**1.INTRODUCTION**

* 1. **INTRODUCTION:**

***INPLIMENTATION***

This project is aimed at developing a web application for the Army Welfare Department. The system is an web application that can be accessed throughout the Army Welfare Department with proper login provided. This system can be used as an web application for post the jobs and to manage the bio data of army persons. Army persons logging should be able to upload their bio data. The key feature of this project is that it is a onetime registration. Our project provides the facility of maintaining the details of the army persons. It also provides a requested list of candidates to recruit the students based on given query. Administrator logging in may also search any information put up by the army persons. This project will aid Army Welfare Department.

* 1. **PROBLEM DEFINITION:**

This project is a web application for the Army Welfare Department.. It is used to post the all kind of jobs for Ex-army person and to manage the profile of army persons. Army persons login should be able to upload their profile. It is used to post a notice for user.

**2. SYSTEM NALYSIS**

**2.1. ANALYSIS:**

The prime focus of the feasibility study is evaluating the practicality of the proposed database keeping in the mind a number of factors. The following factors are taken into account before deciding in favor of the new system. The system study phase studies the problem, identifies alternate solutions evaluate those solutions and finally recommends the best solution. The system study gives the structure and function of the system.

**2.2. EXISTING SYSTEM:**

In this type this may new one web service portal for Karnataka army welfare department. This existing system is based on army well fare, this project not yet implemented for army based pension and after retirement job offers information’s. This is maintained by army welfare department in Dharwad, as they say its need to be made available to online to register for updates and pension and new government schemes and news. Existing system was completely based on paper work based and still it needs to be updated.

**Limitations of Existing System:**

* Paper based information collection by army.
* Job information’s will be posted on only office notice boards.
* Government offices need to post army job needs in paper only.
* Army welfare department needs to be made online so that all works get done faster because work process here slow.
* Government jobs and schemes will only be updated in newspapers by paid ads.

**2.3. PROPOSED SYSTEM:**

Using this web application we can help track of army retired peoples information and their current status. This web application is will be helpful for admin or head offer at army welfare department in Dharwad. Basically they collect army info through papers , jobs and news updates posted on notice boards will be avoided by using this applications. It is used to post the jobs and their requirements to Ex-army people. It provide the army employment news for user. There type users may get benefits out of it.

* **Army Welfare Department Officers :**

Government Officer provides the information about jobs and their requirements for Ex-army persons. Registration is compulsory for Government Office. Government Officer will also update their information and profile. They can send feedback.

* **Ex-Army Seeker :**

Ex-army Seeker can view the all information about the jobs and their requirement. Registration is compulsory for Ex-army Seeker. Ex-army Seeker can also update their information and profile. They can send feedback.

**Advantages of Proposed System:**

* Retired army persons can get all information about jobs easily.
* Reduced the paper work save the time.
* Government jobs and schemes will only be updated in side web portal.
* Army welfare department made online so that all works get done faster.

**3. SYSTEM DESIGN**

**3.1. DESIGNING:**

System design is the first step in moving from problems domain towards solution domain. The goal of the design is to produce a model or the representation of the system, which can be later used to build that system. At the first level of the design, the focus is on deciding which modules is needed for the system, the specification of these modules and how the modules should be interconnected this called system design or top level design.

We are making three modules for our system, they are

* **Admin :**

Admin is the high level administrative person who will be responsible to maintain data in the database. If need he can make query using SQL language. He can check the reports and make their final decision. He can update post the jobs and their requirements and updates. Even he can have direct contact with the officers and Ex-army people.

* **Officers:**

Army welfare department Officers can provide the information about jobs and their requirements for Ex-army persons. He can also view and delete the posted jobs. Registration is compulsory for Government Office. Government Office module will also update their information and profile.

* **Ex-Army:**

Ex-army Persons can view the all information about the jobs and their requirement. Registration is compulsory for Ex-army Person. Ex-army Person can also update their information and profile. They can send feedback. He can view the notices.

**3.2. DATA FLOW DIAGRAM (DFD):**

Data flow diagram is a simple pictorial representation or model for system behavior. It specifies, “What is to be done but not how is to be done”. It describes the logical structure of the system. It relates data or information to various processes of the system. It follows top down system.

**DATA FLOW DIAGRAM NOTATIONS:**

* **Functional Processing:**

It is represented by an oval. The processing or main transaction is specified By this notation.

* **Data Base:**

It is represented by one open end rectangle. The database used in the system is specified by this notation.

* **Data Flow:**

It is represented by an arrow line and name of the data is specified by the Side of the line as label. This is used for data movement.

* + **Source Or Sink:**

It is represented by one end triangle. It is used for specifying from where data comes and where it reaches.

**Steps Required In DFD:**

* Identify system, processing transformations. Transactions concerned with reading, validating and formatting inputs.
* Identify input transformations. Transformations concerned with reading, validating and formatting inputs.
* Identify output transformations. Transformations concerned with formatting and writing output.
* Function.

**3.3. DATA FLOW DIAGRAM LEVELS:**

**DFD Level 0:**

DHARWAD ARMY WELFARE DEPARTMENT

**DFD Level 1:**

* **DFD Level 1: Admin**

Seeker

Feedback

Notice

Vacancy

Company

Admin

* **DFD Level 1: Officer**

Notice

Vacancy

Company

Officer

* **DFD Level 1: Ex-army**

Seeker

Vacancy

Notice

Feedback

Ex-army

**3.4. ENTITY-RELATIONSHIP:**

An Entity-Relationship model is the result of using the systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data. The data is represented as a component (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterized them. Diagrams created to represent these entities, attributes and relationships graphically are called Entity-Relationship Diagrams.

An E-R model is typically implemented as a database. In the case of relation database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such as pointers are the physical implementation of the relationships.

**ENTITY-RELATIONSHIPS NOTATIONS:**

* **Entity:**

It is represented by one end rectangle. It is used for specifying a thing capable of an independent existence that can be uniquely identified.

* **Attribute:**

It is represented by oval. It is used for specifying the fields of entity.

* **Relation:**

It is represented by rhombus. It is used for specifying the relation between two entities.

* **Primary Key:**

It is represented by oval. It is used for specifying every entity must have minimal set of uniquely identifying attributes, which is called the entity’s primary key.

