CHAPTER–1

INTRODUCTION

1.1 Java: The Programming language

The Java programming language happens to be a high level language. There are certain characteristics that define this language; which are often defined as “Java buzzwords”. Below are the following buzzwords of the language:

- Simple: Provides simplicity to the programmer
- Object Oriented: It is built around an object ; which is an entity with some characterstic and a certain behaviour
- Distributed : provides the concept of shared resources
- Interpreted: slow functioning of interpreter over a complier is overcome by approach called JIT(Just In Time)
- Robust : portability is the reason
- Secure :Due to abstraction property
- Architecture neutral: provides platform independency. Bytecode is responsible for the same
- Portable: also responsible for platform independency
- High Performance: yields better performance
- Multithreading : added and a major feature, light weight threads

You either compile or interpret a program in most programming languages, so that you can run it on your computer. The Java programming language is unusual in that a program into an intermediate language called Java byte codes - the platform independent codes interpreted by the interpreter on the java platform.

Why to use java

- First it provides a rich platform
- Second, a single source code
- Also supports lots of open utilities
- And Cross Platform
- Lastly , Standardized API

1.2 HTML: For Front End

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

1.3 CSS: For Front End

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of

web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

1.4 JAVA SCRIPT: For Front End

JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, object-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS,

JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMAScript specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA.

As a multi- paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

1.5 About the project

Today every work is getting more and more specialized. Hiring a specialist for any work has many advantages like work is done more efficiently and faster. These specialists have more experience in the desired field as compare to an ordinary workers. For example today I have a requirement of translating a educational video into four languages within two weeks. This work can be done in our country without much difficulty. But I do not know who are people who can do that work. In today’s fast world where things are to be done yesterday ,the number

of such requirements across the country would be immense. Here again the issue of matching of the work requirement with the people who can do the work.

OBJECTIVE :

The main objective is to provide a specialized worker to the customer as getting work done by a specialist have many advantages over an ordinary worker.

The project contain following modules:-

Module 1 – Freelancer Registration

Module 2 – User Window

Module 3 – Customer Window

Module 4 – Profile

Module 5 – Change Password

Module 6 – Logout Module

Module 7 – Adminstrator Login

Module 8 – Worker Avalible

Module 9 – Assigned Work

Module 10 – List of Customers

Module 11 –Do hire Registration

Module 12 – Forget Password

CHAPTER – 2

SOFTWARE AND HARDWARE REQUIRNMENT

2.1 Hardware requirement:

Minimum Requirements

Processors: Intel Pentium Pro or processor running at 133Hz

Hard Disk: 128 GB Hard Disk

RAM: Client Level -minimum 512 MB

Recommended Requirement for peak performance

Processors: Intel Pentium III running at 450 Hz

RAM: Client Level -minimum 1 GB

Display Type: SVGA Color Monitors

Keyboard: Enhanced 104 Standard

Mouse: PS/2 Button /USB mouse

2.2 Software requirement:

Operating System: WINDOWS 2003/XP/7/8/10

Tools Used:

For Runnable File : JAVA Runtime Enviroment (JRE)

For Coded File: JCreator for java

Front end: HTML, CSS, JAVA SCRIPT

Back end: Oracle, JAVA Servlets, Jsp

CHAPTER – 3

REVIEW OF LITERATURE

3.1 HTML: For Front End

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

3.2 CSS : For Front End

Before CSS, nearly all presentational attributes of HTML documents were contained within the HTML markup. All font colors, background styles, element alignments, borders and sizes had to be explicitly described, often repeatedly, within the HTML. CSS lets authors move much of that information to another file, the style sheet, resulting in considerably simpler HTML. For example, headings (h1 elements), sub-headings (h2), sub-sub-headings (h3), etc.,

are defined structurally using HTML. In print and on the screen, choice of font, size, color and emphasis for these elements is presentational.

Before CSS, document authors who wanted to assign such typographic characteristics to, say, all h2 headings had to repeat HTML presentational markup for each occurrence of that heading type. This made documents more complex, larger, and more error-prone and difficult to maintain. CSS allows the separation of presentation from structure. CSS can define color, font, text alignment, size, borders, spacing, layout and many other typographic characteristics, and can do so independently for on-screen and printed views. CSS also defines non-visual styles, such as reading speed and emphasis for aural text readers. The W3C has now deprecated the use of all presentational HTML markup.

CSS was first proposed by Håkon Wium Lie on October 10, 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside World Wide Web Consortium resulted in the first W3C CSS Recommendation (CSS1) being released in 1996. In particular, Bert Bos' proposal was influential; he became co-author of CSS1 and is regarded as co-creator of CSS.

3.3 JAVA SCRIPT : For Front End

JavaScript typically relies on a run-time environment (e.g., a Web browser) to provide objects and methods by which scripts can interact with the environment (e.g., a webpage DOM). It also relies on the run-time environment to provide the ability to include/import scripts (e.g., HTML <script> elements). This is not a language feature per se, but it is common in most JavaScript implementations.

JavaScript processes messages from a queue one at a time. Upon loading a new message, JavaScript calls a function associated with that message, which creates a call stack frame (the function's arguments and local variables). The call stack shrinks and grows based on the function's needs. Upon function completion, when the stack is empty, JavaScript proceeds to the next message in the queue. This is called the event loop, described as "run to completion" because each message is fully processed before the next message is considered. However, the language's concurrency model describes the event loop as non-blocking: program input/output

is performed using events and callback functions. This means, for instance, that JavaScript can process a mouse click while waiting for a database query to return information.

A common misconception is that JavaScript is similar or closely related to Java. It is true that both have a C-like syntax (the C language being their most immediate common ancestor language). They also are both typically sandboxed (when used inside a browser), and JavaScript was designed with Java's syntax and standard library in mind. In particular, all Java keywords were reserved in original JavaScript, JavaScript's standard library follows Java's naming conventions, and JavaScript's Math and Date objects are based on classes from Java 1.0,[116] but the similarities end there.

Java and JavaScript both first appeared on 23 May 1995, but Java was developed by James Gosling of Sun Microsystems, and JavaScript by Brendan Eich of NetScape Communications.

3.4 JAVA at the Back end:

Originally developed by James Gosling at Sun Microsystem which has recently merged with Oracle Corporation), Java was actually released in the year 1995. Deriving its syntax and various other features from C and C++, while fewer low-level facilities than either of them ,

JAVA gained popularity by addition of several new features to the it such as multithreading and the concepts like packages etc. But what become its biggest highlight was its platform independency. Applications get compiled to bytecode; forming a class file run on JVM.The computer architecture is not considered .That is how the feature works .Java is object oriented language which means the entire execution and work is done around an entity having some characteristics of itself and a certain behavior, the code or procedure is not given priority. Its specification is that it has few implementations dependencies also basis of WORA concept that stands for "write once ,run anywhere"(WORA), that means there is no need for re compilation of coder.

Editions of Java

There are 3 editions of Java:

J2SE: Standard edition also called “CORE JAVA”

J2EE: Enterprise edition also called “ADVANCED JAVA”

Java Servlet

A Java servlet processes or stores a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol. Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL rewriting.

3.5 Ms Access at the Back End

Microsoft Access is a database management system (DBMS) from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions or sold separately.

Microsoft Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases.[1]

Software developers, data architects and power users can use Microsoft Access to develop application software. Like other Microsoft Office applications, Access is supported by Visual Basic for Applications (VBA), an object-based programming language that can reference a variety of objects including DAO (Data Access Objects), ActiveX Data Objects, and many other ActiveX components.

CHAPTER – 4

SOFTWARE REQUIREMENT ANALYSIS

4.1 Modules And Their Functionality

1ST MODULE

- The first module is Registration.
- Here Person can register Directly .
- Plus Directly Login from Here.
- It is also having a Horizontal navigation bar.

2nd MODULE

- The second module is user window.
- Here If a user is valid then he knows about the jobs for which he is short Listed.

3rd MODULE

- The third module is the Customer window.
- Here the Functionality is provided to the customers so that they can see the workers available in nearly Location as per Filter Selected.

