**MAJOR PROJECT ON**

**Work [Or] Hire**

Bachelor of Computer Application

**By**

**Debarghaya Bhaumik**

Under the guidance of

Saubhik Bandopadhyay

Certificate of Recommendation

This is to certify that I , Debarghaya Bhaumik has completed his project work titled “ **Work [OR] Hire** ”, under direct supervision and guidance of my mentor Saubhik Bandopadhyay .



### Goal of the project

The main objective of Work Or Hire project is job provider(Hirer) can post jobs, job seeker(Bidder) can choose jobs as their skill, job search. Our project aims at making the tasks of job provider(Hirer) and job seeker(Bidder) easy which includes registration as you want to work, show all project, update profile, register as you want to hire, post jobs and admin can update his own profile, set online quiz. Moreover selecting a employee according to their feedback or rating and selecting a project of your choice.

The entire system will help job seeker to find job and job provider(Hirer) to choose efficient job-seeker(Bidder) according to their project need at minimal cost. In this system there is no third party interference.

### Organization of the Project

* **Table Information**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Purpose** | **Page No.** |
| TABLE 1 | Value of constant of organic product | 4 |
| TABLE 2 | Value of 7 effort adjustment factor | 5 |
| TABLE 3 | Phases of time analysis | 6 |
| TABLE 4 | General description of input and output | 23-24 |
| TABLE 5 | User interfaces description | 25 |
| TABLE 6 | Software used at developer site | 27 |
| TABLE 7 | Software requirement at client site | 27 |
| TABLE 8 | Hardware used at developer site | 27 |
| TABLE 9 | Hardware requirement at developer site | 27 |
| TABLE 10 | Result set analysis | 49 |
| TABLE 11 | List of validation check in our project | 97 |

* **Image Information**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Purpose** | **Page No** |
| Image 1 | Project schedule | 7 |
| Image 2 | Gantt Chart of the project | 7 |
| Image 3 | Team structure | 8 |
| Image 4 | HTML architecture | 13 |
| Image 5 | My SQL database | 15 |
| Image 6 | My SQL architecture | 16 |
| Image 7 | Users table | 28 |
| Image 8 | User table | 28 |
| Image 9 | Quiz category table | 29 |
| Image 10 | Quiz question table | 29 |
| Image 11 | Quiz option table | 29 |
| Image 12 | Project table | 30 |
| Image 13 | Project skills table | 30 |
| Image 14 | Rating details table | 31 |
| Image 15 | Bidding details table | 31 |
| Image 16 | Skills table | 32 |
| Image 17 | Exam result table | 32 |
| Image 18 | Exam result details table | 32 |
| Image 19 | Payment table | 33 |
| Image 20 | Entity relationship diagram | 34 |
| Image 21 | Home | 35 |
| Image 22 | Registration page | 35 |
| Image 23 | Login page | 36 |
| Image 24 | Edit profile page | 36 |
| Image 25 | Use case diagram | 37 |
| Image 26 | 0 Level DFD | 38 |
| Image 27 | 1 Level DFD for Admin module | 39 |
| Image 28 | 1 Level DFD for Hirer module | 40 |
| Image 29 | 1 Level DFD for Bidder module | 41 |
| Image 30 | Flow chart diagram | 42 |
| Image 31 | Home | 92 |
| Image 32 | Registration | 92 |
| Image 33 | Login | 93 |
| Image 34 | Edit profile | 93 |
| Image 35 | Quiz category | 94 |
| Image 36 | Add quiz category | 94 |
| Image 37 | Quiz category list | 95 |
| Image 38 | Post project | 95 |
| Image 39 | My leads project | 96 |

### Acknowledgement

We would like to express our profound gratitude to our trainer Mr. ParthoMallick, RajdeepGhatak, SubharanjanBasu and the entire faculty of Bachelor of Computer Application for their continuous guidance, valuable suggestions and constant moral support, encouragement and advices throughout the period of the progress in project. We are also grateful to them for providing us all the help and necessary resources.

We also convey our teachers of respecting department for providing us all needed support and laboratory, as and when required. Lastly, we would like to thank all the people who had been a constant source of motivation for us.

**Contents**

|  |  |  |
| --- | --- | --- |
| **Major Project On**  ***Work [OR] Hire*** | | |
| 1 | Introduction | 1 |
| 2 | Scope of the Project | 2-3 |
| 3 | Concepts and Problem Analysis   * 1. Cost Analysis[Use COCOMO model]   2. Time Analysis   3. Team Structure   4. Software Configuration Manager   5. Qualty Assurance Plan   6. Risk Management | 4-12 |
| 4 | Theoretical Background | 13-20 |
| 5 | Software Requirement Specification | 21-28 |
| 6 | Design/Solution/Methodology  6.1Data Design 6.2Interface Design 6.3Architectural Design  6.4Procedural Design | 29-42 |
| 7 | Coding Standard Followed and Assumptions | 43-48 |
| 8 | Result Set Analysis | 49 |
| 9 | Testing | 50-52 |
| 10 | Future Scope of the Project | 53-54 |
| 11 | Conclusion | 55 |
| 12 | Reference And Bibliography | 56 |
| 13 | Appendix   1. Appendix 1 2. Appendix 2 3. Appendix 3 | 57-96 |

# Introduction

**1**

### Introduction

Work Or Hire is a project which aims to provide job to job- seeker(Bidder) as their skill wise. This Work Or Hire used by job- provider(Hirer) to post their projects and choose job-seeker(Bidder) according to their project needs at minimal cost, they can give feedback to job-seeker(Bidder). Job-seeker(Bidder) can choose job as their requirements. Any person who wants to work they can just register themselves and search for job as their skills.

Overall this project of ours is being developed to help job seeker(Bidder) to search jobs and job provider(Hirer) to complete their project at minimal cost.

# Scope of the Project

**2**

### Scope of the Project

This Work Or Hire helps job-seeker(Bidder) and job-provider(Hirer) to find jobs and complete their projects. In this system there is no geographical bound the job-provider(Hirer) can post their projects and choose job-seeker(Bidder) as their project need. It also has a facility that job-seeker(Bidder) can login to their profile and see their completed projects and admin can set the quiz for the job-seeker.

**Benefits:**

*Admin:*

1. set-up for online quiz(add category, question or set answer)
2. share website link to social Network Like Facebook for increasing number of Job- seeker(Bidder) as well as Job-provider (Hirer).
3. Update profile.

*Job-provider(Bidder):*

1. Register as they want to hire.
2. Post a new job.
3. Show all posted, completed and expired jobs.
4. Show all worker details who want to response of a particular job.
5. According to worker feedback, rating or percentage got through online quiz taken by admin, hire a worker.
6. Pay and give feedback or rating to the hired person.
7. Can show all payment details.
8. Update profile.

*Job-seeker(Hirer):*

1. Register as they want to work.
2. Show all jobs posted by provider.
3. Show all completed projects.
4. Can also show their payment details.
5. Give online quiz.

**Applications :**

The different areas where we can use this application:

* It can be used in all type of companies and modifications can be easily done according to requirements.

# 4. Concepts and problem Analysis

**3**

### Concepts and Problem Analysis

* 1. **Cost Analysis (using COCOMO model)**

COCOMO (Constructive Cost Estimation Model) was proposed by bohem, 1981. Bohem postulated that any software development project can be classified in any one of the following categories based on the development complexity: organic, semidetached and embaded. In order to classify a product into the identified categories, Bohem requires us to consider not only the characteristics of the identified categories, Bohem requires us to consider not only the characteristics of the product but also those of the development team and the development environment.

So in case of our project (Work Or Hire), it is actually a development project in Semidetached type which deals with developing a well – understood website application, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

The COCOMO model gives an approximate estimate of the project parameters, in our project, we have calculated Intermediate COCOMO, where this COCOMO estimation model is given by the expression –

* + - **Effort1=a1\*(KLOC)^a2PM**
    - **Effort2=EAF \*Effort1PM**
    - **Tdev=b1\* (Effort)^b2 Months**

Where

* KLOC is estimated size of the software product expressed in Kilo Lines of Code.
* A1, a2, b1, b2 are constants for each category of software products.

•Tdev is the estimated time to develop the software, expressed in months.

* Effort is the total effort required to develop the software product, expressed in person months(PMs).

Here we first calculate Effort1 which is calculated depend on value a1 and number of estimated Lines of Code, then we have calculated Effort2 depend on effort adjustment Factor(EAF) and previous value of Effort i.e. Effort1. Finally Tdev is calculated. So, the effort and duration values computed by COCOMO are the Values for doing the work in shortest time without unduly increasing manpower cost.

**Calculations**

Here the values of a1, a2, b1, b2 depends on the type of software products of project. We have just told that our software project is organic type, so according to that the values of a1, a2, b1, b2 are following

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | A2 | B1 | B2 |
| 2.4 | 1.05 | 2.5 | 0.38 |

Table 1

In our project estimated lines of code is 3500(approx). so the estimated size of the software project is 3.5 KLOC(1LOC – 1000 KLOC).

We know that,Effort1= a1\*(KLOC)^a2 PM

=2.4\*(3.5)^1.05 PM

=8.94298829 PM

Again we have considered 7 Effort Adjustment Factors(EAF) and taken high ratings of these 7 factors which are following.

|  |  |
| --- | --- |
| Analyst capability | 0.75 |
| Application Experience | 0.91 |
| Use of Software Tool | 0.91 |
| Programming Language Experience | 0.95 |
| Required turnaround Time | 1.01 |
| Product Complexity | 1.05 |
| Required Software Reliability | 1.05 |

Table 2 We know that,Effort2= EAF \*Effort1 PM

= (0.75\*0\*91\*0.91\*0.95\*1.01\*1.05\*1.05)\*8.94298829 PM

= 5.875574178 PM

Finally

Tdev = b1\*(Effort2)^b2 Months

= 2.5\*(5.875)^0.38 Months

= 4.899626779 Months

###### Time Analysis

The project task scheduling is taken up in order to decide which task would be done when inside the schedule. We have already calculated the development time of the project that is 147 days approximately. These days are distributed in the following way.

|  |  |
| --- | --- |
| **Phases** | **Duration(days)** |
| Analysis | 5(approx) |
| Design | 10(approx) |
| Coding | 30(approx) |
| Testing and Implementation | 20(approx) |
| Documentation | 15(approx) |