**ENTITY-RELATIONSHIP MODELING:**

* **Two related entities:**
* **An entity with an attribute:**

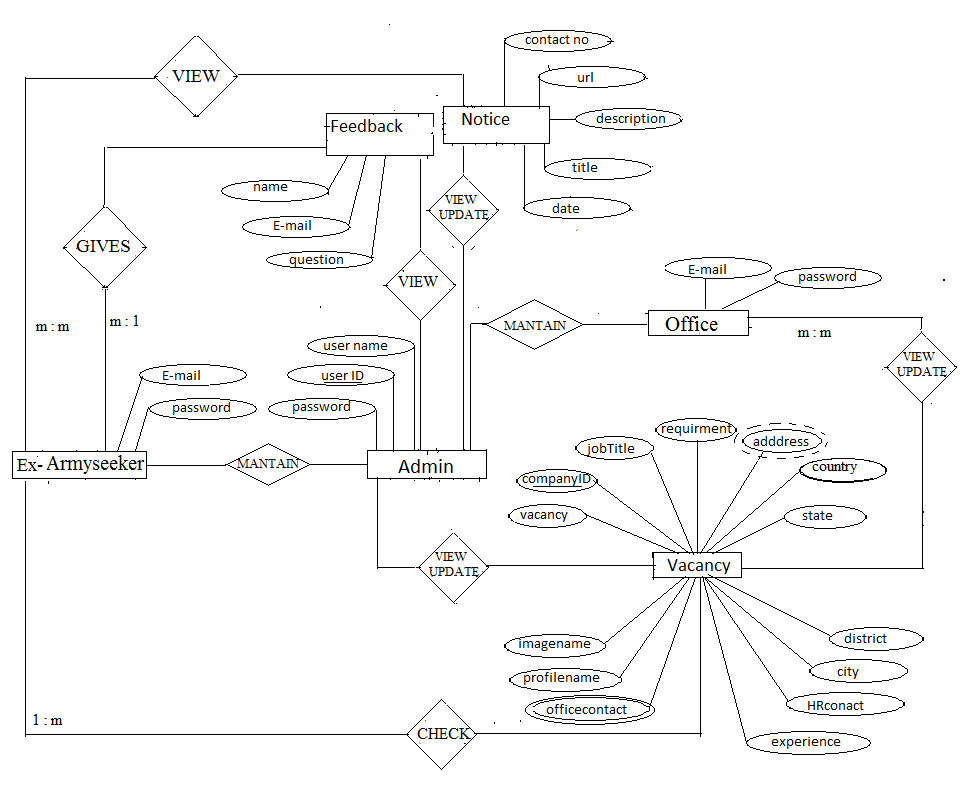
* **Cardinality ratio 1:N for E1:E2 in R :**

**E2**

**E1**

**R**

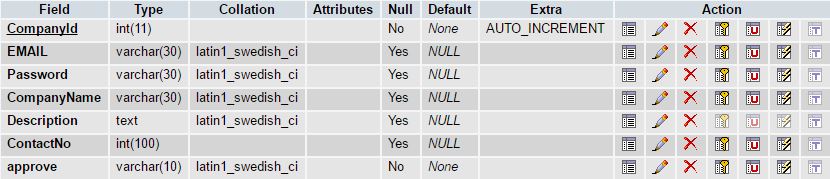
**ENTITY-RELATIONSHIP Diagram:**

****

**3.5. DATABASE DESIGN:**

To run the system smoothly we are designing the following database tables.

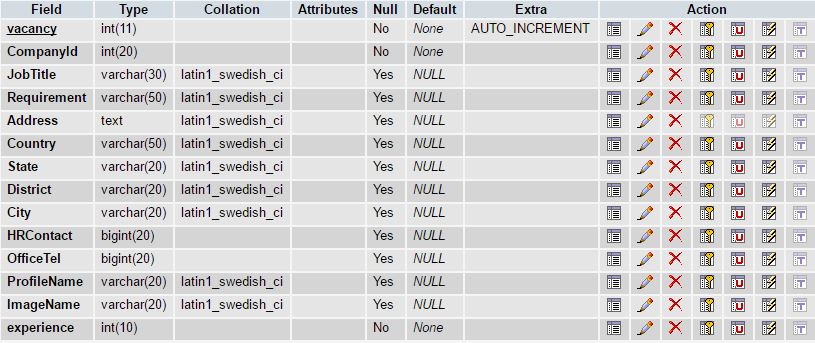
* **Table Company:**

****

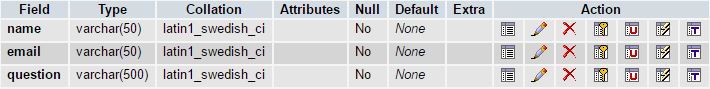
* **Table Seeker:**

****

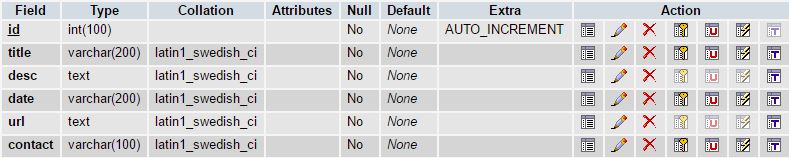
* **Table vacancy:**

****

* **Table Feedback:**

****

* **Table Notice:**

****

**4. SYSTEM REQUIREMENTS**

**4.1. HARDWARE AND SOFTWARE REQUIRMENTS:**

**HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| Processor | Intel Pentium or Greater |
| RAM Capacity | 512MB or Greater |
| Processor speed | 250MHz to 667MHz |
| Hard disk | 20GB or­­ Higher |

**SOFTWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| Operating System | Windows 98/2000/XP/NT |
| Font-End Tool | HTML, CSS |
| Back-End Tool | MySQL |
| Client side script | JavaScript |
| Web browser | Mozilla Firefox/ Google chrome/ Opera |
| Server side script | PHP |
| Language | PHP |

**4.2. FUNCTIONAL REQUIREMENTS:**

Functional requirements are capabilities that a system must exhibit in order to solve a problem. Functional requirements can be sub-classified in:

* **Data Requirements:**

Data Requirements also known as conceptual requirements, content requirements or storage requirements. These requirements establish how information is stored and administrated by the application.

* **Interface Requirements (To The User)** :

Interface Requirements also known as interaction requirements or user’s requirements. They give an answer to how the user is going to interact with the Web application.

* **Navigational Requirements:**

Navigational Requirements represent users’ navigation needs through the hyperspace.

* **Personalization Requirements:**

Personalization Requirementsalso known as customization or adaptation requirements. They describe how a Web application has to (dynamically) adapt itself, depending on the user or environment profile.

* **Transactional Requirements:**

Transactional Requirements also known as internal functional requirements or service requirements, express what the Web application has to compute internally, without considering interface and interaction aspects.

**4.3. NON-FUNCTIONAL REQUIREMENTS:**

Non-Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

**Performance:**

* Performance requirements define acceptable response times for system functionality.
* The load time for user interface screens shall take no longer than ten seconds.
* The log in information is stored.
* Queries shall return results within five seconds.

**Reliability:**

Specify the factors required to establish the required reliability of the software system at time of delivery.

### 

**5. IMPLIMENTATION**

**5.1. PHP AND MYSQL:**

* **PHP:**

**Hypertext Preprocessor (PHP)** lets you separate the dynamic part of your pages from the static HTML. We simply write the regular html in the normal manner, using whatever Web-page-building tools you normally use. We then enclose the code for the c parts in special tags, most of which start with "<?php" and end with "?>".

We normally give your file a .php extension, and typically install it in any ace you could place a normal Web page. Although what you write often looks more a regular html file than a servlet, behind the scenes, the PHP page just gets converted to a normal servlet, with the static html simply being printed to the output stream associated with the servlet’s service method.

This is normally done the first time the page is requested, and developers can simply request the page themselves when first installing it if they want to be sure that the first real user doesn't get a momentary delay when the PHP page is translated to a servlet and the servlet is compiled and loaded. Many Web servers let you define aliases that so that a URL that appears to reference an html file really points to a servlet or PHP page.