4th MODULE

- The fourth module is the Profile.
- The respected Person can see the profile and all the detaills for itself.

5th MODULE

- The fifth module is the Change Password.
- In this module a person can change his password using confirmation of old password.

6th MODULE

- The sixth module is the Logout Module.

7th MODULE

- The seventh module is of Adminstrator Login.
- Adminstrator can login and can see all the internal details of the activities going in the Organisation.

8th MODULE

- The eighth module is List of Job seekers avalible..
- This module provides avaliblity of all the specialised workers avalible or registered on the website but on to the Admin Login.

9th MODULE

- The ninth module is the List of all the Assigned Work
- All the work provided to which worker.

10th MODULE

- The tenth module is the List of Customers.
- It provides admin the data about all the Customers registered.

11th MODULE

- The sixth module is the Customer Registration.
- If the user is new to the website and do not have the username and password for the login then he/she will have to fill few details and will be provided with the username and the password.

12th MODULE

- The module is the Forgot Password.
- If any user forgets his/her password then it can be retrieved back by providing some basic details like the username,email address and the security question's answer

CHAPTER - 5

SOFTWARE DESIGN

5.1 Data flow diagram (DFD)

A graph showing the flow of data values from their source in objects through process that transport them to their destination in other objects is a Data flow diagram.

It is the basic or preliminary step which will create an overview of the entire system.

It will give a bird's eye view of the entire system. It shows that how the modules will open and flow in which the actor will get the modules according to the data given by him.

This is the given flow of the compleate project in the diagram described below :-

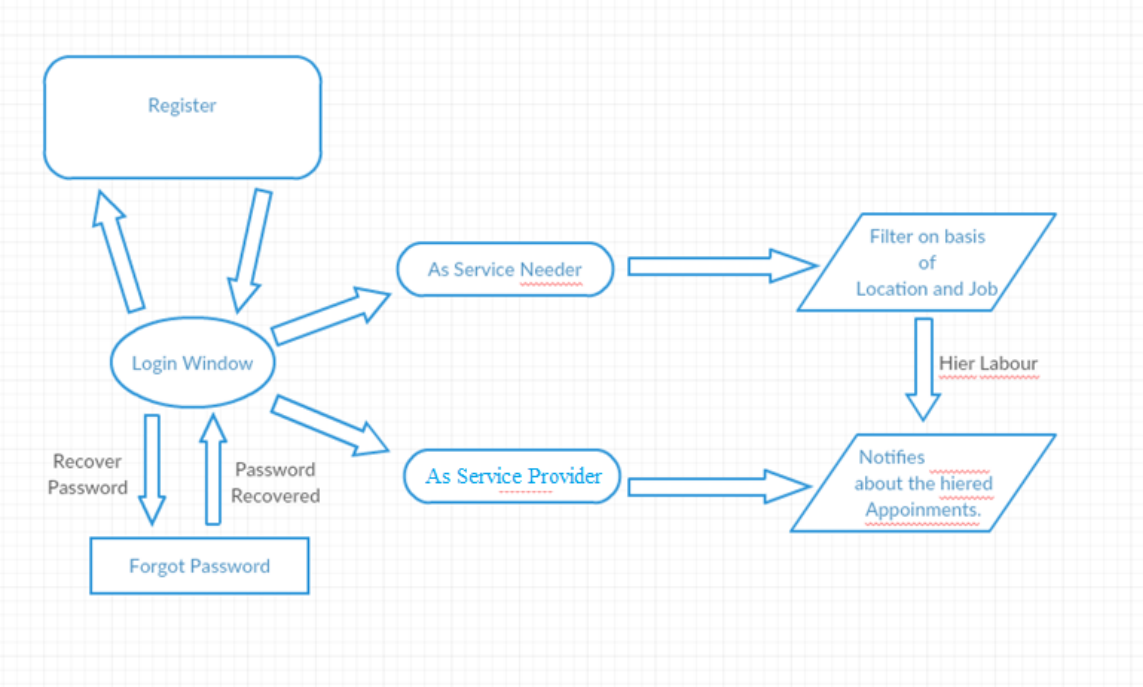

Figure 5.1 Flowchart Diagram

STEPS :-

- The user will at first visit to the homepage of the website.
- If the user is a registered then they can directly log in with the help of the username and password and if not then the user will have to register themselves.
- The User can see the appointments got by the Customers.
- Accordingly the respective Customer can also book the Person on Basis of Search Filters.
- Both have authority to change there respective profile details and passwords.
- If the particular job to a particular person is not available then he can send the information to the Administrator about his requirements.

5.2 Database:

Tables and datatypes: -

Table Needer:-

Needer

UserId	Name	Address	City	Pno	Username	Password	Favplayer	Add New Field
1	abc	igigig	Sonipat	999136161	awe	jbnnblln		
2	abc	kmkh	Sonipat	688786	hvhvh	dydyd		
3	absjb	wkkwdqkl	Sonipat	234424	jbdjjkq	ejkwekj		
4	absjb	wkkwdqkl	Sonipat	234424	aa	aa		
5	absjb	wkkwdqkl	Sonipat	234424	jbdjjkq	ejkwekj		
6	absjb	wkkwdqkl	Sonipat	234424	jbdjjkq	fsfwfwfe	slflsnlf	
7	Aman	690 Model tow	Panipat	999999999	Aman19	aman	aman	

Table 5.2.1: Needer

Table Labour :-

Needer | Labour

Userid	Name	Location	Pno	Skills	Pay	Per	Username	Password	Favplayer
101	Abhinav	Samalkha	92564646	Electrician	1504	Day	Abhinavj796	123123	abhinav
102	Abhinav	Samalkha	92564646	Electrician	1504	Day	Abhinavj796	123123	abhinav
103	Abhinav	Samalkha	92564646	Electrician	1504	Day	Abhinavj796	123123	abhinav
104	Asd	Sonipat	2323	Plumber	2323	Day	dfwf	fwfef	w43ew
105	Ajay Mishra	Panipat	92578456	Carpenter	800	Day	Ajay19	ajay	ajay

Table 5.2.2: Labour

Table Login :-

Needer | Labour | Login

Userid	Username	Password	Favplayer	Type	Add New Field
101	Abhinavj796	123123	abhinav	Labour	
102	Abhinavj796	123123	abhinav	Labour	
103	Abhinavj796	123123	abhinav	Labour	
104	abc	xyz	w43ew	Labour	
105	Ajay19	123	ajay	Labour	
4	aa	aaa	dsdsf	Needer	
5	jbdjjkq	ejkwekj		Needer	
6	jbdjjkq	fsfwfwfe	slflsnlf	Needer	
7	Aman19	aman	aman	Needer	

Table 5.2.3: Login

Table Work Assigned :-

Needer | Labour | Login | work

labourid	neederid	custname	location	doj	nod	wage	mobno
234424	101	6	jbdjjkq	wkkwdqkl	15-06-17	2	3008
234424	101	6	jbdjjkq	wkkwdqkl	18-01-18	12	18048
999999999	105	7	Aman19	690 Model tow	19/01/18	14	11200
101	16	trdrtd	tydh	fjfjfj	kfii	15	hfgf
107	165	ddh	sg	rrsgsg	dhgdhgd	trwrtwt	trsgfsq
105	7	Aman19	690 Model tow	1/07/17	10	8000	999999999
106	strstr	ddt	ytdthd1	dtyd	dtfh	fhjjhgk	kgk

Table 5.2.4: Work Assigned

CHAPTER – 6

CODING

Code for the homepage and login page :-

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/my.css">

<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css
" rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"><
/script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>



<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script>
<script src="js/bootstrap.min.js"></script>



<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.m
in.css" >
<!-- Optional theme -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" >
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.j
s"></script>
<link rel="stylesheet" href="form.css" >
<script src="form.js"></script>
```

```
<link rel="stylesheet" type="text/css"
href=https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css>
<link rel="stylesheet" type="text/css" href="custom.css">
<link rel="stylesheet" type="text/css"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">



