from the user; he may a job seeker or a job provider. Both of them are potential clients to our Resume Art. A user should be registered regardless of whether he is a job seeker or a provider. In this module we register the user and grab as many details as possible about the user.

**Designing and implementing Resume Database, job database**

The aim of the module is to create a resume for all his job seekers asking the details about the experience, education, skills, affiliations and references for the project

.

1. **Designing a Recruitment System**

The aim of the module is to design a dynamic search engine for the Resume art data base which can provide data for the job seekers and job providers.

**2. System Development Life Cycle**

**2.1 Feasibility Study**

An initial investigation culminates in a proposal that determines whether a alternative system is feasible. When approved, the proposal initiates a feasibility study that describes and evaluates candidate system and provides for the selection of the best system that meets performance requirements. To do feasibility study, we need to consider the economic, technical and behavioral factors in system development. Many feasibility studies are disillusioning for both users and analysts. First, the study often presupposes that when the feasibility document is being prepared, the analyst is in a position to evaluate solutions. Second, most studies tend to overlook the confusion inherent in system development. The feasibility study is to serve as decision document; it must answer three key questions:

* Is there a new and better way to do the job that will benefit the user?
* What are the cost and savings of the alternatives?
* What is recommended?

A feasibility study is defined as an evaluation or analysis of the potential impact of a proposed project. A feasibility study is conducted to assist decision makers in or determining whether or not to implement a particular project program. It is performed to describe and evaluate candidate system and to provide for the selection of best system that meets system performance requirement.

During the course of completion of this project work the complete analysis of the proposed system was done. In the analyzing task a complete care about the feasibility of the proposed system was taken. The following feasibility analysis was carried during the course of this project work on “social Site”.

* Economic Feasibility
* Technical feasibility
* Operational feasibility

**2.1.1 Economic Feasibility**

The cost involved in designing and implementation of the proposed system is as follows:

**Analysis and Design Cost**

The cost of analysis & designing can be worked out by calculating the number of human days spend on the analysis and designing of the project and then multiplying the number of days with the cost of human day.

**Programming Cost**

The cost is also calculated by calculating the number of human days spends on the coding of the project and then multiplying the number of the days with the cost of human day.

**Stationary and Miscellaneous Expenses**

The cost of computer stationary is less than the cost of other not computer based stationary.

**2.1.2 Technical Feasibility**

A study of functions, performance and constraints which may affect the ability to achieve an acceptable system included in it. Technical analysis begins with an assessment of the technical reliability of the proposed system.

* What technologies are required for accomplished system function and performance?
* What new materials, methods, algorithms, or processes are required and what is their development risk?

As the software is very much economically feasible, then it is really important for it to be technically sound. The software will be build among:

|  |  |
| --- | --- |
| Front-End | HTML,CSS |
| Logical development | PHP |
| Back-End | MYSQL |

The existing hardware and software facilities support the proposed system. Computers and storage media are available and the software can

Be developed. Basic technical requirement of the system and all aspects that the existing system facilities.

Hardware: There’s need of Pentium machines with windows 2007 Server and printer for reports.

Manpower: The technical and non-technical staff required to implementing this system.

**2.1.3 Operational Feasibility**

The present system is operationally feasible, as it has become easy to have details regarding which user has logged on the system and time of login, time of logout, and what information is transferred.

The above details regarding the feasibility study show that the design of proposed system is very effective.

**About Project**

Before the analysis is done, the feasibility study needs to be done so as to check whether the system is feasible to run on the machines or not. Each and every aspect was judged and after this study, the project team had been assigned which includes many groups each assigned with team leader.

The various schedules were assigned keeping in mind the cost and performance of the system. The hardware requirements were taken into considerations so that the system runs on a proper platform.

Each and every system requires that the cost being taken into consideration. Our system too had considered this. The system is considered best which is being produce with better efficiency with reasonable cost and we have tried to achieve this target so as to make our system efficient in every aspect.

**3. SYSTEM Analysis**

**3.1 Requirement Specification**

A requirement is a feature that must be included in the system, before the actual design and implementation start; getting to know the system to be implemented is of prime importance. Main emphasis should be on:

* The output of the system
* The output expected from the system
* The people involved in the working of the system
* The volume of DATA(INPUTS) and the amount of INFORMATION (OUPTUTS) that will be involved

With respect to the system itself, the following facts should be taken into consideration:

* The major process involved
* The main points of application
* The processing rules for the collected data
* The expectations that may be present
* The checks that should be in place in order to avoid wrong entries

Basically, Structure Analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specifications that are easily understandable to the user. The structured tools focus on the tools – which are essentially the DFD, Structured English, Decision Trees & Decision Tables. The objective is to build a new document, called System Specification. This document provides the basis for design and implementation. It must be noted that this phase focuses on the logical development of the system and not the physical development. We must establish a man–machine interface in this phase.

Tasks which are performed as analyst:

* Gathered all facts about the present system from the employees
* Studied the strengths and weaknesses of the current system
* Determined what must be done to solve the problem
* Prepare the functional specification document

In order to reduce the time, there is need for computerized system that can retrieve data, insert data, and update existing data. This software is developed with the aim of reducing time, manpower and cost so that the records can be maintained easily. The volume of work and complexity are increasing year by year. This system reduces complexity and workload.

**3.2 System Requirements**

**3.2.1 Hardware & Software Requirements**

|  |  |
| --- | --- |
| **Client Side Tools** | |
| **Processor** | PC with a Pentium III-class processor, 600 MHz processor  Recommended: 1 GHz processor |
| **RAM** | 192 MB or onwards Recommended. |
| **Hard Disk** | 1GB of available space required on system drive of available or more. |
| **Operating System** | Windows XP Service Pack 2, Windows Server 2003 Service Pack 1, or later versions  Window 7, 8 or later versions support will be provided |
| **Access** | MySQL v5.1.36. or above |
| **Browser** | Mozilla Firefox 8.0 /Internet Explorer 9.0 Onwards with Player Plug-in 9.0 or onwards. Flash player chrome 15 or above |

|  |  |
| --- | --- |
| **Server Side Tools** | |
| **Processor** | PC with a Pentium III-class processor, 600 MHz processor  Recommended: 1 GHz processor |
| **RAM** | 512 MB or onwards Recommended. |
| **Hard Disk** | 80 GB |
| **Operating System** | Windows XP Service Pack 2, Windows Server 2003 Service Pack 1, or later versions  Window 7, 8 or later versions support will be provided |
| **Software** | MySQL, PHP, Apache server or Wamp server, Microsoft Excel 2003 onwards XAMMP Server (optional) |
| **Browser** | Mozilla Firefox 2.0 /Internet Explorer 7.0 Onwards with Player Plug-in 9.0 or onwards. Flash player chrome 15 or above |

|  |  |
| --- | --- |
| **Server Side Tools** | |
| **Processor** | PC with a Pentium III-class processor, 600 MHz processor  Recommended: 1 GHz processor |
| **RAM** | 512 MB or onwards Recommended. |
| **Hard Disk** | 80 GB |
| **Operating System** | Windows XP Service Pack 2, Windows Server 2003 Service Pack 1, or later versions  Window 7, 8 or later versions support will be provided |
| **Software** | MySQL, PHP, Apache server or Wamp server, Microsoft Excel 2003 onwards XAMMP Server (optional) |
| **Browser** | Mozilla Firefox 2.0 /Internet Explorer 7.0 Onwards with Player Plug-in 9.0 or onwards. Flash player chrome 15 or above |

## TOOLS/PLATFORM:-

Brief description of the platform used is described below.

* **PHP (Hypertext Preprocessor)**

PHP is best summarized as an embedded server - side Web-scripting language that provides developers with the capability to quickly and efficiently build dynamic Web applications. PHP bears a close resemblance; both syntactically and grammatically, to the C programming language, although developers haven’t been shy to integrate features from a multitude of languages, including Perl, Java, and C++. Several of these valuable borrowed features include regular expression parsing, powerful array-handling capabilities, an object-oriented methodology, and vast database support. For writing applications that extend beyond the traditional, static methodology of Web page development (that is, HTML), PHP can also serve as a valuable tool for creating and managing dynamic content, embedded directly beside the likes of JavaScript, Style sheets, WML (Wireless Markup Language) and many other useful languages. Providing hundreds of predefined functions, PHP is capable of handling just about anything a developer can dream of Extensive support is offered for graphic creation and manipulation, mathematical calculations, ecommerce, and burgeoning technologies such as Extensible Markup Language (XML), open

Database connectivity (ODBC),and Macromedia Shockwave. This vast range of capabilities eliminates the need for the tedious and costly integration of several third-party modules, making PHP the tool of choice for developers worldwide. One of the main strengths of PHP is the fact that because it can be embedded directly alongside HTML code, there is no need to write a program that has many commands just to output the HTML. HTML and PHP can be used interchangeably as needed, working alongside one another in unison. With PHP, we can simply do the following:

<html>

<title><? print “Hello world!”; ?></title>

</html>

And Hello world! will be displayed in the Web page title

bar. Interestingly, the single line print statement is enclosed in what are commonly known as PHP’s escape characters (<?… ?>) is a complete program. No need for lengthy prefacing code or inclusion of libraries; the only required code is what is

needed to get the job done!

* **Characteristics of PHP**

PHP is about providing the programmer with the necessary tools to get the job done in a quick and efficient fashion. Five important characteristics make

PHP’s practical nature possible:

* Familiarity
* Simplicity
* Efficiency
* Security
* Flexibility
* One final characteristic makes PHP particularly interesting: it’s free!

# Familiarity

Programmers from many backgrounds will find themselves already accustomed to the PHP language. Many of the language’s constructs are borrowed from C and Perl, and in many cases PHP code is almost indistinguishable from that found in the typical C or Pascal program. This minimizes the learning curve considerably.

# Simplicity

A PHP script can consist of 10,000 lines or one line: whatever you need to get the job done. There is no need to include libraries, special compilation directives, or anything of the sort. The PHP engine simply begins executing the code after the first escape sequence (<?) and continues until it passes the closing escape sequence (?>). If the code is syntactically correct, it will be executed exactly.

# Efficiency

Efficiency is an extremely important consideration for working in a multi-user environment such as the WWW. PHP 4.0 introduced resource allocation mechanisms and more pronounced support for object-oriented programming, in addition to session management features. Reference counting has also been introduced in the latest version, eliminating unnecessary memory allocation.

# Security

PHP provides developers and administrators with a flexible and efficient set of security safeguards. These safeguards can be divided into two frames of reference: system level and application level.

# System-Level Security Safeguards

PHP furnishes a number of security mechanisms that administrators can manipulate, providing for the maximum amount of freedom and security when PHP is properly configured. PHP can be run in what is known as safe mode, which can limit users’ attempts to exploit the PHP implementation in many important ways. Limits can also be placed on maximum execution time and memory usage, which if not controlled can have adverse affects on server performance. Much as with a cgi-bin folder, administrators can also place restrictions on the locations in which users can view and execute PHP scripts and use PHP scripts to view guarded server information, such as the password file. Application-Level Security Safeguards Several trusted data encryption options are supported in PHP’s predefined function set. PHP is also compatible with many third-party applications, allowing for easy-integration with secure ecommerce technologies. Another advantage is that the PHP source code is not viewable through the browser because the script is completely parsed before it is sent back to the requesting user. This benefit of PHP’s server-side architecture prevents the loss of creative scripts to users.