* **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. 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, and Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

Third-party proprietary and free graphical administration applications (or "front ends") are available that integrate with MySQL and enable users to work with database structure and data visually.

**5.2. FRONTEND:**

**HTML:**

**Hypertext Markup Language (HTML)** is a method of describing the format of documents which allows them to be viewed on computer screens. HTML documents are displayed by web browsers, programs which can navigate across networks and display a wide variety of types of information. HTML pages can be developed to be simple text or to be complex multimedia extravaganzas containing sound, moving images, virtual reality, and Java applets.

The global publishing format of the Internet is HTML. It allows authors to use not only text but also format that text with headings, lists, and tables, and to include still images, video, and sound within text. Readers can access pages of information from anywhere in the world at the click of a mouse-button. Information can be downloaded to the reader’s own PC or workstation. HTML pages can also be used for entering data and as the front-end for commercial transactions.

**Features of HTML:**

* It is not a programming language.
* It is not a data description language.
* It is simple to understand and implement.
* HTML constructs a very easy to comprehend, and can be used effectively by anybody.
* The methodology used by HTML to mark up information is independent of its representation on a particular hardware or software architecture.
* HTML syntax is a worldwide standard.

**PHP:**

**Hypertext Preprocessor (PHP)** lets you separate the dynamic part of your pages from the static HTML. We simply write the regular html in the normal manner, using whatever Web-page-building tools you normally use. We then enclose the code for the c parts in special tags, most of which start with "<? php" and end with "?>".

We normally give your file a .php extension, and typically install it in any ace you could place a normal Web page. Although what you write often looks more a regular html file than a servlet, behind the scenes, the PHP page just gets converted to a normal servlet, with the static html simply being printed to the output stream associated with the servlet’s service method.

This is normally done the first time the page is requested, and developers can simply request the page themselves when first installing it if they want to be sure that the first real user doesn't get a momentary delay when the PHP page is translated to a servlet and the servlet is compiled and loaded. Many Web servers let you define aliases that so that a URL that appears to reference an html file really points to a servlet or PHP page.

**Portability:**

**PHP** is an open-source server side scripting language designed for Web development to produce dynamic web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It has also evolved to include a command line interface capability and can be used in standalone graphical applications.

PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. PHP was a competitor to Microsoft's Active Server Pages (ASP) server-side script engine and similar languages.

**Composition:**

It was mentioned earlier that the PHP architecture could include reusable PHP components. The architecture also allows for the embedding of a scripting language directly into the PHP file. The components current supported include sessions and Serves. As the default scripting language, PHP Pages use the C Programming language,JavaScript,HTML. This means that scripting on the server side can take advantage of the full set of capabilities that the C programming language offers.

**Processing:**

A PHP file is essentially an HTML document with PHP scripting or tags. It may have associated components in the form of .class or it may not. The use of components is not required.

The PHP file has a .php extension to identify it to the server as a Hypertext Preprocessor Pages file. Before the page is served, the PHP syntax is parsed and processed into a servlet on the server side. The servlet that is generated, outputs real content in straight HTML for responding to the customer. Because it is standard HTML, the dynamically generated response looks no different to the customer browser than a static response.

**PHP Scripting Elements:**

PHP scripting elements let you insert PHP code into the servlet that will be generated from the current PHP page. There are three forms:

1. Expressions of the form <? php= expression?> that are evaluated and inserted into the input,

2. Script lets of the form <? php code?> that are inserted into the servlet's service method.

**Access Models:**

PHP file may be accessed in at least two different ways:

* A client request comes directly into a PHP page. In this scenario, suppose the page accessed reusable "include.php" components that perform particular well-defined computations like accessing a database. The page uses such Beans to generate dynamic content and present it back to the client.
* A request comes through a servlet. The servlet generates the dynamic content. To handle the response to the client, the servlet creates a file and stores the dynamic content (sometimes called the result set) the servlet then invokes a PHP.

PHP is best summarized as an embedded server-side web scripting language that provides the developers with the capability to quickly and efficiently build web applications. PHP bears a close resemblance, bothsynatactically and grammarly, 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 expressions, parsing, powerful array handling capabilities, an object oriented methodology and vast database support.

PHP can also serve as a valuable tool for creating and managing a dynamic content, embedded directly beside the likes of JavaScript, Style sheets, WML (wireless markup language) and many other useful languages. Providing a hundreds of predefined functions.

Extensive support is offered for graphic creation and manipulation, mathematical calculations, ecommerce, and burgeoning technologies such as extensive markup language (XML), open database connectivity (ODBC) and macromedia sock ware.

In this project we have implemented three-tier model, commands are sent to a "middle tier" of Services, which then send SQL statements to the database.

**Advantages of PHP:**

* PHP can generate dynamic page content
* PHP can create, open, read, write, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can restrict users to access some pages on your website
* PHP can encrypt data
* PHP runs on different platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.).
* PHP has support for a wide range of databases.

**How PHP Works?**

PHP pages exist in 3 forms or versions:

* PHP source code consists of text file with an extension of php and contains a mix of HTML template code, C language statements and Java script
* Directives and actions that describe how to generate a web page to service a particular request.
* PHP source code: the jsp container translates the jsp source code into the source code for an equivalent Java Servlet as needed..

**What Is PHP SCRIPT?**

* PHP Script is embedded into html.
* It is browser dependent.
* PHP script depends on the web browser to support it. If the browser doesn't support it.
* PHP Script code will be ignored. Internet Explorer 3.0 and Netscape Navigator 2.0 onwards support PHP script
* It is an interpreted language, loosely typed, object based language.

**Why PHP?**

* PHP runs on different platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP has support for a wide range of databases.
* PHP is free. Download it from the official PHP resource.
* PHP is easy to learn and runs efficiently on the server side

**Client Side Framework:**

The client side framework includes the following:

* Web Browser
* HTML client extension ( active x controls and Netscape plugging)
* Scripts language (PHP Script) PHP Script role in web application development.

**Client Side Application:**

PHP Script has good capabilities when working with HTML tags & java script. For certain cases JavaScript provides a programming backbone with which to develop application.

**Data Validation:**

PHP provides the means for basic data validation before it is sent to the server. Whether the values entered are correct or not or whether all the fields in a form are filled out or not can be checked before sending data to web server, if PHP is not used then data is sent to web server, and the web server would response with a message that the data sent to it is incorrect or incomplete. Thus, PHP ensures data validation and also reduces the network traffic.

**PHP Data Base Connectivity:**

In an enterprise computing which is largely the black art of managing huge databases? People associated with the enterprise need to be able to use and update the data easily, quickly and securely.

PHP Data Base Connectivity is a standard SQL database access interface providing uniform access to a wide range of relational databases. It also provides a common base on which higher level tools and interfaces can be built.

<?php $var=$\_POST[""]

$con=mysql\_connect("localhost","root",""); mysql\_select\_db("guru",$con); $sql="insert into tablename values()";

mysql\_query($sql); ?>

Mysql\_select\_db does the following things:

* Establish a connection with a database
* Send SQL statements
* Process the results.

**Connection:**

A connection object represents a connection with a database. A connection session includes the SQL statements that are executed and the results that are returned over the connection. A single application can have one or more connections with a single database, or it can have connections with many different databases.