Table 3

###### Project Schedule

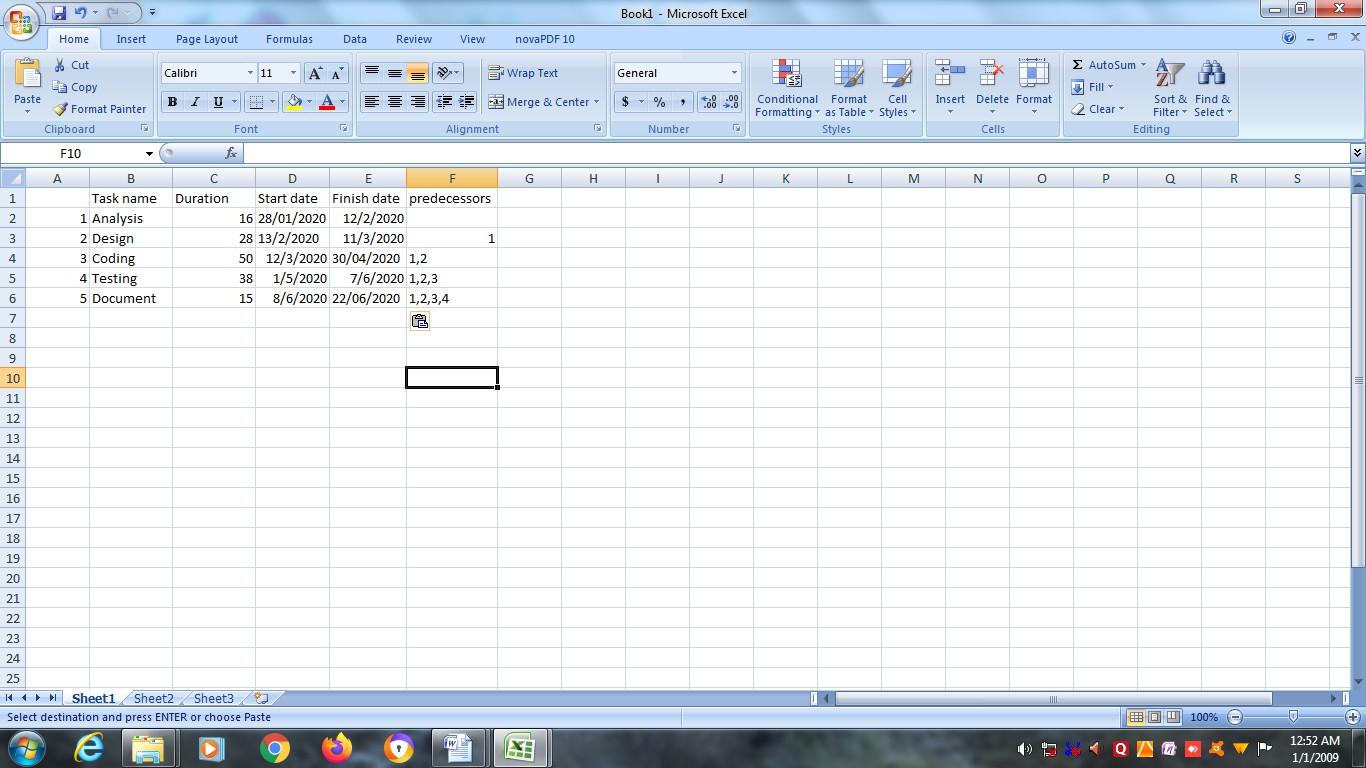
****

IMAGE 1

###### Gantt chart of the Project Schedule

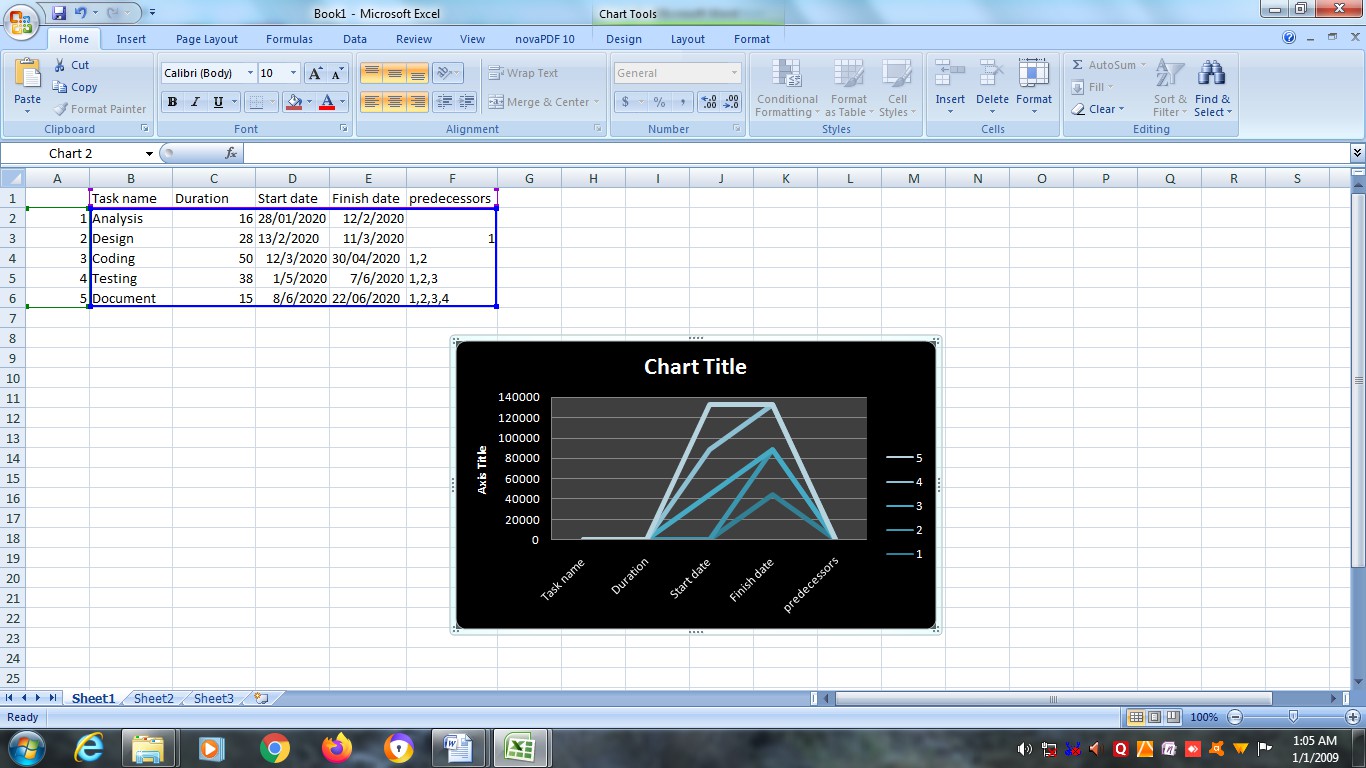
****

IMAGE 2

###### Team structure

The project is a group project. As we are three members in this group Dalia Mridha, Barsha Podder, Brahmanand Pandey, during the course of the project we have continuously interacted with each other.The project is product of our joint efforts and hardwork.Hence our team structure is as follows:

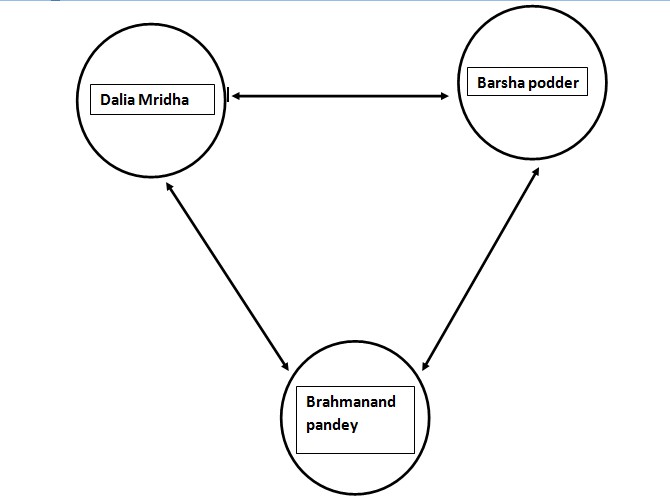


IMAGE 3

###### Software Configuration Management

Change continuously take place in a software project. Change due to evolution of work products. As the project proceeds change due to bugs being found and fixed and change due to requirement modification. Software configuration Management (SCM) are the practices are procedures for administrating the source code. Producing software development builds. Controlling change and managing software configuration. Specifically, SCM ensures the integrity, reliability and reproducibility of developing software products from conception to release. The goal of SCM that are followed during building the project:

* **Configuration Identification:** The team configuration identification or CI refers to the fundamental structural unit of a configuration management system. The objective of this system is to avoid the introduction of errors related to lack of testing as well as incompatibilities with other Cls.
* **Configuration Control:** Implementing a controlled change process. In

the project, our mentor reviewed the changes in our project.

* **Configuration Status Accounting:** Recording and reporting all the necessary information on the status of the development process to the project guide.
* **Configuration Audition:** Ensuring the configuration contain all their

intended parts and are sound with respect to their specifying documents. Including requirements and architectural specification.

* **Build Management:** Managing the process and tools used for builds. One

of the most important steps of a software build is the compilations process.

* **Process Management:** Process Management is the application of

knowledge, skills, tools, techniques and systems to define, control, report and improves the process with the goal to meet the requirements profitably.

* **Environment Management:** Managing the software and hardware that host

the system.

* **Teamwork:** Teamwork is work performed by a team towards a common goal. Teamwork is a joint action by more than one people, in which each

person contribute with different skills and express his or her individual interests and opinions to the unity and efficiency in order to achieve the goal.

* **Defect Tracking:** Making sure every defect traceability back to that source.
  1. **Quality Assurance Plan**

The Quality Assurance (QA) plan involves or selecting standards that should be applied to the software developments process or software product. These standards may be embedded in procedures or processes that are applied during development. Tools that embed knowledge of the quality standards may supports processes.

These are two types of standard that may be established as part of the Quality Assurance Plan:

* **Product Standards:** These are standards that apply to the software product being developed. They include document standards such as the structure of the requirements document which should be produced. Documentation standards such as standard comment header for an object class definition and coding standards which define how a programming language should be used.
* **Process Standards:** These are standards that define the processes which should be followed during software development. They may include definitions of specification. Design and validation process and a description of the documents which must be generated in the course of the process.

These is a very close relationship between product and process standards, the product standards apply to the outputs of the software process and, in many cases the process standards include specific process activities that ensure that product standards are followed.

* 1. **Risk Managements**

A risk unfavorable circumstances or event that might hamper the successful or timely completion of a project. Risk implied that there is a possibility that something negative may happen. In the context of software project, negative implies that there is an adverse effect on cost, quality or schedule. Risk managements an attempt to minimize the impact of risk. The risk management activities help to eradicate risk before it can harm the productivity of the software project with only 28% of software projects finishing on time and on budget, risk and management of risk play an important role in software development. In our project, we face some risks which are in following:

* **Knowledge Risk:** We developed our project on LIBRARY MANAGEMENT SYSTEM. The platform we have chosen to develop the project is- PHP,MYSQL server. Now the main fact is that we have knowledge on these but don’t have enough knowledge on these software or platforms which we will be working on. In that case, it is most important to gather knowledge on this software.
* **Resource Risk:** Initially we don’t have these required platforms (PHP, MYSQL) which we need to develop the project. We have taken 2 or 3 weeks to collect all these software and install all of them, even though during the installation we have faced certain regarding these software, but we have finally solved it successfully.
* **Time Risk:** Time is a main factor for any project. In our project we are giving 5 months to complete our project. But the main problems that we have taken half a month to gather all knowledge about the platforms, install them and operate them. Rest of the months we have taken for coding, designing the forms and binding the project.

# 4.Theoretical Background

### Theoretical Background

##### HTML:

###### HTML Architecture:

* + hypertext markup Language (HTML) is a markup language for creating WebPages. WebPages are usually viewed in a web browser. They can include writing, links, pictures, and even sound and video.
  + Cascading style sheets (CSS) and JavaScript can include in HTML code. CSS is used to change how a Webpage looks. JavaScript is used to add features to WebPages and make them more interactive.
  + HTML was made by the World Web Consortium (W3C).

###### Features of HTML:

* + - Text and all places where it is used, i.e. paragraph, heading, quotes, etc.
    - Multimedia: images, audio, and video.
    - The so-called canvas, which allows you to “draw” on the Web

Page using JavaScript code.

* + - Traditionally, HTML is learned along with CSS and JavaScript, in what we call the web development trio. Together, they make the foundation behind the internet as we see it today.
    - There are some “spinoffs” of HTML, which are used in

different circumstance as well:

* + - * **DHTML:** Dynamic Hyper Text Markup Language, same as normal HTML but more flexible.
      * **XML:** Extensible Markup Language, uused for

documentation and creating layout files during Android development.

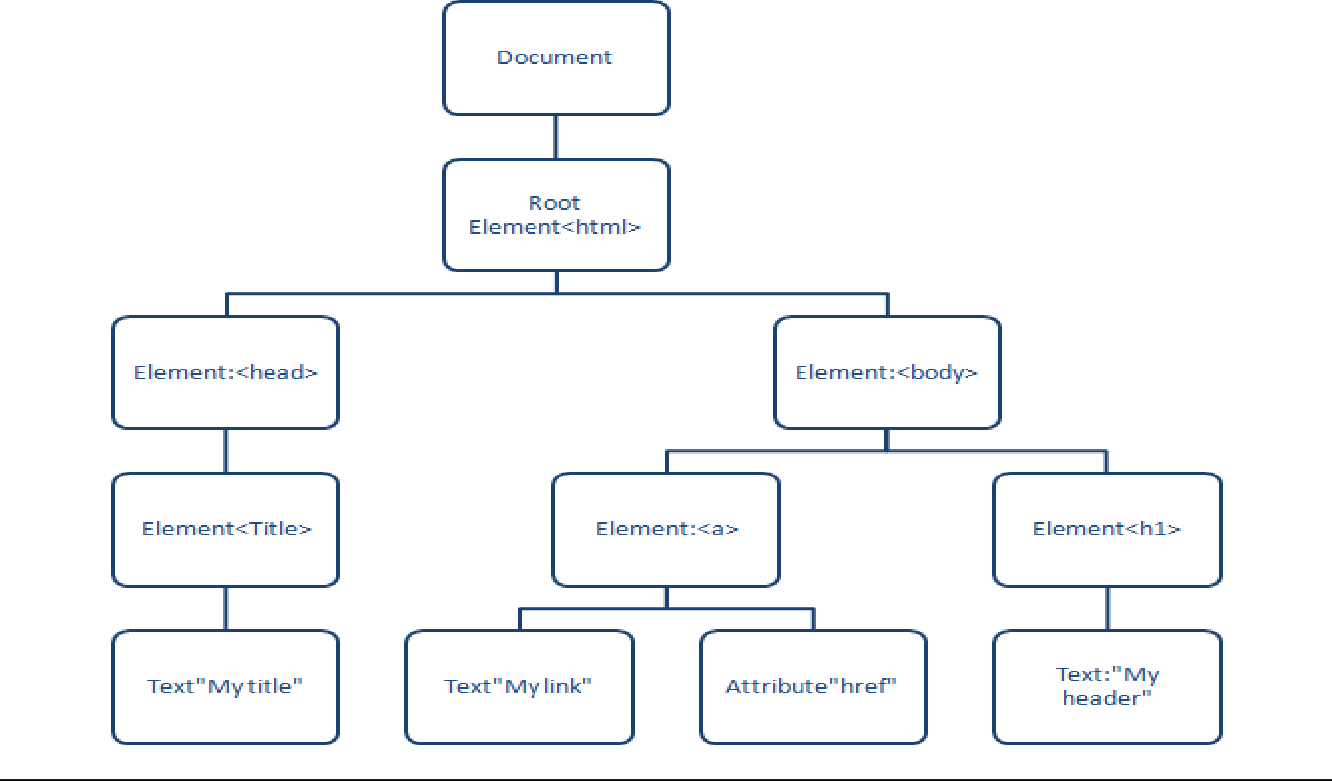


IMAGE 4

##### CSS:

###### CSS Architecture:

* + **Predictable:** predictable CSS means your rules behave as you’d expect. When you add or update a rule, it shouldn’t affect parts of your site that you didn’t intend. On small sites that rarely

change, this isn’t as important, but on large sites with tens or

hundreds of pages, predictable CSS is a must.

