**CHAPTER 1**

**INTRODUCTION**

### **An Overview**

MYJOB.COM is an effective tool to provide help for common people when they meet several problems for finding a job. When the users are searching for a job they get confused with lots of criteria’s they followed in a job oriented websites. I propose a system that will immediately solve the problem faced for a job search. This Project is aimed at developing an online search Portal for the Placement Dept. of the college. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as an Online Job Portal for the Placement Dept of the college to inform the student about the new job vacancies information with regards to placement.

* 1. **Objective of the project**

MYJOB.COM is a web application built in JAVA. It provide lots of functionality different from other job portal websites . It have different modules including Admin, Candidate/Job seeker, College, Company. It provides the candidates, ability to register and search for jobs, also they get job notifications as per their qualification. Also the registered colleges can view the job notifications post by the company they can directly contact by the companies as given in the notification field. Here all colleges can register and this college account is managed by the corresponding placement officers of the colleges. The Employers should also register in this site and company could post new vacancies , also they can post the result of the interview they conducted . Admin is the main head to manage these candidates,colleges and employers.

**CHAPTER 2**

**SYSTEM ANALYSIS**

System analysis is the detailed study of the existing system. It is conducted to identify the user requirements, evaluate the system concept for feasibility, perform economical and technical analysis, allocate functions to hardware, software, people, database and other system elements, establish cost and schedule constraints and create a system definition. System analysis gives the structure and functioning of the system. It helps to understated the problem and emphasize what is needed from the system. The various techniques used in the phase are observation, interview, record searching and discussions.

2.1 **Existing System**

The existing system is a manual system with limitations in accuracy, expense, low speed ,efficiency and unformatted outputs. In the existing system, all data processing is done manually. All the files and record books are replaced by the software system. When there are a lot of issues such as retrieval and storage of the information and keeping track of them becomes a tedious task. By implementing a computerized system, the limitation in the present system will be reduced. Manpower can be reduced to a great extent and efficiency and accuracy can be increased to manifold. More over consumption of time can be reduced to far greater extend by the implementation of the proposed system.

**2.2 Proposed System**

In the proposed system we propose to computerize the above-mentioned activities. In the existing system, all data processing is done manually. When there are a lot of issues such as retrieval and storage of the information and keeping track of them becomes a tedious task. By implementing a computerized system, the limitation in the present system will be reduced. Manpower can be reduced to a great extent and efficiency and accuracy can be increased to manifold. More over consumption of time can be reduced to far greater extend by the implementation of the proposed system.

The proposed computer based information system is aimed at eliminating the limitations of the existing system via increasing speed of information retrieval, by increasing processing speed, data security and hence give accurate reports. The proposed system fully satisfies user needs. All most all the limitations of the existing system are

reduced in a great manner. Details are stored in various files from which data can be accessed and manipulated easily.In the proposed system there are two sections one is administration part in which the following processes are carried out.

**MERITS OF PROPOSED SYSTEM**

* A lot of manual work which is time consuming and tedious can be reduced.
* Eliminates wrong entries and hence give accurate report.
* Reduce paper work and extra cost.
* Data redundancy can be avoided.
* Data security.
* Information retrieval is faster.
* User friendliness.
* Flexibility.
* The number of registers can be reduced.
* The new system provides data consistency and integrity so that data at any time represent the current status.
* The system should provide better security and control over data.

**2.3Feasibility Study**

The main objective of the feasibility study is to treat the technical, operational, logical and economic feasibility of developing the computerized system. All systems are feasible, given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at System Study Phase itself. The feasibility study to be conducted for this project involves:

* + - 1. Technical Feasibility.
      2. Operational Feasibility.
      3. Economic Feasibility
    1. **Technical Feasibility**

Technical feasibility includes the hardware and software facilities for the successful completion of the projects. The latest hardware and software facility for the successful completion of the project. With this latest hardware and software support the system will perform extremely well. The system is available through Google play store.

**2.3.2 Operational Feasibility**

The user finds no difficulty in operations. Emergency situation is handled by the system onwards its fully automated. System provides situation. This system, will handles the request in a better way and make the process easier thus, it is sure that the system developed is operationally feasible.

**2.3.3 Economic Feasibility**

In the economic feasibility the development cost of the system is evaluated weighing it against the ultimate benefit derived from the new system. It is found that the benefit, from the new system would be more than the cost and time involved in its development. The tools selected for developing the software are PHP 2005, SQL Server

2005, Windows XP which are all readily / freely available in the market. The report generation module which has direct interface with the end user (viewer) can run in a windows environment. It is coded in such a way that not much change occurs while

operations are done. The system is menu driven and meaningful texts are given as hyperlinks; it is user friendly and easy for the staff as well as the users to operate.

**2.4 PROJECT PLANNING**

Project Planning is an aspect of Project Management that focuses a lot on Project Integration. The project plan reflects the current status of all project activities and is used to monitor and control the project. The Project Planning tasks ensure that various elements of the Project are coordinated and therefore guide the project execution.

Project planning helps in

* Facilitating communication
* Monitoring /measuring the project progress and
* Provides overall documentations of assumptions/planning decisions

The project planning phases can be broadly classified as follows

* Development of the project plan
* Execution of the project plan
* Change control and corrective actions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **April 2017** | **May 2017** | **June 2017** | **July 2017** | **Aug 2017** |
|  |  |  |  |  |

**Planning**

**Requirement**

**Analysis**

**Design**

**Development**

**Testing**

**Implementat**

**ion**

Fig 2.4 Gantt chart

**Chapter 3**

**ENVIRONMENT**

**3.1 SYSTEM SPECIFICATION**

**3.1.1 Software Specification**

|  |  |  |
| --- | --- | --- |
| **3.1.1.1 Client Side** |  |  |
| Operating System | : | Windows XP/Higher |
| Front End | : | HTML |
| Back End | : | MySQL |
| Editor | : | Netbeans |
| **3.1.1.2 Server Side** |  |  |
| Operating System | : | Windows XP/Higher |
| Front End | : | HTML |
| Back End | : | MySQL |
| Editor | : | Netbeans |

**3.1.2 Hardware Specification**

|  |  |  |
| --- | --- | --- |
| **3.1.2.1 Client Side** |  |  |
| Processor | : | 800 MHz ARM 11 |
| RAM | : | 128 MB |
| Display | : | TFT capacitive touch screen |
| Card slot | : | microSD |
| Mobile Phone | : | Mobile Phone with Android OS |
| **3.1.2.2 Server Side** |  |  |
| Processor | : | Intel Core i3 (5th Gen) |
| Clock Speed | : | 2.86GHZ Processor |
| Hard disk | : | 20 GB |
| RAM | : | 4GB |
| Display | : | Color Monitor |

Compatible Keyboard and Mouse

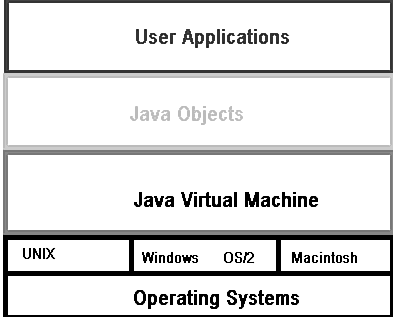
**3.2 Software Technology Review**

**3.2.1 Java**

Back in 1990, a gentleman by the name of James Gosling was given the task of creating programs to control consumer electronics. Gosling and his team of people at Sun Microsystems started designing their software using C++, the language that most programmers were praising as the next big thing because of its object-oriented nature. Gosling, however, quickly found that C++ was not suitable for the projects he and his team had in mind. They ran into trouble with complicated aspects of C++ such as multiple inheritances of classes and with program bugs such as memory leaks. Gosling soon decided that he was going to have to come up with his own, simplified computer language that would avoid all the problems he had with C++.

## 3.2.1.1 Introducing Java

By now, everyone may be curious why Java is considered such a powerful tool for Internet development projects. We already know that Java is a simplified version of C++. Anyone who has struggled with learning C++ knows that the key word in the previous sentence is "simplified." C++ added so much to the C language that even professional programmers often have difficulty making the transition.



According to Sun Microsystems, Java is "simple, object-oriented, statically typed, compiled, architecture neutral, multi-threaded, garbage collected, robust, secure, and extensible." That's a mouthful, but this description of Java probably doesn't help to understand the language much. The following list of Java's attributes, however, should clear out some of the cobwebs:

* Simple. Java's developers deliberately left out many of the unnecessary features of other high-level programming languages. For example, Java does not support pointer math, implicit type casting, structures or unions, operator overloading, templates, header files, or multiple inheritance.
* Object-oriented. Just like C++, Java uses classes to organize code into logical modules. At runtime, a program creates objects from the classes. Java classes can inherit from other classes, but multiple inheritances, wherein a class inherits methods and fields from more than one class, are not allowed.
* Statically typed. All objects used in a program must be declared before they are used. This enables the Java compiler to locate and report type conflicts.
* Compiled. Before user can run a program written in the Java language, the Java compiler must compile the program. The compilation results in a "byte-code" file that, while similar to a machine-code file, can be executed under any operating system that has a Java interpreter. This interpreter reads in the byte-code file and translates the byte-code commands into machine-language commands that can be directly executed by the machine that's running the Java program. One could say, then, that Java is both a compiled and interpreted language.
* Multi-threaded. Java programs can contain multiple threads of execution, which enables programs to handle several tasks concurrently. For example, a multi-threaded program can render an image on the screen in one thread while continuing to accept keyboard input from the user in the main thread. All applications have at least one thread, which represents the program's main path of execution.
* Garbage collected. Java programs do their own garbage collection, which means that programs are not required to delete objects that they allocate in memory. This relieves programmers of virtually all memory-management problems.
* Robust. Because the Java interpreter checks all system access performed within a program, Java programs cannot crash the system. Instead, when a serious error is discovered, Java programs create an exception. This exception can be captured and managed by the program without any risk of bringing down the system.
* Secure. The Java system not only verifies all memory access but also ensures that no viruses are hitching a ride with a running applet. Because pointers are not supported by the Java language, programs cannot gain access to areas of the system for which they have no authorization.
* Extensible. Java programs support native methods, which are functions written in another language, usually C++. Support for native methods enables programmers to write functions that may execute faster than the equivalent functions written in Java. Native methods are dynamically linked to the Java program; that is, they are associated with the program at runtime. As the Java language is further refined for speed, native methods will probably be unnecessary.
* Well-understood. The Java language is based upon technology that's been developed over many years. For this reason, Java can be quickly and easily understood by anyone with experience with modern programming languages such as C++.

As one can tell from the preceding list of features, a great deal of thought went into creating a language that would be fairly easy to use but still provide the most powerful features of a modern language like C++. Thanks to features such as automatic garbage collection, programmers can spend more time developing their programs rather than wasting valuable man-hours hunting for hard-to-find memory-allocation bugs. However, features such as Java's object-oriented nature, as well as its ability to handle multiple threads of execution, ensure that the language is both up-to-date and powerful.

**3.2.1.2** **The Java Developer's Kit**

Java is actually more than a computer language; it's also a programming environment that includes a complete set of programming tools. These tools include a compiler, an interpreter, a debugger, a disassembler, a profiler, and more. To create a Java program, first use a text editor to create the source-code file. Users write the source code, of course, in the Java language. After completing the source code, which is always saved with a .java file extension, compile the program into its byte-code format, the file for which has the .class file extension. It is the .class file that the interpreter loads and executes. Because the byte-code files are fully portable between operating systems, they can be executed on any system that has a Java interpreter.

After compiling and running a Java program, one may discover that the source code needs modification. The Java debugger can help to find errors, whereas the Java profiler provides handy information about program. If users run into a compiled Java program that like to see in source-code form, the Java disassembler will do the translation. Java also includes a program that creates the files need to take advantage of native methods (functions written in another language, such as C++). There's even a program that can create HTML documents from Java source-code files. Although all the development tools are DOS applications-that is, they don't run under Windows-they provide a complete environment for creating and managing Java projects. The development environment for Java is surprisingly rich, especially considering that the Java programming language is fairly new. Most Java developers and programmers find that the JDK contains everything they need to get started creating powerful Java programs. This is because the developer's kit includes the Java programming language core functionality, the Java Application Programming Interface (API) complete with multiple package sets, and essential tools for creating Java programs. The Java Developer's Kit is currently available for most operating systems. One can obtain the Sun Solaris, Windows 95/NT, and Macintosh versions directly from JavaSoft. Because the Java Soft FTP site is extremely busy.

The heart of the Java programming language is contained in a set of packages called java.lang, which is a part of the Java Application Programming Interface. Although the java.lang package provides the core functionality of the Java programming language, it is not the only package included in the Java Developer's Kit.

The JDK includes the following packages: java.applet, java.awt, java.awt.image, java.awt.peer, java.io, java.lang, java.net, and java.util. These packages provide everything to start creating powerful Java applications quickly. The JDK also includes an additional package called sun.tools.debug, which is designed to make the application-debugging process easier.

|  |  |  |
| --- | --- | --- |
| Package | Package Name | Description |
| java.applet | Applet | A set of classes that relate to the applet environment and are generally used when viewing applets |
| java.awt | Abstract Windowing Toolkit | A set of classes that provide graphical interface tools such as buttons, controls, scrollbars, and windows |
| java.awt.image | AWT Image | A set of classes related to using images |
| java.awt.peer | AWT Peer | A set of classes for AWT peer classes and methods |
| java.sql | Database connectivity | A set of classes that enable developers to write Java applications that access databases |
| java.io | I/O | A set of classes that provide standard input/output and file I/O utilities |
| java.lang | Language | The core set of classes for the Java language that provide basic functions, such as string and array handling |
| java.net | Network | A set of classes that provide tools for accessing networks by protocols, such as FTP, Telnet, and HTTP |
| java.util | Utility | A set of classes that provide core utility functions such as encoding/decoding, hash tables, and stacks |
| sun.tools.debug | Debug | A set of classes that provide debugging functions and tools |

## 3.2.1.3 Java Programs

Java is first and foremost an object-oriented programming language. Many programmers are surprised when they discover how easy it is to follow sound object-oriented design practices with Java. Java can be used to create two types of programs: applets and stand-alone applications. An Applet is simply a part of a Web page, just as an image or a line of text can be. Just as a browser takes care of displaying an image referenced in an HTML document, a Java-enabled browser locates and runs an Applet . When Java-capable Web browser loads the HTML document, the Java applet is also loaded and executed. Using applets, one can do everything from adding animated graphics to our Web pages to creating complete games and utilities that can be executed over the Internet. Some applets that have already been created with Java include Bar Chart, which embeds a configurable bar chart in an HTML document; Crossword Puzzle, which enables users to solve a crossword puzzle on the Web; and LED Sign, which presents a scrolling, computerized message to viewers of the Web page within which the applet is embedded.

