Project Report

On

JOB PORTAL

Bachelor in Computer Applications

From

Mahatma Gandhi Kashi Vidyapeeth, Varanasi

Submitted By

Saksham Gupta

(Enroll. No. KA2K20/125407067)

Under the able guidance of

Deepak Singh

MAHADEV P.G. COLLEGE, Varanasi

C E R T I F I C A T E

This is to certify that this project entitled **"JOB PORTAL"** submitted in partial fulfillment of the degree of Bachelor of Computer Applications to **Mahatma Gandhi Kashi Vidyapeeth, Varanasi** through **Mahadev P.G. College**, Varanasi done by Mr. Saksham Gupta Roll no 12520407067 is an authentic work carried out by him at Mahadev P.G. College ,Varanasi under my guidance.

Signature of the Student **Signature of the Guide**

A C K N O W L E D G E M E N T

Many persons have contributed to make this software on "**JOB PORTAL"** a reality. I would especially like to express my appreciation to my parents for his unstinted support, encouragement and his painstakingly and meticulous effort towards developing this software.

I acknowledge the help and cooperation received from all the faculty members of ITC TECHNOSOFT. Several colleagues and students have contributed directly and indirectly to the contents this software, as they had given me numerous ideas. Their criticism gave me the much-needed hints about the areas that needed elaboration and amendments and also to present them with greater clarity.

I sincerely express my gratitude to **Mr. Deepak Singh** the present Managing Director of MSDREAM Technologies for his support and help in the final preparation of this Report.

Finally, I wish to express my sincere thanks to all my family members, especially my Parents for their constant moral support and Encouragement.

I would Welcome Constructive Suggestions to improve this software, which can be implemented in my further attempts.

Thanking you!

Saksham Gupta

Enrollment No. KA2K20/125407067

C O N T E N T

P R E F A C E

When A Computer Software succeeds-when it meets the needs of the people who use it, when it performs flawlessly over along period of time, when it is easy to modify and even easier to use-it can and does change things for the better. But when software fails-when its users are dissatisfied, when it is error prone, when it is difficult to change and even harder to use-bad things can and do happen.

We all want to build software that makes things better, avoiding the bad things that lurk in the shadow of failed efforts. To succeed we need discipline when software is designed and built. Many individuals and companies still develop software haphazardly, even as they build systems to service the most advanced technologies of the day. As a result, the quality of the software that we produce suffers and bad things happen.

This project report is intended to serve as a guide to the software developed on **JOB PORTAL** . I have tried to follow the principles and rules as suggested by the software engineers as far as possible, in order to make this software a Successful one.

The report starts with a comprehensive introduction to the project undertaken as its very First Section. It includes objectives and scope of the project; about the front-end tool used i.e. PHP and the back-end tool i.e. MYSQL. The second part presents and discusses the theoretical background of the project. The third section encompasses all the problems of the software that includes what is expected from the software, the demands and the requirements of the end-users. The fourth part is the System analysis and design section. This part focuses on requirements analysis and specification, analysis issues, detailed procedures and the database designs. In The fifth section, different approaches to formal evaluation and review techniques are explored. The sixth section highlights the methodology adopted for this project. The seventh part suggests the steps required to implement the software on the user machine. The eighth part discusses the hardware and software requirements of the user machines. The ninth part deals with the cost benefit analysis. The tenth section contains the data flow diagrams. The next section is the flowchart part. The next section is for the entity relationship diagram of the project. The thirteenth part explains the methodology used for testing. The fourteenth section is the test report. The fifteenth part is the most important part of the project i.e. the code for the software. The sixteenth part is the user manual section. The seventeenth part is the annexure for the topic that includes some details about the organization, the data dictionary, definitions, acronyms and abbreviations used in the report. The final section is the reference part that contains a list of the books and reports that were referred during the development of the project and the report as well.

The emphasis in this report is to document the important concepts and techniques used for the successful development of this project.

Objectives & Scope of the Project

INTRODUCTION & OBJECTIVE

Introduction

THIS PROJECT IS AIMED AT DEVELOPING AN ONLINE APPLICATION FOR THE JOBS. THE SYSTEM IS AN ONLINE APPLICATION THAT CAN BE ACCESS THROUGHOUT THE ORGANIZATION AND OUTSIDE AS WELL WITH PROPER LOGIN PROVIDED. THIS SYSTEM CAN BE USED AS AN APPLICATION FOR THE JOBS THAT'S WHY WE NAMED OUR PROJECT AS "JOB PORTAL". THROUGH OUR ONLINE PROJECT WE WANT TO PROVIDE JOBS TO THE FRESHER'S AS WELL AS TO THE EXPERIENCED PERSON BECAUSE AFTER RESSETION EVERYONE IS IN NEED OF JOBS. THE JOB PORTAL THAT WE DECIDED TO DEVELOP IS BASED ON THE PHP TECHNOLOGY AND OUR PROJECT IS WEB BASED.

OBJECTIVE

OUR PROJECT PROVIDES THE FACILITY OF MAINTAINING THE INFORMATION OF THE JOBSEEKER AND ALSO CONTAIN THE COMPANIES DETAILS WHO WOULD LIKE TO RECRUIT THE PERSONS ACCORDING TO THEIR ELIGIBILITY CRITERIA. THE MAIN AIM OF OUR PROJECT IS TO MAKE JOB SEARCH EASY, FAST & FLEXIBLE. THE RECRUITER CAN EASILY RECRUITED A PERSON ELIGIBLE FOR THE AVAILABLE JOB AND JOBSEEKER WILL GET A GOOD JOB FOR THAT THEY ARE ELIGIBLE.

Theoretical Background

THEORETICAL BACKGROUND

The "JOB PORTAL" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

This application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is need for the user to use this system. Thus by this all it proves is user – friendly.

This software is provided as an online-only resource so that it may be continually extended and updated. This document begins with a description of the separate applications that the sample application comprises, describes the modular structure of the JOB PORTAL application, and provides an in-depth description of several pieces of the modules.

We provide three logins: --

- Jobseeker Login: -
- Recruiter Login: -
- Admin Login: -

Problem Definition

PROBLEM DEFINITION

PROBLEM STATEMENT:

- JOB PORTAL is an online social network service by which user can establish a network among the people residing in all over the world. All the information can be easily accessed.
- This system provides users to maintain complete record of himself/herself and many more.

NEED OF THE SOFTWARE:

This software is developed keeping in mind the above-mentioned problems. The needs and requirements of the end users are also kept in mind while designing this software

This software is provided as an online-only resource so that it may be continually extended and updated. This document begins with a description of the separate applications that the sample application comprises, describes the modular structure of the JOB PORTAL application, and provides an in-depth description of several pieces of the modules

System Analysis & Design

SYSTEM ANALYSIS

Requirements analysis is a software engineering task that bridges the gap between system level requirements engineering and software design. Requirements engineering activities result in the specification of software's operational characteristics (function, data and behavior), indicate software's interface with other system elements, and establish constraints that software must meet. The most commonly used requirements technique is to conduct a meeting or interview. The first meeting between a software engineer (the analyst) and the customer can be likened to the awkwardness of a first date between two adolescents. Neither person knows what to say or ask; both are worried that they do say will be misinterpreted; both are thinking about where it might lead (both likely have radically different expectations here); both want to get the thing over with, but at the same time, both want it to be a success.

Gause and Weinberg suggest that the analyst start by asking CONTEXT-FREE QUESTIONS. That is, a set of questions that will lead to a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of the first encounter itself.

The goal of the requirements gathering activity is to collect all relevant information from the customer regarding the product to be developed with a view to clearly understanding the customer requirements and weeding out the incompleteness and inconsistencies in these requirements. The requirements analysis activity is begun by collecting all relevant data regarding the product to be developed from the users of the product and from the customer through interviews and discussions.

Data collection is done by taking the copies of the documents involved in its working from the organization.

EXTERNAL INTERFACE REQUIREMENTS

The user interface of software is responsible for all the interactions with the user. Almost every software has a user interface. Many users often judge a software product based on its user interface. An interface that is difficult to use leads to higher level of user errors and ultimately leads to user dissatisfaction. Users become particularly irritated when a system behaves in unexpected ways i.e. issued commands do not carry out actions according to the intuitive expectations of the user. Therefore, sufficient care and attention should be paid to the design of the user interface of any

software product. Development of a good user interface usually takes significant portion of the total system development effort.

This software is developed keeping in mind the basic characteristics of a good user interface. Some features of this software can be stated as:-

- It is very easy to learn. It does not require its users to memorize commands. Neither the users are asked to remember information from one screen to another while performing various tasks using the interface.
- The time and user effort necessary to initiate and execute different commands is minimal.
- It is very attractive to use. It catches user attention and fancy. It is a Graphical-based user interface.
- The users can seek guidance and online help when they are unaware of some features of the software.

PERFORMANCE CONSTRAINTS

- For 100 Entries in the database it takes 2-3 seconds approximately.

DESIGN CONSTRAINTS

SOFTWARE CONSTRAINTS

- The software is to run under Windows Operating System or above versions.

HARDWARE CONSTRAINTS

- The software will run on an Intel Workstation with 1 GB RAM, Running on Windows.

ACCEPTANCE CRITERIA

- Before finally implementing the software at the user's site, the developer must demonstrate that the system works on the source data and other information. The developer will have to show through Test Cases that all the conditions and requirements are satisfied.

FESIBILITY ANALYSIS

Feasibility study is conducted to select the best system that meets the performance requirements. This entails an identification, description, and evaluation of the candidate system, and the section of the best system for the job.

Many feasibility studies are disillusioning for both user and analyst. First the study often pre-supposes that when feasibility of the documents is being prepared, the analysis is in position to evaluate solutions. Second most studies tend to overlook the confusion inherent in the system development … the constraints and the assumed attitudes. If the feasibility study is to serve as decision document, it must answer three key questions: -

* Is there new and better way to do a job that will benefit the user?

* What are the cost and saving of the alternatives?

* What is recommended?

The most successful system projects are not necessary or most visible in business but rather those that truly need user expectations. More projects failed because of inflated expectations than for any other reasons …

There main considerations are involved in feasibility analysis**: -**

TECHNICAL

Technical feasibility centers on the current system and to what extent it can support the proposed system, it includes current computer system specifications such as hardware, software etc. it also involves financial considerations to accommodate the technical enhancements. If the budget is serious constraint, then the project is judged not feasible.