**Sending Statement:**

Once a connection is established, it is used to pass SQL statements to its underlying database. mysql\_query() sends the SQL statements. This provides a great deal of flexibility, allowing the use of database-specific statements or even Non-SQL statements. It requires, however, that the user be responsible for making sure that the underlying database can process the SQL statements being sent and suffer the consequences if it cannot.

**Session:**

This is the ***Http Session*** object associated with the request. Recall that sessions are created automatically, so this variable is bound even if there was no incoming session reference. The one exception is if you use the ***session*** attribute of the page directive to turn sessions off, in which case attempts to reference the session variable cause errors at the time the JSP page is translated into a servlet.

**5.3. BACKEND:**

**WAMP SERVER:**

**Windows Apache MySQL Pup (WAMP)**

**Access:**

Create a PHP Web Application or Mobile Application using Net Beans Build and run project, this will automatically launch the WAMP as default.

**Viewing Web Applications:**

http://localhost/

**Directory Structure:**

The typical and default directory hierarchy of a Tomcat installation comprises the following:

* Bin - startup, shutdown and other scripts and executable.
* Common - common classes that Catalina and web applications can use.
* Conf - XML files and related DTDs to configure Tomcat.
* Logs - Catalina and application logs.
* Server - classes used only by Catalina.
* Shared - classes shared by all web applications.
* Webapps - directory containing the web applications.
* Work – temporary storage for files directories.

A web application is basically a web site that:

* "Knows who you are"--it doesn't just give you static pages, it interacts with you.
* Can permanently change data (such as in a database).
* A web application can consist of multiple pieces.
* Static web pages (possibly containing forms).
* Servlets: PHP.

WAMP organizes all these parts into a single directory structure for each web application.

**The Flow That Takes Place Is:**

* The user submits an HTML form
* Wamp finds the servlet based on the URL and the employment. descriptor (web.xml) and passes the request to the servlet
* The servlet computes a response
* Either: The servlet writes an HTML page containing the response
* The servlet forwards the response to the JSP
* The PHP embeds the response in an HTML page
* Wamp returns the HTML page to the user.

**Status:**

Wamp is available at the php.net binary downloads page. The wamp server is a PHP based Web Application container that was created to run Servlets and PHP Pages in Web applications. As part of wamp open source, it has nearly become the industry accepted standard reference implementation for both the Servlets and PHP.

Wamp server includes the apache tomcat .Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation.

It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Tomcat is a web server that supports servlets and PHPs. Tomcat comes with the Jasper compiler that compiles PHPs into servlets. The Tomcat servlet engine is often used in combination with an Apache webserver or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high availability environments. Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

**OVERVIEW OF 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.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.

MySQL Workbench is available in two editions, the regular free and open source

*Community Edition* which may be downloaded from the MySQL website, and the proprietary

*Standard Edition* which extends and improves the feature set of the Community Edition.Third-party proprietary and free graphical administration applications (or "front ends") are available that integrate with MySQL and enable users to work with database structure and data visually. Some well-known front ends, in alphabetical order, are:

* **Adminer –** a free MySQL front end written in one PHP script, capable of managing multiple databases, with many CSS skins available.
* **DaDaBIK –** a customizable CRUD front-end to MySQL. W ritten in PHP. Commercial.
* **DBEdit –** a free front end for MySQL and other databases.
* **dbForge GUI Tools —** a set of tools for database man agement that includes separate applications for schema comparison and synchronization, data comparison and synchronization, and building queries.
* **Heidi SQL –** a full featured free front end that runs on Windows, and can connect to local or remote MySQL servers to manage databases, tables, column structure, and individual data records. Also supports specialised GUI features for date/time fields and enumerated multiple-value fields.
* **Libre Office Base -** LibreOffice Base allows the creation and management of databases, preparation of forms and reports that provide end users easy access to data. Like Access it.

Can be used as a front-end for various database systems, including Access databases (JET):

ODBC data sources, and MySQL or PostgreSQL.

* **Navicat –** a series of proprietary graphical databas e management applications, developed for Windows, Macintosh and Linux.
* **OpenOffice.org –** OpenOffice.org Base can manage MyS QL databases if the entire suite is installed. Free and open-source.
* **phpMyAdmin –** a free Web-based front end widely installed by web hosts, since it is developed in PHP and is included in the LAMP stack, MAMP, XAMPP and WAMP software bundle installers.
* **SQLBuddy** - a free Web-based front end, developed in PHP.
* **Sequel** **Pro** - a free, open-source front end for Mac OS X.
* **SQLYog** - a free community-developed UI for MySQL.

MySQL is written in C and C++. Its SQL parser is written in yacc and a home-brewed lexicalanalyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java.

In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

**Features:**

* Cross-platform support
* Updatable Views
* Transactions with the InnoDB, and Cluster storage engines; savepoints with InnoDB.

We can import and export the database

**Why MySQL?**

* Free as in Freedom - Released with GPL version 2 license (though a different license can be bought from Oracle, see below)
* Cost - Free!
* Support - Online tutorials, forums, mailing list (lists.mysql.com), paid support contracts.
* Speed - One of the fastest databases available.
* Functionality - supports most of ANSI SQL commands.
* Ease of use - less need of training / retraining.
* Portability - easily import / export from Excel and other databases
* Scalable - Useful for both small as well as large databases containing billions of records and terabytes of data in hundreds of thousands of tables.
* Permission Control - selectively grant or revoke permissions to users.

**Limitations:**

* Like other SQL databases, MySQL does not currently comply with the full SQL standard for some of the implemented functionality, including foreign key references when using some storage engines other than the 'standard' InnoDB.
* Triggers are currently limited to one per action / timing, i.e. maximum one after insert and one before insert on the same table. There are no triggers on views.
* MySQL, like most other transactional relational databases, is strongly limited by hard disk performance. This is especially true in terms of write latency.Given the recent appearance of very affordable consumer grade SATA interface Solid-state drives that offer zero mechanical latency, a fivefold speedup over even an eight drive RAID array can be had for a smaller investment.

**SELECT Statement:**

This statement is used to select certain attributes from a table.

Format: SELECT <ATTRIBUTES> FROM <TABLENAME> WHERE CONDITION;

**INSERT Statement:**

Enables the user to enter data directly into the table.

Format: INSERT INTO TABLENAME (ATTRIBUTE NAMES…)

VALUES (ATTRIBUTE VALUES);

**UPDATE Statement:**

Enables to modify the data already stored in the table.

Format: UPDATE TABLENAME

SET COLUMNNAME=VALUE

WHERE CONDITION

**DELETE Statement:**

Enables you to remove selected rows of data from a single table.

Format: DELETE FROM TABLENAME

WHERE CONDITION

**VIEW Statement:**

A view is a personalized presentation of data from one or more tables. View does not contain or store data, but they take data from the tables on which they are based, called base tables.

As the tables the views can also be queried, updated, inserted and Deleted, with some restrictions.

Format: CREATE VIEW *VIEWNAME* AS

SELECT *FIELDS*

FROM *TABLENAME*

WHERE *CONDITION*

**Indexes:**

An index is an optional structure associated to tables that increases the data recovery performance. An index is created for one or more columns of a table. After being created, an index is automatically updated and used by ORACLE.