One of Java's major strengths is that one can use the language to create dynamic content for your Web pages. That is, thanks to Java applets, Web pages are no longer limited to the tricks one can perform with HTML. Now Web pages can do just about anything users want them to. All need to do is write the appropriate applets.

But writing Java applets is only half the story. How Web page's users obtain and run the applets is equally as important. It's only write the applet (or use someone else's applet), but also to provide users access to the applet. Basically, Web pages can contain two types of applets: local and remote. In this section, one can learn the difference between these applet types, which are named after the location at which they are stored.

Local Applets:-A local applet is one that is stored on own computer system .When Web page must find a local applet, it doesn't need to retrieve information from the Internet-in fact, browser doesn't even need to be connected to the Internet at that time.

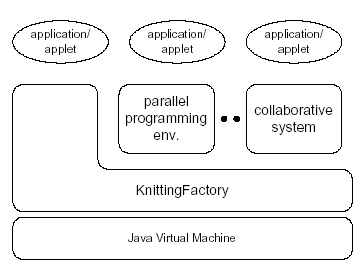


Figure – Applets/Application programs operate on Java Virtual Machine

Remote Applets:-A remote applet is one that is located on another computer system. This computer system may be located in the building next door or it may be on the other side of the world-it makes no difference to Java-compatible browser. No matter where the remote applet is located, it's downloaded onto computer via the Internet. Browser must, of course, be connected to the Internet at the time it needs to display the remote applet.

To reference a remote applet in Web page, users must know the applet's URL (where it's located on the Web) and any attributes and parameters that users need to supply in order to display the applet correctly. If users didn't write the applet, they’ll need to find the document that describes the applet's attributes and parameters. The applet’s author usually writes this document. It composes an HTML <applet> tag that accesses a remote applet.

**3.2.2**  **The Primary Characteristics of Java**

The ease with which one can transition to Java from C/C++ and Java's object-oriented capabilities are only the icing on the cake. Java has many powerful characteristics, most of which are:

* Architecture-neutral
* Distributed
* Dynamic
* Interpreted and compiled
* Multithreaded
* Network-ready and compatible
* Object-oriented
* Portable
* Robust
* Secure

These characteristics are the source of Java's power and the reason for Java's widespread popularity. Many of these characteristics are interrelated and are a direct outgrowth of Java's development for the following:

* Distributed networked environments
* High performance
* Easy reuse of code
* Security

## 3.2.3 Object-Oriented Programming

Object-oriented programming enables to think of program elements as objects. In the case of a window object, users don't need to know the details of how it works, nor do users need to know about the window's private data fields. Users need to know only how to call the various functions (called methods in Java) that make the window operate. Consider the car object discussed in the previous section. To drive a car, users don't have to know the details of how a car works. Users need to know only how to drive it. What's going on under the hood is none of business. (And, if one casually try to make it your business, plan to face an amused mechanic who will have to straighten out mess)

But OOP is a lot more than just a way to hide the details of a program. To learn about OOP, one needs to understand three main concepts that are the backbone of OOP.

These concepts, which are covered in the following sections, are: encapsulation, inheritance, and polymorphism.

### **Objects**

The fundamental unit in object-oriented programming is the object. Languages that follow object-oriented concepts describe the interaction among objects. All objects have a state and a behavior.

The state of an object pertains to data elements and their associated values. Everything the object knows about these elements and values describes the state of the object. Data elements associated with objects are called instance variables.

The behavior of an object depends on the actions the object can perform on the instance variables defined within the object. In procedural programming, such a construct would be called a function. In object-oriented terminology, this construct is called a method. A method belongs to the class it is a member of, and one can use a method when he need to perform a specific action more than once.

Thus, the state of an object depends on the things the object knows, and the behavior of the object depends on the actions the object can perform. If user creates a software object that models the television, the object would have variables describing the television's current state, such as it is on, the current channel setting is 8, the current volume setting is 23, and there is no input coming from the remote control. The object would also have methods that describe the permissible actions, such as turn the television

on or off, change the channel, change the volume, and accept input from the remote control.

### **Encapsulation**

One major difference between conventional structured programming and object-oriented programming is a handy thing called encapsulation. Encapsulation enables to hide, inside the object, both the data fields and the methods that act on that data. (In fact, data fields and methods are the two main elements of an object in the Java programming language.) After do this, users can control access to the data, forcing programs to retrieve or modify data only through the object's interface. In strict object-oriented design, an object's data is always private to the object. Other parts of a program should never have direct access to that data. How does this data hiding differ from a structured-programming approach? After all, one can always hide data inside functions, just by making that data local to the function. A problem arises; however, when one want to make the data of one function available to other functions. The way to do this in a structured program is to make the data global to the program, which gives any function access to it. It seems that users could use another level of scope-one that would make your data global to the functions that need it-but still prevent other functions from gaining access. Encapsulation does just that. In an object, the encapsulated data members are global to the object's methods, yet they are local to the object. They are not global variables.

### **Classes as Data Types**

An object is just an instance of a data type. For example, when you declare a variable of type int, you're creating an instance of the int data type. A class is like a data type in that it is the blueprint upon which an object is based. When users need a new object in a program, one can create a class, which is a kind of template for the object. Then, in program, users create an instance of the class. This instance is called an object.

Classes are really nothing more than user-defined data types. As with any data type, users can have as many instances of the class as users want. For example, users can have more than one window in a Windows application, each with its own contents.Classes encapsulate objects. A single class can be used to instantiate multiple objects. This means that one can have many active objects or instances of a class. The object describing the functions of television is an instance of a class of objects called television.

Each object within a class retains its own states and behaviors. By encapsulating objects within a class structure, one can group sets of objects by type. The Java Application Programming Interface (API) describes many classes. Each class in the API specifies a set of objects that perform related functions and share common characteristics. Classes programmer create can serve a similar purpose For example, think again about the integer data type (int). It's absurd to think that a program can have only one integer. One can declare many integers, just about all. The same is true of classes. After one defines a new class, it can create many instances of the class. Each instance (called an object) normally has full access to the class's methods and gets its own copy of the data members.

**Inheritance**

Inheritance enables to create a class that is similar to a previously defined class, but one that still has some of its own properties. Consider a car-simulation program. Suppose that anyone have a class for a regular car, but now users want to create a car that has a high-speed passing gear. In a traditional program, users might have to modify the existing code extensively and might introduce bugs into code that worked fine before changes.

To avoid these hassles, one can use the object-oriented approach: Create a new class by inheritance. This new class inherits all the data and methods from the tested base class. Users can control the level of inheritance with the public, private, and protected keywords.

### **Libraries**

In C++ and other programming languages, a collection of related classes or functions is called a library. Java puts a twist on the concept of libraries by using the term package to describe a collection of related classes. Just as classes encapsulate objects, packages encapsulate classes in Java.

### **Polymorphism**

The last major feature of object-oriented programming is polymorphism. By using polymorphism, one can create new objects that perform the same functions as the base object but which perform one or more of these functions in a different way. For example, user may have a shape object that draws a circle on the screen. By using polymorphism, one can create a shape object that draws a rectangle instead. User does this by creating a new version of the method that draws the shape on the screen. Both the old circle drawing and the new rectangle-drawing method have the same name but accomplish the drawing in a different way.

**3.2.4 Introduction to Mysql**

What is it?

* Command line SQL and PL\SQL language interface to the database
* It can be used as an interactive interface or it can be driven from scripts.
* Its equivalents in other databases are as:
  + Ingres: SQL
  + Sybase and SQLServer: isql
  + IBM DB2:db2
  + PostgresQL: psql
  + MySQL:mysql

What is it for?

* Connect to Oracle
* Create, change, delete database objects and data
* Ad hoc query data objects and data
* Verify the functioning of your applications

**Overview of Application Architecture**

### There are two common ways to architect a database: client/server or multitier. As internet computing becomes more prevalent in computing environments, many database management systems are moving to a multitier environment.

### ***Client/Server Architecture***

### Multiprocessing uses more than one processor for a set of related jobs. Distributed processing reduces the load on a single processor by allowing different processors to concentrate on a subset of related tasks, thus improving the performance and capabilities of the system as a whole.

### An Oracle database system can easily take advantage of distributed processing by using its client/server architecture. In this architecture, the database system is divided into two parts: a front-end or a client, and a back-end or a server.

##### The Client

##### The client is a database application that initiates a request for an operation to be performed on the database server. It requests, processes, and presents data managed by the server. The client workstation can be optimized for its job. For example, it might not need large disk capacity, or it might benefit from graphic capabilities.Often, the client runs on a different computer than the database server, generally on a PC. Many clients can simultaneously run against one server.

##### The Server

##### The server runs Oracle software and handles the functions required for concurrent, shared data access. The server receives and processes the SQL and PL/SQL statements that originate from client applications. The computer that manages the server can be optimized for its duties. For example, it can have large disk capacity and fast processors.

#### 

#### Multitier Architecture: Application Servers

A multitier architecture has the following components:

* A client or initiator process that starts an operation
* One or more application servers that perform parts of the operation. An application server provides access to the data for the client and performs some of the query processing, thus removing some of the load from the database server. It can serve as an interface between clients and multiple database servers, including providing an additional level of security.
* An end or database server that stores most of the data used in the operation

This architecture enables use of an application server to do the following:

* Validate the credentials of a client, such as a Web browser
* Connect to an Oracle database server
* Perform the requested operation on behalf of the client

If proxy authentication is being used, then the identity of the client is maintained throughout all tiers of the connection.

**Overview of Physical Database Structures**

The following sections explain the physical database structures of an Oracle database, including datafiles, redo log files, and control files.

*Data files*

Every Oracle database has one or more physical datafiles. The datafiles contain all the database data. The data of logical database structures, such as tables and indexes, is physically stored in the datafiles allocated for a database.The characteristics of data files are:

* A data files can be associated with only one database.

### Datafiles can have certain characteristics set to let them automatically extend when the database runs out of space.

### One or more data files form a logical unit of database storage called a table space.

Data in a data files is read, as needed, during normal database operation and stored in the memory cache of Oracle. For example, assume that a user wants to access some data in a table of a database. If the requested information is not already in the memory cache for the database, then it is read from the appropriate datafiles and stored in memory.Modified or new data is not necessarily written to a data file immediately. To reduce the amount of disk access and to increase performance, data is pooled in memory and written to the appropriate data files all at once, as determined by the [database writer process (DBWn)](http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/glossary.htm#i996724) background process.

**3.2.5 Java Data Base Connectivity (JDBC)**

Sun developed a single API for data base access-JDBC. They kept three main goals in mind:

* + 1. JDBC should be a SQL-Level API.
    2. JDBC should capitalize on the experience of existing database API’s.
    3. JDBC should be simple.

JDBC is a SQL level API that allows you to embed SQL statements as arguments to the methods in JDBC interface. To enable you to do this in a database independent fashion, JDBC requires database venders to furnish a runtime implementation of its interface.

These implementations route your SQL calls to the database in the proprietary fashion it recognizes.Java provides database programmers with the following features they have traditionally lacked:

1. Easy object to relational mapping.
2. Database independence.
3. Distributed computing.

Java’s database connectivity allows you to worry about the translation of relational data into objects instead of worrying about how you are getting that data. The JVM provides an application with a guaranteed runtime environment; no administration is needed for the configuration of that environment for individual applications.

JDBC architecture as follows:

1. JDBC Driver Manager.
2. JDBC Driver.
3. JDBC-ODBC Bridge.
4. Application.

JDBC ARCHITECTURE

|  |
| --- |
| APPLICATION |
| JDBC NET DRIVER |
| JDBC NATIVE DRIVER |
| JDBC DRIVER MANAGER |
| NATIVE PROTOCOL JDBC DRIVER |
| JDBC ODBC DRIVER |
| ODBC |

## ->Appln🡪JDBC🡪 JDBC-ODBC Driver🡪 ODBC->Database

JDBC DriverManager: Function of the driver manager is to find out available drivers in the system and connect the application to the appropriate database, whenever a connection is requested. However, to help the driver manager identify different types of drivers, each driver should be registered with the driver manager.

JDBC Driver: Function of the JDBC Driver is to accept the SQL calls from the application and convert them into native calls to the database. However, in this process it may take help from some other drivers or even servers, which depends on the type of JDBC Driver. It also is possible that the total functionality of the database server could be built into the driver itself.

JDBC-ODBC Bridge: Sun soft provides a special JDBC Driver called JDBC-ODBC Bridge driver, which can be used to connect to any existing database, that is ODBC complaint.

Application: Application is a java program that needs the information to be modified in some database or wants to retrieve the information.

**CREATING A CONNECTION and CONNECTING A DATABASE:**

Java.Sql.Driver: JDBC is a launching point for data base connectivity by responding to driver Manager. Driver Manager connection requests and providing information about the implementation in question.

Java.Sql.DriverManager: DriverManager is class instead of an interface. Its main responsibilities are to maintain the list of Driver implementation and present an application with one that matches a requested URL. The driver manager provides register Driver () and deregister Driver (), which allow the Driver implementation to register itself with the Driver Manager or remove itself from that list. You can get an enumeration of register drivers through the getDriver().

Java.Sql.Connection: The Connection class represents a single logical database connection. In other words we use the connection class for sending a series of sql statement to the database and managing the committing or aborting of those statements.

PreparedStatement: The prepared Statement enables sql statements to contain the parameters like a function definition and can execute a single statement repeatedly with different values for those parameters. The act of assigning values to parameters is called binding parameters. You might want to use a prepared statement in updating a group of objects stored on same table.

This interface is used to retrieve the data from the database in second form. This interface will prepare a SKELTON statement at compile time. To execute the SKELTON we have to set the values for the statement and execute it at run time.

**Chapter 4**

**SYSTEM DESIGN**

System design involves translating information requirements and conceptual design into technical specification and general flow of processing. After the project idea is specified, related information is gathered and a way to evolve the system is developed. The process starts with a study of the outputs required of the system, the inputs given, the data that is to be stored, how to store the data, the data structured for storage.

**4.1 Input Design**

Input design is the method by which valid data are accepted from the user. This part of the designing requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify these errors. Inaccurate input data are the most common cause of errors in data processing. Input design consists of the following processes:-

* Designing graphical user entry screen is easy to use.
* Designing procedures and functions to valid the data as per business rules.
* Designing functions needed to store data into a usable form for processing.
* Designing the common integrated functions that can be used by all other users when needed.

**4.1.1 Input Objectives**

Controlling Amount of Input: Wherever user input is required, giving possible input values as default in that area reduces the amount of user keystrokes. Thus the user can pass on to next data without much typing. This makes the data entry much fast and error free. When the user has the format of input to be given, it will be very easy for the user to give input in the same format.