# Flexibility

Because PHP is an embedded language, it is extremely

flexible towards meeting the needs of the developer. Although PHP is generally touted as being used in conjunction solely with HTML, it can also be integrated alongside languages like JavaScript, WML, XML, and many others. Additionally, as with most other mainstream languages, wisely planned PHP applications can be easily expanded as needed. Browser dependency is not an issue because PHP scripts are compiled entirely on the server side before being sent to the user. In fact, PHP scripts can be sent to just about any kind of device containing a browser, including cell phones, personal digital assistant (PDA) devices, pagers, laptops, not to mention the traditional PC. People who want to develop shell-based applications can also execute PHP from the command line. Since PHP contains no server-specific code, users are not limited to a specific and perhaps unfamiliar Web server. Apache, Microsoft IIs, Netscape Enterprise Server, Stronghold, and Zeus are all fair game for PHP's server integration. Because of the various platforms that these servers operate on, PHP is largely platform independent, available for such platforms as UNIX, Solaris, FreeBSD, and Windows 95/98/NT. Finally, PHP offers access to external components, such as Enterprise Java Beans and Win32 COM objects.

**HTML**

Hypertext Markup Language (HTML) is a language for describing how pages of text, graphics, and other information are organized. Hypertext means text stored in electronic form with cross-reference links between pages.

HTML pages are standard interface to the Internet. A web browser just retrieves a file and put it on the screen. It actually assembles the component parts of a page and arranges those parts according to commands hidden in the text by the author of the file. Those commands are written in the HTML. The most popular HTML compatible browsers are Internet Explorer.

* **MySQL**

In today's connected world, data and the systems that manage that data must always be secure yet available to our users. With MySQL , users and information technology (IT) professionals across your organization will benefit from reduced application downtime, increased scalability and performance, and tight yet flexible security controls. MySQL also includes many new and improved capabilities to help make your IT staff more productive.

**3.3 Lifecycle Model**

Types of Life Cycle model:

* + Prototype Model
  + Waterfall Model
  + Iterative Enhancement Model
  + The Spiral Model
  + Dynamic System Development Model

**Prototype Model**

The goal of prototyping based development is to counter the first two limitations of the waterfall model discussed earlier. The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. Development of the prototype obviously undergoes design, coding and testing. But each of these phases is not done very formally or thoroughly. By using this prototype, the client can get an "actual feel" of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.

**Iterative Enhancement Model**

The iterative enhancement life cycle model counters the third limitation of the waterfall model and tries to combine the benefits of both prototyping and the waterfall model. The basic idea is that the software should be developed in increments, where each increment adds some functional capability to the system until the full system is implemented. At each step extensions and design modifications can be made. An advantage of this approach is that it can result in better testing, since testing each increment is likely to be easier than testing entire system like in the waterfall model. Furthermore, as in prototyping, the increments provide feedback to the client which is useful for determining the final requirements of the system.

**Spiral Model**

This is a recent model that has been proposed by Boehm. As the name suggests, the activities in this model can be organized like a spiral. The spiral has many cycles. The radial dimension represents the cumulative cost incurred in accomplishing the steps dome so far and the angular dimension represents the progress made in completing each cycle of the spiral. The structure of the spiral model is shown in the figure given below. Each cycle in the spiral begins with the identification of objectives for that cycle and the different alternatives are possible for achieving the objectives and the imposed constraints.

**Waterfall Model**

We have followed “**Waterfall model**” for software engineering. This method suggests a systematic, sequential approach to software development that being at the system level and progress through analysis, design, coding, testing and maintenance. This model is also called “**Linear Sequential Model**” as one have to follow all the above five stages sequentially. Also called the **“Classic Life Cycle”** is the oldest and the most widely used paradigm for Software Engineering.

System

Engineering

Requirement

Analysis

**Design**

**Coding**

**Testing**

**Maintenance**

**Reason for selecting this model**

* If all the five stages mentioned above are followed sequentially than one may able to prepare the efficient software.
* It is general practice to do complex analysis and review core functionality before actual coding and testing.
* On starting we have specify maintain only employee
* After one month the requirement for the remaining module was given. So, the same steps has to repeat so we need to **move back to start from analysis** stack & that is the main reason to use waterfall model.

**3.4 Task Dependency Diagram**

**Testing**

**Coding & integrating Module**

**Scheduling**

**Designing For Master Module**

**Estimation**

**Requirement Analysis For Master Module**

**Requirement Gathering for Master Module**

**Delivery Confirmation**

**3.6 DFD’s and E-R Diagram**

## DATA FLOW DIAGRAMS:-

It is a graphical representation of flow of data through a system. The basis of DFD is a data flow graph, which pictorially represents transformation on data. DFD’s can represent the system at any level of abstraction.

**Note:**

Represents External Entity

Represents the Process

Represents the flow of Control

Represents Close Database



**Employer DFD**



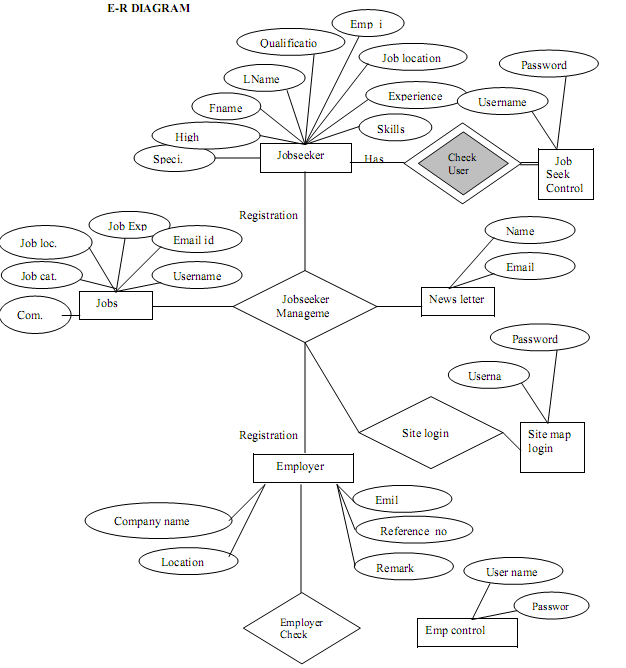
**Job Seeker DFD**

E-R-DIAGRAM:-

The primary purpose of the E-R diagram is to represent the data objects and their relationships.

VARIOUS COMPONENTS USED IN THE GIVEN E-R DIAGRAM:-

|  |  |  |
| --- | --- | --- |
| SYMBOL NAME | SYMBOL | MEANING |
| Rectangle |  | It represents entity sets. |
| Diamond |  | It represents relationship sets. |



**4. System Design and Coding**

**4.1 Goals and activities**

The following design goals that are applicable to virtually every web app regardless of application domain, size or complexity:

**Simplicity**

Although it may seem old-fashioned, the aphorism “all things in moderation” applies to web apps. There is tendency among some designers to provide the end-user with “too much”-exhaustive content, extreme visuals, intrusive animation, enormous web pages.

**Consistency**

This design goal applies to virtually every element of the design model. Content should be constructed consistently. Graphic design should present a consistent look across all parts of the web app. Architectural Design should establish templates that led to consistent hypermedia structures.

**Identity**

The aesthetic, interface and navigational design of a web app must be consistent with the application domain for which it is to be built. The web app architecture will be entirely different interfaces will be constructed to accommodate different categories of users; navigation will be organized to accomplish different objectives.

**Robustness**

Based on the identity that has been established, a web app often makes an implicit “promise” to a user. The user expects robust content and functions that are relevant to the user’s need.

**Navigability**

In web app navigation should be simple and consistent. It should also be designed in a manner that is intuitive and predictable. That is, the user should understand how to move about the web app without having to search for navigation links.

**Compatibility**

A web app will be used in a variety of environment (e.g. different hardware, Internet connection types, operating System, browsers) and must be designed to be compatible with each.

**Design Activity**

In web app there are following Design activity:

**Interface Design**

It describes the structure and organization of the user interface includes a representation of system layout, a definition of the modes of interaction and a description of navigation mechanisms.

**Aesthetic Design**

This is also called Graphic Design. It describe the “look and feel” of the web app. Includes color schemes, geometric layout, size, font and placement, the use of graphics and related aesthetic decisions.

**Content Design**

It defines the layout, structure and outline for all content that is presented as part of the web app. establishes the relationships between content object.

**Navigation Design**

It represents the navigational flow between content objects and for all web app functions.

**Architecture Design**

It identifies the overall hypermedia structure for the web app.

**4.2 Database Structures**

According to the structure the system will be built into the following basic modules.

* **Login:** According to the requirement login is provided to all new users, members and even Admin. This module is integrated with Restriction of unauthorized access . Each user can only access those data, which are required for his/her work.
* **Registration**:-It is a core of this web Portal. It involves creating account, so that all new members can get registered and post his or her scrapes on net member can take unique profile according to his or her choice.

**Scrap Module**—the scrap module provides facility of sending scrap and receiving scraps.

**Security Module**—the security module requires a user to sign on before accessing certain screens, and manages the sign on process

* **Profile Module**—the profile module enables user to change his/her profile and friends ‘profile. The module facilitate user to maintain various type of profiles like personal, professional, social.
* **Friend Invitation Module**—the friend invitation module provides feature of sending invitation to his/her friends.
  + - **Search Your Friend**
    - Name
    - School/College Name
    - City
    - Location
    - Key skills/Key words
    - **CBMS Module:-**

**Key Features of CBMS**

### Store the friend’s details

### Store the Scraps details

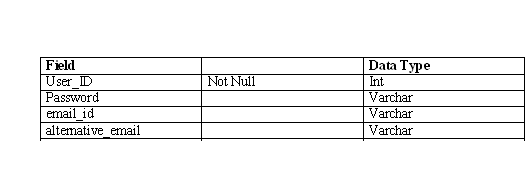
### Block- Listed / Rejected persons

### Raise the community

### View the report

**TABLE NAME:-**

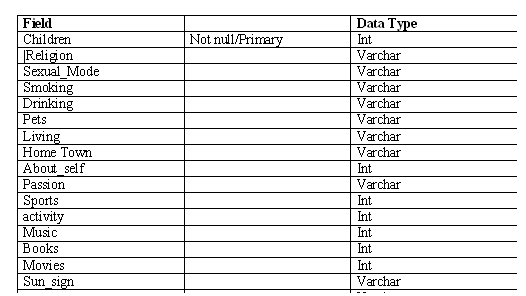
**User\_ Login\_ Detail**



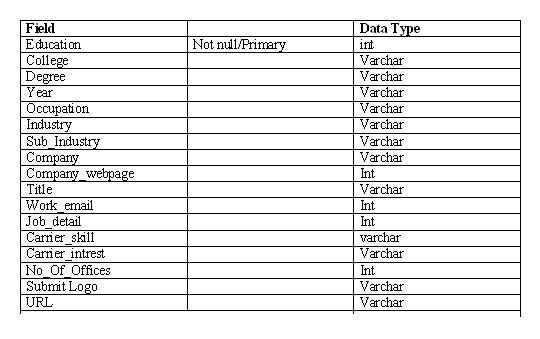
**User\_ Personal\_ Details**

|  |  |  |
| --- | --- | --- |
| Name |  | Varchar |
| Gender |  | Varchar |
| Location |  | Varchar |
| E\_Mail |  | Int |
| Phone\_off |  | Int |
| Address |  | Int |
| Mobile |  | Int |
| E\_Mail |  | Varchar |
| City |  | Varchar |
| State |  | Varchar |
| Postel\_code |  | Varchar |
| Country |  | Varchar |
| User\_id |  | Varchar |
|  |  |  |