Though the system is developed in the generalized form, which covers all the procedures and operations carried out in any manual library management system. The version used in the system is PHP and MYSQL.

The features embedded in the system are latest and according to the need of the client. Such as the online entry for the complains about h/w and s/w. The backend used is the latest **MYSQL version**, which also supports to view the database contents, relationship with the connected database including the primary, foreign key. Any up gradations needed can be easily made in the source code, thus decreasing the headache of changing each and every code. If in near future, the latest version of PHP and MYSQL Server to be installed then, the source code handling the connection of the database can easily be modified. The codes are easily compatible for the changes, as the latest version doesn't affect the core code.

- MYSQL Server can manage large amount of data and is simple and secure.
- Using PHP helps us to design the look of our application and its windows.
- PHP full open database connectivity (ODBC) with MYSQL Server.
- Data accessibility, response and output require less time.

BEHAVIORAL

People are inherently resistance to change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover, transfers, restraining, and changes in job status. Therefore, it is understandable that the introduction of the candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

The routine operations that are handled manually in the organization take much time in processing. It is very cumbersome job to search about any specific hardware component status, details about the complains of particular date, lab wise details from the huge records of complains. The manual maintenance of the records may kill the time of the Administrator.

So to ease the processing time, which basically includes the complaint list and generating the reports etc. this system provides all these features at just the click of the mouse/pressing of the keys. So it's now the job of this system to handle the operations. The system is capable of handling bulk of records easily. Although, it is compatible to the changes, which might occur, needed for the better and fast services. The facilities provided are knowledge of handling the computer system.

ECONOMICAL

Economic analysis is the most frequently used method for evaluating the candidate system. More commonly known as cost of Benefit Analysis, the procedure is to determine the benefits and savings that are expected from the candidate system and compare them with the costs. If benefit outweighs the cost then the decision is made to design and implementation otherwise further justification or alterations are made in the proposed system.

This project doesn't have many hardware requirements, thus, it requires less costing to install the software on the whole.

Though, from the point of economy, the manual handling of the hardware component is much cheaper and best as compared to computerized systems. This approach normally works very well in any ordinary organization. The major problem starts when the no. of hardware components are starts growing with a time. Manual system needs various registers/books to maintain the daily complain entry, hardware entry done. In case of any misplacement of hardware component, the concerned registers have to be searched for the verification of identifying the status of that component. It is very cumbersome job to maintain all these manually. So it is very easy to maintain all these in the proposed system.

- **COST ANALYSIS-**
  - The cost to conduct investigation was negligible, as the center manager and teachers of center provided most of information.
  - The cost of essential hardware and software requirement is not very expensive.

- Moreover, hardware like Pentium I PC and software like PHP and MYSQL are easily available in the market.

➢ BENEFITS AND SAVINGS-

Cost of the maintenance of the proposed system is negligible.

- Money is saved as paper work is minimized.
- Records are easily entered and retrieved.
- Time is saved as all the work can be done by a simple mouse click.
- The proposed system is fully automated and hence easy to use.
- Since benefits out base the cost, hence our project is economically feasible.

OPERATIONAL FESIBILITY

In this we determine what change will be brought in system, new skills required and other human organization and political aspects.

- Non-Programmers can easily use our software. However, it is desirable that the user has the basic knowledge of the computers.
- Without making any changes in the rules and regulations of the existing system proposed system can easily adopted.

LEGAL FEASIBILITY-

Our Project does not infringe with known acts, status or any pending legislation. Hence it is legally feasible.

SOCIAL FEASIBILITY-

Since, the proposed system is inexpensive, easy to install, reduces human errors, fast and simple, hence socially accepted by all organization and institutes being

directly affected by it.

SYSTEM PLANNING

PROJECT EVALUATION AND REVIEW TECHNIQUE

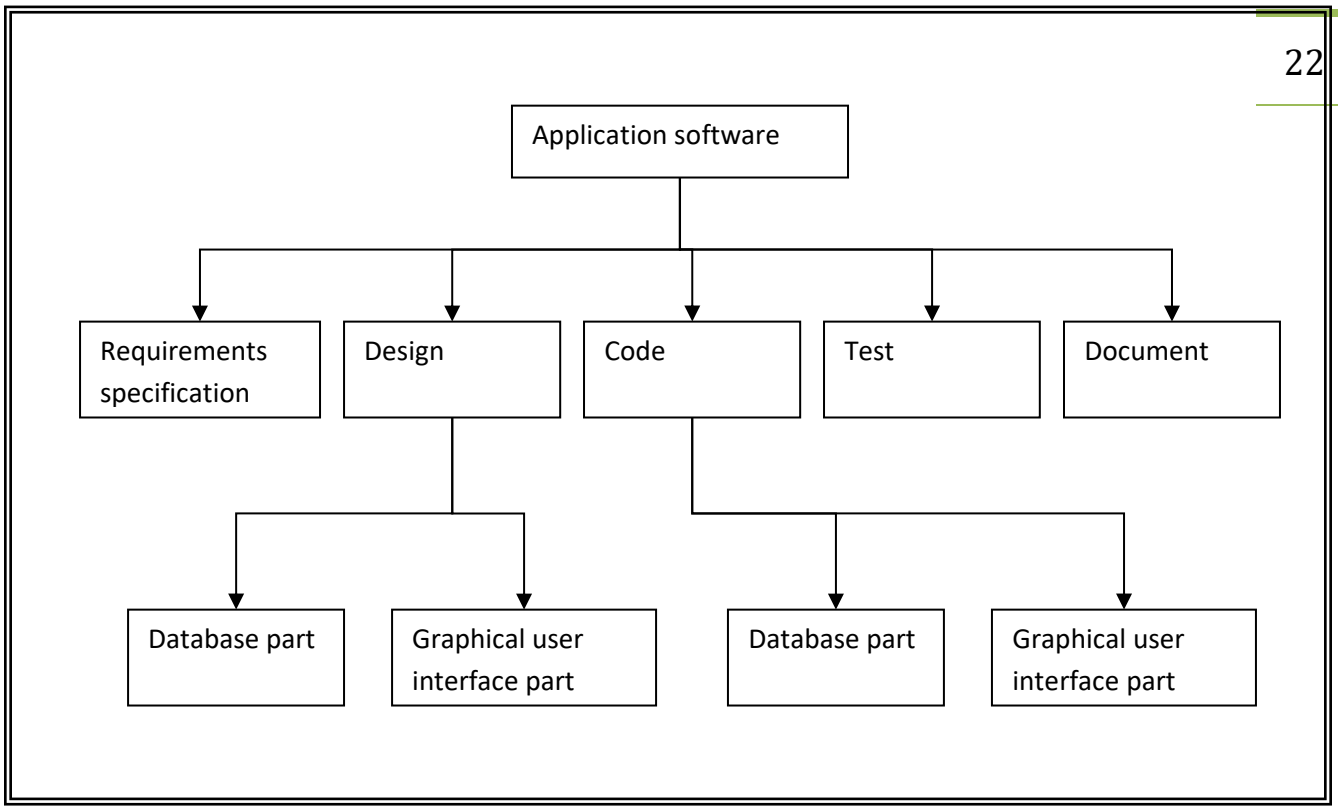
Introduction:

Software project managers take the overall responsibility of steering a project to success. This surely is a very hazy job description. But, it is very difficult to objectively describe the job responsibilities of a project manager.

WORK BREAKDOWN STRUCTURE

Work breakdown structure is used to decompose a given task set recursively into small activities. WBS provides a notation for representing the major tasks needed to be carried out in order to solve a problem. The root of the tree is labeled by the problem name. Each node of the tree is broken down into smaller activities that are made the children of the node. Each activity is recursively decomposed into smaller sub-activities until at the leaf level; the activities require approx. two weeks to develop. If a task is broken down into a large number of very small activities, these can be distributed to a large number of engineers. If the activity ordering permits, the solutions to these can be carried out independently. Thus, it becomes possible to develop the product faster.

The following figure represents the WBS of application software:



PERT CHART

PERT shows precedence relationships among the tasks and various stages of a project. By the helps of PERT chart, a project manager can identify the activities and the amount of time they require, show their interrelationships, specify their sequence, and have a meant of monitoring progress on the project.

PERT makes use of tasks. Like milestone charts, it shows achievements. These achievements however are not task achievements. They are terminal achievements, called EVENTS.

Each activity/Task of the project is represented by a directional are (more commonly known as arrow) pointing in the direction of progress in this project.
The circles represent the beginning or completion of a task. The nodes at the network (also referred as events) establish the relationship among the different activities of the project. The rules are available for construction networks:

1. Each activity is represented by one and only One arrow in the network.
2. Each action must be identifying by two distinct nodes.

This is the table of 'Activity' and it's estimated time duration, which are used to accomplish the project "Firendsword.CO.IN.

Activity	Preprocessor(s)	Duration (In Days)
A: Study		
B: Analysis		
C: Form Design		
D: Coding and Testing		
E: Database Design		
F: Connectivity G: Test with Database		
H: Documentation		
I: Error trapping		
J: Final Test		

Methodology Used

INTRODUCTION

A software life cycle is the series of identifiable stages that a Software product undergoes during its lifetime. The first stage in the life Cycle of any software product is usually the feasibility study stage. The Subsequent stages are: requirement analysis and specification, design, coding, testing and maintenance.

Each of these stages is called a life cycle Phase. A life cycle model represents all the activities required to make a software product transit through its life cycle phases. It also captures the order in which these activities are to be undertaken. The most strict life cycle model used is the Classical Life Cycle Model. However, in any practical software development work, it is not possible to strictly follow the classical waterfall model from every phase to its preceding phases.

METHODOLOGY ADOPTED

- Structured system analysis techniques had been adopted for the analysis of the algorithm and software development.
- Structured system design techniques had been adopted for the design of the algorithm and software development.
- Prototyping model for initial implementation had been used for early testing and module development.
- Prototyping model had been used for the development of the Graphical User Interface.
- Incremental Model had been used for the development of the software.
- Structured development life cycle had been used for this project.
- PERT chart is the primary Tool used for system planning.
- Project cost had been estimated as a function of comparing the cost of similar available products, effort estimation on the project, resources acquired for the project and COCOMO Model as the theoretical reference.
- Test plan was created to form the strategy of testing. This includes the decision of testing techniques, decision of testing tools, and decision of milestones when the testing will commence.