* + **Reusable:** CSS rules should be abstract and decoupled enough that you can build new components quickly from existing parts without having to recode patterns and problems you’ve already solved.
  + **Maintainable:** When new components and features need to be added, updated or rearranged on your site, doing so shouldn’t require refactoring existing CSS. Adding component X to the page shouldn’t break component Y by its mere presence.
  + **Scalable:** AS your site grows in size and complexity it usually requires more developments to maintain. Scalable CSS means it can be easily managed by single person or a large engineering team. It also means your site’s CSS architecture is easily approachable without requiring an enormous learning curve. Just because you’re the only developer touching the CSS today doesn’t mean that will always be the case.

###### CSS Features:

* + Selectors
  + Box model
  + Backgrounds and Borders
  + Image values and Replaced Content
  + Text effects
  + 2D/3D Transformations
  + Animation
  + Multiple columns Layouts
  + User Interface

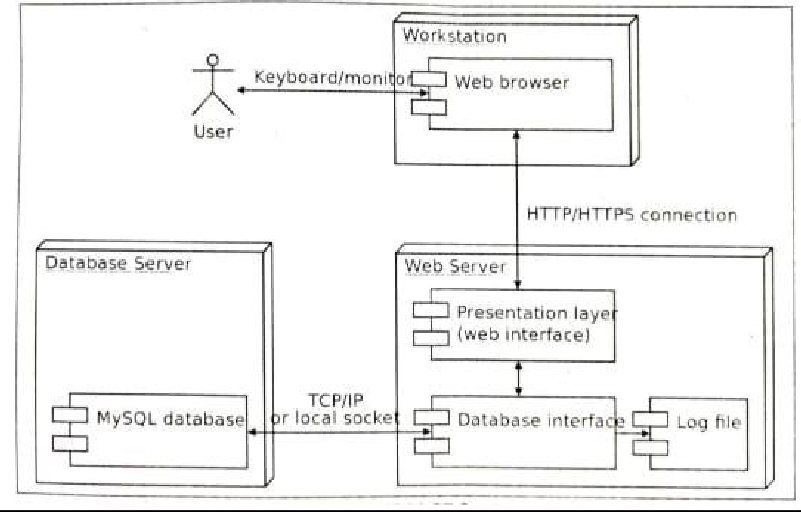


IMAGE 5

##### DATABASE CONNECTIVITY WITH SQL SERVER AND MYSQL

SQL server is an SQL, compliant RDBMS. SQL compliant means it uses the ANSI( American National Standard Institute) version of structured Query Language or ‘SQL’. Structured Query Language is a command that allow us to modify or retrieve information from the database.

Client server means that SQL server is designed to store data in the central location(the server) and deliver it on demand to numerous other locations (the client). SQL server is also a Relational Database management System(RDBMS).

###### Features of SQL Server:

* + Information representation.
  + Unique definitions of rows.
  + Guaranteed access.
  + High level update. Insert and Delete.
  + Retrieving information from the database.
  + Accept query language statement.
  + Enforcing security specifications.
  + Enforcing data integrity specifications.
  + Enforcing transactions consistency.
  + Managing data sharing.
  + Optimizing queries.

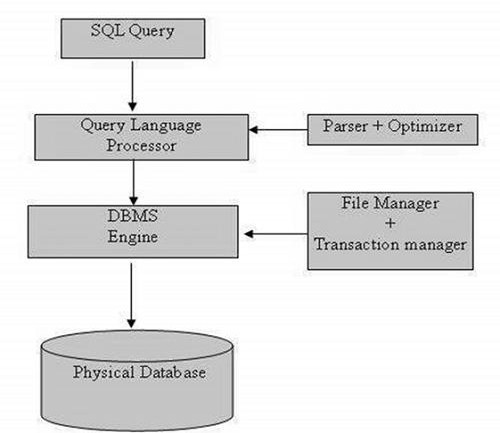


Image 6

###### MySQL architecture:

Many testers know how to write SQL queries and how to fetch data from the tables. But only a few understand the internal architecture and MySQL interacts with storage engines. This blog discusses the MySQL architecture, different layers and journey of SQL statement in MySQL data.

###### MySQL architecture is broken into three layers-

* Application layer.
* Logical layer.
* Physical layer.

###### Features of MySQL:

* + Cross-platform support.
  + Stored procedures, using a procedural language that closely adheres to SQL/PSM.
  + Triggers.
  + Updatable views.
  + Information schema.
  + Performance schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.
  + A set of SQL mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
  + Query caching.
  + Sub-SELECTs.
  + Built-in replication.
  + Full-text indexing and searching Embedded database library.

##### BOOTSTRAP

Build responsive, mobile-first projects on the web with the world’s

most popular front-end componenet library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on JQuery.

* **Features of Bootstrap:**
  + Update print styles and Utility Classes. There is a great improvement in the rendering of printed pages.
  + Additive Border Utilities.
  + Sass map updations.
  + Added documentation.
  + New responsive classes.
  + Endorsement for Bootstrap approach.
* **Features of Java:**
  + Object-oriented.
  + Simple.
  + Secured.
  + Platform independent.

**19**

* + Robust.
  + Portable.
  + Architecture neutral.
  + Dynamic.
  + Interrupted.
  + High performance.
  + Multithreaded.
  + Distributed.
* **Features of server and client side programming:**
  + Querying the database.
  + Operations over databases.
  + Access/Write other servers.
  + Interact with other servers.
  + Structure web applications.
  + Process uses input. For example if user input is a text in search box, run a search algorithm on data stored on server and send the results.

# 5. Software Requirement Specification

### Software Requirement Specification

###### Introduction

* 1. Purpose

In general, purpose of the system is to describe its requirements according to its intended users. The system should be designed in such a way that non-technical persons can also use the system i.e. it must have a high user friendly environment.

* 1. Scope

This project is aimed at developing the Work Or Hire which wouldn’t have any geographical bound. This is an internet based application that can be accessed throughout any company or by job-provider(Hirer). In this system the job- provider(Hirer) can post jobs. Job-seeker(Bidder) can search for jobs as their skills.

References

The references of the following websites and

books have been taken during development of the project and documentation.

###### Software engineering book by Rajiv Mall.

* + - [**www.stackoverflow.com**](http://www.stackoverflow.com/)
    - [**www.w3schools.com**](http://www.w3schools.com/)
  1. Developer’s Responsibility Overview

During the development of the system and after the development it is the responsibility of the user that the system runs properly on the user that the system runs properly on the user machine and it does not undergoes through any exception.

So to handle any errors there are exception handlers that are used to catch the errors occurred during run time and thus it is the developer’s duty to handle the exception properly. In this system all the exceptions are handled properly by the developer and thus the chance of error is very little. Also to provide an easy UI for the user is an important issue that must be implemented by the desktop designers. Our system is designed to provide an easy and user-friendly interface so that no extra learning is required and also some instruction.

###### General Description

Describe the general factors that product and its requirements. This section does not state specific requirements. Instead it provides a background for these, which are defined in late sections, and makes them easier to understand.

###### Product Function Overview

This website provides a computerized version of online job portal system which will benefit the job-seekers as well as the job-providers. It makes entire process online where jobseekers can search for vacancies. It also has a facility for job-seekers to find different type of jobs they want and job-provider can login and can see their posted projects.

**FUNCTIONALITIES:**

###### Admin:

* Login in his profile.
* Can set quiz category.
* Can set quiz list.
* Can set quiz questions.
* Can update own profile.
* Can logout.

###### Job-seekers(Bidder):

* Can register as job-seeker(Bidder).
* Can login in there profile.
* Search for jobs.
* Can show their completed projects.
* Can quiz the online exam.
* Show their payments.
* Can update own profie.
* Logout.

###### Job-providers(Hirer):

* Can register as job-provider(Hirer).
* Can login in there profile.
* Can post projects.
* Can show their projects.
* Can show their all completed projects and expired projects.
* Can show there payment details.
* Can update own profile.
* Logout.

###### User Characteristics

* + - Primary users of the system will be admin and job- seeker(Bidder).

###### Functional Requirements

* 1. **general Description of Inputs and Outputs**

As the system, is an online project and also interacts with the user to function properly it takes inputs from the user and provides output to them.

Following table shows the inputs and outputs that are generated.

|  |  |  |
| --- | --- | --- |
| **Screen Name** | **Input** | **Output** |
| Login(Admin) | Email-id, password | If correct then redirect to the profile page and give access to all relevant record regarding set quiz category, set quiz question etc. else invalid login  message. |
| Login(Hirer) | Email-id, password | If correct then redirect to the profile page and give access to all relevant record regarding post projects, show all completed projects and expired projects etc. else invalid login  message. |
| Login(Bidder) | Email-id, password | If correct then redirect to the profile page and give access to all relevant record regarding search jobs, show all completed projects and can take quiz  etc. else invalid login message. |
| Registration(Hirer, Bidder) | Name, Email, Ph no., address, DOB,  password, Profile  picture, Type | If all data are enter correctly then the people will register successfully or else enter field message will show. |
| Post project | Project name, skill, starting date, project type, minimum rate, maximum  rate | If all data are entered correctly then project will be posted or error message will show. |

|  |  |  |
| --- | --- | --- |
| Add quiz category | Quiz  category |  |
| Add quiz questions | Quiz category, question, option, set  answer |  |

**Table 4**

###### Functional Requirements

This Work Or Hire is designed to full-fill some requirements that were not fulfilled by the other job portal system. These requirements are needed to be full-filled to make the system function correctly.

* In this online job portal system job-provider(Hirer) can choose worker as their own choice at minimal cost.
* In this online job portal system job-seeker(Bidder) can

choose project as their own choice as their skill wise.

* This online system will keep record of job-seeker’s completed projects and payment details. Also keep records of job-provider’s all completed projects and all expired projects.

###### External Interface Requirements

* 1. **User Interface**

Some of the user interface are describe in the following table.

|  |  |
| --- | --- |
| **Major Forms Used** | **Description** |
| Login(Admin) | Admin can login by their email-id and password and can set quiz category, set quiz list, set quiz list  and can update their own profile. |
| Login(Hirer) | Hirer can login to their account by their registered email-id and password. They can post jobs and show all projects, completed projects, expired projects and they can also show their payment details  and can update own profile. |
| Login(Bidder) | Bidder can login to their profile by their registered email-id, password. They can search for jobs, can show their all completed jobs, they can quiz the online exam and can update  their own profile. |
| Registration(Hirer, Bidder) | People who wants to register in our site can register by entering all required data and type as which  they want to register. |
| Update profile(For all members) | People can update their profile by  entering all required data. |
| Post project | Job-provider can post their projects by entering project name, skill, starting date, minimum rate,  maximum rate, project type. |
| Quiz category | Admin can set quiz category by  entering quiz category field. |
| Quiz question | Admin can set quiz question by  entering quiz category, question, options and answer of the question. |

**Table 5**

* 1. Error Messages

Yet the system has been tested properly and the system generated errors have been checked by using exception handling mechanism but there can be also some error occur at the runtime unexpectedly due to server failure. Some of them are:

* Here we have provided some user defined error messages for particular warning message for particular case when a user properly log in then “login successfully” alert message will shown otherwise “Invalid password”, “please enter valid email” alert message will shown.
* On the time of registration if any person didn’t enter proper email then “please enter valid email” alert message will shown. Or if phone number is wrong then “phone number is only on digits” alert message will shown. If the email id already exists then “email id already exists” message will shown.
* If the admin doesn’t enter the quiz category then “please select quiz category name” will shown.
* If hirer wants to hire more than one person for a project then “you have already hired a person for this project” alert message will shown.

1. Performance constraints

The performance constraints part includes the functionalities that are used to run the project smoothly and faster. In our system we have used application variables to save the values that are required to redirect to next pages during any page redirection and also have cleared those application values after closing. Also the pages are redirect based on checking the application values are automatically deleted. Another significance to use application variable is that multiple pages at a time.