# User \_Social\_ Detail



**User \_Professional\_ detail**

****

**Scrap \_details**

|  |  |  |
| --- | --- | --- |
| Field |  | **Data Type** |
| Scrap\_id | Not Null | Int |
| Image\_url |  | Varchar |
| Doa |  | Varchar |
| Screp\_message |  | Varchar |
| To\_user |  | Number |
| From\_user |  | Varchar |
| Video\_url |  | Varchar |
|  |  |  |

**4.3 GENERIC CODES:-**

**Index.php:**

<? php

session\_start();

require 'Libs/config.php';

include 'Libs/header.php';?>

<div class='container'>

<div class='col-md-3 col-sm-3'>

<div class="list-group">

<a href="search.php?q=php&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item active">Php Jobs</a>

<a href="search.php?q=HTML&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">HTML Jobs</a>

<a href="search.php?q=jquery&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">jQuery Jobs</a>

<a href="search.php?q=css&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">CSS Jobs</a>

<a href="search.php?q=.net&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">.Net Jobs</a>

<a href="search.php?q=asp&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">ASP Jobs</a>

</div>

<div class='well well-sm well-statics'>

<h1>Our Statics</h1>

<table class='table table-bordered table-hover table-striped'>

<tr>

<th>Total User</th>

<?php

$totalUsers = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts"));

?>

<td><?php echo $totalUsers['count'];?></td>

</tr>

<tr>

<th>Total Employer</th>

<?php

totalEyr = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts WHERE accountType='employer'"));

?>

<td><?php echo $totalEyr['count'];?></td>

</tr>

<tr>

<th>Total Employee</th>

<?php

$totalEyee = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts WHERE accountType='employee'"));

?>

<td><?php echo $totalEyee['count'];?></td>

</tr>

<tr>

<th>Jobs</th>

<?php

$totalJobs = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobs"));

?>

<td><?php echo $totalJobs['count'];?></td>

</tr>

</table>

</div>

</div>

<div class='col-md-9 col-sm-9'>

<div class='well well-sm'>

<h4 class='text-center mt0'>Search Jobs</h4>

<form action='search.php' name='jobSearch' id='jobSearch' method='get' role='form'>

<div class='form-group'>

<div class="input-group">

<input type="text" name='q' class="form-control" placeholder="eg. php, net, java" aria-label="...">

<div class="input-group-btn">

<input type='hidden' name='jobExp' value='noway'>

<input type='hidden' name='jobCat' value='noway'>

<input type='hidden' name='jobOrder' value='ASC'>

<input type='submit' name='searchJob' value='Search' class='btn btn-primary'>

</div>

</div>

</div>

</form>

</div>

<div class="well well-sm well-white">

<h1 class='text-center mt0 font24'>Jobs By Categories</h1>

<hr class='mt0'>

<div class='row'>

<?php

$jobCategories = mysql\_query("select \* from jobCategories ORDER BY rand() LIMIT 28");

$i=0;

while($jobByCats=mysql\_fetch\_array($jobCategories)):

$i++;

?>

<?php if($i==1||$i==8||$i==15||$i==22): ?>

<div class='col-md-3 col-sm-3'>

<ul class='list-unstyled fa-ul'>

<?php endif; ?>

<li><a href='search.php?q=&jobCat=<?php echo $jobByCats['id']; ?>&jobExp=noway&jobOrder=ASC&searchJob=Search'><i class='fa-li fa fa-caret-right'></i><?php echo $jobByCats['name']; ?></a></li>

<?php if($i==7||$i==14||$i==21||$i==28): ?>

</ul>

</div>

<?php endif; ?>

<?php endwhile;

?>

</div>

</div>

<h1 class='headingStrip mb0'> Latest Jobs</h1>

<table class='table table-bordered table-hover table-striped'>

<?php

$latestJobs = mysql\_query("SELECT \* FROM jobs ORDER BY id DESC LIMIT 7");

while($latestJob=mysql\_fetch\_array($latestJobs)):

echo "<tr>";

echo "<td>";

echo "<span class='jobTitle'>".$latestJob['jobTitle']."</span><br>";

echo "<span class='jobDisc'>";

$words = explode(" ",$latestJob['jobDisc']);

echo implode(" ",array\_splice($words,0,20));

echo "<br>";

echo "<a href='jobView.php?i=".$latestJob['id']."' class='btn btn-xs btn-primary'>Read More..</a>";

secho "</span>";

echo "</td>";

?>

<td><?php echo date('d M,Y',$latestJob['postedOn']);?></td>

</tr>

<?php

endwhile; ?>

</table></div>

</div>

<?php

include 'Libs/footer.php';

?>

**Registration:**

<?php

require 'Libs/config.php';

include 'Libs/header.php';

?>

<div class='container'>

<div class='col-md-6 col-sm-6 col-md-offset-3 col-sm-offset-3'>

<form name='registerForm' id='registerForm' action='<?php echo $\_SERVER['PHP\_SELF'];?>' class='well well-sm' method='post'>

<?php

if(isset($\_POST['registerSubmit'])){

$fName = $\_POST['fName'];

$lName = $\_POST['lName'];

$email = $\_POST['email'];

$mob = $\_POST['mob'];

$accountType = $\_POST['accountType'];

if(empty($fName)){

$er = "Please Enter Your <b>First Name</b>";

$e = 1;

}else if(empty($lName)){

$er = "Please Enter Your <b>Last Name</b>";

$e = 1;

}else if(empty($email)){

$er = "Please Enter Your <b>Email Address</b>";

$e = 1;

}else if(!empty($mob)&&strlen($mob)!=10){

$er = "Please Enter Valid <b>Moile No</b>";

$e = 1;

}else{

$er = "Registration <b>Successful,</b> Please Login Now";

$e = 0;

}

if($e==1){

echo "<div class='alert alert-danger'>";

echo "<p>$er</p>";

echo "</div>";

}else{

$username = $fName.rand(222,39999);

$pass = rand(3333,9999);

$sql = "INSERT INTO accounts SET username='$username', password='$pass', fName='$fName', lName='$lName', email='$email', phoneNo='$mob', accountType='$accountType'";

if(mysql\_query($sql)){

echo "<div class='alert alert-success'>";

echo "<p>$er</p>";

echo "<p>Username: <b>$username</b><br>Password: <b>$pass</b></p>";

echo "</div>";

}else{

echo "<div class='alert alert-danger'>";

echo "<p>Something Is Wrong Here, Notable to Create Record In Database</p>";

echo "</div>";

}

}

}

?>

<div class='row'>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='fName'>First Name:</label>

<input type='text' name='fName' id='fName' class='form-control' placeholder='Please Enter Your First Name' required>

</div>

</div>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='lName'>Last Name:</label>

<input type='text' name='lName' id='lName' class='form-control' placeholder='Please Enter Your Last Name' required>

</div>

</div>

</div>

<div class='form-group'>

<label for='email'>Email:</label>

<input type='email' name='email' id='email' class='form-control' placeholder='Please Enter Your Email' required>

</div>

<div class='form-group'>

<label for='mob'>Moile No:</label>

<input type='number' name='mob' id='mob' class='form-control' placeholder='Please Enter Your Email'>

</div>

<div class='row'>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label>

<input type='radio' value='employee' name='accountType' id='accountType' required> Job Seaker

</label>

</div>

</div>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label>

<input type='radio' value='employer' name='accountType' id='accountType'> Job Poster

</label>

</div>

</div>

</div>

<div class='form-group'>

<input type='submit' name='registerSubmit' id='registerSubmit' class='btn btn-primary btn-sm' value='Register Now!'>

</div>

</form>

</div>

</div>

<?php

include 'Libs/footer.php';

?>

**User\_Account\_Page:**

<?php

require 'Libs/config.php';

session\_start();

if(!isset($\_SESSION['islogin'])){

header('Location: index.php');

exit;

}

include 'Libs/header.php';

if($userDetails['accountType']=='employer'):?>

<div class='container'>

<div class='row'>

<div class='col-md-2 col-sm-3'>

<div class="list-group">

<a href="<?php echo $websiteBase;?>/dashboard.php" class="list-group-item active">Dashboard</a>

<a href="<?php echo $websiteBase;?>/jobsByMe.php" class="list-group-item">My Jobs</a>

<a href="<?php echo $websiteBase;?>/myCompanyProfile.php" class="list-group-item">My Companies</a>

<a href="<?php echo $websiteBase;?>/changePass.php" class="list-group-item">Change Password</a>

</div>

</div>

<div class='col-md-10 col-sm-9 minSide'>

<div class='well well-sm'>

<h4 class='m0'>Employer Job Posting Dashboard</h4>

</div>

<div class='row'>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Job Posted</h5>

<hr class='mt0'>

<?php

$totalJobs = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobs"));

?>

<h1 class='text-center mb0'><?=$totalJobs['count'];?></h5>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Job By Me</h5>

<hr class='mt0'>

<?php

$totalJobsByMe = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobs WHERE userID=".$\_SESSION['islogin']));

?>

<h1 class='text-center mb0'><?=$totalJobsByMe['count'];?></h5>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Job Seakers</h5>

<hr class='mt0'>

<?php

$totalJobsSeakrs = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts WHERE accountType='employee'"));

?>

<h1 class='text-center mb0'><?=$totalJobsSeakrs['count'];?></h5>

</div>

</div>

</div>

</div>

</div>

</div>

<?php elseif($userDetails['accountType']=='employee'): ?>

<div class='container'>

<div class='row'>

<div class='col-md-2 col-sm-3'>

<div class="list-group">

<a href="<?php echo $websiteBase;?>/dashboard.php" class="list-group-item active">Dashboard</a>

<a href="<?php echo $websiteBase;?>/jobsAppliedByMe.php" class="list-group-item">My Applied Jobs</a>

<a href="<?php echo $websiteBase;?>/myProfile.php" class="list-group-item">My Resume</a>

<a href="<?php echo $websiteBase;?>/changePass.php" class="list-group-item">Change Password</a>

</div>

</div>

<div class='col-md-10 col-sm-9 minSide'>

<div class='well well-sm'>

<h4 class='m0'>Job Seaker Dashboard</h4>

</div>

<div class='row'>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Job Posted</h5>

<hr class='mt0'>

<?php

$totalJobs = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobs"));

?>

<h1 class='text-center mb0'><?=$totalJobs['count'];?></h5>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Application</h5>

<hr class='mt0'>

<?php

$totalJobsApplication = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobApplications"));