Avoiding Delay: A processing delay resulting from data entry operations is called a bottleneck. Such bottlenecks are made obsolete in this project by breaking up the amount of data to be entered in each form into different smaller and simpler forms.

Avoiding Errors in Data: The rate at which errors occur depends on the quantity of the data. As told in the above objective these errors are reduced by making the number of data to be entered in each form is reduced.

Avoiding Extra Steps: To fulfill any operation the user have no need to do complex steps, instead any operation can be done with simple easy to use steps.

**4.2 Output design**

Output design is one of the most important features of the information system. When the output is not of good quality, the users will be averse to use the newly designed system and may not use the system. There are many types of outputs, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used.

Outputs from computer system are required primarily to communicate the results of processing to users. They are also used to provide a permanent hard copy of the results for later consultation.

**4.2.1 Output Objectives**

The output from an information system should accomplish one or more of the following objectives:

* Convey information about past activities, current status, or projections of the future.
* Signal important events, opportunities, problems or warnings
* Trigger an action
* Confirm an action

**4.3 Data Flow Diagram**

Data Flow Diagram (DFD) representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources. The purpose of DFD is to provide a semantic bridge between users and system developers. The diagram is the basis of structured system analysis.

A level 0 DFD, also called a fundamental system model or a context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e., Level 1 DFD. Each of the processes represented at Level 1 are sub functions of overall system depicted in the context model.

Any processes, which are complex in Level 1, will be further represented into sub functions in the next level, i.e., in level 2.Data flow diagrams illustrate how data is processed by a system in terms of inputs, and outputs. Represent major components or functions with Circles. Actions for input by a user or a system go in Rectangular Boxes. Databases are represented by Parallel lines enclosing a phrase corner.

**Data Flow Diagram – The Rules**

1. External Entities

External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system’s inputs and outputs.

2. Processes

When a naming processes, avoid glossing over them, without really understanding their role. Indications that this has been done are the use of vague terms in the descriptive title area like ‘process’ or ‘update’. The most important thing is that the description must be meaningful to whoever will be using the diagram.

3. Data Flows

Double-headed arrows can be used (to show two-way flows) on all but bottom level diagrams. Furthermore, in common with most of the other symbols used, a data flow at a particular level of a diagram may be decomposed to multiple data flows at lower levels.

4. Data Store

Data store represent stores of data within the systems and are represented by open rectangle. Data Flows represent the movements of data between other components and are shown by arrows.

Rectangular box defined a source of destination of the system

A circle stands for processes that convert data into information

Data Flow

A Parallel line is a data store

Candidate

Admin

Company

College

Request

Request

Request

Request

Response

Response

Response

Response

**Fig. 4.3.1 Context Level (Level 0)**

jobpool

Cand\_reg

feedback

Candidate

email

password

login

profile

jobs

Applied list

login

Candi-box

apply

feedback

Profile

Result

result

registration

Upload resume

**Fig. 4.3.2 Candidate (Level 1)**

colg\_reg

College

email

password

login

registration

jobs

login

profile

feedback

jobpool

feedback

Applied list

apply

**Fig. 4.3.3 College (Level 1)**

comp\_reg

Company

email

password

login

registration

result

login

jobs

jobpool

result

feedback

feedback

profile

**Fig. 4.3.4 Company (Level 1)**

Admin

email

password

login

login

Cand\_reg

Candidate details

Comp\_reg

Colg\_reg

Comp\_reg

Colg\_reg

Company details

College details

Candidate

College

Company

Feedback

feedback

Cand\_reg

**Fig. 4.3.5 Admin (Level 1)**

# MODULE DESCRIPTION

* + 1. **Candidate Module**

1.Registration

New Users can register.

2.Login

After the verification of admin the approved candidate can login using their e-mail and password.

3.Apply job

They can view new job notifications and they can apply .

4.Review of applied job

They can view the vacancies they already applied in the site.

5.Profile Editing

They can change their profile details and change their passwords also.

6.Send Feedback

They can send feedback to the admin.

**4.4.2. College Module**

1.Registration

New colleges can register their college

2.Login

After the verification of admin the approved colleges can login using e-mail and password.

3.Apply Job

Colleges can view new job notifications and they can apply .

4.Review of applied job

They can view the vacancies they already applied in the site.

5.Profile Editing

They can change their profile details and change their passwords also.

6.Send Feedback

They can send feedback to the admin.

**4.4.** **3. Company Module**

1.Registration

New companies can register .

2.Login

After the verification of admin the approved companies can login using e-mail and password.

3.Post New Vacancies

Companies can post the vacancies in their company.

4.Post Final Results.

They can publish the final results of interviews conducted on their company

5.Profile Editing

They can change their profile details and change their passwords also.

6.Send Feedback

They can send feedback to the admin.

**4.4.4. Admin Module**

1.Verification of Candidate

Admin verify the new Candidates , approved candidates can only login otherwise they cannot use the system.

2.Verification of College

Admin verify the new Colleges , approved colleges can only login otherwise they cannot use the system.

3.Verification of Company

Admin verify the new Companies , approved companies can only login otherwise they cannot use the system.

4.View the list of Candidates

Admin can view all the candidate details.

5.View the list of Companies

Admin can view all the company details.

6.View the list of Colleges

Admin can view all the college details.

7.View the feedbacks

Admin can view all the feedbacks from candidates, colleges and companies.

**4.5 Analysis Tools**

System analysis is the process of collecting and interpreting facts, understanding problems and using the information to suggest improvements on the system. This will help to understand the existing system and determine how computers make its operation more effective.

**4.5.1. Use Case Diagram**

In software engineering, a use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The use case diagram shows the position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the customer/owner/user and the development team.

Candidate

**Fig. 4.5.1.1 Use Case Diagram for Candidate**

College

**Fig. 4.5.1.2 Use Case Diagram for College**

Company

**Fig. 4.5.1.3 Use Case Diagram for Company**

Admin

**Fig. 4.5.1.4 Use Case Diagram for Admin**

**4.6 Architectural Design**

Server Side

Component

Mobile Equipment

Database

Registration

Location

Information

Communicated to

Server

Login

Request for

access

Load

complaint

Location based

Computation

Location

Information

identified

Generate Alert

Display

Content on

Mobile

**4.7 Database Design**

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. The main objectives of databases are data integrity and data independence . A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users Database Management System (DBMS) allow the data to be protected and organized separately from other resources like hardware, software, and programs. DBMS is a software package, which contains components that are not found other data management packages. The significant of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. That is the difference between the logical and physical data.

The main objectives covered in database design are:

* Controlled redundancy
* Data independence
* Accuracy and integrity
* Privacy and security
* Performance

The design process consists of the following steps

* Determine the purpose of the database: This helps prepare for the remaining steps.
* Find and organize the information required: Gather all of the types of information to record in the database.
* Divide the information into tables: Divide information items into major entities or subjects.
* Turn information items into columns: Decide what information needs to be stored in each table. Each item becomes a field, and is displayed as a column in the table.
* Specify primary keys: Choose each table’s primary key. The primary key is a column, or a set of columns, that is used to uniquely identify each row.
* Set up the table relationships: Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.
* Refine the design: Analyze the design for errors. Create tables and add a few records of sample data. Check if results come from the tables as expected. Make adjustments to the design, as needed.
* Apply the normalization rules: Apply the data normalization rules to see if tables are structured correctly. Make adjustments to the tables, as needed.

**4.7.1 Normalization**

Normalization is the process of decomposing the attributes in an application. This results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

Update anomalies: If data items are scattered and are not linked to each other properly, then it could lead to strange situations. For example, when we try to update one data item having its copies scattered over several places, a few instances get updated properly while a few others are left with old values. Such instances leave the database in an inconsistent state.

Deletion anomalies: We tried to delete a record, but parts of it were left undeleted because of unawareness, the data is also saved somewhere else.

Insert anomalies: We tried to insert data in a record that does not exist at all.

Normalization is a method to remove all these anomalies and bring the database to a consistent state and free from any kind of anomalies. The main objectives of normalization are:

* Eliminate data redundancy
* Ensure data integrity

**4.7.1.1 First Normal Form**

Tables are said to be in first normal form when:

* Each table has a primary key.
* No single attribute (column) has multiple values.
* The non-key attribute depend on the primary key.

To convert the table to 1Nf: Remove repeating data i.e.multi-valued attributes and create separate tables for those attributes.

**4.7.1.2 Second Normal Form**

Tables are said to be in second normal form when:

* It is in 1NF.
* If the primary key is composite of attributes (contains multiple columns),the non key attributes must depend on the whole key.

If composite key is used and a partial dependency is found hence remove the partially dependent attribute and place it in a separate table along with the determinant as the primary key for the new table.

**4.7.1.3 Third Normal Form**

Tables are said to be in third normal form when:

* It is in 2NF.
* No transitive functional dependency.

Transitive dependency is defined as follows.

If A->B, B->C then A->C.

The database tables used in this project are given below

1. Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| Email | Varchar | 30 | Not null | Email ID |
| Password | Varchar | 10 | Not null | Password |
| Type | Varchar | 10 | Not null | User type |

1. Candi\_reg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| Cand\_id | Varchar | 10 | Primary key | Candidate id |
| Name | Varchar | 30 | Not null | Name |
| Email | Varchar | 30 | Not null | Email |
| Mobile | Varchar | 12 | Not null | Mobile |
| Password | Varchar | 10 | Not null | Password |

1. Candibox

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| Can\_id | Varchar | 20 | Foreign key | Candidate id |
| Name | Varchar | 50 | Not null | Name |
| Gender | Varchar | 20 | Not null | Gender |
| Dob | Date | 20 | Not null | Date of birth |
| Mobile No | Varchar | 12 | Not null | Mobile number |
| Email | Varchar | 50 | Not null | Email |
| Address | Varchar | 100 | Not null | Address |
| Key Skills | Varchar | 100 | Not null | Key skills |
| Password | Varchar | 20 | Not null | Password |
| Photoup | Varchar | 100 | Not null | Photo upload |
| Resumup | varchar | 100 | Not null | Resume upload |

1. collg\_reg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| Col\_id | Varchar | 10 | Primary key | College id |
| Name | Varchar | 20 | Not null | College name |
| Officer | Varchar | 20 | Not null | Placement officer |
| Email | Varchar | 20 | Not null | Email |
| Address | Varchar | 50 | Not null | Address |
| Mobile No | Varchar | 12 | Not null | Mobile No |
| Password | Varchar | 10 | Not null | Password |

1. comp\_reg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| Com\_id | varchar | 10 | Primary key | Company id |
| Name | varchar | 30 | Not null | Company name |
| Owner | varchar | 30 | Not null | Owner |
| Email | varchar | 30 | Not null | Email |
| Mobile No | varchar | 12 | Not null | Mobile number |
| Address | varchar | 50 | Not null | Address |
| Password | varchar | 10 | Not null | Password |

1. jobpool

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| J\_id | varchar | 10 | Primary key | Job id |
| Com\_email | varchar | 10 | Foreign key | Company id |
| Job Title | varchar | 20 | Not null | Job title |
| Qualification | varchar | 20 | Not null | Qualification |
| Experience | varchar | 20 | Not null | Experience |
| Job Location | varchar | 30 | Not null | Job location |
| Salary | varchar | 10 | Not null | Salary |
| Venu,time | Date | 10 | Not null | Interview date time |
| Last date | Date | 10 | Not null | Last date of apply |

1. apply

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| A\_id | Varchar | 10 | Primary key | Feedback id |
| Candi\_name | Varchar | 50 | not null | Candidate name |
| Candi\_email | Varchar | 100 | Not null | candidate email |
| Company | Varchar | 100 | Not nul | Company |
| Resume | Varchar | 100 | Not null | Resume |
| Date | Date |  | Not null | Apply date |

1. Feedback

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraints | Description |
| f\_id | Varchar | 10 | Primary key | Feedback id |
| Name | Varchar | 50 | not null | User name |
| Email | Varchar | 100 | Not null | User email |
| Msg | Varchar | 100 | Not nul | Feedback message |
| Date | Date |  | Not null | Feedback date |

**Chapter 5**

**SYSTEM TESTING AND IMPLEMENTATION**

System testing is the process of checking whether the developed system is working according to the original objectives and requirements. Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification. Design and coding .Testing presents an interesting anomaly for the software. Testing is vital to the success of the system. Errors can be injected at any stage during the development .system testing makes logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. During testing the program to be tested is executed with set of test data and the output of the program for the test data is evaluated to determine if the program is performing as expected. A series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**5.1 TESTING METHODS**

Software testing methods are traditionally divided into black box testing and white box testing.

**5.1.1 White Box testing**

White box testing is a testing in which the software tester has knowledge of the inner workings structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

**White box testing steps:**

* Create test plans.
* Profile the application block.
* Test the internal subroutines.
* Test loops and conditional statements.
* Perform security testing**.**

**5.1.2 Black Box Testing**

Black box testing is a testing in which software without any knowledge of the inner workings structure or languages of the module being tested Black box tests, as most other kinds of tests ,must be written from a definitive source document, such as specifications or requirement document .It is a testing in which the software under test is treated as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Black box testing steps**

* Create test plans.
* Test the external interfaces.
* Perform load testing.
* Perform stress testing.
* Perform security testing.
* Perform globalization testing

**5.2 LEVELS OF TESTING**

Before the system is ready for the user to use it, a series of testing are performed for the proposed system. The testing steps are:

* Unit Testing.
* Integration Testing
* Validation Testing.
* Output Testing.
* User acceptance Testing.

**5.2.1 Unit Testing:**

This is the first level of testing. In this different modules are tested against the specification produced during design of the module. Unit testing is done during the coding the coding phases for testing the logical correctness of the modules. It refers to the verification of a single program module in an isolated environment. Unit testing first focuses on the modules independently of one other to locate errors. After coding each dialogue is tested and run individually. All unnecessary coding were removed and it was ensured that all the modules have worked as the programmer expected. Logical errors found were corrected.

**5.2.2 Integration Testing:**

This data can be lost across an interface. Integration testing is a systematic testing for constructing the program structure. Conducting the test is uncovering errors associated within the interface. The objective is to take the unit tested modules and build a program structure. All the modules combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes.

**5.2.3 Validation Testing:**

At the end of integration testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. The software is completely assembled as a package.

**5.2.4 Output Testing:**

After performing the validation testing, the next test is the output test of the proposed system, since no system could be useful if it does not provide the required output in a specific format. The output format on the screen is found to be as expected by the user. After coding each dialogue is tested and run individually. All unnecessary coding were removed and it was ensured that all the modules have worked as the programmer expected. Logical errors found were corrected.