###### Design Constraints

* 1. Software constraints

Software used by developer’s-

|  |  |
| --- | --- |
| Operating System | Windows |
| Programming Language | Core Java |
| Front End | Eclipse, Dia Software |
| Back End | Oracle 10g |

Table 6

Minimum Software Requirements for Client’s –

|  |  |
| --- | --- |
| Operating System | Any GUI based operating system |
| Software | Any of Eclipse,Netbeans,VStdio |

Table 7

* 1. Hardware Constraints Hardware used by developer’s -

|  |  |
| --- | --- |
| Processor | Intel Core TM i3 |
| Ram | 1GB |
| Cache Memory | 512KB |
| Hard disk | 4GB |
| Keyboard | Microsoft compatible 101 or more  keyboard |

Table 8

Minimum Hardware Requirements for Client’s -

|  |  |
| --- | --- |
| Processor | Any processor |
| Ram | 512MB |

Table 9

# Design/Solution/ Methodology

* 1. **Data design**

##### USERS

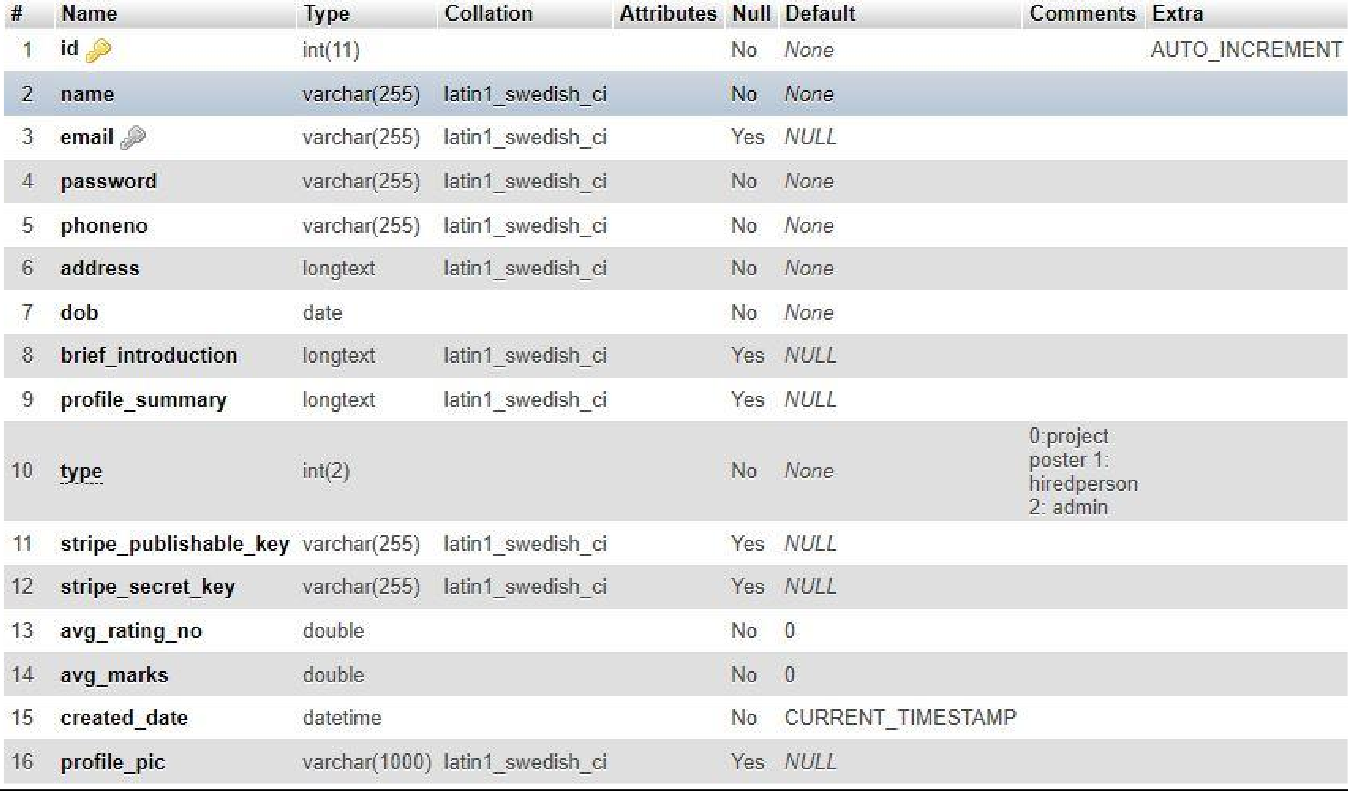
****

IMAGE 7

##### USER

****

IMAGE 8

##### QUIZ CATEGORY

****

IMAGE 9

##### QUIZ QUESTION

****

IMAGE 10

##### QUIZ OPTION

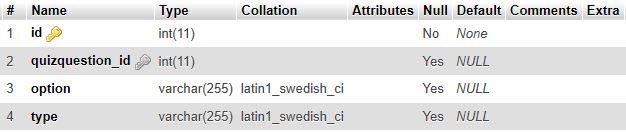
****

IMAGE 11

##### PROJECT

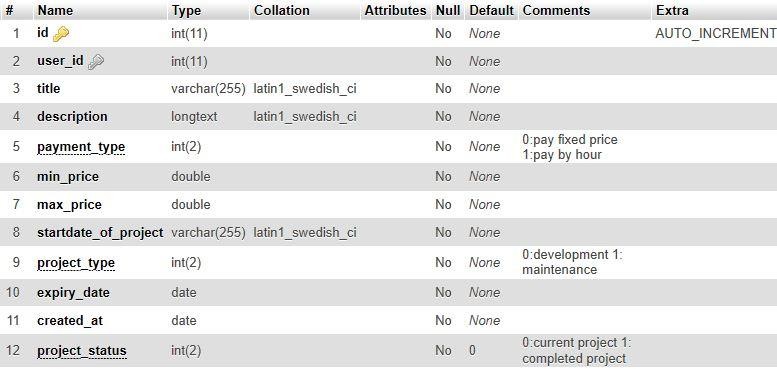
****

IMAGE 12

##### PROJECT SKILLS

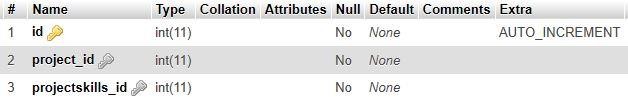
****

IMAGE 13

##### RATING DETAILS

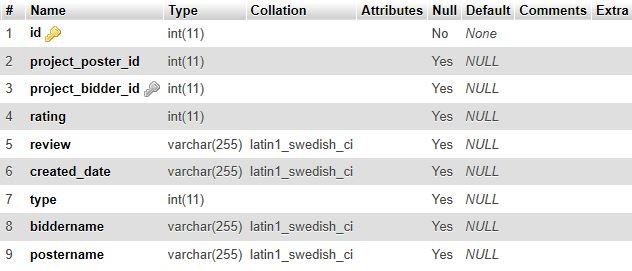
****

IMAGE 14

##### BIDDING DETAILS

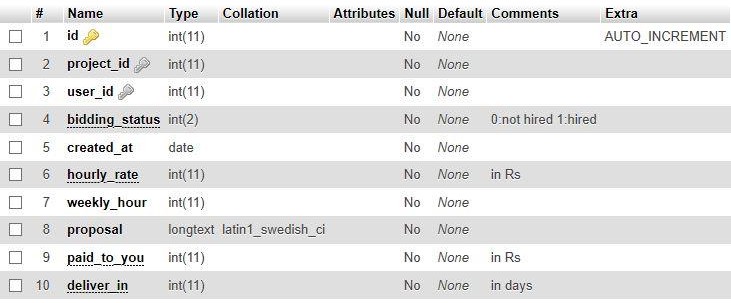
****

IMAGE 15

##### SKILLS

****

IMAGE 16

##### EXAM RESULTS

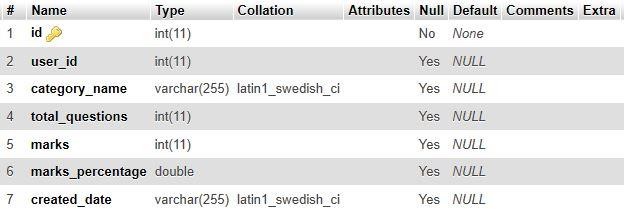
****

IMAGE 17

##### EXAM RESULT DETAILS

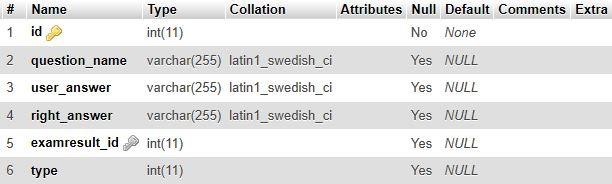
****

IMAGE 18

##### PAYMENT

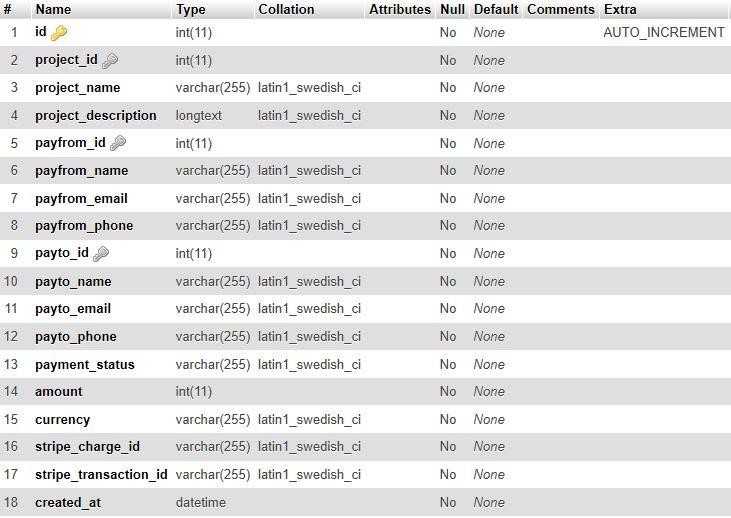
****

IMAGE 19

###### ERD diagram

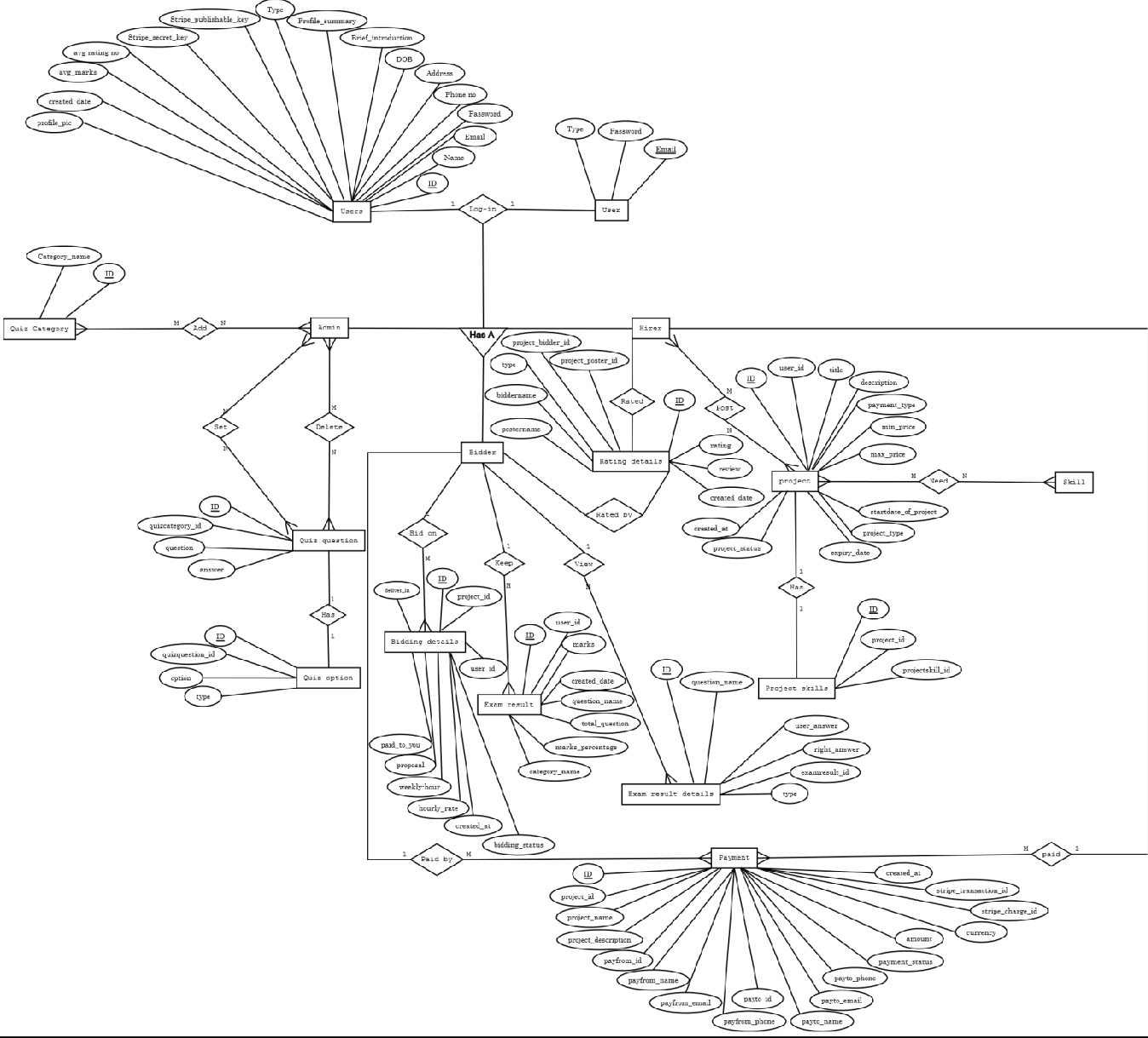
****

IMAGE 20

###### Interface Design

* **Home**

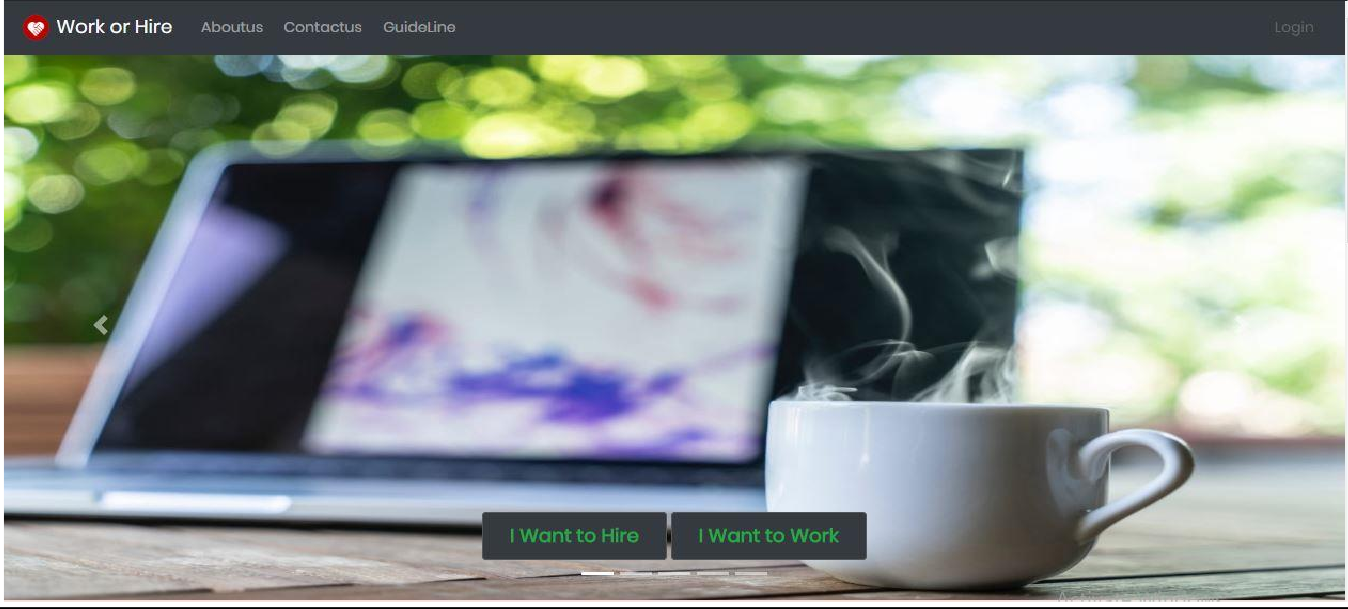
****

IMAGE 21

###### Registration

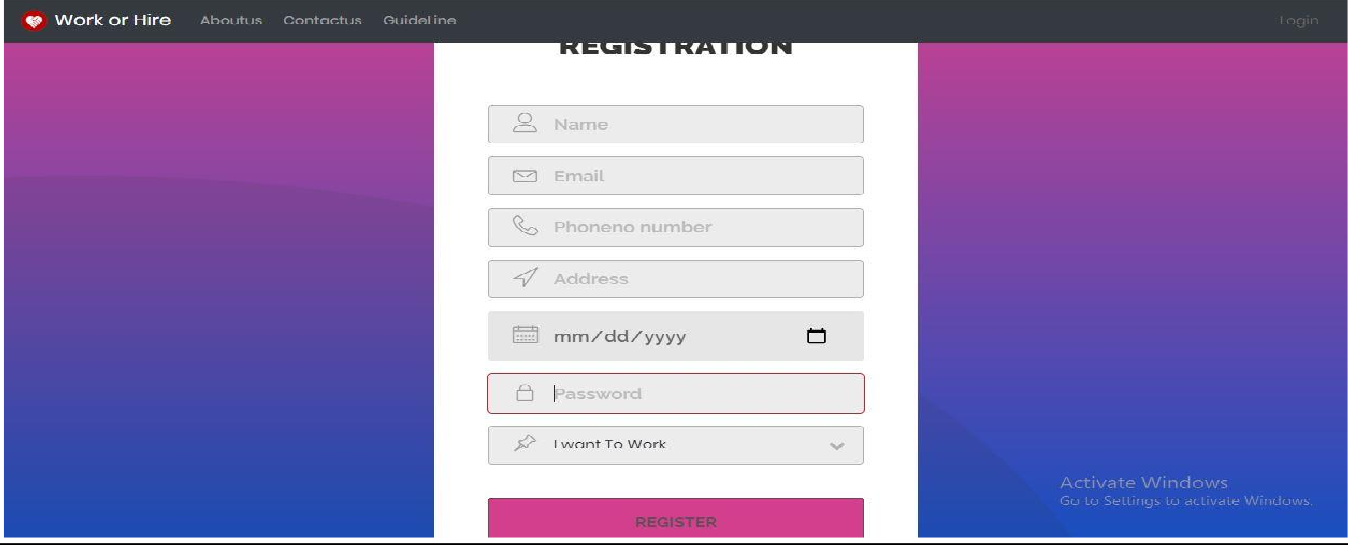
****

IMAGE 22

###### Login

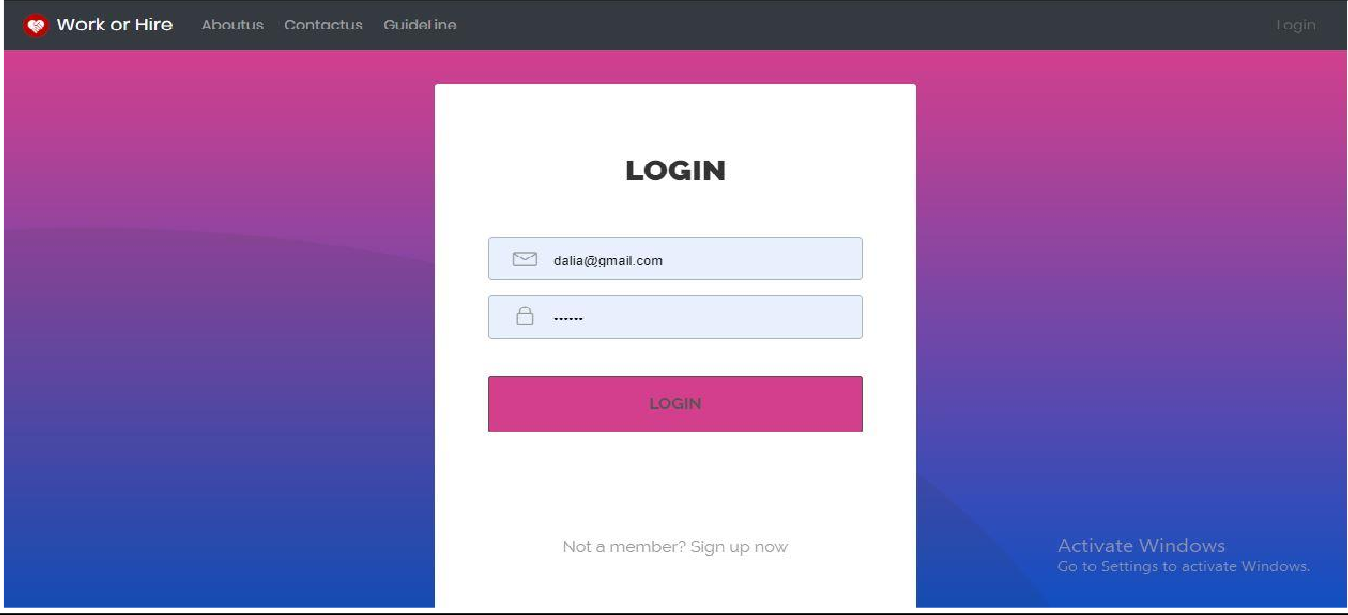
****

IMAGE 23

###### Edit profile

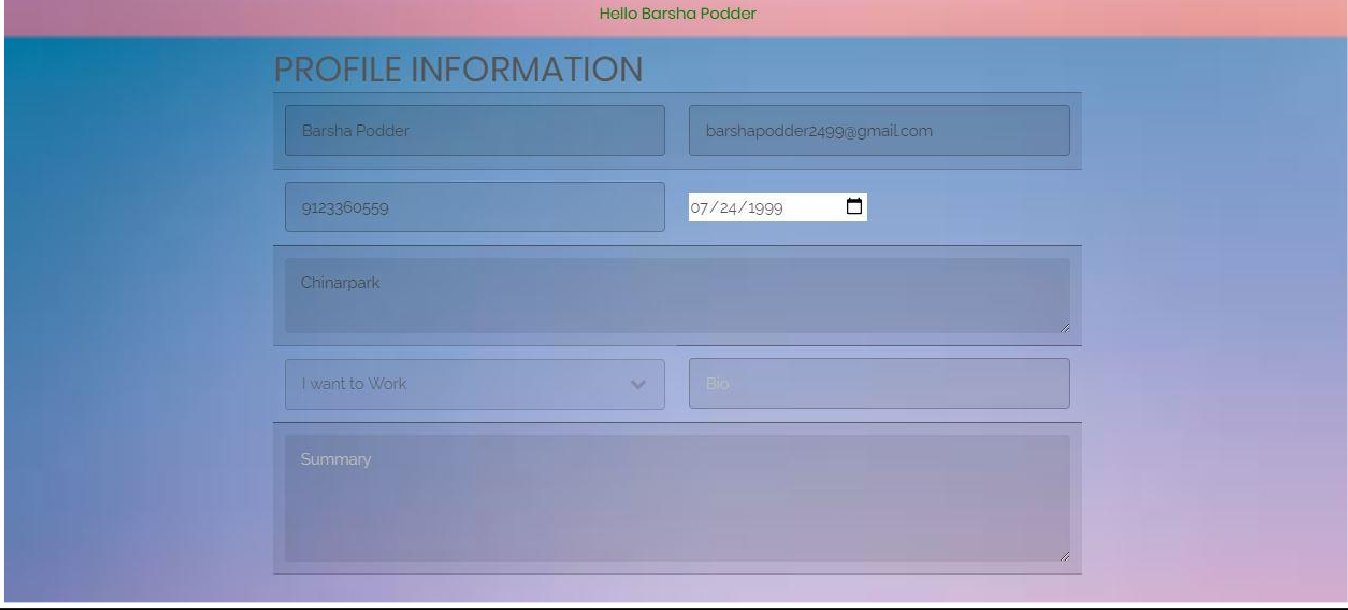
****

IMAGE 24

###### Architectural Design

* **Use case diagram**

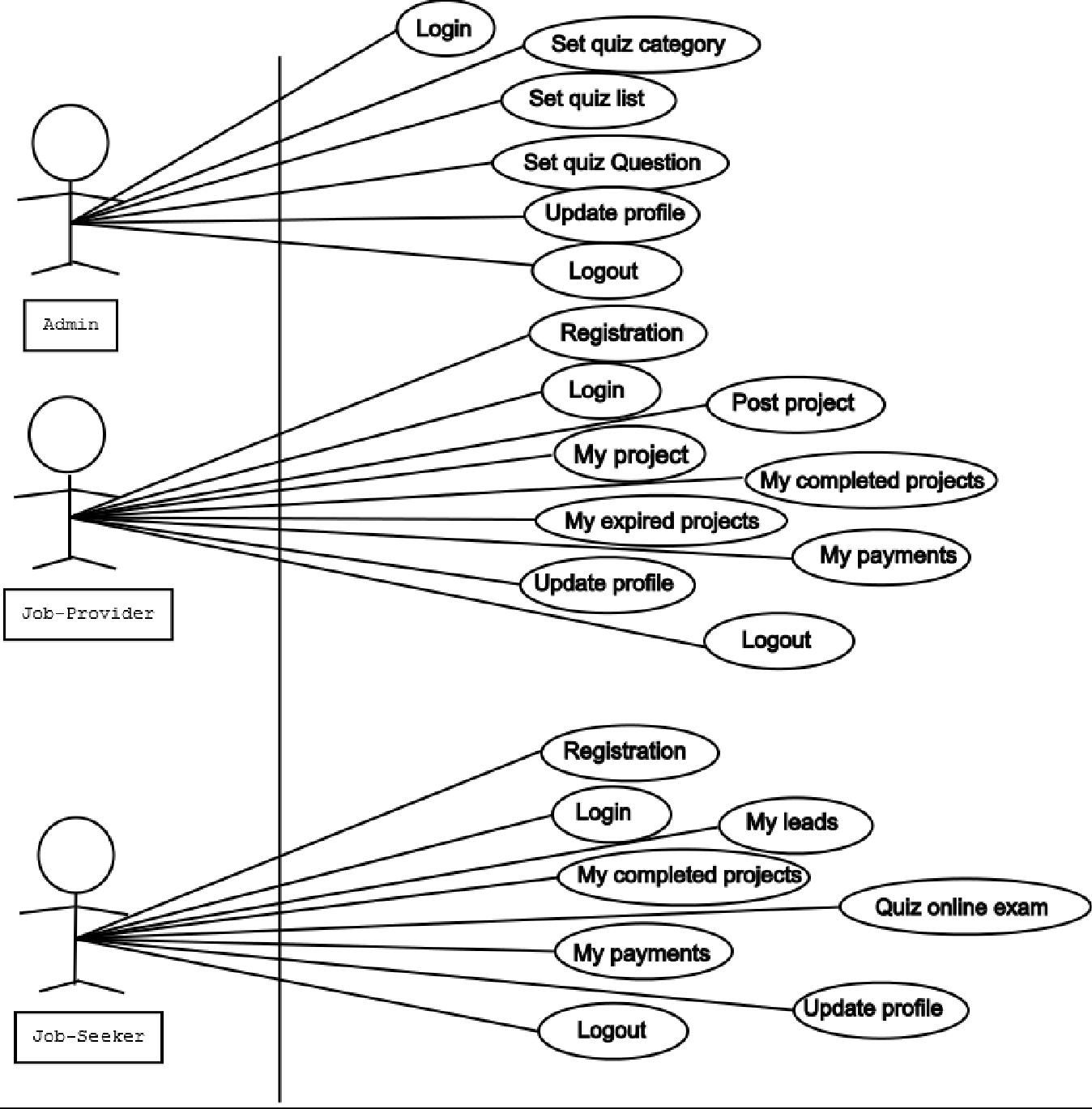
****

IMAGE 25

###### Procedural Design

* **0 level DFD**

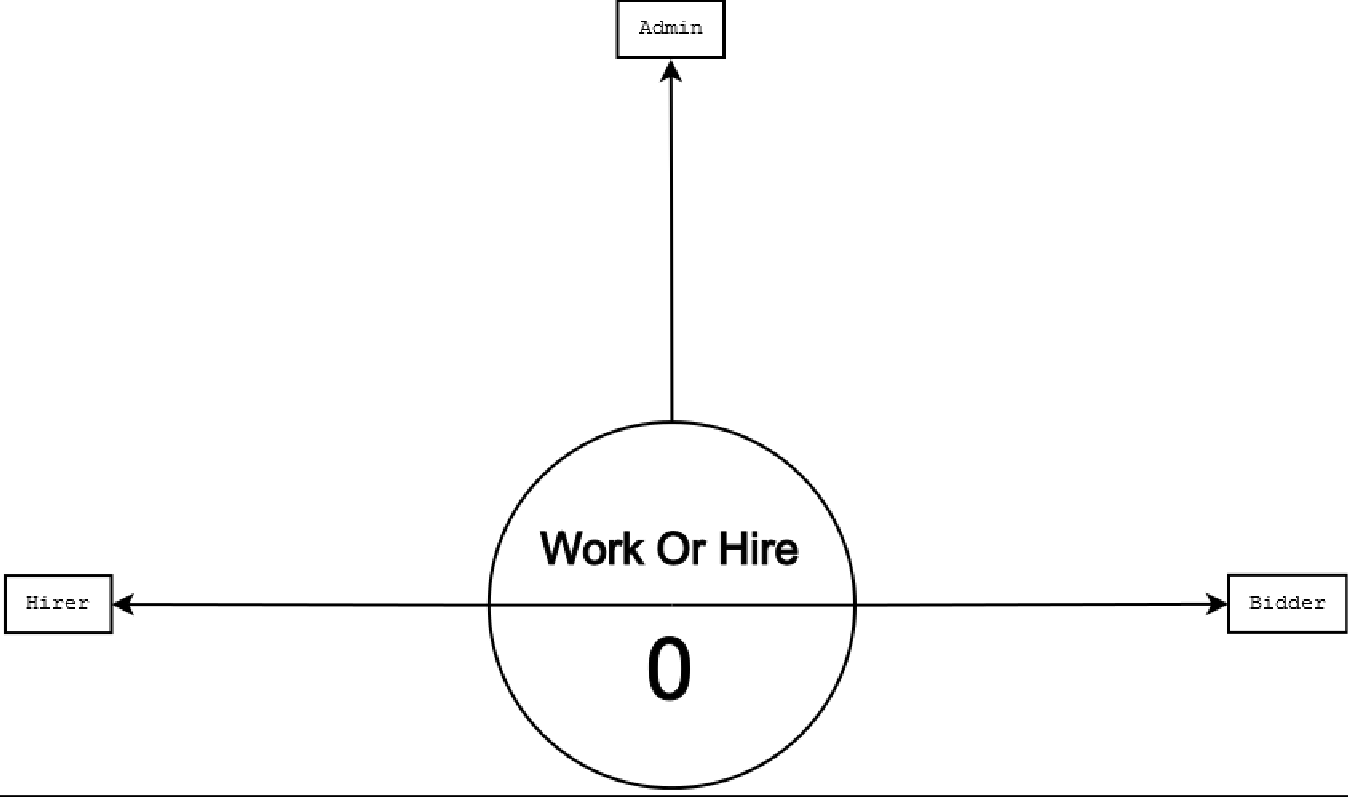
****

IMAGE 26

###### 1.1 level DFD for Admin module

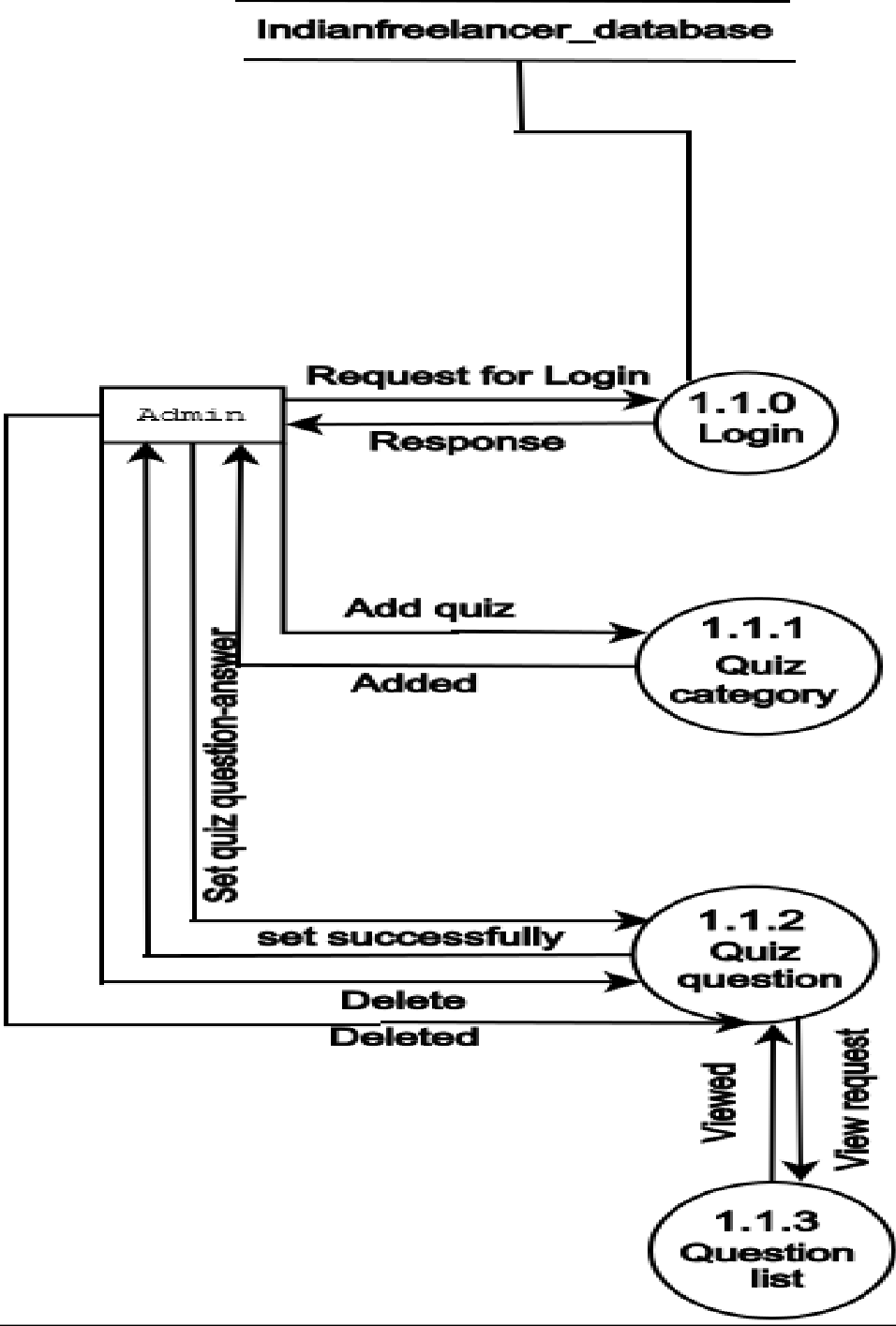
****

IMAGE 27

###### 1.2 Level DFD for Hirer module

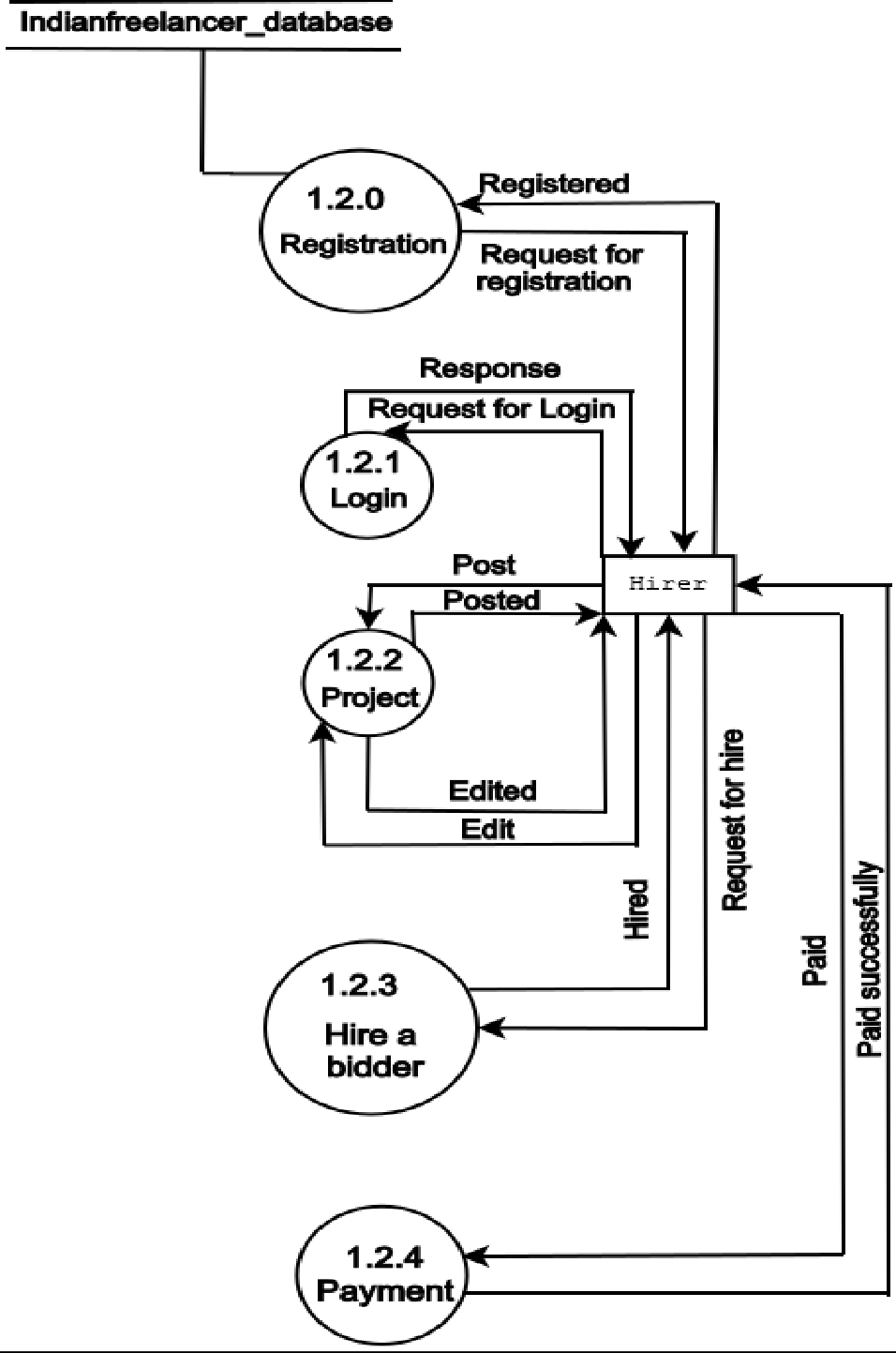
****

IMAGE 28

###### 1.3 Level DFD for Bidder module

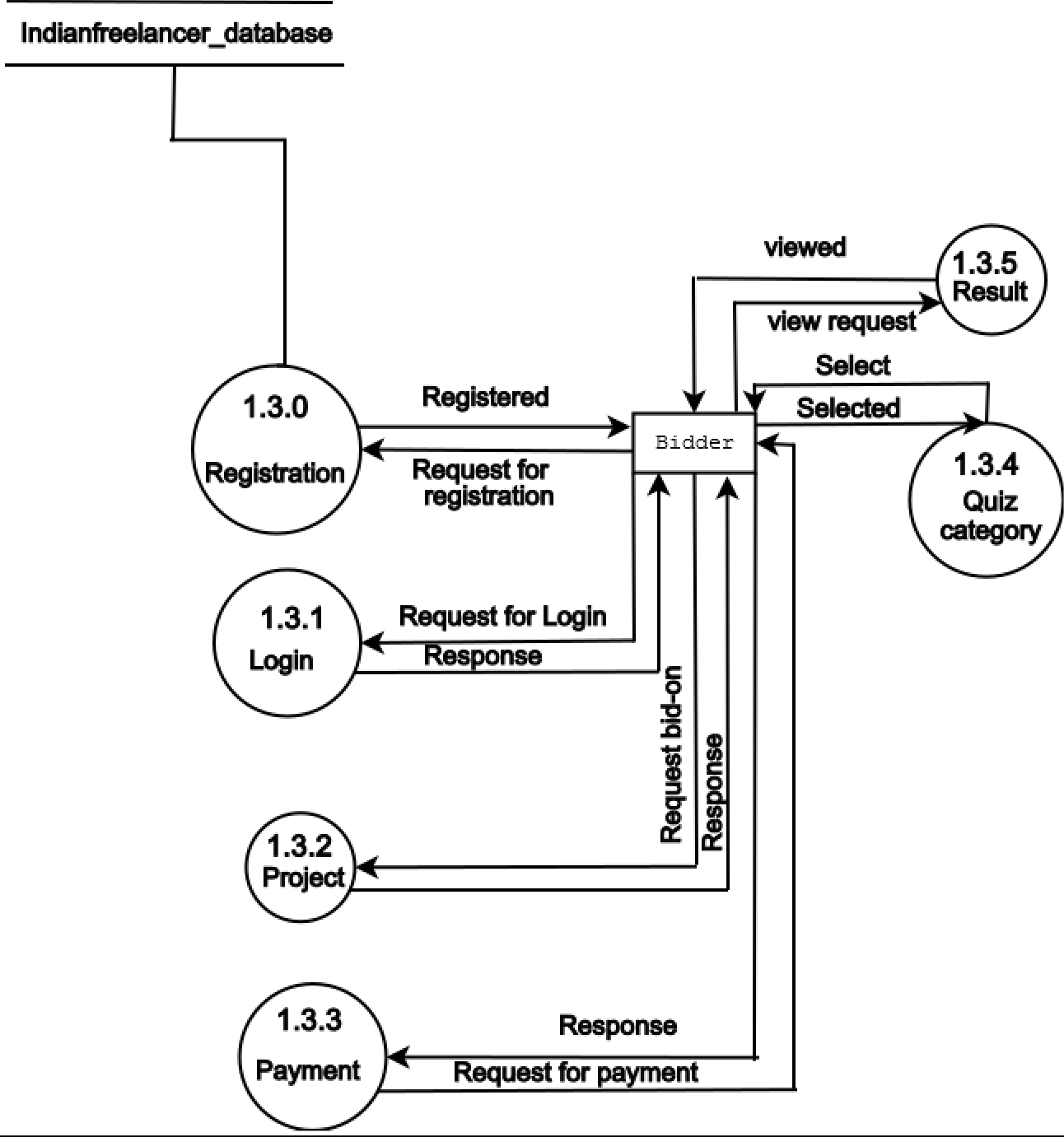
****

IMAGE 29

###### Flow Diagram

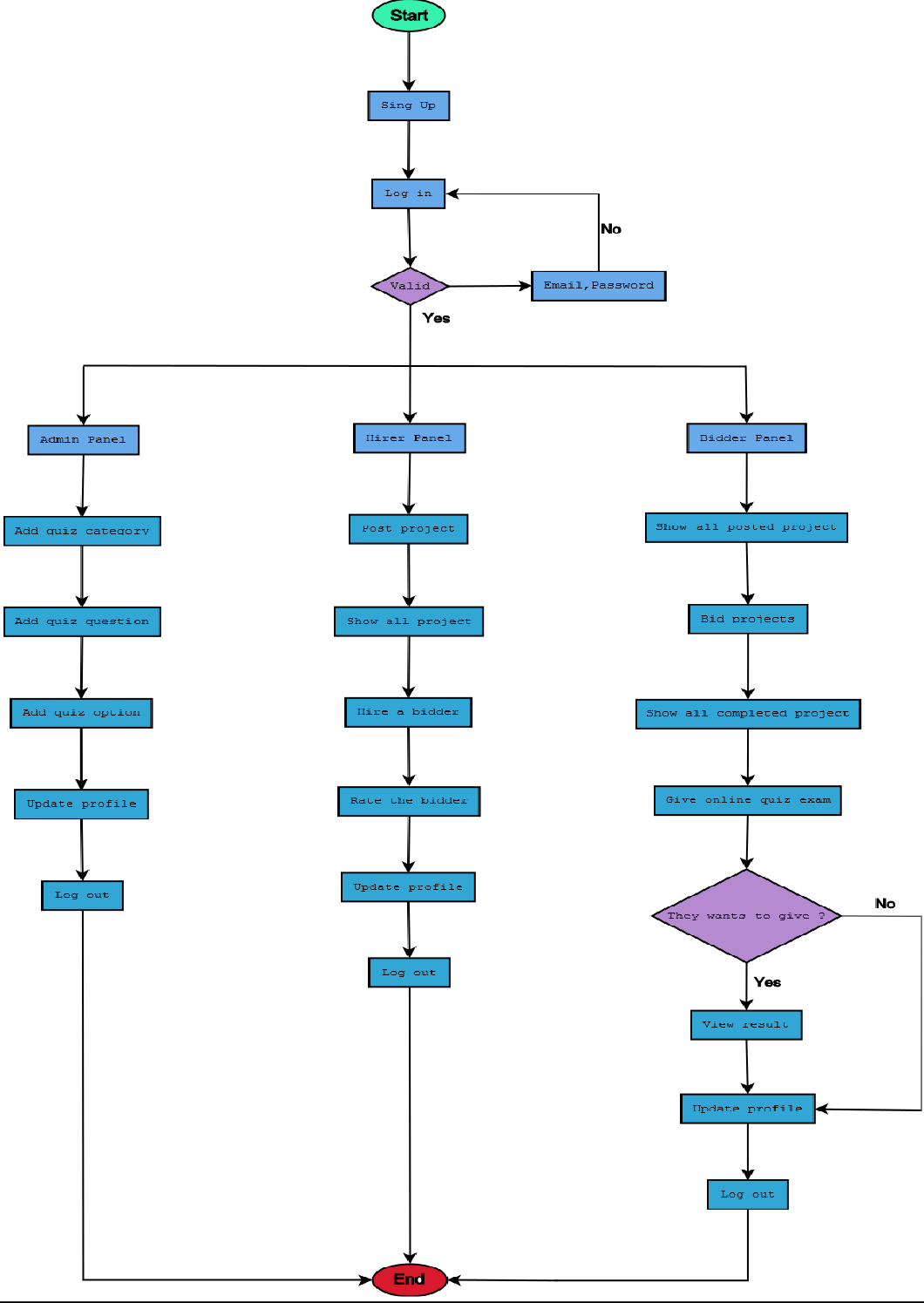
****

IMAGE 30

# Coding Standard Followed And Assumptions

##### CODING STANDARD

The main goal of the recommendation is to improve readability and thereby the understanding and the maintainability and the general quality of the code. It is impossible to cover all the specific cases in a general guide and the programmer should be flexible.