ANALYSIS METHODS

- Structured system analysis techniques are used for the analysis of algorithm and development.
- Background analysis: concept of mining and about the algorithm is learned through research papers on Internet and experts theory.
- Fact-finding: facts about the algorithm are collected from the books and Internet.
- Fact analysis: facts are analyzed through different examples related to algorithm.
- DFDs and flowchart are used to model the algorithm and logic design.

DESIGN METHODS

- Structured design techniques are used to design the software.
- An activity of all phases of a project is the walk-through. It is an interchange of ideas among the team members. In design walk-through, the purpose is to recognize as many problems in the software as possible while they are still “paper tigers”
- Activity diagram are used then to define the behavior of mechanism.
- Layering has been done to refine the architecture. Initially, two layers are introduced, system layer and application layer.
- Demo version is created on these lines of design techniques to demonstrate the feasibility of design methods.
- Database design is according to the implementation of algorithms.

CODING METHODS

- Event driven programming language PHP had been used for coding the modules and programs.
- Structured English and pseudo-codes are used to closely refine the mechanisms using the facility of defined objects.
- Various stubs had been used to facilitate incremental coding followed by testing.
- The basic philosophy followed at this stage:" code one line followed by rigorous testing".
- Integrated development environment of .Net had been used for the development of various modules in integrated manner.
- Incremental compilation had been used to compile and test on which work was in progress.
- Stepwise refinement technique had been used to code the modules.

TESTING METHODS

- Structured testing tools had been used for testing the programs.
- Small programs are developed to test the individual modules.
- Each module is tested by the development of appropriate functionality.
- Groups of people were assigned to test the software functionality.
- The same group is responsible to test the reliability of software by using various inputs and techniques, on various parts and functionalities.
- The group had done Alpha testing.
- Software is distributed to doctors and scientist to Beta-test the entire software.

System Implementation

SYSTEM IMPLEMENTATION

Direct Implementation Method is used to implement the system at the user's site. No special Equipment (Hardware Device) is required to install the

Software. The only requirement is the user must have a Printer Installed & connected to the system for the generation of the various receipts & the reports.

Platform Used

JOB PORTAL System Specification:

The system specification is the final work produced by the system and requirements engineer. It serves as the foundation for hardware engineering, software engineering, database engineering and human engineering.

Once requirements have been gathered explore each requirements in relationship to others, examine requirements for consistency, omissions and ambiguity.

Software platform for development phase

- Operating System : Windows-11Pro
- Editor : Visual Studio Code
- RDBMS/Backend : MYSQL
- Server : Apache/2.4.51 (Win64)
- Server side Scripting language : PHP Version 7.4.26
- Front-end Scripting language : HTML, CSS, Java script
- Browser : Google Chrome

Hardware platform

- Processor : Pentium III or above
- Solid State Drive(SSD) : 512GB
- RAM : 4GB

Client Side

On the client side sole requirements is any HTML compatible browser which supports Java script, XML. For example Microsoft Internet Explorer, Mozilla Firefox, Opera, Microsoft Edge etc.

HTML: - HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML markup.

JAVA SCRIPT: - JavaScript (JS) is an interpreted computer programming language. It was originally implemented as part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that was displayed.

JavaScript is a prototype-based scripting language that is dynamic, weakly typed, and has first-class functions. Its syntax was influenced by the language C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

JavaScript's use in applications outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and frameworks built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications.

JavaScript was formalized in the ECMAScript language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to computational objects within a host environment.

CSS:- Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

PHP: - PHP is a server-side scripting language designed for Web development but also used as a general-purpose programming language. PHP is now installed on more than 20 million Web sites and 1 million Web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group.While PHP originally stood for Personal Home Page, it is now said to stand for PHP: Hypertext Preprocessor, a recursive acronym.

PHP code is interpreted by a Web server with a PHP processor module which generates the resulting Web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform, free of charge.

MySQL: - MySQL is the world's most used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

It is named after co-founder Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

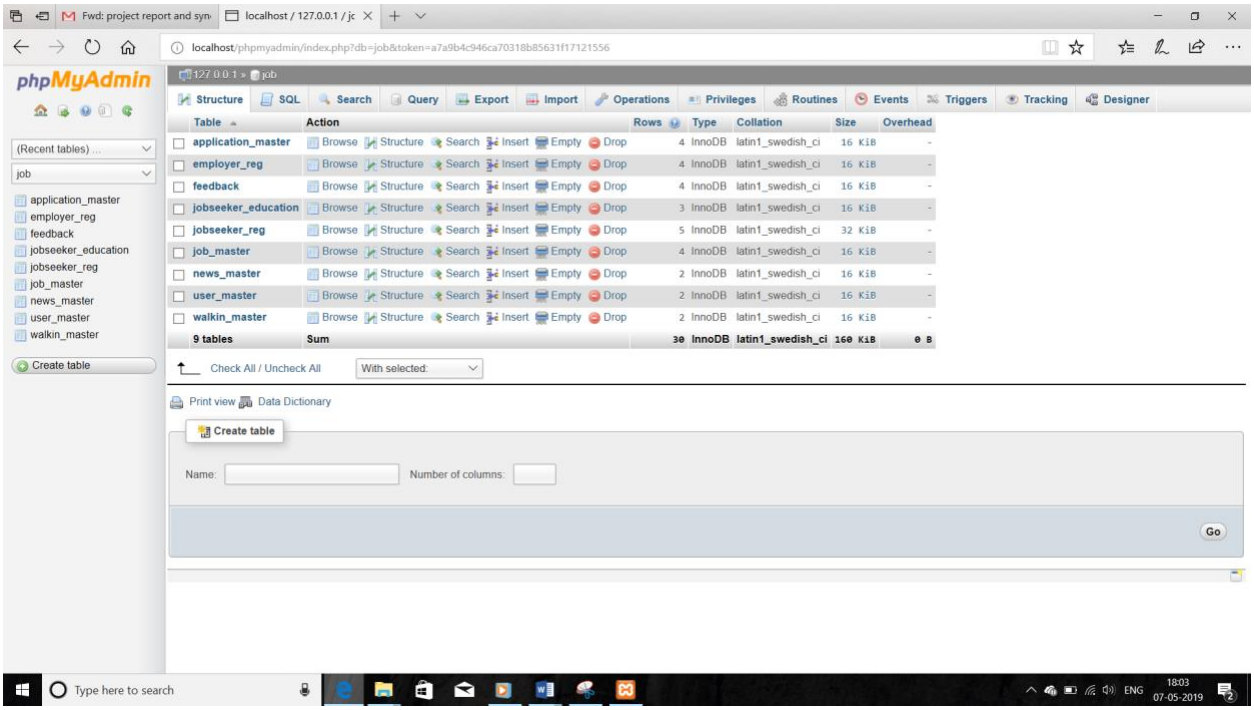
For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, Nokia.com, and YouTube.

MySQL Interface: - MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical Interface: - The official MySQL Workbench is a free integrated environment developed by MySQL AB, which enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

Using the Graphical User Interface

The graphical user interface is a feature of PHPMyadmin.



To start the graphical user interface:

1. Type http://localhost/phpmyadmin/

2. Enter your user name and password. Click Log In.
3. The PHP MyAdmin application window appears.

Command line:- MySQL ships with many command line tools, from which the main interface is 'mysql' client. Third-parties have also developed tools to manage, optimize, monitor and backup a MySQL server, some listed below. All these tools work on *NIX type operating systems, and some of them also on Windows.

Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl. Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Fedora and Ubuntu as well. As of late 2011, Maatkit is no longer developed, but Percona has continued development under the Percona Toolkit brand.

XtraBackup - Open Source MySQL hot backup software. Some notable features include hot, non-locking backups for InnoDB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of IO operations per second, etc.

MySQL::Replication - a replacement for MySQL's built-in replication, developed in Perl. MySQL::Replication can be used to create a peer-to-peer, multi-master MySQL replication network.

MySQL Command List

List of all MySQL commands:

? (\?) Synonym for `help'.

clear (\c) Clear command.

connect (\r) Reconnect to the server. Optional arguments are db and host.

delimiter (\d) Set statement delimiter.

edit (\e) Edit command with $EDITOR.

ego (\G) Send command to mysql server, display result vertically.

exit (\q) Exit mysql. Same as quit.

go (\g) Send command to mysql server.

help (\h) Display this help.

nopager (\n) Disable pager, print to stdout.

notee (\t) Don't write into outfile.

pager (\P) Set PAGER [to_pager]. Print the query results via PAGER.

print (\p) Print current command.

prompt (\R) Change your mysql prompt.

quit (\q) Quit mysql.

rehash (\#) Rebuild completion hash.

source (\.) Execute an SQL script file. Takes a file name as an argument.

status (\s) Get status information from the server.

system (\!) Execute a system shell command.

tee (\T) Set outfile [to_outfile]. Append everything into given

outfile.

use (\u) Use another database. Takes database name as argument.

charset (\C) Switch to another charset. Might be needed for processing

binlog with multi-byte charsets.

warnings (\W) Show warnings after every statement.

nowarning (\w) Don't show warnings after every statement.

System Maintenance

SYSTEM MAINTENANCE

System maintenance denotes any changes made to a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product. However, most products need maintenance due to the wear and tear caused by use. On the other hand, software products do not need maintenance on this count, but need maintenance to correct errors, enhance features, port to new platforms, etc.

Software maintenance is becoming an important activity of large number of organizations. This is no surprise, given the rate of hardware obsolescence, the immortality of a software product *per se*, and the demand of the user community to see the existing software products run on newer platforms, run in newer environments, and/or with enhanced features. When the hardware platform changes and a software product perform some low-level functions, maintenance is necessary. Also, whenever the support environment of a software changes, the software product requires re-work to cope with the newer interface. Thus, every software product continues to evolve after its development through maintenance efforts.