?>

<h1 class='text-center mb0'><?=$totalJobsApplication['count'];?></h5>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Applied By Me</h5>

<hr class='mt0'>

<?php

$totalJobsApplied = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobApplications WHERE userID=".$\_SESSION['islogin']));

?>

<h1 class='text-center mb0'><?=$totalJobsApplied['count'];?></h5>

</div>

</div>

</div>

</div>

</div>

</div>

<?php elseif($userDetails['accountType']=='admin'): ?>

<div class='container'>

<div class='row'>

<div class='col-md-2 col-sm-3'>

<div class="list-group">

<a href="<?php echo $websiteBase;?>/dashboard.php" class="list-group-item active">Dashboard</a>

<a href="<?php echo $websiteBase;?>/jobsAll.php" class="list-group-item">All Jobs</a>

<a href="<?php echo $websiteBase;?>/applicationsAll.php" class="list-group-item">All Application</a>

<a href="<?php echo $websiteBase;?>/users.php" class="list-group-item">All Users</a>

<a href="<?php echo $websiteBase;?>/users.php?filter=employer" class="list-group-item">Employer Users</a>

<a href="<?php echo $websiteBase;?>/users.php?filter=employee" class="list-group-item">Job Seekers Users</a>

<a href="<?php echo $websiteBase;?>/changePass.php" class="list-group-item">Change Password</a>

</div>

</div>

<div class='col-md-10 col-sm-9 minSide'>

<div class='well well-sm'>

<h4 class='m0'>Admin Dashboard</h4>

</div>

<div class='row'>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Job Posted</h5>

<hr class='mt0'>

<?php

$totalJobs = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobs"));

?>

<h1 class='text-center mb0'><?=$totalJobs['count'];?></h5>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Application</h5>

<hr class='mt0'>

<?php

$totalJobsApplication = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobApplications"));

?>

<h1 class='text-center mb0'><?=$totalJobsApplication['count'];?></h5>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='well well-sm well-white'>

<h5 class='m0'>Total Applied By Me</h5>

<hr class='mt0'>

<?php

$totalJobsApplied = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobApplications WHERE userID=".$\_SESSION['islogin']));

?>

<h1 class='text-center mb0'><?=$totalJobsApplied['count'];?></h5>

</div>

</div>

</div>

<div class='row text-center'>

<img src='assets/img/Admin-icon.png'>

</div>

</div>

</div>

</div>

<?php else: ?>

<?php

endif;

include 'Libs/footer.php';

?>

**User\_Inbox\_page :**

<?php

require 'Libs/config.php';

session\_start();

if(!isset($\_SESSION['islogin'])){

header('Location: index.php');

exit;

}

include 'Libs/header.php';

if($userDetails['accountType']=='employer'):?>

<div class='container'>

<div class='row'>

<div class='col-md-2 col-sm-3'>

<div class="list-group">

<a href="<?php echo $websiteBase;?>/dashboard.php" class="list-group-item ">Dashboard</a>

<a href="<?php echo $websiteBase;?>/jobsByMe.php" class="list-group-item active">My Jobs</a>

<a href="<?php echo $websiteBase;?>/myCompanyProfile.php" class="list-group-item">My Companies</a>

<a href="<?php echo $websiteBase;?>/changePass.php" class="list-group-item">Change Password</a>

</div>

</div>

<div class='col-md-10 col-sm-9 minSide'>

<div class='row'>

<div class='well well-sm'>

<h4 class='m0'>Jobs Posted By Me

<a href='<?php echo $websiteBase;?>/postJob.php' class='btn btn-primary btn-xs pull-right'>Post New Job</a>

</h4>

</div>

</div>

<div class='row'>

<table class='table table-bordered table-striped table-hover'>

<thead>

<tr>

<th>#</th>

<th>Job Title</th>

<th>Posted On</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<?php

$userID = $\_SESSION['islogin'];

$sSQL = mysql\_query("SELECT \* FROM jobs WHERE userID='$userID'");

$i =0;

while($job=mysql\_fetch\_array($sSQL)):

$i++;

?>

<tr>

<td><?php echo $i; ?></td>

<td><?php echo $job['jobTitle'];?></td>

<td><?php echo date('d M,Y h:i',$job['postedOn']);?></td>

<td class='btn-group'>

<a class='btn btn-xs btn-primary' href='jobView.php?i=<?php echo $job['id']; ?>'><span class='glyphicon glyphicon-eye-open'></span></a>

<a class='btn btn-xs btn-danger' href='jobDelete.php?i=<?php echo $job['id']; ?>&back=jobsByMe'><span class='glyphicon glyphicon-trash'></span></a>

</td>

</tr>

<?php endwhile; ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<?php

else: ?>

<?php

header("Location: dashboard.php");

endif;

include 'Libs/footer.php';

?>

**Company\_Profile :**

<?php

session\_start();

if(!isset($\_SESSION['islogin'])){

header('Location: index.php');

exit;

}

require 'Libs/config.php';

include 'Libs/header.php';

if($userDetails['accountType']=='employer'):?>

<div class='container'>

<div class='row'>

<div class='col-md-2 col-sm-3'>

<div class="list-group">

<a href="<?php echo $websiteBase;?>/dashboard.php" class="list-group-item ">Dashboard</a>

<a href="<?php echo $websiteBase;?>/jobsByMe.php" class="list-group-item ">My Jobs</a>

<a href="<?php echo $websiteBase;?>/myCompanyProfile.php" class="list-group-item active">My Companie</a>

<a href="<?php echo $websiteBase;?>/changePass.php" class="list-group-item">Change Password</a>

</div>

</div>

<div class='col-md-10 col-sm-9'>

<div class='row'>

<div class='well well-sm'>

<h4 class='m0'>Update Company Info</h4>

</div>

</div>

<div class='row'>

<form class='well well-sm' action='<?php echo $\_SERVER['PHP\_SELF'];?>' method='post'>

<?php

$userID = $\_SESSION['islogin'];

if(isset($\_POST['companySubmit'])){

$companyName = $\_POST['companyName'];

$companyEmail = $\_POST['companyEmail'];

$companyMob = $\_POST['companyMob'];

$companyAddress = $\_POST['companyAddress'];

$companyDisc = $\_POST['companyDisc'];

if(empty($companyName)){

$er = "Please Enter Your <b>Company Name</b>";

$e = 1;

}else if(empty($companyEmail)){

$er = "Please Enter Your <b>Email</b>";

$e = 1;

}else if(empty($companyMob)){

$er = "Please Enter <b>Contact No</b>";

$e = 1;

}else if(empty($companyAddress)){

$er = "Please Enter <b>Company Address</b>";

$e = 1;

}else if(empty($companyDisc)){

$er = "Please Enter <b>Company Discription</b>";

$e = 1;

}else{

$er = "Record Updated <b>Successfully</b>";

$e = 0;

}

if($e==1){

echo "<div class='alert alert-danger'>";

echo "<p>$er</p>";

echo "</div>";

}else{

$haveCompanyRecord = mysql\_fetch\_array(mysql\_query("SELECT count(\*) FROM companies WHERE userID='$userID'"));

if($haveCompanyRecord[0]==0){

mysql\_query("INSERT INTO companies SET userID='$userID', name='$companyName', email='$companyEmail', contactNo='$companyMob', address='$companyAddress', info='$companyDisc'");

}else{

mysql\_query("UPDATE companies SET name='$companyName', email='$companyEmail', contactNo='$companyMob', address='$companyAddress', info='$companyDisc' WHERE userID='$userID'");

}

echo "<div class='alert alert-success'>";

echo "<p>$er</p>";

echo "</div>";

}

}

$companyDetails = mysql\_fetch\_array(mysql\_query("SELECT \* FROM companies WHERE userID='$userID'"));

?>

<div class='row'>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='companyName'>Company Name:</label>

<input type='text' value='<?php echo $companyDetails['name'];?>' name='companyName' id='companyName' placeholder='Enter Company Name' class='form-control' required>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='companyEmail'>Email:</label>

<input type='email' value='<?php echo $companyDetails['email'];?>' name='companyEmail' id='companyEmail' placeholder='Enter Company Contact Email' class='form-control' required>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='companyMob'>Contact No:</label>

<input type='number' value='<?php echo $companyDetails['contactNo'];?>' name='companyMob' id='companyMob' placeholder='Enter Company Contact No' class='form-control' required>

</div>

</div>

</div>

<div class='row'>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='companyAddress'>Company Address:</label>

textarea rows='7' name='companyAddress' id='companyAddress' class='form-control' placeholder='Address your company' required><?php echo $companyDetails['address'];?></textarea>

</div>

</div>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='companyDisc'>Company Discription:</label>

<textarea rows='7' name='companyDisc' id='companyDisc' class='form-control' placeholder='Discribe your company' required><?php echo $companyDetails['info'];?></textarea>

</div>

</div>

</div>

<div class='form-group'>  
<input type='submit' value='Update Now!' name='companySubmit' class='btn btn-primary btn-sm'>

</div>

</form>

</div>

</div>

</div>

</div>

<?php else: ?>

<?php

header("Location: dashboard.php");

endif;

include 'Libs/footer.php';

?>

**Delete\_job :**

<?php

session\_start();

if(!isset($\_SESSION['islogin'])){

header('Location: index.php');

exit;

}

require 'Libs/config.php';

include 'Libs/header.php';

if($userDetails['accountType']=='employer'):?>

<div class='container'>

<div class='row'>

<div class='col-md-2 col-sm-3'>

<div class="list-group">

<a href="<?php echo $websiteBase;?>/dashboard.php" class="list-group-item ">Dashboard</a>

<a href="<?php echo $websiteBase;?>/jobsByMe.php" class="list-group-item ">My Jobs</a>

<a href="<?php echo $websiteBase;?>/myCompanyProfile.php" class="list-group-item active">My Companie</a>

<a href="<?php echo $websiteBase;?>/changePass.php" class="list-group-item">Change Password</a>

</div>

</div>

<div class='col-md-10 col-sm-9'>

<div class='row'>

<div class='well well-sm'>

<h4 class='m0'>Update Company Info</h4>

</div>

</div>

<div class='row'>

<form class='well well-sm' action='<?php echo $\_SERVER['PHP\_SELF'];?>' method='post'>

<?php

$userID = $\_SESSION['islogin'];

if(isset($\_POST['companySubmit'])){

$companyName = $\_POST['companyName'];

$companyEmail = $\_POST['companyEmail'];

$companyMob = $\_POST['companyMob'];

$companyAddress = $\_POST['companyAddress'];

$companyDisc = $\_POST['companyDisc'];

if(empty($companyName)){

$er = "Please Enter Your <b>Company Name</b>";

$e = 1;

}else if(empty($companyEmail)){

$er = "Please Enter Your <b>Email</b>";

$e = 1;

}else if(empty($companyMob)){

$er = "Please Enter <b>Contact No</b>";

$e = 1;

}else if(empty($companyAddress)){

$er = "Please Enter <b>Company Address</b>";

$e = 1;

}else if(empty($companyDisc)){

$er = "Please Enter <b>Company Discription</b>";

$e = 1;