##### NAMING CONVENTION

###### Names representing packages should be in all lower case

mypackage, com.company.application.ui Package naming convention used by Sun for the Java core packages. The initial package name representing the domain name must be in lower case.

###### Names representing types must be nouns and written in mixed case starting with upper case

Line,AudioSystem Common practice in the Java development community and also the type naming convention used by Sun for the Java core packages.

###### Variable names must be in mixed case starting with lower case

Makes variables easy to distinguish from types, and effectively resolves potential naming collision as in the declaration line.

###### Names representing constants (final variables) must be all uppercase using underscore to separate words

e.g. MAX\_ITERATIONS, COLOR\_RED

###### Names representing methods must be verbs and written in mixed case starting with lower case

getName(), computeTotalWidth()

Common practice in the Java development community and also the naming convention used by Sun for the Java core packages. This is identical to variable names, but methods in Java are already distinguishable from variables by their specific form.

###### Private class variables should have underscore suffix

class Person

{

private String name\_; ...

}

Apart from its name and its type, the scope of a variable is its most important feature. Indicating class scope by using underscoreclass variables from local scratch variables.class variables are considered to have higher significance than method variables.

A side effect of the underscore naming convention is that it nicely resolves the problem of finding reasonable variable names for setter methods:

voidsetName(String name)

{ name\_ = name;}

An issue is whether the underscore should be added as a prefix or as a suffix.

###### Generic variables should have the same name as their type

**45**

voidsetTopic(Topic topic)

// NOT: void setTopic(Topic value)

// NOT: void setTopic(Topic aTopic)

// NOT: void setTopic(Topic t) void connect(Database database)

// NOT: void connect(Database db)

It reduces complexity by reducing the number of terms and names used. Also makes it easy to deduce the type given a variable name only.

###### Variables with a large scope should have long names; variables with a small scope can have short names

Scratch variables used for temporary storage or indices are best kept short. A programmer reading such variables should be able to assume that its value is not used outside a few lines of code. Common scratch variables for integers are i, j, k, m, n and for characters c and d.

**The terms get/set must be used where an attribute is accessed directly**

employee.getName();employee.setName(name);matrix.getElement(2, 4); matrix.setElement(2, 4, value);

Common practice in the Java community and the convention used by Sun for the Java core packages.

###### The is prefix should be used for boolean variables and methods

isSet, isVisible, isFinished, isFound, isOpen