Maintenance covers a wide range of activities including correcting, coding and designing errors, updating documentation and test data and upgrading user support. Enhancement means adding, modifying or re-developing the code to support changes in the specifications. It is necessary to keep up with the changing user needs and the operational environment. The software is designed with the view of easy updating to the software. Future advancements can be done easily through the review of product. The software architecture is strong enough to provide enhancement in functionality, performance and reliability. Functions in the software are designed in such away that they dynamically update on addition of new and revised modules. To append new features in this software, the databases for student details, examination details are sufficient enough for the task. Thus the software is capable enough of being enhanced easily whenever requires by the user.

Cost & Benefit Analysis

COST & BENEFITS ANALYSIS

- **Cost of PC/XT, Printer, Voltage Regulator= Rs. 50,000**
- **Cost of Space (Nil)= no extra space allocated**
- **Cost of System Analysis/ Programmers/ Consultants for two Months= Rs. 10,000**
- **Stationery Cost/ Floppy Cost/ Maintenance/ Electricity= Rs. 900 Per Month**
- **Capital Cost= Rs. 20,000**
- **Recurring Cost= Rs. 800 per month**

Benefits (Direct Savings)

- **Stationery Cost=Rs. 10,000**

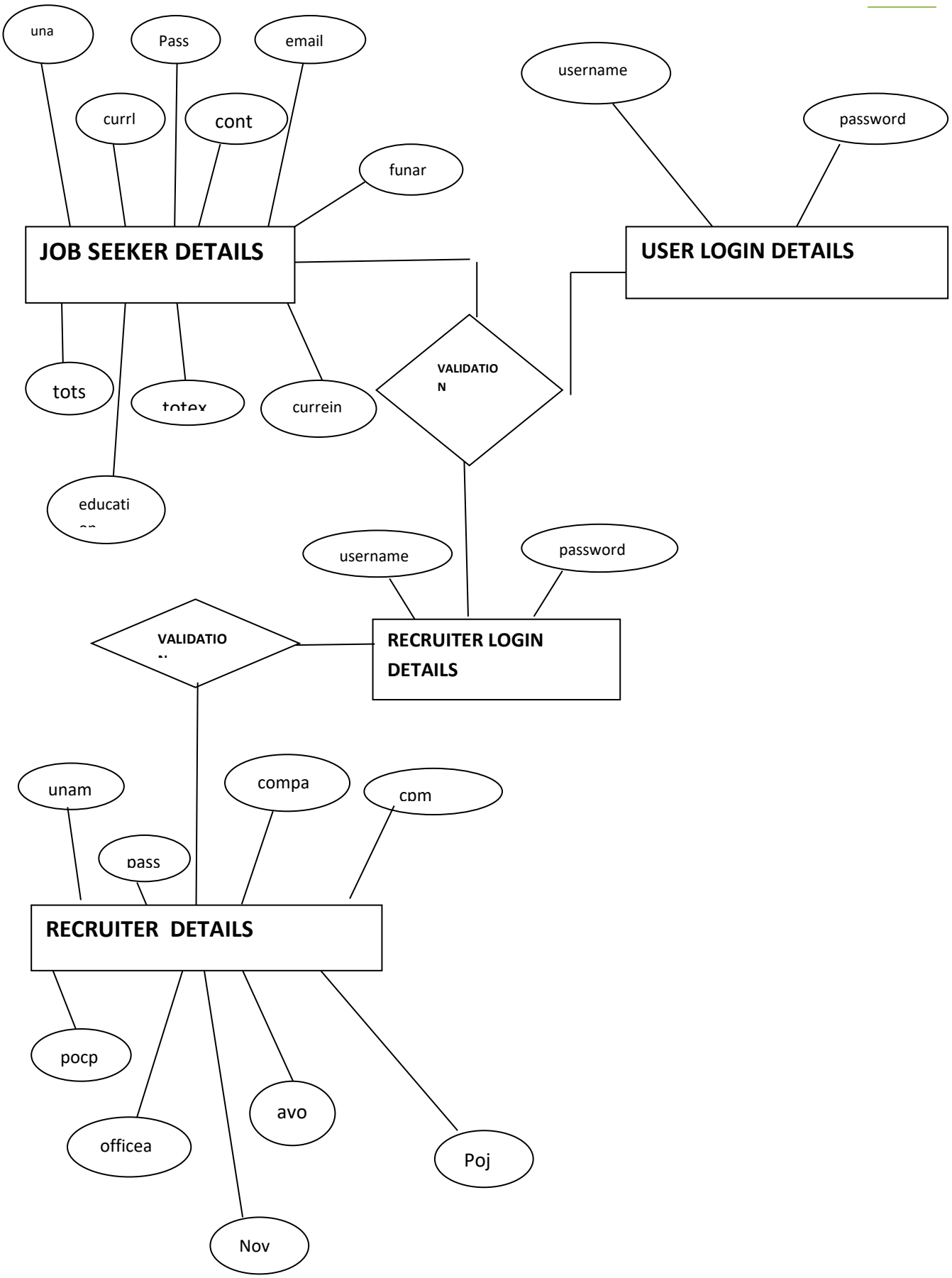
Intangible Benefits

- **Better Performance reflected in the Admission Procedure and Information Retrieval.**

  - **Total Effort = 1000 man-hours**
  - **Total Benefits= Rs. 10,000 per month**
  - **Recurring Cost= Rs. 800 per month**
  - **Net Benefit per Month= Rs. 9200**
  - **Total Cost= Rs. 20, 000**

Life Cycle of the Project

ER-Diagram


una
Pass
email
currl
cont
funar
JOB SEEKER DETAILS
username
password
USER LOGIN DETAILS
tots
totex
currein
VALIDATIO
N
educati
username
password
VALIDATIO
RECRUITER LOGIN
DETAILS
unam
compa
com
pass
RECRUITER DETAILS
pocp
officea
avo
Poj
Nov

DATABASE TABLE

Application Master

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**ApplicationId**	int(11)			No	*None*	AUTO_INCREMENT
2	**JobSeekId**	int(11)			No	*None*	
3	**JobId**	int(11)			No	*None*	
4	**Status**	varchar(30)	latin1_swedish_ci		No	*None*	
5	**Description**	varchar(200)	latin1_swedish_ci		No	*None*	

Employee Registration

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**EmployerId**	int(11)			No	*None*	AUTO_INCREMENT
2	**CompanyName**	varchar(20)	latin1_swedish_ci		No	*None*	
3	**ContactPerson**	varchar(20)	latin1_swedish_ci		No	*None*	
4	**Address**	varchar(100)	latin1_swedish_ci		No	*None*	
5	**City**	varchar(20)	latin1_swedish_ci		No	*None*	
6	**Email**	varchar(40)	latin1_swedish_ci		No	*None*	
7	**Mobile**	bigint(20)			No	*None*	
8	**Area_Work**	varchar(40)	latin1_swedish_ci		No	*None*	
9	**Status**	varchar(10)	latin1_swedish_ci		No	*None*	
10	**UserName**	varchar(20)	latin1_swedish_ci		No	*None*	
11	**Password**	varchar(20)	latin1_swedish_ci		No	*None*	
12	**Question**	varchar(100)	latin1_swedish_ci		No	*None*	
13	**Answer**	varchar(50)	latin1_swedish_ci		No	*None*	

JobSeeker Registration

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**JobSeekId**	int(11)			No	*None*	AUTO_INCREMENT
2	**JobSeekerName**	varchar(20)	latin1_swedish_ci		No	*None*	
3	**Address**	varchar(100)	latin1_swedish_ci		No	*None*	
4	**City**	varchar(20)	latin1_swedish_ci		No	*None*	
5	**Email**	varchar(40)	latin1_swedish_ci		No	*None*	
6	**Mobile**	bigint(20)			No	*None*	
7	**Qualification**	varchar(20)	latin1_swedish_ci		No	*None*	
8	**Gender**	varchar(10)	latin1_swedish_ci		No	*None*	
9	**BirthDate**	date			No	*None*	
10	**Resume**	varchar(200)	latin1_swedish_ci		No	*None*	
11	**Status**	varchar(10)	latin1_swedish_ci		No	*None*	
12	**UserName**	varchar(20)	latin1_swedish_ci		No	*None*	
13	**Password**	varchar(20)	latin1_swedish_ci		No	*None*	
14	**Question**	varchar(100)	latin1_swedish_ci		No	*None*	
15	**Answer**	varchar(50)	latin1_swedish_ci		No	*None*	

JobSeeker Education

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**EduId**	int(11)			No	*None*	AUTO_INCREMENT
2	**JobSeekId**	int(11)			No	*None*	
3	**Degree**	varchar(20)	latin1_swedish_ci		No	*None*	
4	**University**	varchar(100)	latin1_swedish_ci		No	*None*	
5	**PassingYear**	mediumint(9)			No	*None*	
6	**Percentage**	float			No	*None*	

Walkin Master

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**WalkInId**	int(11)			No	*None*	AUTO_INCREMENT
2	**CompanyName**	varchar(20)	latin1_swedish_ci		No	*None*	
3	**JobTitle**	varchar(50)	latin1_swedish_ci		No	*None*	
4	**Vacancy**	int(11)			No	*None*	
5	**MinQualification**	varchar(50)	latin1_swedish_ci		No	*None*	
6	**Description**	varchar(100)	latin1_swedish_ci		No	*None*	
7	**InterviewDate**	date			No	*None*	
8	**InterviewTime**	time			No	*None*	

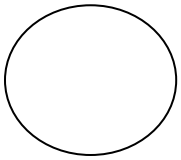
Data Flow Diagram

INTRODUCTION

The DFD also known as the Bubble Chart is a simple graphical formalism that can be used to represent a system in terms of the input data to the system. Various processing carried out on these data, and the output data generated by the system. The main reason why the DFD technique is so popular is probably because of the fact that DFD is a very simple formalism-it is simple to understand and use. A DFD uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among these functions. Starting with a set of high-level functions that a system performs, a DFD model hierarchically represents various sub functions. The five different types of primitive symbols used for constructing DFDs are:

SYMBOLS USED:

PROCESS:

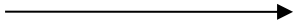


A function is represented using a circle. This symbol is called a process or a bubble. Bubbles are annotated with the names of the corresponding functions.