}else{

$er = "Record Updated <b>Successfully</b>";

$e = 0;

}

if($e==1){

echo "<div class='alert alert-danger'>";

echo "<p>$er</p>";

echo "</div>";

}else{

$haveCompanyRecord = mysql\_fetch\_array(mysql\_query("SELECT count(\*) FROM companies WHERE userID='$userID'"));

if($haveCompanyRecord[0]==0){

mysql\_query("INSERT INTO companies SET userID='$userID', name='$companyName', email='$companyEmail', contactNo='$companyMob', address='$companyAddress', info='$companyDisc'");

}else{

mysql\_query("UPDATE companies SET name='$companyName', email='$companyEmail', contactNo='$companyMob', address='$companyAddress', info='$companyDisc' WHERE userID='$userID'");

}

echo "<div class='alert alert-success'>";

echo "<p>$er</p>";

echo "</div>";

}

}

$companyDetails = mysql\_fetch\_array(mysql\_query("SELECT \* FROM companies WHERE userID='$userID'"));

?>

<div class='row'>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='companyName'>Company Name:</label>

<input type='text' value='<?php echo $companyDetails['name'];?>' name='companyName' id='companyName' placeholder='Enter Company Name' class='form-control' required>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='companyEmail'>Email:</label>

<input type='email' value='<?php echo $companyDetails['email'];?>' name='companyEmail' id='companyEmail' placeholder='Enter Company Contact Email' class='form-control' required>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='companyMob'>Contact No:</label>

<input type='number' value='<?php echo $companyDetails['contactNo'];?>' name='companyMob' id='companyMob' placeholder='Enter Company Contact No' class='form-control' required>

</div>

</div>

</div>

<div class='row'>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='companyAddress'>Company Address:</label>

<textarea rows='7' name='companyAddress' id='companyAddress' class='form-control' placeholder='Address your company' required><?php echo $companyDetails['address'];?></textarea>

</div>

</div>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='companyDisc'>Company Discription:</label>

<textarea rows='7' name='companyDisc' id='companyDisc' class='form-control' placeholder='Discribe your company' required><?php echo $companyDetails['info'];?></textarea>

</div>

</div>

</div>

<div class='form-group'>

<input type='submit' value='Update Now!' name='companySubmit' class='btn btn-primary btn-sm'>

</div>

</form>

</div>

</div>

</div>

</div>

<?php else: ?>

<?php

header("Location: dashboard.php");

endif;

include 'Libs/footer.php';

?>

**Search\_Job:**

<?php

session\_start();

require 'Libs/config.php';

include 'Libs/header.php';

?>

<div class='container'>

<div class='col-md-3 col-sm-3'>

<div class="list-group">

<a href="search.php?q=php&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item active">Php Jobs</a>

<a href="search.php?q=HTML&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">HTML Jobs</a>

<a href="search.php?q=jquery&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">jQuery Jobs</a>

<a href="search.php?q=css&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">CSS Jobs</a>

<a href="search.php?q=.net&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">.Net Jobs</a>

<a href="search.php?q=asp&jobExp=noway&jobCat=noway&jobOrder=ASC&searchJob=Search" class="list-group-item">ASP Jobs</a>

</div>

<div class='well well-sm well-statics'>

<h1>Our Statics</h1>

<table class='table table-bordered table-hover table-striped'>

<tr>

<th>Total User</th>

<?php

$totalUsers = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts"));

?>

<td><?php echo $totalUsers['count'];?></td>

</tr>

<tr>

<th>Total Employer</th>

<?php

$totalEyr = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts WHERE accountType='employer'"));

?>

<td><?php echo $totalEyr['count'];?></td>

</tr>

<tr>

<th>Total Employee</th>

<?php

$totalEyee = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM accounts WHERE accountType='employee'"));

?>

<td><?php echo $totalEyee['count'];?></td>

</tr>

<tr>

<th>Jobs</th>

<?php

$totalJobs = mysql\_fetch\_array(mysql\_query("SELECT count(\*) as count FROM jobs"));

?>

<td><?php echo $totalJobs['count'];?></td>

</tr>

</table>

</div>

</div>

<div class='col-md-9 col-sm-9'>

<div class='well well-sm'>

<h4 class='text-center mt0'>Search Jobs</h4>

<form action='search.php' name='jobSearch' id='jobSearch' method='get' role='form'>

<div class='form-group'>

<input type="text" name='q' value='<?php echo empty($\_GET['q']) ? '' : $\_GET['q']; ?>' class="form-control" placeholder="eg. php, net, java">

</div>

<div class='row'>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='jobExp'>Experience:</label><?php $jobExpSQL = "SELECT \* FROM jobExperience";$jobExps = mysql\_query($jobExpSQL); ?>

<select name='jobExp' id='jobExp' class='form-control'> <option value='noway'>Any</option>

<?php while($jobExp=mysql\_fetch\_array($jobExps)): ?>

<?php $jobExpSlct = ($\_GET['jobExp']==$jobExp['id']) ? 'selected' : ''; ?>

<option value='<?php echo $jobExp['id'];?>' $jobExpSlct><?php echo $jobExp['title']; ?></option>

<?php endwhile; ?>

</select>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='jobCat'>Category:</label>

<?php $jobCatSQL = "SELECT \* FROM jobCategories";

$jobCats = mysql\_query($jobCatSQL); ?>

<select name='jobCat' id='jobCat' class='form-control'>

<option value='noway'>Any</option>

<?php while($jobCat=mysql\_fetch\_array($jobCats)): ?>

<option value='<?php echo $jobCat['id'];?>'><?php echo $jobCat['name']; ?></option>

<?php endwhile; ?>

</select>

</div>

</div>

<div class='col-md-4 col-sm-4'>

<div class='form-group'>

<label for='jobOrder'>Order:</label>

<select name='jobOrder' id='jobOrder' class='form-control'>

<option value='ASC'>NEW - OLD</option>

<option value='DESC'>OLD - NEW</option>

</select>

</div>

</div>

</div>

<div class='form-group'>

<input type='submit' name='searchJob' value='Search' class='btn btn-primary'>

</div>

</form>

</div>

<?php

if(isset($\_GET['searchJob'])):

$q = $\_GET['q'];

$jobExp = $\_GET['jobExp'];

$jobCat = $\_GET['jobCat'];

$jobOrder = $\_GET['jobOrder'];

if(empty($q)):

$theSeacrhSQL = "SELECT \* FROM jobs";

if($jobCat!='noway'):

$theSeacrhSQL .= " WHERE jobCat =".$jobCat;

$xxx = "AND";

else:

$xxx = "WHERE";

endif;

if($jobExp!='noway'):

$theSeacrhSQL .= " ".$xxx." jobExp =".$jobExp;

endif;

if($jobOrder=='ASC'||$jobOrder=='DESC'):

$theSeacrhSQL .= " ORDER BY id ".$jobOrder;

endif;

else:

$theSeacrhSQL = "SELECT \* FROM jobs WHERE jobDisc LIKE '%$q%'";

if($jobExp!='noway'):

$theSeacrhSQL .= " AND jobExp =".$jobExp;

endif;

if($jobCat!='noway'):

$theSeacrhSQL .= " AND jobCat =".$jobCat;

endif;

if($jobOrder=='ASC'||$jobOrder=='DESC'):

$theSeacrhSQL .= " ORDER BY id ".$jobOrder;

endif;

endif;

$searchResult = mysql\_query($theSeacrhSQL);

if($searchResult):

echo "<table class='table table-striped table-border table-hover'>";

$i=1;

while($theSeacrh=mysql\_fetch\_array($searchResult)):

echo "<tr>";

echo "<td>$i</td>";

echo "<td>";

echo "<span class='jobTitle'>".$theSeacrh['jobTitle']."</span><br>";

echo "<span class='jobDisc'>";

$words = explode(" ",$theSeacrh['jobDisc']);

echo implode(" ",array\_splice($words,0,20));

echo "<br>";

echo "<a href='jobView.php?i=".$theSeacrh['id']."' class='btn btn-xs btn-primary'>Read More..</a>";

echo "</span>";

echo "</td>";

echo "<td>".date('d M,Y h:i',$theSeacrh['postedOn'])."</td>";

echo "</tr>";

$i++;

endwhile;

echo "</table>";

else:

echo "<div class='well well-sm'>";

echo "<h1>Sorry, No Result Found</h1>";

echo "</div>";

endifendif;

?>

</div>

</div>

<?php

include 'Libs/footer.php';

?>

**Contact\_Us :**

<?php

session\_start();

require 'Libs/config.php';

include 'Libs/header.php';

?>

<div class='container'>

<div class='col-md-8 col-sm-8'>

<?php

if(isset($contactSubmit)):

$name = $\_POST['name'];

$mob = $\_POST['mob'];

$email = $\_POST['email'];

$message = $\_POST['message'];

$errorMessage = "";

$errorFlag = 1;

if(empty($name)):

$errorMessage = "Please Enter Your Name";

$errorFlag = 1;

elseif(empty($mob)||is\_int($mob)||strlen($mob)!=10):

$errorMessage = "Please Enter A Valid Mobile No";

$errorFlag = 1;

elseif(empty($email)||!filter\_var($email,FILTER\_VALIDATE\_EMAIL)):

$errorMessage = "Please Enter A Valid Email Address";

$errorFlag = 1;

elseif(empty($message)||strlen($message)<10):

$errorMessage = "Please Enter Message and Should be of minimum 10 words";

$errorFlag = 1;

else:

$errorMessage = "Mail Successfully Send";

$errorFlag = 0;

endif;

if($errorFlag==1):

echo "<div class='alert alert-danger'>";

echo $errorMessage;

echo "</div>";

else:

echo "<div class='alert alert-success'>";

echo $errorMessage;

echo "</div>";

endif;

endif;

?>

<form action='<?php echo $\_SERVER['PHP\_SELF'];?>' method='post' name='contactUs' id='contactUs' class='well well-sm'>

<div class='row'>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='name'>You Name:</label>

<input type='text' name='name' id='name' class='form-control' placeholder='Please Enter Your Name' required>

</div>

</div>

<div class='col-md-6 col-sm-6'>

<div class='form-group'>

<label for='mob'>Mobile No:</label>

<input type='number' name='mob' id='mob' class='form-control' placeholder='Please Enter Your Mobile No'>

</div>

</div>

</div>

<div class='form-group'>

<label for='email'>Email:</label>

<input type='email' name='email' id='email' class='form-control' placeholder='Please Enter Your Email' required>

</div>

<div class='form-group'>

<label for='message'>Message:</label>

<textarea name='message' rows='7' id='message' class='form-control' placeholder='Please Enter Your Email' required> </textarea>

</div>

<div class='form-group'>

<input type='submit' name='contactSubmit' id='contactSubmit' value='Send Enquiry' class='btn btn-primary btn-sm'>

</div>

</form>

</div>

<div class='col-md-4 col-sm-4'>

<iframe width="100%" height="250" frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="http://maps.google.com/maps?f=q