This is the naming convention for boolean methods and variables used by Sun for the Java core packages. Using the is prefix solves a

common problem of choosing bad Boolean names like status or flag. isStatus or isFlag simply doesn't fit, and the programmer is forced to chose more meaningful names.

###### The term find can be used in methods where something is looked up

vertex.findNearestVertex(); matrix.findSmallestElement();

Consistent use of the term enhances readability. The n prefix should be used for variables representing a number of objects. E.g. nPoints, nLines etc.

**The Iterator variables should be called i, j, k etc**. for (Iterator i = points.iterator(); i.hasNext(); ) { : } for (int i = 0; i <nTables; i++) { : }

The notation is taken from mathematics where it is an established convention for indicating iterators. Variables named j, k etc. should be used for nested loops only.

###### User Defined Exception classes should be suffixed with Exception class

AccessException extends Exception { : }

Exception classes are really not part of the main design of the program, and naming them like this makes them stand out relative to the other classes. This standard is followed by Sun in the basic Java library.

###### Functions (methods returning an object) should be named after what they return and procedures (void methods) after what they do

Increase readability. Makes it clear what the unit should do and especially all the things it is not supposed to do. This again makes it easier to keep the code clean of side effects.

###### Java source files should have the extension .java

**Special characters like TAB and page break must be avoided**

These characters are bound to cause problem for editors, printers, terminal emulators or debuggers when used in a multi-programmer, multi-platform environment.

###### Variables should be initialized where they are declared and they should be declared in the smallest scope possible

This ensures that variables are valid at any time. Sometimes it is impossible to initialize a variable to a valid value where it is declared. In these cases it should be left uninitialized rather than initialized to some phony value.

###### Variables must never have dual meaning

Enhances readability by ensuring all concepts are represented uniquely. Reduce chance of error by side effects.

###### Floating point constants should always be written with a digit before the decimal point

double total = 0.5; // NOT: double total = .5;

The 0.5 is a lot more readable than .5; there is no way it can be mixed with the integer.

###### Arrays should be declared with their brackets next to the type

double[] vertex; // NOT: double vertex[]; public static void main(String[] arguments) public double[] computeVertex()

The reason for is twofold. First, the array-ness is a feature of the class, not the variable. Second, when returning an array from a method, it is not possible to have the brackets with other than the type (as shown in the last example).

##### ASSUMPTIONS

* Job-provider can post jobs.