EXTERNAL ENTITY:

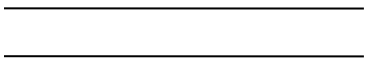
An external entity such as a librarian, a library member, etc. is represented by a rectangle. The external entities are essentially those physical entities external to the software system that interact with the system by inputting data to the system or by consuming the data produced by the system. In addition to the human users, the external entity symbols can be used to represent external hardware and software such as application software.

DATA FLOW:



A directed arc or an arrow is used as a data flow symbol. A data flow symbol represents the data flow occurring between two processes, or between an external entity and a process, in the direction of the data flow arrow. Data flow symbols are usually annotated with the corresponding data names.

DATA STORE:



A data store represents a logical file. It is represented using two parallel lines. A logical file can represent either a data store symbol, which can represent either a data structure, or a physical file on disk. Each data store is connected to a process by means of a data flow symbol. The direction of the data flow arrow shows whether data is being read from or written into a data store. A arrow flowing in or out of a data store implicitly represents the entire data of the data store and hence connecting to a data store need not be annotated with the name of the corresponding data items.

OUTPUT SYMBOL:

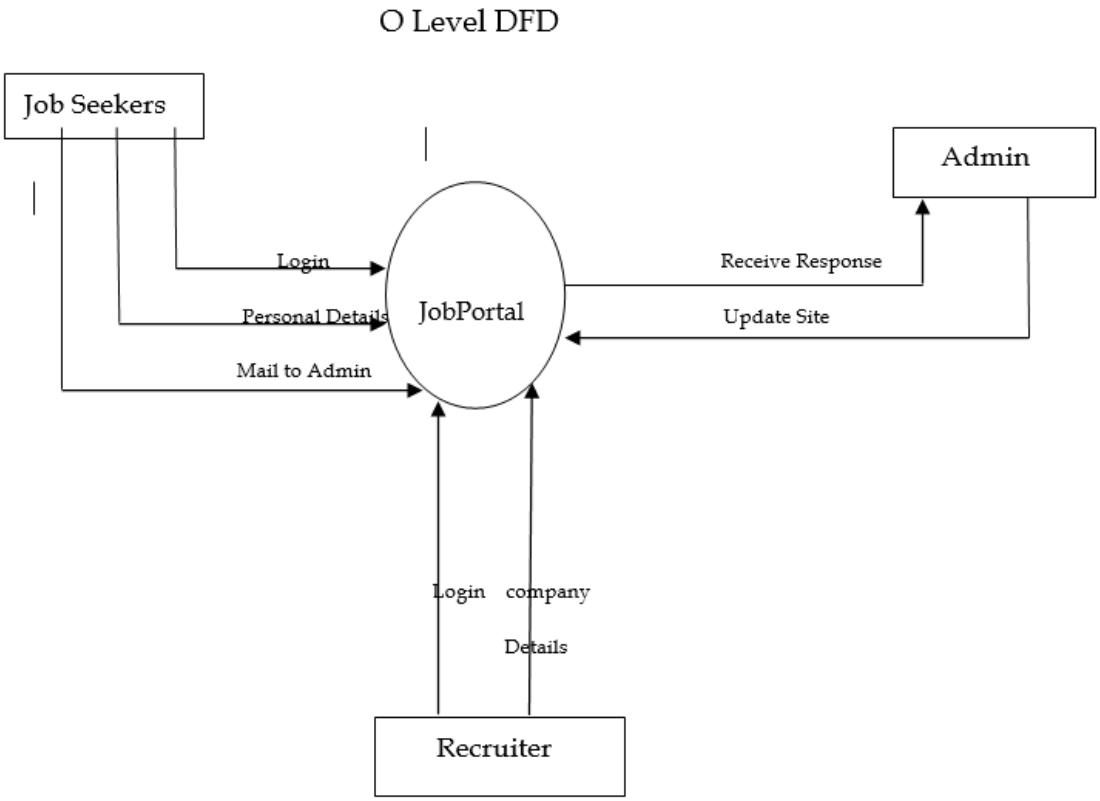
The output symbol is used when a hard copy is produced and the user of the copies cannot be clearly specified or there are several users of the output.

CONTEXT DIAGRAM

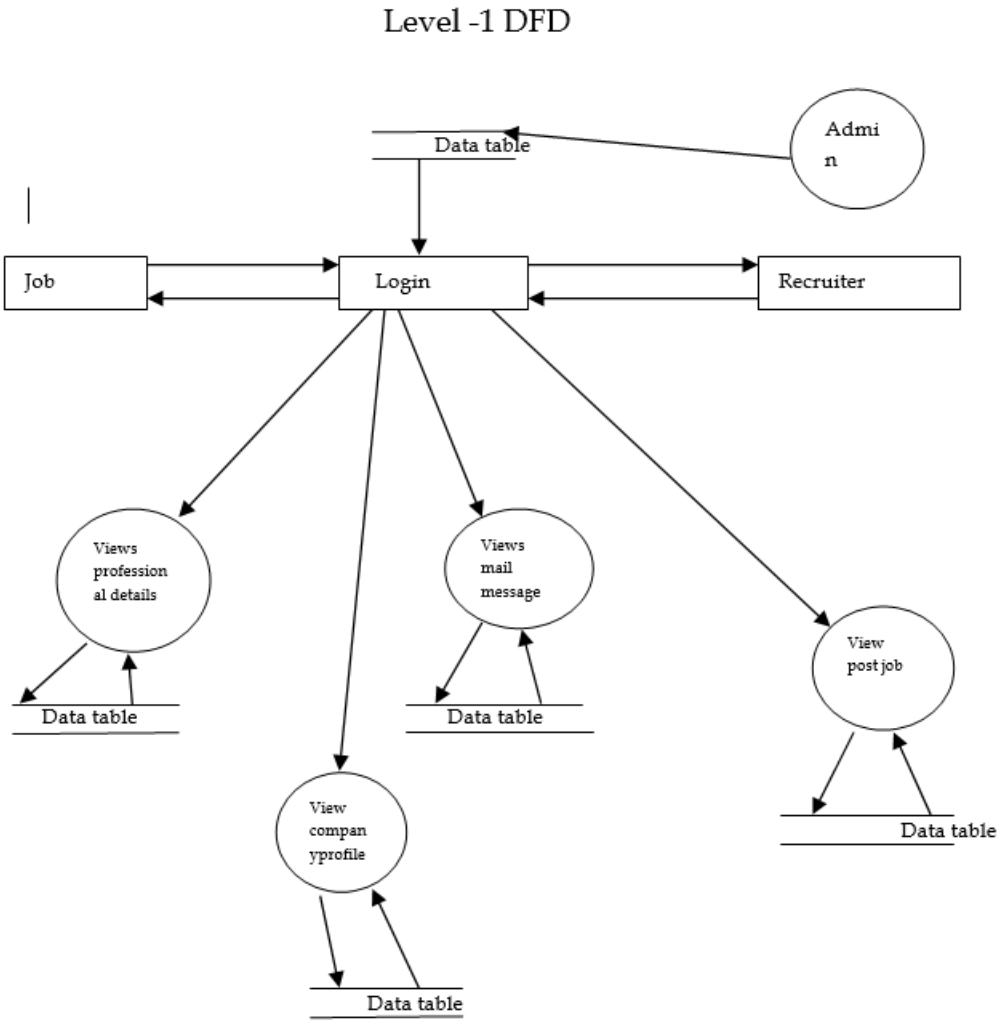
The context diagram is the most abstract data flow representation of a system. It represents the entire system as a single bubble. This bubble is labeled according to the main function of the

system. The various external entities with which the system interacts and the data flows occurring between the system and the external entities are also represented

Context Level DFD



Level-1 DFD


Level -1 DFD
Admi
n
Data table
Job
Login
Recruiter
Views
profession
al details
Views
mail
message
View
post job
Data table
Data table
Data table
View
compan
yprofile
Data table

