**SYNOPSIS INDEX**

|  |  |  |
| --- | --- | --- |
| ***Sl No.*** | ***Subject*** | ***Page No.*** |
| ***1.*** | Synopsis Title | 2 |
| ***2.*** | Introduction | 3-4 |
| ***3.*** | Objective | 5 |
| ***4.*** | Category | 6 |
| ***5.*** | Tools/Platform | 7 |
| ***6.*** | Data Flow Diagram | 8-13 |
| ***7.*** | ER-Diagram | 14 |
| ***8.*** | Number of Modules | 15-16 |
| ***9.*** | Data Structure | 17-19 |
| ***10.*** | Outputs/Reports | 20 |
| ***11.*** | Validation | 21 |
| ***12.*** | Future Scope | 22 |

### Build Carrier Placements

**ABSTRACT**:In current competitive scenario every business establishment needs quality processes to increase their efficiency as well as improve their productivity. It is of vital importance that manual, time consuming & monotonous operations be automated so as to streamline the working of an organization. It is keeping in mind this business philosophy (mantra) that we propose an online automated solution for **Build Carrier Placements.** Our system will deal with all the aspects of any Consultant and Placement Service provider as well as report generation.

**PROPOSED SYSTEM: -**

1. This is a web application intended to provide automated solution to the Job Seekers, Employers along with the Consultant and Placements Service Provider.

**(2)** As this is completely web based, the Job Seeker can search job at any time as

well the Employer can also post jobs at any time on this system.

**(3)** The system will work in three ways:-

* + - * + From job seeker’s End
        + From employer’s End
        + From Admin End.

In the first way the job seeker can log on to the system via Internet. In this case the job seeker can enter in the system after the authentication of the seeker-id. After the authentication he/she could able to apply for any job that suits their profile. They can modify their profile.

While in the second way, the Employer can also logged in to the system with their proper login credentials. Once they get logged in they can post new jobs in **Build Carrier,** they can Edit/Modify the earlier posted job. Employer can even delete any jobs from this application.

**(4)** When the job seeker/Employer create their profile in this system, the system will allot job-seeker-id/employer-id respectively to the. When the employer posts any job to this system, the system will generate a new job-id for that particular job.

**(5)** Beside job seeker and employer the system has one more important entity this is Admin.

The main responsibility of **Admin** in this system is that once the job is posted by the employer it will remain in pending status until the admin will not accept this job.

Once the **Admin** will accept the job, it will automatically get posted to the system.

Various responsibilities of **Admin** are listed below:-

* Grant permission to newly arrived job.
* Add/Remove details of job-seeker/employers.
* Edit the priority of employer.

**(6)** Status of any job seeker/employer/job can easily be determined by using their respective ids.

**(7)** The system provides various online reports for the concerned senior officers of

**Build Carrier Placements** that helps in decision-making.

**(8)** Nobody have right to delete any records excepts **Admin** or without **Admin** permissions.

The objectives of developing the **Build Carrier Placements** are as given below: -

**(1)** To store data of all the job seeker/employer who are registered with this organization.

**(2)** It will work as a bridge connection in between Employers and Job Seekers in efficient way.

In essence it means that the Employer will not have to search the resources for long time and vice versa for the job seekers.

**(3)** To facilitate easy and efficient retrieval of data as and when required for study by the senior officials in this organization.

**(4)** To generate and print out the various reports. The online MIS reports are generated for the concerned senior officers of **Build Carrier Placements** to monitor the progress of organization in concern of their sponsors and income from them.

**(5)** The **tracking** of previous job seeker/employers feedbacks can also make the company (organization) aware of their changing needs.

**(6)** Data security is maintained to relatively high level by implementing it at Database level, so as to ensure that only authorized users have access to confidential client information.

* MS Office 2000 Professional or Higher
* .NET Framework 2.0 or Higher
* SQL Server 2000 or Higher
* VB Script
* HTML
* JavaScript Enabled
* Internet Connection

## **Hardware Requirements:**

* Pentium III 1.5 GHz and Above
* 258 MB SDRAM or More
* 40 GB HDD

## **Software Requirements:**

* Windows NT Server
* MS office 2000 Professional
* Internet Information System (IIS)
* Active Server Pages (ASP.NET)
* SQL Server 2000
* VB Script
* HTML

###### Dataflow Diagram (*DFDs*)

Admin

Job-Seeker

Employer

Send Information regarding Job

Submit Emp\_Id

Seeker\_Id & PWD & PWD

Job-Apply Post Jobs

Query Query

Response

Various

Reports

Submit

Admin\_Id

& PWD Reports

**(CONTEXT LEVEL DFD)**

**Abbreviation used in all DFDs**

**Emp\_Id :** Employer Id

**PWD :** Password

Send Information

Employer

Job-Seeker

Profile Valid

Create Job-Id

Store Date Invalid Invalid & Time

Emp-id

Logout & Pwd D1 Master database

Store Recods

Valid cust

Valid user Submitted By

Seeker

Login

Invalid

Admin-id Accepts

Valid & Pwd

Seeker-Id

Admin

Valid

Admin

Store

Details Generate

Check

Closes

Delete Records

D1Master Database

Closed

Check from

Print

Details of Customers & Problems

Various report

**(FIRST LEVEL DFD)**

**(1) User Login Process**

Valid Job-Seeker

Valid Store

Employer

D1 Master Database

Store

Job-Id.

Admin

**(2) Job Posting Process**

Valid Reported to

Admin

Employer

Posting

Pending Take Details

of Job-Id & D1 Master Database

Emp\_Id

Job

Update

**(3) Job Apply Process**

Job Seeker Search Take Job Details

Send Mail D1 Master Database

Update Status

Employers

**(4) Mailbox/Delete Job process: -**

Post and send Information

Checking status

Not approved/send

Job

Status

Detail

Send

Mail

To

Employer

Job Seeker

To

Store Data

D1 Master Database

Store not approved Details

Store Modified Details

Store Job posting Details

**(5) Report Generation**

Report 1

Report 2

Report 3

Valid

Admin**/**

Employers/ Report 4

Job Seeker

Report 5

Take various

Details

D1 Master Database

Report 1: Report on Job Submitted by Employers

Report 2: Report on Employer (for Admin)

Report 3: Report on Job Seeker (for Admin)

Report 4: Report on job applied from Job Seeker

Report 5: Report on Current Status of Job posted so far.

Mail

Register

Job-Seeker

Build Carrier

Admin

Employer

Mailbox

Apply

Request

Check Mailbox

Job

Post

Delete

1. **Create Profile Module:-**

This Module touches the two entities in this system:-

* Job Seeker
* Employer

Job Seeker can create their profile on this system and further they can apply for any job as per their profile.

Creating profile for the job seeker is important as because the job seeker can only apply for any job after get logged in to this system.

Almost same like job seeker entity, this module will work same for the employer entity also. Employer can only post any job to this system when they will have their own Employer ID.

**(2) User Login Module:-** This module is for login by the various users. The User can be a Job Seeker, Employer or Admin. This module is integrated with Restriction of unauthorized access module. Each user can only access those data, which are required for his/her work.

**(2) Mailbox Module:-** Again this Module will touches the two modules:-

* Job Seeker
* Employer

Job Seeker/Employer both will get facility of mailbox in their account and in case of job seeker they will receive mails if any new job posted on this system which matches their profile.

In case of Employer, they will receive new mail when any job seeker apply for any job posted by the respective employer. In return the employer can send a reply mail also to the applied job seeker profile.

**(3) Job Search Module:-** This module is for the Job Seeker. The Job Seeker can search jobs posted on this system according to job category and experience.

1. **Job Post Module:-** This module is for Employer. The Employer can post job on this system and detailed requirements for that job. They further can modify this posted job details based on the job-id.

1. **Admin Panel Module:-** This module is for Admin. The admin continually check the various status of Job-seeker/Employer/Jobs on this system. Admin will grant the posting of any new job.

Admin will have the rights to modify/delete any job posted on this system.

**(6)Restriction of Unauthorized access Module:-** Unauthorized Customers/Users are restricted to access in the system.

**(7)Change of privileges Module:-** Privileges will be granted on the basis of groups created by the Admin. They can also be revoked in case the granted privileges are misused.

**(8)Report generation Module:-** This module generates various reports.

1. **JobSeeker\_Profile: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Seeker\_Id | NVARCHAR | 20 | Identification of Job Seeker |
| First\_Name | NVARCHAR | 20 | Job Seeker First Name |
| Last\_Name | NVARCHAR | 20 | Job Seeker Last Name |
| Address | NVARCHAR | 500 | Address of the Job Seeker |
| City | NVARCHAR | 20 | Job Seeker’s City |
| State | NVARCHAR | 20 | Job Seeker’s State |
| Contact\_No | BIGINT | 15 | Job Seeker’s Contact# |
| Email | NVARCHAR | 100 | Job Seeker’s Email\_Id |
| DoB | NVARCHAR | 20 | Job Seeker’s Date of Birth |
| Edu\_Qualification | NVARCHAR | 1000 | Educational Qualification |
| Pro\_Qualification | NVARCHAR | 1000 | Professional Qualification |
| OtherQualification | NVARCHAR | 1000 | Other Qualification |
| Experience | NVARCHAR | 20 | Work Experience of Seeker |
| Job\_Category | NVARCHAR | 1000 | Category of Job looking for |
| Password | NVARCHAR | 100 | Password of the Seeker |

1. **Employer: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **SIZE** | **Description** |
| Employer\_Id | NVARCHAR | 20 | Identification of Employer |
| E\_Name | NVARCHAR | 50 | Employer Name |
| Com\_Name | NVARCHAR | 50 | Employer’s Company Name |
| Com\_Address | NVARCHAR | 100 | Employer’s Company Address |
| City | NVARCHAR | 100 | Employer’s City |
| State | NVARCHAR | 100 | Employer’s State |
| Contact\_No | BIGINT | 15 | Contact Detauls for Employer |
| Com\_Website | NVARCHAR | 100 | Company Website |
| Email\_Id | NVARCHAR | 100 | Employer Email id |

**3. Job\_Category: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Category\_ID | NVARCHAR | 10 | Job Category ID |
| Job\_ Category | NVARCHAR | 50 | Job Category |
| Status | BIT | 1 | Status of Job |

**4. Job\_Details:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Row\_Id | BIGINT | 10 | Identify Row in Table |
| Job\_Id | NVARCHAR | 25 | Job\_Id from New Job |
| Job\_Title | NVARCHAR | 100 | Job Title |
| Job\_Desc | NVARCHAR | 1000 | Description of Job |
| Emp\_Id | NVARCHAR | 25 | Job Submitted by which Emp. |
| Job\_Category | NVARCHAR | 1000 | Category of the Job |
| City | NVARCHAR | 100 | City |
| State | NVARCHAR | 100 | State |
| Exp\_Req | NVARCHAR | 25 | How much Experience required |
| No\_Of\_Days | INT | 4 | For how many days job will be there in A1 Jobs site |
| Posting\_Date | NVARCHAR | 20 | On Which date job is posted |
| Job\_Status | BIT | 1 | Define the Status of Job |

**5. Job\_Transaction: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Job\_Id | NVARCHAR | 20 | Identify Job in Process |
| Emp\_Id | NVARCHAR | 20 | Posted by which Emp. |
| Seeker\_Id | NVARCHAR | 20 | Applied by which Seeker |
| Emp\_Status | BIT | 1 | Status of Emp. |
| Seeker\_Status | BIT | 1 | Status of Seeker |

**6. Seeker\_Inbox: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Seeker\_Id | NVARCHAR | 20 | Identify the Job Seeker |
| Msg\_Id | NVARCHAR | 20 | Message Id for the Message |
| From | NVARCHAR | 50 | Message send from Which Company |
| Msg\_Subject | NVARCHAR | 1000 | Subject of the Message |
| Msg\_Desc | NVARCHAR | 1000 | Description of Message |
| Msg\_Status | BIT | 1 | Identification Status of Message |
| Msg\_Date | DATETIME | 20 | Date of Message |

7. Seeker\_SentMail: -

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Seeker\_Id | NVARCHAR | 20 | Identification Job Seeker |
| Msg\_Id | NVARCHAR | 20 | Identification Message for Seeker |
| Msg\_Subject | NVARCHAR | 1000 | Subject of Message |
| Msg\_Desc | NVARCHAR | 1000 | Description of Message |
| Msg\_Date | NVARCHAR | 20 | Date of Msg |

8. Seeker\_Trash: -

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Seeker\_Id | NVARCHAR | 20 | Identification Job Seeker |
| Msg\_Id | NVARCHAR | 20 | Identification Message for Seeker |
| Msg\_Subject | NVARCHAR | 1000 | Subject of Message |
| Msg\_Desc | NVARCHAR | 1000 | Description of Message |
| Msg\_Date | NVARCHAR | 20 | Date of Msg |

9. Employer\_Inbox: -

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Description** |
| Emp\_Id | NVARCHAR | 20 | Identification Employer |
| Job\_Id | NVARCHAR | 20 | Identification Job |
| Mail\_Id | NVARCHAR | 20 | Identify Message |
| Seeker\_Id | NVARCHAR | 20 | Identification Job Seeker |

**(1) Report on Job Submitted by Employer (For Admin):** This report contains how many Jobs are posted by Employer group by Emp\_Id. Admin will approve the posting based on this report.

**(2) Report on Employer (For Admin):** This report contains the Number of Employer created on this system with their Priority Level.

**(3) Report on Job Seeker (For Admin):** This report contains the number of Job Seeker created on this system.

**(4) Report on Jobs Status (For Employers):** This report contains how many Job are posted by them and with what status they are now.

**(5) Report on Applied Jobs (For Job Seeker):** This report contains total no. of Jobs applied by the Job Seeker with **Pending/Ok** Status.

1. **Date Validation**:The validation on date data type has been specified to be of the format DD/MM/YY. Any other format is unacceptable.
2. **Time validation:** The validation on time data type has been specified to be of the format hours-minutes-seconds. Any other format is unacceptable.
3. **Number field validation:** The field specified with number as then their data- type will not accept character.
4. **User Authentication:** When a Customer/user logs on to the system to access data from tables and database, the Id & password needs to be checked.
5. **Password change Validation:** Only authorized users are allowed to change the password and the process requires asking the old password before changing it to the new one.
6. **Unique Job-Id:** The Job-Id should not be edited. It must be unique.
7. **Unique Job-Seeker-Id/Employer-Id:** The Job-Seeker-Id/Employer-Id must be

unique.

1. Enhancement of our proposed system would be in terms of connectivity with the rest of the departments so that information is made available throughout the organization which leads to efficiency and speeding up of a number of processes in the organization.

Members in the organization will access this system at network level.

1. After successfully implementing the system, it can be added in the system that automatically displays all the Jobs Status along with the Employers and Job Seekers when the Admin Login to the System.
2. When the developed system runs successfully, it can be also be used in other areas with certain changes according to the requirements of that areas.
3. Online Seeker/Employer register facility could be brought on Web that would enable the Seeker/Employer to register and in one side the Employers will be able to post the jobs while the Job Seeker will be able to apply for the Jobs in the system itself by entering their Id(s)., thus reducing the dependencies between Job seekers and Employers.
4. Integration of **OLIP** (On-Line Interview Process) for automated telephonic and Cam’s Interview between the Employers and Job Seeker for any job will even reduce resources used for conducting interviews and many other factors.

\*\*\*\*\*\*\*\*\*

**Scope of future application**

**PROJECT INDEX**

|  |  |  |
| --- | --- | --- |
| SI No. | Subject | Page No. |
| 1 | **Introduction** | **24-25** |
| 2 | **Objective** | **26** |
| 3 | **System Analysis** | **27-28** |
| 4 | **Feasibility Study** | **29-35** |
| 5 | **S/W Engineering Paradigm Applied** | **36-39** |
| 6 | **S/W and H/W Requirement Specification** | **40-50** |
| 7 | **System Design(DFD and ERD)** | **51-89** |
| 8 | **Coding** | **90-173** |
| 9 | **Code Efficiency** | **174-179** |
| 10 | **Optimiztion of Code** | **180-181** |
| 11 | **Validation Check** | **182** |
| 12 | **Implementaion and Maintenance** | **183-186** |
| 13 | **Testing** | **187-196** |
| 14 | **Sysetm Security Measures** | **197-200** |
| 15 | **Cost Estimation of the Project** | **201-204** |
| 16 | **Pert Chart and Gantt Chart** | **205-209** |
| 17 | **Future Scope of the Project** | **210** |
| 18 | **Bibliography** | **211** |
| 19 | **Appendices(if any and glossery)** | **212-213** |

**2. INTRODUCTION**

**ABSTRACT**: In current competitive scenario every business establishment needs quality processes to increase their efficiency as well as improve their productivity. It is of vital importance that manual, time consuming & monotonous operations be automated so as to streamline the working of an organization. It is keeping in mind this business philosophy (mantra) that we propose a Online automated solutions for **Build Carrier Placements.** Our system will deal with all the aspects of any Consultant and Placement Service provider as well as report generation.

**PROPOSED SYSTEM: -**

1. This is a web application intended to provide automated solution to the Job Seekers, Employers along with the Consultant and Placements Service Provider.

**(2)** As this is completely web based, the Job Seeker can search job at any time as

well the Employer can also post jobs at any time on this system.

**(3)** The system will work in three ways:-

* + - * + From job seeker’s End
        + From employer’s End
        + From Admin End.

In the first way the job seeker can log on to the system via Internet. In this case the job seeker can enter in the system after the authentication of the seeker-id. After the authentication he/she could able to apply for any job that suits their profile. They can modify their profile. While in the second way, the

Employer can also logged in to the system with their proper login credentials. Once they get logged in they can post new jobs in **Build Carrier Placements,** they can Edit/Modify the earlier posted job. Employer can even delete any jobs from this application.

**(4)** When the job seeker/Employer create their profile in this system, the system will allot job-seeker-id/employer-id respectively to the. When the employer posts any job to this system, the system will generate a new job-id for that particular job.

**(5)** Beside job seeker and employer the system has one more important entity this is Admin.The main responsibility of **Admin** in this system is that once the job is posted by the employer it will remain in pending status until the admin will not accept this job.Once the **Admin** will accept the job, it will automatically get posted to the system.

Various responsibilities of **Admin** are listed below:-

* Grant permission to newly arrived job.
* Add/Remove details of job-seeker/employers.
* Edit the priority of employer.

**(6)** Status of any job seeker/employer/job can easily be determined by using their respective ids.

**(7)** The system provides various online reports for the concerned senior officers of

**Build Carrier Placements** that helps in decision-making.

**(8)** Nobody have right to delete any records excepts **Admin** or without **Admin** permissions.

## **3. Objective of the Project**

**🏱**The objectives of developing the **Build Carrier Placements** are as given below: -

**(1)** To store data of all the job seeker/employer who are registered with this organization.

**(2)** It will work as a bridge connection in between Employers and Job Seekers in efficient way.In essence it means that the Employer will not have to search the resources for long time and vice versa for the job seekers.

**(3)** To facilitate easy and efficient retrieval of data as and when required for study by the senior officials in this organization.

**(4)** To generate and print out the various reports. The online MIS reports are generated for the concerned senior officers of **Build Carrier Placements** to monitor the progress of organization in concern of their sponsors and income from them.

**(5)** The **tracking** of previous job seeker/employers feedbacks can also make the company (organization) aware of their changing needs.

**(6)** Data security is maintained to relatively high level by implementing it at Database level, so as to ensure that only authorized users have access to confidential client information.

## **4*.* SYSTEM ANALYSIS**

System Analysis is a management technique, which helps in designing a new system or improving an existing system. System Analysis is the process of gathering and interpreting facts, diagnosing problems (if any), using information to recommend improvements to the system. There are four basic elements of system analysis: - Output, Input, Files, processes. For computerization of any system, the existing system must be thoroughly being understood to determine “how the computer can be best used to make its operation most effective”. This is acquired by analyzing existing system.

4.1 *Identification of Need*

In current competitive scenario every business establishment needs quality processes to increase their efficiency as well as improve their productivity. It is of vital importance that manual, time consuming & monotonous operations be automated so as to streamline the working of an organization.**Build Carrier Placements** is completely web based idea of providing interactions between Job Seekers and Employers.It’s platform will allow seeker to search job online, they don’t have to go every company for job. Same for the employers they don’t need to wait for right candidate for long time.All these things can be easily done here with very less time and efforts.

## 4.2 *Preliminary Investigation*

The purpose of Preliminary Investigation is to evaluate project requests. It is the collection of details to describe the business system in all respect. Preliminary Investigation also examine project feasibility, the likelihood the system will be useful to the organization. Preliminary Investigation helps to accomplish the following objectives: -

1. Clarify and understand the project request.
2. Determine the size of the project.
3. Assess costs and benefits of alternative approaches.
4. Determine the feasibility of alternative approaches.

We worked a lot to investigate the requirements because it is the crucial part of the S/W Development Life Cycle. We got some details by examining organization charts and studying written operating procedures. The procedures helped in various important steps in receiving, assigning, managing, and closing the Problem/Complaint.

We also observed the activities of the system directly. During the on-site observation, we saw the office environment, work load of the system and users, method of work, and the facilities provided by the organization. This information helped us to understand how the system should operate. But after interviewing the persons, who is affected by the system, we got more details that further explain the project and shown whether assistance is merited economically, operationally and technically.

## **5. FEASIBILITY STUDY**

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. I have taken two weeks in feasibility study. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to proceed the project. It helped in taking decisions such as which software to use, hardware combinations, etc.

The following is the process diagram for feasibility analysis. In the diagram, the feasibility analysis starts with the user set of requirements. With this, the existing system is also observed. The next step is to check for the deficiencies in the existing system. By evaluating the above points a fresh idea is conceived to define and quantify the required goals. The user consent is very important for the new plan. Along with, for implementing the new system, the ability of the organization is also checked. Besides that, a set of alternatives and their feasibility is also considered in case of any failure in the proposed system. Thus, feasibility study is an important part in software development.



**(Figure – 1. PROCESS DIAGRAM FOR FEASIBILITY ANALYSIS)**

In the SDLC (Systems Development Life Cycle) of our project we maintained a number of feasibility checkpoints between the two phases of the SDLC.

**These checkpoints indicate that the management decision to be made after a phase is complete.**

The feasibility checkpoints in our project were as follows:

1. Survey phase checkpoint
2. Study phase checkpoint
3. Selection phase checkpoint
4. Acquisition phase checkpoint

## 5.1 *Technical Feasibility*

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

This project is feasible on technical remarks also, as the proposed system is more beneficiary in terms of having a sound proof system with new technical components installed on the system. The proposed system can run on any machines supporting **Windows** and **Internet** services and works on the best software and hardware that had been used while designing the system so it would be feasible in all technical terms of feasibility.

**Technical Feasibility Addresses three major issues:**

1. **Is the proposed Technology or Solution Practical?**

The technologies used are matured enough so that they can be applied to our problems. The practicality of the solution we have developed is proved with the use of the technologies we have chosen. The technologies such as ASP.NET, IIS, C# and the compatible H/Ws are so familiar with the today’s knowledge based industry that anyone can easily be compatible to the proposed environment.

1. **Do we currently posses the necessary technology?**

We first make sure that whether the required technologies are available to us or nor. If they are available then we must ask if we have the capacity. For instance, “Will our current Printer be able to handle the new reports and forms required of a new system?

**(c) Do we possess the necessary Technical Expertise and is the schedule reasonable?**

This consideration of technical feasibility is often forgotten during feasibility analysis. We may have the technology, but that doesn’t mean we have the skills required to properly apply that technology.

As far as our project is concerned we have the necessary expertise so that the proposed solution can be made feasible. Some projects are initiated with specific deadlines. In our case first we have given three months time but due to some problems regarding time and the constraints of expertise it has been extended to six months. Now there are some organizational constraints that have not yet given us the opportunity to install the system.

## 5.2 *Economical Feasibility*

Economical feasibility determines whether there are sufficient benefits in creating to make the cost acceptable or is the cost of the system too high. As this signifies cost-benefit analysis and savings.On the behalf of the cost-benefit analysis, the proposed system is feasible and is economical regarding its pre-assumed cost for making a system.

During the economical feasibility test we maintained the balance between the Operational and Economical feasibilities, as the two were the conflicting. For example the solution that provides the best operational impact for the end-users may also be the most expensive and, therefore, the least economically feasible.

We classified the costs of **BUILD CARRIER PLACEMENTS** according to the phase in which they occur. As we know that the system development costs are usually one-time costs that will not recur after the project has been completed. For calculating the Development costs we evaluated certain cost categories viz.

1. Personnel costs
2. Computer usage
3. Training
4. Supply and equipments costs
5. Cost of any new computer equipments and software.

## 5.3 *Operational Feasibility*

Operational feasibility criteria measure the urgency of the problem (survey and study phases) or the acceptability of a solution (selection, acquisition and design phases). How do you measure operational feasibility? There are two aspects of operational feasibility to be considered:

1. **Is the problem worth solving or will the solution to the problem work?**

There are certain measures, which decide, the effectiveness of the system. These measures can be collectively called as **PIECES.**

**P (Performance):** The BUILD CARRIER PLACEMENTS adequate throughput and response time.

**I (Information):** The BUILD CARRIER PLACEMENTS provide end-users and managers with tamely, pertinent, accurate, and usefully formatted information.

E **(Economy):** The BUILD CARRIER PLACEMENTS offer adequate service level and capacity to reduce the costs of the business or increase the profits of the business.

**C (Control):** The BUILD CARRIER PLACEMENTS offer adequate controls to protect against fraud and embezzlement and to guarantee the accuracy and security of the data and information.

**E (Efficiency):** The BUILD CARRIER PLACEMENTS make maximum use of available resources including people, time, flow of forms, minimum processing delays and the like.

**S (Services):** The BUILD CARRIER PLACEMENTS provide desirable and reliable

service to those who need it. The BUILD CARRIER PLACEMENTS is flexible and

xpandable.

**(b) How Do the End-Users and Managers feel about the problem (Solution)?**

It is not only important to evaluate whether a system can work. We must also evaluate whether a system will work. A workable solution might fail because of end-user management resistance.

In case of our project BUILD CARRIER PLACEMENTS we have examined all the concerns that can further affect its operational feasibility. The following points will explore those concerns:

1. The BUILD CARRIER PLACEMENTS has complete support of the management of **ORGANIZATION**. As far as organizational constraints are concerned we have maintained the path that never infringes those constraints.
2. The BUILD CARRIER PLACEMENTS has made the role of end-users an easiest one. The end-users feel comfortable and upgraded with the BUILD CARRIER PLACEMENTS as there is universal tendency that people tend to resist change. We have overcome this problem with the convincing effort made to aware the end-users and management about the betterment of the services that the BUILD CARRIER PLACEMENTS will provide the customers and the users.
3. As any new system requires its own working environment, so is the case with the BUILD CARRIER PLACEMENTS. The new environment that is compatible with the BUILD CARRIER PLACEMENTS requires user training. The training will be made, so that end-users and management adapt to the change.

## **6. SOFTWARE ENGINEERING PARADIGM APPLIED**

The development strategy that encompasses the process, methods, and tools and the generic phases is called Software Engineering Paradigm. The s/w paradigm for software is chosen based on the nature of the project and application, the method and tools to be used, and the controls and deliverables that are required. All software development can be characterized as a problem-solving loops (fig. 2) in which four distinct stages are encountered: status quo, problem definition, technical development, and solution integration.

Problem

Solving

Technical

Development

Solution

Integration

**(Figure - 2) Problem Solving Loop**

Status quo represents the current state of affairs, Problem definition identifies the specific problem to be solved, technical development solves the problem through the application of some technology, and solution integration delivers the results to those who requested the solution in the first place.

There are various software paradigms, but we used **Waterfall model** (the linear sequential model), which states that the phases are organized in a linear order. The Waterfall model suggests a systematic, sequential approach to s/w development that begins at the system level and progresses through analysis, design, coding, testing, and maintenance and support as shown in below (fig.3.)

System

Analysis

System

Design

Coding

Testing and Integration

Operations and

Support

Installation

Requirement Document & Project plan

System & Detailed

Design document

Programs

Test plan,

Report & manuals

Installation

Report

**(Figure- 3) Waterfall model**

The sequence of activities performed in a software development project with the Waterfall model is: system analysis, system design, coding, testing & integration, installation, and maintenance. For a successful project resulting in a successful product, all phases listed in the waterfall model must be performed. Any different ordering of the phases will result in a less successful software product.