Format: CREATE INDEX *INDEXNAME* ON TABLENAME

(NAMEOFTHECOLUMN ASC/DESC,..)

**Relational Database Management System:**

In an RDBMS, a database is considered to be a collection of interrelated data and programs. The data in a database has to be related. The programs in a database perform the role manipulating this data. A database management system is software that takes care of maintaining the database. It acts as the interface between the database and the user.

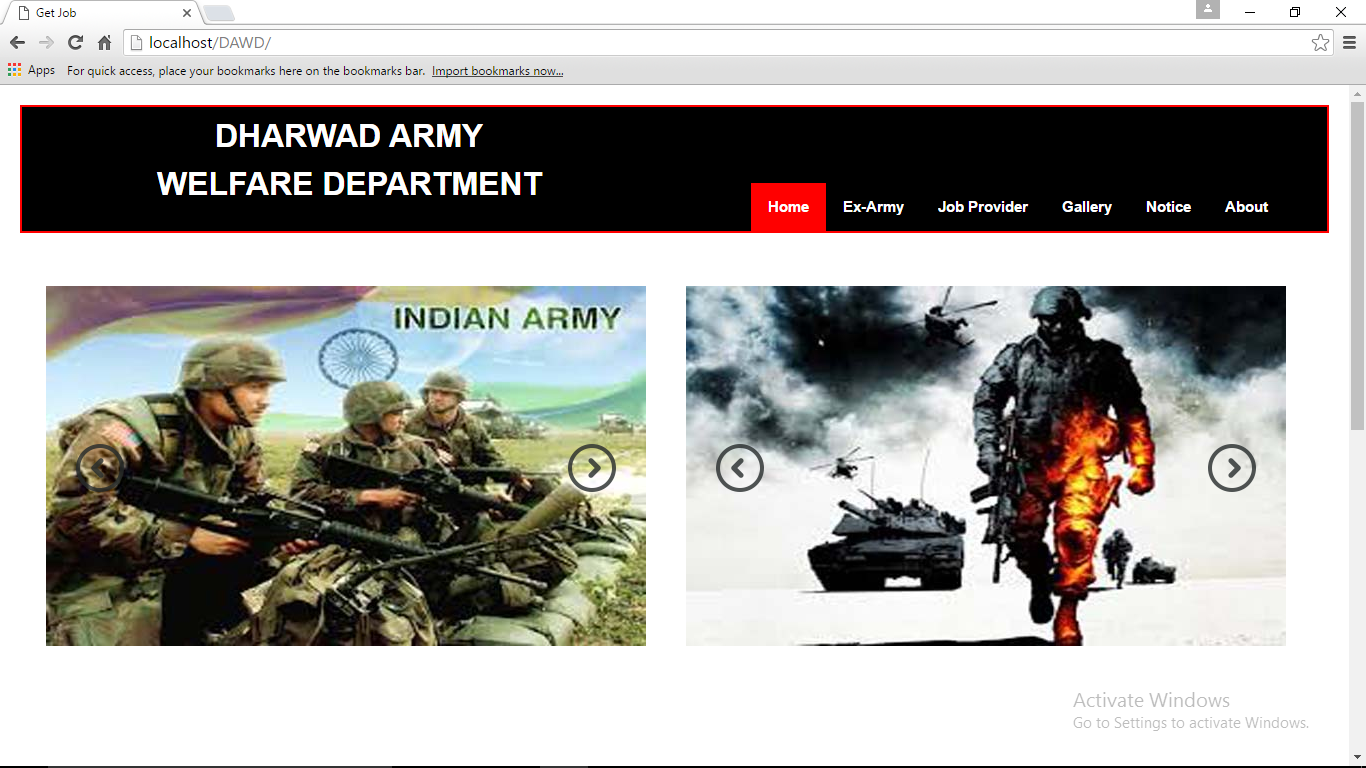
A database that is designed on relational model of database is called a ‘Relational Database’ and the software that helps maintain those databases is called ‘Relational database Management System’. In the relational data model, the data in a database is an organized in ‘relations’. A Relation is synonymous with a ‘table ’. A table consists of columns and rows, in which are referred to as fields and records in DBMS terms and attributes and tuples in RDBMS.

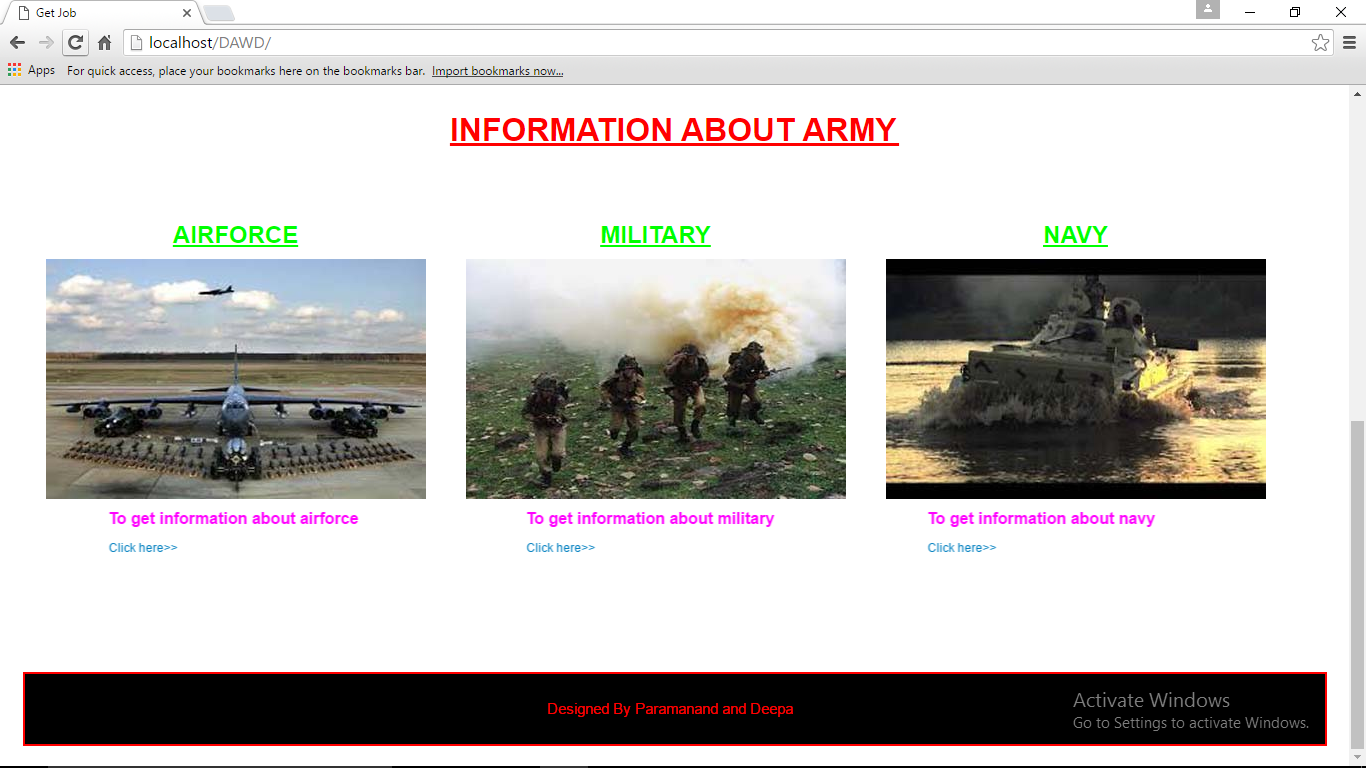
**Users of an RDBMS:**

* **Database Designers :** These would be the people who analyze the kind of data that is to be stored in the database, and would design the structure of the database.
* **Database Administrator :** More popularly called the DBA, this would be a person who monitors operations on a database and ensures that it is maintained efficiently.
* **Application Developer :** This category of people takes care of writing programs for accessing the database.
* **End User :** Entry of data and manipulation of the data is taken care of by the end user.