* Job-seeker choose projects as there skills.

# Result Set Analysis

**49**

|  |  |
| --- | --- |
| **Form Name** | **Purpose** |
| 1.Home page | Providing opportunity to choose login or registration. |
| 2.Login | Providing opportunity that by entering password and email-id they  can login to their profile. |
| 3.Register | Providing opportunity that by filling all the required data people can  register as which they want to register. |
| 4.Edit profile | Providing opportunity to user that they can edit their profile details. |
| 5. Edit profile  picture | Providing opportunity to user that they can update their profile  picture. |
| 6.Post project | Providing opportunity to hirer that they can post their projects. |
| 7.My projects | Providing opportunity to hirer and bidder that hirer can show their  posted projects and bidder can show their completed projects. |
| 8.Completed  projects | Providing opportunity to hirer and bidder that they can show their  completed projects. |
| 9.Expired  projects | Providing opportunity to hirer that they can show their expired  projects. |
| 10.Payment | Providing opportunity to hirer that they can pay bidders who  completed their projects. |
| 11.Online quiz  exam | Providing opportunity to all bidders that they can take online quiz. |
| 12.Show exam  result | Providing opportunity to bidders that they can show their quiz result. |
| 13.Place bid | Providing opportunity to bidders that they can bid for the projects. |
| 14.View hired  person | Providing opportunity to hirer that they can show the hired person for  a particular projects. |
| 15.View all  bids | Providing opportunity to hirer that can show all bids for their projects  and choose a particular bidder. |
| 16.Rating  submit | Providing opportunity to hirer that they can rate bidders who work  for them. |
| 17.Add quiz  category | Providing opportunity to admin that they can add quiz category by  selecting a particular category. |
| 18.Add quiz  question | Providing opportunity to admin that they can add quiz question and  set answer. |
| 19.View quiz  category list | Providing opportunity to admin that they can show quiz category,  questions and options. |

**Table 10**

# Testing

## Testing

The aim of the system testing process was to determine all defects in our project. The program was subjected to a set of inputs and various observation were made and based on these observations it will be decided whether the program behaves as expected or not. Our projects went through two levels of testing.

1. Unit testing.
2. Integration testing.

###### Unit Testing

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module we need to provide complete environment i.e. besides the module we would require. The produces belonging to other modules that the module under test calls

* + - Non local data structures that module accesses.
    - A procedure to call the functions of the module under test with appropriate parameters.

Unit testing was done on each and every module.