There are a number of project outputs in waterfall model that is produced to produce a successful product:

* Requirement documents and project plan
* System and detailed design
* Programs (code)
* Test plan, test reports and manuals
* Installation reports

**LIMITATIONS OF WATERFALL MODEL:**

1. The waterfall model assumes that the requirements of a system can be baselined before the design begins. This is possible for system designed to automate an existing manual system. For our system, **(BUILD CARRIER PLACEMENTS)** this is a new system, determining the requirement is difficult, as the user does not even know the requirements.
2. Feezing the requirements usually requires choosing the hardware.

The waterfall model stipulates that the requirements be completely specified before the rest of the development can proceed.It is a document driven process that requires formal documents at the end of each phase. This approach tends to make the process documentation-heavy and is not suitable for many applications (interactive applications).

The waterfall model is the most widely used process model. It is suited for routine types of projects where the requirements are well known, i.e. if the developing organization is quite familiar with the problem domain and the requirements for the software are quite clear, the waterfall model works well. This applies on our project.

A Closed paradigm structures a team, along a traditional hierarchy of authorities. Such teams can work well when producing software that is quite similar to past efforts, but they will be less likely to be innovative when working within the closed paradigm. Only two persons, I and my co-developer is concerned with this project, so, I and he is playing the role of system analyst, programmer as well as testing engineer interchangeably.

## **7. S/W AND H/W REQUIREMENT SPECIFICATION**

## ***Project Category***

### Internet

**Hardware Requirements**:

* Pentium III 1.5 GHz and Above
* 128 MB SDRAM or More
* 20 GB HDD
* Printer
* Power Backup
* Telephone and Internet Connection

## **Software Requirements:**

* Windows NT Server or Higher
* Internet Information System (IIS)
* .NET Framework
* SQL Server 2000
* C#
* HTML

**ABOUT ACTIVE SERVER PAGE (ASP.NET):**

**Introduction:** Active Server Pages are Microsoft’s solution to creating dynamic web pages.An ASP.NET file can contain text, HTML tags and scripts. Scripts in an ASP.NET file are executed on the server.

**What is ASP.NET?**

ASP.NET is a program that runs at server. With Windows 2000/XP IIS plays the role of server. IIS comes as a free component with Windows 2000/XP. The detailed discussion about IIS has been given as separate topic.

Microsoft ASP.NET is not just an upgrade—not by a long shot. ASP.NET

provides the most advanced Web development platform created to date. What’s more, ASP.NET has been rebuilt from the ground up to create an entirely new and more flexible infrastructure for Web development.What makes ASP.NET so revolutionary is that it’s based on Microsoft’s new.NET platform, or more accurately the Microsoft .NET Framework.

In ASP.NET ***Client-Server Model*** is implemented. A client computer requests some needed information from Server computer. The Server returns this information and the client acts on it. The ***client*** is a web browser on the Internet. With the Internet the server is a particular web server. A ***web server*** is a computer that contains all the web pages for a particular web site and has special software installed to send these web pages to web browsers that request them.

The client can’t tell the difference between an ASP.NET page and a static web page because it receives just HTML text in both cases. When the web server processes an ASP.NET page, all the programmatic code is interpreted on the server-none of it is sent to the client. The web plays a more active role when an ASP.NET page is requested by the client.

Server

Client requests ASP page

Server returns HTML

The Server locates the ASP file on the hard drive and parses it, removing all ASP scripts and replacing it with HTML text.

**(Figure – 4.Client/Server Interaction for ASP.NET files)**

**ASP.NET OBJECT HIERARCHY:**

Response Object

Request Object

**Session Object**





This hierarchy diagram shows the process of creating and serving ASP.NET Pages.

**Application Object**

Client

**Session Object**

Client

**Client Request**

**Server Response**

Server

**Server Object**

**.NETError Object**

**Client Request**

**Server Response**

**(Figure – 5. Process of creating and serving ASP.NET pages)**

1. **Response Object:** This object is used to send output. The write method sends output to the user’s web browser.
2. **Request Object:** This object is used to retrieve data from the client. When the client’s web browser makes a request for a particular page, it sends some information along to the server. That data is packaged together in the Request object.
3. **Application Object:** Application is used to share information among several clients visiting the same group of pages. In ASP.NET the term application refers to all the .ASP.NET pages in a directory and its subdirectories. Only one instance of the Application Object is created per application. It is shared among all the clients accessing that application.
4. **Session Object:** A session on the other hand, refers to a single client accessing an application. Therefore a new instance of the Session object is created for each session. Session is important to carrying information as a client travels between pages because Session variables persist for the entire session. One page can store data into a Session variable, and that data can be accessed from other pages in the session.
5. **Server Object:** The server object provides us with a series of methods and properties that are useful in scripting with ASP.NET. The most obvious is the Server.Create Object method,which allows us to properly instantiate other COM objects on the server within the context of the current page or session.Thereare also methods to translate strings into the correct format for use in URLs and in HTML, by converting non-legal characters to the correct legal equivalent.
6. **ASP.NET Error Object:** This object is a new object in ASP.NET 3.0, and is available through the Get Last Error method of the Server Object. It provides a range of detailed information about the last error that occurred in ASP.NET.
7. **Object Context Object:** This object is used to link ASP.NET and the Microsoft Transaction Server. MTS is used to make web sites more scalable and improve the performance of other components.

**ABOUT HTML:**

Hypertext Markup Language (HTML) is a language for describing how pages of text, graphics and other information are organized. Hypertext means text stored in electronic form with cross-reference links between pages. An HTML page contains HTML tags, which are embedded commands that supply information about the page’s structure, appearance, and contents. Web browsers use this information to determine how to display the page.

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

**ABOUT C#:**

C# is a simple, modern, object oriented, and type-safe programming language derived from C and C++. It will immediately be familiar to C and C++ programmers. C# aims

to combine the high productivity of Visual Basic and the raw power of C++.Visual C# .NET is Microsoft’s C# development tool. It includes an interactive development environment, visual designers for building Windows and Web applications, a compiler, and a debugger. Visual C# .NET is part of a suite of products, called Visual Studio .NET, that also includes Visual Basic .NET, Visual C++ .NET, and the JScript scripting language. All of these languages provide access to the Microsoft .NET Framework, which includes a common execution engine and a rich class library. The.NET Framework defines a “Common Language Specification” (CLS), a sort of lingua franca that ensures seamless interoperability between CLS-compliant languages and class libraries. For C# developers, this means that even though C# is a new language, it has complete access to the same rich class libraries that are used by seasoned tools such as Visual Basic .NET and Visual C++ .NET. C# itself does not include a class library.

**ABOUT WINDOWS 2000 PROFESSIONAL:**

WINDOWS 2000 PROFESSIONAL has many features that will make using the Computer easy, effective and entertaining. For instance, you can use Remote Desktop to access your work computer and its resources from home and to view files

and documents on the computer’s desktop from a co-workers computer. WINDOWS 2000 PROFESSIONAL extensive online help for all OS as well as a digital tour to help you discover the possibilities awaiting you. There are tools that can used to get the most out of the computer experience and other technology that run in the background making the computer run more efficiently and reliably.

With WINDOWS 2000 PROFESSIONAL, you can work with your files and folders more efficiently. You can perform tasks with a more intuitive Web page interface, and view file and folder details in new and helpful ways. There are more options to associate certain file types with specific programs and working files and folders offline has never been easier. Folders and files are compressed quickly and easily, so they take up less space on the computer.

WINDOWS 2000 PROFESSIONAL combines a bright, fresh new look with a simple-to-use design. The desktop and taskbar are less cluttered. The Start menu provides easier access to the programs. There are more options for customizing desktop environment.

WINDOWS 2000 PROFESSIONAL offers many new and more effective features and technologies. Remote Desktop allows accessing Windows session from another computer, just as if you were in front of your computer. Features like Windows File Protection and System Restore will keep from accidentally deleting important files and will return computer to its previous state if something goes wrong. If a system or program error is encountered, a report can be send to Microsoft and

NetMeeting can be used for virtual meeting with anyone, anywhere .If a flat computer

screen is in use, then you can choose to display screen fonts with ClearType, the latest breakthrough technology in font clarity from Microsoft. The Dual View features enables to use a separate monitor with the laptop computer.

With WINDOWS 2000 PROFESSIONAL home networking never been easier. The Network Setup Wizard can be used to quickly setup own network. You can

share your Internet connection with all the computers on your network, and feel comfortable knowing your computer is protected by an Internet Connection Firewall.

WINDOWS 2000 PROFESSIONAL makes it easy to set up and manage computer accounts for everyone who uses your computer. Now multiple users can switch between accounts without having to restart the computer. Help is provided for remembering forgotten password, store multiple and protect copy of WINDOWS 2000 PROFESSIONAL against piracy.

WINDOWS 2000 PROFESSIONAL includes the newest version of MSN Explorer, with its full package of Micros of services and the most recent release of Internet Explorer.

**About Internet Information Services (IIS):**

Internet Information Services (IIS) is the Windows component that makes it easy to publish information and bring business applications to the Web. IIS makes it easy for you to create a strong platform for network applications and communications. Internet Information Services 5.1 has many features to help Web administrators to create scalable, flexible Web applications.

* Security
* Administration
* Programmability
* Internet Standards

Microsoft Internet Information Services 5.0 and 5.1 comply with the HTTP 1.1 standard, including features such as PUT and DELETE, the ability to customize HTTP error messages, and support for custom HTTP headers.

IIS 5.1 offers greater protection and increased reliability for your Web applications. By default, IIS runs all of your applications in a common or pooled process that is separate from core IIS processes.

In IIS 5.1, administrators and application developers have the ability to add custom objects, properties, and methods to the existing ADSI provider, giving administrators even more flexibility in configuring their sites

Internet Information Services (IIS) makes it easy for you to publish information on the Internet or your intranet. IIS includes a broad range of administrative features for managing Web sites and your Web Server. With programmatic features like Active Server Pages (ASP.NET), you can create and deploy scalable, flexible Web applications.

### ABOUT SQL SERVER 2000

SQL Server is one of the most popular RDBMS of today. Its Popularity is owned to a number of factors like its ability to support dozens to thousands of simultaneous users, its ability to handle high transaction rates and its ability to access databases that ranges in size from several hundred megabytes to many gigabytes. SQL Server 8 is based on new architecture; know as NCA (Network Computing Architecture). NCA is a three-tried architecture as opposed to client/server architecture that is two tired.

**FEATURES OF SQL 2000**

* **Security Management:** SQL Server provides a controlled access to data to users buy providing a combination of privileges.
* **Backup and Recovery:** SQL Server provided sophisticated security backup and recovery routines.
* **Open connectivity:** SQL Server provides open connectivity to and from other vendor’s software such as Microsoft. Also SQL Server database can be access by various front-end software’s such as Microsoft Visual Basic ®, Power Builder etc.
* **Space Management:** In SQL Server once can flexibly allocate disk spaces for data storage and can control them subsequently. SQL Server 8 is designed with special feature of data warehousing.

The reason for choosing SQL SERVER 7.0 as RDBMS is because; one of SQL's greatest benefits is that it is a truly cross-platform language and a cross-product language. It is also what programmers refer to as a high-level or fourth-generation language (4GL), in which a large amount of work can be in fewer lines of code. SQL is the de facto standard language used to manipulate and retrieve data from these relational databases.

## **8. SYSTEM DESIGN**

The objective of the system design is to deliver the requirements as specified in the feasibility report. System design involves first logical design (logical design) and then physical construction (detailed design) of the system. The logical design describes the structure and characteristics of features, such as the outputs, inputs, files, databases, and procedures. The physical construction produces actual program software, files, and a working system.

**1. Data Design:** It transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the ER-Diagram and the detailed data

content depicted in the Data-Dictionary provide the basis for the data design activity.

The **ER-Diagram** of **BUILD CARRIER PLACEMENTS** has been given in the section 8.1 of System Design.

The complete **Data Structure** of the system along with the relationships has been depicted in the relationship Diagram attached with the document.

**DATA STRUCTURE (LIST OF TABLES USED)**

* 1. **JobSeeker\_Profile:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Constraints** | **Description** |
| Seeker\_Id | NVARCHAR | 20 | Primary Key | Identification of Job Seeker |
| First\_Name | NVARCHAR | 20 |  | Job Seeker First Name |
| Last\_Name | NVARCHAR | 20 |  | Job Seeker Last Name |
| Address | NVARCHAR | 500 |  | Address of the Job Seeker |
| City | NVARCHAR | 20 |  | Job Seeker’s City |
| State | NVARCHAR | 20 |  | Job Seeker’s State |
| Contact\_No | BIGINT | 15 |  | Job Seeker’s Contact# |
| Email | NVARCHAR | 100 |  | Job Seeker’s Email\_Id |
| DoB | NVARCHAR | 20 |  | Job Seeker’s Date of Birth |
| Edu\_Qualification | NVARCHAR | 1000 |  | Educational Qualification |
| Pro\_Qualification | NVARCHAR | 1000 |  | Professional Qualification |
| OtherQualification | NVARCHAR | 1000 |  | Other Qualification |
| Resume\_Title | NVARCHAR | 50 |  | Resume Title |
| Experience | NVARCHAR | 20 |  | Work Experience of Seeker |
| Job\_Category | NVARCHAR | 1000 |  | Category of Job looking for |
| Password | NVARCHAR | 100 |  | Password of the Seeker |

**2.Seeker\_Inbox:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **FieldType** | **Size** | **Constraints** | **Description** |
| RowID | Int | -- |  | Maintaining Rows in Table |
| Seeker\_Id | NVARCHAR | 20 | Foreign Key | Identify the Job Seeker |
| Mail\_Id | NVARCHAR | 100 | Foreign Key | Identify the Employer |
| MsgFrom | NVARCHAR | 50 |  | Message send from Which Company |
| Msg\_Subject | NVARCHAR | 1000 |  | Subject of the Message |
| Msg\_Desc | NVARCHAR | 1000 |  | Description of Message |
| Msg\_Status | BIT | 1 |  | Identification Status of Message |
| Msg\_Date | DATETIME | 20 |  | Date of Message |

**3.Seeker\_SentMail:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **FieldType** | **Size** | **Constraints** | **Description** |
| Seeker\_Id | NVARCHAR | 20 | Foreign Key | Identification Job Seeker |
| Mail\_Id | NVARCHAR | 20 | Foreign Key | Identification Message for Seeker |
| MsgFrom | NVARCHAR | 100 |  | Sender MailID |
| Msg\_Subject | NVARCHAR | 1000 |  | Subject of Message |
| Msg\_Desc | NVARCHAR | 1000 |  | Description of Message |
| Msg\_Date | NVARCHAR | 20 |  | Date of Msg |

**4.Seeker\_Trash:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Constraints** | **Size** | **Description** |
| Seeker\_Id | NVARCHAR | Foreign Key | 20 | Identification Job Seeker |
| Mail\_Id | NVARCHAR | Foreign Key | 20 | Identification Message for Seeker |
| MsgFrom | NVARCHAR |  | 100 | Mail Came From |
| Msg\_Subject | NVARCHAR |  | 1000 | Subject of Message |
| Msg\_Desc | NVARCHAR |  | 1000 | Description of Message |
| Msg\_Date | NVARCHAR |  | 20 | Date of Msg |

**5.Employee:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **FieldType** | **Size** | **Constraints** | **Description** |
| RowID | BIGINT | -- |  | Identification of Rows in Table |
| Employer\_Id | NVARCHAR | 20 | Primary Key | Identification Coloumn for Employer |
| E\_Name | NVARCHAR | 100 |  | Employers Name |
| E\_Address | NVARCHAR | 1000 |  | Address of Employer |
| E\_ContactNo | NVARCHAR | 50 |  | Contact number of Employer |
| Comp\_Name | NVARCHAR | 500 |  | Name of Company |
| Comp\_Address | NVARCHAR | 1000 |  | Address of Company |
| City | NVARCHAR | 100 |  | City Name |
| State | NVARCHAR | 100 |  | State Name |
| Contact\_No | NVARCHAR | 50 |  | Company Contact Number |
| Comp\_Website | NVARCHAR | 50 |  | Website for Company |
| Email\_Id | NVARCHAR | 100 |  | Employer Mail ID |
| Password | NVARCHAR | 100 |  | Employer Password |

**6.EmployerPendingRequest:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **Field Type** | **Constraints** | **Size** | **Description** |
| RowID | BIGINT |  | -- | Identification Column of Table |
| Employer\_Id | NVARCHAR | Foreign Key | 20 | Employer ID |
| Seeker\_ID | NVARCHAR | Foreign Key | 10 | Seeker ID |
| Job\_ID | NVARCHAR | Foreign Key | 1000 | Job ID |
| JobTitle | NVARCHAR |  | 1000 | Title of Job |
| Date | NVARCHAR |  | 20 | Date of Job Posted by Employer |
| Status | BIT |  | 1 | To Maintain Status of Job |

**7.Job\_Dtls:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Constraints** | **Description** |
| RowID | BIGINT | -- |  | Identification of Rows in Table |
| Job\_Id | NVARCHAR | 20 | Primary Key | Uniquie Job ID |
| Job\_Title | NVARCHAR | 100 |  | Job Title |
| Job\_Desc | NVARCHAR | 1000 |  | Description of Job |
| Employer\_ID | NVARCHAR | 50 | Foreign Key | Employer ID |
| Job\_Category | NVARCHAR | 500 |  | What is the Category of Job |
| City | NVARCHAR | 100 |  | City Name |
| State | NVARCHAR | 100 |  | State Name |
| Exp\_Req | NVARCHAR | 50 |  | Experience required for the post |
| No\_Of\_Days | INT | -- |  | For how many days job will visible on **A1 Jobs’s website** |
| Posting\_Date | NVARCHAR | 100 |  | Posting Date of Job |
| Expiry\_Date | NVARCHAR | 100 |  | Expiry Date of Job |
| JobStatus | BIT | 1 |  | To Maintain Job Status |

**8.JobCategory:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Constraints** | **Description** |
| CategoryID | NVARCHAR | 50 | Primary Key | Unique Category Id of Job |
| JobCategory | NVARCHAR | 50 |  | Category Name Of Job |
| Status | BIT | 1 |  | Maintaining Category Status |

**9.job\_transaction:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Constraints** | **Description** |
| Job\_Id | NVARCHAR | 50 | Foreign Key | Job ID |
| Employer\_Id | NVARCHAR | 50 | Foreign Key | Employee ID |
| Seeker\_Id | NVARCHAR | 20 | Foreign Key | Seeker ID |
| E\_Status | BIT | 1 |  | Employee Status in Job Transaction |
| S\_Status | BIT | 1 |  | Seeker Status in Job Transaction |

**10.AdminLogin:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Size** | **Constraints** | **Description** |
| Login\_Id | NVARCHAR | 50 | Primary Key | Login ID for Admin |
| LoginPWD | NVARCHAR | 50 |  | Password for Admin |

### COMPLETE STRUCTURE OF THE PROGRAM

**ERD:-**

Mail

Register

Job-Seeker

Build Carrier

Admin

Employer

Mailbox

Apply

Request

Check Mailbox

Job

Post

Delete

**2. Input/Output Design:**

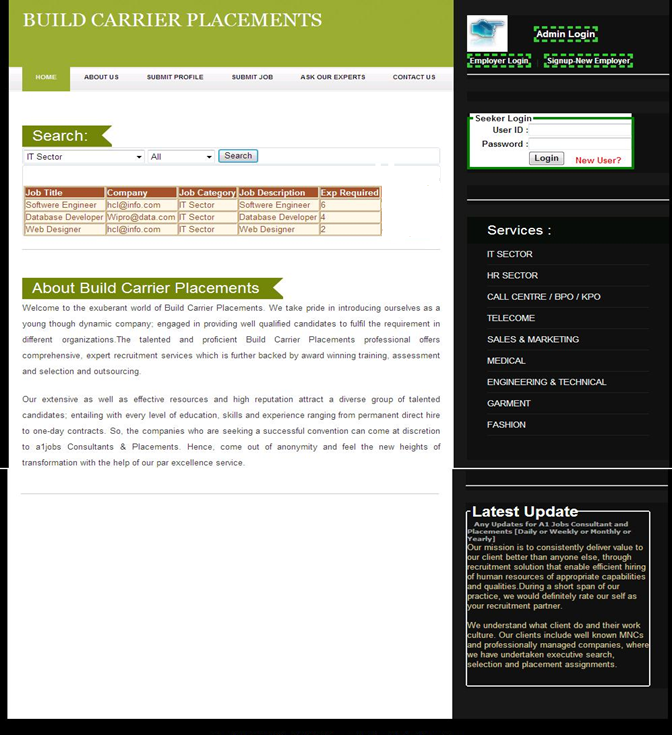
* **Input design**: The input design play very significant role in getting the correct output. It covers all phases of input from creation of initial data to actual entering the data to the system for processing. The objectives of input design are:
* Controlling amount of data
* Avoiding delay
* Avoiding errors in data
* Avoiding extra steps
* Keeping the process simple

**Input Verification and Control:** If incorrect data enters into the system it is very costly to make the necessary correction. There are following methods, which are used to verify data entering into the system as input:

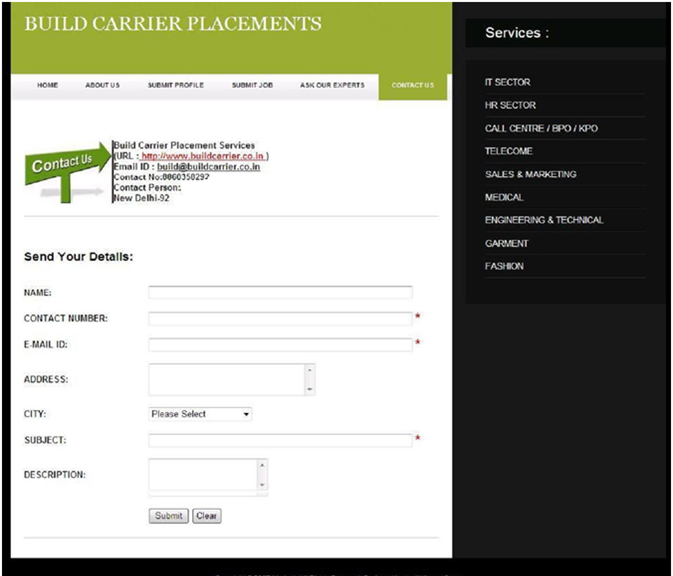
2.Verification of Data type: Some input fields should contain only numeric data while others should contain only character data The constraint of DateTime datatypehasbeenmaintained.

1. Use of self-checking number: Our system has been programmed to reject numbers that have been transposed of have one or more wrong digits. Check digits and self-checking number routines can be effectively used for numbers in a series such as SeekerID, E\_ID, CategoryID and JobID.

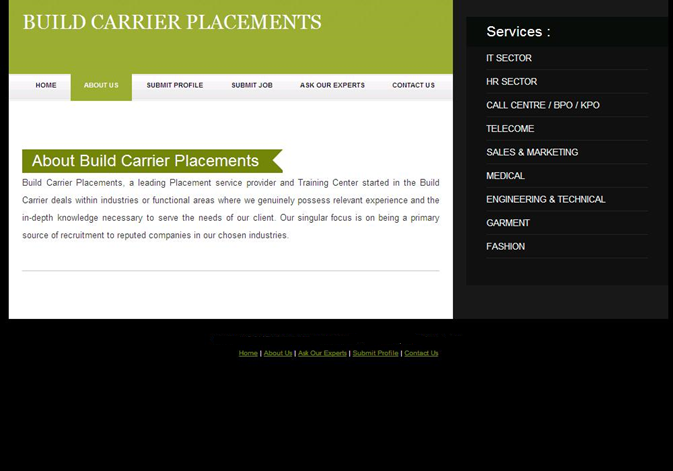
**Home Page for Build Carrier Placements:**

****

**Contact Us :**



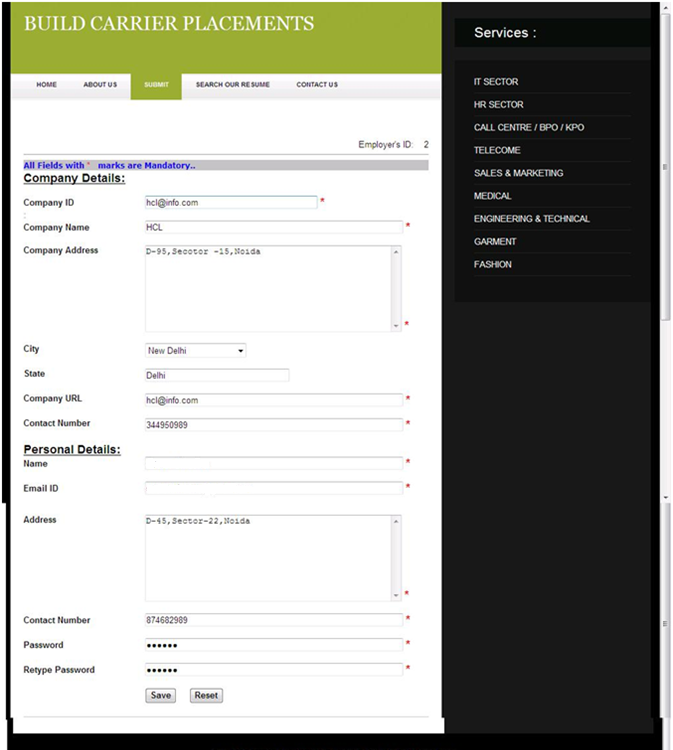
**About us Page**



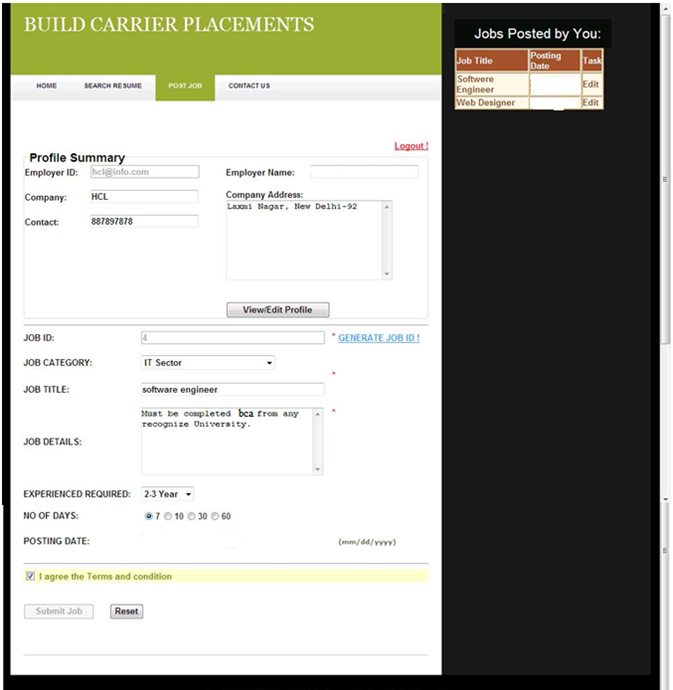
**Our Experts Page**



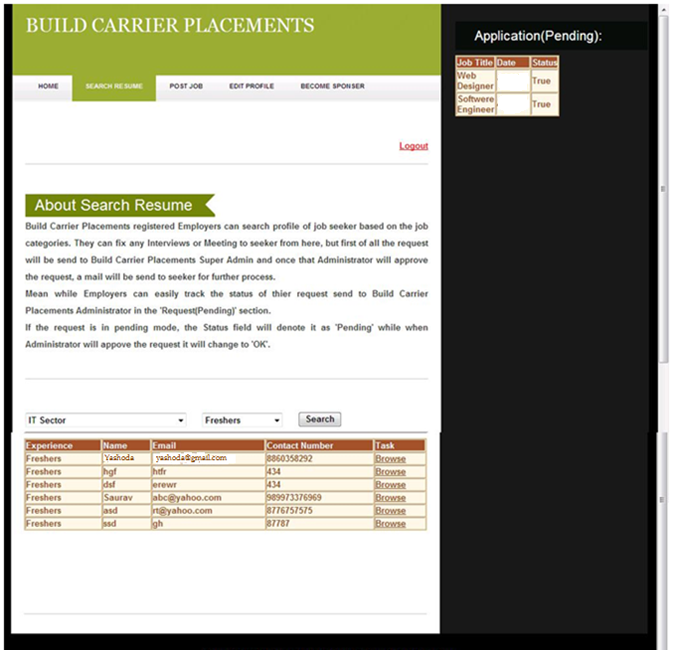
**New Employer SignUp on Build Carrier Placements :**