**5.2.5 User Acceptance Testing:**

Acceptance test refers to acceptance of data into the system for processing. The acceptance test contributes to the consistency and smooth working of the system. The system under consideration is tested for users at a time of developing and making changes whenever required.

**5.3 TEST CASES**

In the project various test cases are implemented. They are given below.

**5.3.1 Validation Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Procedure** | **Expected Result** | **Condition** | **Remark** |
|  |  |  |  |
| Register User | Display Error message | Blank | Pass |
|  |  |  |  |
| Enter Name | Alphabet only | Enter digit | Pass |
|  |  |  |  |
| Enter Mobile number | Number only | Enter number | Pass |
|  |  |  |  |
| Email id | Display Error message | Do not enter @ and dot | Pass |
|  |  | Symbol |  |
|  |  |  |  |

Table : 6.3.1 Validation test cases

**5.3.2 Functional Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Being** | **Initial System State** | **Input** | **Expected Output** |
| **Tested** |  |  |  |
|  |  |  |  |
| System is started | System is off | Activate the on State | System request |
|  |  |  | username and |
|  |  |  | password for login |
| System read the | System is asking for | Enter username and | System displays a |
| username and | username and | Password | message |
| Password | password |  |  |
|  |  |  |  |
| System handle an | System asking for | Enter the incorrect | The invalid |
| invalid username | username and | username or password | username or |
| or password | password |  | password extension |
|  |  |  | is performed |
| System handle an | System asking for | Invalid location | Logout |
| invalid | Location on |  |  |
|  |  |  |  |
| System perform a | System is displaying | Registered user can send resume directly,  colleges can view and get the information the upcoming vacancies | System send a |
| Legitimate | the various job posted by the company |  | notification to the |
| Various |  |  | email that the |
| Search of job |  |  | complaint is |
| Properly |  |  | completed |
| System verifies | System is verifying the registered user’s | If the system approve the user their details are stored | System give a provision for the registered users |
| that user’s |  | If the system declaim a user registration they are deleted from the table |  |
|  |  |  |  |
|  |  |  |  |
| fulfill the registration |  |  |  |
| Profile | System ask for edit | Current password, | Change your |
| Editing | profile | new and conform | password |
|  |  | Password | successfully |
|  |  |  |  |
| Feedback is sent to the admin by users | System send feedback to the admin | System can view the feedback | Feedback is sended and viewed successfully |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table: 6.3.2 Functional test cases

**5.4 TEST RESULT**

Test results emphasize how the actual results differed from the expected results. This suggests the need for re-testing, and to discover the source of differences. The test phase of systems development process involves the defining of the criteria by which the system will be tested and measuring the criteria against the acceptable failure rate. Individual modules are tested during the development itself. Errors detected are corrected and re-tested, and the project leader has verified the compliance. Each input, output and processes are tested to verify that it performs as specified in the design. The units in the system are re-compiled and errors found are corrected as indicated by the compiler. The tests are repeated until all known errors are eliminated and the program matched the design specifications. Separate tests are performed to ensure that program units are properly interfaced with each other to form a complete system. All the test cases passed successfully. No defects encountered

**5.5 IMPLEMENTATION**

Implementation is the stage of project when the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on existing practices shifts to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. The implementation stage is a systems project in its own right. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, training of staff in the changeover procedure and evaluation of changeover methods.

**5.5.1 Implementation Planning**

The implementation of a system involves people from different departments and system analysts are confronted with the practical problem of controlling the activities of people outside their own data processing department. Prior to this point in the project, system analyst has interviewed department staff with the permission of their respective managers. The implementation coordination committee should be responsible for a successful implementation. The composition of committee is important. There should be at least one representative of each department affected by the changes and other members should be co-opted for discussions of specific topics.

**CHAPTER 7**

**CONCLUSION**

The project was successfully completed within the time span allotted. Every effort has been made to present the system in more user friendly manner. All the activities provide a feeling like an easy walk over to the user who is interfacing with the system. A trial run of the system has been made and is giving good results.

The software has been developed in JSP. All the modules are tested separately and put together to form the main system the system has been developed in an attractive dialogs fashion. So user with minimum knowledge about the computers can operate the system easily. By using this software the user and administrator are able to communicate in a faster and better way and will be more secure so that the system is developed in an efficient manner from unauthorized access.

**APPENDIX-A- SCREEN SHOTS**

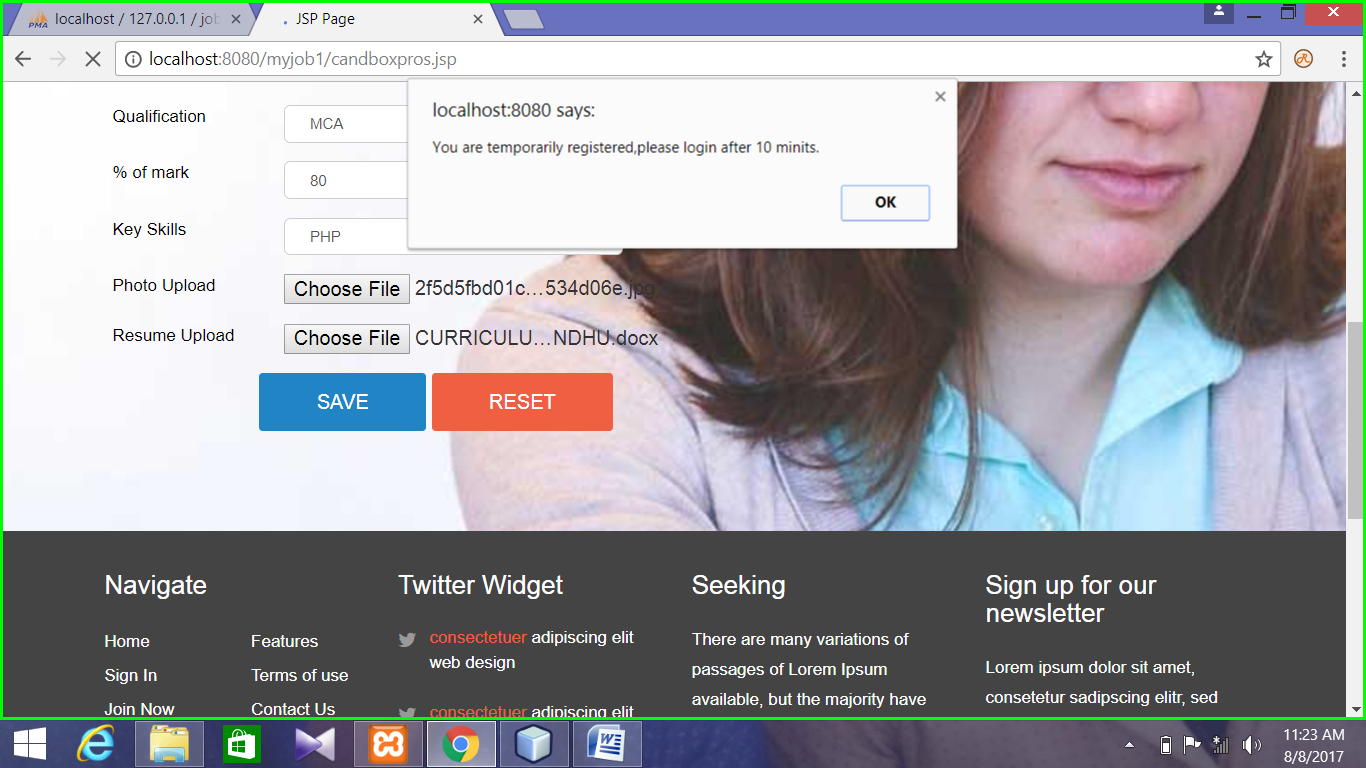
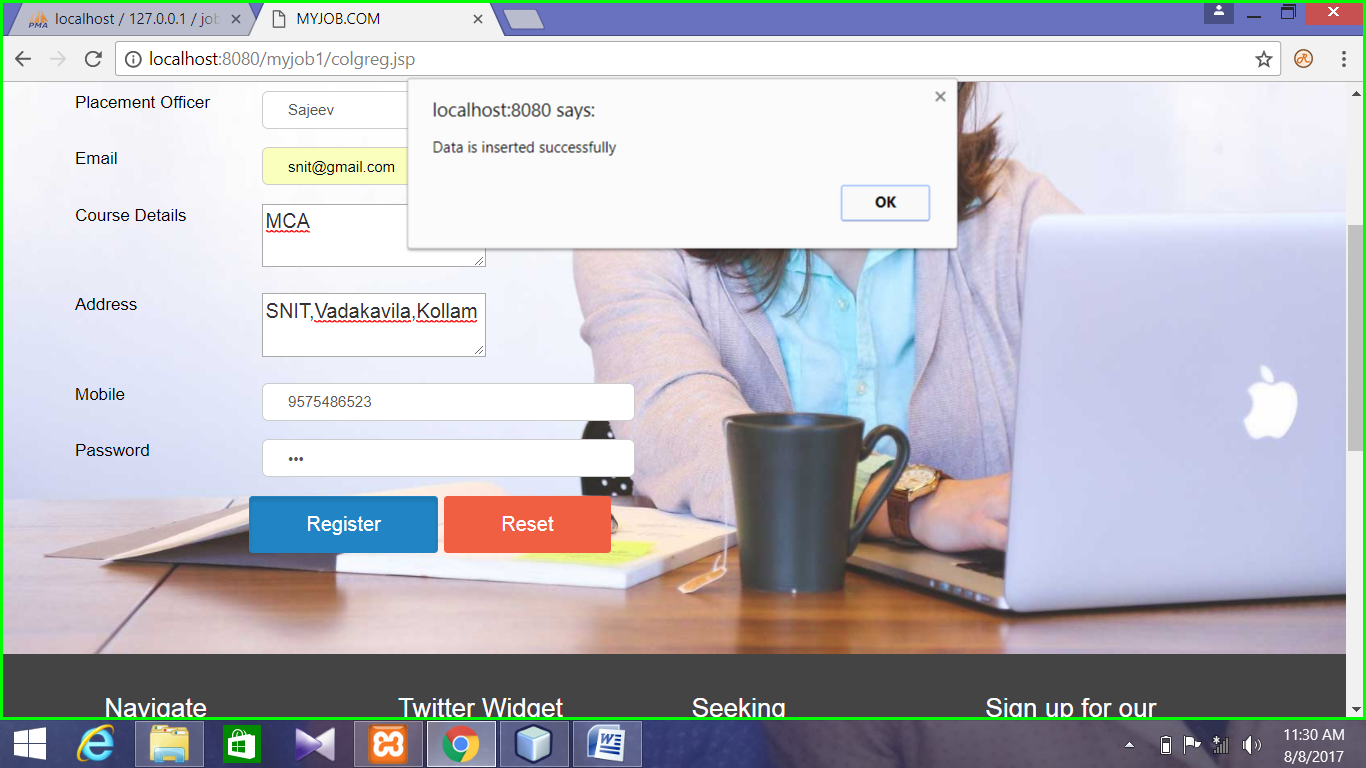
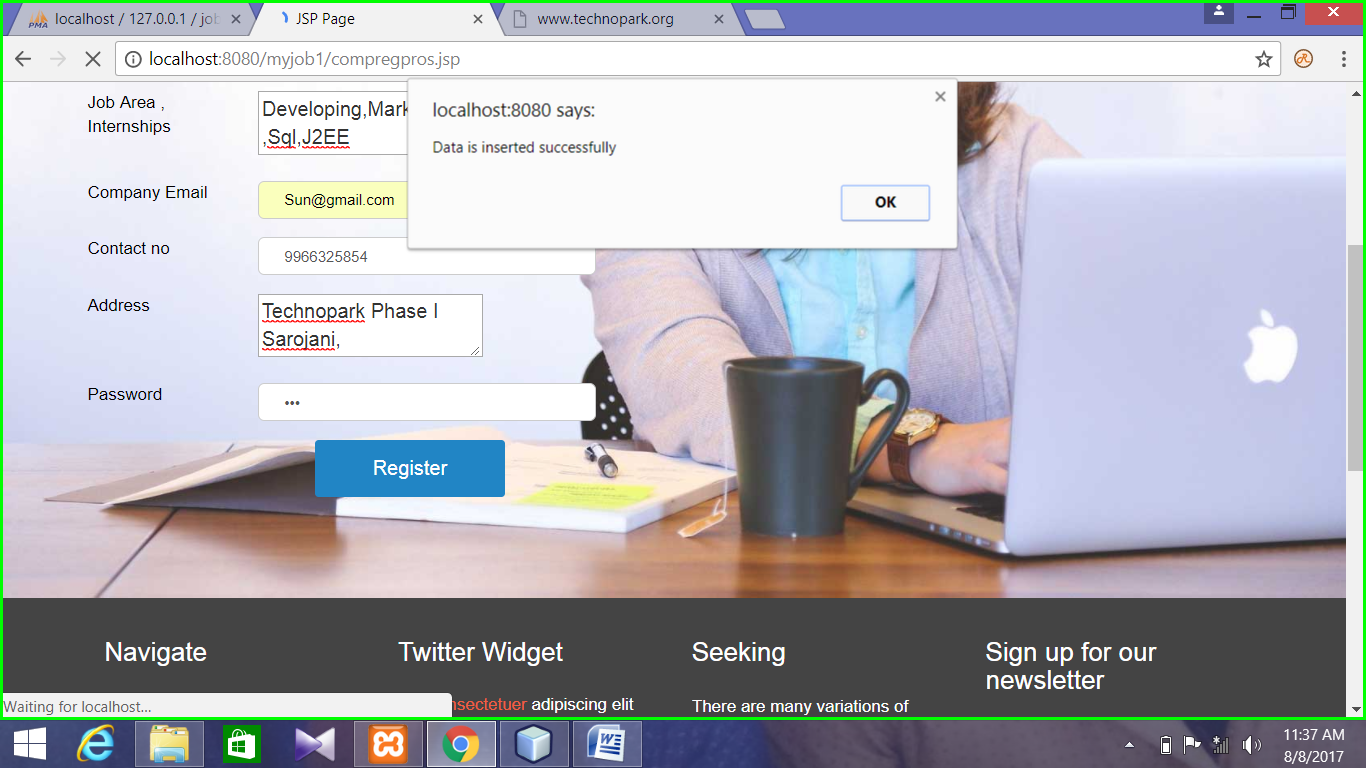