<style>
body {
position: relative;
}
#section1 {padding-top:50px;height:500px;color: #fff; background-
color: #1E88E5;}
#section2 {padding-top:50px;height:500px;color: #fff; background-
color: #a3c2c2;}
#section3 {padding-top:50px;height:500px;color: #fff; background-
color: #ff9800;}
#section41 {padding-top:50px;height:500px;color: #fff; background-
color: #00bcd4;}
#section42 {padding-top:50px;height:500px;color: #fff; background-
color: #009688;}

.btn-file {
position: relative;
overflow: hidden;
}
.btn-file input[type=file] {
position: absolute;
top: 0;
right: 0;
min-width: 10%;
min-height: 10%;
font-size: 100px;
text-align: right;
filter: alpha(opacity=0);
opacity: 0;
outline: none;
background: white;
cursor: inherit;
display: block;
}

#img-upload{
width: 40%;
}

</style>
```

```
</head>
<body data-spy="scroll" data-target=".navbar" data-offset="50">

<nav class="navbar navbar-inverse navbar-fixed-top">
<div class="container-fluid">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#myNavbar">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="#">The Perfect Work</a>
</div>
<div>
<div class="collapse navbar-collapse" id="myNavbar">
<ul class="nav navbar-nav">
<li><a href="#section1">Home</a></li>
<li><a href="#section2">Register</a></li>
<li><a href="#section3">Our Team</a></li>
<li><a href="#section4">Contact Us</a></li>
<li class="dropdown"><a class="dropdown-toggle visible-xs" data-
toggle="dropdown" href="#"><i class="glyphicon glyphicon-user"></i>
Login<span class="caret"></span></a>
<ul class="dropdown-menu">
<li>
<form>
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-
user"></i></span>
<input id="email" type="text" class="form-control" name="email"
placeholder="Email">
</div>
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-
lock"></i></span>
<input id="password" type="password" class="form-control"
name="password" placeholder="Password">
</div>
<a href="#">Forgot Password?</a>
<label class="radio-inline" style="color:white">
<input type="radio" name="category">Get Hired
</label>
<label class="radio-inline" style="color:white">
<input type="radio" name="category">Do Hire
</label>
<input type="submit" class="btn btn-primary radio-inline">
</form>
</li>
</ul>
```

```
</li>
</ul>
</div>
</div>
</div>
</nav>

<div id="section1" class="container-fluid" style="background-
image:url('image/b1.jpg'); background-size:cover">
<h1>The Perfect Work:</h1>
<div class="col-xs-8">
<h2>CAREERS</h2>
<p>
If there’s one thing that unites the people who work here, it’s this:
we believe that bicycles change lives. We’re always looking for
passionate people—in all disciplines, from engineers and designers to
sales and administration roles—who share in this belief and are
looking to play an active role in inspiring and getting more people
on bikes. Are you ready to make a difference and move your career
forward?
</p> </div>
```

Code for the Admin :-

```
<div id="section" style="background-
image:url('508374604.jpg');background-size:100% 100%;">
<br><br/>
<CENTER><img src="ad5.jpg">
<Br><Br>
<Font Color = 'Brown' >
<H2> Administrator Page</H2> </Font>
<Form Method ="Post" Action = "Check1" OnSubmit="return check(this)">
<Font Color = 'Red'>
User Name</Font>
<Input Type = "text" Value="" Name = "nm">
<Br><Br><BR> <Font Color = 'Red'> Password </Font>
<Input Type = "password" Value="" Name = "pass">
<Br><Br>
<Font Color = 'Green'>
<Input Type = "submit" Value="Login" >
</Font>
</Form>
<BR>
<A HREF = "index.html"> Login Page </A>

</Center>
<marquee><font size =6 color = brown ><i>Specify Login
information</i></font></marquee>
```

```
</div>
```

Code for the college Admin Home page:-

```
<div id="section" style="background-
image:url('508374604.jpg');background-size:100% 100%;">
<br><br/>
<CENTER><img src="ad5.jpg">
<Br><Br>
<Font Color = 'Brown' >
<H2> Administrator Page</H2> </Font>
<Form Method ="Post" Action = "Check1" OnSubmit="return check(this)">
<Font Color = 'Red'>
User Name</Font>
<Input Type = "text" Value="" Name = "nm">
<Br><Br><BR> <Font Color = 'Red'> Password </Font>
<Input Type = "password" Value="" Name = "pass">
<Br><Br>
<Font Color = 'Green'>
<Input Type = "submit" Value="Login" >
</Font>
</Form>
<BR>
<A HREF = "index.html"> Login Page </A>

</Center>
<marquee><font size =6 color = brown ><i>Specify Login
information</i></font></marquee>
</div>
```

Servlet code for the login page :-

```
<div id="section" style="background-
image:url('508374604.jpg');background-size:100% 100%;" >

<Center>

<Form Method ="Post" Action = "Register" OnSubmit="return
check(this)">
<br>
<b><i> Needer Registration Form</i></b>
<fieldset>
<Table>
<tr><td> Name </td> <td><Input Type = "text" Value="" Name =
"cnm"></td></tr>
<tr><td> Address </td> <td><TextArea Value="" Name =
"add"></Textarea></td></tr>
<tr><td> City</td> <td>
<Select Name ="city" >
```

```
<option value = "Sonipat" Selected> Sonipat
             &nbs
p      
<option value = "Panipat"> Panipat
<option value = "Samalkha"> Samalkha
<option value = "Gaunaur"> Gaunaur
<option value = "Rai"> Rai
</Select></td></tr>
<tr><td> Candidate Phone </td> <td><Input Type = "text" Value="" Name
= "ph"></td></tr>

<tr><td> User Name </td> <td><Input Type = "text" Value="" Name =
"nm"></td></tr>

<tr><td> Password </td> <td><Input Type = "password" Value="" Name =
"pass"></td></tr>

<tr><td>  Fav. Player  </td> <td><Input Type = "text" Value="" Name =
"pnm"></td></tr>
<tr><td>

<Input Type = "submit" Value="Register" style="width:
100%;"></td></tr></table>
</fieldset>
</Form>

</Center>
<marquee><font size =6 color = brown ><i>Specify Login information
for Registration</i></font></marquee>
</div>
```

Code for the Contact Us :-

```
<div id="section" style="background-
image:url('508374604.jpg');background-size:100% 100%;">

<h2>CONTACT US</h2>
<p>You can contact us anytime whenever you have a problem.we will be
happy to help you.</p>
<br>

<div class="row">
<div class="column">
<div class="card">
<img src="cu1.jpg" alt="Contactus1" style="width:100%">
<div class="container">
<h2>ADDRESS</h2>
<p class="title">OFFICE</p>
<p>D-216,SECTOR 63,Noida,Uttar Pradesh,822101</p>
<p>P.I.E.T,Samalkha,Panipat,haryana,132102</p>
```

```
</div>
</div>
</div>

<div class="column">
<div class="card">
<img src="cu2.png" alt="Contactus2" style="width:100%">
<div class="container">
<h2>Call us</h2>
<p class="title">TELEPHONE</p>
<p><b>Landline</b> - 011-123456</p>
<p><b>mobile</b> - 0123456789</p>
<p><b>Toll free</b> -012001200</p>
</div>
</div>
</div>
<div class="column">
<div class="card">
<img src="cu3.jpg" alt="Contactus3" style="width:100%">
<div class="container">
<h2>Write Us</h2>
<p class="title">Email id</p>
<p>abc.compleatesolution.gmail.com</p>
<p>compleatesolution@gmail.com</p>
</div>
</div>
</div>
</div>