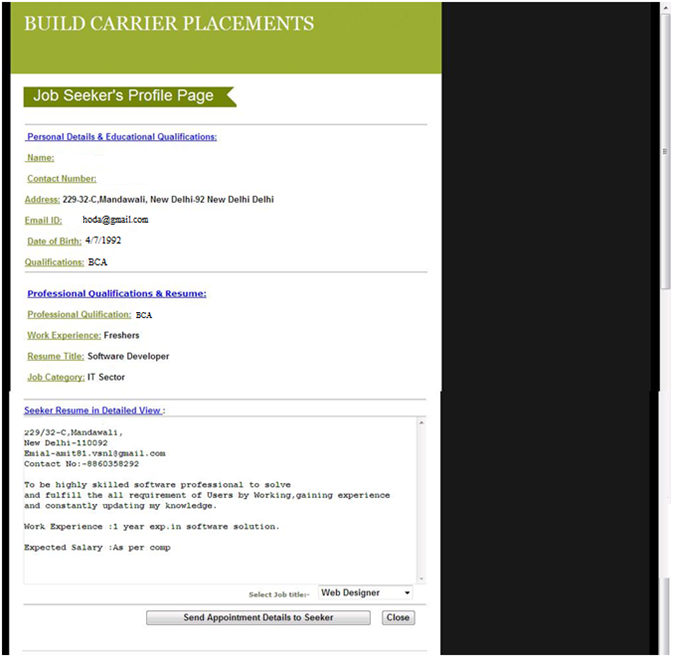
**Post Job Page**



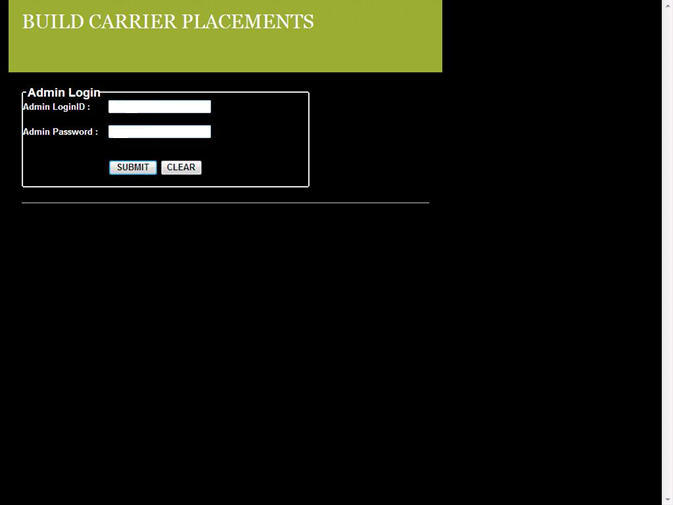
**Search Resume Page**

****

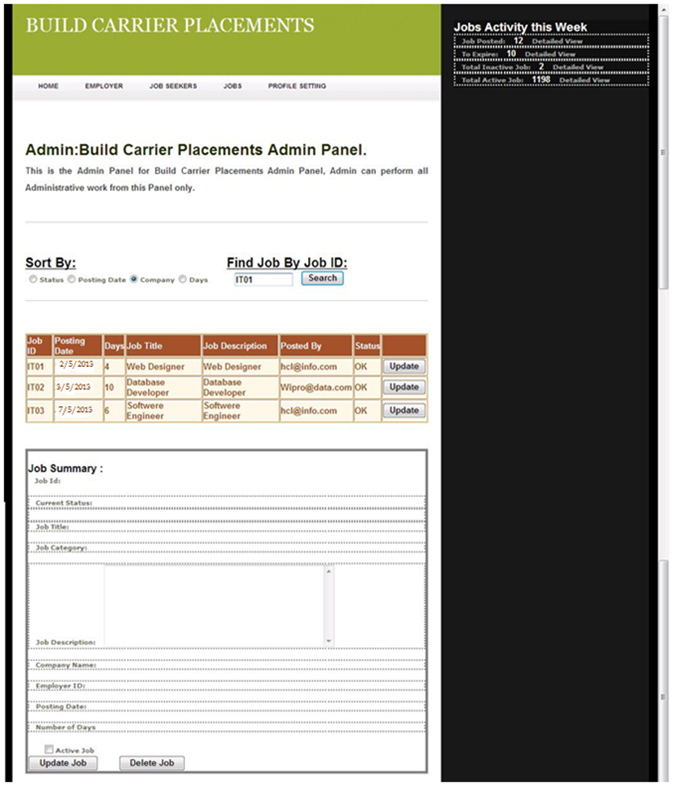
**Seeker Resume in Detailed View (***SeekerProfile.aspx***)**

****

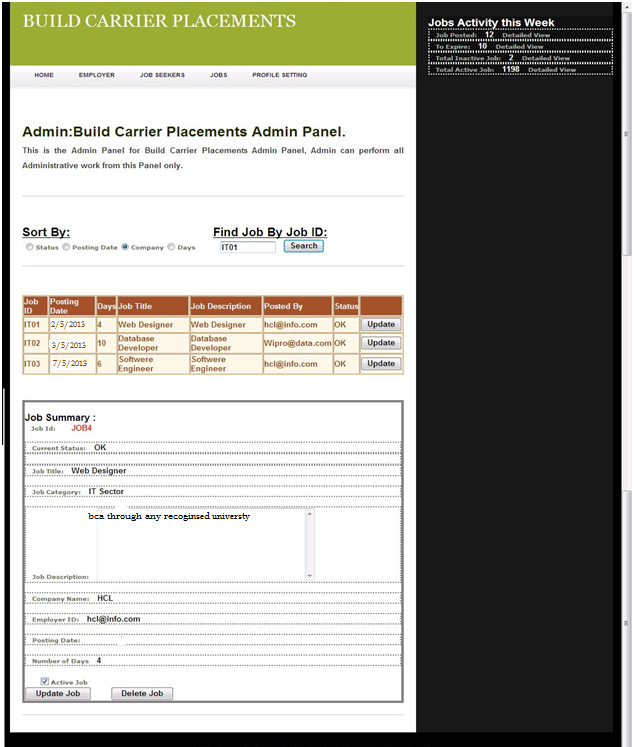
**Admin Login Page (**AdminLogin.aspx**)**



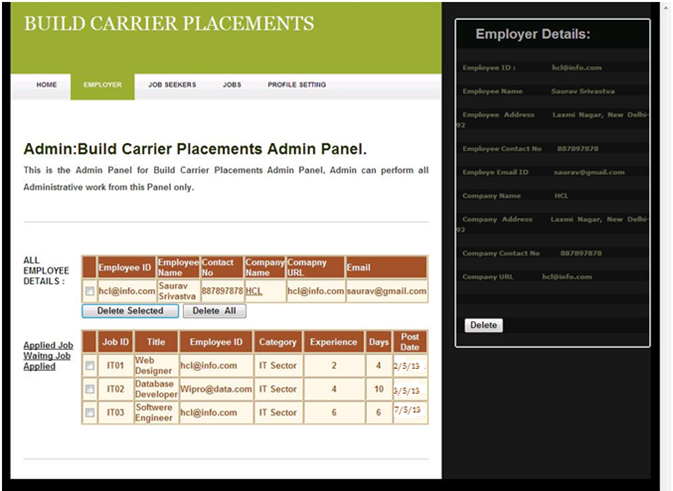
**Admin Panel Home Page (**AdminPanel.aspx**)**



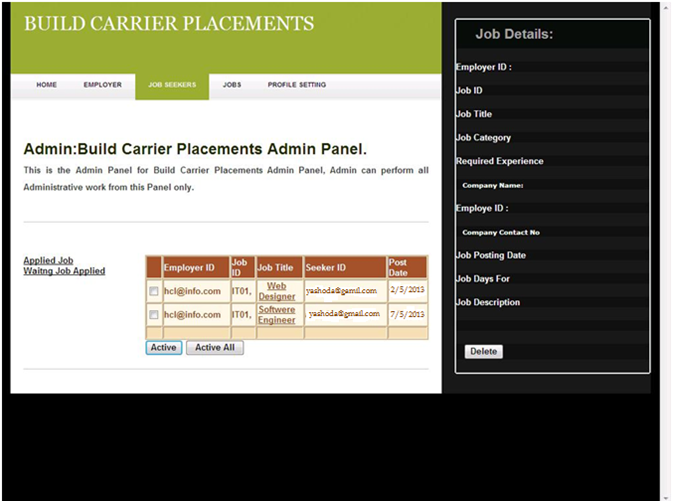
**Update Job From Admin End (**AdminPanel.aspx**)**



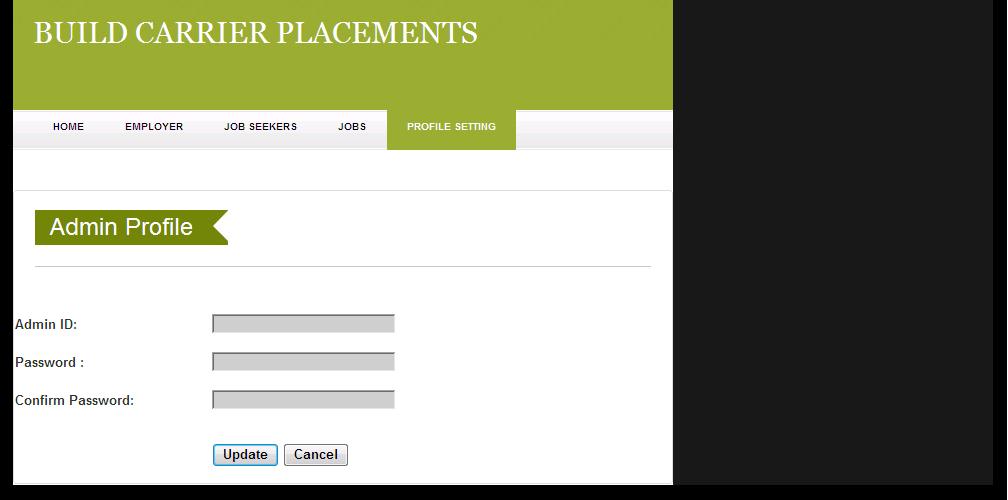
**Employer Section in Admin Panel (**Employer.aspx**)**



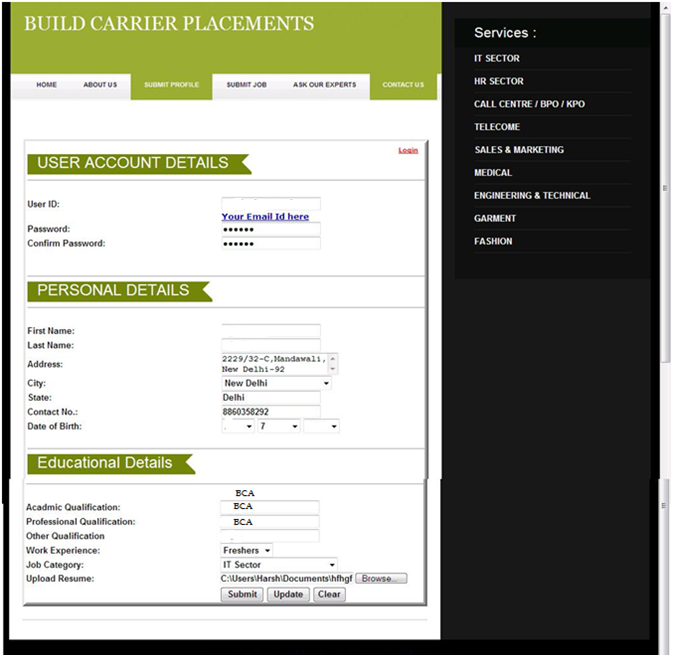
**Job Seeker Page in Admin Panel (**JobSeeker.aspx**)**



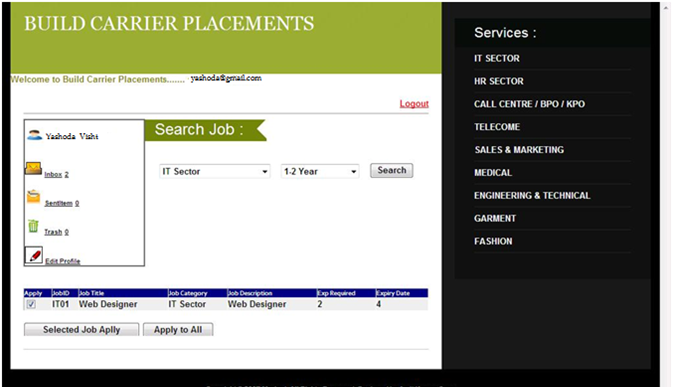
**Profile Setting for AdminPanel(ProfileSetting.aspx)**



**Job Seeker New Registration Page (**Registration\_step1.aspx**)**

****

**Job Seeker home page (**BuildCarrier\_User.aspx**)**



* **Output Design**: Output of this system **(BUILD CARRIER PLACEMENTS)** is various forms. The most common are reports, displays on screen, printed forms. Outputs can be divided in two parts:-

1. **Application Output**: Application outputs are of three types:-
   1. Output as a basis for decision making: Management for decision-making processes generally requires this type of output. Reports are used for decision-making.
   2. Output as a requirement to meet a functional objective
   3. Statutory Output: The organization is required to produce reports and forms as required by law.
2. **Operating Output:**
3. Operating outputs are generated for use of data processing staff and give various indications as to how the system operates. Example: system logs, error messages, status indicators, etc**.**

**Screen to Input Date to Produce Report on List of Complaints**

This screen is used to input the date on which the report on list of customer has been generated.

1. **File design**: Files are the heart of a computer system. The selection of a particular file depends on the nature of the application for which it will be used. There are various types of files such as master, transaction, table, report, etc. The JOB DETAILS contains all the records of the job. The

EMPLOYER contains the records of the Company and Job.JOB SEEKER contains the record of our own records . The JOB CATEGORY contains the record of job category.

3. Component level design (Procedural design):

Component level Design establishes the algorithmic detail required to manipulate data structures, effect communication between software components via their interface, and implement the processing algorithms allocated to each component Design representations of data, architecture, and interfaces form the foundation for component-level design.

STRUCTURED DESIGN METHODOLOGIES

Structured design proceeds from top down. Structured design is an attempt to minimize the complexity and make a problem manageable by subdividing it into smaller segments, which is called modularization or decomposition. Structured design is a set of techniques, guidelines, and a method for making program coding, testing, and maintenance easier by reducing the complexity of programs**. Structured design reduces the complexity by breaking the programs into small pieces called modules.**

**MODULAR DESIGN CONCEPTS**

**(i) Functional Independence:** The concept of functional independence is a direct outgrowth of modularity and the concepts of abstraction and information hiding.

The principle of information hiding suggests that modules be “characterized by design decisions that (each) hides from all others”. In other words modules should be specified and designed so that information (procedure and data) contained within a module is inaccessible to other modules that have no need for such information. Hiding implies that effective modularity can be achieved by defining a set of independent modules that communicate with one another only that information necessary to achieve software function. Abstraction helps to define the procedural (or informational) entities that make up the software. As most data and procedure are hidden from other parts of the software inadvertent errors introduced during modification are less likely to propagate to other locations within the software.Functional independence is achieved by developing modules with “single-minded” function and an “aversion” to excessive interaction with other modules.

**Advantages:** Independent modules are easier to maintain (and test) because secondary effects caused by design or code modification are limited, error propagation is reduced, and reusable modules are possible.Thus with taking utmost care of this concept we have maintained functional independence in our project BUILD CARRIER PLACEMENTS at some extent that required interaction among different modules is maintained.

**(ii) Cohesion:** Cohesion of a module represents how tightly bound the internal elements of the module are to one another. Cohesion of a module gives the designer an idea about whether the different elements of a module belong together in the same module.

**(iii) Coupling:** Coupling is a measure of interconnection among modules in a software structure. Coupling depends on the interface complexity between modules, the point at which entry or reference is made to a module, and what data pass across the interface. In software design, we strive for lowest possible coupling. Simple connectivity among modules results in software that is easier to understand and less prone to a “ripple effect” when errors occur at one location and propagate through a system.

* **Data coupling:** Data coupling means simple argument list (data) is passed and a one to one correspondence exists. A variation of data coupling is found when a portion of a data structure rather than simple arguments is passed via a module interface.
* **Control coupling:** When a “control flag” (a variable that controls decisions in a subordinate or superordinate module) is passed between modules.
* **External coupling:** It is a relatively high level of coupling occurs when modules are tied to an environment external to software.
* **Common coupling:** When a number of modules reference a global data area. In A1 BUILD CARRIER PLACEMENTS we have maintained the use of global data but restricted ourselves against the common consequences of this coupling.
* **Content coupling:** The highest degree of coupling, content coupling occurs when one module makes use of data or control information maintained within the boundary of another module. Secondarily, content coupling occurs when branches are made into the middle of a module. As this type of coupling makes

software complex so in BUILD CARRIER PLACEMENTS we have tried our best to avoid such coupling.

**As** the **cohesion** and **coupling** are clearly related. Usually the greater the cohesion of each module in a system, the lower the coupling between modules is. So we have maintained a balance between these two engineering concepts.

**SPECIFICATION OF MODULES:** Module specification is the major part of system design specification. All modules in the system should be identified when the system design is complete, and these modules should be specified in the document. To specify a module, the design document must specify.

1. **The abstract behavior of the module:** specifying the module’s functionality or its input/output.
2. **The interface of the module:** All data items, their types, and whether they are for input and /or output.
3. **All other modules used by the module being specified:** This information is quiet useful in maintaining and understanding the design.
   * 1. **JOB SEEKER/EMPLOYER Login Module:**

**The abstract behavior:** This module is for login by the Job Seeker/Employer. The Job Seeker/Employer can directly login through this module. This module is integrated with Restriction of unauthorized access module. Each user can only access those data, which are required for his/her work.

**Process Design Language (PDL) For This Module:**

DISPLAY SeekerID/EmployerID, Seeker/Employer PASSWORD

READ SeekerID/ EmployerID, Seeker/ Employer PASSWORD

DO WHILE NOT EOF

IF UID= SeekerID/EmployerID THEN

IF PWD= Seeker/Employer PASSWORD

THEN

DISPLAY REQUIRED PAGE

ELSE

DISPLAY “ACCESS DENIED”

END IF

END IF

END DO

**DFD for this module:**

UserId

Pwd

User Successfully

Logged into the

System

1. **New Signup / Update Profile: (Job Seeker & Employer)**

**The abstract behavior**: This module is for the Job Seeker and the Employer for New Signup or Update their exsisting Profile in “Build Carrier Placements”. The system automatically generates the Job Seeker ID and Employer ID.

**Process Design Language (PDL) For This Module:**

READ NAME, ADDRESS, PHONENO, EMAILID, EXPERIENCE, JOB CATEGORY, QUALIFICATIONS, COMAPANY PROFILE.

PROCEDURE CONFIRM ()

IF LEN (MAILID)=0 THEN

MESSAGE “ID CANNOT BE EMPTY”

END IF

IF LEN (NAME)=0 THEN

MESSAGE “NAME CANNOT BE EMPTY”

END IF

IF LEN (ADDRESS)=0 THEN

MESSAGE “ADDRESS CANNOT BE EMPTY”

IF NOT ISNUMERIC (PHONENO) THEN

MESSAGE “PHONE NUMBER SHOULD BE NUMERIC”

END IF

IF LEN (EMAILID)=0 THEN

MESSAGE “EMAIL ID CANNOT BE EMPTY”

ELSE

POSITION=INSTR (1,EMAILID,”@”)

IF POSITION=0 THEN

MESSAGE “INVALID EMAIL ID”

END IF

END IF

IF LEN (PTYPE)=0 THEN

MESSAGE “PROBLEM TYPE CANNOT BE EMPTY”

END IF

IF LEN (CALLDATE)=0 THEN

MESSAGE “CALL DATE CANNOT BE BLANK”

END IF

CONFIRM

END PROCEDURE

STORE THE INFORMATION

END

**DFD for this module:**

Valid Job-Seeker

Store

Valid

Employer

D1 Master Database

Store

Job ID

Admin

1. **Job Posting Process:**

**The abstract behavior:** This module is for the Employer. The Employer can post job on Build Carrier using this module and it will be get visible over Build Carrier when admin will verify this posted job, Employer will savi this in the Master database.

**Process Design Language (PDL) For This Module:**

PROCESS LOGIN

GENERATE NEW JOBID

POST JOB WITH COMPLETED DESCRIPTION

EDIT POSTED JOB.

**DFD for this Module:**

Valid Reported to

Admin

Employer

Posting

Pending Take Details

of Job-Id & D1 Master Database

Emp\_Id

Job

Update

1. **Job Search and Job Apply Processing:**

**The abstract behavior:** This module is for the Job Seeker. The Job Seeker will search job in this module and can apply from here itself. Admin will get request from Job Seeker and vierfy it’s request then the Employer will get informed about seeker request.

**Process Design Language (PDL) For This Module:**

SEEKER LOGIN

SEEKER OPEN SEARCH PAGE

CHECK FOR NEW POSTED JOB/SEARCH JOB

APPLY TO MATCHING REQUIRMENT

SEND THE INFORMATION TO THE ADMIN

STORE STATUS

END PROCESS

**DFD for this Module:**

Job Seeker Search Take Job Details

Send Mail D1 Master Database

Update Status

Employers

1. **Mailbox/Delete Job Process:**

**The abstract behavior:** This module is for Job Seeker. The Seeker can chaeck his/her mail and job status. He/she will get updates in his mailbox. The seeker can delete job or mail in this module.

\*\* He/She can even restore deleted job

**Process Design Language (PDL) For This Module:**

JOB/MAIL CHECK-STATUS

CHECK STATUS OF THE JOB / MAILS

IF STATUS = OK THEN

JOB APPLIED TO EMPLOYER

CLOSE THE PAGE

ELSE

SEND REMINDER TO THE ADMIN FOR NEXT TRANSACTION

END IF

IF STATUS = NOT OK THEN

DELETE THE SELECTED MAIL

END IF

END PROCESS

**DFD of this Module:**

Post and send Information

Checking status

Not approved/send

Job

Status

Detail

Send

Mail

To

Employer

Job Seeker

To

Store Data

D1 Master Database

Store not approved Details

Store Modified Details

Store Job posting Details

1. **Change of privileges Module: (ADMIN)**

**The abstract behavior:** Admin can anytime edit/update Seeker/Employer or job in this module.He can delete any job from here .

1. **Report generation Module:**

**The abstract behavior:** This module generates various reports.

**DFD for this Module:**

Report 1

Report 2

Report 3

Valid

Admin**/**

Employers/ Report 4

Job Seeker

Report 5

Take various

Details

D1 Master Database

Report 1: Report on Job Submitted by Employers

Report 2: Report on Employer (for Admin)

Report 3: Report on Job Seeker (for Admin)

Report 4: Report on job applied from Job Seeker

Report 5: Report on Current Status of Job posted so far.

**Quality Assurance**

This project has much Quality Assurance check points throughout the entire development process, to ensure all work is performed to the highest accepted industry standards and practices. During development, the main emphasis is not only on the final product quality, but also the intermediate products, even though in a project the ultimate interest is in the quality of delivered product.

Various constraints are put inside the project. These validation checks make the project work more efficiently. As discussed earlier, not only the final product is considered to be important, rather the intermediate products are also kept in mind. Constraints like primary key for each clients, stock and calls details & calls attended are worked on, so that each of them is having unique identity. Also other validation checks are performed on various attributes of the entities like name, address, phone number etc.

**Client Review**

Client involvement is important in all aspects of design and development of every successful project. To this end, client is encouraged to understand and input his reviews throughout the project. Upon completion of each phase, a client review is conducted. Reviews serve to verify that the program is instructionally sound, meets established milestones and budget and reflects the expectations and goals of the client.

**4. Control design:**

The control design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc. This will ensure that the system is functioning as per plan. One of the controls the access control is achieved by UserId and password.

## **All the steps of system design are fully implemented in the project**.

## **9. CODING**

**List of Coding:**

1. Web.Config
2. DataUtility.cs
3. City.cs
4. Default.css
5. BuildCarrier\_User.aspx.cs
6. A1Jobs\_Login.aspx.cs
7. Aboutus.aspx.cs
8. Contactus.aspx.cs
9. EmployerRegistration.aspx.cs
10. [Home.aspx.cs](http://Home.aspx.cs)
11. OurExperts.aspx.cs
12. PostJob.aspx.cs
13. RegistrationStep1.aspx.cs
14. SearchJob.aspx.cs
15. SearchResume.aspx.cs
16. SeekerProfile.aspx.cs
17. Submit.aspx.cs

## **9. 1 Web.Config**

In Web.Config; we define the ConnectionString.

This will provide a great modularity to our project.

<?xml version="1.0"?>

<!--

Note: As an alternative to hand editing this file you can use the

web admin tool to configure settings for your application. Use

the Website->Asp.Net Configuration option in Visual Studio.

A full list of settings and comments can be found in

machine.config.comments usually located in

\Windows\Microsoft.Net\Framework\v2.x\Config

-->

<configuration>

<configSections>

<sectionGroup name="system.web.extensions" type="System.Web.Configuration.SystemWebExtensionsSectionGroup, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">

<sectionGroup name="scripting" type="System.Web.Configuration.ScriptingSectionGroup, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">

<section name="scriptResourceHandler" type="System.Web.Configuration.ScriptingScriptResourceHandlerSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" allowDefinition="MachineToApplication"/>

<sectionGroup name="webServices" type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">

<section name="jsonSerialization" type="System.Web.Configuration.ScriptingJsonSerializationSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" allowDefinition="Everywhere"/>

<section name="profileService" type="System.Web.Configuration.ScriptingProfileServiceSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" allowDefinition="MachineToApplication"/>

<section name="authenticationService" type="System.Web.Configuration.ScriptingAuthenticationServiceSection,

System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" allowDefinition="MachineToApplication"/>

<section name="roleService"

type="System.Web.Configuration.ScriptingRoleServiceSection, System.Web.Extensions,

Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" allowDefinition="MachineToApplication"/></sectionGroup></sectionGroup></sectionGroup></configSections><appSettings/>

<connectionStrings>

<connectionStrings>

<add name="Harsh" connectionString="server=HARSH2007\SA;database=BuildCareer; uid=sa; password=harsh2007"/>

</connectionStrings>

<system.net>

<mailSettings>

<smtp from="sales@techinsightvision.com">

<network host="mail.techinsightvision.com" password="allowme" userName="sales@techinsightvision.com"/>

</smtp>

</mailSettings>

</system.net>

<system.web>

<customErrors mode="Off"></customErrors>

<!--

Set compilation debug="true" to insert debugging

symbols into the compiled page. Because this

affects performance, set this value to true only

during development.

-->

<compilation debug="true">

<assemblies>

<add assembly="System.Core, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/></assemblies></compilation>

<!--

The <authentication> section enables configuration

of the security authentication mode used by

ASP.NET to identify an incoming user.

-->

<authentication mode="Windows"/>

<!--

The <customErrors> section enables configuration

of what to do if/when an unhandled error occurs

during the execution of a request. Specifically,

it enables developers to configure html error pages

to be displayed in place of a error stack trace.

<customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">

<error statusCode="403" redirect="NoAccess.htm" />

<error statusCode="404" redirect="FileNotFound.htm" />

</customErrors>

-->

<pages>

<controls>

<add tagPrefix="asp" namespace="System.Web.UI" assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add tagPrefix="asp" namespace="System.Web.UI.WebControls" assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/></controls></pages>

<httpHandlers>

<remove verb="\*" path="\*.asmx"/>

<add verb="\*" path="\*.asmx" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add verb="\*" path="\*\_AppService.axd" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add verb="GET,HEAD" path="ScriptResource.axd" validate="false" type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/></httpHandlers>

<httpModules>

<add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/></httpModules></system.web>

<system.codedom>

<compilers>

<compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CSharp.CSharpCodeProvider,System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" warningLevel="4">

<providerOption name="CompilerVersion" value="v3.5"/>

<providerOption name="WarnAsError" value="false"/></compiler>

<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.VisualBasic.VBCodeProvider, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" warningLevel="4">

<providerOption name="CompilerVersion" value="v3.5"/>

<providerOption name="OptionInfer" value="true"/>

<providerOption name="WarnAsError" value="false"/></compiler></compilers></system.codedom>

<system.webServer>

<validation validateIntegratedModeConfiguration="false"/>

<modules>

<remove name="ScriptModule"/>

<add name="ScriptModule" preCondition="managedHandler" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/></modules>

<handlers>

<remove name="WebServiceHandlerFactory-Integrated"/>

<remove name="ScriptHandlerFactory"/>

<remove name="ScriptHandlerFactoryAppServices"/>

<remove name="ScriptResource"/>

<add name="ScriptHandlerFactory" verb="\*" path="\*.asmx" preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add name="ScriptHandlerFactoryAppServices" verb="\*" path="\*\_AppService.axd" preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add name="ScriptResource" verb="GET,HEAD" path="ScriptResource.axd" preCondition="integratedMode" type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/></handlers></system.webServer>

<runtime>

<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">

<dependentAssembly>

<assemblyIdentity name="System.Web.Extensions" publicKeyToken="31bf3856ad364e35"/>

<bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/></dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Extensions.Design" publicKeyToken="31bf3856ad364e35"/>

<bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="3.5.0.0"/></dependentAssembly></assemblyBinding></runtime>

</configuration>

## **9. 2 DataUtility.cs**

We have a folder in the project named ‘Component’. Here we define a class file named “DataUtility.cs”. In this file we define few functions which are used throughout the project. Name of some of the functions are ExecuteSql (string SqlStr). This function will actually execute the Sql Query which we are using from every front-end forms. Here we are also importing some namespaces.

using System;

using System.Configuration;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Data;

using System.Data.SqlClient;

using System.Data.SqlTypes;

/// <summary>

/// Summary description for DataUtility

/// </summary>

///

namespace BuildCarrier.Components

{

public class DataUtility

{

string strCon = ConfigurationManager.ConnectionStrings["Harsh"].ConnectionString;

public SqlCommand DataCom;

public SqlConnection con;

public void OpenConnection()

{

con = new SqlConnection(strCon);

con.Open();

if (con == null)

{

DataCom = new SqlCommand();

con.Open();

DataCom.Connection = con;

}

}

public void CloseConnection()

{

con = new SqlConnection(strCon);

con.Close();

}

public DataTable GetDataTable(string sql)

{

OpenConnection();

SqlDataAdapter da = new SqlDataAdapter(sql,con)

DataTable dt = new DataTable();

da.Fill(dt);

CloseConnection();

return dt;

}

public int executeSql(string strsql)

{

OpenConnection();

int val;

DataCom.CommandType = CommandType.Text;

DataCom.CommandText = strsql;

val = this.DataCom.ExecuteNonQuery();

CloseConnection();

return val;

}

}

}

## **9. 3 City.cs**

using System;

using System.Data;

using System.Configuration;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Collections;

using System.Data.SqlClient;

public class City

{

string strCon =ConfigurationManager.ConnectionStrings["Harsh"].ConnectionString;

public City()

{

//

// TODO: Add constructor logic here

//

}

public ArrayList IndCity()

{

ArrayList IndCityList = new ArrayList();

IndCityList.Add("Please Select");

IndCityList.Add("Any Where India");

IndCityList.Add("Agartala");

IndCityList.Add("Agra");

IndCityList.Add("Ahmedabad");

IndCityList.Add("Ahmednagar");

IndCityList.Add("Aizwal");

IndCityList.Add("Ajmer");

IndCityList.Add("Allahabad");

IndCityList.Add("Ambala");

IndCityList.Add("Amritsar");

IndCityList.Add("Andaman and Nicobar");

IndCityList.Add("Andhra Pradesh");

IndCityList.Add("Arunachal Pradesh");

IndCityList.Add("Assam");

IndCityList.Add("Aurangabad");

IndCityList.Add("Bangalore");

IndCityList.Add("Baroda Vadodara");

IndCityList.Add("Belgaum");

IndCityList.Add("Bhilai");

IndCityList.Add("Bhopal")

IndCityList.Add("Bhubaneshwar");

IndCityList.Add("Bihar");

IndCityList.Add("Chandigarh");

IndCityList.Add("Chattisgarh");

IndCityList.Add("Chennai Madras");

IndCityList.Add("Coimbatore");

IndCityList.Add("Cuttack");

IndCityList.Add("Dadra Nagar Haveli");

IndCityList.Add("Daman and Diu");

IndCityList.Add("Darjeeling");

IndCityList.Add("Dehradun");

IndCityList.Add("New Delhi");

IndCityList.Add("Delhi NCR");

IndCityList.Add("Dhanbad");

IndCityList.Add("Dispur");

IndCityList.Add("Erode");

IndCityList.Add("Faridabad");

IndCityList.Add("Gandhi Nagar");

IndCityList.Add("Gangtok");

IndCityList.Add("Ghaziabad");

IndCityList.Add("Goa");

IndCityList.Add("Gujarat");

IndCityList.Add("Gurgaon");

IndCityList.Add("Guwahati");

IndCityList.Add("Gwalior");

IndCityList.Add("Haridwar");

IndCityList.Add("Haryana");

IndCityList.Add("Himachal Pradesh");

IndCityList.Add("Hissar");

IndCityList.Add("Hubli");

IndCityList.Add("Hyderabad");

IndCityList.Add("Imphal");

IndCityList.Add("Indore");

IndCityList.Add("Itanagar");

IndCityList.Add("Jabalpur");

IndCityList.Add("Jaipur");

IndCityList.Add("Jalandhar");

IndCityList.Add("Jalgaon");

IndCityList.Add("Jammu");

IndCityList.Add("Jammu Kashmir");

IndCityList.Add("Jamnagar");

IndCityList.Add("Jamshedpur");

IndCityList.Add("Jharkhand");

IndCityList.Add("Jodhpur");

IndCityList.Add("Kalyan");

IndCityList.Add("Kanpur");

IndCityList.Add("Kanyakumari");

IndCityList.Add("Karnataka");

IndCityList.Add("Kerala");

IndCityList.Add("Kochi");

IndCityList.Add("Kohima");

IndCityList.Add("Kolkata");

IndCityList.Add("Kollam");

IndCityList.Add("Kottayam");

IndCityList.Add("Kullu");

IndCityList.Add("Lakshadweep");

IndCityList.Add("Ludhiana");

IndCityList.Add("Madurai");

IndCityList.Add("Madhya Pradesh");

IndCityList.Add("Manali");

IndCityList.Add("Mangalore");

IndCityList.Add("Manipur");

IndCityList.Add("Mathura");

IndCityList.Add("Meerut");

IndCityList.Add("Meghalaya");

IndCityList.Add("Mizoram");

IndCityList.Add("Mumbai");

IndCityList.Add("Mysore");

IndCityList.Add("Nagaland");

IndCityList.Add("Nagpur");

IndCityList.Add("Nainital");

IndCityList.Add("Nasik");

IndCityList.Add("Navi Mumbai");

IndCityList.Add("Noida");

IndCityList.Add("Ooty");

IndCityList.Add("Orissa");

IndCityList.Add("Others");

IndCityList.Add("Palghar");

IndCityList.Add("Panipat");

IndCityList.Add("Panjim");

IndCityList.Add("Patiala");

IndCityList.Add("Patna");

IndCityList.Add("Pondicherry");

IndCityList.Add("Port Blair");

IndCityList.Add("Pune");

IndCityList.Add("Punjab");

IndCityList.Add("Puri");

IndCityList.Add("Raipur");

IndCityList.Add("Rajasthan");

IndCityList.Add("Rajkot");

IndCityList.Add("Ranchi");

IndCityList.Add("Rourkela");

IndCityList.Add("Salem");

IndCityList.Add("Secunderabad");

IndCityList.Add("Shillong");

IndCityList.Add("Shimla");

IndCityList.Add("Sikkim");

IndCityList.Add("SHIL");

IndCityList.Add("SHIM");

IndCityList.Add("SIKK");

IndCityList.Add("Silchar");

IndCityList.Add("Siliguri");

IndCityList.Add("Sonipat");

IndCityList.Add("Srinagar");

IndCityList.Add("Surat");

IndCityList.Add("Tamil Nadu");

IndCityList.Add("Thane");

IndCityList.Add("Thiruvananthapuram");

IndCityList.Add("Tiruchirapali");

IndCityList.Add("Trichy");

IndCityList.Add("Tripura");

IndCityList.Add("Udaipur");

IndCityList.Add("Uttar Pradesh");

IndCityList.Add("Uttaranchal");

IndCityList.Add("Valsad");

IndCityList.Add("Varanasi");

IndCityList.Add("Vijayawada");

IndCityList.Add("Vishakhapatnam");

IndCityList.Add("West Bengal");

IndCityList.Add("Other");

return IndCityList;

**9.4 Default.css**

**Default.css**

‘BUILD CARRIER PLACEMENTS ‘ .css is an important .css file. This file is using here to provide better look to the project. Like the Scrollbars of project; which is ‘WHITE’. In this project only because of that BUILD CARRIER PLACEMENTS.css file. Similarly we are using for every Web control that we are used in the project. The main advantage of using this file is that if there will be any need to change the looking of the project we only have to change the coding in this BUILD CARRIER PLACMENTS .css file ; this will automatically affect the whole project.

/\*

Design by Free CSS Templates

http://www.freecsstemplates.org

Released for free under a Creative Commons Attribution 2.5 License

\*/

\* {

margin: 0;

padding: 0;

}

body {

margin: 0px;

background: #000000;

text-align: justify;

font-family: Arial, Helvetica, sans-serif;

font-size: 13px;

color: #363636;

}

h1, h2, h3 {

color: White;

}

h1 {

}

h2 {

}

h3 {

}

p, blockquote, ul, ol {

margin-bottom: 20px;

line-height: 2em;

}

p

{

}

blockquote {

}

ul, ol, li {

margin: 0px;

padding: 0px;

list-style: none;

}

a {

text-decoration: underline;

color: #74870B;

}

a:hover {

text-decoration: none;

color: #74870B;

}

/\* Wrapper \*/

#wrapper {

width: 980px;

margin: 0 auto;

background: url(images/img01.gif) repeat-y left top;

}

/\* Header \*/

#header {

width: 980px;

height: 110px;

margin: 0 auto;

background: url(images/img02.jpg) no-repeat left top;

}

/\* Logo \*/

#logo {

float: left;

width: 640px;

height: 76px;

margin: 0px;

padding: 15px 0px 0px 20px;

}

#logo h1 {

margin: 0;

padding: 0;

color:White;

font: normal 30px Georgia, "Times New Roman", Times, serif;

}

#logo h2 {

margin: -2px 0 0 0;

padding: 0px 0px 0px 3px;

text-transform: uppercase;

letter-spacing: 2px;

font-size: 10px;

font-weight: bold;

color: #444444;

}

#logo a {

text-decoration: none;

color: #FFFFFF;

}

/\* Menu \*/

#menu {

width: 660px;

height: 40px;

margin: 0 auto;

padding-bottom: 40px;

background: url(images/img03.jpg) no-repeat left top;

}

#menu ul {

margin: 0px 0px 0px 20px;

padding: 0px 0px 0px 0px;

list-style: none;

line-height: normal;

}

#menu li {

display: inline;

text-align: center;

}

#menu a {

display: block;

float: left;

height: 30px;

margin: 0px 1px 0px 0px;

padding: 10px 20px 0px 20px;

text-decoration: none;

text-align: center;

text-transform: uppercase;

font-size: 10px;

font-weight: bold;

color: #000000;

}

#menu a:hover, #menu .active a {

background: #9BAF31;

color: #FFFFFF;

}

#splash {

width: 940px;

height: 410px;

margin: 0 auto;

}

/\* Search \*/

#header-search {

width: 940px;

height: 51px;

margin: 0 auto;

padding: 0px 0px 0px 0px;

}

#header-search form {

float: right;

width: 300px;

margin: 0;

padding: 12px 0px 0px 0px;

}

#header-search fieldset {

margin: 0;

padding: 0;

border: none;

}

#header-search-text {

width: 200px;

padding: 2px 3px;

background: #1E1E1E;

border: none;

color: #4D4D4D;

}

#header-search-submit {

background: none;

border: none;

font-family: Arial, Helvetica, sans-serif;

font-size: 12px;

color: #E4E4E4;

}

#search {

height: 51px;

margin: 0 auto;

padding: 0px 0px 60px 0px;

}

#search form {

margin: 0;

padding: 12px 0px 0 0;

}

#search fieldset {

margin: 0;

padding: 0;

border: none;

}

#search input {

float: left;

font: 12px Arial, Helvetica, sans-serif;

}

#search-text {

width: 160px;

height: 18px;

margin-left: 30px;

padding: 3px 0 3px 5px;

border: none;

color: #000000;

}

#search-submit {

margin-left: 10px;

padding: 4px 4px;

border: none;

background: #9BAF31;

color: #FFFFFF;

}

/\* Page \*/

#page {

width: 980px;

margin: 0 auto;

}

#page-bgtop {

}

/\* Content \*/

#content {

float: left;

width: 660px;

}

.post {

margin: 0px 0px 30px 0px;

}

.post .title {

margin: 0px;

padding: 0px 0px 5px 0px;

color: #232F01;

}

.post .title a {

padding: 4px 35px 4px 15px;

background: #74870B url(images/img04.jpg) no-repeat right top;

text-decoration: none;

font-weight: normal;

color: #FFFFFF;

}

.post .entry {

padding: 20px 0px;

margin: 0px 20px;

border-bottom: 1px solid #C7C7C7;

}

.post img {

float: left;

padding: 15px 0px;

}

.post .meta {

text-align: right;

padding: 0px 0 20px 0;

font-weight: bold;

color: #202020;

}

.post .byline {

float: right;

margin-top: -30px;

padding-right: 20px;

font-size: 12px;

font-weight: bold;

color: #363636;

}

.post .more {

display: block;

width: 81px;

height: 30px;

background: url(images/img05.gif) no-repeat left top;

text-decoration: none;

text-transform: uppercase;

text-align: center;

font-size: 10px;

font-weight: bold;

color: #FFF;

}

/\* Sidebar \*/

#sidebar {

float: right;

width: 300px;

margin: -85px 0px 0px 0px;

padding: 0px 0px 30px 0px;

}

#sidebar ul {

margin: 0;

padding: 0;

list-style: none;

line-height: normal;

}

#sidebar li {

margin-bottom: 1px;

background: #101010;

margin-bottom: 20px;

}

#sidebar li ul {

margin: 0px;

padding: 0px 30px 40px 30px;

}

#sidebar li li {

margin: 0;

padding: 9px 0px;

border: none;

border-bottom: 1px solid #252726;

}

#sidebar h2 {

margin: 0px;

padding: 10px 0px 10px 30px;

background: #0C0C0C;

font-size: 160%;

font-weight: normal;

color: #FFFFFF;

}

#sidebar h3 {

font-size: 77%;

color: #110404;

}

#sidebar p {

margin: 0;

line-height: normal;

color: #D4C792;

}

#sidebar a {

border: none;

text-decoration: none;

color: #A1A1A1;

}

#sidebar a:hover {

text-decoration: underline;

}

/\* Submenu \*/

#submenu {

}

/\* News \*/

#news {

}

#news a {

font-size: 85%;

}

/\* Footer \*/

#footer {

width: 980px;

margin: 0 auto;

padding: 20px 0px;

color: #FFFFFF;

}

#footer p {

margin: 0;

text-align: center;

font-size: 77%;

}

#footer a {

text-decoration: underline;

}

#footer a:hover {

text-decoration: none;

}

.text\_Link {

font-family: Verdana, Arial, Helvetica, sans-serif;

font-size: 10px;

font-style: normal;

font-weight: bold;

font-variant: normal;

color: #6B6854;

text-decoration: none;

padding-right: 10px;

padding-left: 10px;

text-align: justify;

text-decoration:none;

}

.TDLabel

{background:#6B6854; color:#F3F1DF;}

.text\_about {

font-family: Verdana, Arial, Helvetica, sans-serif;

font-size: 12px;

font-style: normal;

font-weight: normal;

font-variant: normal;

color: #000000;

text-decoration: none;

padding-right: 10px;

padding-left: 10px;

text-align: justify;

}

.A1Button

{

color: #000080;

font-size: 12pt;

font-family:Verdana;

border:solid 1px #B3CAE1;

margin-left: 0px;

height: 20px;

}

## **9.5 Build Carrier\_User.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.IO;

using System.Data.SqlClient;

using System.Web.Services;

using System.Net.Mail;

using System.Net.Mime;

using BuildCarrier.Components;

public partial class BuildCarrier\_User : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

string JobCategory = string.Empty;

string Emply\_ID = string.Empty;

string Email = string.Empty;

string Exp = string.Empty;

string S\_ID = string.Empty;

int chk;

SqlCommand cmd = new SqlCommand();

public static int profile, flag;

protected void Page\_Load(object sender,EventArgs e)

{

if (Request.Cookies["SeekerID"] == null)

{

Response.Redirect("BuildCarrier\_Login.aspx");

}

if (!Page.IsPostBack)

{

fnBindJobCategory();

fnBindCount();

fnBindUserDtls("Seeker\_Inbox");

flag = 1;

Panel1.Visible = false;

pnlJobDtls.Visible = false;

lblBox.Text = "Inbox";

lblmessage.Text = "";

lblError.Text = "";

lblmessage.Visible = false;

Label1.Text = Request.Cookies["SeekerID"].Value;

}

}

private void fnBindJobCategory()

{

DataTable dtBindJobCategory = new DataTable();

string strBindJobCategory = string.Empty;

strBindJobCategory = "select Category\_ID,JobCategory from Job\_Category";

try

{

dtBindJobCategory = dut.GetDataTable(strBindJobCategory);

if (dtBindJobCategory.Rows.Count > 0)

{

ddlJobCategory.DataSource = dtBindJobCategory;

ddlJobCategory.DataTextField = "JobCategory";

ddlJobCategory.DataValueField = "Category\_ID";

ddlJobCategory.DataBind();

}

}

catch (Exception e)

{

}

}

//Number of Message fnBindCount()

private void fnBindCount()

{

try

{

DataTable dtinboxcount = new DataTable();

string strcount = "select count(\*) from Seeker\_Inbox where Seeker\_Id = " + "'" + Request.Cookies["SeekerID"].Value.ToString() + "'";

dtinboxcount = dut.GetDataTable(strcount);

if (dtinboxcount.Rows.Count > 0)

{

lblInboxCount.Text = dtinboxcount.Rows[0][0].ToString();

}

DataTable dtsencount = new DataTable();

string strsentcount = "select count(\*) from Seeker\_SentMail where Seeker\_Id = " + "'" + Request.Cookies["SeekerID"].Value.ToString() + "'";

dtsencount = dut.GetDataTable(strsentcount);

if (dtsencount.Rows.Count > 0)

{

lblOutboxCount.Text = dtsencount.Rows[0][0].ToString();

}

DataTable dtThrascount = new DataTable();

string strThrascount = "select count(\*) from Seeker\_Trash where Seeker\_Id = " + "'" + Request.Cookies["SeekerID"].Value.ToString() + "'";

dtThrascount = dut.GetDataTable(strThrascount);

if (dtsencount.Rows.Count > 0)

{

lblTrashCount.Text = dtThrascount.Rows[0][0].ToString();

}

}

catch (Exception ex)

{ }

}

//-----fnBindJobDtls()---------------

private void fnBindJob()

{

try

{

DataTable dtJob = new DataTable();

string strJob;

if (chk != 1)

{

strJob = "select \* from Job\_Details where Job\_Category='" + ddlJobCategory.SelectedItem.Text + "' and JobStatus=1 order by exp\_req desc ";

dtJob = dut.GetDataTable(strJob);

if (dtJob.Rows.Count > 0)

{

pnlJobDtls.Visible = true;

Panel1.Visible = false;

pnlInfo.Visible = false;

GVJobDtls.DataSource = dtJob;

GVJobDtls.DataBind();

pnlapplyControl.Visible = true;

}

else

{

GVJobDtls.Dispose();

GVJobDtls.DataBind();

dtJob.Dispose();

lblError.Text = "There are no Job Here in this Section....";

pnlapplyControl.Visible = false;

}

}

else

{

strJob = "select \* from Job\_Details where Job\_Category='" + JobCategory + "' and exp\_req='" + Exp + "' and Job\_Status=1 order by exp\_req desc ";

dtJob = dut.GetDataTable(strJob);

if (dtJob.Rows.Count > 0)

{

lblBox.Text = "";

pnlJobDtls.Visible = true;

Panel1.Visible = false;

pnlInfo.Visible = false;

GVJobDtls.DataSource = dtJob;

GVJobDtls.DataBind();

pnlapplyControl.Visible = true;

}

else

{

GVJobDtls.Dispose();

GVJobDtls.DataBind();

dtJob.Dispose();

lblError.Text = "There are no Job Here in this Section....";

pnlapplyControl.Visible = false;

}

}

}

catch (Exception ex)

{ }

}

private void fnBindUserDtls(string tablename)

{

string UserDtls=string.Empty;

try

{

lblmessage.Text = "";

//flag = 1;

lblMenuUser.Text = Request.Cookies["UserName"].Value.ToString();

S\_ID = Request.Cookies["SeekerID"].Value.ToString();

if (flag == 0 || flag == 1)

{

UserDtls = "select Msg\_From,Seeker\_Id,Msg\_Subject,Msg\_Date from " + tablename + " where Seeker\_Id = " + "'" + S\_ID + "'";

}

else if (flag == 2)

{

UserDtls = "select Msg\_From,Seeker\_Id,Msg\_Subject,Msg\_Date from " + tablename + " where Seeker\_Id = " + "'" + S\_ID + "'"; ;

}

else if (flag == 3)

{

UserDtls = "select Msg\_From,Seeker\_Id,Msg\_Subject,Msg\_Date from " + tablename + " where Seeker\_Id = " + "'" + S\_ID + "'"; ;

}

DataTable dtUserDtls = new DataTable();

dtUserDtls=dut.GetDataTable(UserDtls);

if (dtUserDtls.Rows.Count>0)

{

GridView1.DataSource = dtUserDtls;

GridView1.DataBind();

PnlControl.Visible = true;

}

else

{

PnlControl.Visible = false;

lblmessage.Visible = true;

lblmessage.Text = "Records Empty";

GridView1.DataSource = null;

GridView1.DataBind();

}

}

catch (Exception ex)

{ }

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

JobCategory = "IT Sector";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

JobCategory = "HR Sector";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

JobCategory = "B P O";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

JobCategory = "Telecome";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

JobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

JobCategory = "Medical";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

JobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

JobCategory = "Garment";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

JobCategory = "Fashion";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void btnCheckAll\_Click(object sender, EventArgs e)

{

foreach (GridViewRow gvr in GridView1.Rows)

{

CheckBox cb = (CheckBox)gvr.FindControl("CheckBox1");

cb.Checked = true;

}

}

protected void btnUnCheckAll\_Click(object sender, EventArgs e)

{

foreach (GridViewRow gvr in GridView1.Rows)

{

CheckBox cb = (CheckBox)gvr.FindControl("CheckBox1");

cb.Checked = false;

}

}

protected void btnDeleteCheck\_Click(object sender, EventArgs e)

{

try

{

string currentRow;

ArrayList Fromlist = new ArrayList();

foreach (GridViewRow row in GridView1.Rows)

{

CheckBox cb = (CheckBox)row.FindControl("CheckBox1");

if (cb.Checked)

{

btnDeleteCheck.Enabled = true;

currentRow = row.Cells[1].Text;

Fromlist.Add(currentRow);

}

else

{

lblmessage.Text = "Please Select Job for Apply..";

}

}

int i;

string strID = string.Empty;

for (i = 0; i < Fromlist.Count; i++)

{

strID += Fromlist[i].ToString() + ",";

}

if (flag == 1)

{

DataTable Dt = new DataTable();

string strQuerDel = "sp\_InboxDel '" + strID + "'";

Dt = dut.GetDataTable(strQuerDel);

if (Dt.Rows.Count > 0)

{

fnBindUserDtls("Seeker\_Inbox");

}

else

{ }

}

else if (flag == 3)

{

DataTable Dt = new DataTable();

string strQuerDel = "sp\_ThrashDel'" + strID + "'";

Dt = dut.GetDataTable(strQuerDel);

if (Dt.Rows.Count>0)

{

fnBindUserDtls("Seeker\_Trash");

}

else

{ }

}

else if (flag == 2)

{

DataTable Dt = new DataTable();

string strQuerDel = "sp\_SentBoxDel '" + strID + "'";

Dt = dut.GetDataTable(strQuerDel);

if (Dt.Rows.Count> 0)

{

fnBindUserDtls("Seeker\_SentMail");

}

}

}

catch (Exception ex)

{ }

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

string strsql = string.Empty;

Panel1.Visible = true;

int index = Convert.ToInt32(e.CommandArgument);

if (e.CommandName == "sub")

{

GridViewRow row = GridView1.Rows[index];

string id = Convert.ToString(row.Cells[2].Text);

if (lblBox.Text == "Inbox")

{

strsql = "select \* from Seeker\_Inbox where Seeker\_Id='" + id + "'";

}

else if (lblBox.Text == "Sent Mail")

{

strsql = "select \* from Seeker\_SentMail where Seeker\_Id='" + id + "'";

}

else

{

strsql = "select \* from Seeker\_Trash where Seeker\_Id='" + id + "'";

}

DataTable Dt = new DataTable();

Dt = dut.GetDataTable(strsql);

if (Dt.Rows.Count>0)

{

Panel1.Visible = true;

lblmsgFrom.Text = Dt.Rows[0]["Msg\_From"].ToString();

lblmsgTo.Text = Dt.Rows[0]["Seeker\_Id"].ToString();

lblSubject.Text =Dt.Rows[0]["Msg\_Subject"].ToString();

txtSubject.Text = Dt.Rows[0]["Msg\_Desc"].ToString();

}

else

{

Panel1.Visible = false;

}

}

}

protected void GridView1\_SelectedIndexChanging(object sender, GridViewSelectEventArgs e)

{

GridView1.PageIndex = e.NewSelectedIndex;

fnBindUserDtls("TableName");

}

protected void lnkbtnInbox\_Click(object sender, EventArgs e)

{

flag = 1;

pnlInfo.Visible = true;

pnlJobDtls.Visible = false;

lblBox.Text = "Inbox";

fnBindUserDtls("Seeker\_Inbox");

Panel1.Visible = false;

}

protected void lnkbtnOutbox\_Click(object sender, EventArgs e)

{

pnlInfo.Visible = true;

pnlJobDtls.Visible = false;

flag = 2;

lblBox.Text = "Sent Mail";

fnBindUserDtls("Seeker\_SentMail");

Panel1.Visible = false;

}

protected void lnkbtnThrash\_Click(object sender, EventArgs e)

{

pnlInfo.Visible = true;

pnlJobDtls.Visible = false;

flag = 3;

lblBox.Text = "Thrash";

fnBindUserDtls("Seeker\_Trash");

Panel1.Visible = false;

}

protected void ImgBtnProfile\_Click(object sender, ImageClickEventArgs e)

{

profile = 1;

Response.Redirect("Registration\_step1.aspx");

}

protected void lblInboxCount\_Click(object sender, EventArgs e)

{

pnlInfo.Visible = true;

pnlJobDtls.Visible = false;

lblBox.Text = "Inbox";

fnBindUserDtls("Seeker\_Inbox");

Panel1.Visible = false;

}

protected void lblOutboxCount\_Click(object sender, EventArgs e)

{

pnlInfo.Visible = true;

pnlJobDtls.Visible = false;

lblBox.Text = "Sent Mail";

fnBindUserDtls("Seeker\_SentMail");

Panel1.Visible = false;

}

protected void lblTrashCount\_Click(object sender, EventArgs e)

{

pnlInfo.Visible = true;

pnlJobDtls.Visible = false;

lblBox.Text = "Trash";

fnBindUserDtls("Seeker\_Trash");

}

protected void lnkEdit\_Click(object sender, EventArgs e)

{

profile = 1;

Response.Redirect("Registration\_step1.aspx");

}

protected void GVJobDtls\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GVJobDtls.PageIndex = e.NewPageIndex;

}

string strID;

protected void btnSelectedApply\_Click(object sender, EventArgs e)

{

DataTable DtGridValue = new DataTable();

string CrRow;

ArrayList frmList = new ArrayList();

foreach (GridViewRow row in GVJobDtls.Rows)

{

CheckBox cb = (CheckBox)row.FindControl("chkJobAply");

if (cb.Checked)

{

CrRow = row.Cells[1].Text;

Emply\_ID = CrRow;

frmList.Add(CrRow);

}

else

{

lblError.Text = "There is No Job Selected";

}

}

int i;

for (i = 0; i < frmList.Count; i++)

{

strID += frmList[i].ToString() + ",";

}

fnJobApplyed();

}

private void fnJobApplyed()

{

try

{

DataTable Dt = new DataTable();

string idemp = string.Empty;

idemp = "select Emp\_ID from Job\_Details where Job\_Id='" +Emply\_ID+"'";

Dt=dut.GetDataTable(idemp);

if(Dt.Rows.Count>0)

{

Emply\_ID=Dt.Rows[0]["Emp\_Id"].ToString();

}

lblError.Text = "";

if (Request.Cookies["SeekerID"].Value != null)

{

string strinsert = "sp\_jobTransaction"+"'"+ strID +"'"+","

+"'"+Emply\_ID+"'"+","

+"'"+ Request.Cookies["SeekerID"].Value + "'";

Dt = dut.GetDataTable(strinsert);

if (Dt.Rows.Count < 0)

{

lblError.Text = "Please select the Job.";

}

else

{

lblError.Text = "Job Applied";

}

}

else

Response.Redirect("Registration\_step1.aspx");

}

catch (Exception ex)

{ }

}

protected void btnAllApply\_Click(object sender, EventArgs e)

{

foreach (GridViewRow gvr in GVJobDtls.Rows)

{

CheckBox cb = (CheckBox)gvr.FindControl("chkJobAply");

cb.Checked = true;

}

DataUtility DutGridValue = new DataUtility();

DataTable DtGridValue = new DataTable();

string CrRow;

ArrayList frmList = new ArrayList();

foreach (GridViewRow row in GVJobDtls.Rows)

{

CheckBox cb = (CheckBox)row.FindControl("chkJobAply");

if (cb.Checked)

{

CrRow = row.Cells[1].Text;

Emply\_ID = CrRow;

frmList.Add(CrRow);

}

else

{

lblError.Text = "There is No Job Selected";

}

}

int i;

for (i = 0; i < frmList.Count; i++)

{

strID += frmList[i].ToString() + ",";

}

fnJobApplyed();

}

protected void LinkButton1\_Click(object sender, EventArgs e)

{

if (Request.Cookies["SeekerID"] != null)

Response.Cookies["SeekerID"].Expires = DateTime.Now.AddYears(-30);

if (Request.Cookies["UserName"] != null)

Response.Cookies["UserName"].Expires = DateTime.Now.AddYears(-30);

Response.Redirect("Home.aspx");

}

protected void btnSearch\_Click(object sender, EventArgs e)

{

chk = 1;

if (ddlJobCategory.SelectedItem.Text == "Job Category")//|| ddlExperience.Text == "Experience")

{ }

else

{

JobCategory = ddlJobCategory.SelectedItem.Text;

Exp = ddlExperience.SelectedItem.Value;

fnBindJob();

}

}

protected void btnReply\_Click(object sender, EventArgs e)

{

DataTable dtSent = new DataTable();

string crDate = DateTime.Now.ToShortDateString();

string id = Request.Cookies["SeekerID"].Value.ToString();

string strSentInsert = " insert into Seeker\_SentMail (Seeker\_Id,MsgFrom,Mail\_Id,Msg\_Subject,Msg\_Desc,Msg\_Date)values('" + id + "','" +

this.lblmsgFrom.Text + "','" + this.lblmsgTo.Text + "','" + this.lblSubject.Text + "','" + this.txtSubject.Text + "','" + crDate + "')";

dtSent = dut.GetDataTable(strSentInsert);

if (dtSent.Rows.Count > 0)

{ }

}

}

**9.6** **A1Jobs*\_Login.aspx.cs***

Any Authorized user can get enter through this (BuildCarrier\_Login.aspx) only.