Input & Output Screens

Index Page

About Us Page

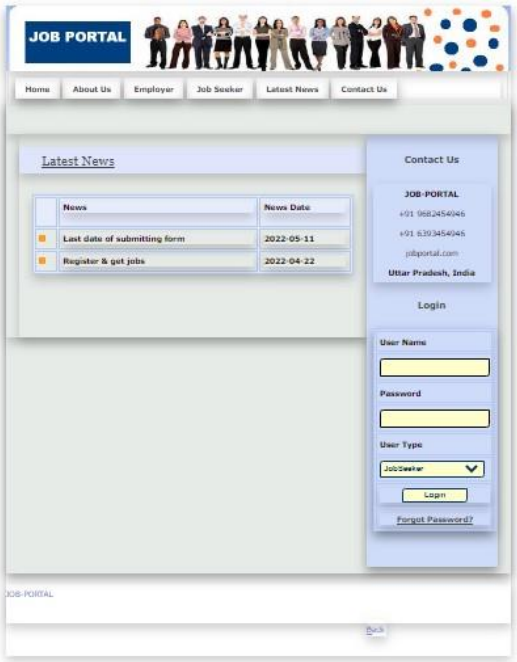
Employer Page

Employer Registration Form

Job Seekers Page

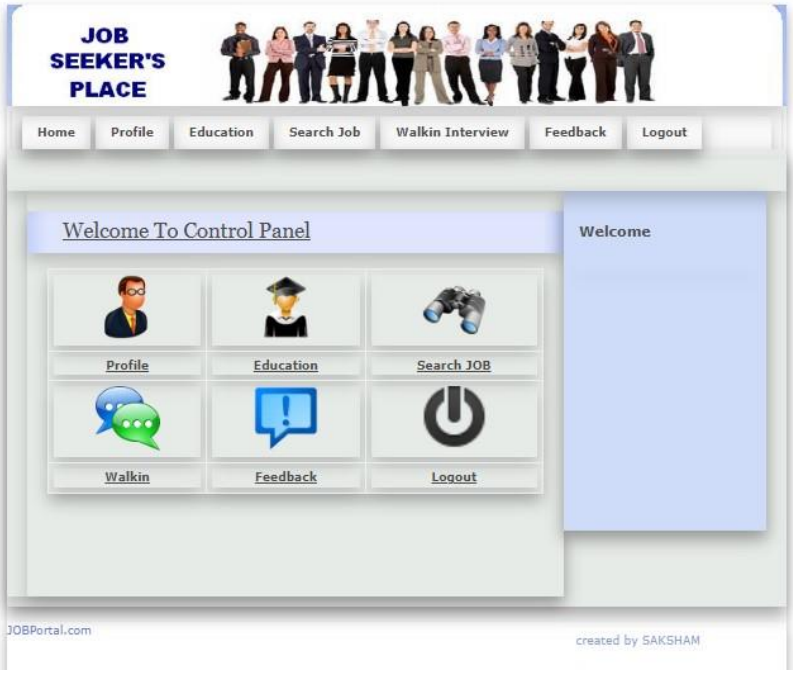
Job Seeker Registration Page

Latest News

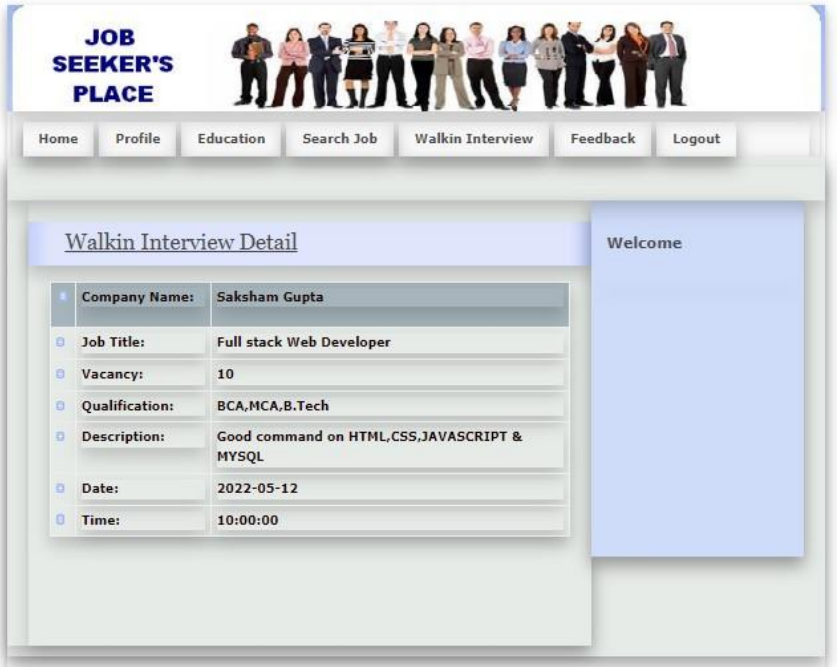


Contact Us Page

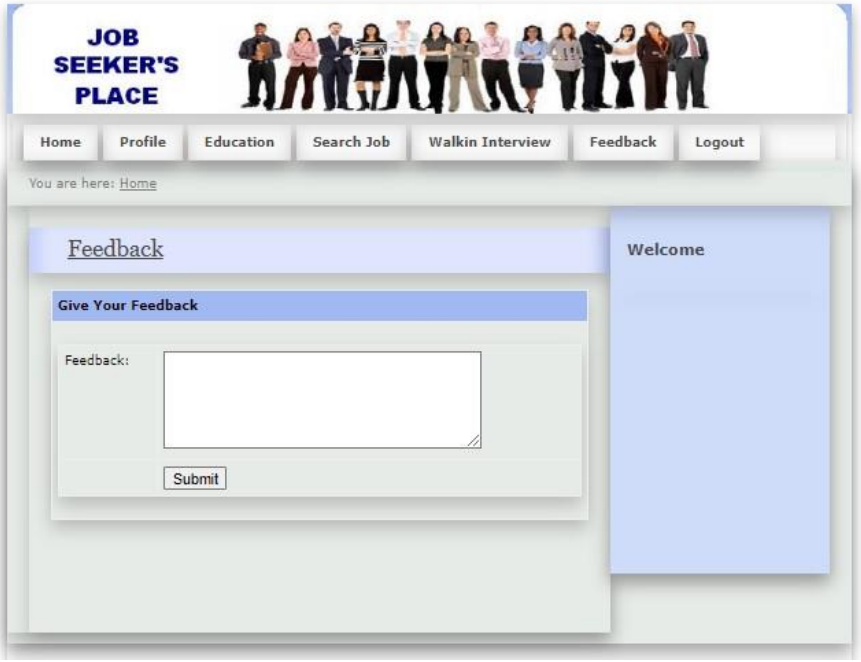
Job Seeker Place Page



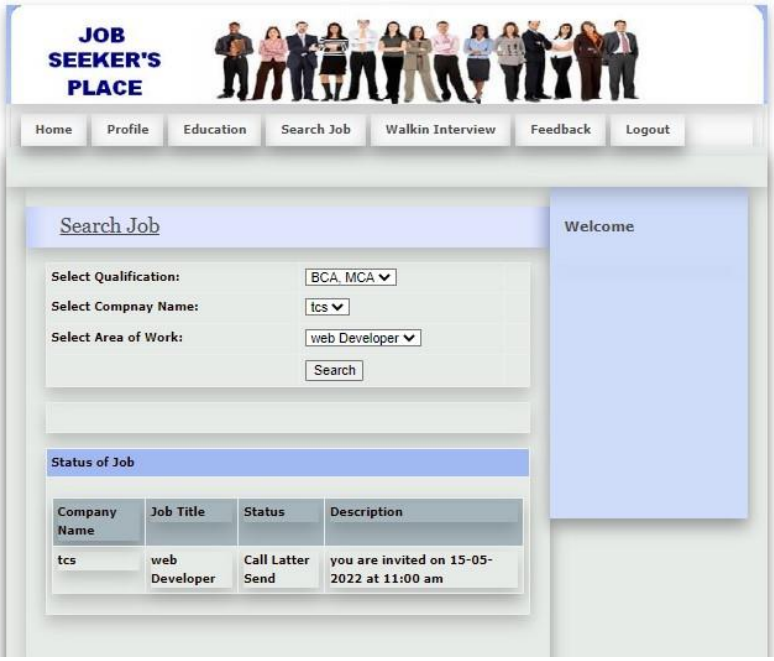
Walkin Interview Page



Feedback Page

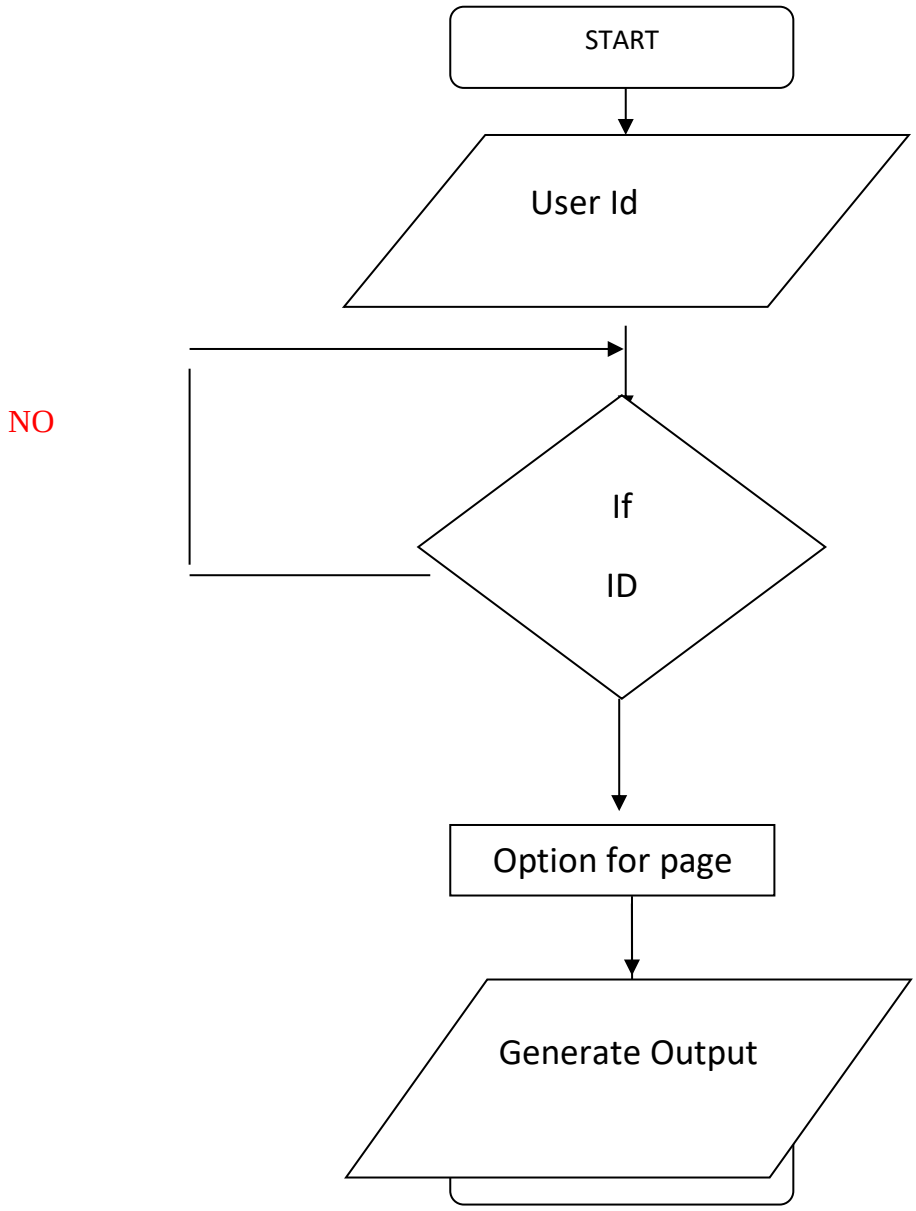


Search Job Page



Process Involved

LOGIN PROCESS:



TESTING METHODOLOGY

- Test plan has been created to guide the overall testing process.
- Modular testing has been used to test each module.
- Black box testing has been used to test the functionality of each module.
- PHP debugger has been used as a tool to black box test functional behavior of module.
- PHP debugger has also been used for white box testing by comparing step-by-step execution against white box cases.
- On-line testing of the software by live entering the details of five
- Different students.