</div>
```

Servlet Code Labour Registration:

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Register1 extends HttpServlet
{
Connection con;
PreparedStatement ps1,ps2;
Statement stmt;
ResultSet rs;
int cn;
public void init() throws ServletException
{
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection("jdbc:odbc:ab","","");
```

```
ps1 = con.prepareStatement( "Insert into Labour values
(?,?,?,?,?,?,?,?,?,?)");
ps2 = con.prepareStatement( "Insert into Login values (?,?,?,?,?)");

stmt = con.createStatement();

rs = stmt.executeQuery ("Select  Max(Userid) + 1  from Labour");
if(rs.next( ))
{
cn = rs.getInt(1);
System.out.println(cn);
if(cn == 0)
cn=101;
}
}catch(Exception e)
{
System.out.println("Error");
}
}

public void doPost(HttpServletRequest request,
HttpServletResponse response)
throws IOException,ServletException
{
response.setContentType("text/HTML");
PrintWriter out = response.getWriter();

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

String add=request.getParameter("add");
String ph=request.getParameter("ph");

String skill=request.getParameter("skill");
String pay=request.getParameter("pay");
String per=request.getParameter("per");


String nm=request.getParameter("nm");
String pass=request.getParameter("pass");
String fav=request.getParameter("fav");

try{
ps1.setInt(1, cn);
ps1.setString(2, cnm);
ps1.setString(3, add);
ps1.setString(4, ph);
ps1.setString(5, skill);
ps1.setString(6, pay);
ps1.setString(7, per);
ps1.setString(8,nm);
```

```
ps1.setString(9,pass);
ps1.setString(10,fav);
ps1.executeUpdate();

ps2.setInt(1, cn);
ps2.setString(2, nm);
ps2.setString(3, pass);
ps2.setString(4, fav);
ps2.setString(5, "Labour");

ps2.executeUpdate();

}catch(Exception e)
{
out.println("Unable to store information"+e);

return;
}
out.println("<HTML>");
out.println("<HEAD>");

out.println("     <style>\n" +
"#header {\n" +
"    background-color:#0073e6;\n" +
"    color:white;\n" +
"    padding:15px;\n" +
"    height:50px;\n" +
"}    \n" +
" .right {\n" +
"    position: absolute;\n" +
"    right: 0px;\n" +
"    width: 700px;\n" +
"    padding:23px;\n" +
"   text-align:center;\n" +
"}\n" +
".left {\n" +
"    float:left;\n" +
"    position: absolute;\n" +
"    left: 0px;\n" +
"    width: 700px;\n" +
"    }\n" +
"    \n" +
"   #footer {\n" +
"    background-color:black;\n" +
"    color:white;\n" +
"    clear:both;\n" +
"    text-align:center;\n" +
"   padding:5px;                                    \n" +
"}\n" +
"    </style>\n" +
```

```
"    ");

out.println("<TITLE>Successful</TITLE>");
out.println("</HEAD>");
out.println("<Body Bgcolor = \"#C0C0C0\" text=\"#ffffcc\">");

out.println("  <div id=\"header\">\n" +
"      <div  class=\"left\">        <a
href=\"index.html\"><img       src=\"mgnrega.gif\"       width=\"600\"
height=\"50\" alt=\"c6\"/></a></div>\n" +
"        <div class=\"right\">\n" +
"            \n" +
"        </div>\n" +
"    </div>\n");

out.println(" <div  style=\"background-image:url('k.jpg');background-
size:100% 100%;\" >\n" +
"          <br/>");
out.println("<CENTER><img src='company-registration.jpg'>");
out.println("<BR><BR><BR>");
out.println("<H1> Thank you for Registration. <BR>");
out.println(" your Registration no. is " + cn +" </H1>");
out.println("<Br><Br> <Form method = get Action = 'index.html'>");
out.println("<Br><Br><Input    Type='Submit'    Value='Goto    Login
Screen'></CENTER>");
out.println("</Form>");
out.println("<marquee><font   size   =6   color   =   brown   ><i>Thank
You</i></font></marquee></div>");

out.println("  <div id=\"footer\">\n" +
"Copyright ©Mgnrega\n" +
"</div>  ");

out.println("</BODY>");
out.println("</HTML>");
out.flush();
}
}
```

Servlet Code For Customer Registration:

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Register extends HttpServlet
{
Connection con;
PreparedStatement ps1,ps2;
Statement stmt;
```

```
ResultSet rs;
int cm;
public void init() throws ServletException
{
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection("jdbc:odbc:ab","","");
ps1   =   con.prepareStatement(   "Insert   into   Needer   values
(?,?,?,?,?,?,?,?)");
ps2 = con.prepareStatement( "Insert into Login values (?,?,?,?,?)");

stmt = con.createStatement();

rs = stmt.executeQuery ("Select  Max(Userid) + 1  from Needer");
if(rs.next( ))
{
cm = rs.getInt(1);
System.out.println(cm);
if(cm == 0)
cm=1;
}
}catch(Exception e)
{
System.out.println("Error");
}
}

public void doPost(HttpServletRequest request,
HttpServletResponse response)
throws IOException,ServletException
{
response.setContentType("text/HTML");
PrintWriter out = response.getWriter();

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

String add=request.getParameter("add");
String city=request.getParameter("city");
String ph=request.getParameter("ph");

String nm=request.getParameter("nm");
String pass=request.getParameter("pass");
String fav=request.getParameter("pnm");
System.out.println(fav);
try{
ps1.setInt(1, cm);
ps1.setString(2, cnm);
ps1.setString(3, add);
ps1.setString(4, city);
ps1.setString(5, ph);
```

```
ps1.setString(6, nm);
ps1.setString(7, pass);
ps1.setString(8, fav);
ps1.executeUpdate();

ps2.setInt(1, cm);
ps2.setString(2, nm);
ps2.setString(3, pass);
ps2.setString(4, fav);
ps2.setString(5, "Needer");


ps2.executeUpdate();

}catch(Exception e)
{
out.println("Unable to store information " + e);

return;
}
out.println("<HTML>");
out.println("<HEAD>");
out.println("     <style>\n" +
"#header {\n" +
"    background-color:#0073e6;\n" +
"    color:white;\n" +
"    padding:15px;\n" +
"    height:50px;\n" +
"}    \n" +
" .right {\n" +
"    position: absolute;\n" +
"    right: 0px;\n" +
"    width: 700px;\n" +
"    padding:23px;\n" +
"   text-align:center;\n" +
"}\n" +
".left {\n" +
"    float:left;\n" +
"    position: absolute;\n" +
"    left: 0px;\n" +
"    width: 700px;\n" +
"    }\n" +
"    \n" +
"   #footer {\n" +
"    background-color:black;\n" +
"    color:white;\n" +
"    clear:both;\n" +
"    text-align:center;\n" +
"   padding:5px;                                    \n" +
"}\n" +
```

```
"        input[type=submit] {\n" +
"     width: 12%;\n" +
"    background-color: #00cccc;\n" +
"    color: white;\n" +
"    padding: 4px;\n" +
"    border: none;\n" +
"    border-radius: 4px;\n" +
"    cursor: pointer;\n" +
"}\n" +
"\n" +
"input[type=submit]:hover {\n" +
"    background-color: #009999;\n" +
"}</style>\n" +
"    ");

out.println("<TITLE>Successful</TITLE>");
out.println("</HEAD>");
out.println("<Body Bgcolor = \"#C0C0C0\" text=\"#ffffcc\">");

out.println("  <div id=\"header\">\n" +
"      <div  class=\"left\">        <a
href=\"index.html\"><img      src=\"mgnrega.gif\"      width=\"600\"
height=\"50\" alt=\"c6\"/></a></div>\n" +
"        <div class=\"right\">\n" +
"            \n" +
"        </div>\n" +
"    </div>\n");

out.println("  <div  style=\"background-image:url('k.jpg');background-
size:100% 100%;\" >\n" +
"          <br/>");
out.println("<CENTER><img src='company-registration.jpg'>");
out.println("<BR><BR><BR>");
out.println("<H1> Thank you for Registration. <BR>");
out.println(" your Registration no. is " + cm +" </H1>");
out.println("<Br><Br> <Form method = get Action = 'index.html'>");
out.println("<Br><Br><Input    Type='Submit'    Value='Goto    Login
Screen'></CENTER>");
out.println("</Form>");
out.println("<marquee><font   size   =6   color   =   brown   ><i>Thank
You</i></font></marquee></div>");

out.println("  <div id=\"footer\">\n" +
"Copyright ©Mgnrega\n" +
"</div>  ");

out.println("</BODY>");
out.println("</HTML>");
out.flush();
}
```

```
}
```

Servlet Code for Labour Window:

```
<%@ page language="java" contentType="text/html"
import="java.util.*"
import="java.sql.*"%>

<HEAD>
<style>


.dropbtn {
background-color: #0059b3;
color: white;
padding: 7px;
font-size: 15px;
border: none;
cursor: pointer;
}

.dropdown {
position: relative;
display: inline-block;
}

.dropdown-content {
display: none;
position: absolute;
background-color: #f9f9f9;
min-width: 160px;
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
}

.dropdown-content a {
color: black;
padding: 12px 16px;
text-decoration: none;
display: block;
}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {
display: block;
}

.dropdown:hover .dropbtn {
background-color: #0080ff;
```

```
}


#myTable {
border-collapse: collapse;

border: 5px solid #ddd;
font-size: 18px;
}

#myTable th, #myTable td {
text-align: left;
padding: 12px;
}

#myTable tr {
border-bottom: 3px solid #ddd;
}

#myTable tr.header, #myTable tr:hover {
background-color: #d98cb3;
}

#header {
background-color:#0073e6;
color:white;
padding:15px;
height:50px;
}
.right {
position: absolute;
right: 0px;
width: 700px;
padding:23px;
text-align:center;
}
.left {
float:left;
position: absolute;
left: 0px;
width: 700px;
}

#footer {
background-color:black;
color:white;
clear:both;
text-align:center;
padding:5px;
}
```

```
</style>

<TITLE>Labour Window</TITLE>
</HEAD>
<Body Bgcolor = "#C0C0C0" text="#ffffcc">


<%

String cc=(String) session.getAttribute("code");
String n=(String) session.getAttribute("nm");%>

<div id="header">
<div class="left">          <img
src="mgnrega.gif" width="600" height="50" alt="c6"/></div>
<div class="right">

         
        
        
        
        
        
        
              

<div class="dropdown">
<button class="dropbtn">UserId: <%= cc %></BUTTON>
<div class="dropdown-content">
<a href="Status.jsp">Home</a>
</div>
</div>         

<div class="dropdown">
<button class="dropbtn">Name: <%= n %></BUTTON>
<div class="dropdown-content">
<a href="LabourProfile.jsp">Profile</a>
<a href="ChangePass.jsp">Change Pass</a>
<a href="Logout.jsp">Logout</a>
</div>
</div>         

</div>
</div>

<CENTER>
<div style="background-image:url('slide_04.jpg');background-size:100%
100%;" >
<br>
<img src="13.jpg">
```

```
<%
Connection con;
ResultSet rs =null;
Statement stmt;
Vector items= new Vector( );;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection("jdbc:odbc:ab","","");
stmt = con.createStatement( );
rs = stmt.executeQuery("SELECT count(labourid) FROM work where
labourid=\'"+cc+"\'");
int a=0;
while(rs.next())
{
a=rs.getInt(1);
}
if(a==0)
{
%>
<center><h2>You Don't Have any bookings till now</h2></CENTER>
<%
}
else
{
%>

<H1>Your Bookings</H1>
<H2>Your appointment details are listed below:</H2>
<BR><BR>
<table id="myTable">
<TR>
<TH BGCOLOR="orange">User Code</TH>
<TH BGCOLOR="orange">Customer Name</TH>
<TH BGCOLOR="orange">Location</TH>
<TH BGCOLOR="orange">Date of Joining</TH>
<TH BGCOLOR="orange">No. of Days</TH>
<TH BGCOLOR="orange">Wage</TH>
<TH BGCOLOR="orange">Mobile No.</TH>

</TR>
<%
rs = stmt.executeQuery("SELECT * FROM work where
labourid=\'"+cc+"\'");
while(rs.next( ))
{
%>
<TR>
<TD><%= rs.getString("labourid") %></TD>
<TD><%= rs.getString("custname") %></TD>
<TD><%= rs.getString("location") %></TD>
```

```
<TD><%= rs.getString("doj") %></TD>
<TD><%= rs.getString("nod") %></TD>
<TD><%= rs.getString("wage")%></TD>
<TD><%= rs.getString("mobno") %></TD>
</TR>