Here we are importing the ‘component’ folder, So that we can use all the function which we defined in ‘DataUtility.cs’ (A Class file created in the component folder). We can use any function in any form by defining a object.

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Net.Mail;

using System.Net.Mime;

using System.IO;

using System.Text;

using BuildCarrier.Components;

public partial class BuildCarrier\_Login : System.Web.UI.Page

{

string JobCategory = string.Empty;

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

pnlpwdforget.Visible = false;

if (Request.QueryString["PWD"] != null)

{

string Status = string.Empty;

try

{

Status = Request.QueryString["PWD"].ToString();

if (Status != "")

{

DataTable dtLogin = new DataTable();

string User\_FName = string.Empty;

string User\_LName = string.Empty;

string User\_Name = string.Empty;

string User\_Email = string.Empty;

string strLogin = "select Seeker\_Id,First\_Name,Last\_Name,Password,Email from jobseeker\_profile WHERE Seeker\_Id = " + "'"

+ Request.QueryString["ID"].ToString() +

"' AND Password = " + "'" + Request.QueryString["PWD"].ToString() + "'";

dtLogin = dut.GetDataTable(strLogin);

if (dtLogin.Rows.Count > 0)

{

Response.Cookies["SeekerID"].Value = dtLogin.Rows[0]["Seeker\_Id"].ToString();

User\_FName = dtLogin.Rows[0]["First\_Name"].ToString();

User\_LName = dtLogin.Rows[0]["Last\_Name"].ToString();

User\_Email = dtLogin.Rows[0]["Email"].ToString();

User\_Name = User\_FName + " " + User\_LName;

Response.Cookies["UserName"].Value = User\_Name.ToString();

Response.Cookies["UserEmail"].Value = User\_Email.ToString();

Response.Redirect("BuildCarrier\_Login.aspx");

}

else

Response.Redirect("http://www.techinsightvision.com/IVT\_login.aspx");

}

}

catch (Exception ex)

{

Response.Write(ex.ToString());

}

}

}

}

potected void lnkbtnForgetPwd\_Click(object sender, EventArgs e)

{

pnlpwdforget.Visible = true;

}

protected void lnkbtnNewUser\_Click(object sender, EventArgs e)

{

if (ddluserType.SelectedItem.Text == "Employeer")

{

Response.Redirect("EmployerRegistration.aspx");

}

else

Response.Redirect("Registration\_step1.aspx");

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

JobCategory = "IT Sector";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

JobCategory = "HR Sector";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

JobCategory = "B P O";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

JobCategory = "Telecome";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

JobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

JobCategory = "Medical";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

JobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

JobCategory = "Garment";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

JobCategory = "Fashion";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void btnCancel\_Click(object sender, EventArgs e)

{

txtUserId.Text = "";

txtPwd.Text = "";

txtUserId.Focus();

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

if (txtUserId.Text != "" && txtPwd.Text != "")

{

if (ddluserType.SelectedItem.Text == "Job Seeker")

{

DataTable dtLogin = new DataTable();

string User\_FName = string.Empty;

string User\_LName = string.Empty;

string User\_Name = string.Empty;

string User\_Email = string.Empty;

string strLogin = "select Seeker\_Id,First\_Name,Last\_Name,Password,Email from jobseeker\_profile WHERE Seeker\_Id =" + "'" + this.txtUserId.Text + "' AND Password = " + "'" + this.txtPwd.Text + "'";

dtLogin = dut.GetDataTable(strLogin);

if (dtLogin.Rows.Count > 0)

{

Response.Cookies["SeekerID"].Value = dtLogin.Rows[0]["Seeker\_Id"].ToString();

User\_FName = dtLogin.Rows[0]["First\_Name"].ToString();

User\_LName = dtLogin.Rows[0]["Last\_Name"].ToString();

User\_Email = dtLogin.Rows[0]["Email"].ToString();

User\_Name = User\_FName + " " + User\_LName;

Response.Cookies["UserName"].Value = User\_Name.ToString();

Response.Cookies["UserEmail"].Value = User\_Email.ToString();

Response.Redirect("BuildCarrier\_User.aspx");

}

else

{

lblErrormsg.Text = "Invalid User ID and Password !";

txtUserId.Text = "";

txtPwd.Text = "";

txtUserId.Focus();

}

}

else

{

DataUtility dutLogin = new DataUtility();

DataTable dtLogin = new DataTable();

string strLogin = "select id, password from employers WHERE id = " + "'" + this.txtUserId.Text + "' AND password = " + "'" + this.txtPwd.Text + "'";

dtLogin = dutLogin.GetDataTable(strLogin);

if (dtLogin.Rows.Count > 0)

{

Response.Cookies["EmployerID"].Value = Convert.ToString(dtLogin.Rows[0]["id"]);

Response.Redirect("PostJob.aspx");

}

else

{

lblErrormsg.Text = "Invalid User ID and Password !";

txtUserId.Text = "";

txtPwd.Text = "";

txtUserId.Focus();

}

}

}

else

{

lblErrormsg.Text = "Invalid User ID and Password !";

txtUserId.Text = "";

txtPwd.Text = "";

txtUserId.Focus();

}

}

string id, pwd;

protected void btnpwdGo\_Click(object sender, EventArgs e)

{

try

{

lblpwdError.Text = "";

if (txtEmailid.Text != "")

{

DataUtility dutpwd = new DataUtility();

DataTable dtpwd = new DataTable();

string strpwd = "select Seeker\_Id,Password,Email from JobSeeker\_Profile where Email='" + txtEmailid.Text + "'";

dtpwd = dutpwd.GetDataTable(strpwd);

if (dtpwd.Rows.Count > 0)

{

id = dtpwd.Rows[0]["Seeker\_Id"].ToString();

pwd = dtpwd.Rows[0]["Password"].ToString();

UserQuery();

}

else

lblpwdError.Text = "You are not member of Build Carrier.Plz Subscribe for LogIN !";

}

else

lblpwdError.Text = "Email ID compulsory.. !";

}

catch (Exception ex)

{

}

}

public void UserQuery()

{

string strbody = "<html><body><table><tr><td><b><font color=black> Login ID : </font></b><font color=blue>" + id;

strbody += "</font><tr><td><b><font color=black> Login Password : </font></b><font color=blue>" + pwd;

strbody = strbody + "</p></td></tr></table></html>";

SmtpClient smtpClient = new SmtpClient();

MailMessage message = new MailMessage();

try

{

MailAddress fromAddress = new MailAddress("amit81.vsnl@gmial.com.com");

smtpClient.Host = "webmail.techinsightvision.com";

smtpClient.Port = 25;

message.From = fromAddress;

message.To.Add(this.txtEmailid.Text);

message.Subject = "Build Carrier account details.";

message.IsBodyHtml = true;

message.Body = strbody;

smtpClient.Send(message);

lblpwdError.Visible = true;

lblpwdError.Text = "Your Login Details Has been Sent in your mail id";

}

catch (Exception ex)

{

lblpwdError.Visible = true;

lblpwdError.Text = ex.Message.ToString();

}

}

}

## **9. 7 Aboutus.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class Aboutus : System.Web.UI.Page

{

string jobCategory = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

jobCategory = "IT Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

jobCategory = "HR Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

jobCategory = "B P O";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

jobCategory = "Telecome";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

jobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

jobCategory = "Medical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

jobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

jobCategory = "Garment";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

jobCategory = "Fashion";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

}

## **9. 8 Contact.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Text;

using System.Web.Services;

using System.Net.Mail;

using System.Net.Mime;

using System.IO;

using System.Text.RegularExpressions;

using BuildCarrier.Components;

public partial class Contactus : System.Web.UI.Page

{

string jobCategory = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

fnBindCity();

}

}

private void fnBindCity()

{

City cityList = new City();

ArrayList indcity = new ArrayList();

ddlCity.Items.Clear();

indcity = cityList.IndCity();

for (int i = 0; i < indcity.Count; i++)

{

ddlCity.Items.Add(indcity[i].ToString());

}

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

jobCategory = "IT Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

jobCategory = "HR Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

jobCategory = "B P O";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

jobCategory = "Telecome";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

jobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

jobCategory = "Medical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

jobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

jobCategory = "Garment";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

jobCategory = "Fashion";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void btnsubmit\_Click(object sender, EventArgs e)

{

if (txtEmail.Text != "" & txtContact.Text != "" & txtSubject.Text != "")

{

btnsubmit.Attributes.Add("onclick", "return validate()");

string pattern = @"^[a-z][a-z|0-9|]\*([\_][a-z|0-9]+)\*([.][a-z|" +

@"0-9]+([\_][a-z|0-9]+)\*)?@[a-z][a-z|0-9|]\*\.([a-z]" +

@"[a-z|0-9]\*(\.[a-z][a-z|0-9]\*)?)$";

System.Text.RegularExpressions.Match match =

Regex.Match(txtEmail.Text.Trim(), pattern, RegexOptions.IgnoreCase);

if (match.Success)

{

lblErrormsg.Text = "";

}

else

lblErrormsg.Text = "Invalid Email ID..";

SendUserQuery();

}

else

{

lblErrormsg.Text = "Some Fields are blank";

}

}

public void SendUserQuery()

{

string strbody = "<html><body><table><tr><td><b><font color=black> Name : </font></b><font color=blue>" + txtName.Text;

strbody += "</font><tr><td><b><font color=black> Phone No. : </font></b><font color=blue>" + txtContact.Text;

strbody += "</font><tr><td><b><font color=black> Email - ID : </font></b><font color=blue>" + txtEmail.Text;

strbody += "</font><tr><td><b><font color=black> Subject : </font></b><font color=blue>" + txtSubject.Text;

strbody += "</font><tr><td><b><font color=black> Reason For Contact . : </font></b><font color=blue>" + txtDescription.Text;

strbody = strbody + "</p></td></tr></table></html>";

SmtpClient smtpClient = new SmtpClient();

MailMessage message = new MailMessage();

try

{

MailAddress fromAddress = new MailAddress(txtEmail.Text);

smtpClient.Host = "webmail.techinsightvision.com";

//smtpClient.Host = "mail.techinsightvision.com";

smtpClient.Port = 25;

message.From = fromAddress;

message.To.Add("info@techinsightvision.com");

message.Subject = txtSubject.Text;

message.IsBodyHtml = true;

message.Body = strbody;

smtpClient.Send(message);

lblErrormsg.Visible = true;

lblErrormsg.Text = "Your Mail Has been Sent Successfully....";

reset();

}

catch (Exception ex)

{

lblErrormsg.Visible = true;

lblErrormsg.Text = ex.Message.ToString();

}

}

private void reset()

{

this.txtName.Text = "";

this.txtContact.Text = "";

this.txtEmail.Text = "";

this.txtSubject.Text = "";

this.txtDescription.Text = "";

this.txtName.Focus();

}

}

## **9.9 EmployerRegistration.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.IO;

using System.Data.SqlClient;

using System.Net.Mail;

using System.Net.Mime;

using System.Text;

using BuildCarrier.Components;

public partial class EmployerRegistration : System.Web.UI.Page

{

//int strNewRowID = 0;

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

if (Request.Cookies["EmployerID"] != null)

fnBindEmploerDtls();

else

{

#region -- New Employer Id

try

{

DataTable dtGetEmpID = new DataTable();

string strGetEmpID = string.Empty;

strGetEmpID = "GetEmpID";

dtGetEmpID = dut.GetDataTable(strGetEmpID);

if (dtGetEmpID.Rows.Count != 0)

{

lblHiddenEmpId.Text = dtGetEmpID.Rows[0]["NewEmpID"].ToString();

this.lblEmpID.Text = dtGetEmpID.Rows[0]["NewEmpID"].ToString();

//dtGetEmpID.Rows[0]["NewEmpID"].ToString();

}

else

{ }

}

catch (Exception ex)

{

}

#endregion

}

}

}

#region -- fnBindEmploerDtls()

private void fnBindEmploerDtls()

{

try

{

DataTable dtGetEmployersDtls = new DataTable();

string strGetEmpDtls = RowID,Employer\_Id,E\_Name,Com\_Name,Com\_Address,City,State,Contact\_No,Com\_Website,Email\_Id FROM Employer where Employer\_Id

= " + "'" + Request.Cookies["EmployerID"].Value + "'";

dtGetEmployersDtls = dut.GetDataTable(strGetEmpDtls);

if (dtGetEmployersDtls.Rows.Count != 0)

{

this.lblEmpID.Text = dtGetEmployersDtls.Rows[0]["Employer\_id"].ToString();

this.txtCmpName.Text = dtGetEmployersDtls.Rows[0]["com\_name"].ToString();

this.txtCompanyAddress.Text = dtGetEmployersDtls.Rows[0]["com\_address"].ToString();

this.txtCompanyURL.Text = dtGetEmployersDtls.Rows[0]["com\_website"].ToString();

this.txtCompNumber.Text = dtGetEmployersDtls.Rows[0]["contact\_no"].ToString();

this.txtEmpName.Text = dtGetEmployersDtls.Rows[0]["e\_name"].ToString();

this.txtEmpMailID.Text =

dtGetEmployersDtls.Rows[0]["email\_id"].ToString();

this.txtEmpAddress.Text = dtGetEmployersDtls.Rows[0]["comp\_address"].ToString();

this.txtEmpContactNumber.Text = dtGetEmployersDtls.Rows[0]["Contact\_No"].ToString();

}

else

{ }

}

catch (Exception ex)

{

}

}

#endregion

protected void btnSave\_Click(object sender, EventArgs e)

{

if (this.txtCmpName.Text == "" || this.txtCompanyAddress.Text == "" || this.txtCompanyURL.Text == "" || this.txtCompNumber.Text == "" || this.txtEmpAddress.Text == "" || this.txtEmpContactNumber.Text == "" || this.txtEmpMailID.Text == "" || this.txtEmpName.Text == "" || this.txtEmpPassword.Text == "" || this.txtEmpPassword2.Text == "")

btnSave.Attributes.Add("onclick", "return Validate()");

else

{

//dut

if (Request.Cookies["A1EmployerID"] == null)

{

string strInsert = string.Empty;

strInsert = "INSERT INTO Employers(RowID, id, e\_name, comp\_address, Contact\_No, com\_name, com\_address, contact\_no, com\_website, email\_id, password)" +

" VALUES (" + Convert.ToInt16(lblHiddenEmpId.Text) + "," + "'" + this.lblEmpID.Text + "'" + "," + "'" + this.txtEmpName.Text + "'" + "," + "'" + this.txtEmpAddress.Text + "'" + "," +

"'" + this.txtEmpContactNumber.Text + "'" + "," + "'" + this.txtCmpName.Text + "'" + "," + "'" + this.txtCompanyAddress.Text + "'" + "," + "'" + this.txtCompNumber.Text + "'" +

"," + "'" + this.txtCompanyURL.Text + "'" + "," + "'" + this.txtEmpMailID.Text + "'" + "," + "'" + this.txtEmpPassword2.Text + "')";

try

{

int InsertResponse = dut.executeSql(strInsert);

if (InsertResponse != 0)

{

this.lblmsg.Text = "Profile Created Sucessfully, Login Credentials has been sent to your mailid";

fnSendDtls();

fnClearFields();

}

}

catch (Exception ex)

{ }

}

else

{

string strUpdate = string.Empty;

strUpdate = "UPDATE Employer SET e\_name = " + "'" + this.txtEmpName.Text + "'" + "," + "com\_address = " + "'" + this.txtEmpAddress.Text + "'" + "," +

"Contact\_No = " + "'" + this.txtEmpContactNumber.Text + "'" + "," + "com\_name = " + "'" + this.txtCmpName.Text + "'" + "," + " com\_address = "

+ "'" + this.txtCompanyAddress.Text + "'" + "," + "contact\_no = " + "'" + this.txtEmpContactNumber.Text + "'" + "," + "com\_website = " +

"'" + this.txtCompanyURL.Text + "'" + "," + "email\_id = " + "'" + this.txtEmpMailID.Text + "'" + "WHERE Employer\_id = " + "'" +

Request.Cookies["EmployerID"].Value + "'";

try

{

int ResponseUpdate = dut.executeSql(strUpdate);

if (ResponseUpdate != 0)

{

this.lblmsg.Text = "Profile Updated Sucessfully..";

fnClearFields();

}

else

{ }

}

catch (Exception ex)

{ }

}

}

}

#region -------fnSendDtls()

private void fnSendDtls()

{

string strbody = "<html><body><table><tr><td><b><font color=black> Login ID : </font></b><font color=blue>" + lblEmpID.Text;

//strbody += "</font><tr><td><b><font color=black> Login - ID : </font></b><font color=blue>" + id;

strbody += "</font><tr><td><b><font color=black> Login Password : </font></b><font color=blue>" + txtEmpPassword2.Text;

strbody = strbody + "</p></td></tr></table></html>";

SmtpClient smtpClient = new SmtpClient();

MailMessage message = new MailMessage();

try

{

MailAddress fromAddress = new MailAddress("sales@techinsightvision.com");

smtpClient.Host = "webmail.techinsightvision.com";

smtpClient.Port = 25;

message.From = fromAddress;

message.To.Add(this.txtEmpMailID.Text);

//message.To.Add("ibrar\_star@yahoo.com");

message.Subject = "A1 Jobs account details.";

//message.Subject = id;

message.IsBodyHtml = true;

message.Body = strbody;

smtpClient.Send(message);

//lblpwdError.Visible = true;

//lblpwdError.Text = "Your Login Details Has been Sent in your mail id";

//reset();

}

catch (Exception ex)

{

//lblpwdError.Visible = true;

lblmsg.Text = ex.Message.ToString();

}

}

#endregion

#region ------------fnClearFields-------

private void fnClearFields()

{

this.txtCmpName.Text = "";

this.txtCompanyURL.Text = "";

this.txtCompNumber.Text = "";

this.txtCompanyAddress.Text = "";

this.txtEmpName.Text = "";

this.txtEmpAddress.Text = "";

this.txtEmpContactNumber.Text = "";

this.txtEmpMailID.Text = "";

this.txtEmpPassword.Text = "";

this.txtEmpPassword2.Text = "";

this.txtCmpName.Focus();

}

#endreg

## **9. 10 Home.aspx.cs**

Using System;

Using Sysetm.data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using A1Jobs.Components;

using BuildCarrier.Components;

public partial class Home : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

Response.Cookies["SeekerID"].Expires = DateTime.Now.AddYears(-1);

fnBindJobCategory();

}

}

protected void btnLogin\_Click(object sender, EventArgs e)

{

if (this.txtUserID.Text != "" && this.txtPassword.Text != "")

{

DataTable Dt = new DataTable();

string GetUserDtls = string.Empty;

string User\_FName = string.Empty;

string User\_LName = string.Empty;

string User\_Name = string.Empty;

GetUserDtls = "SELECT Seeker\_Id,First\_Name,Last\_Name FROM jobseeker\_profile WHERE Seeker\_Id = " + "'" + this.txtUserID.Text + "' AND Password = " + "'" + this.txtPassword.Text + "'";

Dt = Dut.GetDataTable(GetUserDtls);

if (Dt.Rows.Count > 0)

{

Response.Cookies["SeekerID"].Value = Dt.Rows[0]["Seeker\_Id"].ToString();

User\_FName = Dt.Rows[0]["First\_Name"].ToString();

User\_LName = Dt.Rows[0]["Last\_Name"].ToString();

User\_Name = User\_FName + " " + User\_LName;

Response.Cookies["UserName"].Value = User\_Name.ToString();

Response.Redirect("BuildCarrier\_User.aspx");

}

else

{

this.lblMsg.Text = "Invalid Login Credentials..";

this.txtPassword.Text = "";

this.txtUserID.Text = "";

this.txtUserID.Focus();

}

}

else {

this.lblMsg.Text = "Please Enter your UserID / Password";

}

}

protected void lnkEmployerLogin\_Click(object sender, EventArgs e)

{

this.pnlEmpLogin.Visible = true;

this.txtELoginID.Focus();

}

protected void imgEClose\_Click(object sender, ImageClickEventArgs e)

{

this.pnlEmpLogin.Visible = false;

}

protected void btnELogin\_Click(object sender, EventArgs e)

{

if (this.txtELoginID.Text == "" || this.txtEPassword.Text == "")

{

btnELogin.Attributes.Add("onclick", "return alert('LoginID & Password are Mandatory');");

this.txtELoginID.Focus();

}

else

{

try

{

DataTable dtEmpLogin = new DataTable();

string strEmpLogin = "SELECT id FROM Employers WHERE employer\_id = " + "'" + this.txtELoginID.Text + "'" + " and Password=" + "'" + this.txtEPassword.Text + "'";

dtEmpLogin = Dut.GetDataTable(strEmpLogin);

if (dtEmpLogin.Rows.Count != 0)

{

Response.Cookies["EmployerID"].Value = Convert.ToString(dtEmpLogin.Rows[0]["id"]);

Response.Redirect("PostJob.aspx");

}

else

{

}

}

catch (Exception ex)

{ }

}

}

protected void lbtnNewUser\_Click(object sender, EventArgs e)

{

if (Request.Cookies["SeekerID"] != null)

Request.Cookies["SeekerID"].Expires = DateTime.Now.AddYears(-30);

Response.Redirect("Registration\_step1.aspx");

}

protected void btnSearch\_Click(object sender, EventArgs e)

{

//if (ddlJobCategory.Text == "Job Category" )//|| ddlExperience.Text == "Experience")

//{ }

//else

//{

// Response.Cookies["Category"].Value = ddlJobCategory.SelectedItem.Text;

// Response.Cookies["Experience"].Value =

ddlExperience.SelectedItem.Text;

// Response.Redirect("SearchJob.aspx");

//}

}

protected void lnkNewEmployer\_Click(object sender, EventArgs e)

{

if (Request.Cookies["EmployerID"] != null)

Response.Cookies["EmployerID"].Expires = DateTime.Now.AddYears(-30);

Response.Redirect("EmployerRegistration.aspx");

}

#region -- fnBindJobCategory()

private void fnBindJobCategory()

{

DataTable dtBindJobCategory = new DataTable();

string strBindJobCategory = string.Empty;

strBindJobCategory = "SELECT CategoryID, JobCategory FROM JobCategory Order By CategoryID";

try

{

dtBindJobCategory = Dut.GetDataTable(strBindJobCategory);

if (dtBindJobCategory.Rows.Count > 0)

{

this.ddlJobCategory.DataSource = dtBindJobCategory;

this.ddlJobCategory.DataTextField = "JobCategory";

this.ddlJobCategory.DataValueField = "CategoryID";

this.ddlJobCategory.DataBind();

}

else

{

}

}

catch (Exception ex)

{ }

}

#endregion

string JobCategory = string.Empty;

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

JobCategory = "IT Sector";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

JobCategory = "HR Sector";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

JobCategory = "B P O";

Response.Cookies["Job"].Value = JobCategory.ToString(); Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

JobCategory = "Telecome";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

JobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

JobCategory = "Medical";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

JobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

JobCategory = "Garment";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

JobCategory = "Fashion";

Response.Cookies["Job"].Value = JobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

}

protected void imbtnSearch\_Click(object sender, ImageClickEventArgs e)

{

if (ddlJobCategory.Text == "Job Category")// || ddlExperience.Text == "Experience")

{ }

else

{

Response.Cookies["Category"].Value = ddlJobCategory.SelectedItem.Text;

Response.Cookies["Experience"].Value = ddlExperience.SelectedItem.Text;

Response.Redirect("SearchJob.aspx");

}

protected void LinkButton2\_Click(object sender, EventArgs e)

{

Response.Redirect("~/AdminPanel/AdminLogin.aspx");

}

protected void lnkbtnforgetpwd\_Click(object sender, EventArgs e)

Response.Redirect("BuildCarrier\_Login.aspx");

}

**9.11 OurExpert.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Net.Mail;

using System.Net.Mime;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using BuildCarrier.Components;

public partial class OurExpert : System.Web.UI.Page

{

string jobCategory = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

fnBindCity();

}

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

try

{

if (txtSubject.Text != "" && txtEmail.Text != "")

{

btnSubmit.Attributes.Add("onclick", "return validate()"); string pattern = @"^[a-z][a-z|0-9|]\*([\_][a-z|0-9]+)\*([.][a-z|" +

@"0-9]+([\_][a-z|0-9]+)\*)?@[a-z][a-z|0-9|]\*\.([a-z]" +

@"[a-z|0-9]\*(\.[a-z][a-z|0-9]\*)?)$";

System.Text.RegularExpressions.Match match =

Regex.Match(txtEmail.Text.Trim(), pattern, RegexOptions.IgnoreCase);

if (match.Success)

{

lblMessage.Text = "";

}

else

lblMessage.Text = "Invalid Email ID";

SendUserQuery();

}

else

lblMessage.Text = "";

}

catch (Exception ex)

{ }

}

public void SendUserQuery()

{

string strbody = "<html><body><table><tr><td><b><font color=black> Name : </font></b><font color=blue>" + txtName.Text;

strbody += "</font><tr><td><b><font color=black> Phone No. : </font></b><font color=blue>" + txtContact.Text;

strbody += "</font><tr><td><b><font color=black> Email - ID : </font></b><font color=blue>" + txtEmail.Text;

strbody += "</font><tr><td><b><font color=black> Subject : </font></b><font color=blue>" + txtSubject.Text;

strbody += "</font><tr><td><b><font color=black> Reason For Contact . : </font></b><font color=blue>" + txtQueryDtls.Text;

strbody = strbody + "</p></td></tr></table></html>";

SmtpClient smtpClient = new SmtpClient();