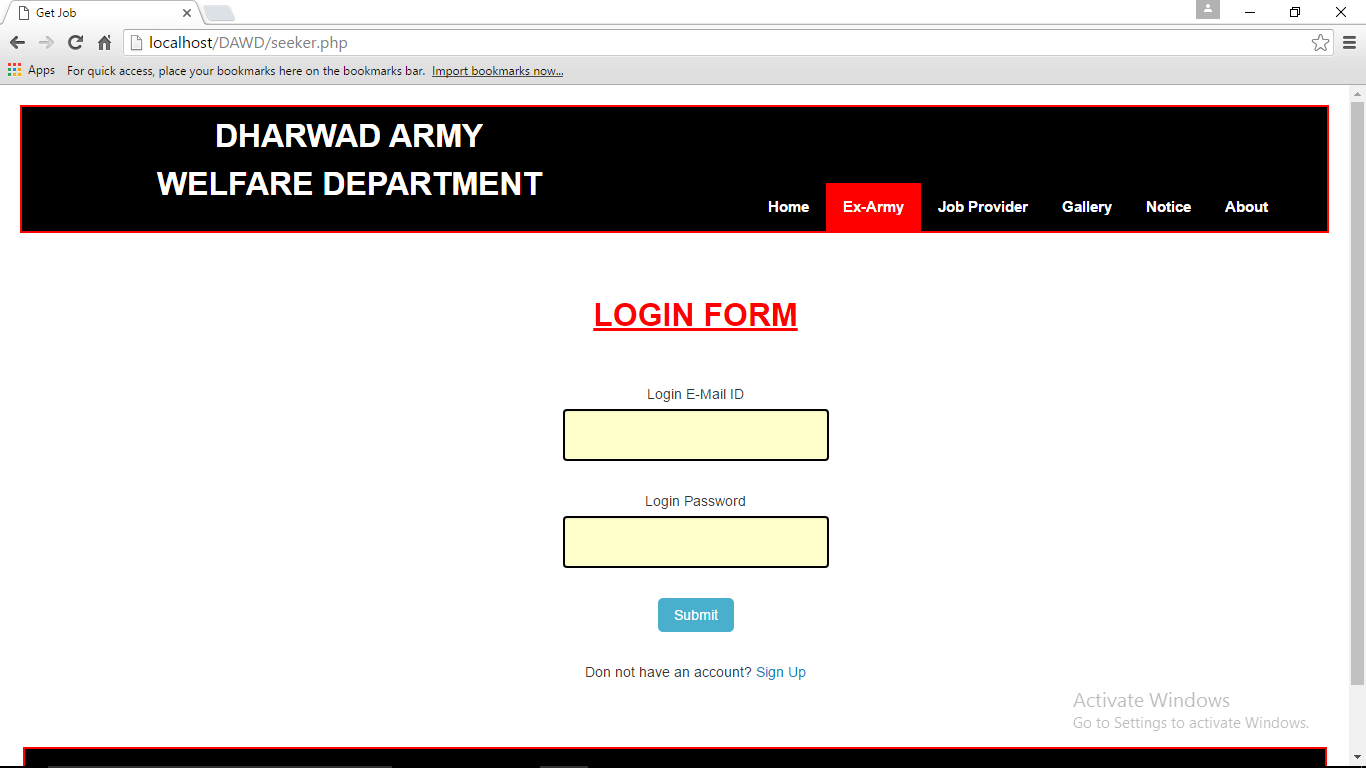
**5.4. SCREENSHOTS:**

* **Home Page:**

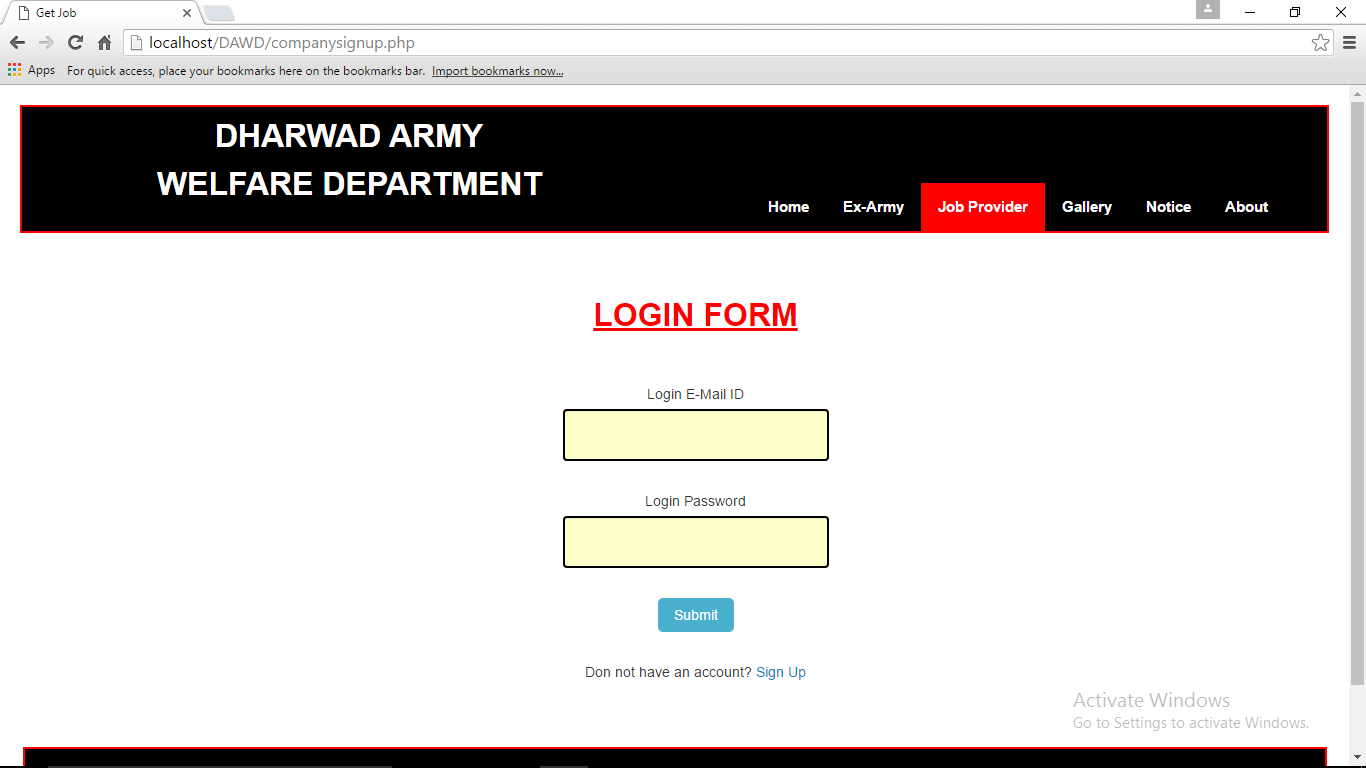




* **Login Page for Ex-army:**



* **Login Page for Officer:**



**5.5. CODING**

**LOGIN:**

<?php

if($\_SERVER['REQUEST\_METHOD']=='POST')