&amp;source=s\_q

&amp;hl=en

&amp;geocode=

&amp;abauth=523c47c1mykXIlzZj8OqgayUcQI-a\_Lnm6Y

&amp;authuser=0

&amp;q=Uttam nagar, delhi,+India

&amp;aq=

&amp;vps=3

&amp;jsv=465b

&amp;sll=28.879496,76.58711

&amp;sspn=0.010203,0.021136

&amp;vpsrc=0

&amp;t=h

&amp;num=10

&amp;abstate=A:actbar-saveto

&amp;output=embed">

</iframe>

</div>

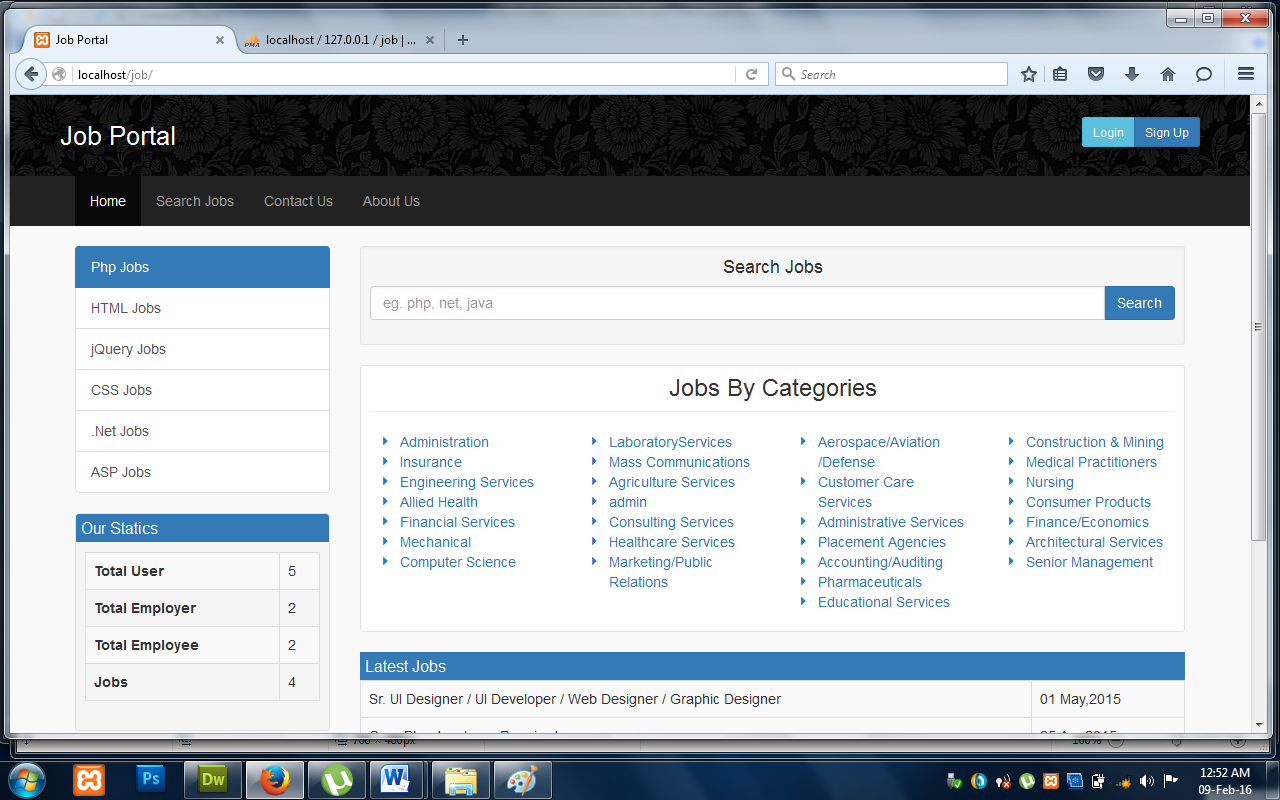
</div>

<?php

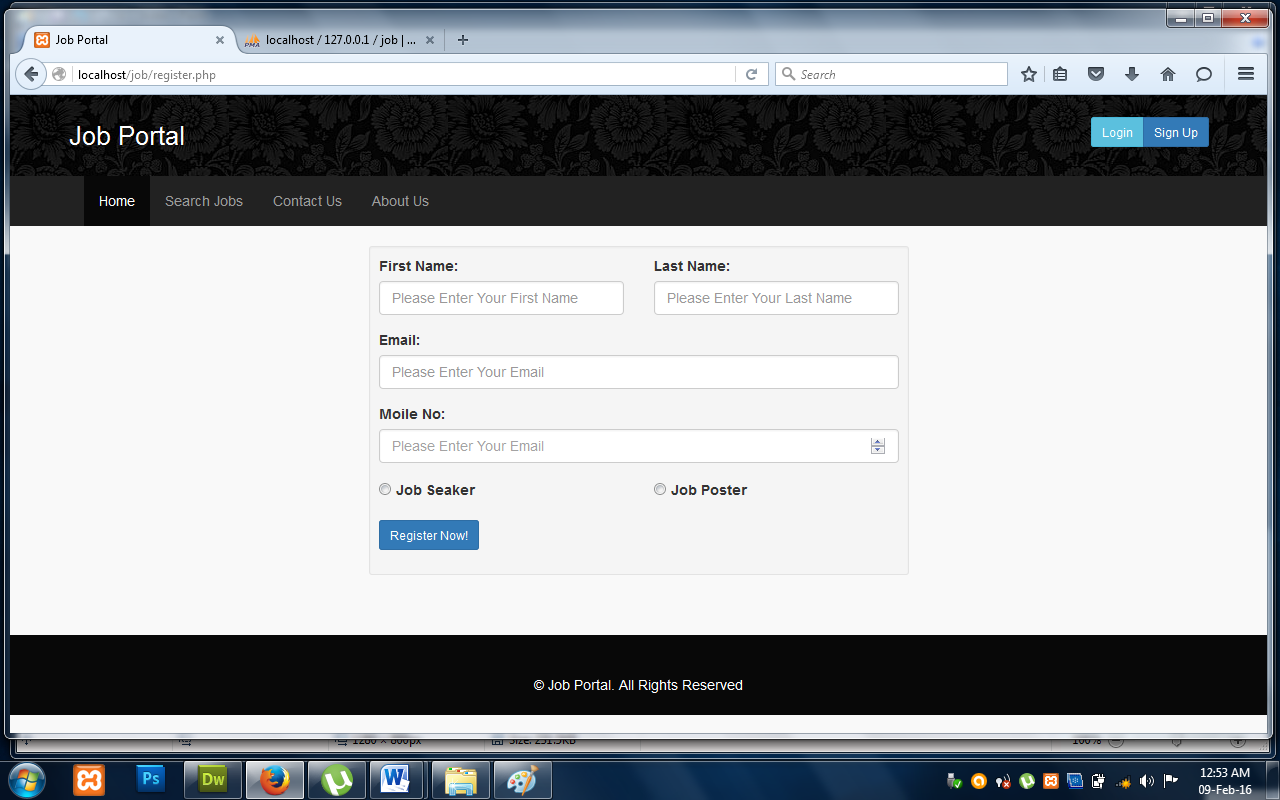
include ' Lib /footer.php';