<%
}}
%>
</TABLE>
<br>
</div>
<div id="footer">
Copyright©Mahatama Gandhi National Rural Employment Guarantee Act
</div>
</CENTER>
</BODY>
</HTML>
```

CHAPTER – 7

OUTPUTS

- **HOME SCREEN FRAME :-**

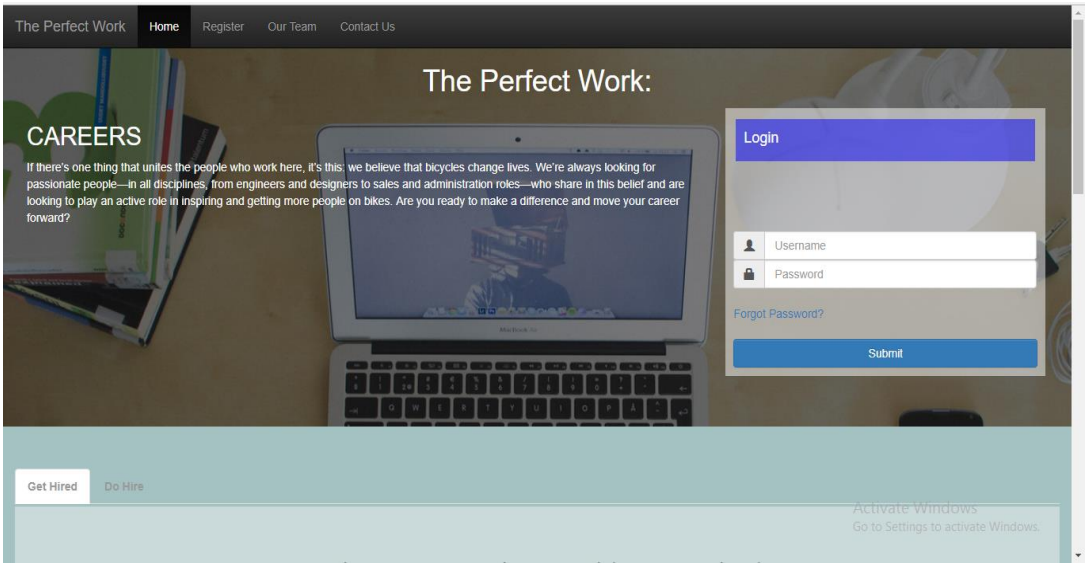

Figure 7.1: Home Screen

- **REGISTER (NAME) :-**

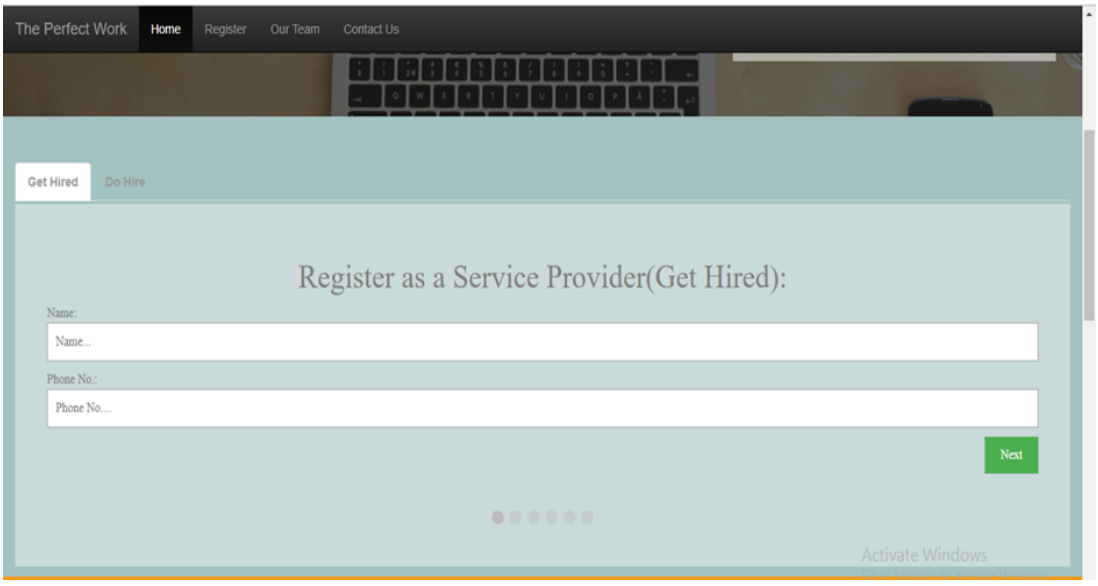

Figure 7.2: Register Frame

- **SELECT WORK :-**

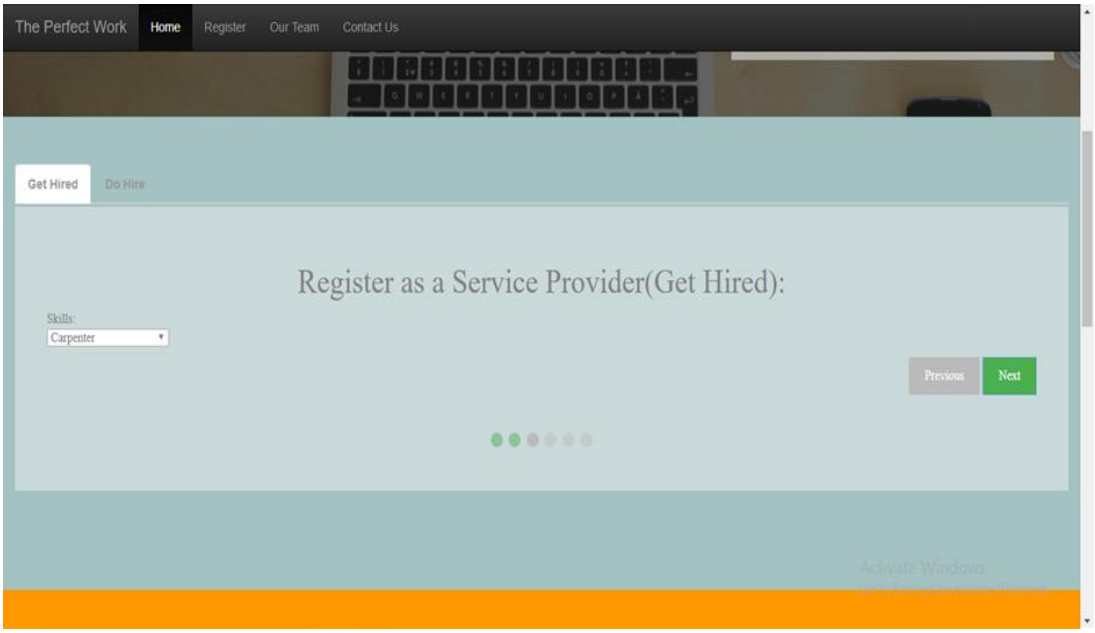

Figure 7.3: Select work frame

- **PAYMENT DETAILS**

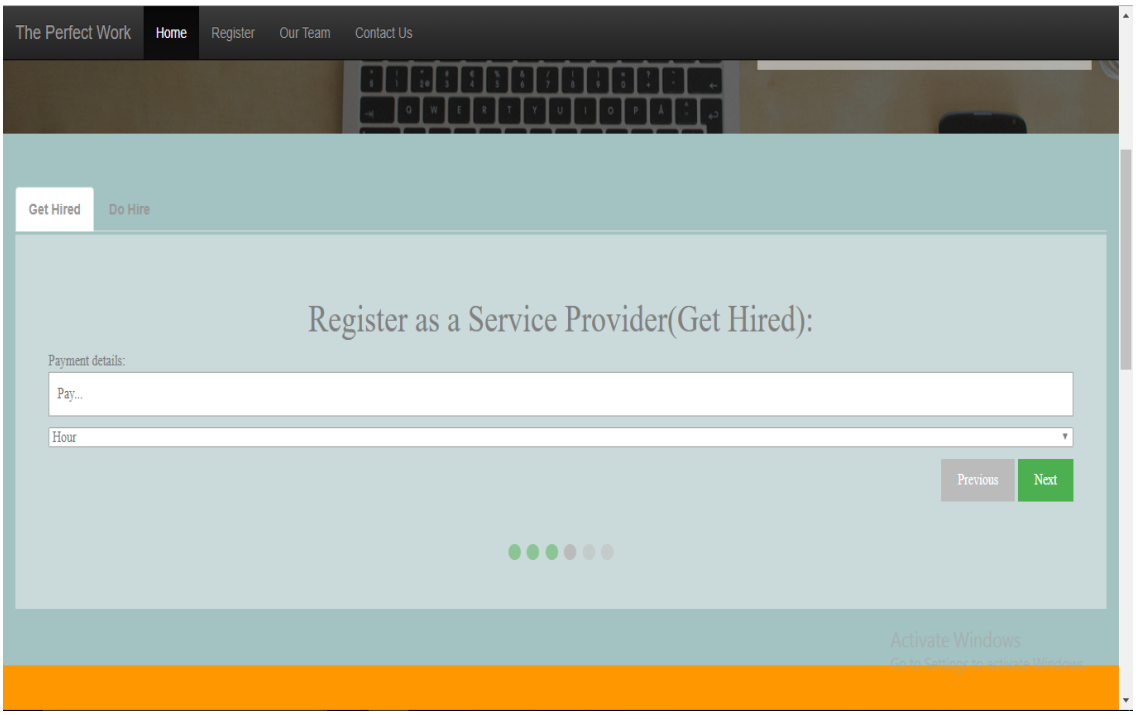

Figure 7.4: Payment Details

- **REGISTER(USERNAME & PASSWORD)**

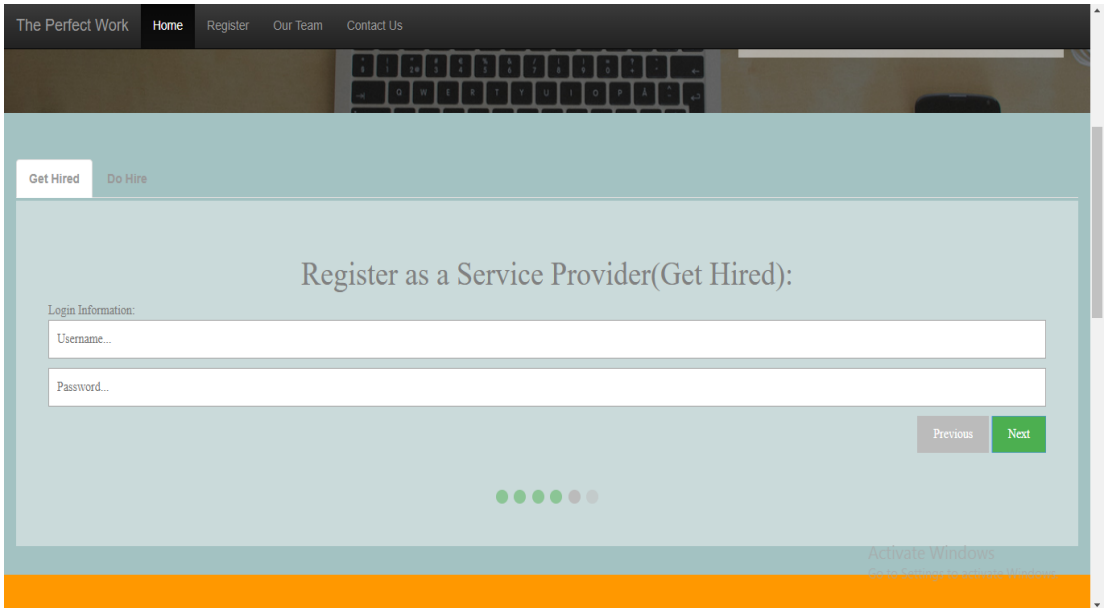

Figure 7.5:Username and password registration

- **SECURITY QUESTION**

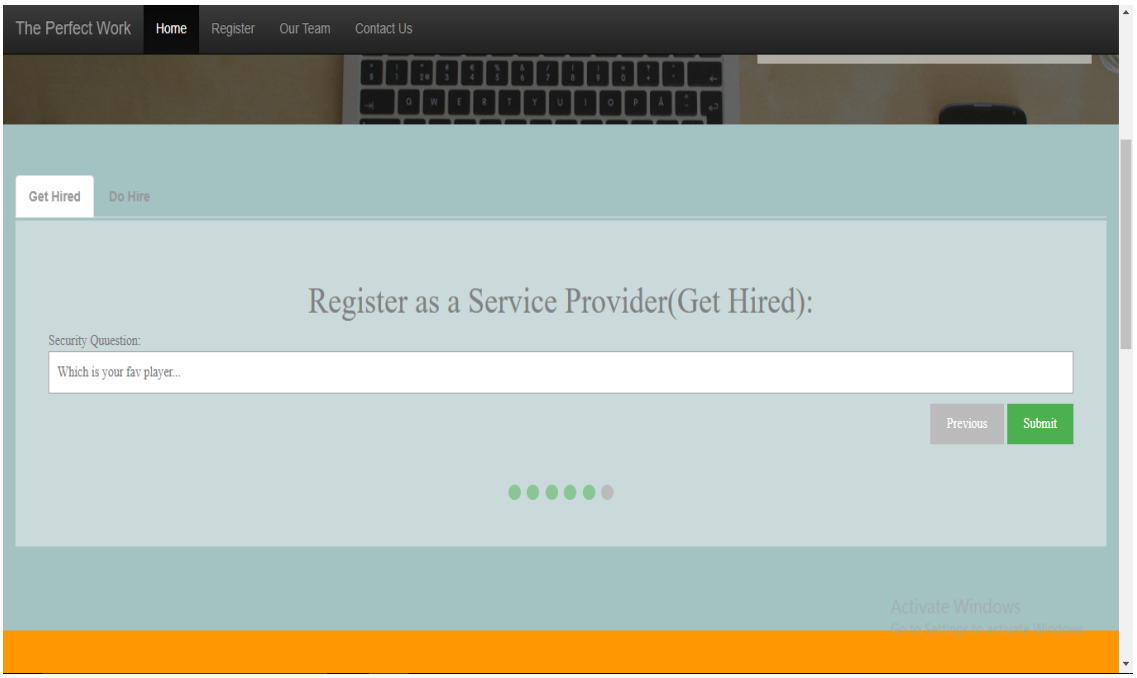