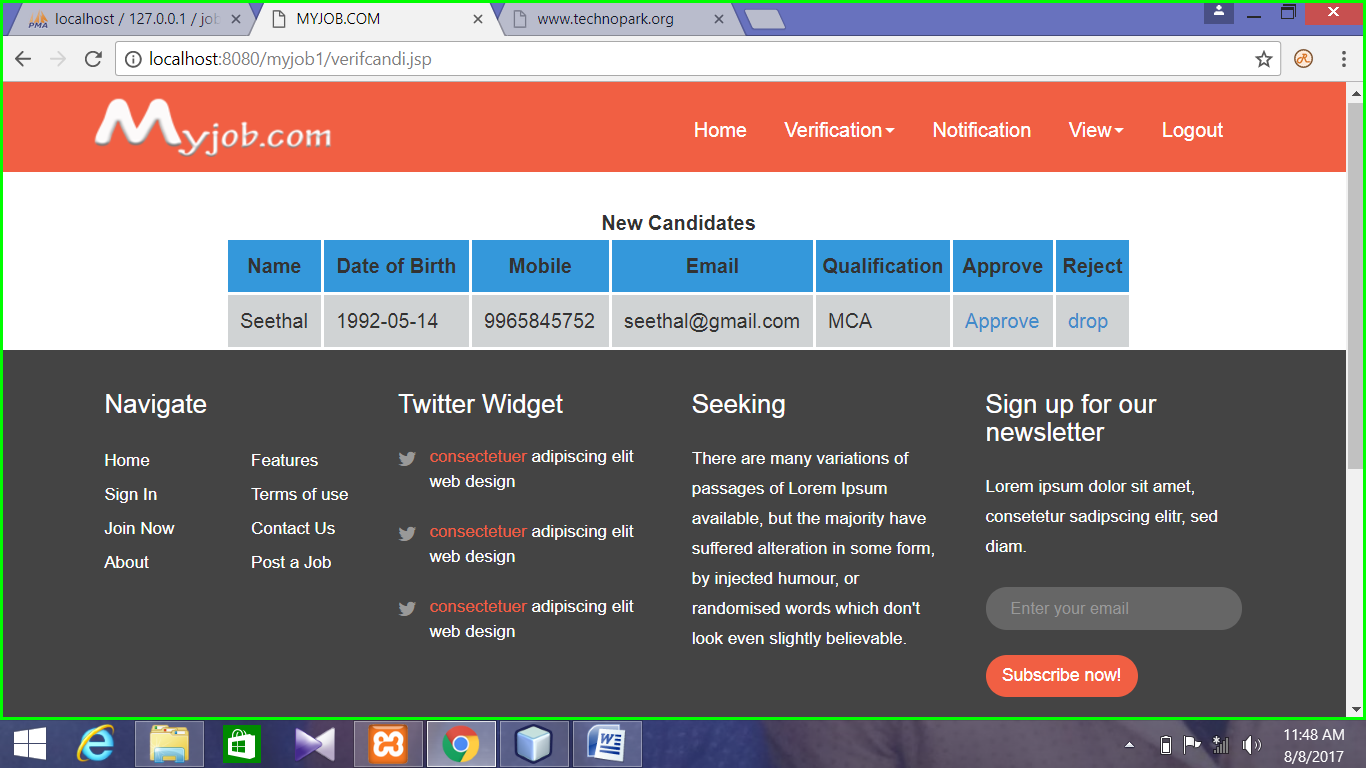
Fig A.1 **Candidate/ Job seeker Registration**