TEST CASE-1

TEST NO. : 1

TEST TYPE : UNIT TESTING

INPUT : PASSWORD

OBJECTIVE : CHECKING PASSWORD SECURITY

EXPECTED OUTPUT: ACCESS TO AUTHORIZED USERS ONLY

ACTUAL OUTPUT : PASSWORD SECURITY SUCCESSFUL

RESULT : ACCESS TO ONLY AUTHORIZED USERS

T E S T C A S E -2

TEST NO. : 1

TEST TYPE : UNIT TESTING

INPUT : TELEPHONE NO., MOBILE NO, ETC.

OBJECTIVE : ALPHABETS IN NUMERIC VALUES

EXPECTED OUTPUT : ERROR MESSAGE

ACTUAL OUTPUT : ERROR MESSAGE

RESULT : ONLY INTEGERS ARE ACCEPTED FOR THE CONTACT NUMBERS

Code

Index.php

```
<?php
session_start();
if(isset($_SESSION['$UserName'])){
    header('location:Admin/index.php');
}
if(isset($_SESSION['$UserName_job'])){
    header('location:JobSeeker/index.php');
}
if(isset($_SESSION['$UserName_emp'])){
    header('location:Employer/index.php');
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>

    <title>JOB PORTAL BY SAKSHAM</title>

    <link rel="index" href="./" title="Home" />
    <link rel="stylesheet" media="screen,projection" type="text/css"
href="./css/main.css" />
    <link rel="stylesheet" media="print" type="text/css" href="./css/print.css"
/>
    <link rel="stylesheet" media="aural" type="text/css" href="./css/aural.css"
/>
    <style type="text/css">
<!--
```

```
.style1 {
    color: #000066;
    font-weight: bold;
}
.style2 {
    font-size: medium;
    font-weight: bold;
}
-->
    </style>
</head>

<body id="www-url-cz">
<!-- Main -->
<div id="main" class="box">
<?php
include "Header.php"
?>
<?php
include "menu.php"
?>
<!-- Page (2 columns) -->
    <div id="page" class="box">
    <div id="page-in" class="box">

        <div id="strip" class="box noprint">

            <!-- RSS feeds -->
            <hr class="noscreen" />

            <!-- Breadcrumbs -->
            <p id="breadcrumbs"> </p>
          <hr class="noscreen" />

        </div> <!-- /strip -->

        <!-- Content -->
        <div id="content">


            <!-- /article -->

            <hr class="noscreen" />
```

```
            <!-- /article -->

            <hr class="noscreen" />

            <!-- Article -->

            <!-- /article -->

            <hr class="noscreen" />

            <!-- Article -->
            <div class="article">
                <h2><span><a href="#">Welcome To Job Portal System by
SAKSHAM</a></span></h2>
                <h3><marquee>Welcome To Job Portal System by
SAKSHAM</marquee></h3>


                <p> <span class="style2">W</span>elcome to online Job Portal. It
provides facility to the Job Seeker to search for various jobs as per his
qualification. Here Job Seeker can registered himself on the web portal and
create his profile along with his educational information. Job Seeker can search
various jobs and apply for the Job.</p>
              <p>This Portal is also designed for the various employer who
required to recruit employees in their organization. Employer can registered
himself on the web portal and then he can upload information of various job
vacancies in their organization. Employeer can view the applications of Job
Seeker and send call latter to the job seekers.</p>
              <p align="right"> <img src="design/banner-4.jpg" alt="" width="510"
height="300" /></p>
              <p class="btn-more box noprint"> </p>
          </div> <!-- /article -->

            <hr class="noscreen" />

        </div> <!-- /content -->

<?php
include "right.php"
?>

    </div> <!-- /page-in -->
    </div> <!-- /page -->
```

```
<?php
include "footer.php"
?>
</div> <!-- /main -->

</body>
</html>
```

Jobseeker.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="content-language" content="cs" />
    <meta name="robots" content="all,follow" />


    <title>JOB PORTAL BY SAKSHAM</title>
    <meta name="description" content="..." />
    <meta name="keywords" content="..." />

    <link rel="index" href="./" title="Home" />
    <link rel="stylesheet" media="screen,projection" type="text/css"
href="./css/main.css" />
    <link rel="stylesheet" media="print" type="text/css" href="./css/print.css"
/>
    <link rel="stylesheet" media="aural" type="text/css" href="./css/aural.css"
/>
    <style type="text/css">
<!--
.style1 {
    color: #000066;
    font-weight: bold;
}
.style2 {font-weight: bold}
-->
    </style>
</head>

<body id="www-url-cz">
```

```
<SCRIPT language="JavaScript1.2" src="gen_validation.js"></SCRIPT>
<SCRIPT language="JavaScript1.2">
var arrFormValidation=
            [
                   [//Name
                         ["minlen=1",
        "Please Enter Name "
                         ]

                     ],
                   [//Address
                          ["minlen=1",
        "Please Enter Address "
                         ]

                   ],
                   [//Country




                   ],
                   [//State


                   ],
                   [//City


                   ],
                   [//Mobile

                         ["num",
        "Please Enter valid Mobile "
                         ],
                         ["minlen=10",
        "Please Enter valid Mobile "
                         ]



                   ],
                   [//Email
                          ["minlen=1",
        "Please Enter Email "
```

```
                        ],
                        ["email",
        "Please Enter valid email "
                        ]

                    ],
                    [//ID


                    ],
                    [//TDType


// Specify the query to execute
$sql = "select * from jobseeker_reg where Status='Confirm'";
// Execute query



$result = mysqli_query($con,$sql);


//var_dump($result);

// Loop through each records
while($row = mysqli_fetch_array($result))
{
$Name=$row['JobSeekerName'];
$City=$row['City'];
$Email=$row['Email'];

?>
<tr>
<td class="style3"><div align="left" class="style9 style5"><strong><?php echo
$Name;?></strong></div></td>
<td class="style3"><div align="left" class="style9 style5"><strong><?php echo
$City;?></strong></div></td>
<td class="style3"><div align="left" class="style9 style5"><strong><?php echo
$Email;?></strong></div></td>
</tr>
<?php
}
// Retrieve Number of records returned
$records = mysqli_num_rows($result);
?>
```

```
<?php
// Close the connection
mysqli_close($con);
?>
</table>
    </td>
  </tr>
</table>
                </p>

                <div align="center"><a href="JobSeekerReg.php"><strong>New Job
Seeker? Register Here</strong></a>                    </div>
                <p class="btn-more box noprint"> </p>
          </div> <!-- /article -->

            <hr class="noscreen" />

        </div> <!-- /content -->

<?php
include "right.php"
?>

    </div> <!-- /page-in -->
    </div> <!-- /page -->


<?php
include "footer.php"
?>
</div> <!-- /main -->

</body>
</html>
```

EmployeInsert.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
<?php

    $CompnayName=$_POST['txtName'];
    $ContactPerson=$_POST['txtPerson'];
    $Address=$_POST['txtAddress'];
    $City=$_POST['txtCity'];
    $Email=$_POST['txtEmail'];
    $Mobile=$_POST['txtMobile'];
    $Area=$_POST['txtAreaWork'];
    $Status="Pending";
    $UserName=$_POST['txtUserName'];
    $Password=$_POST['txtPassword'];
    $UserType="Employer";
    $Question=$_POST['cmbQue'];
    $Answer=$_POST['txtAnswer'];
    // Establish Connection with MYSQL
    $con = mysqli_connect ("localhost","root","","job");

    // Specify the query to Insert Record
    $sql = "insert into
employer_reg(CompanyName,ContactPerson,Address,City,Email,Mobile,Area_Work,Status
,UserName,Password,Question,Answer)
values('".$CompnayName."','".$ContactPerson."','".$Address."','".$City."','".$Ema
il."',".$Mobile.",'".$Area."','".$Status."','".$UserName."','".$Password."','".$Q
uestion."','".$Answer."')";
    // execute query

    mysqli_query ($con,$sql);
    // Close The Connection
    mysqli_close ($con);

    echo '<script type="text/javascript">alert("Registration Completed
Succesfully");window.location=\'index.php\';</script>';

?>
</body>
</html>
```

JobSeekerInsert.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
<?php

    $Name=$_POST['txtName'];
    $Address=$_POST['txtAddress'];
    $City=$_POST['txtCity'];
    $Email=$_POST['txtEmail'];
    $Mobile=$_POST['txtMobile'];
    $Qualification=$_POST['txtQualification'];
    $Gender=$_POST['cmbGender'];
    $BirthDate=$_POST['txtBirthDate'];
    $path1 = $_FILES["txtFile"]["name"];
    $Status="Pending";
    $UserName=$_POST['txtUserName'];
    $Password=$_POST['txtPassword'];
    $Question=$_POST['cmbQue'];
    $Answer=$_POST['txtAnswer'];
    $UserType="JobSeeker";
    if ($Qualification=="Other")
    {
        $Qualification=$_POST['txtOther'];
    }
    //   move_uploaded_file($_FILES["txtFile"]["tmp_name"],"upload/"  .$_FILES["t
xtFile"]["name"]);
    // Establish Connection with MYSQL
    $con = mysqli_connect ("localhost","root","","job");
    // Select Database  mysql_select_db("job", $con);
    // Specify the query to Insert Record
//  $sql = "insert into
jobSeeker_reg(JobSeekerName,Address,City,Email,Mobile,Qualification,Gender,BirthD
ate,Resume,Status,UserName,Password,Question,Answer) values(
//'".$Name."','".$Address."','".$City."','".$Email."',".$Mobile.",'".$Qualificati
on."','".$Gender."',
//'".$BirthDate."','".$path1."','".$Status."','".$UserName."','".$Password."','".
$Question."','".$Answer."')";
```

```
$sql="insert into
jobseeker_reg(JobSeekerName,Address,City,Email,Mobile,Qualification,Gender,BirthD
ate,Status,UserName,Password,Question,Answer) VALUES (
'$Name','$Address','$City','$Email','$Mobile','$Qualification','$Gender','$BirthD
ate','$Status','$UserName','$Password','$Question','$Answer'

)";
    // execute query

// var_dump($sql);
    if(mysqli_query ($con,$sql)){



        echo '<script type="text/javascript">alert("Registration Completed
Succesfully");window.location=\'index.php\';</script>';
    }


mysqli_close ($con);