{

include 'connect123\_db.php';

require 'connect.php';

$errors=array();

if(empty($\_POST['email']))

{

$errors[]='Enter Your Email Address';

}

else

{

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

}

if(empty($\_POST['pass1']))

{

$errors[]='Enter Password';

}

else

{

$pass=$\_POST['pass1'];

}

if(empty($errors))

{

$q="SELECT \* FROM `seeker` WHERE `EMAIL`='$email' AND `Password`='$pass' AND `approve`='yes'";

$r=mysqli\_query($dbc,$q);

if($r)

{

if(mysqli\_num\_rows($r)!=0)

{

$q1="SELECT \* FROM `seeker` WHERE `EMAIL`='$email' AND `Password`='$pass' AND `State`='NULL' AND `approve`='yes'";

$r1=mysqli\_query($dbc,$q1);

$row=mysqli\_fetch\_array($r1,MYSQLI\_ASSOC);

session\_start();

if(mysqli\_num\_rows($r1)!=0)

{

$\_SESSION['seekerid']=$row['SeekerId'];

load('Seeker1.php');

}

else

{

$row2=mysqli\_fetch\_array($r,MYSQLI\_ASSOC);

$\_SESSION['seekerid']=$row2['SeekerId'];

load('update\_seeker.php');

}

}

else

{

echo 'Email address and Password Does not Exists.Click here to

<a href="seeker.php">Register</a> <br/>OR Wait for admin to

give authentication';

}

}

mysqli\_close($dbc);

exit();

}

else

{

echo '<h1> Error!</h1>

<p>The following error(s) occured:<br>';

foreach($errors as $msg)

{

echo "-$msg<br>";

}

echo 'please try again';

mysqli\_close($dbc);

}

}

?>

**REGISTRATION:**

<?php

if($\_SERVER['REQUEST\_METHOD']=='POST')

{

/\* DB BY KK \*/

/\*

$dbc=mysqli\_connect('localhost','root','','job')

OR die(mysqli\_connect\_error());

mysqli\_set\_charset($dbc,'utf\_8'); \*/

include 'connect123\_db.php';

$errors=array();

if(empty($\_POST['email']))

{

echo $errors[]='Enter Your Email Address';

}

else

{

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

}

if(!empty($\_POST['pass1']))

{

if($\_POST['pass1']==$\_POST['pass2'])

{

$pass=$\_POST['pass1'];

}

else

{

$errors[]='Password Did not Matched';

}

}

else

{

$errors[]='Enter Password';

}

if(empty($errors))

{

$q="SELECT \* FROM `seeker` WHERE `EMAIL`='$email'";

$r=mysqli\_query($dbc,$q);

if(mysqli\_num\_rows($r)!=0)

{

$errors[]='Email address already registered';

}

}

if(empty($errors))

{

$q="INSERT INTO seeker (EMAIL, Password, FirstName, MidleName,

LastName, DOB, Gender, State, Address, District, City, MobileNo, LandNo,

Experience, Skills, FunArea, BasicEdu, MasterEdu, ResumeName,

ImageName,approve) VALUE('$email','$pass','NULL','NULL','NULL', 'NULL', 'NULL','NULL', 'NULL','NULL','NULL','NULL','NULL','NULL',

'NULL','NULL','NULL', 'NULL','NULL','NULL','no') ";

$r=mysqli\_query($dbc,$q);

if($r)

{

?>

<div class="list-container clearfix">

<h3 class="title-heading">Registered Successfull</h3>

<div class="span6 ">

<article class="property-item clearfix">

<?php echo 'Thank You For Registering With DAW CELL. You May LOGIN Later After Admin Approvals.

<a href="seeker.php">Login</a>'; ?>

</article>

</div>

<?php

}

mysqli\_close($dbc);

exit();

}

else

{

?>

<div class="list-container clearfix">

<h3 class="title-heading">Error Occured</h3>

<div class="span6 ">

<article class="property-item clearfix">

<?php

echo '<p class="tip">The folowing errors(s)occured:<br/>';

foreach($errors as $msg)

{

$message="-$msg<br/>";

echo $message;

}

echo 'please try again.<a href="seeker.php">Click Here</a>

<i class="icon- remove"></i></p>';

?>

</article>

</div>

<?php

mysqli\_close($dbc);

}

}

?>

**SEARCH:**

<?php

if($\_SERVER['REQUEST\_METHOD']=='POST')

{

$errors=array();

if(empty($\_POST['job']))

{

$errors[]= 'Not entered job description';

}

else

{

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

}

if(empty($\_POST['state']))

{

$errors[]='Not entered State';

}

else

{

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

}

if($\_POST['experience']==0)

{

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

}

else

{

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

}

if(empty($errors))

{

$q="SELECT \* FROM `vacancy` WHERE `JobTitle`='$job'

AND `State`='$state' AND `experience`='$experience'";

$r=mysql\_query($q);

if($r)

{

$fetch=mysql\_num\_rows($r);

?>

<h3 class="title-heading">Available Jobs<span id="jobs-counter">

(<?php echo $fetch; ?>)</span></h3>

<div class="about-agent clearfix">

<?php

if($fetch>0)

{

?>

<?php

while($row=mysql\_fetch\_array($r))

{

$com=$row['CompanyId'];

$q3=mysql\_query("SELECT \* FROM `company`

WHERE `CompanyId`='$com'");

if((mysql\_num\_rows($q3))>0)

{

while($row2=mysql\_fetch\_array($q3))

{

?>

<div class="span6 ">

<article class="property-item clearfix">

<h5 class="price">

<?php

echo $row2['CompanyName'].'</h5>';

}

}

else

{

echo '<div class="span6 ">

<article class="property-item clearfix">

<h5 class="price">Get Job</h5>';

}

?>

<figure>

<a href="#" >

<img src="upload/Company/<?php echo $row['ImageName'];?>"

class="attachment-property-thumb-image wp-post-image" >

</a>

</figure>

<div class="detail">

<h4><span>

Job Title:<?php echo $row['JobTitle'];?>

</span></h4>

<h4><span>

Requirement:<?php echo $row['Requirement'];?>

</span></h4>

<p><span>

<?php echo $row['ProfileName'];?>

</span></p>

<h4><span>

<?php echo $row['Address'];?>,

<?php echo $row['City'];?>,

<?php echo $row['District'];?>,

<?php echo $row['State'];?>,

<?php echo $row['Country'];?></span></h4>

<?php $vacancy=$row['vacancy'];?>

<?php $compid=$row['CompanyId'];?>

</p>

</div>

<div class="follow-agent clearfix">

<?php

echo "<a class=\"real-btn btn\" href=\"seekerapply.php?vacancy=$vacancy\">Apply</a>";