Figure 7.6:Security Question

- **REGISTRATION COMPLETE**

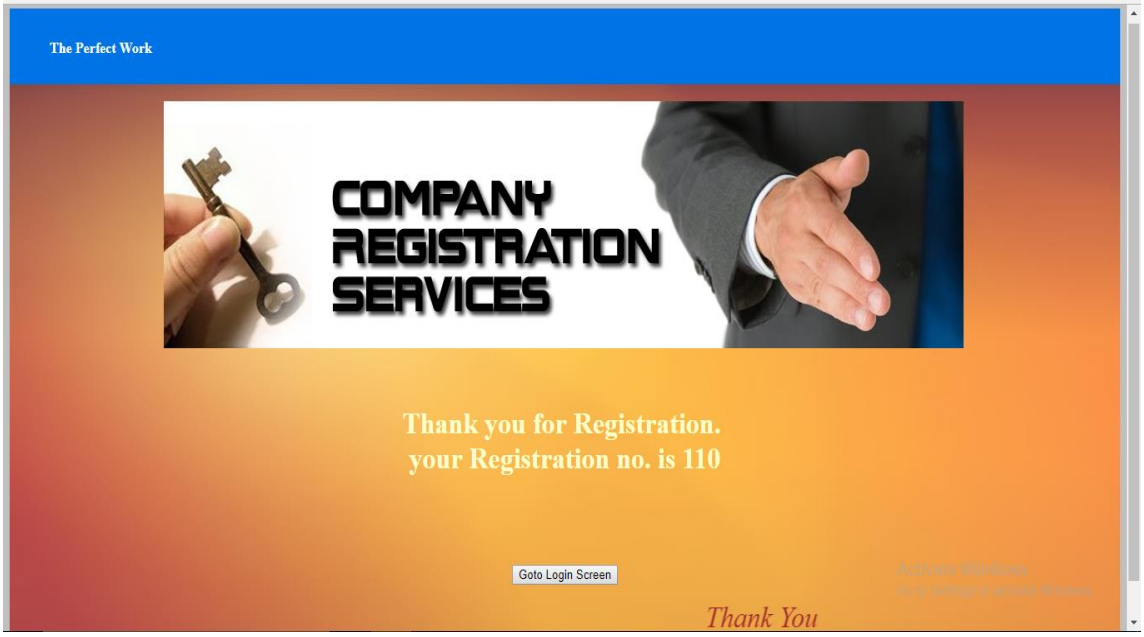

Figure 7.7:Registration complete

- **USER WINDOW**

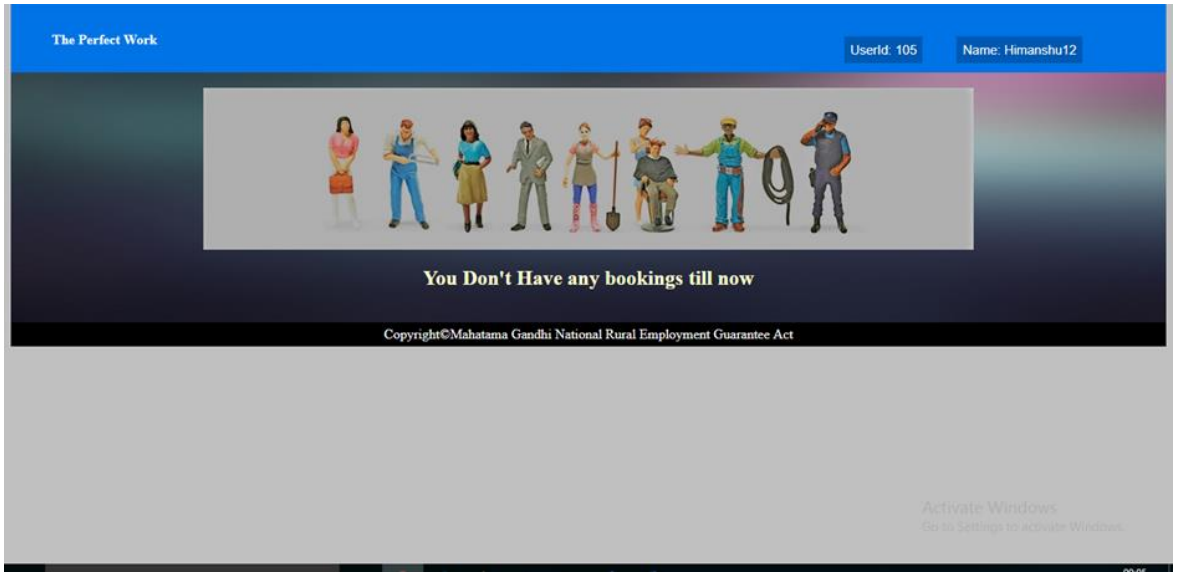

Figure 7.8:User window

- **LOGOUT FRAME**

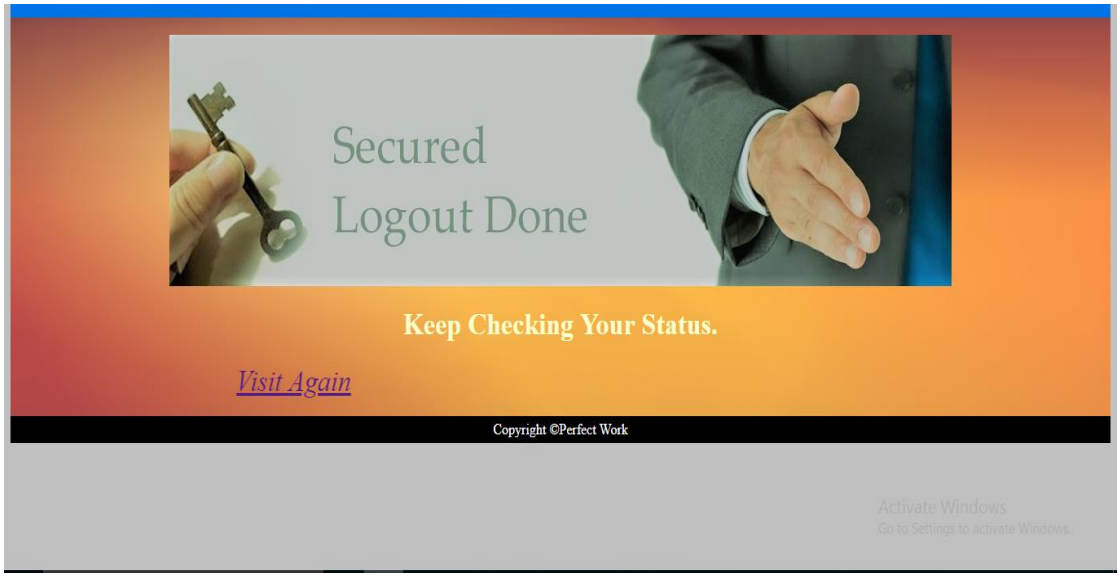

Figure 7.9:Logout frame

- **MEET OUR TEAM**

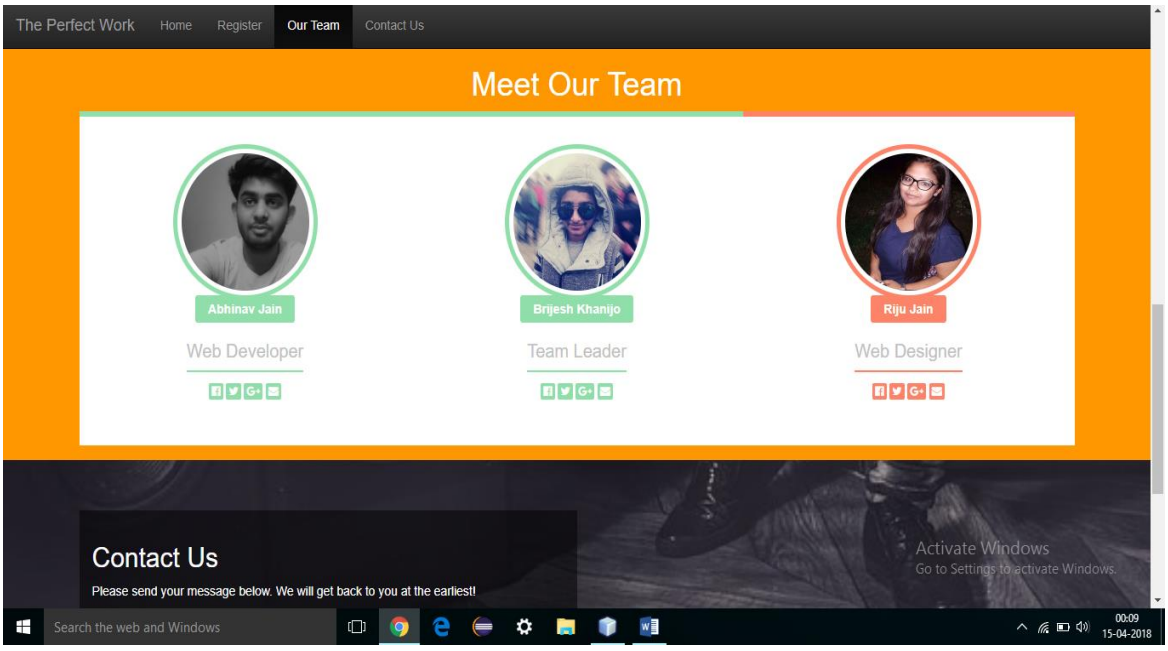

Figure 7.10:Meet our team

- **CONTACT US**

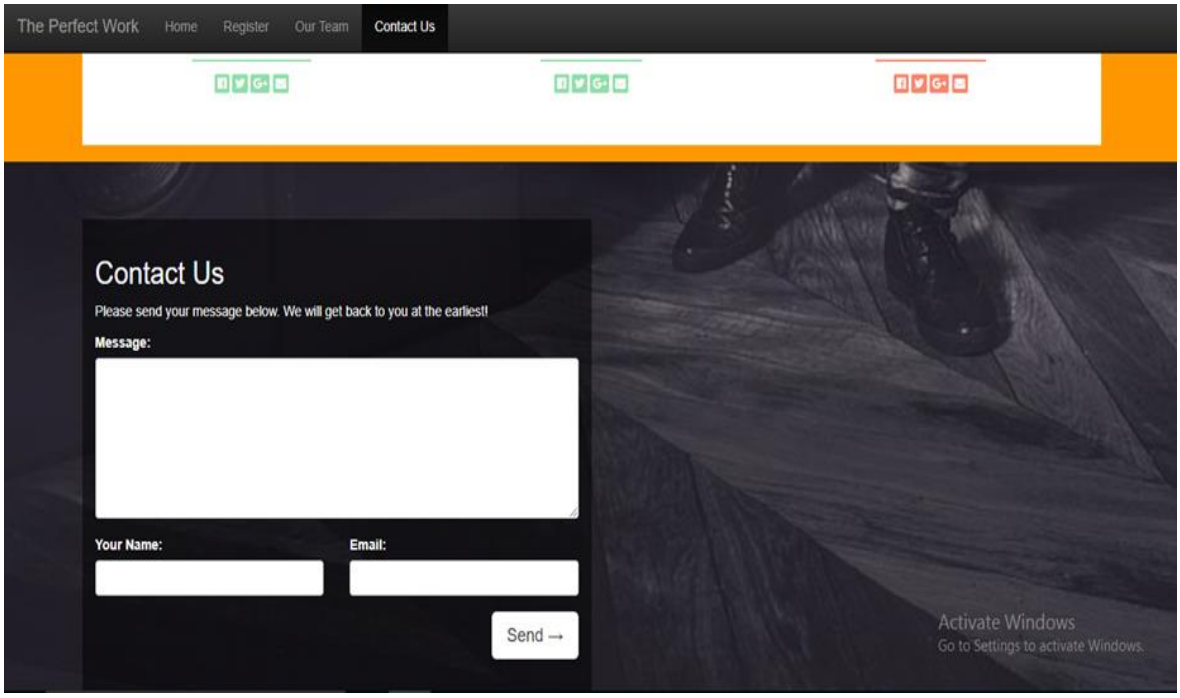

Figure 7.11:Contact us

CHAPTER – 8

CONCLUSION

Specialist knowledge is a critical asset for individuals, businesses and societies. The importance of such knowledge is even more pronounced in a dynamic, globalized world. Building knowledge on basis of basic skills early on, by broadening and improving the quality of early childhood, is essential. Matching specialist knowledge and jobs requirements has become a high-priority policy concern.

To conclude, with the economy gaining pace again, the time is ripe to initiate ambitious and visionary skilling and employment programs, which would enable the economy to perform to its potential.

REFERENCES

- https://martonrecruitment.com/news/importance-of-specialisation
- https://digitalcommons.unl.edu/cgi/viewcontent.cgi?referer=https://www.google.co.in/&httpsredir=1&article=3470&context=libphilprac
- https://www.google.co.in/search?q=importance+of+specialised+workers&oq=importance+of+specialised+workers&aqs=chrome..69i57j0.27775j1j1&sourceid=chrome&ie=UTF-8