    // Close The Connection


?>
</body>
</html>
```

JobSeekerReg.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="content-language" content="cs" />
    <meta name="robots" content="all,follow" />


    <title>Welcome To Job Portal by SAKSHAM</title>
    <meta name="description" content="..." />
    <meta name="keywords" content="..." />

    <link rel="index" href="./" title="Home" />
```

```
    <link rel="stylesheet" media="screen,projection" type="text/css"
href="./css/main.css" />
    <link rel="stylesheet" media="print" type="text/css" href="./css/print.css"
/>
    <link rel="stylesheet" media="aural" type="text/css" href="./css/aural.css"
/>
    <style type="text/css">
<!--
.style1 {
    color: #000066;
    font-weight: bold;
}
-->
    </style>
    <style type="text/css">

.ds_box {
    background-color: #FFF;
    border: 1px solid #000;
    position: absolute;
    z-index: 32767;
}

.ds_tbl {
    background-color: #FFF;
}

.ds_head {
    background-color: #333;
    color: #FFF;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
    font-weight: bold;
    text-align: center;
    letter-spacing: 2px;
}

.ds_subhead {
    background-color: #CCC;
    color: #000;
    font-size: 12px;
    font-weight: bold;
    text-align: center;
    font-family: Arial, Helvetica, sans-serif;
    width: 32px;
```

```
}

.ds_cell {
    background-color: #EEE;
    color: #000;
    font-size: 13px;
    text-align: center;
    font-family: Arial, Helvetica, sans-serif;
    padding: 5px;
    cursor: pointer;
}
    // Hide the calendar.
    ds_hi();
    // Set the value of it, if we can.
    if (typeof(ds_element.value) != 'undefined') {
        ds_element.value = ds_format_date(d, m, y);
    // Maybe we want to set the HTML in it.
    } else if (typeof(ds_element.innerHTML) != 'undefined') {
        ds_element.innerHTML = ds_format_date(d, m, y);
    // I don't know how should we display it, just alert it to user.
    } else {
        alert (ds_format_date(d, m, y));
    }
}

// And here is the end.

// ]]> -->
</script>
    <script src="SpryAssets/SpryValidationTextarea.js"
type="text/javascript"></script>
    <link href="SpryAssets/SpryValidationTextarea.css" rel="stylesheet"
type="text/css" />

</head>

<body id="www-url-cz">
<SCRIPT language="JavaScript1.2" src="gen_validation.js"></SCRIPT>
<SCRIPT language="JavaScript1.2">
var arrFormValidation=
            [
                  [//Name
                      ["minlen=1",
        "Please Enter Name"
                       ]
```

```
            ],
          [//Address
                ["minlen=1",
"Please Enter Address"
                ]

          ],
          [//City
               ["minlen=1",
"Please Enter City"
                ]
          ],
          [//Email

                ["minlen=1",
"Please Enter Email "
                ],
                ["email",
"Please Enter valid email "
                ]
          ],
          [//Mobile
                 ["num",
"Please Enter valid Mobile "
                ],
                ["minlen=10",
"Please Enter valid Mobile "
                ],
                ["maxlen=10",
"Please Enter valid Mobile "
                ]
          ],
          [//Qual




          ],
          [//Other


          ],
          [//Gender
```

```
                ],
                [//Birthdate

                     ["minlen=1",
          "Please Enter Birthdate "
                     ]

                ],
                [//Upload

                  ["minlen=1",
          "Please Upload Marksheet "
                     ]
                ],
                [//User
                     ["minlen=1",
          "Please Enter UserName "
                     ]


                ],
                [//Password
                    ["minlen=1",
          "Please Enter Password "
                     ]


                ],
                 [//Que


                ],
                 [//Answer

                     ["minlen=1",
          "Please Enter Answer "
                     ]

                ]

            ];
</SCRIPT>
<!-- Main -->
<div id="main" class="box">
```

```
<?php
include "Header.php"
?>
<?php
include "menu.php"
?>
<!-- Page (2 columns) -->
    <div id="page" class="box">
    <div id="page-in" class="box">

        <div id="strip" class="box noprint">

            <!-- RSS feeds -->
            <hr class="noscreen" />

            <!-- Breadcrumbs -->
            <p id="breadcrumbs"> </p>
          <hr class="noscreen" />

        </div> <!-- /strip -->

        <!-- Content -->
        <div id="content">


            <!-- /article -->

            <hr class="noscreen" />


            <!-- /article -->

            <hr class="noscreen" />

            <!-- Article -->

            <!-- /article -->

            <hr class="noscreen" />

            <!-- Article -->
            <div class="article">
                <h2><span><a href="#">Job Seeker Registration
Form</a></span></h2>
```

```
                    <div class="login">

                <form action="JobSeekerInsert.php" method="post" onSubmit="return
validateForm(this,arrFormValidation);" enctype="multipart/form-data" id="form2">
                  <table width="100%" border="0" cellspacing="0" cellpadding="0">
                    <tr>
                      <td>JobSeeker Name:</td>
                      <td><span id="sprytextfield3">
                        <label>
                        <input type="text" name="txtName" id="txtName" />
                        </label>
                      <span class="textfieldRequiredMsg">Enter
Name</span></span></td>
                    </tr>

                    <tr>
                      <td>Address:</td>
                      <td><span id="sprytextarea1">
                        <label>
                        <textarea name="txtAddress" id="txtAddress" cols="45"
rows="5"></textarea>
                        </label>
                      <span class="textareaRequiredMsg">Enter
Address</span></span></td>
                    </tr>
                    <tr>
                      <td>City:</td>
                      <td><span id="sprytextfield4">
                        <label>
                        <input type="text" name="txtCity" id="txtCity" />
                        </label>
                      <span class="textfieldRequiredMsg">Enter
City</span></span></td>
                    </tr>
                    <tr>
                      <td>Email:</td>
                      <td><span id="sprytextfield5">
                        <label>
                        <input type="text" name="txtEmail" id="txtEmail" />
                        </label>
                      <span class="textfieldRequiredMsg">Enter Email
Id</span></span></td>
                    </tr>
                    <tr>
```

```
                    <td>Mobile:</td>
                    <td><span id="sprytextfield6">
                      <label>
                      <input type="text" name="txtMobile" id="txtMobile" />
                      </label>
                    <span class="textfieldRequiredMsg">Enter
Mobile</span></span></td>
                  </tr>
                  <tr>
                    <td>Qualification:</td>
                    <td><label>
                  LOGIN.PHP

session_start();
$UserName=$_POST['txtUser'];
$Password=$_POST['txtPass'];
$UserType=$_POST['cmbUser'];
if($UserType=="Administrator")
{
$con = mysqli_connect("localhost","root","","job");

$sql = "select * from user_master where UserName='".$UserName."' and
Password='".$Password."'";
$result = mysqli_query($con,$sql);
$records = mysqli_num_rows($result);
$row = mysqli_fetch_array($result);
if ($records==0)
{
echo '<script type="text/javascript">alert("Wrong UserName or
Password");window.location=\'index.php\';</script>';
}
else
{
    $_SESSION['$UserName']=$UserName;
header("location:Admin/index.php");
}
mysqli_close($con);
}
else if($UserType=="JobSeeker")
{
$con = mysqli_connect("localhost","root","","job");
$sql = "select * from jobseeker_reg where UserName='".$UserName."' and
Password='".$Password."' and Status='Confirm'";
$result = mysqli_query($con,$sql);
$records = mysqli_num_rows($result);
```

```
$row = mysqli_fetch_array($result);
if ($records==0)
{
echo '<script type="text/javascript">alert("Wrong UserName or
Password");window.location=\'index.php\';</script>';
}
else
{
$_SESSION['ID']=$row['JobSeekId'];
$_SESSION['Name']=$row['JobSeekerName'];
$_SESSION['$UserName_job']=$UserName;
header("location:JobSeeker/index.php");
}
mysqli_close($con);
}
else
{
$con = mysqli_connect("localhost","root","","job");
$sql = "select * from employer_reg where UserName='".$UserName."' and
Password='".$Password."' and Status='Confirm'";
$result = mysqli_query($con,$sql);
$records = mysqli_num_rows($result);
$row = mysqli_fetch_array($result);
if ($records==0)
{
echo '<script type="text/javascript">alert("Wrong UserName or
Password");window.location=\'index.php\';</script>';
}
else
{
    $_SESSION['ID']=$row['EmployerId'];
$_SESSION['Name']=$row['CompanyName'];
$_SESSION['$UserName_emp']=$UserName;
header("location:Employer/index.php");
}
mysqli_close($con);
}
?>

</body>
</html>
```

News.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="content-language" content="cs" />
    <meta name="robots" content="all,follow" />


    <title>JOB PORTAL</title>
    <meta name="description" content="..." />
    <meta name="keywords" content="..." />

    <link rel="index" href="./" title="Home" />
    <link rel="stylesheet" media="screen,projection" type="text/css"
href="./css/main.css" />
    <link rel="stylesheet" media="print" type="text/css" href="./css/print.css"
/>
    <link rel="stylesheet" media="aural" type="text/css" href="./css/aural.css"
/>
    <style type="text/css">
<!--
.style1 {
  color: #000066;
  font-weight: bold;
}
.style2 {font-weight: bold}
-->
    </style>
</head>

<body id="www-url-cz">
<!-- Main -->
<div id="main" class="box">
<?php
include "Header.php"
?>
<?php
include "menu.php"
?>
<!-- Page (2 columns) -->
    <div id="page" class="box">
    <div id="page-in" class="box">

        <div id="strip" class="box noprint">
```

```
            <!-- RSS feeds -->
            <hr class="noscreen" />

            <!-- Breadcrumbs -->
            <p id="breadcrumbs"> </p>
          <hr class="noscreen" />

        </div> <!-- /strip -->

        <!-- Content -->
        <div id="content">


<?php
include "right.php"
?>

    </div> <!-- /page-in -->
    </div> <!-- /page -->


<?php
include "footer.php"
?>
</div> <!-- /main -->

</body>
</html>
```

Limitation

LIMITATION

- This system can’t provide on line video conferencing..
- This system not providing chat facility.
- This system not provides the facility to users so that they can transfer various types of information to their friends.
- This system not making on in different languages.

Bibliography

- The Complete Reference MySQL, Vikram Vaswani.
- Complete Reference HTML, S.Eric
- jQuery Cook Book, Tim Hudson.
- www.php.net
- www.w3schools.com