MailMessage message = new MailMessage();

try

{

MailAddress fromAddress = new MailAddress(txtEmail.Text);

smtpClient.Host = "webmail.techinsightvision.com";

//smtpClient.Host = "mail.techinsightvision.com";

smtpClient.Port = 25;

message.From = fromAddress;

message.To.Add("info@techinsightvision.com");

message.Subject = txtSubject.Text;

message.IsBodyHtml = true;

message.Body = strbody;

smtpClient.Send(message);

lblMessage.Visible = true;

lblMessage.Text = "Your Mail Has been Sent Successfully....";

fnClear();

}

catch (Exception ex)

{

lblMessage.Visible = true;

lblMessage.Text = ex.Message.ToString();

}

}

private void fnClear()

{

this.txtSubject.Text = "";

this.txtName.Text = "";

this.txtContact.Text = "";

this.txtEmail.Text = "";

this.txtFax.Text = "";

this.txtAddress.Text = "";

this.txtQueryDtls.Text = "";

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

jobCategory = "IT Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

jobCategory = "HR Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

jobCategory = "B P O";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

jobCategory = "Telecome";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

jobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

jobCategory = "Medical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

jobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

jobCategory = "Garment";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

jobCategory = "Fashion";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

private void fnBindCity()

{

City cityList = new City();

ArrayList indcity = new ArrayList();

ddlCity.Items.Clear();

indcity = cityList.IndCity();

for (int i = 0; i < indcity.Count; i++)

{

ddlCity.Items.Add(indcity[i].ToString());

}

}

}

## **9. 12 PostJob.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using BuildCarrier.Components;

public partial class JobSeeker\_Registration\_step1 : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

protected void Page\_Load(object sender, EventArgs e)

{

/////////////////////////////////////////////////////////////////

//this.txtEmpID.Text = "e\_01";

/////////////////////////////////////////////////////////////////

if (!IsPostBack)

{

if (Request.Cookies["EmployerID"] != null)

{

#region -- Bind Employers Details..

fnEmployerDtls();

#endregion

#region -- Bind Jobs Posted by Logged in Employers

BindPostedJobs();

#endregion

if (this.chkTerms.Checked == true)

this.btnSubmit.Enabled = true;

else

this.btnSubmit.Enabled = false;

this.txtPostingDate.Text = DateTime.Now.Month.ToString() + "/" +

DateTime.Now.Day.ToString() + "/" +

DateTime.Now.Year.ToString();

DataTable dtBindJobCategory = new DataTable();

string strBindJobCategory = string.Empty;

try

{

#region -- Job Category

strBindJobCategory = "SELECT Category\_ID, JobCategory FROM Job\_Category Order By Category\_ID";

dtBindJobCategory = dut.GetDataTable(strBindJobCategory);

if (dtBindJobCategory.Rows.Count > 0)

{

this.ddlJobCategory.DataSource = dtBindJobCategory;

this.ddlJobCategory.DataTextField = "JobCategory";

this.ddlJobCategory.DataValueField = "Category\_ID";

this.ddlJobCategory.DataBind();

}

else

{

}

#endregion

#region -- Job ID

//BindJobID();

#endregion

}

catch (Exception ex)

{

}

else

this.lnkLogout.Text = "Signup for New Employers..";

}

}

#region -- fnEmployerDtls()

private void fnEmployerDtls()

{

try

{

//Response.Cookies["A1EmployerID"].Value

DataTable dtGetEmpDtls = new DataTable();

string strGetEmpDtls = "SELECT employer\_id, e\_name, com\_name, com\_address, contact\_no from Employers where id = " + "'" + Request.Cookies["EmployerID"].Value + "'";

dtGetEmpDtls = dut.GetDataTable(strGetEmpDtls);

if (dtGetEmpDtls.Rows.Count != 0)

{

this.txtEmpID.Text = dtGetEmpDtls.Rows[0]["employer\_id"].ToString();

this.txtEmpName.Text = dtGetEmpDtls.Rows[0]["e\_name"].ToString();

this.txtCompany.Text = dtGetEmpDtls.Rows[0]["com\_name"].ToString();

this.txtCompanyAddress.Text = dtGetEmpDtls.Rows[0]["com\_address"].ToString();

this.txtContactNum.Text = dtGetEmpDtls.Rows[0]["contact\_no"].ToString();

}

}

catch (Exception ex)

{ }

}

#endregion

#region -- BindPostedJobs()

private void BindPostedJobs()

{

try

{

DataTable dtPostedJobs = new DataTable();

string strPostJobs = string.Empty;

strPostJobs = "SELECT Job\_ID, Job\_Title, Posting\_Date FROM Job\_Details WHERE emp\_id = " + "'" + Request.Cookies["EmployerID"].Value + "'" + " ORDER BY Posting\_Date DESC";

dtPostedJobs = dut.GetDataTable(strPostJobs);

if (dtPostedJobs.Rows.Count > 0)

{

dgPostedJob.DataSource = dtPostedJobs;

dgPostedJob.DataBind();

}

else

{

}

}

catch (Exception ex)

{

}

}

#endregion

#region -- BindJobID()

private void BindJobID()

{

try

{

//if (this.txtJobID.Text == "")

//{

this.lnkPostJob.Visible = true;

DataTable dtJobID = new DataTable();

string strJobID = string.Empty;

strJobID = "GetNewJobID";

dtJobID = dut.GetDataTable(strJobID);

if (dtJobID.Rows.Count > 0)

{

this.txtJobID.Text = dtJobID.Rows[0]["NowJobID"].ToString();

}

else

{

}

//}

//else

//{

// this.lnkPostJob.Visible = true;

//}

}

catch (Exception ex)

{

}

}

#endregion

protected void btnSubmit\_Click(object sender, EventArgs e)

{

#region -- Code to Post New Job

#region -- Validations

if (txtJobID.Text == "")

this.lblMsgJobID.Text = "Generate JOB ID First";

else

this.lblMsgJobID.Text = "\*";

if (txtJobTitle.Text == "")

this.lblMsgTitle.Text = "Fill Job Title";

else

this.lblMsgTitle.Text = "\*";

if (txtJobDtls.Text == "")

this.lblMsgDtls.Text = "Fill Job Description";

else

this.lblMsgDtls.Text = "\*";

#endregion

if (txtJobTitle.Text != "" && txtJobDtls.Text != "")

{

try

{

DataUtility dutPostNewJob = new DataUtility();

string strPostNewJob = string.Empty;

strPostNewJob = "UPDATE job\_details SET job\_title = '" + this.txtJobTitle.Text + "'" + "," + "job\_desc = " + "'" + this.txtJobDtls.Text + "'" + "," +

"emp\_id = " + "'" + this.txtEmpID.Text + "'" + "," + "job\_category = " + "'" + this.ddlJobCategory.SelectedItem + "'" + "," + " exp\_req = " + "'" + this.ddlExperience.SelectedItem +

"'" + "," + " no\_of\_days = " + "'" + this.rdbDays.SelectedItem + "'" + "," + " posting\_date = " + "'" + this.txtPostingDate.Text + "'" +

" WHERE Job\_Id = " + "'" + this.txtJobID.Text + "'";

int ExecuteSQL = dut.executeSql(strPostNewJob);

if (ExecuteSQL == 0)

{

this.pnlJobSummary.Visible = false;

this.lblMsg.Text = "Error Occured while Saving Records..";

}

else

{

this.pnlJobSummary.Visible = true;

this.txtPostedJobID.Text = this.txtJobID.Text;

this.txtPostedJobTitle.Text = this.txtJobTitle.Text;

this.txtPostedJobDesc.Text = this.txtJobDtls.Text;

this.txtPostedJobCategory.Text = this.ddlJobCategory.SelectedItem.ToString();

this.lblExpiryDay.Text = this.rdbDays.SelectedItem.ToString();

BindPostedJobs();

ResetControls();

this.lnkPostJob.Visible = true;

}

}

catch (Exception ex)

{

}

}

#endregion

}

private void ResetControls()

{

this.txtJobTitle.Text = "";

this.txtJobID.Text = "";

this.txtJobDtls.Text = "";

this.ddlJobCategory.SelectedIndex = 0;

this.ddlExperience.SelectedIndex = 0;

this.rdbDays.SelectedIndex = 0;

this.chkTerms.Checked = false;

this.btnSubmit.Enabled = false;

this.lnkPostJob.Visible = true;

}

protected void chkTerms\_CheckedChanged(object sender, EventArgs e)

{

if (Request.Cookies["EmployerID"] != null)

if (this.chkTerms.Checked == true)

this.btnSubmit.Enabled = true;

else

this.btnSubmit.Enabled = false;

else

this.btnSubmit.Enabled = false;

}

protected void btnReset\_Click(object sender, EventArgs e)

{

ResetControls();

}

protected void lnkPostJob\_Click(object sender, EventArgs e)

{

this.pnlJobSummary.Visible = false;

BindJobID();

}

protected void btnEditJob\_Click(object sender, EventArgs e)

{

lnkPostJob.Visible = false;

string strJobID = this.txtPostedJobID.Text;

GetJobIDWise(strJobID);

}

protected void dgPostedJob\_EditCommand(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)

{

try

{

ViewState["i"] = this.dgPostedJob.DataKeys[e.Item.ItemIndex];

string strtest = ViewState["i"].ToString();

GetJobIDWise(strtest);

}

catch (Exception ex)

{

}

}

private void GetJobIDWise(string strSQL)

{

DataTable dtGetJobDtls = new DataTable();

string strGetJobDtls = string.Empty;

strGetJobDtls = "SELECT job\_id, Job\_Title, job\_desc, job\_category, exp\_req, no\_of\_days FROM Job\_Details WHERE Job\_ID = " +

"'" + strSQL + "'";

dtGetJobDtls = dut.GetDataTable(strGetJobDtls);

if (dtGetJobDtls.Rows.Count > 0)

{

this.txtJobID.Text = dtGetJobDtls.Rows[0]["Job\_Id"].ToString();

this.txtJobTitle.Text = dtGetJobDtls.Rows[0]["Job\_Title"].ToString();

this.txtJobDtls.Text = dtGetJobDtls.Rows[0]["job\_desc"].ToString();

this.ddlExperience.ClearSelection();

this.ddlExperience.Items.FindByText(dtGetJobDtls.Rows[0]["exp\_req"].ToString()).Selected = true;

this.ddlJobCategory.ClearSelection();

this.ddlJobCategory.Items.FindByText(dtGetJobDtls.Rows[0]["job\_category"].ToString()).Selected = true;

this.rdbDays.ClearSelection();

this.rdbDays.Items.FindByText(dtGetJobDtls.Rows[0]["no\_of\_days"].ToString()).Selected = true;

}

lnkPostJob.Visible = false;

}

protected void dgPostedJob\_PageIndexChanged(object source, DataGridPageChangedEventArgs e)

{

dgPostedJob.CurrentPageIndex = e.NewPageIndex;

dgPostedJob.DataBind();

}

protected void lnkLogout\_Click(object sender, EventArgs e)

{

//Response.Cookies["A1EmployerID"].Expires = DateTime.Now.AddYears(-30);

if (lnkLogout.Text == "Logout")

{

if (Request.Cookies["EmployerID"] != null)

Response.Cookies["EmployerID"].Expires = DateTime.Now.AddYears(-30);

Response.Redirect("Home.aspx");

}

else

Response.Redirect("EmployerRegistration.aspx");

}

protected void btnEdit\_Click(object sender, EventArgs e)

{

if (Request.Cookies["EmployerID"].Value != null)

Response.Redirect("EmployerRegistration.aspx");

else

Response.Redirect("Home.aspx");

}

}

## **9. 13 RegistrationStep1.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.IO;

using System.Data.SqlClient;

using System.Net.Mail;

using System.Net.Mime;

using System.Text;

using System.Text.RegularExpressions;

using BuildCarrier.Components;

public partial class JobSeeker\_Registration\_step1 : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

string str1;

public static string DOB;

string dd = string.Empty;

string mm = string.Empty;

string yy = string.Empty;

SqlCommand cmd = new SqlCommand();

SqlDataReader dr;

string jobCategory = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

fnDOB();

fnBindCity();

fnBindJobCategory();

ddlDay.Visible = true;

ddlMonth.Visible = true;

ddlYear.Visible = true;

lblDOB.Visible = false;

if (Request.Cookies["SeekerID"] == null)

{

}

else

{

fnBindProfile();

}

}

}

#region -- fnBindJobCategory()

private void fnBindJobCategory()

{

DataTable dtBindJobCategory = new DataTable();

string strBindJobCategory = string.Empty;

strBindJobCategory = "SELECT Category\_ID, JobCategory FROM Job\_Category Order By Category\_ID";

try

{

dtBindJobCategory = Dut.GetDataTable(strBindJobCategory);

if (dtBindJobCategory.Rows.Count > 0)

{

this.ddlJobCategory.DataSource = dtBindJobCategory;

this.ddlJobCategory.DataTextField = "JobCategory";

this.ddlJobCategory.DataValueField = "Category\_ID";

this.ddlJobCategory.DataBind();

}

else

{

}

}

catch (Exception ex)

{ }

}

#endregion

protected void btnSubmit\_Click(object sender, EventArgs e)

{

fnSubmitProfile();

}

private void funSubmitProfile()

{

dd = ddlDay.SelectedItem.Text;

mm = ddlMonth.SelectedItem.Text;

yy = ddlYear.SelectedItem.Text;

DOB = dd + "/" + mm + "/" + yy;

string insertSeekerProfile = string.Empty;

Int64 contact = Convert.ToInt64(txtSeekerContact.Text);

insertSeekerProfile = "usp\_InsertSeekerDtls " +

"'" + txtSeekerEmail.Text + "'" + ","

+ "'" + txtSeekerFName.Text + "'" + ","

+ "'" + txtSeekerLName.Text + "'" + ","

+ "'" + txtSeekerAddress.Text + "'" + ","

+ "'" + ddlCity.SelectedItem.Text.ToString() + "'" + ","

+ "'" + txtSeekerState.Text + "'" + ","

+ contact + ","

+ "'" + txtSeekerEmail.Text + "'" + ","

+ "'" + DOB.ToString() + "'" + ","

+ "'" + txtSeekerAcdQualification.Text + "'" + ","

+ "'" + txtSeekerProfQualification.Text + "'" + ","

+ "'" + txtSeekerOtherQualification.Text + "'" + ","

+ "'" + ddlExperience.SelectedItem.Text.ToString() + "'" + ","

+ "'" + ddlJobCategory.SelectedItem.Text.ToString() + "'" + ","

+ "'" + txtSeekerPwd.Text + "'";

try

{

DataTable dtRes = new DataTable();

dtRes = dut.GetDataTable(insertSeekerProfile);

if (dtRes.Rows.Count > 0)

{

fnClear();

fnResumePath();

fnSendUserID();

}

else

{

this.txtSeekerEmail.Text = "";

this.txtSeekerEmail.Focus();

}

}

#region ----fnSendUser()------

private void fnSendUserID()

{

//string strbody = "<html><body><table><tr><td><b><font color=black> Name : </font></b><font color=blue>" + "anjum0326@gmail.com";

string strbody = "<html><body><table><tr><td><b><font color=black> Your Login ID : </font></b><font color=blue>" + txtUserID.Text;

//strbody += "</font><tr><td><b><font color=black> Login - ID : </font></b><font color=blue>" + id;

strbody += "</font><tr><td><b><font color=black> Login Password : </font></b><font color=blue>" + txtSeekerPwd.Text;

strbody = strbody + "</p></td></tr></table></html>";

SmtpClient smtpClient = new SmtpClient();

MailMessage message = new MailMessage();

try

{

MailAddress fromAddress = new MailAddress("sales@techinsightvision.com");

smtpClient.Host = "webmail.techinsightvision.com";

smtpClient.Port = 25;

message.From = fromAddress;

message.To.Add(this.txtSeekerEmail.Text);

//message.To.Add("ibrar\_star@yahoo.com");

message.Subject = "A1 Jobs account details.";

//message.Subject = id;

message.IsBodyHtml = true;

message.Body = strbody;

smtpClient.Send(message);

lblError.Visible = true;

lblError.Text = "Your Login Details Has been Sent in your mail id";

//reset();

}

catch (Exception ex)

{

lblError.Visible = true;

lblError.Text = ex.Message.ToString();

}

}

#endregion

private void fnResumePath()

{

try

{

SqlConnection con = new SqlConnection();

if (FileUpload1.PostedFile != null)

{

FileUpload1.SaveAs(MapPath("~/Resume/" + this.txtSeekerEmail.Text + ".doc"));

}

else

{

}

}

catch (Exception ex)

{ }

}

private void fnBindCity()

{

City cityList = new City();

ArrayList indcity = new ArrayList();

ddlCity.Items.Clear();

indcity = cityList.IndCity();

for (int i = 0; i < indcity.Count; i++)

{

ddlCity.Items.Add(indcity[i].ToString());

}

}

private void fnDOB()

{

for (int i = 1; i <= 31; i++)

{

ddlDay.Items.Add(i.ToString());

dd = ddlDay.Text;

}

for (int i = 1; i <= 12; i++)

{

ddlMonth.Items.Add(i.ToString());

mm = ddlMonth.Text;

}

for (int i = 1970; i <= 2000; i++)

{

ddlYear.Items.Add(i.ToString());

yy = ddlYear.Text;

}

DOB = dd + "/" + mm + "/" + yy;

}

private void fnBindProfile()

{

try

{

ddlDay.Visible = false;

ddlMonth.Visible = false;

ddlYear.Visible = false;

lblDOB.Visible = true;

this.lnkBtnLogin.Text = "Logout";

string strbindProfile = string.Empty;

DataTable dtBindProfile = new DataTable();

strbindProfile = "select \* from jobseeker\_profile where Seeker\_Id='" + Request.Cookies["SeekerID"].Value + "'";

dtBindProfile = dut.GetDataTable(strbindProfile);

if (dtBindProfile.Rows.Count != 0)

{

this.txtUserID.Text = dtBindProfile.Rows[0]["Seeker\_Id"].ToString();

this.txtSeekerPwd.Text = dtBindProfile.Rows[0]["Password"].ToString();

this.txtSeekerFName.Text = dtBindProfile.Rows[0]["First\_Name"].ToString();

this.txtSeekerLName.Text = dtBindProfile.Rows[0]["Last\_Name"].ToString();

this.txtSeekerAddress.Text = dtBindProfile.Rows[0]["Address"].ToString();

this.ddlCity.Text = dtBindProfile.Rows[0]["City"].ToString();

this.txtSeekerState.Text = dtBindProfile.Rows[0]["State"].ToString();

this.txtSeekerContact.Text = dtBindProfile.Rows[0]["Contact\_No"].ToString();

this.lblDOB.Text = dtBindProfile.Rows[0]["DoB"].ToString();

this.txtSeekerAcdQualification.Text = dtBindProfile.Rows[0]["Edu\_Qualification"].ToString();

this.txtSeekerProfQualification.Text = dtBindProfile.Rows[0]["Pro\_Qualification"].ToString();

this.txtSeekerOtherQualification.Text = dtBindProfile.Rows[0]["OtherQualification"].ToString();

this.ddlExperience.Text = dtBindProfile.Rows[0]["Experience"].ToString();

this.ddlJobCategory.Text = dtBindProfile.Rows[0]["Job\_Category"].ToString();

}

}

catch (Exception ex)

{

}

}

protected void btnUpdate\_Click(object sender, EventArgs e)

{

fnUpdateProfile();

}

private void fnUpdateProfile()

{

try

{

string ID = txtUserID.Text;

SqlConnection con = new SqlConnection(strCon);

cmd.Connection = con;

con.Open();

cmd.CommandText = "update jobseeker\_profile set Address='" + txtSeekerAddress.Text + "',city='" + ddlCity.Text + "',state='" + txtSeekerState.Text + "',contact\_no='" + txtSeekerContact.Text + "',Email='" + txtSeekerEmail.Text + "',edu\_qualification='" + txtSeekerAcdQualification.Text + "',pro\_qualification='" + txtSeekerProfQualification.Text + "',OtherQualification='" + txtSeekerOtherQualification.Text + "',experience='" + ddlExperience.Text + "',job\_category='" + txtSeekerJobCategory.Text + "' where Seeker\_Id='" + ID + "'";

cmd.ExecuteNonQuery();

con.Close();

}

catch (Exception ex)

{ }

}

protected void btnClear\_Click(object sender, EventArgs e)

{

fnClear();

}

private void fnClear()

{

txtUserID.Text = "";

txtSeekerPwd.Text = "";

txtSeekerLName.Text = "";

txtSeekerFName.Text = "";

txtSeekerContact.Text = "";

txtSeekerDOB.Text = "";

txtSeekerEmail.Text = "";

txtSeekerAddress.Text = "";

ddlCity.SelectedItem.Text = "Select City";

ddlExperience.SelectedItem.Text = "Fresher";

txtSeekerJobCategory.Text = "";

txtSeekerOtherQualification.Text = "";

txtSeekerProfQualification.Text = "";

txtSeekerAcdQualification.Text = "";

txtResumeTitle.Text = "";

txtSeekerState.Text = "";

ddlDay.SelectedItem.Text = "Day";

ddlMonth.SelectedItem.Text = "Month";

ddlYear.SelectedItem.Text = "Year";

}

private void fnBindFile()

{

FileInfo strFile;

strFile = new FileInfo(Server.MapPath("~/Resume/" + this.txtUserID.Text + ".doc"));

str1 = strFile.ToString();

if (strFile.Exists)

{

//lnkbtnFile.Text = "Your Resume<A HREF=" + str1 + "></A>";

Response.Clear();

Response.AppendHeader("Content-Disposition", "attachment; filename=" + strFile.Name);

Response.AppendHeader("Content-Length", strFile.Length.ToString());

Response.ContentType = "application/octet-stream";

Response.WriteFile(strFile.FullName);

Response.End();

}

}

protected void lnkbtnFile\_Click(object sender, EventArgs e)

{

fnBindFile(); ;

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

jobCategory = "IT Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

jobCategory = "HR Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

{

jobCategory = "B P O";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

jobCategory = "Telecome";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{

jobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

jobCategory = "Medical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

jobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

jobCategory = "Garment";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

jobCategory = "Fashion";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void LinkButton1\_Click(object sender, EventArgs e)

{

Response.Redirect("A1Jobs\_Login.aspx");

}

}

## **9. 14 SearchJob.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Data.SqlClient;

using System.Web.Services;

using System.Text;

using System.Net.Mail;

using System.Net.Mime;

using System.IO;

using BuildCarrier.Components;

public partial class SearchJob : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

string jobCategory = string.Empty;

string Exp = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

fnBindJobCategory();

fnSearchjob()

}

}

#region -- fnBindJobCategory()

private void fnBindJobCategory()

{

DataTable dtBindJobCategory = new DataTable();

string strBindJobCategory = string.Empty;

strBindJobCategory = "SELECT Category\_ID, JobCategory FROM Job\_Category Order By CategoryID";

try

{

dtBindJobCategory = Dut.GetDataTable(strBindJobCategory);

if (dtBindJobCategory.Rows.Count > 0)

{

this.ddlJobCategory.DataSource = dtBindJobCategory;

this.ddlJobCategory.DataTextField = "JobCategory";

this.ddlJobCategory.DataValueField = "Category\_ID";

this.ddlJobCategory.DataBind();

}

else

{

}

}

catch (Exception ex)

{ }

}

#endregion

//string JobCategory = string.Empty;

private void fnSearchjob()

{

try

{

fnBindJob();

}

catch (Exception ex)

{}

}

private void fnBindJob()

{

try

{

DataUtility dut = new DataUtility();

DataTable dtJob = new DataTable();

string strJob;

string exp = Request.Cookies["Experience"].Value.ToString();

if (exp == "All")

{

strJob = "select \* from job\_details where job\_category='" + Request.Cookies["Category"].Value + "' and Job\_Status=1 order by exp\_req desc ";

}

else

{

strJob = "select \* from job\_details where job\_category='" + Request.Cookies["Category"].Value + "' and exp\_req='" + Request.Cookies["Experience"].Value + "' and Job\_Status=1 order by exp\_req desc ";

}

dtJob = dut.GetDataTable(strJob);

if (dtJob.Rows.Count > 0)

{

pnlControl.Visible = true;

GVJob.DataSource = dtJob;

GVJob.DataBind();

}

else

{

pnlControl.Visible = false;

}

}

catch (Exception ex)

{

pnlControl.Visible = false;

}

}

protected void btnApllyAll\_Click(object sender, EventArgs e)

{

}

protected void GVJob\_PageIndexChanged(object sender, EventArgs e)

{

}

protected void GVJob\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GVJob.PageIndex = e.NewPageIndex;

fnBindJob();

string strID;

protected void btnSelectedApply\_Click(object sender, EventArgs e)

{

DataUtility DutGridValue = new DataUtility();

DataTable DtGridValue = new DataTable();

string CrRow;

ArrayList frmList = new ArrayList();

foreach (GridViewRow row in GVJob.Rows)

{

CheckBox cb = (CheckBox)row.FindControl("chkJobAply");

if (cb.Checked)

{

CrRow = row.Cells[1].Text

frmList.Add(CrRow);

}

else

{

//lblErrormsg.Text = "There is No Job Selected";

}

}

int i;

for (i = 0; i < frmList.Count; i++)

{

strID += frmList[i].ToString() + ",";

//string query1 = "select e\_id from job\_dtls where job\_title='" + frmList[i].ToString() + "'";

}

fnJobApplyed();

}

private void fnJobApplyed()

{

try

{

DataUtility Dut = new DataUtility();

DataTable Dt = new DataTable();

if (Request.Cookies["SeekerID"] != null)

{

string strinsert = "sp\_jobTransaction '" + strID + "'" + ",'" + Request.Cookies["SeekerID"].Value + "'";

int responce = Dut.executeSql(strinsert);

if (responce != 0)

{

lblErrormsg.Text = "Job Applied";

}

else

lblErrormsg.Text = "Operation Failed";

}

else

Response.Redirect("BuildCarrier\_Login.aspx");

}

catch (Exception ex)

{ }

}

protected void btnAllApply\_Click(object sender, EventArgs e)

{

foreach (GridViewRow gvr in GVJob.Rows)

{

CheckBox cb = (CheckBox)gvr.FindControl("chkJobAply");

cb.Checked = true; }

DataUtility DutGridValue = new DataUtility();

DataTable DtGridValue = new DataTable();

string CrRow;

ArrayList frmList = new ArrayList();

foreach (GridViewRow row in GVJob.Rows)

{

CheckBox cb = (CheckBox)row.FindControl("chkJobAply");

if (cb.Checked)

{

CrRow = row.Cells[1].Text;

frmList.Add(CrRow);

}

else

{

//lblErrormsg.Text = "There is No Job Selected";

}

}

int i;

for (i = 0; i < frmList.Count; i++)

{

strID += frmList[i].ToString() + ",";

//string query1 = "select e\_id from job\_dtls where job\_title='" + frmList[i].ToString() + "'";

}

fnJobApplyed();

}

protected void lnkbntIT\_Click(object sender, EventArgs e)

{

//jobCategory = "";

jobCategory = "IT Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnHR\_Click(object sender, EventArgs e)

{

jobCategory = "HR Sector";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnBPO\_Click(object sender, EventArgs e)

jobCategory = "B P O";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnTelcom\_Click(object sender, EventArgs e)

{

jobCategory = "Telecome";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnSales\_Click(object sender, EventArgs e)

{jobCategory = "Sales/Marketing";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnMedical\_Click(object sender, EventArgs e)

{

jobCategory = "Medical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnEng\_Click(object sender, EventArgs e)

{

jobCategory = "Engineering/Technical";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnGarment\_Click(object sender, EventArgs e)

{

jobCategory = "Garment";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void lnkbtnFashion\_Click(object sender, EventArgs e)

{

jobCategory = "Fashion";

Response.Cookies["Job"].Value = jobCategory.ToString();

Response.Redirect("Submit.aspx");

}

protected void imbtnSearch\_Click(object sender, ImageClickEventArgs e)

{

try

{

DataUtility dutSrc = new DataUtility();

DataTable dtSrc = new DataTable();

string strJob;

string exp = ddlExperience.SelectedItem.Text;

if (exp == "All")

{

strJob = "select \* from job\_details where job\_category='" + ddlJobCategory.SelectedItem.Text + "' and Job\_Status=1 order by exp\_req desc ";

}

else

{

strJob = "select \* from job\_details where job\_category='" + ddlJobCategory.SelectedItem.Text + "' and exp\_req='" + ddlExperience.SelectedItem.Text + "' and Job\_Status=1 order by exp\_req desc ";

}

dtSrc = dutSrc.GetDataTable(strJob);

if (dtSrc.Rows.Count > 0)