?>

**4.4 Snapshots**:

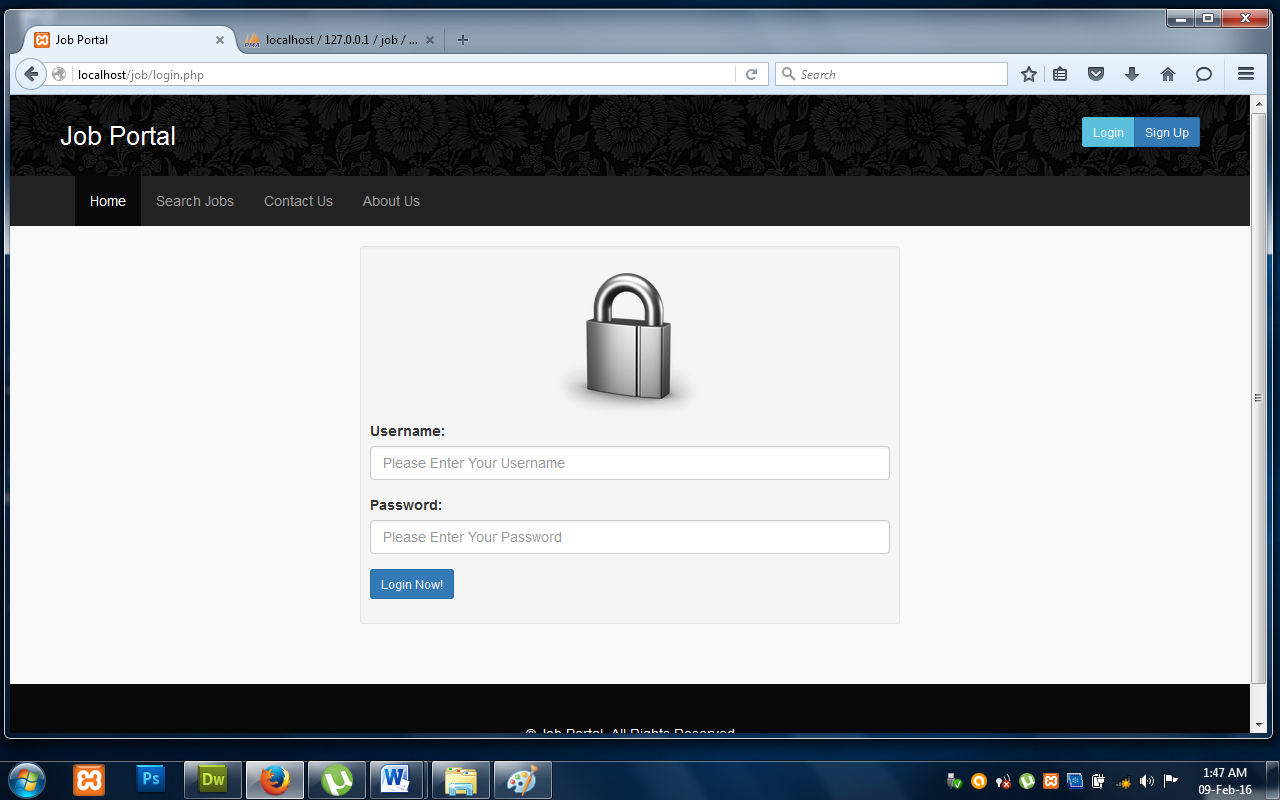
Index\_ page:-

**Figure 4.4.1 Welcome to Index page**

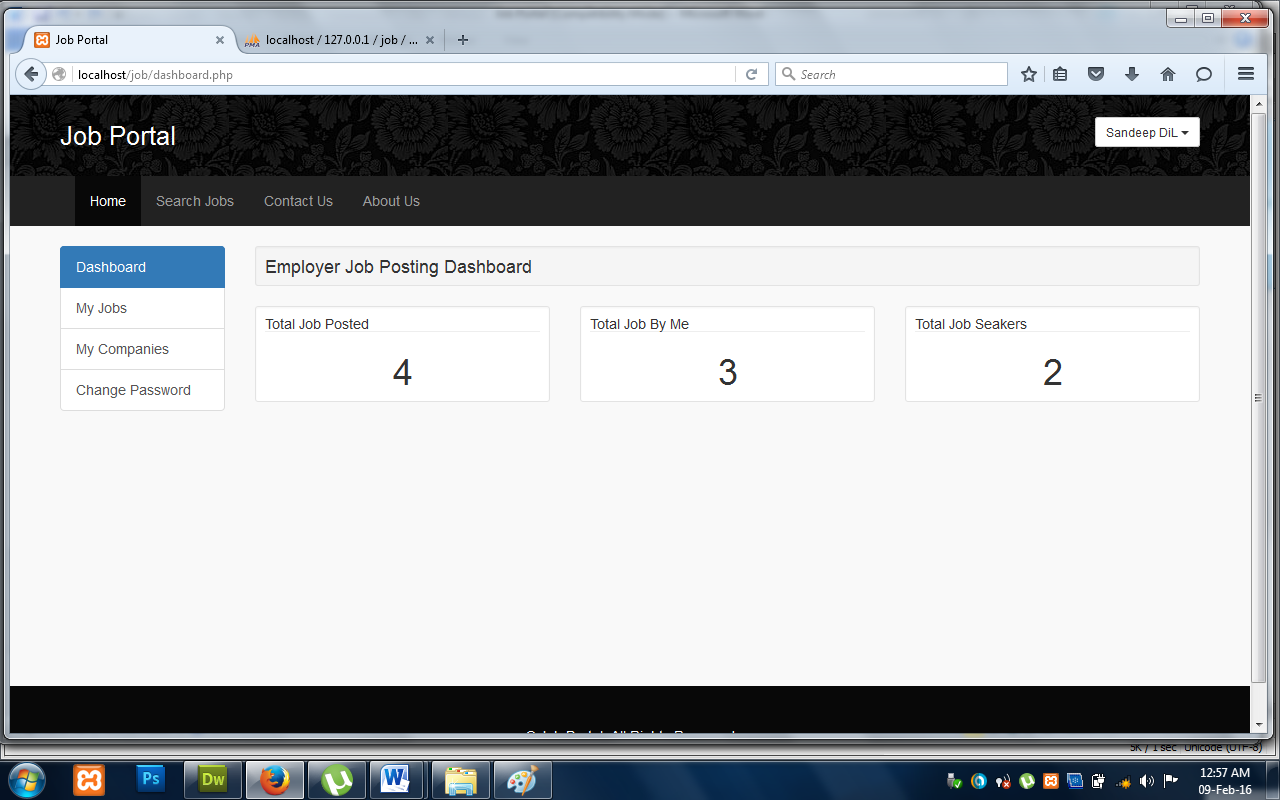
**** User\_ Registration\_ page:-

**Figure 4.4.2 Welcome to User Registration**

User\_ Login\_ page:-

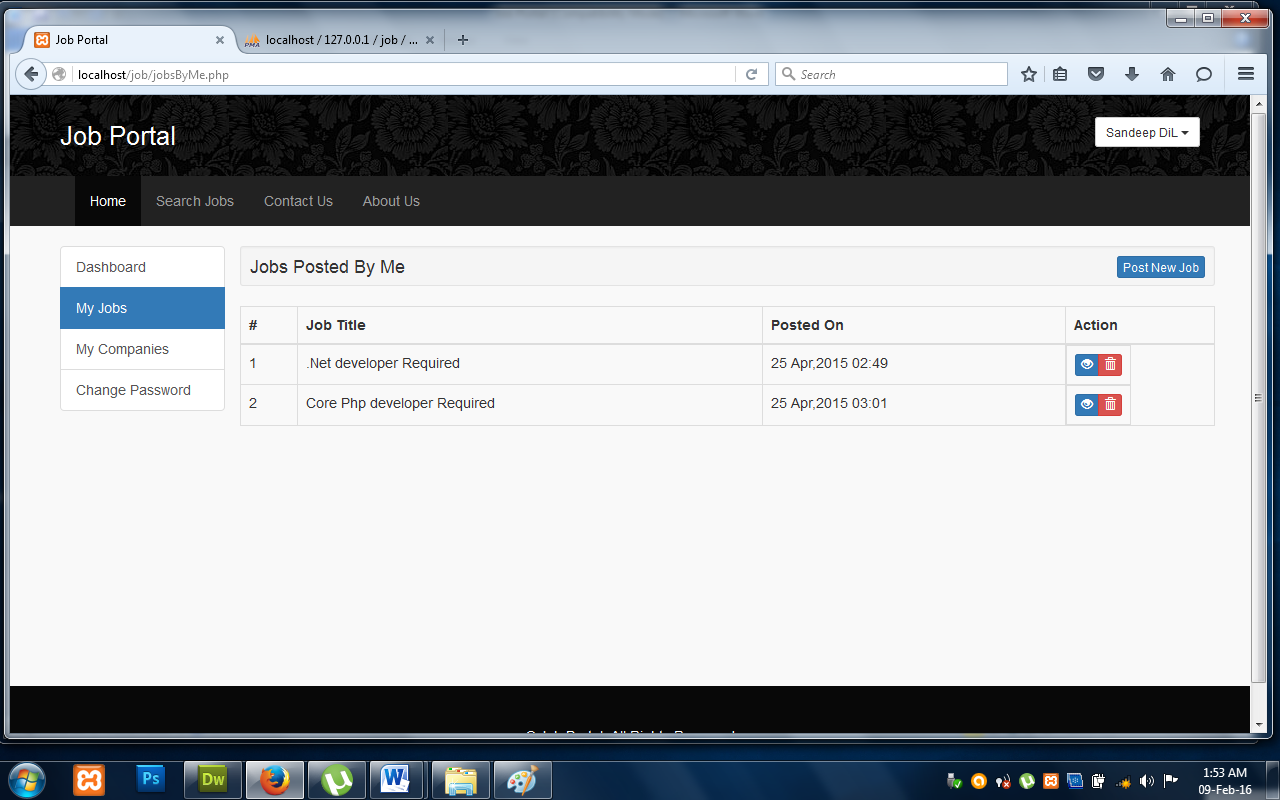


**Figure 4.4.3 Welcome to User Login**

****User\_ Dashboard:-

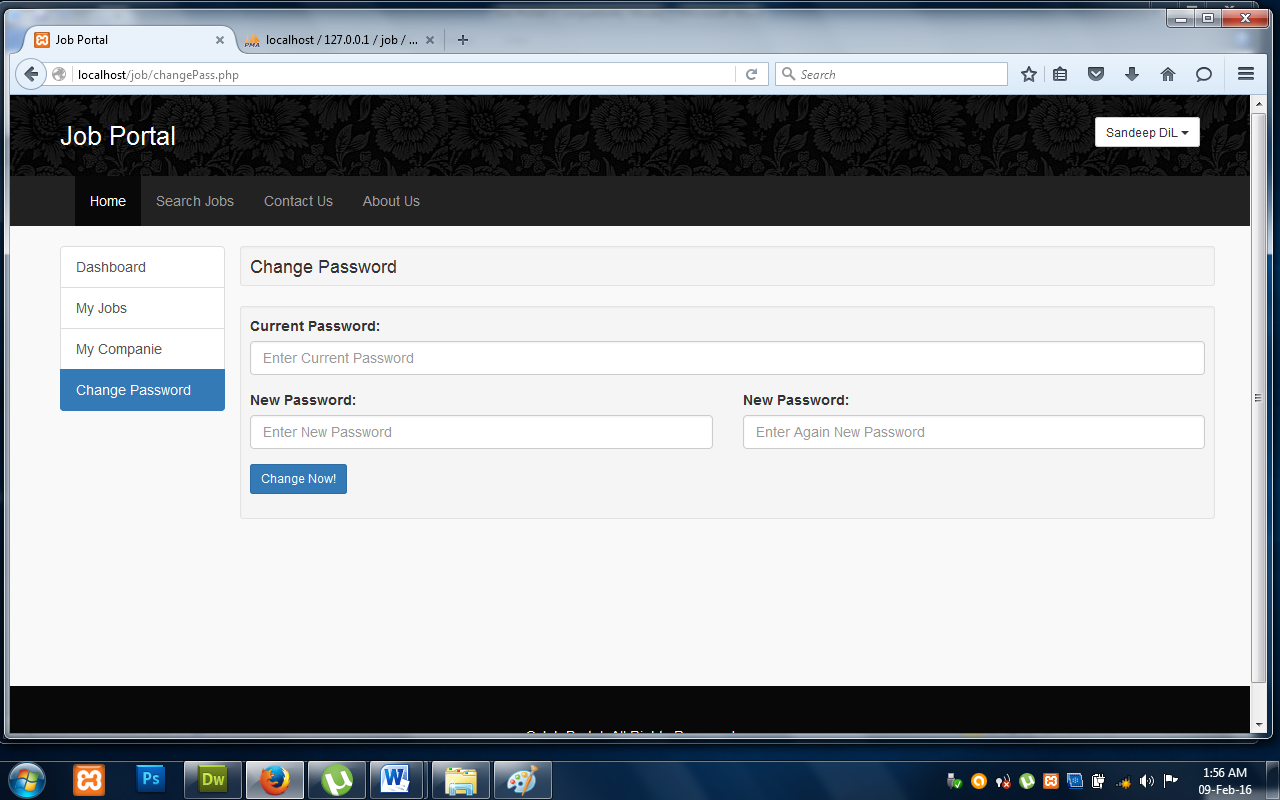
**Figure 4.4.4 Welcome to User Dashboard**