****

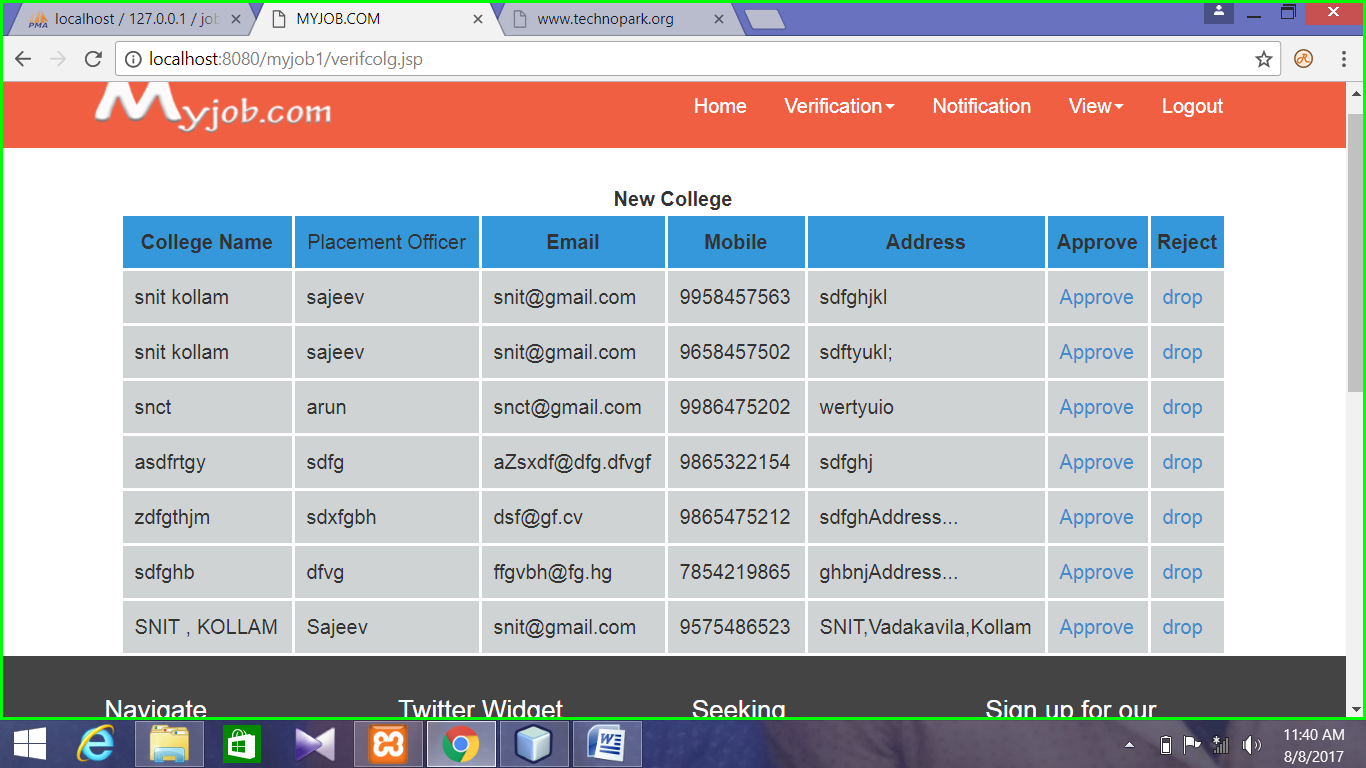
**Fig A.2 College Registration**

****

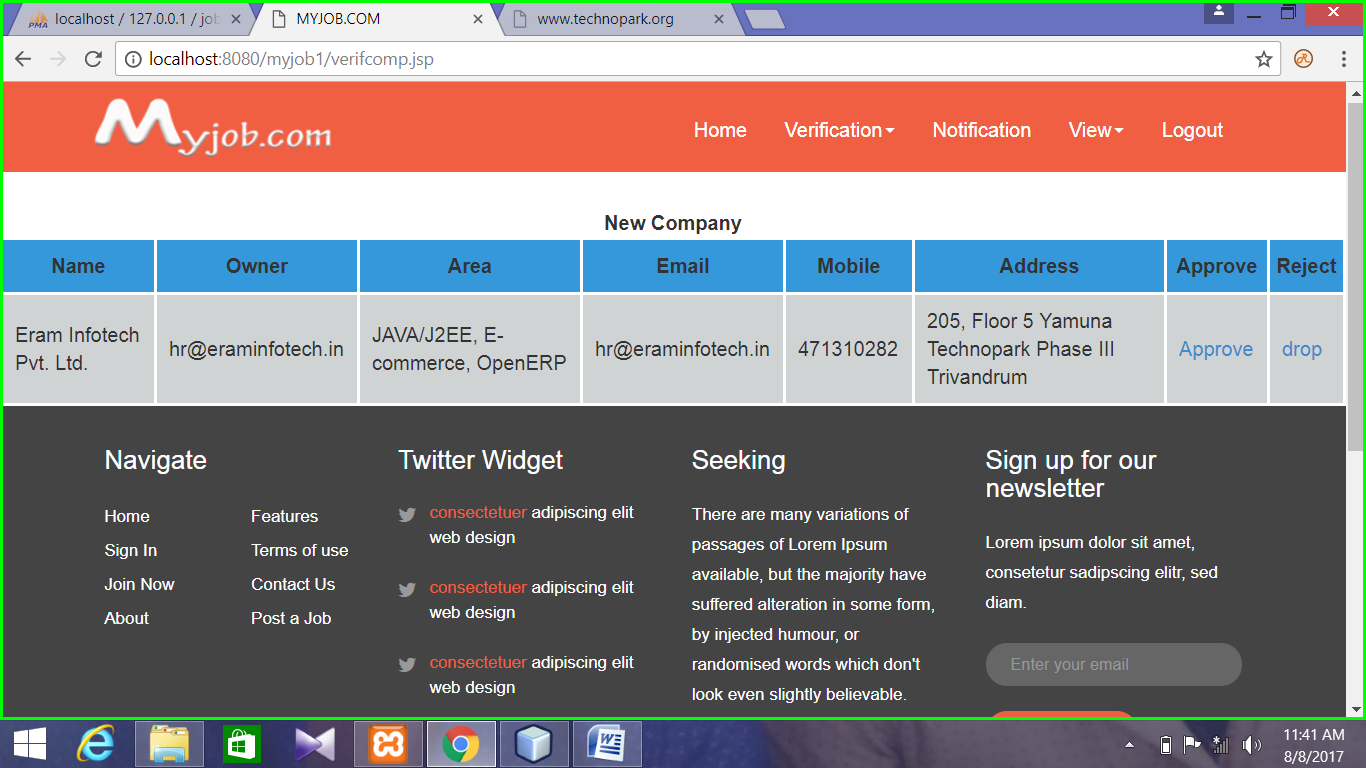
**Fig A.3 Company Registration**

****

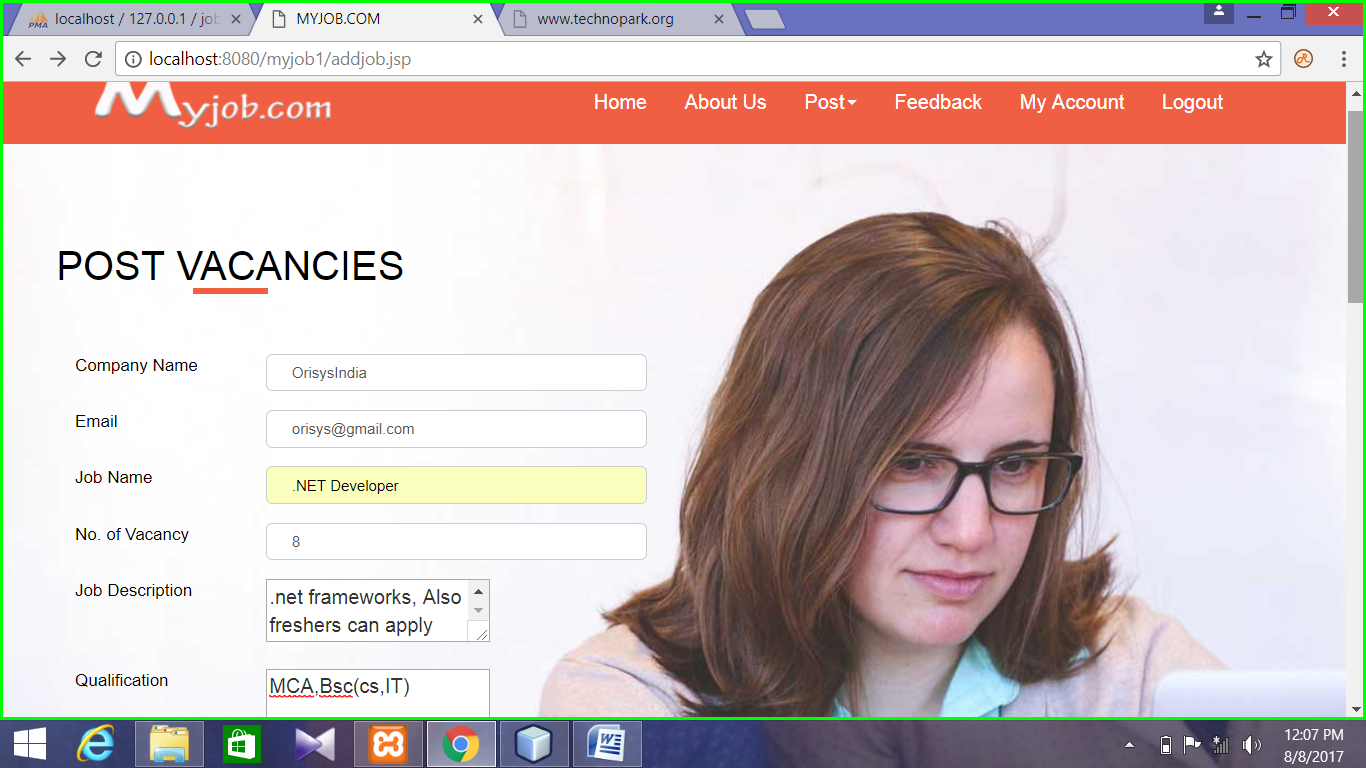
**FigA.4 Verification of Candidates**



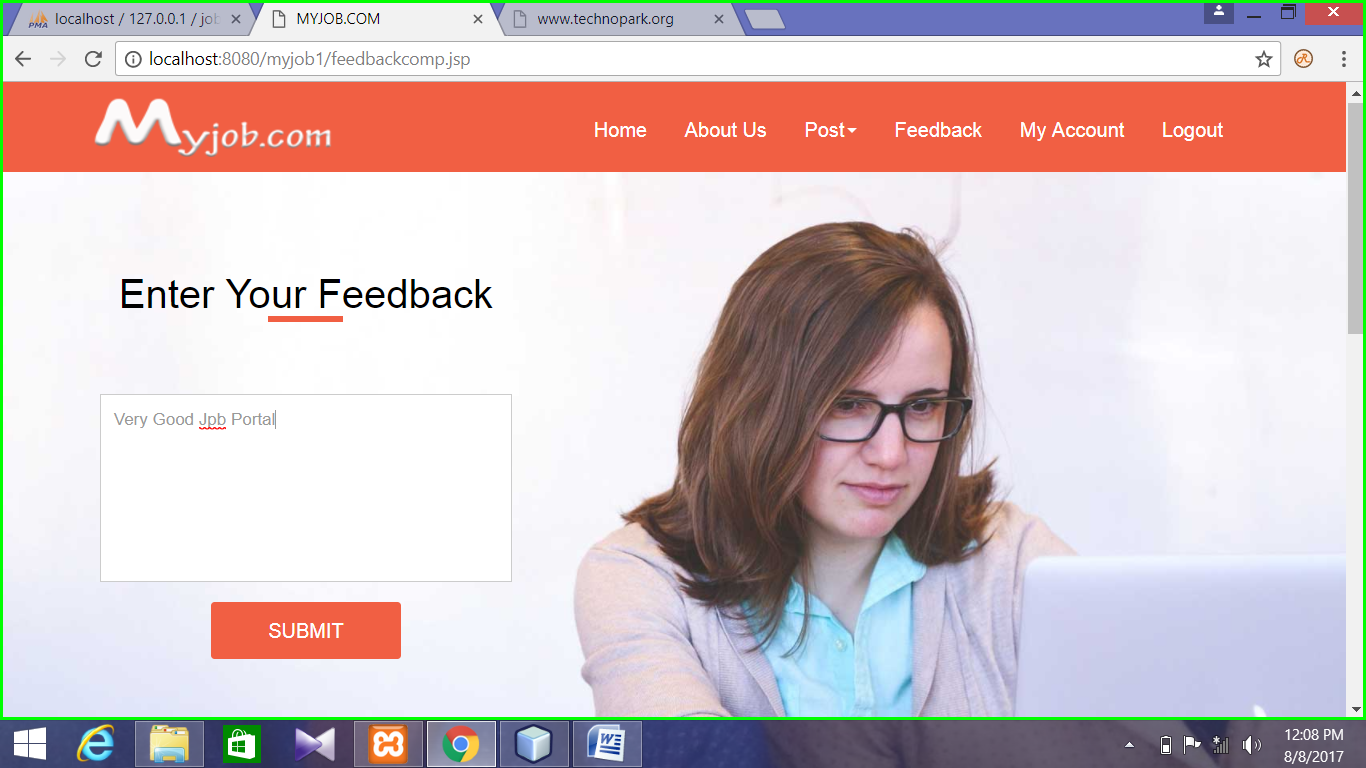
**FigA.5 Verification of Colleges**

****

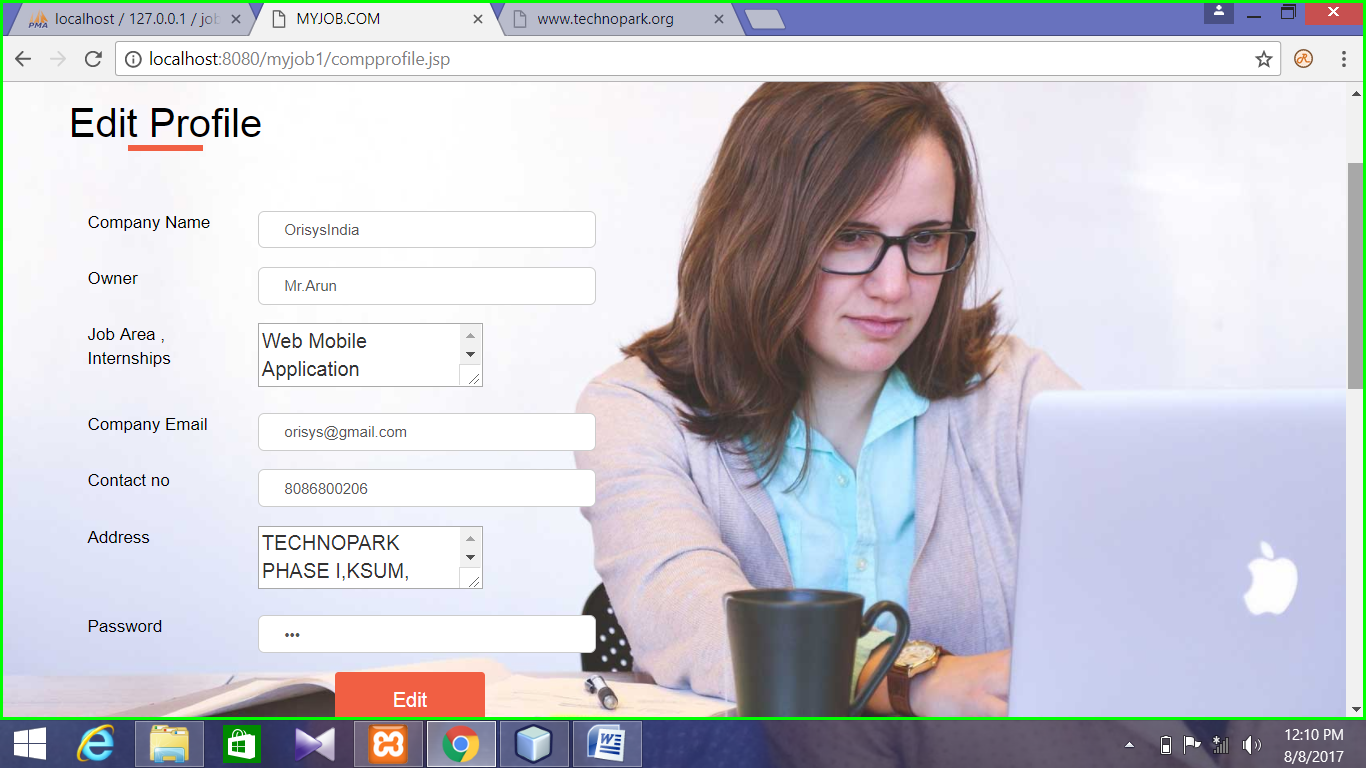
**FigA.5 Verification of Companies**



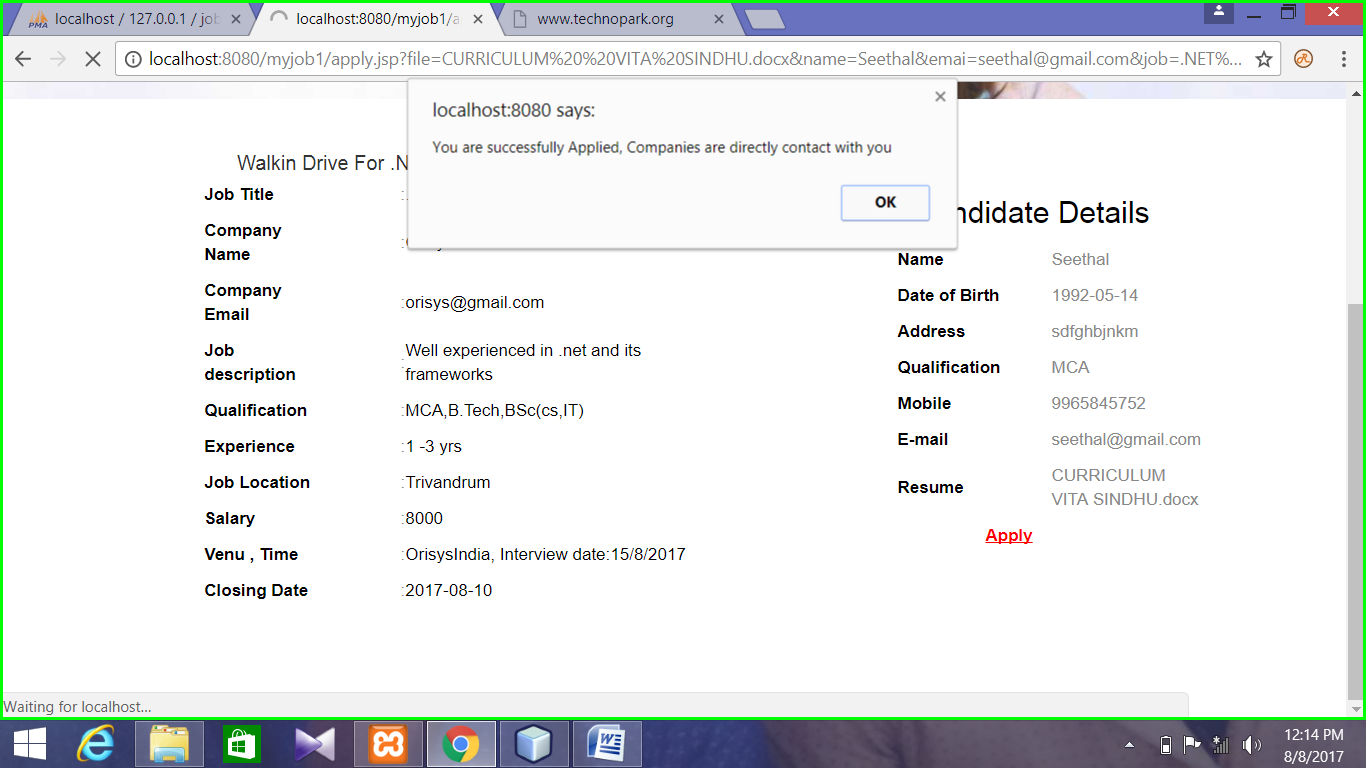
**FigA.6 Post new Vacancies**

****

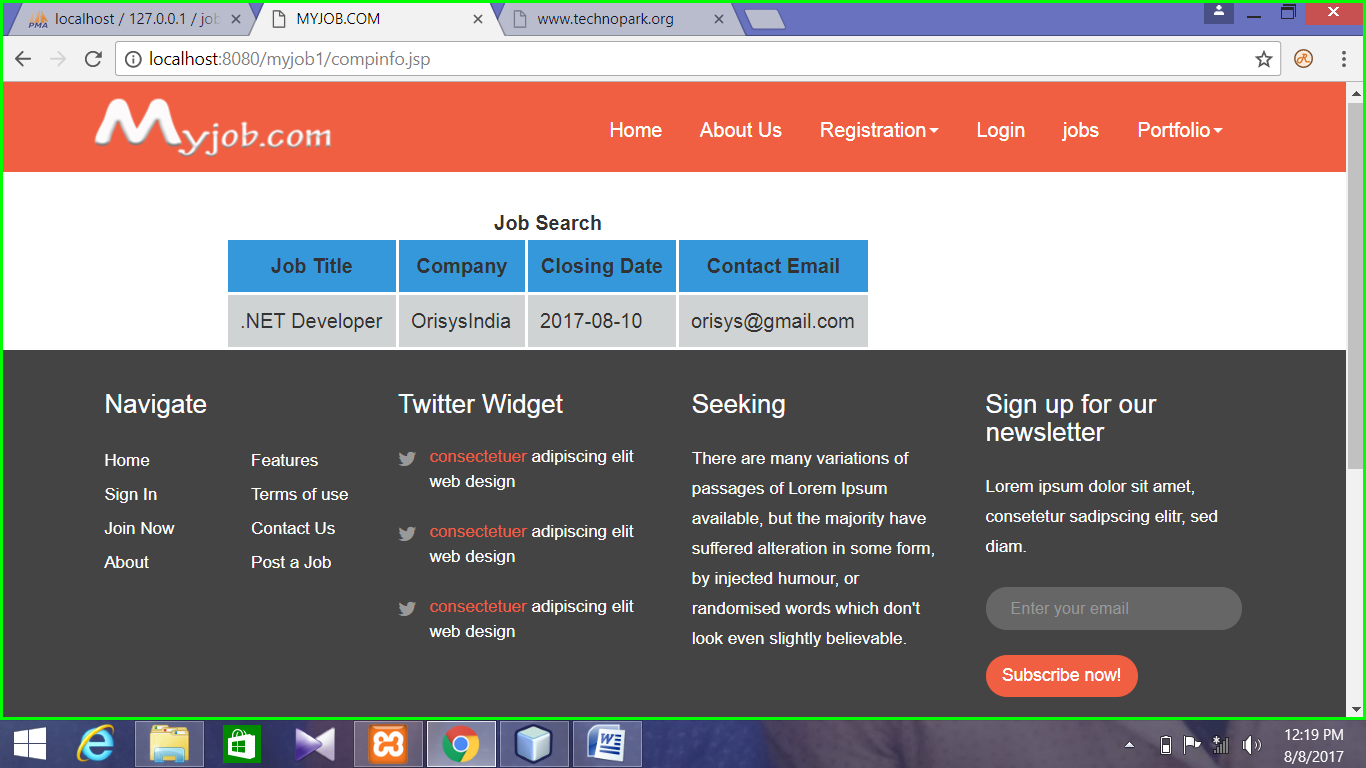
**FigA.7 Send Feedback Message**

****

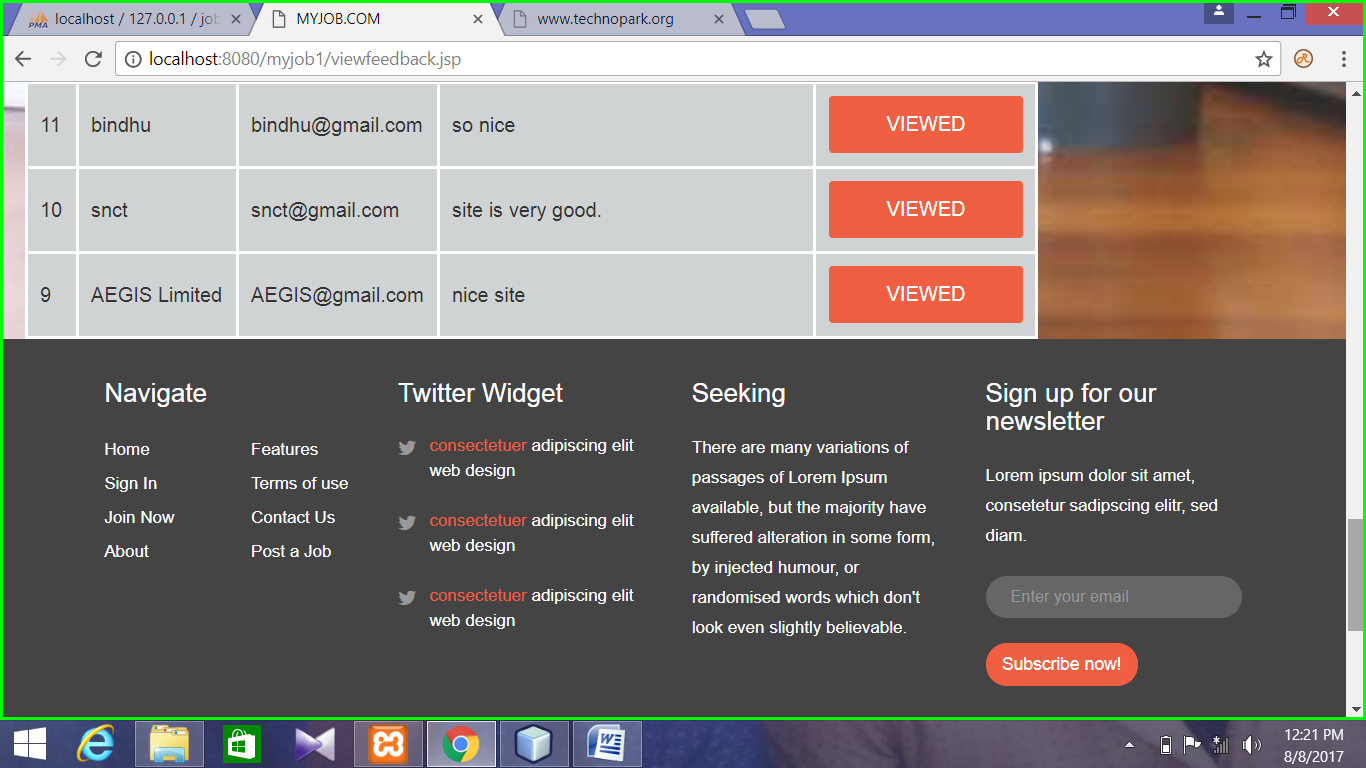
**FigA.8 Edit Company Profile**

****

**FigA.9 Apply Job**

****

**FigA.10 Posted Vacancy Information**

****

**FigA.11 View Feedbacks**

**APPENDIX B – SAMPLE CODE**

HomePage

<!DOCTYPE HTML>

<html>

<head>

<title>Myjob.com</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="Seeking Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,

Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />

<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>

<link href="css/bootstrap-3.1.1.min.css" rel='stylesheet' type='text/css' />

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<!-- Custom Theme files -->

<link href="css/style.css" rel='stylesheet' type='text/css' />

<link href='//fonts.googleapis.com/css?family=Roboto:100,200,300,400,500,600,700,800,900' rel='stylesheet' type='text/css'>

<!----font-Awesome----->

<link href="css/font-awesome.css" rel="stylesheet">

<!----font-Awesome----->

</head>

<body>

<nav class="navbar navbar-default" role="navigation">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="index.jsp"><img src="images/myjobnew.png" alt=""/></a>

</div>

<!--/.navbar-header-->

<div class="navbar-collapse collapse" id="bs-example-navbar-collapse-1" style="height: 1px;">

<ul class="nav navbar-nav">

<!-- <li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Jobs<b class="caret"></b></a>

<ul class="dropdown-menu">

<li><a href="location.html">Contract Jobs</a></li>

<li><a href="location.html">Walkin Jobs</a></li>

<li><a href="location.html">Jobs by Location</a></li>

<li><a href="location.html">Jobs by Function</a></li>

<li><a href="location.html">Jobs by Industry</a></li>

<li><a href="location.html">Jobs by Company</a></li>

</ul>

</li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Services<b class="caret"></b></a>

<ul class="dropdown-menu multi-column columns-3">

<div class="row">

<div class="col-sm-4">

<ul class="multi-column-dropdown">

<li><a href="services.html">Action</a></li>

<li><a href="services.html">Another action</a></li>

<li><a href="services.html">Something else here</a></li>

<li class="divider"></li>

<li><a href="services.html">Separated link</a></li>

<li class="divider"></li>

<li><a href="services.html">One more separated link</a></li>

</ul>

</div>

<div class="col-sm-4">

<ul class="multi-column-dropdown">

<li><a href="services.html">Action</a></li>

<li><a href="services.html">Another action</a></li>

<li><a href="services.html">Something else here</a></li>

<li class="divider"></li>

<li><a href="services.html">Separated link</a></li>

<li class="divider"></li>

<li><a href="services.html">One more separated link</a></li>

</ul>

</div>

<div class="col-sm-4">

<ul class="multi-column-dropdown">

<li><a href="services.html">Action</a></li>

<li><a href="services.html">Another action</a></li>

<li><a href="services.html">Something else here</a></li>

<li class="divider"></li>

<li><a href="services.html">Separated link</a></li>

<li class="divider"></li>

<li><a href="services.html">One more separated link</a></li>

</ul>

</div>

</div>

</ul>

</li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Recruiters<b class="caret"></b></a>

<ul class="dropdown-menu">

<li><a href="login.html">Recruiter Updates</a></li>

<li><a href="recruiters.html">Recruiters you are following</a></li>

<li><a href="codes.html">Shortcodes</a></li>

</ul>

</li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">More<b class="caret"></b></a>

<ul class="dropdown-menu">

<li><a href="jobs.html">Walk-ins</a></li>

<li><a href="jobs.html">Bpo Jobs</a></li>

<li><a href="jobs.html">Teaching Jobs</a></li>

<li><a href="jobs.html">Diploma Jobs</a></li>

<li><a href="jobs.html">Tech Support</a></li>

<li><a href="jobs.html">Finance Jobs</a></li>

<li><a href="jobs.html">Part time Jobs</a></li>

<li><a href="jobs.html">Health Care</a></li>

<li><a href="jobs.html">Hospitality</a></li>

<li><a href="jobs.html">Internships</a></li>

<li><a href="jobs.html">Research Jobs</a></li>

<li><a href="jobs.html">Defence Jobs</a></li>

</ul>

</li> -->

<li><a href="index.jsp">Home</a></li>

<li><a href="#">About Us</a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Registration<b class="caret"></b></a>

<ul class="dropdown-menu">

<li><a href="candireg.jsp">Candidate</a></li>

<li><a href="colgreg.jsp">College</a></li>

<li><a href="compreg.jsp">company</a></li>

</ul>

</li>

<li><a href="login1.jsp">Login</a></li>

<li><a href="compinfo.jsp">Jobs</a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Portfolio<b class="caret"></b></a>

<ul class="dropdown-menu">

<li><a href="location.html">Gallary</a></li>

</ul>

</li>

</ul>

</div>

<div class="clearfix"> </div>

</div>

<!--/.navbar-collapse-->

</nav>

<div class="banner">

<div class="container">

<div id="search\_wrapper">

<div id="search\_form" class="clearfix">

<h1>Start your job search</h1>

<p>

<input type="text" class="text" placeholder=" " value="Enter Keyword(s)" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Enter Keyword(s)';}">

<input type="text" class="text" placeholder=" " value="Location" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Location';}">

<label class="btn2 btn-2 btn2-1b"><input type="submit" value="Find Jobs"></label>

</p>

<!-- <h2 class="title">top Countries &amp; searches</h2>-->

</div>

<!-- <div id="city\_1" class="clearfix">

<ul class="orange">

<li>

<a href="#">Japan</a>

</li>

<li>

<a href="#" title="Kenya Jobs">Kenya</a>

</li>

<li>

<a href="#" title="Australia Jobs">Australia</a>

</li>

<li>

<a href="#" title="Poland Jobs">Poland</a>

</li>

<li>

<a href="#" title="Oman Jobs">Oman</a>

</li>

<li>

<a href="#" title="Pakistan jobs">Pakistan</a>

</li>

<li>

<a href="#" title="Saudi Arbia Jobs">Saudi Arbia</a>

</li>

<li>

<a href="#" title="Doha Jobs">Doha</a>

</li>

</ul>

<ul class="orange">

<li>

<a href="#" title="Russia Jobs">Russia</a>

</li>

<li>

<a href="#" title="Germany Jobs">Germany</a>

</li>

<li>

<a href="#" title="Canada Jobs">Canada</a>

</li>

<li>

<a href="#" title="Hong Kong Jobs">Hong Kong</a>

</li>

<li>

<a href="#" title="Spain Jobs">Spain</a>

</li>

<li>

<a href="#" title="Thailand Jobs">Thailand</a>

</li>

<li>

<a href="#" title="Norway Jobs">Norway</a>

</li>

<li>

<a href="#" title="Srilanka Jobs">Srilanka</a>

</li>

</ul>

<ul class="blue">

<li>

<a href="#" title="Information Technology">Mexico</a>

</li>

<li>

<a href="#" title="Fresher">Fresher</a>

</li>

<li>

<a href="#" title="Engineering">Engineering</a>

</li>

<li>

<a href="#" title="Finance Jobs">Finance</a>

</li>

<li>

<a href="#" title="Teaching">Teaching</a>

</li>

<li>

<a href="#" title="Software">Software</a>

</li>

<li>

<a href="#" title="Accounting">Accounting</a>

</li>

<li>

<a href="#" title="Data Entry">Data Entry</a>

</li>

</ul>

<ul class="blue">

<li>

<a href="#" title="Marketing Jobs">Marketing </a>

</li>

<li>

<a href="#" title="Freelancer">Freelancer</a>

</li>

<li>

<a href="#" title="Internet Jobs">Internet</a>

</li>

<li>

<a href="#" title="Sales">Sales</a>

</li>

<li>

<a href="#" title="Legal">Legal</a>

</li>

<li>

<a href="#" title="HR">HR</a>

</li>

<li>

<a href="#" title="MBA">MBA</a>

</li>

<li>

<a href="#" title="Pharma">Pharma</a>

</li>

</ul>

</div>-->

</div>

</div>

</div>

<div class="container">

<div class="grid\_1">

<h3>Featured Employers</h3>

<ul id="flexiselDemo3">

<li><img src="images/c1.gif" class="img-responsive" /></li>

<li><img src="images/c2.gif" class="img-responsive" /></li>

<li><img src="images/c3.gif" class="img-responsive" /></li>

<li><img src="images/c4.gif" class="img-responsive" /></li>

<li><img src="images/c5.gif" class="img-responsive" /></li>

<li><img src="images/c6.gif" class="img-responsive" /></li>

</ul>

<script type="text/javascript">

$(window).load(function() {

$("#flexiselDemo3").flexisel({

visibleItems: 6,

animationSpeed: 1000,

autoPlay:false,

autoPlaySpeed: 3000,

pauseOnHover: true,

enableResponsiveBreakpoints: true,

responsiveBreakpoints: {

portrait: {

changePoint:480,

visibleItems: 1

},

landscape: {

changePoint:640,

visibleItems: 2

},

tablet: {

changePoint:768,

visibleItems: 3

}

}

});

});

</script>

<script type="text/javascript" src="js/jquery.flexisel.js"></script>

</div>

<div class="single">

<div class="col-md-4">

<div class="col\_3">

<h3>Todays Jobs</h3>

<ul class="list\_1">

<li><a href="#">Department of Health - Western Australia</a></li>

<li><a href="#">Australian Nursing Agency currently require experiences</a></li>

<li><a href="#">Russia Nursing Agency currently require experiences</a></li>

<li><a href="#">The Government of Western Saudi Arbia</a></li>

<li><a href="#">Department of Health - Western Australia</a></li>

<li><a href="#">Australian Nursing Agency currently require experiences</a></li>

<li><a href="#">Russia Nursing Agency currently require experiences</a></li>

<li><a href="#">The Scientific Publishing Services in Saudi Arbia</a></li>

<li><a href="#">BPO Private Limited in Canada</a></li>

<li><a href="#">Executive Tracks Associates in Pakistan</a></li>

<li><a href="#">Pyramid IT Consulting Pvt. Ltd. in Pakistan</a></li>

</ul>

</div>

<div class="col\_3">

<h3>Jobs by Category</h3>

<ul class="list\_2">

<li><a href="#">Railway Recruitment</a></li>

<li><a href="#">Air Force Jobs</a></li>

<li><a href="#">Police Jobs</a></li>

<li><a href="#">Intelligence Bureau Jobs</a></li>

<li><a href="#">Army Jobs</a></li>

<li><a href="#">Navy Jobs</a></li>

<li><a href="#">BSNL Jobs</a></li>

<li><a href="#">Software Jobs</a></li>

<li><a href="#">Research Jobs</a></li>

</ul>

</div>

<div class="widget">

<h3>Take The Seeking Poll!</h3>

<div class="widget-content">

<div class="seeking-answer">

<span class="seeking-answer-group">

<span class="seeking-answer-input">

<input class="seeking-radiobutton" type="radio">

</span>

<label for="" class="seeking-input-label">

<span class="seeking-answer-span">Frequently</span>

</label>

</span>

<span class="seeking-answer-group">

<span class="seeking-answer-input">

<input class="seeking-radiobutton" type="radio">

</span>

<label for="" class="seeking-input-label">

<span class="seeking-answer-span">Interviewing</span>

</label>

</span>

<span class="seeking-answer-group">

<span class="seeking-answer-input">

<input class="seeking-radiobutton" type="radio">

</span>

<label for="" class="seeking-input-label">

<span class="seeking-answer-span">Leaving a familiar workplace</span>

</label>

</span>

<div class="seeking\_vote">

<a class="seeking-vote-button">Vote</a>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-8">

<div class="col\_1">

<div class="col-sm-4 row\_2">

<a href="single.html"><img src="images/a1.jpg" class="img-responsive" alt=""/></a>

</div>

<div class="col-sm-8 row\_1">

<h4><a href="single.html">It is a long established fact</a></h4>

<h6>SIt is a long <span class="dot">·</span> Jul. 31, 2015</h6>

<p>There are many variations of passages of Lorem Ipsum available, but the majority have suffered</p>

<div class="social">

<a class="btn\_1" href="#">

<i class="fa fa-facebook fb"></i>

<span class="share1 fb">Share</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-twitter tw"></i>

<span class="share1">Tweet</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-google-plus google"></i>

<span class="share1 google">Share</span>

</a>

</div>

</div>

<div class="clearfix"> </div>

</div>

<div class="col\_1">

<div class="col-sm-4 row\_2">

<a href="single.html"><img src="images/a2.jpg" class="img-responsive" alt=""/></a>

</div>

<div class="col-sm-8 row\_1">

<h4><a href="single.html">Lorem Ipsum is simply dummy</a></h4>

<h6>SIt is a long <span class="dot">·</span> Jul. 31, 2015</h6>

<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.</p>

<div class="social">

<a class="btn\_1" href="#">

<i class="fa fa-facebook fb"></i>

<span class="share1 fb">Share</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-twitter tw"></i>

<span class="share1">Tweet</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-google-plus google"></i>

<span class="share1 google">Share</span>

</a>

</div>

</div>

<div class="clearfix"> </div>

</div>

<div class="col\_1">

<div class="col-sm-4 row\_2">

<a href="single.html"><img src="images/a3.jpg" class="img-responsive" alt=""/></a>

</div>

<div class="col-sm-8 row\_1">

<h4><a href="single.html">There are many variations</a></h4>

<h6>SIt is a long <span class="dot">·</span> Jul. 31, 2015</h6>

<p>YBut I must explain to you how all this mistaken idea of denouncing pleasure.</p>

<div class="social">

<a class="btn\_1" href="#">

<i class="fa fa-facebook fb"></i>

<span class="share1 fb">Share</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-twitter tw"></i>

<span class="share1">Tweet</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-google-plus google"></i>