{

pnlControl.Visible = true;

GVJob.DataSource = dtSrc;

GVJob.DataBind();

}

else

{

pnlControl.Visible = false;

}

}

catch (Exception ex)

{

pnlControl.Visible = false;

}

}

}

## **9. 15 SearchResume.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using BuildCarrier.Components;

public partial class SarchResume : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

string empID = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

try

{

if (!IsPostBack)

{

if (Request.Cookies["EmployerID"] != null)

{

empID = Request.Cookies["EmployerID"].Value.ToString();

#region -- Bind Job Category & Pending Job Status

fnBindJobCategory();

fnBindJobs();

#endregion

fnBindS\_Pending();

}

else

{

this.btnSearch.Enabled = false;

ddlCategory.Items.Add("No Session Found");

this.lnkPendingApplication.Text = "No Session Found";

this.lnkPendingApplication.Enabled = false;

this.ddlCategory.Enabled = false;

fnBindS\_Pending();

}

}

}

catch (Exception ex)

{ }

}

#region -- fnBindJobCategory()

private void fnBindJobCategory()

{

DataTable dtBindJobCategory = new DataTable();

string strBindJobCategory = string.Empty;

strBindJobCategory = "SELECT Category\_ID, JobCategory FROM Job\_Category Order By Category\_ID";

try

{

dtBindJobCategory = dut.GetDataTable(strBindJobCategory);

if (dtBindJobCategory.Rows.Count > 0)

{

this.ddlCategory.DataSource = dtBindJobCategory;

this.ddlCategory.DataTextField = "JobCategory";

this.ddlCategory.DataValueField = "Category\_ID";

this.ddlCategory.DataBind();

}

else

{

}

}

catch (Exception ex)

{ }

}

#endregion

#region -- fnBindJobs()

private void fnBindJobs()

{

if (Request.Cookies["EmployerID"] != null)

{

DataTable dtEmployersJob = new DataTable();

string strEmployersJob = string.Empty;

strEmployersJob = "SELECT RowID, EmpID, SeekerID, JobID, JobTitle, Date, Status = CASE WHEN Status = 0 THEN 'Pending' ELSE 'OK' END"

+ " FROM EmployerPendingRequest WHERE EmpID = " + "'" + Request.Cookies["EmployerID"].Value + "'" + " ORDER BY Date ASC";

try

{

dtEmployersJob = dut.GetDataTable(strEmployersJob);

if (dtEmployersJob.Rows.Count != 0)

{

dgPendingRequest.DataSource = dtEmployersJob;

dgPendingRequest.DataBind();

}

else

{

}

}

catch (Exception ex)

{

}

}

}

#endregion

protected void btnSearch\_Click(object sender, EventArgs e)

{

string strSearch = string.Empty;

DataTable dtGetResume = new DataTable();

try

{

strSearch = "SELECT Seeker\_id, Resume\_Title, Experience, (First\_Name + ' ' + Last\_Name) as Name, Email, Contact\_no FROM JobSeeker\_Profile WHERE " +

"job\_category LIKE " + "'%" + this.ddlCategory.SelectedItem.Text + "%'" + " AND Experience LIKE " + "'%" + this.ddlExperience.SelectedItem.Text + "%'";

dtGetResume = dut.GetDataTable(strSearch);

if (dtGetResume.Rows.Count != 0)

{

dgSearchResume.DataSource = dtGetResume;

dgSearchResume.DataBind();

}

else

{

dtGetResume.Clear();

dgSearchResume.DataSource = dtGetResume;

dgSearchResume.DataBind();

}

}

catch (Exception ex)

{

}

}

protected void dgSearchResume\_EditCommand(object source, DataGridCommandEventArgs e)

{

try

{

ViewState["i"] = this.dgSearchResume.DataKeys[e.Item.ItemIndex];

if (Request.Cookies["JobSeekerID"] != null)

Response.Cookies["JobSeekerID"].Expires = DateTime.Now.AddYears(-30);

Response.Cookies["JobSeekerID"].Value = ViewState["i"].ToString();

if (Request.Cookies["LoggedInJobSeekerID"] == null)

OpenNewWindow("SeekerProfile.aspx");

//Response.Write("<script type='text/javascript'>detailedresults=window.open('SeekerProfile.aspx');</script>");

//System.Diagnostics.Process.Start("iexplore.exe", "www.techinsightvision.com");

else

{

CloseWindow("SeekerProfile.aspx");

if (Request.Cookies["LoggedInJobSeekerID"] != null)

//Response.Cookies["LoggedInJobSeekerID"].Expires = DateTime.Now.AddYears(-30);

//System.Diagnostics.Process.Start("iexplore.exe", "../A1jobs/SeekerProfile.aspx");

OpenNewWindow("SeekerProfile.aspx");

}

}

catch (Exception ex)

{

}

}

private void OpenNewWindow(string url)

{

ClientScript.RegisterStartupScript(this.GetType(), "newWindow", String.Format("<script>window.open('{0}');</script>", url) );

}

private void CloseWindow(string URL)

{

ClientScript.RegisterStartupScript(this.GetType(), "closeWindow", String.Format("<script>window.close('{0}');</script>", URL));

}

protected void imgRfresh\_Click(object sender, ImageClickEventArgs e)

{

fnBindJobs();

}

protected void lnkbtnLogout\_Click(object sender, EventArgs e)

{

if (Request.Cookies["EmployerID"] != null)

Response.Cookies["EmployerID"].Expires = DateTime.Now.AddYears(-30);

if (Request.Cookies["JobSeekerID"] != null)

Response.Cookies["JobSeekerID"].Expires = DateTime.Now.AddYears(-30);

Response.Redirect("Home.aspx");

}

#region --fnBindS\_Pending--

private void fnBindS\_Pending()

{

DataUtility dutPending = new DataUtility();

DataTable dtPending = new DataTable();

string strPending = "select job\_id,seeker\_id from job\_transaction where emp\_status='0' and emp\_id='" + empID + "' ";

dtPending = dutPending.GetDataTable(strPending);

if (dtPending.Rows.Count > 0)

{

GVpending.DataSource = dtPending;

GVpending.DataBind();

}

}

#endregion

protected void GVpending\_RowCommand(object sender, GridViewCommandEventArgs e)

{

try

{

int index = Convert.ToInt32(e.CommandArgument);

if (e.CommandName == "id")

{

GridViewRow row = GVpending.Rows[index];

string id = Convert.ToString(row.Cells[1].Text);

//if (Request.Cookies["JobSeekerID"] != null)

// Response.Cookies["JobSeekerID"].Expires = DateTime.Now.AddYears(-30);

Response.Cookies["JobSeekerID"].Value = id.ToString();

OpenNewWindow("SeekerProfile.aspx");

//if (Request.Cookies["LoggedInJobSeekerID"] == null)

// OpenNewWindow("../A1Job/SeekerProfile.aspx");

//else

//{

// CloseWindow("../A1Job/SeekerProfile.aspx");

// if (Request.Cookies["LoggedInJobSeekerID"] != null)

// OpenNewWindow("../A1Job/SeekerProfile.aspx");

//}

}

}

catch (Exception ex)

{ }

}

}

## **9. 16 SeekerProfile.aspx.cs**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Data.SqlClient;

using Microsoft.Office.Core;

using BuildCarrier.Components;

public partial class SeekerProfile : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

protected void Page\_Load(object sender, EventArgs e)

{

if (Request.Cookies["JobSeekerID"] == null)

{

fnResetPage();

this.lblMsg.Text = "No Session Found.. Please try again !!";

}

else

{

Response.Cookies["LoggedInJobSeekerID"].Value = Request.Cookies["JobSeekerID"].Value;

OpenResume();

fnBindJobTitle();

DataTable dtFetchSeekerProfile = new DataTable();

string strFetchSeekerProfile = string.Empty;

strFetchSeekerProfile = "SELECT First\_Name + ' ' + Last\_Name AS Name, Address + ' ' + city + ' ' + state AS Address, contact\_no, Email, "

+ "DoB, edu\_qualification, pro\_qualification, OtherQualification, Resume\_Title, experience, job\_category FROM jobseeker\_profile "

+ "WHERE Seeker\_ID = " + "'" + Request.Cookies["JobSeekerID"].Value + "'";

try

{

dtFetchSeekerProfile = dut.GetDataTable(strFetchSeekerProfile);

if (dtFetchSeekerProfile.Rows.Count != 0)

{

this.lblSeekerName.Text =

dtFetchSeekerProfile.Rows[0]["Name"].ToString();

this.lblSeekerAddress.Text = dtFetchSeekerProfile.Rows[0]["Address"].ToString();

u.lblDOB.Text =

dtFetchSeekerProfile.Rows[0]["DoB"].ToString();

this.lblSeekerContactNumber.Text = dtFetchSeekerProfile.Rows[0]["contact\_no"].ToString();

this.lblSeekerEmail.Text = dtFetchSeekerProfile.Rows[0]["Email"].ToString();

this.lblEduQualifications.Text = dtFetchSeekerProfile.Rows[0]["edu\_qualification"].ToString();

this.lblSeekerProQualification.Text = dtFetchSeekerProfile.Rows[0]["pro\_qualification"].ToString();

this.lblExperience.Text = dtFetchSeekerProfile.Rows[0]["experience"].ToString();

this.lblResumeTitle.Text = dtFetchSeekerProfile.Rows[0]["Resume\_Title"].ToString();

this.lblJobCategory.Text = dtFetchSeekerProfile.Rows[0]["job\_category"].ToString();

//this.txtResume.Text = "";

}

else

{

}

}

catch (Exception ex)

{

}

}

}

private void fnResetPage()

{

this.lblSeekerName.Text = "";

this.lblSeekerAddress.Text = "";

this.lblDOB.Text = "";

this.lblSeekerContactNumber.Text = "";

this.lblSeekerEmail.Text = "";

this.lblEduQualifications.Text = "";

this.lblSeekerProQualification.Text = "";

this.lblExperience.Text = "";

this.lblResumeTitle.Text = "";

this.lblJobCategory.Text = "";

this.txtResume.Text = "";

}

protected void btnSendMail\_Click(object sender, EventArgs e)

{

if (Request.Cookies["JobSeekerID"] != null)

Response.Cookies["JobSeekerID"].Expires = DateTime.Now.AddYears(-30);

if (Request.Cookies["LoggedInJobSeekerID"] != null)

Response.Cookies["LoggedInJobSeekerID"].Expires = DateTime.Now.AddYears(-30);

if (Request.Cookies["JobSeekerID"] != null && Request.Cookies["LoggedInJobSeekerID"] != null)

{

string strCurrentDate = DateTime.Now.Month.ToString() + "/" +

DateTime.Now.Day.ToString() + "/" +

DateTime.Now.Year.ToString();

string strInsertRequest = string.Empty;

strInsertRequest = "InsertEmployerRequest " + "'" +

Request.Cookies["EmployerID"].Value + "'" + "," + "'" + Request.Cookies["JobSeekerID"].Value + "'"

+ "," + "'" + this.ddlJobList.SelectedItem.Value.ToString() + "'" + "," + "'" + this.ddlJobList.SelectedItem.Text.ToString() + "'"

+ "," + "'" + strCurrentDate + "'";

try

{

int InsertResponse = dut.executeSql(strInsertRequest);

if (InsertResponse != 0)

{

DataUtility dutinboxinsert = new DataUtility();

DataTable dtinboxinsert = new DataTable();

// string strInboxinsert = "

this.lblMsg.Text = "Request Sent successfully to A1 Jobs administrator...";

}

else

{

this.lblMsg.Text = "Request Not Sent.. Please Try again !!";

}

}

catch (Exception ex)

{

}

}

}

protected void btnClose\_Click(object sender, EventArgs e)

{

if (Request.Cookies["JobSeekerID"] != null)

Response.Cookies["JobSeekerID"].Expires = DateTime.Now.AddYears(-30);

if (Request.Cookies["LoggedInJobSeekerID"] != null)

Response.Cookies["LoggedInJobSeekerID"].Expires = DateTime.Now.AddYears(-30);

}

private void OpenResume()

{

Word.ApplicationClass WordApp = new Word.ApplicationClass();

//string filePath = Server.MapPath("..\\Jobseeker\\" + Ide + ".doc");

try

{

string filePath = Server.MapPath("..\\A1job\\Resume\\" + Request.Cookies["JobSeekerID"].Value + ".doc");

object file = filePath;

object nullobj = System.Reflection.Missing.Value;

Word.Document doc = WordApp.Documents.Open(ref file,

ref nullobj, ref nullobj, ref nullobj, ref nullobj, ref nullobj,

ref nullobj, ref nullobj, ref nullobj, ref nullobj, ref nullobj,

ref nullobj, ref nullobj, ref nullobj, ref nullobj, ref nullobj);

Word.Document doc1 = WordApp.ActiveDocument;

string m\_Content = doc1.Content.Text;

// the content is stored into the textbox.

txtResume.Text = m\_Content;

doc.Close(ref nullobj, ref nullobj, ref nullobj);

}

catch (Exception ex)

{

}

}

#region --- fnBindJobTitle()

private void fnBindJobTitle()

{

DataTable dtJobTitle = new DataTable();

string strJobTitle = string.Empty;

strJobTitle = "SELECT job\_id, job\_title FROM job\_details WHERE emp\_id = " + "'" + Request.Cookies["A1EmployerID"].Value + "'" + " AND JobStatus = 1";

try

{

dtJobTitle = dut.GetDataTable(strJobTitle);

if (dtJobTitle.Rows.Count != 0)

{

this.ddlJobList.DataSource = dtJobTitle;

this.ddlJobList.DataTextField = "job\_title";

this.ddlJobList.DataValueField = "job\_id";

this.ddlJobList.DataBind();

}

else

{

this.ddlJobList.Text = "No Jobs from Your Account";

}

}

catch (Exception ex)

{

}

}

#endregion

}

## **9.17 Submit.aspx.cs**

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

using System.Data.SqlClient;

using System.IO;

using BuildCarrier.Components;

public partial class Submit : System.Web.UI.Page

{

BuildCarrier.Components.DataUtility dut = new BuildCarrier.Components.DataUtility();

string S\_ID = string.Empty;

string job\_cat = string.Empty;

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

//job\_cat = Request.Cookies["Job"].Value.ToString();

if (Request.Cookies["Job"].Value.ToString() != "")

{

lblSectorHeading.Text = Request.Cookies["Job"].Value.ToString();

fnBindDetails();

}

}

}

#region------------fnBindDetails()-------------

private void fnBindDetail

if (lblSectorHeading.Text == "IT Sector")

{

lblDtls.Text = "HR sector is the latest entry in the market of India. HR sector has made an impact in the masses in the very short passage of time.Young boys and girls are stepping with an enormous energy into such a wonderful sector, as here one gets the full income for his labour and smart work. We are the perfect browser for any kind of services for the HR sector.We act as the traffic police and show you the rightmost path that leads to your goal.";

fnBindJob();

}

‘

else if (lblSectorHeading.Text == "Banking/Finance")

{

lblDtls.Text = "With the entry of private players in the banking and finance sector, the banking and finance sector is growing with a rapid speed.The reason is that they have simplified the policies and attracted common masses to step in. banking and finance sector is welcoming the merit and endowments and proffering them a good payment for their jobs.Before jumping in such a profitable sector, you must come to us first as we tell you what is best for you.We do this with the expertise gained with the vast experience in the market. No other can persuade the technical terms in most simplified words than us.";

fnBindJob();

}

else if (lblSectorHeading.Text == "HR Sector")

{

lblDtls.Text = "With the entry of private players in the banking and finance sector, the banking and finance sector is growing with a rapid speed.The reason is that they have simplified the policies and attracted common masses to step in. banking and finance sector is welcoming the merit and endowments and proffering them a good payment for their jobs.Before jumping in such a profitable sector, you must come to us first as we tell you what is best for you.We do this with the expertise gained with the vast experience in the market. No other can persuade the technical terms in most simplified words than us.";

fnBindJob();

}

else if (lblSectorHeading.Text == "B P O")

{

lblDtls.Text = "";

fnBindJob();

}

else if (lblSectorHeading.Text == "Telecome")

{

lblDtls.Text = "";

fnBindJob();

}

else if (lblSectorHeading.Text == "Sales/Marketing")

{

lblDtls.Text = "";

fnBindJob();

}

else if (lblSectorHeading.Text == "Medical")

{

lblDtls.Text = "";

fnBindJob();

}

else if (lblSectorHeading.Text == "Engineering/Technical")

{

lblDtls.Text = "";

fnBindJob();

}

else if (lblSectorHeading.Text == "Garment")

{

lblDtls.Text = "";

fnBindJob();

}

else if (lblSectorHeading.Text == "Fashion")

{

lblDtls.Text = "";

fnBindJob();

}

else

lblSectorHeading.Text = "A1 Services";

}

#endregion

protected void lbtnItSec\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "IT Sector";

lblDtls.Text = "IT Recruitment Agency with a proven track record of recruiting for many of the India's leading companies and we have a wide range of IT Jobs available.";

lblDescriptionCategory.Text = "IT Director Analyst Programmer , IT Manager, Software Developer, Programme Manager, Network Manager, Project Manager, Helpdesk Manager, Business Analyst, Helpdesk Support, Test Analyst, Network Support Analyst, Test Manager, Network Administrator, Oracle Database Administrator, SQL Server DBA, Web Developer, Web Designer, Webmaster eCommerce / Web Manager, Service Delivery Manager, IT Trainer ";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnBanking\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Banking/Finance";

lblDtls.Text = "With the entry of private players in the banking and finance sector, the banking and finance sector is growing with a rapid speed.The reason is that they have simplified the policies and attracted common masses to step in. banking and finance sector is welcoming the merit and endowments and proffering them a good payment for their jobs.Before jumping in such a profitable sector, you must come to us first as we tell you what is best for you.We do this with the expertise gained with the vast experience in the market. No other can persuade the technical terms in most simplified words than us.";

//imservicetype.ImageUrl = "~/images/Finance.jpg";

fnBindJob();

}

protected void lbtnHrSec\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "HR Sector";

lblDtls.Text = "HR sector is the latest entry in the market of India. HR sector has made an impact in the masses in the very short passage of time.Young boys and girls are stepping with an enormous energy into such a wonderful sector, as here one gets the full income for his labour and smart work. We are the perfect browser for any kind of services for the HR sector.We act as the traffic police and show you the rightmost path that leads to your goal.";

//imservicetype.ImageUrl = "~/images/Hr.jpg";

fnBindJob();

}

protected void lbtnCallCentre\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "B P O";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnTelcome\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Telecome";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnSales\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Sales/Marketing";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnMedical\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Medical";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnTechnical\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Engineering/Technical";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnGarment\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Garment";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void lbtnFashion\_Click(object sender, EventArgs e)

{

lblSectorHeading.Text = "Fashion";

lblDtls.Text = "";

//imservicetype.ImageUrl = "";

fnBindJob();

}

protected void GVJobDtls\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GVJobDtls.PageIndex = e.NewPageIndex;

}

protected void GVJobDtls\_PageIndexChanged(object sender, EventArgs e)

{

}

private void fnBindJob()

{

try

{

DataUtility dut = new DataUtility();

DataTable dtJob = new DataTable();

string strJob;

strJob = "select \* from job\_details where job\_category='" + this.lblSectorHeading.Text + "' and Job\_Status=1 order by exp\_req desc ";

dtJob = dut.GetDataTable(strJob);

if (dtJob.Rows.Count > 0)

{

//pnl1.Visible = true;

GVJobDtls.DataSource = dtJob;

GVJobDtls.DataBind();

pnlControl.Visible = true;

}

else

{

GVJobDtls.Dispose();

GVJobDtls.DataBind();

dtJob.Dispose();

//pnl1.Visible = false;

pnlControl.Visible = false;

}

}

catch (Exception ex)

{ }

}

protected void btnSelectedApply\_Click(object sender, EventArgs e)

{

Response.Redirect("BuildCarrier\_Login.aspx");

}

protected void btnAllApply\_Click(object sender, EventArgs e)

{

Response.Redirect("BuildCarrier\_Login.aspx");

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

Response.Redirect("BuildCarrier\_Login.aspx");

}

}

**10 Functions and Stored Procedures used in A1Jobs**

**FUNCTION** :- *SplittAll*

set ANSI\_NULLS ON

set QUOTED\_IDENTIFIER ON

go

CREATE FUNCTION [dbo].[SplitAll]

(

@RowData nvarchar(4000),

@SplitOn nvarchar(5)

)

RETURNS @RtnValue table

(

Id int identity(1,1),

Data int

)

AS

BEGIN

Declare @Cnt int

Set @Cnt = 1

-- Set WebId = @WebsiteId

While (Charindex(@SplitOn,@RowData)>0)

Begin

Insert Into @RtnValue(Data)

Select

Data = Convert(int, ltrim(rtrim(Substring(@RowData,1,Charindex(@SplitOn,@RowData)-1))))

Set @RowData = Substring(@RowData,Charindex(@SplitOn,@RowData)+1,len(@RowData))

Set @Cnt = @Cnt + 1

End

Insert Into @RtnValue (data)

Select Data = ltrim(rtrim(@RowData))

Return

END

**LIST OF PROCEDURES:-**

1. InsertSeekerDetails
2. InsertEmloyerDetails
3. InsertEmployerRequest
4. GetNewJobID
5. GetJobWeekWise
6. GetEmpID

*InsertSeekerDetails:-*

USE [BuildCareer]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[usp\_InsertSeekerDtls] Script Date: 03/29/2011 04:15:17 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROC [dbo].[usp\_InsertSeekerDtls]

@UserID AS NVARCHAR(40),

@FirstName AS NVARCHAR(40),

@LastName AS NVARCHAR(40),

@Address as nvarchar(1000),

@City as nvarchar(40),

@State as nvarchar(40),

@Contact\_No as bigint,

@Email as nvarchar(200),

@DOB as nvarchar(40),

@Edu\_Qualification as nvarchar(2000),

@Pro\_Qualification as nvarchar(2000),

@OtherQualification as nvarchar(2000),

@Experience as nvarchar(40),

@Job\_Category as nvarchar(2000),

@Password as nvarchar(200)

AS

--DECLARE @SeekerID AS VARCHAR(10)-- = 'BC'

if exists(select top 1 \* from JobSeeker\_Profile where Seeker\_Id = @UserID )

--SET @SeekerID = @SeekerID + Convert(char,(select COUNT(\*) from JobSeeker\_Profile) + 1)

SELECT 'Error: Seeker Mail-ID already exists !!'

else

--SET @SeekerID = @SeekerID + Convert(char,'01')

INSERT INTO JobSeeker\_Profile (Seeker\_Id,First\_Name,Last\_Name,Address,City,

State,Contact\_No,Email,DOB,Edu\_Qualification,Pro\_Qualification,

OtherQualification,Experience,Job\_Category,Password)

values (@UserID,@FirstName,@LastName,@Address,@City,

@State,@Contact\_No,@Email,@DOB,@Edu\_Qualification,@Pro\_Qualification,

@OtherQualification,@Experience,@Job\_Category,@Password)

GO

***InsertEmployerDetails:-***

USE [BuildCareer]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[insert\_Employer] Script Date: 03/28/2011 00:55:06 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[insert\_Employer]

@RowID bigint,

@Employer\_Id nvarchar(20),

@E\_Name nvarchar(50),

@Com\_Name nvarchar(50),

@Com\_Address nvarchar(100),

@City nvarchar(100),

@State nvarchar(100),

@Contact\_No bigint,

@Com\_Website nvarchar(100),

@Email\_Id nvarchar(100),

@Password nvarchar(100)

as

if not exists(select Employer\_Id from Employer where Employer\_Id=@Employer\_Id)

INSERT INTO Employer(RowID,Employer\_Id,E\_Name,Com\_Name,Com\_Address,City,State,Contact\_No,Com\_Website,Email\_Id,Password)

VALUES(@RowID,@Employer\_Id,@E\_Name,@Com\_Name,@Com\_Address,@City,@State,@Contact\_No,@Com\_Website,@Email\_Id,@Password)

else

select 'Already Exist'

GO

*InsertEmployerRequest:-*

set ANSI\_NULLS ON

set QUOTED\_IDENTIFIER ON

go

ALTER PROC [dbo].[InsertEmployerRequest]

@EmpID NVARCHAR(50),

@SeekerID NVARCHAR(50),

@JobID NVARCHAR(50),

@JobTitle NVARCHAR(50),

@Date NVARCHAR(20)

AS

DECLARE @RowID INT

SET @RowID = ( SELECT MAX( RowID ) FROM EmployerPendingRequest )

IF(@RowID IS NULL)

SET @RowID = 1

ELSE

SET @RowID = @RowID + 1

INSERT INTO EmployerPendingRequest ( RowID, EmpID, SeekerID, JobID, JobTitle, Date )

VALUES(@RowID, @EmpID, @SeekerID, @JobID, @JobTitle, @Date )

*GetNewJobID:-*

set ANSI\_NULLS ON

set QUOTED\_IDENTIFIER ON

go

ALTER PROC [dbo].[GetNewJobID]

AS

DECLARE @MaxRowID INT

DECLARE @NewJobID NVARCHAR(20)

SET @MaxRowID = (SELECT MAX(RowID) FROM job\_dtls)

IF( @MaxRowID IS NULL )

BEGIN

SET @NewJobID = 'Job1'

INSERT INTO job\_dtls (RowID, Job\_ID) VALUES (1, @NewJobID)

SELECT @NewJobID AS NowJobID

END --IF( @MaxRowID = 0 )

ELSE

BEGIN

SET @MaxRowID = @MaxRowID + 1

SET @NewJobID = LTRIM(RTRIM('Job' + LTRIM(RTRIM(CONVERT(NVARCHAR(30), @MaxRowID)))))

INSERT INTO job\_dtls (RowID, Job\_ID) VALUES (@MaxRowID, @NewJobID)

SELECT @NewJobID AS NowJobID

END

*GetJobWeekWise:-*

set ANSI\_NULLS ON

set QUOTED\_IDENTIFIER ON

go

-- GetJobWeekWise '7/9/2009'

ALTER proc [dbo].[GetJobWeekWise] @CurrentDate NVARCHAR(10)

AS

DECLARE @StartDate DATETIME

DECLARE @EndDate DATETIME

DECLARE @Day INT

DECLARE @EndDay INT

SET @StartDate = ( SELECT CONVERT( DATETIME, LTRIM(RTRIM(@CurrentDate)) ) )

SET @Day = ( SELECT DATEPART(dw, @StartDate))

PRINT @Day

SET @StartDate = ( SELECT DATEADD( dd, -(@Day-1), @StartDate ) )

SET @EndDay = 6 - @Day

SET @Day = @Day + @EndDay

PRINT @Day

SET @EndDate = ( SELECT DATEADD( dd, @Day, @StartDate ) )

PRINT @StartDate

PRINT @EndDate

DECLARE @TotalActiveJob INT

DECLARE @TotalInActiveJob INT

DECLARE @JobsPosted\_Week INT

DECLARE @JobsExpire\_Week INT

DECLARE @No\_Of\_Days INT

DECLARE @ExpiryDate DATETIME

--SET @ExpiryDate = (SELECT COUNT(\*) FROM )

SET @TotalActiveJob = (SELECT COUNT(\*) FROM job\_dtls WHERE JobStatus = 1)

SET @TotalInActiveJob = (SELECT COUNT(\*) FROM job\_dtls WHERE JobStatus = 0)

SET @JobsPosted\_Week = (SELECT COUNT(\*)from job\_dtls where convert(datetime,posting\_date) between @StartDate and @EndDate)

SET @JobsExpire\_Week = (SELECT COUNT(\*)from job\_dtls where convert(datetime,Expiry\_Date) between @StartDate and @EndDate)

IF( @TotalActiveJob IS NULL )

SET @TotalActiveJob = '0'

select @TotalActiveJob AS TotalActiveJob,

@TotalInActiveJob AS TotalInActiveJob,

@JobsPosted\_Week AS JobsPosted\_Week,

@JobsExpire\_Week AS JobsExpire\_Week,

job\_id,

job\_title,

job\_descriptoin,

e\_id,

job\_category,

exp\_req,

no\_of\_days,

posting\_date,

JobStatus

INTO Temp1

from job\_dtls where convert(datetime,posting\_date) between @StartDate and

@EndDate

IF @@Rowcount = 0

BEGIN

DECLARE @job\_id INT

DECLARE @job\_title INT

DECLARE @job\_descriptoin INT

DECLARE @e\_id INT

DECLARE @job\_category INT

DECLARE @exp\_req INT

DECLARE @no\_of\_days1 INT

DECLARE @posting\_date INT

DECLARE @JobStatus INT

select @TotalActiveJob AS TotalActiveJob,

@TotalInActiveJob AS TotalInActiveJob,

@JobsPosted\_Week AS JobsPosted\_Week,

@JobsExpire\_Week AS JobsExpire\_Week,

@job\_id AS job\_id,

@job\_title AS job\_title,

@job\_descriptoin AS job\_descriptoin,

@e\_id AS e\_id,

@job\_category AS job\_category,

@exp\_req AS exp\_req,

@no\_of\_days1 AS no\_of\_days,

@posting\_date AS posting\_date,

@JobStatus AS JobStatus

END

ELSE

select @TotalActiveJob AS TotalActiveJob,

@TotalInActiveJob AS TotalInActiveJob,

@JobsPosted\_Week AS JobsPosted\_Week,

@JobsExpire\_Week AS JobsExpire\_Week,

job\_id,

job\_title,

job\_descriptoin,

e\_id,

job\_category,

exp\_req,

no\_of\_days,

posting\_date,

JobStatus

from job\_dtls where convert(datetime,posting\_date) between @StartDate and @EndDate

*GetEmpID:-*

set ANSI\_NULLS ON

set QUOTED\_IDENTIFIER ON

go

ALTER PROC [dbo].[GetEmpID]

AS

DECLARE @MaxEmpID INT

DECLARE @NewEmpID INT

SET @MaxEmpID = (SELECT MAX(RowID) FROM Employers)

IF (@MaxEmpID IS NULL)

SET @NewEmpID = 1

ELSE

SET @NewEmpID = @MaxEmpID + 1

SELECT @NewEmpID AS NewEmpID

**11 CODE OPTIMIZATION & EFFICIENCY**

The below mentioned steps where taken to optimize the code:

1. **Use of Class Modules (Components):** Class modules were used for reusability of the code, this helps our application to achieve easy maintenance and separate compilation can be reflected to all the dependent modules.
2. **Use of General Variables and Objects in module:** Module was used for declaring general variables and objects (ex. Connection, Record set etc.). Single connection was opened in the general module so that common connection can be shared by record sets in different form modules.
3. **Check on use of unnecessary variables:** Precautions are taken to prevent the use of unnecessary variables or objects; same variables are used in case of holding the similar data.
4. **Opening/ Closing Objects:** Objects are assigned only when they are to be used to prevent wastage of memory, and it is closed when its necessity is assumed to be vanished.
5. **Disconnected Record sets:** Maximum resource is used to retain connection of record set with the connection object, so we took care to disconnect the record set and the connection is resumed when need is felt.
6. **Connection Pooling:** When connection is closed explicitly or implicitly is is not flushed from the memory and it is moved to a pool from where it is picked

again when a request for same connection comes to the application, rather than creating a fresh connection.