User\_ Inbox:-

****

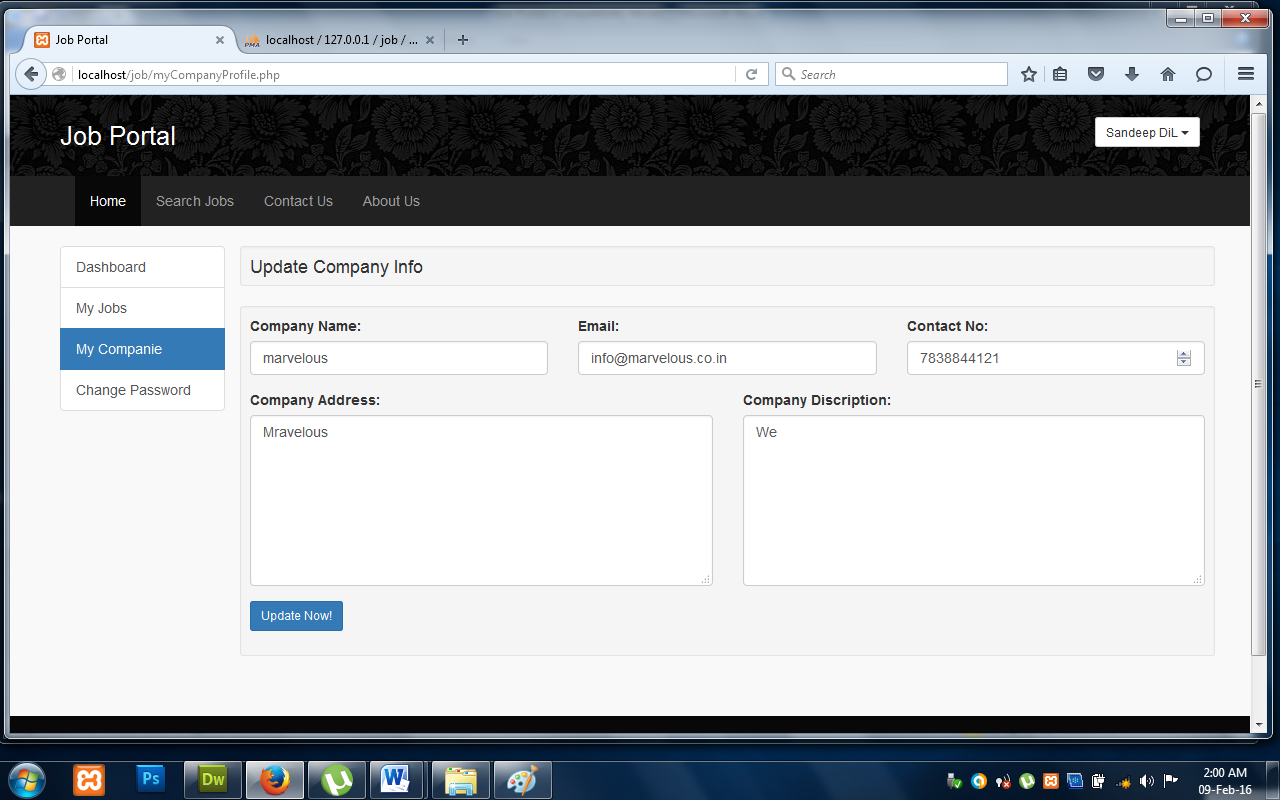
**Figure 4.4.5 Welcome to User Inbox**

User\_Change\_Password :-



**Figure 4.4.6 Welcome to User Change Password**

User\_ Job\_ update :-



**Figure 4.4.7 Welcome to User Job Update**

**5. System Testing and Implementation**

**5.1 Objective of Testing**

Testing is a technique that uncovers error in the web app. There are following objectives of Testing:

* The content model for the web app is reviewed to uncover errors.
* The interface model is reviewed to ensure that all the use-cases can be accommodated.
* The design model for the web app is reviewed to uncover navigation errors.
* The user interface is tested to uncover errors in presentation and navigation mechanics.
* Selected functional components are unit tested.
* Performance tests are conducted.
* The web app is tested by a controlled and monitored population of end-users; the results of their interaction with the system are evaluated for content and navigation errors.

There are seven testing steps that are performed during this web app:

**Content Testing**

It attempts to uncover errors in content. This testing activity is similar in many respects to copy-editing for a written document. In fact, a large web site might enlist the services of a professional copy editor to uncover typographical errors, grammatical mistakes, errors in content consistency, errors in graphical representations and cross referencing errors.

**Interface Testing**

It exercises interaction mechanisms and validates aesthetic aspects of the user interface. The intent is to uncover errors that result from poorly implemented interaction mechanisms or omissions, inconsistency or ambiguities that have been introduced into the interface inadvertently.

**Navigation Testing**

Navigation Testing applies use-cases, derived as part of the analysis activity, in the design of test cases that exercise each usage scenario against the navigation design. Navigation mechanisms implemented within the interface layout are tested against use-cases to ensure that any errors that impede completion of use-cases are identified and corrected.

**Component Testing**

It exercises content and functional units within the web app. The “unit” of choice within the content architecture is the web page. Each web page encapsulates content, navigation links, and processing elements. A “unit” within the web app architecture may be defined functional component that provides service directly to an end user or an infrastructure component that enables the web app to perform all of its capabilities. It also called “Function Testing”.

**Security Testing**

It incorporates a series of tests designed to exploit vulnerabilities in the web app and its environment. The intent is to demonstrate that a security breach is possible.

**GUI Testing Input/output Testing**

GUI testing is done to ensure the uniform look on feel of the user interface components across the application. All major elements of the graphical interface such as windows, mouse operations etc were validated during GUI testing, various selections were made through mouse and keyboard to ensure that it works both ways. It was tested that appropriate message appear to guide the user through the course of action. It was checked whether all the required outputs are generated and are in the desired and proper format. Also it should serve the purpose for which the application was designed.

**Integration Testing**

Navigation and component testing are used as integration testing. The strategy for integration testing depends on the content and web app architecture that has been chosen. If the content architecture has been designed with a linear, grid or simple hierarchical structure, it is possible to integrate web pages in much the same way as we integrate modules for conventional software. Based testing can be used to integrate the set of web pages required to respond to a user event. Regression Testing is applied to ensure that no side effects occur. Cluster testing integrates a set of collaborating pages. Test cases are derived to uncover errors in the collaborations.

**Configure ration Testing**

It attempts to uncover errors that are specific to a particular client or server environment. A cross reference matrix that defines all probable operating systems, browsers, hardware platforms and communication protocols is created. Tests are then conducted to uncover errors associated with each possible configure ration.

**Performance Testing**

It encompasses a series of tests that are designed to assess 1.how web app response time and reliability are affected by increased user traffic,2.which web app components are responsible for performance degradation and what usage characteristics cause degradation to occur and how performance degradation impacts overall web app objectives and requirements.

**Software Testing:**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is an exposure of a system to trial input to see whether software meets correct output. Testing cannot be determined whether software meets user’s needs, only whether it appears to confirm to requirements. Testing can show that a system is free of errors, only that it contains error. Testing finds errors, it does not correct errors. Software success is a quality product, on time and within cost. Through testing can reveal critical mistakes. Testing should therefore,

* Validate Performance
* Detects Errors
* Identify Inconsistencies

**Test Objective:**

* There is strong evidence that effective requirement management leads to overall project cost savings. The three primary reasons for this are,
* Requirement errors typically cost well over 10 times more to repair than other errors.
* Requirement errors typically comprise over 40% of all errors in a software project.
* Small reduction in the number of requirement errors pays big dividend in avoided rework costs and schedule delays.
* System are not designed as entire systems nor are they tested as single systems the analyst must perform both unit and system testing. For this different level testing are use:

**Unit Testing:**

* In unit testing Module is tested separately and the programmer simultaneously along with the coding of the module performs it.
* In unit testing the analyst tests the programs making up a system. For this reason, unit testing is sometime called program testing. Unit testing gives stress on modules independently of one another, to find errors. This helps the tester in detecting errors in coding and logic that are contained within that module alone. The errors resulting from the interaction between modules are initially avoided.
* Unit testing can be performed from the bottom up, Starting with smallest and lowest-level modules and proceeding one at a time., for each module in Bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embedded within the larger system.

**System Testing:**

* This is performed after the system is put together. The system is tested against the system requirement to check if all the requirements are met and if the system performs of specify by the requirements.
* Testing is an important function to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully activated. Another reason for system testing is its utility as a user-oriented vehicle before implementation.

**System Testing Consists of Following Five Steps:**

1. **Program Testing :**

* A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. It is the responsibility of a programmer to have an error free program. At the time of testing the system, there exist two types of errors that should be checked. These errors are

a) Syntax error

b) Logic error

* A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are shown through error messages generated by the Computer.
* A logic error, on the other hand, deals with incorrect data fields out of range items, and invalid combinations. Since the logical errors are not detected by compile, the programmer must examine the output carefully to detect them.
* When program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of the instructions, must be traced to determine the problem. Breaking the program down into self- contained portion, each of which can be checked at certain key points, facilitates the process.

1. **String Testing :**

* Programs are invariably related to one another and interact in total system. Each program is tested to see whether it confirms to related programs in the system. Each part of the system is tested against the entire module with both test and live data before the whole system is ready to be tested.

1. **System Testing:**

* System testing is designed to uncover weaknesses that were not found in earlier tests. This includes forced system failure and validation of total system, as its user in the operational environment will implement it. Under this testing, generally we take low volumes of transactions based on live data. This volume is increased until the maximum level for each transaction type is related. The total system is also tested for recovery and fall back after various major failures to ensure that no data are lost during the emergency.

**User Acceptance Testing:**

* An acceptance test has the objective if selling the user on the validity and reliability of the system. It verifies that the system’s procedures operate to system specifications and that the integrity of important data is maintained. Performance of and acceptance of the system. After that a comprehensive test report is prepared. This report shows the system’s tolerance, performance range, error rate and accuracy.

**Acceptance Testing:**

* Finally the acceptance testing is perform to demonstrate the system to the client on the real life data of the client and on the operating system used by the client testing is an externally critical and time consuming activity it requires proper planning the process starts with a test plant that must be performed and specifies guide line for testing than for different unit the test is case specification document is produce in which list of all different test cases with expected outputs are put together during testing the specified test cases are executed and actual results are compared with expected output the final output of the testing phases is the test report and error report.

**5.2 Test Case:**

A **test case** in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. The mechanism for determining whether a software program or system has passed or failed such a test is known as a **test MYSQL**.

A test case is usually a single step, or occasionally a sequence of steps, to test the correct behavior/functionalities, features of an application. An expected result or expected outcome is usually given.

Additional information that may be included:

Test case name

Test case objective

Expected result

Actual o/p

The test cases in our project are as follows:-

**Log In:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Value | Valid/Invalid | Result |
| Us User Name | Null | Invalid | Please Enter Your Username |
| Pa Password | Null | Invalid | Please Enter Your Password |
| User Name | Punit@unitbook.com | Valid | Redirect Towards your system |
| P Password | Hello | Invalid | Password Length Minimum 6 character long |
| User Type | Null | Invalid | Please select a valid user type |
| User Type | Faculty | Valid | Redirect Toward your system |

**Appendices**

**Steps to Run Project Are as Follows**

1. Install the XAMP Server if not installed already.
2. Install the SQL server for data base management Or MYSQL.
3. Open the Web site and then use the different services offered in this project.
4. We get the log in page by giving the authentication logged in to the system.
5. From the menu select the type of module you want to work on.
6. Another submenu will open showing the various options related to the chosen module.
7. The output will be shown to the user accordingly.
8. We can Send the receive Mail for the Account.

REFERENCES

**Books Referred:**

* BEGINNING PHP 5 ---DAVE MERCER
* BLACK BOOK HTML ---WILEY DREAMTECH
* PHP AND MYSQL WEB DEVELOPMENT --- LUKEWELLING,LAURA
* MICROSOFT SQL SERVER-2000 ---RANKIN, PAUL & JENSEN
* SQL SERVER-2000 ---DUSAN PETKOVIC
* PHP IN A NUTSHELL --- PAUL HUDSON
* **Websites Referred:**
* [http://www.w3schools.com/PHP](http://www.w3schools.com/PHP%20)
* <http://ific.uv.es/informatica/manuales/php/>
* <http://www.sunilb.com/php/php-tutorials>
* http://[www.mysql.com](http://www.mysql.com)/
* <http://wiki.cihar.com/>

CONCLUSION

A Job Portal system is developed using PHP & SQL fully meets the objectives of the systems for which it has been developed. The system is operated at a high level of efficiency & all the job seekers &user associated with the system understands its advantage. The system solves the problem. It was intended to solve as requirement specification.

# LIMITATIONS

The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.

Training for simple computer operations is necessary for the users working on the system.