?>

</div>

</article>

</div>

<?php

}

}

else

{

?>

<div class="list-container clearfix">

<div class="span6 ">

<article class="property-item clearfix">

<?php echo 'No result found for your searched result';?>

</article>

</div>

<?php

}

}

else

{

echo mysql\_error();

}

}

Else

{

?>

<div class="list-container clearfix">

<h3 class="title-heading">Error Occured</h3>

<div class="span6 ">

<article class="property-item clearfix">

<?php

echo '<p class="tip">The folowing errors(s)occured:<br/>';

foreach($errors as $msg)

{

$message="-$msg<br/>";

echo $message;

}

echo 'please try again<i class="icon-remove"></i></p>';

?>

</article>

</div>

<?php

}

}

?>

**6. TESTING**

**6.1. TESTING:**

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

**Who Does Testing?**

It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in the context of the given requirements. Moreover, developers also conduct testing which is called Unit Testing. In most cases, following professionals are involved in testing of a system within their respective capacities:

* Software Tester
* Software Developer
* Project Lead/Manager
* End User

Different companies have difference designations for people who test the software on the basis of their experience and knowledge such as Software Tester, Software Quality Assurance Engineer, and QA Analyst etc. It is not possible to test the software at any time during its cycle. The next two sections state when testing should be started and when to end it during the SDLC.

An early start to testing reduces the cost, time to rework and error free software that is delivered to the client. However in Software Development Life Cycle (SDLC) testing can be started from the Requirements Gathering phase and lasts till the deployment of the software. However it also depends on the development model that is being used.

For example in Water fallmodel formal testing is conducted in the Testing phase, but in incremental model, testing is performed at the end of every increment/iteration and at the end the whole application is tested.

Testing is done in different forms at every phase of SDLC like during Requirement gathering phase, the analysis and verifications of requirements are also considered testing. Reviewing the design in the design phase with intent to improve the design is also considered as testing. Testing performed by a developer on completion of the code is also categorized as Unit type of testing.

Unlike when to start testing it is difficult to determine when to stop testing, as testing is a never ending process and no one can say that any software is 100% tested. Following are the aspects which should be considered to stop the testing:

* Testing Deadlines.
* Completion of test case execution.
* Completion of Functional and code coverage to a certain point.
* Bug rate falls below a certain level and no high priority bugs are identified.
* Management decision.

**6.2. TESTING METHODS:**

* **Black Box Testing:**

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

* **White Box Testing:**

White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

**6.3. TESTING STRATEGIES:**

There are two general strategies for testing software. There are follows

* **Code Testing :**

This examines the logic of the program. To follow this test, cases are developed such that very path of the program is tested.

* **Specification Testing :**

Specification testing examines the specifications starting what the program should do and how it should perform under various conditions. Then test cases are developed for each conditions and combinations of conditions and to be submitted for processing.

**6.4. STAGES IN THE TESTING PROCESS:**

* **Unit Testing:**

Individual components are tested to ensure that they operate correctly. Each component is tested independently without other system components.

Ex. Checked for Login and Password with the table.

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

**6.5. LIMITATIONS OF UNIT TESTING:**

Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing. There is a limit to the number of scenarios and test data that the developer can use to verify the source code. So after he has exhausted all options there is no choice but to stop unit testing and merge the code segment with other units.

* **Module Testing**:

Module is a collection of dependent components such as an object classes an abstract data type or some looser collection of procedures and functions. A module encapsulates related components so can be tested without other system modules.

* **Subsystem Testing :**

This phase involves testing collection of modules, which have been integrated into subsystems. Subsystems may be independently designed and implemented. The most common problems which arise in the large software systems are subsystems interface mismatches. The subsystem test process should therefore concentrate on the detection of interface errors by rigorously exercising these interfaces.

* **System Testing :**

The subsystems are integrated to make up the entire system. The testing process is concerned with finding errors, which result from unanticipated interactions between subsystems and system components. It is also concerned with validating that the system is functional and non-functional requirements.

**Ex:** Those all subsystems are integrated and checked for inter-dependency between the subsystems.

* **Acceptance Testing :**

This is final stage in testing process before the system is tested for operational use. The system is tested with data supplied by the system procurer rather than simulated test data. Acceptance testing may reveal errors and omissions in the systems requirements definitions because the real data exercises the system in different phase from the test data.

**6.6. TEST CASES AND RESULTS:**

**Here Are The Few Testing Conditions:**

**Test Case 1:**

**Test data:** Blank username.

**Message:** you need to enter your username.

**Test data:** Blank password.

**Message:** you need to enter your password.

**Test data:** Incorrect username or password.

**Message:** Problem logging in.

**Test Case 2:**

**Test data:** Blank text boxes, invalid current password, new password, confirm password.

**Expected result:** please enter current password, please enter new password, please enter confirm password, enter proper current password, enter proper confirm password.

**Observed result**: message saying please enter current password, please enter new password, please enter confirm password, enter proper current password, enter proper confirm password.

**Remark:** input validation succeeded.

**Test Case 3**:

**Test data:** blank text boxes, invalid data for different textboxes

**Expected result:** please enter proper data in the respective field, respective field cannot be empty.

**Observed result:** message saying please enter proper data, field cannot be empty.

**Remark**: input validation succeeded.

**Test Case 4:**

**Test data:** blank text boxes, invalid data for different textboxes

**Expected result:** please enter proper data in the respective filed, respective field cannot be empty Observed result: message saying please enter proper data, filed cannot be empty.

**Remark**: input validation succeeded.

**7. FUTURE ENHANCEMENT:**

* The Ex-army people will apply for jobs.
* User will apply for army employment jobs.
* We will provide the list of Ex-army people.
* We will update the news about air force, navy and military.
* We will make for whole Karnataka.

**8. CONCLUSION:**

Using this web application we can help track of army retired peoples information and their current status. It is used to maintain the army welfare department profile. This web application is will be helpful for admin or head offer at army welfare department in Dharwad. Basically they collect army info through papers, jobs and news updates posted on notice boards will be avoided by using these applications. It is used to post the jobs and their requirements to Ex-army people. It provide the army employment news for user. We can get information about army like history, their eligibility for air force, navy, military.

**9. BIBLIOGRAPHY:**

* Database System concepts(TMH edition) Silberschatz, Korth, Sundarshan
* The Complete Reference - Helbert schildt published fifth edition.
* PHP-MySQL-Dummies-3rd-edition
* MySQL Guide
* YouTube videos on PHP by The New Boston
* Lynda.com. PHP. With. MySQL. Essential. Training.
* Server side programming with PHP web programming

**WEBSITES:**

* http://php.net.in/php/
* http://php/w3schools/products/reference/faqs/index.html
* http://www.apt.jhu.edu/~hall/php/