1. **SQL Optimization:** Extra care is taken for SQL query optimization, for example, rather than writing “SELECT \* FROM <Table Name>” we fetch the required fields only so that time to resolve the field names can be eliminated.
2. **Extensive use of Connection object:** in case of insertion, updation and deletion of the records mainly the connection object is used rather than record set so that the time can be reduced.

**12. VALIDATION CHECKS**

**(1) Date Validation**:The validation on date data type has been specified to be of the format DD/MM/YY. Any other format is unacceptable.

**(2) Time validation:** The validation on time data type has been specified to be of the format hours-minutes-seconds. Any other format is unacceptable.

**(3) Number field validation:** The field specified with number as then their data- type will not accept character.

**(4) User Authentication:** When a Customer/user logs on to the system to access data from tables and database, the Id & password needs to be checked.

**(5) Password change Validation:** Only authorized users are allowed to change the password and the process requires asking the old password before changing it to the new one.

**13. IMPLEMENTATIO AND MAINTENANCE**

**MAMAIAMAIMAINTENANCE**

# **IMPLEMENTATION:**

# Implementation uses the design document to produce code. Demonstration that the program satisfies its specifications validates the code. Typically, sample runs of the program demonstrating the behavior for expected data values and boundary values are required.

# Small programs are written using the model:

# Write/Compile/ Test

# It may take several iterations of the model to produce a working program. As programs get more complicated, testing and debugging alone may not be enough to produce reliable code. Instead, we have to write programs in a manner that will help insure that errors are caught or avoided.

# ***Top-down implementation:***

# Top down implementation begins with the user-invoked module and works toward the modules that do not call any other modules. The implementation may proceed depth-first or breadth-first.

# ***Bottom- Up implementation:***

# Implementation begins with modules that do not call any other modules

# and works toward the main program. Test harness (see below) is used to test individual modules. The main module constitutes the final test harness.

# ***Stub Programming:***

# Stub programming is the implementation analogue of top-down and stepwise refinement. It supports incremental program development by allowing for error and improvement. A stub program is a stripped-down, skeleton version of a final program. It doesn't implement details of the algorithm or fulfill all the job requirements. However, it does contain rough versions of all subprograms and their parameter lists. Furthermore, it can be compiled and run. Extensive use of procedures and parameter are the difference between stub programs and prototypes. Quick and dirty prototypes should be improved--they should be rewritten. A stub program helps demonstrates that a program's structure is plausible. Its procedures and functions are unsophisticated versions of their final forms, but they allow limited use of the entire program. In particular, it may work for a limited data set. Often the high-level procedures are ready to call lower-level code, even if the more detailed subprograms haven't even been written. Such sections of code are commented out. The comment brackets can be moved, call-by-call, as the underlying procedures are actually written.

# **Incremental program development:**

# As program become more complex, changes have a tendency to introduce

# unexpected effects. Incremental programming tries to isolate the effects of changes. We add new features in preference to adding new functions, and add new function rather than writing new programs. The program implementation model becomes:

# ------ Define types/compile/fix;

# ------ Add load and dump functions/compile/test;

# ------ Add first processing function/compile/test/fix;

# ------ Add features/compile/test/fix;

# ------ Add second processing function/compile/test/fix;

# ------ Keep adding features/and compiling/and testing/ and fixing.

**MAINTENANCE:**

Once the software is delivered and deployed, then *maintenance* phase starts. Software requires maintenance because there are some residual errors remaining in the system that must be removed as they discovered. Maintenance involves understanding the existing software (code and related documents), understanding the effect of change, making the changes, testing the new changes, and retesting the old parts that were not changed. The complexity of the maintenance task makes maintenance the most costly activity in the life of software product.

It is believed that almost all software that is developed has residual errors, or bugs, in them. These errors need to be removed when discovered that leads to the software change. This is called Corrective Maintenance. Corrective

maintenance means repairing, processing or performance failures or making alterations because of previously ill-defined problems. Software undergoes change

frequently even without bugs because the software must be upgraded and enhanced to include more features and provide more services. This also requires

modification of the software. The changed software changes the environment, which in turn requires further change. This phenomenon is called the *law of software evaluation*. Maintenance due to this phenomenon is called *adaptive maintenance*. Adaptive maintenance means changing the program function. Perfect maintenance means enhancing the performance or modifying the programs according to the user’s additional or changing needs. The keys to reduce the need for maintenance are:

1. More accurately defining the user’s requirement during system development.
2. Preparation of system documentation in a better way.
3. Using more effective ways for designing processing logic and communicating it to project team members.
4. Making better use of existing tools and techniques.
5. Managing the system engineering process effectively.

**13.1 TESTING TECHNIQUES AND TESTING STRATEGIES**

**14. TESTING (TESTING TECHNIQUES AND TESTING STRATEGIES)**

SYSTEM TESTING:

System testing is the expensive and time-consuming process. There are two strategies for testing software that we use for testing our system: Code Testing and Specification Testing. In Code testing, we developed those cases to execute every instructions and path in the program. In specification testing, we examined the program specification and then wrote test data to determine how the

program operates under specified condition. The different levels of testing are used in the testing process. The basic levels are unit testing, integration testing, system testing, and acceptance testing. These different levels of testing detect different types of faults. The different levels of testing are shown in figure in next page.

Client Acceptance

Needs Testing

Requirements System

Testing

Design Integration

Testing

Code Unit

Testing

**(Figure – 7. Levels of testing)**

We have tested each module separately i.e. have completed unit testing first and system testing was done after combining /linking all different Modules with different menus and thorough testing was done. Testing is a very important part of SDLC and takes approximately 50%of the time.

Once the system is a live one, Maintenance phase is important. Service after sale is a must and users/ clients must be helped after the system is implemented. If he/she faces any problem in using the system, one or two trained persons from developer’s side can be deputed at the client’s site, so as to avoid any problem and if any problem occurs immediate solution may be provided.

OVERVIEW OF TESTING:

Testing: Testing involves executing the program (or part of it) using sample data and inferring from the output whether the software performs correctly or not. This can be done either during module development (unit testing) or when several modules are combined (system testing).

Defect Testing: Defect testing is testing for situation where the program does not meet its fictional specification. Performance testing tests a system's performance or reliability under realistic loads. This may go some way to ensuring that the program meets its non-functional requirements.

Debugging: Debugging is a cycle of detection, location, repair and test. Debugging is a hypothesis testing process. When a bug is detected, the tester must form a hypothesis about the cause and location of the bug. Further examination of the execution of the program (possible including many returns of it) will usually take place to confirm the hypothesis. If the hypothesis is demonstrated to be incorrect, a new hypothesis must be formed. Debugging tools that show the state of the program are useful for this, but inserting print statements is often the only approach. Experienced debuggers use their knowledge of common and/or obscure bugs to facilitate the hypothesis testing process. After fixing a bug, the system must be reset to ensure that the fix has worked and that no other bugs have been introduced. This is called regression testing. In principle, all tests should be performed again but this is often too expensive to do.

TEST PLANNING:

Testing needs to be planned to be cost and time effective. Planning is setting out standards for tests. Test plans set out the context in which individual engineers can place their own work. Typical test plan contains:

OVERVIEW OF TESTING PROCESS:

>Requirements trace ability (to ensure that all requirements are tested)

>List of item to be tested

>Schedule

>Recording procedures so that test results can be audited

>Hardware and software requirements

>Constraints

# **OVERVIEW OF TESTING STRATEGIES:**

# Large system usually tested using a mixture of strategies. Different strategies may be needed for different parts of the system or at a stage of the process.

# ***Top-down testing:***

This approach tests high levels of system before detailed components. This is

an appropriate when developing the system top-down likely to show up structural

design errors early (and therefore cheaply) has advantage that a limited, working

system available early on. Validation (as distinct from verification) can begin early. Its

disadvantage is that stubs needs to be generated (extra effort) and might be

impracticable if component is complex (e.g. converting an array into a linked listunrealistic to generate random list; therefore end up implementing unit anyway). Tet output may be difficult to observe (needs creation of artificial environment). This is not appropriate for OO systems (except within a class).

***Bottom-up testing:***

This is opposite of top-down testing. This testing test low-level unit then works up hierarchy. Its advantages and disadvantages of bottom-up mirror those of top-down. In this testing there is need to write test drivers for each unit. These are as

reusable as the unit itself. Combining top-down development with bottom-up testing means that all parts of system must be implemented before testing can begin, therefore does not accord with incremental approach discussed above. Bottom-up testing less likely to reveal architectural faults early on. However, bottom-up testing of critical low-level components is almost always necessary. Appropriate for OO systems.

***Stress testing:***

Test system's ability to cope with a specified load (e.g. transactions per second). Plan tests to increase load incrementally. Go beyond design limit until system fails (this test particularly important for distributed systems (check degradation as network exchange data).

***Back-to-back testing:***

Comparison of test results from different versions of the system (e.g. compare with prototype, previous version or different configuration). Process - Run first system, saving test case results. Run second system, also saving its results. Compare results files. Note that no differences don’t imply no bugs. Both systems may have made the same mistake.

***Defect testing:*** A successful defect test is a test that causes the system to behave incorrectly. Defect testing is not intended to show that a program meets its specification. If tests don't show up defects it may mean that the tests are not

exhaustive enough. Exhaustive testing is not always practicable. Subset has to be defined (this should be part of the test plan, not left to the individual programmer). Possible methods:

* Usual method is to ensure that every line of code is executed at least once.
* Test capabilities rather than components (e.g. concentrate on tests for data loss over ones for screen layout).
* Test old in preference to new (users less effected by failure of new capabilities).>Test typical cases rather than boundary ones (ensure normal operation works properly).

Three approaches to defect testing. Each is most appropriate to different types of component. Studies show that black box testing is more effective in discovering faults than white-box testing. However, the rate of fault detection (faults detected per unit time) was similar for each approach. Also showed that static code reviewing was more effective and less expensive than defect testing.

***Black-box (Functional) Testing:***

Testing against specification of system or component. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a

# number of classes. E.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and

# output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested. Test cases are chosen to exercise each portion. Also test boundary cases (atypical, extreme, zero) since these frequently show up defects. For completeness, test all combinations of partitions. Black box testing is rarely exhaustive (because one doesn't test every value in an equivalence partition) and sometimes fails to reveal corruption defects caused by "weird" combination of inputs. Black box testing should not be used to try and reveal corruption defects caused, for example, by assigning a pointer to point to an object of the wrong type. Static inspection (or using a better programming language!) is preferable for this.

# ***White-box (structural) Testing:***

# Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has n control statements, there will be up to 2n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring

# assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically this will count how many times each statement is executed. At end, print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

# Data complexity not taken into account.

# Does not test all paths in combination.

# Really only possible at unit and module testing stages because beyond that complexity is too high.

# ***Interface testing:***

# Usually done at integration stage when modules or sub-systems are combined. Objective is to detect errors or invalid assumptions about interfaces between modules. Reason these are not shown up in unit testing is that test case may perpetuate same incorrect assumption made by module designer. Particularly important when OO development has been used. Four types of interface:

# 1. Parameter: data (or occasionally function references) passed from one unit to another.

# 2. Shared memory: block of memory shared between units (e.g. global variable) .One places data there and the other retrieves it.

# 3. Procedural: object-oriented or abstract data type form of interface, encapsulating several procedures.

# 4. Message passing: one sub-system requests a service by passing a message. Client-server interface also used by some OO architectures.

# Three common kinds of interface error:

# Interface misuse: caller gives wrong number/types/order of parameters or sends invalid message.

# Interface misunderstanding: caller misunderstanding specification of called component and provides or receives data in legal but unexpected form.

# Timing errors: producer/consumer of data operate at different speeds and data is accessed before being ready. "Race conditions".

# Common manifestations are when each unit assumes the other one is checking for invalid data (failure to check return status) and the consequences of when such a fault is propagated to other units.

# **TESTING PROCESS**

# Best testing process is to test each subsystem separately, as we have done in my project. Best done during implementation. Best done after small sub-steps of the implementation rather than large chunks. Once each lowest level unit has been tested, units are combined with related units and retested in combination. This proceeds hierarchically bottom-up until the entire system is tested as a whole.

# Typical levels of testing:

# Unit -procedure, function, method

# Module -package, abstract data type, class

# Sub-system - collection of related modules, cluster of classes, method-message paths

# Acceptance testing - whole system with real data (involve customer, user, etc)

# Alpha testing is acceptance testing with a single client (common for bespoke systems).

# Beta testing involves distributing system to potential customers to use and provide feedback. In, this project, Beta testing has been followed. This exposes system to situations and errors that might not be anticipated by us.

**15. SYSTEM SECURITY MEASURES**

It is critical in system development. The analyst has a responsibility to design a workable security system to protect the system from damage, error, and unauthorized access.

* Types of issues related to security:
* System Security
* System Integrity: Proper functioning of hardware & Programs
* System Privacy
* System Confidentiality
* Some threats to system security
* Errors & Omissions
* Disgruntled and dishonest employees
* Fire
* Natural Disaster
* External Attack
* Control Measures:
* Physical security from fire, flood etc
* Database integrity are data validation techniques
* A control measures three passwords, encryption and monitoring users on a regular basis.
* Review the software library periodically to ensure that a complete operating documentation exists for all applications.Also Recovery Measures should be kept in view of any system failure.
* Failures can be
* System crash
* Program bugs
* Data interruption
* Structural damage

We should have roll back facility to recover data. Other alternation can be banking

service, warn back up service etc.

**Reliability:** The system would be developed with user-friendly error messages that would pop-up in case of errors encountered in any of the user screens.

**Availability:** As the site could be accesses at any time and, 24hrs of work time except for times when the system is in the maintenance cycle or the system administrator has locked assess to the system due to specific reasons or the server is not responding at all.

**Correctness:** The correctness of the site will be thoroughly checked during the testing phase where all interfaces and files would be checked to see if they are operating as desired.

**Security:** Only the administrators could access the donors/acceptors record or

could view the suggestions given by the users as using login ids and password

could only access the back-end interface.

**Maintainability:** This site would be regularly maintained and updated to provide the user with the new information related to the blood banks and for donation purpose.

**Usability:** The system would be user-friendly, easily understandable and short user training would be sufficient. The skills that are required to use the system would include knowledge of operating the site and visiting web pages. The user could be trained to maintain the site and update the site on their own and if some help is required they can contact the web designers for this purpose.

**An online system**

In this high tech world, with the help of this site the donors/acceptors record could be easily maintained and accessible and search requests could be retrieved easily.

Thus by understanding the aforesaid threats to the system we have maintained following security and safety measures.

**SECURITY MEASURES:**

* User needs to be authenticated.
* An encryption mechanism is used to ensure better security.
* Only Supervisor is allowed to update or modify the BUILD CARRIER PLACEMENTS database. The end-users are not allowed to access user or customer details, which is ensured by; protection methods, such as password encryption.
* Access privileges have been granted to the users based on their roles in the organization.

**SAFETY MEASURES:**

A log based transaction system is maintained to ensure reliability of the system. The system maintains “Transaction Log” & “Error Log “. In case of a system failure/crash, the BUILD CARRIER PLACEMENTS retrieves the data from the Transaction Log. The problems encountered during the execution of BUILD CARRIER PLACEMENTS are recorded in the Error Log. The Problems encountered could be a network error, message delivery failure, and Database connectivity failures. Backup facility is also provided.

**15. COST ESTIMATION OF THE PROJECT**

The objective of the cost estimation is to enable the client or developer to perform a cost-benefit analysis and for project monitoring and control. The accuracy of the estimate depends on the amount of reliable information about the final product. When the product is delivered, the cost can be accurately determined, as all the data about the project and the resource spent can be fully known by then. The obtainable accuracy of the estimates as it varies with the different phases is shown in below figure – 8.

x

Feasibility Requirement System Detailed Coding and Accepted Analysis Design Design Testing Software

**(Figure – 8. Accuracy of cost estimation)**  **(Figure – 9. Graph of Accuracy of Cost Estimation)**

**Cost of Correcting errors:** According to the established S/W Engineering standard of estimating the cost of correcting errors the phase wise distribution of occurances of errors is as given below: -

Requirement Analysis 20%

Design 30%

Coding 50%

The cost of correcting errors of different phases is not the same and depends on when the error is detected and corrected. The relative cost of correcting requirement errors as a function of where they are detected is shown below figure10.

1000

100

50

10

5

2

Requirements Design Code Development Acceptance Operation

Test Test

Phase In Which Error Was Detected

**(Figure – 10. Cost of correcting errors)**

One can perform cost estimation at any point in the software life cycle. As the cost of the project depends on the nature and characteristics of the project, at any point, the accuracy of the estimate will depend on the amount of reliable information we have about the final product. The figure depicted below shows the accuracy of the cost estimation.On Size estimation of Schedule and Cost of the Project: this approach implies that size is the primary factor for cost; other factors have lesser effect. Here we will discuss one such model called the Constructive Cost Model (COCOMO) developed by Boehm. This model also estimates the total effort in terms of person-months of the technical project staff. The basic steps of this model are as follows:

* Obtain the initial estimate of the development effort from the estimate of thousands of delivered lines of source code (KDLOC).
* Determine a set of multiplying factors from different attributes of the project.
* Adjust the effort estimate by multiplying the initial estimate with all the multiplying factors.

The initial estimate also called nominal estimate is determined by an equation of the form used in the static single-variable models, using KDLOC as the measure of size. To determine the initial effort Ei in person-months the equation used is of the type

Ei = a\*(KDLOC)b

In COCOMO model the values of constants a and b are different with different type of projects. As our project is Organic type the values of a and b are 3.2

and 1.05 respectively. The total thousand delivered code (KDLOC) of our system has been estimated as around 2.

In order to determine the multiplying factors commonly known as cost driver attributes we have taken rating of these attributes according to our

requirements. From these, the effort adjustment factor (EAF) of our project has been estimated as 1.6.

Now the final efforts estimate, E, of our project is obtained by multiplying the initial estimate by the EAF.

i.e., E = EAF\*Ei

The project duration is estimated for an Organic project by the formula

D = 2.5 \* E0.38 and according to this formula I have estimated the project duration 6 months which has been depicted in the Gantt and PERT charts given as separate topic of this project document.

***Note:******This developed project for the partial fulfillment of our BCA programme thus the estimation of commercial terms has not been done.***

**17. PERT CHART AND GANTT CHART**

* **PERT CHART:** Program evaluation and review technique (PERT) is a project scheduling methods that can be applied to software development. A PERT chart is a graph-based chart. It can be used to determine the activities that form the critical path, which if delayed will cause the overall project to delay. This technique is driven by information developed earlier in project planning activities:
  + - Estimates of effort
* A decomposition of the product function
* The selection of the appropriate process model and task set
* Decomposition of tasks

PERT chart provides quantitative tools that allow the software planner to:

1. Determine the critical path;
2. Establish most likely time estimates for individual tasks by applying statistical models;
3. Calculate boundary times that define window for a particular task.

Boundary time calculations can be very useful in software project scheduling. Boundary time calculations lead to a determination of critical path and provide the higher authority with a quantitative method for evaluating progress as tasks are completed. The PERT chart is not conceptually as simple, and the representation is graphically not clear as Gantt charts. Its use is justified in large projects. PERT chart of this system (BUILD CARRIER PLACEMENTS) is shown in figure- (11) below.

D E

A

B F J K L

C

G H

**Figure- (11) PERT Chart of BUILD CARRIER PLACEMENTS**  Critical Path

Activity

Dummy Activity

Node

A = Order Computing Platform

B = Prepare Site

C = Review Specification

D = Install Equipment

E = Test Hardware

F = Training

G = Write Programs

H = Test Programs

I = Test Software

J = Convert System

K = Implementation Follow-up

L=Accept

**Some of the advantages of pert chart are as follows:**

1. It forces the manager to plan.
2. It shows the interrelationship among the tasks in the project and in particular clearly identifies the critical path of the project, thus helping to focus on it. The PERT chart has exposed to cope with a potential problem.
3. It exposes all possible parallelism in the activities & help in allocating resources
4. It allows scheduling and simulation of alternative schedules.
5. It enables the manager to monitor and control the project.

Despite these advantage, PERT is just a tool, and its use doesn’t

automatically guarantee the success of the project. The manager has much latitude in how PERT is used. PERT is highly developed methodology and full description of it is into computer programs and is available as computerized management system.

**GANTT CHART**

Gantt chart is a simple and effective scheduling technique. Gantt chart uses a calendar-oriented chart to represent the schedule. Gantt chart is also called a Timeline chart. A Gantt chart can be developed for the entire project. Each activity is represented as a bar in the calendar, starting from the date of the activity and ending at the ending date for that activity. The start and end of each activity becomes milestone for the project.

Once the information necessary for the generation of a Gantt chart has been input,the majority of software project scheduling tools produce project

tables- a tabular listing of all project tasks, their planned and actual start- and end date, and a variety of related information.

The Gantt chart in the figure is actually an enhanced version of standard Gantt charts. The white part of the bar shows the length of time each task is estimated to take. The gray part shows the slake time, that is, the latest time by which a task must be finished. One-way to view the slake time is that, if necessary, ewe can slide the white area over the gray area without forcing the start if next activity to be delayed.Gantt charts can take different forms depending on their intended use. They are for resource scheduling. When a Bar chart is used as a project control method milestones or check points usually are placed at the completion of each task (they may also placed with in task). They indicate the completion of particular task and the project are on schedule; when a checkpoint is reached, the task just completed and the entire project are reviewed and evaluated. Reviewers ask where resources allocated have been properly utilized and indication of which task must be completed before others are begun and project costs must be accumulated and evaluated using other methods.

The main drawback of the Gantt chart is that it does not depict the dependency relationships among the different activities. Hence, the effect of slippage in one activity on other activities or on the overall project schedule cannot be determined. It

is sufficient for small and medium-sized projects. It is heavily used and easy to understand. The Gantt chart for this system (BUILD CARRIER PLACEMENTS) is shown in figure- (12).

Analysis

Design

Coding

Testing

Installation

Operation

Maintenance

April May June July August Sept. Oct. Nov. D.

Time in Month

**(Figure - 12.GANTT CHART of BUILD CARRIER PLACEMENTS)**

Once we have estimated of the effort and time requirement for the different phases a schedule for the project can be prepared. This schedule will then be used later for monitoring the progress of the project.

**18. FUTURE SCOPE OF THE PROJECT**

1. Enhancement of our proposed system would be in terms of connectivity with the rest of the departments so that information is made available throughout the organization which leads to efficiency and speeding up of a number of processes in the organization.

Members in the organization will access this system at network level.

1. After successfully implementing the system, it can be added in the system that automatically displays all the Jobs Status along with the Employers and Job Seekers when the Admin Login to the System.
2. When the developed system runs successfully, it can be also be used in other areas with certain changes according to the requirements of that areas.
3. Online Seeker/Employer register facility could be brought on Web that would enable the Seeker/Employer to register and in one side the Employers will be able to post the jobs while the Job Seeker will be able to apply for the Jobs in the system itself by entering their Id(s)., thus reducing the dependencies between Job seekers and Employers.
4. egration of **OLIP** (On-Line Interview Process) for automated telephonic and Cam’s Interview between the Employers and Job Seeker for any job will even reduce resources used for conducting interviews and many other factors.

\*\*\*\*\*\*\*\*

**Scope of future application**

**19. BIBLIOGRAPHY**

1. **Software Engineering A Practitioner’s Approach**: ROGER S. PRESSMAN
2. **Programming in ASP.NET :** NIIT.
3. **An Integrated Approach to Software Engineering**: PANKAJ JALOTE
4. **Professional Active Server Pages 3.0**: Wrox Press Ltd.: 15 Authors
5. **HTML: *Sams* Teach Your Self**: Dick Oliver
6. **C# Programming**: Wrox Press Ltd.
7. **SQL Server 2000; Mastering SQL Server 2000**: Mike Gunderloy, Joseph L. Jorden.
8. **Structured Analysis and Design** (**CS-05**) of IGNOU
9. http://www.webdev.wrox.co.uk
10. http://www.microsoft.com

**GLOSSARY**

**URL:** A URL contains a web server or network location, path, and file name. A URL also identifies the protocol that will handle the file, such as HTTP, FTP, or FILE. An absolute URL contains the full address. A relative URL has one or more missing parts. The missing information is taken from the page that contains the URL, e.g., if the protocol and domain are missing the web browser uses the protocol and domain of the current page.

**ActiveX Data Objects (ADO):** A high-level data access programming interface to an underlying data access technology (such as OLE DB), implemented by using the Component Object Model (COM).

**File Transfer Protocol (FTP):** The protocol used for copying files to and from remote computer systems on a network using Transmission Control Protocol/Internet Protocol (TCP/IP), such as the Internet. This protocol also allows users to use FTP commands to work with files, such as listing files and directories on the remote system.

**Hypertext Transfer Protocol (HTTP): The** client/server protocol used to access information on the World Wide Web.

**Localhost:** A placeholder for the name of the computer on which a program is running; localhost uses the reserved loopback IP address 127.0.0.1.

**Server:** A term used for any of the following: a computer on a network that sends files to, or runs applications for, other computers on the network; the software that runs on the server computer and performs the work of serving files or running applications; or, in object-oriented programming, a piece of code that exchanges information with another piece of code upon request

**Web page**   
A World Wide Web document. A Web page typically consists of an HTML file, with associated files for graphics and scripts, in a particular directory on a particular computer (and thus identifiable by a URL).

**Web server**   
In general terms, a computer equipped with the server software that uses Internet protocols such as HTTP and FTP to respond to Web client requests on a TCP/IP network.

**Internet**   
Abbreviation for Internet work. A set of dissimilar computer networks joined together by means of gateways that handle data transfer and the conversion of messages from the sending network to the protocols used by the receiving networks. These networks and gateways use the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols.Originally part of the Defense Advanced Research Projects Agency (DARPA), operated by the U.S. Department of Defense.

**\*\*\*\*\*\*\*\*\*\*\***