<span class="share1 google">Share</span>

</a>

</div>

</div>

<div class="clearfix"> </div>

</div>

<div class="col\_1">

<div class="col-sm-4 row\_2">

<a href="single.html"><img src="images/a4.jpg" class="img-responsive" alt=""/></a>

</div>

<div class="col-sm-8 row\_1">

<h4><a href="single.html">Contrary to popular belief</a></h4>

<h6>SIt is a long <span class="dot">·</span> Jul. 31, 2015</h6>

<p>At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti.</p>

<div class="social">

<a class="btn\_1" href="#">

<i class="fa fa-facebook fb"></i>

<span class="share1 fb">Share</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-twitter tw"></i>

<span class="share1">Tweet</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-google-plus google"></i>

<span class="share1 google">Share</span>

</a>

</div>

</div>

<div class="clearfix"> </div>

</div>

<div class="col\_1">

<div class="col-sm-4 row\_2">

<a href="single.html"><img src="images/a5.jpg" class="img-responsive" alt=""/></a>

</div>

<div class="col-sm-8 row\_1">

<h4><a href="single.html">At vero eos et accusamus</a></h4>

<h6>SIt is a long <span class="dot">·</span> Jul. 31, 2015</h6>

<p>On the other hand, we denounce with righteous indignation and dislike men.</p>

<div class="social">

<a class="btn\_1" href="#">

<i class="fa fa-facebook fb"></i>

<span class="share1 fb">Share</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-twitter tw"></i>

<span class="share1">Tweet</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-google-plus google"></i>

<span class="share1 google">Share</span>

</a>

</div>

</div>

<div class="clearfix"> </div>

</div>

<div class="col\_2">

<div class="col-sm-4 row\_2">

<a href="single.html"><img src="images/a6.jpg" class="img-responsive" alt=""/></a>

</div>

<div class="col-sm-8 row\_1">

<h4><a href="single.html">On the other hand</a></h4>

<h6>SIt is a long <span class="dot">·</span> Jul. 31, 2015</h6>

<p>Contrary to popular belief, Lorem Ipsum is not simply random text.</p>

<div class="social">

<a class="btn\_1" href="#">

<i class="fa fa-facebook fb"></i>

<span class="share1 fb">Share</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-twitter tw"></i>

<span class="share1">Tweet</span>

</a>

<a class="btn\_1" href="#">

<i class="fa fa-google-plus google"></i>

<span class="share1 google">Share</span>

</a>

</div>

</div>

<div class="clearfix"> </div>

</div>

</div>

<div class="clearfix"> </div>

</div>

</div>

<div class="footer">

<div class="container">

<div class="col-md-3 grid\_3">

<h4>Navigate</h4>

<ul class="f\_list f\_list1">

<li><a href="index.html">Home</a></li>

<li><a href="login.html">Sign In</a></li>

<li><a href="login.html">Join Now</a></li>

<li><a href="about.html">About</a></li>

</ul>

<ul class="f\_list">

<li><a href="features.html">Features</a></li>

<li><a href="terms.html">Terms of use</a></li>

<li><a href="contact.html">Contact Us</a></li>

<li><a href="jobs.html">Post a Job</a></li>

</ul>

<div class="clearfix"> </div>

</div>

<div class="col-md-3 grid\_3">

<h4>Twitter Widget</h4>

<div class="footer-list">

<ul>

<li><i class="fa fa-twitter tw1"> </i><p><span class="yellow"><a href="#">consectetuer</a></span> adipiscing elit web design</p></li>

<li><i class="fa fa-twitter tw1"> </i><p><span class="yellow"><a href="#">consectetuer</a></span> adipiscing elit web design</p></li>

<li><i class="fa fa-twitter tw1"> </i><p><span class="yellow"><a href="#">consectetuer</a></span> adipiscing elit web design</p></li>

</ul>

</div>

</div>

<div class="col-md-3 grid\_3">

<h4>Seeking</h4>

<p>There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. </p>

</div>

<div class="col-md-3 grid\_3">

<h4>Sign up for our newsletter</h4>

<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam.</p>

<form>

<input type="text" class="form-control" placeholder="Enter your email">

<button type="button" class="btn red">Subscribe now!</button>

</form>

</div>

<div class="clearfix"> </div>

</div>

</div>

<div class="footer\_bottom">

<div class="container">

<div class="col-sm-2">

<ul class="f\_list2">

<li><a href="jobs.html">Russia Jobs</a></li>

<li><a href="jobs.html">Australia Jobs</a></li>

<li><a href="jobs.html">Srilanka Jobs</a></li>

<li><a href="jobs.html">Poland Jobs</a></li>

</ul>

</div>

<div class="col-sm-2">

<ul class="f\_list2">

<li><a href="jobs.html">New Zealand Jobs</a></li>

<li><a href="jobs.html">Pakistan Jobs</a></li>

<li><a href="jobs.html">Srilanka Jobs</a></li>

<li><a href="jobs.html">Irland Jobs</a></li>

</ul>

</div>

<div class="col-sm-2">

<ul class="f\_list2">

<li><a href="jobs.html">Canada Jobs</a></li>

<li><a href="jobs.html">Germany Jobs</a></li>

<li><a href="jobs.html">China Jobs</a></li>

<li><a href="jobs.html">Nepal Jobs</a></li>

</ul>

</div>

<div class="col-sm-6 footer\_text">

</div>

<div class="clearfix"> </div>

<div class="copy">

</div>

</div>

</div>

</body>

</html>

**Candidate Registration**

Candireg.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ include file="header.jsp" %>

<script>

window.onload = function() {

document.getElementById("myText").focus();

};

function validateForm(whichForm) {

if (whichForm.t1.value == "") {

alert ("College Name cannot be blank.");

return false;

}

else if(whichForm.t2.value == "")

{

alert("Officer name cannot be blank.");

return false;

}

else if(whichForm.t3.value == "")

{

alert("Email cannot be blank.");

return false;

}

else if(whichForm.cou.value == "")

{

alert("course cannot be blank.");

return false;

}

else if(whichForm.add.value == "")

{

alert("Address cannot be blank.");

return false;

}

else if(whichForm.t5.value == "")

{

alert("Mobile cannot be blank.");

return false;

}

else if(whichForm.t6.value == "")

{

alert("Password cannot be blank.");

return false;

}

else {

alert('Data is inserted successfully');

}

}

</script>

<div class="banner">

<!--<div class="container">

<div id="search\_wrapper1">

<div id="search\_form" class="clearfix">

<h1>Start your job search</h1>

<p>

<input type="text" class="text" value="Enter Keyword(s)" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Enter Keyword(s)';}">

<input type="text" class="text" placeholder=" " value="Location" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Location';}">

<label class="btn2 btn-2 btn2-1b"><input type="submit" value="Find Jobs"></label>

</p>

</div>

</div>-->

<div class="container" style="margin-left: 0px;">

<div class="single" style="float: left;">

<div class="form-container" style="margin-left: -50px;">

<h2 style="float: left">Register Form</h2>

<form action="colgregpros.jsp" method="POST">

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="Name">College Name</label>

<div class="col-md-9">

<input type="text" name="t1" id="myText" class="form-control input-sm" size="60px"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="po">Placement Officer</label>

<div class="col-md-9">

<input type="text" name="t2" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="email">Email</label>

<div class="col-md-9">

<input type="text" name="t3" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="course">Course Details</label>

<div class="col-md-9">

<textarea name="cou">Course details....</textarea>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="address">Address</label>

<div class="col-md-9">

<textarea name="add">Address...</textarea>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="mobile">Mobile</label>

<div class="col-md-9">

<input type="text" maxlength="10" name="t5" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="pswd">Password</label>

<div class="col-md-9">

<input type="password" name="t6" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-actions floatRight">

<input type="submit" value="Register" onclick="return validateForm(this.form);" class="btn btn-primary btn-sm">

<input type="submit" value="Reset" class="btn btn-primary btn-sm">

</div>

</div>

</form>

</div>

</div>

</div>

</div>

<%@ include file="footer.jsp" %>

Candiregpros.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.sql.\*"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<%

String name=request.getParameter("name");

String email=request.getParameter("email");

//String mobile=request.getParameter("mobile");

String mobile=request.getParameter("mobile");

String pswd=request.getParameter("pass");

//out.println(mobile);

//out.println(name);

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/job","root","");

Statement stmt=con.createStatement();

// ResultSet rs;

// while(rs.next())

// {

int qry=stmt.executeUpdate("insert into cand\_reg(name,email,mobile,password)values('"+name+"','"+email+"','"+mobile+"','"+pswd+"')");

//out.println("Data is inserted successfully");

response.sendRedirect("candbox.jsp");

//out.println("<script>alert('Data is inserted successfully');window.location='candirgproces.jsp';</script>");

con.close();

//}

}

catch(Exception e)

{

out.println(e);

}

%>

</body>

</html>

Addjob.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.sql.\*,java.util.\*"%>

<%@ include file="headercomp.jsp" %>

<script>

window.onload = function() {

document.getElementById("myText").focus();

};

function validateForm(whichForm) {

if (whichForm.job.value == "") {

alert ("Job cannot be blank.");

return false;

}

else if(whichForm.vac.value == "")

{

alert("No.of Vaccancy cannot be blank.");

return false;

}

else if(whichForm.descri.value == "")

{

alert("Job Description cannot be blank.");

return false;

}

else if(whichForm.email.value == "")

{

alert("Email cannot be blank.");

return false;

}

else if(whichForm.pass.value == "")

{

alert("Password cannot be blank.");

return false;

}

else if(whichForm.add.value == "")

{

alert("Address cannot be blank.");

return false;

}

else if(whichForm.qual.value == "")

{

alert("Qualification cannot be blank.");

return false;

}

else if(whichForm.mark.value == "")

{

alert("Mark cannot be blank.");

return false;

}

else if(whichForm.key.value == "")

{

alert("Key Skills cannot be blank.");

return false;

}

else if(whichForm.photo.value == "")

{

alert("Photo must Upload.");

return false;

}

else if(whichForm.resume.value == "")

{

alert("Resume must Upload.");

return false;

}

else {

alert('Data is stored successfully');

}

}

</script>

<div class="banner">

<!--<div class="container">

<div id="search\_wrapper1">

<div id="search\_form" class="clearfix">

<h1>Start your job search</h1>

<p>

<input type="text" class="text" value="Enter Keyword(s)" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Enter Keyword(s)';}">

<input type="text" class="text" placeholder=" " value="Location" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Location';}">

<label class="btn2 btn-2 btn2-1b"><input type="submit" value="Find Jobs"></label>

</p>

</div>

</div>

</div> -->

<%

try

{

String email=session.getAttribute("cand").toString();

String nam="";

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/job","root","");

Statement stmt=con.createStatement();

ResultSet rst=stmt.executeQuery("select \* from comp\_reg where email='"+email+"' ");

while(rst.next())

{

nam=rst.getString("name");

// String email=rst.getString("email");

/\*

String mobile=rst.getString("mobile");

String pswd=rst.getString("password"); \*/

%>

<div class="container" style="margin-left: 0px;">

<div class="single" style="float: left;">

<div class="form-container" style="margin-left: -50px;">

<h2 style="float: left">POST VACANCIES</h2>

<form action="addjobpros.jsp" method="POST">

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="Name">Company Name</label>

<div class="col-md-9">

<input type="text" name="comname" maxlength="2000" size="60px" id="myText" value=<%out.print(nam);%> class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="Name">Email</label>

<div class="col-md-9">

<input type="text" name="email" size="60px" value=<%out.print(email);%> class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="Name">Job Name</label>

<div class="col-md-9">

<input type="text" name="job" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="vac">No. of Vacancy</label>

<div class="col-md-9">

<input type="text" name="vac" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="descr">Job Description</label>

<div class="col-md-9">

<textarea name="descri" maxlength="10000"> </textarea>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="qual">Qualification</label>

<div class="col-md-9">

<textarea name="qual" maxlength="10000"> </textarea>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="sexp">Experience</label>

<div class="col-md-9">

<textarea name="exp" maxlength="10000"> </textarea>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="job">Job Location</label>

<div class="col-md-9">

<textarea name="loc" maxlength="10000"></textarea>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="sal">Salary</label>

<div class="col-md-9">

<input type="text" name="sal" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="vt">Venu , Time</label>

<div class="col-md-9">

<input type="text" name="vt" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-group col-md-12">

<label class="col-md-3 control-lable" for="ld">Last Date</label>

<div class="col-md-9">

<input type="date" name="ld" class="form-control input-sm"/>

</div>

</div>

</div>

<div class="row">

<div class="form-actions floatRight">

<input type="submit" value="POST" class="btn btn-primary btn-sm">

</div>

</div>

</form>

</div>

</div>

</div>

<% }

}

catch(Exception e)

{

;

}

%>

</div>

<%@ include file="footer.jsp" %>

Addjobpros.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.sql.\*"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<%

String comp=request.getParameter("comname");

String email=request.getParameter("email");

String job=request.getParameter("job");

String vac=request.getParameter("vac");

String desc=request.getParameter("descri");

String qua=request.getParameter("qual");

String exp=request.getParameter("exp");

String loc=request.getParameter("loc");

String salary=request.getParameter("sal");

String vetim=request.getParameter("vt");

String time=request.getParameter("ld");

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/job","root","");

Statement stmt=con.createStatement();

String qry="insert into jobpool(company,email,title,no\_vac,descri,qualification,Experience,loca,salary,venu\_tim,date)values('"+comp+"','"+email+"','"+job+"','"+vac+"','"+desc+"','"+qua+"','"+exp+"','"+loc+"','"+salary+"','"+vetim+"','"+time+"')";

stmt.executeUpdate(qry);

out.println("<script>alert('Vacancy Posted Successfully');window.location='addjob.jsp';</script>");

// out.print("Data is inserted successfully");

//response.sendRedirect("addjob.jsp");

con.close();

}

catch(Exception e)

{

out.println(e);

}

%></body></html>

**REFERENCE**

Book Reference

1. Rajib Mall, Fundamentals of Software Engineering, Eastern Economy Edition, Third Edition, Year 2009.

2. Roger S Pressman, Software Engineering , TataMcgrawHill, Fifth Edition, Year 2004.

3. Jiawei Han & Micheline Kamber, Data Mining Concepts And Techniques, Second Edition

4. V. Rajaraman, Analysis and Design of Information Systems, Eastern Economy Edition, Third Edition, Year 2011.

5. Margaret H Dunham, Data Mining: Introductory and Advanced Topics, Pearson Education, First Edition, Year 2008.

Web Reference

1.www.javatpoints.com

2.www.Tutorials Point.com

3.www.Stack Overflow.com

4.www.w3school.com