1. Testing for login form- This form is used for log in of administrator of the system. In this we enter the Email-id and password if both are correct administrator page will open otherwise if any data is wrong it will redirected back to the login page and will show alert message like “Invalid Email or password”.
2. Testing for registration form- This form is used for registration of any user. In this form we enter name, email, phone number, DOB, password, profile picture and type if all

are valid then the profile will be created otherwise “please enter valid email” or “Phone number only in digits” or “email id already exist” alert message will be shown.

1. Testing for Edit form- This from is used for edit details of any user. In this form we enter name, email, phone number, DOB, password, profile picture, brief introduction and summary and if all information are correct then profile will be updated otherwise “please enter valid email” or “phone number only in digits” alert message will be shown.

###### Test For Admin module

* 1. Set quiz Category Page- This form is used by only admin for set quiz category otherwise “please select quiz category name” alert message will shown.
  2. Set quiz question page- This form is used by only admin for set quiz question and options properly otherwise it will redirect to the page.
  3. Show quiz list page- This page is used only by admin for show all quiz category in list. If there is no quiz question then it will show empty and only admin can add and delete quiz question whenever they want from these page.

###### Test For Hirer module

* + 1. Post project form- this form is used by only hirer for post a project. They have to enter project name, skill, starting date, project type, minimum rate, maximum rate and project details if all data are correct then project will be posted otherwise not.
    2. Edit project- This form is by only hirer for edit a posted project.
    3. Hire a bidder- This page is used by a hirer that they can hire a bidder for a project. If they already hire a bidder for that project and try to hire again then “You’ve already hire a bidder” alert message will be shown.
    4. Rating review- This page will used by hirer to give rating to the bidder who work for their project.

###### Test For Bidder module

1. Bid a project- This page is only used by bidders to bid any project which they want to do.
2. Online exam- This page is used by bidders to take online quiz exam.
3. Show all projects- This page is used by bidders to show all projects, completed projects.

###### Integration Testing

In this type of testing we test various integration of the project module by providing input. The primary objective is to test the module interfaces in order to ensure that no error are occurring when one module invokes the other modules.

# ​Future Scope of the project

**Future Scope of the Project**

The project Work Or Hire gives a new evolution on the era of technology. It makes entire working and hiring process online where bidder can search for job as their skill wise and hirer can hire a bidder and rate them. It has also has a facility that hirer and bidder can login to their profile and edit details. There is a future scope of this facility that many more features such as chat option can be added that hirer and bidder can chat on the website, feature of video conferencing over website can be added, we can also add payment gateway, a feature of online video lectures can be added related to how bidders can improve their skills, and the project is designed in local host there is many limitations in local host if we design it on domain we can improve our project.

The site can be easily implemented. We can add new features as when we require. There is flexibility in all the modules.

###### Software Scope:

Extensibility: This software is extendable in various ways and new features can be added.

Reusability: Reusability is possible as when require in this website. We can update in next version. Reusable software reduces design, coding and testing cost by amortizing effort over several designs. Reducing the amount of code is correct. We follow up both type of reusability: Sharing at newly written code on new projects.

Understandability: A method is understandable if someone other than the creator after a time lapse.

Cost-effectiveness: Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement.

# ​Conclusion

**55**

### Conclusion

The “**Work [OR] Hire**” made entire process online to reduce cost of third- party. The main focus of the project is to work without geographical and time bound. The maintenance of all records of users is stored in database so it can be retrieve easily. The record of bidders like bidding details, payment, completed project and the record of hirer like posted project, completed project are stored in database. We are working on high security with all our users’ data. The editing also made simpler. The user has to just type in the required field and press the update button to update desire field.

The projects are given a particular project id no. so that they can be accessed correctly without errors. Our main aim of the project is work efficiently with skills without having any problem with time, place.

# ​Reference and

**Bibliography**

**56**

### Reference and Bibliography

* Software engineering book by Rajiv Mall
* [www.javatpoint.com](http://www.javatpoint.com/)
* [www.studytonight.com](http://www.studytonight.com/)
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* Bootstrap 5
* W3 schools

# Appendix

## Appendix

* 1. **Appendix I**

## Home Page:

<%@pageimport=*"indianfreelancer.RegisterDao"*%>

<%@pageimport=*"indianfreelancer.Register"*%>

<%@pageimport=*"indianfreelancer.ProjectDao"*%>

<%@pageimport=*"indianfreelancer.Project"*%>

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"java.util.\*"*%>

<!DOCTYPEHTML>

<html>

<title>Home</title>

<style>

*.inxtxt*{

font-family: *serif*; font-size: *19px*; color: *black*; padding: *5px*;

}

</style>

<%@includefile=*"includes/header.jsp"*%>

<%@includefile=*"includes/banner.jsp"*%>

<!-- Highlights -->

<sectionclass=*"wrapper"*>

<divclass=*"inner"*>

<h4style="text-align:*center*;color:*#ea3f0c*;text- decoration:*underline*;margin:*5px*;">Our All Projects :~</h4>

<pstyle="text-align:*center*;color:*black*;" class=*"inxtxt"*>These are our all projects which are posted by users.<br/>Register to bid these projects and then you will get hire.</p>

<divclass=*"highlights"*>

<%

ProjectDao obj=**new**ProjectDao();

List<Project> list=obj.homePageProjectsList(); request.setAttribute("list",list);

RegisterDao obj1=**new**RegisterDao(); List<Register> list1=obj1.homePageUserList();

request.setAttribute("list1",list1);

%>

<%**if**(list.isEmpty()) { out.print( "Currently There is no project available." ); } %>

<c:forEachitems=*"*${ list }*"*var=*"u"*>

<section>

<divclass=*"content"*>

<header>

<h5style="color:*green*">${ u.getTitle() }</h5>

<pstyle="color:*black*;font-weight:*bold*">Posted Date : ${ u.getCreated\_at() }</p>

<pstyle="color:*black*;font-weight:*bold*">Expiry Date : ${ u.getExpiry\_date()}</p>

<c:iftest=*"*${u.getProject\_status()==0}*"*>

<pstyle="color:*#ec4b0d*">Status: Pending<p>

</c:if>

<c:iftest=*"*${u.getProject\_status()==1}*"*>

<pstyle="color:*#ec4b0d*">Status: Completed<p>

</c:if>

weight:*bold*">Posted By : ${ u.getUsername() }</p>

</div>

</section>

</header>

<pstyle="color:*#070796*;font-

</c:forEach>

</div>

</div>

</section>

<%@includefile=*"includes/footer.jsp"*%>

</body>

</html>

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Hirer Completed Projects</title>

</head>

<bodystyle="background-image: *url('front/images/3.jpg')*">

<body>

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

%@pageimport=*"indianfreelancer.Project"*%

<%@pageimport=*"indianfreelancer.ProjectDao"*%>

<%@pageimport=*"java.util.\*"*%>

<%@includefile=*"includes/header.jsp"*%>

<%-- <%@ include file="includes/banner.jsp" %>--%>

<%@includefile=*"checklogin.jsp"*%>

<%

**int** user\_id=Integer.parseInt((String)session.getAttribute("loginid")); List<Project> list=ProjectDao.myCompletedProjectsList(user\_id); request.setAttribute("list",list);

%>

<h3style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*#fc03a9*;text-decoration:*underline*">My Completed Projects </h3>

<divstyle="width:*90%*; margin:*0 auto !important*; margin:*20px*;">

<pstyle="color:*#fc03a9*;margin-bottom:*20px*; text-align:*center*;font-size:*18px*;" >

<%**if**(list.isEmpty()) { out.print( "Currently You have no completed project. Please post your project." ); } %>

</p>

<c:forEachitems=*"*${ list }*"*var=*"u"*>

<divclass=*"container"*>

<divclass=*"card float-left"*style="background-color:*rgba(0,0,0,0.5)*;width: *24rem*;">

<divclass=*"card-wrapper text-center"*style="border: *2px solid #ccc*;border-radius: *5px*;border-color:*#0b8de3*;padding: *10px*;">

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Project Name :</span> ${ u.getTitle() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Project Description :</span> ${ u.getDescription() }</p>

<p>

<spanstyle="font-weight:*bold*">Payment Type :</span>

<c:iftest=*"*${u.getPayment\_type()==0 }*"*>

<c:outvalue=*"Pay Fixed Price"*/>

</c:if>

</p>

<c:iftest=*"*${u.getPayment\_type()==1 }*"*>

<c:outvalue=*"Pay By Hour"*/>

</c:if>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Min Price :</span>Rs ${ u.getMin\_price() }&nbsp&nbsp&nbsp&nbsp

<spanstyle="font-weight:*bold*"> Max Price :</span>Rs ${ u.getMax\_price() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Project Will Start :</span> ${ u.getStartdate\_of\_project() }</p>

<p>

<spanstyle="font-weight:*bold*"> Project Type :</span>

<c:iftest=*"*${u.getProject\_type()==0 }*"*>

<c:outvalue=*"Development”/>*

</c:if>

<c:iftest=*"*${u.getProject\_type()==1 }*"*>

<c:outvalue=*"Maintenance"*/>

</c:if>

</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Post Date :</span> ${ u.getCreated\_at() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Expiry Date :</span> ${ u.getExpiry\_date() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Required Skills :</span> ${ u.getSkills\_name() }</p>

<pstyle="margin-top: *10px*;"><ahref=*"viewhiredperson?pid=*${ u.getId() }*"*class=*"btn btn-outline-warning"*style="padding: *8px*;border-radius: *5px*;">View Hired Person</a>

</p>

</div>

</div>

</div>

</c:forEach>

</div>

<%-- <%@ include file="includes/footer.jsp" %>--%>

</body>

</html>

## Login Page:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtmlPUBLIC"-//W3C//DTD HTML 4.01

Transitional//EN""<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<metahttp-equiv=*"Content-Type"*content=*"text/html; charset=ISO-8859-1"*>

<title>Login</title>

</head>

<body>

<%@includefile=*"includes/header.jsp"*%>

<pstyle="color:*red*" class=*"commonmsg"*><% String logerr=(String)request.getAttribute("loginerrmsg"); **if**(logerr!=**null**)

{

out.print(logerr);

}

%></p>

<pstyle="color:*red*" class=*"commonmsg"*><% String chklogin=(String)request.getAttribute("checkloginmsg"); **if**(chklogin!=**null**)

{

out.print(chklogin);

}

%></p>

<pstyle="color:*green*;" class=*"commonmsg"*>

<% String logoutmsg=(String)request.getParameter("logoutmsg");

**if**(logoutmsg!=**null**)

{

out.print(logoutmsg);

}

%>

</p>

<divclass=*"limiter"*>

<divclass=*"container-login100"*>

<divclass=*"wrap-login100 p-l-50 p-r-50 p-t-77 p-b-30"*>

<formclass=*"login100-form validate- form"*action=*"loginaction"*method=*"post"*>

<spanclass=*"login100-form-title p-b-55"*> Login

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data-validate=*"Valid email is required:* [*ex@abc.xyz"*](mailto:ex@abc.xyz)>

<inputclass=*"input100"*type=*"text"*name=*"useremail"*placeholder=*"Email"*style="pad ding-left: *60px*;">

<pstyle="color:*red*"><% String em=(String)request.getAttribute("emailmsg");

**if**(em!=**null**)

{

out.print(em);

}

%></p>

</div>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-envelope"*></span>

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data-

validate=*"Password is required"*>

<inputclass=*"input100"*type=*"password"*name=*"userpassword"*placeholder=*"Password"* style="padding-left: *60px*;">

<pstyle="color:*red*"><% String ps=(String)request.getAttribute("passmsg");

**if**(ps!=**null**)

{

out.print(ps);

}

%></p>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-lock"*></span>

</span>

</div>

*btn"*name=*"submit"*>

<divclass=*"container-login100-form-btn p-t-25"*>

<buttonclass=*"login100-form-*

Login

</button>

</div>

<divclass=*"text-center w-full p-t-115"*>

<spanclass=*"txt1"*>

Not a member?

</span>

<aclass=*"txt1 bo1 hov1"*href=*"register"*> Sign up now

</div>

</div>

</div>

</div>

</form>

</a>

<%@includefile=*"includes/footer.jsp"*%>

</body>

</html>

## Register Page:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Registrarion</title>

</head>

<body>

<%@includefile=*"includes/header.jsp"*%>

<%-- <%@ include file="includes/banner.jsp" %>--%>

<divclass=*"limiter"*>

<divclass=*"container-login100"*>

<divclass=*"wrap-login100 p-l-50 p-r-50 p-t-77 p-b-30"*>

<formclass=*"login100-form validate- form"*action=*"registeraction"*method=*"post"*>

<spanclass=*"login100-form-title p-b-55"*> Registration

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data-

validate=*"Name is required"*>

<inputclass=*"input100"*type=*"text"*name=*"username"*placeholder=*"Name"*style="paddi ng-left: *60px*;">

<% String nameerr=(String)request.getAttribute("nameerr");

**if**(nameerr!=**null**)

{

out.print(nameerr);

}

%>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-user"*></span>

</span>

</div>

<divclass=*"wrap-input100 validate-input m-b-16"*data- validate=*"Valid email is required:* [*ex@abc.xyz*](mailto:ex@abc.xyz)*"*>

<inputclass=*"input100"*type=*"text"*name=*"useremail"*placeholder=*"Email"*style="pad ding-left: *60px*;">

<pstyle="color:*red*"><% String em=(String)request.getAttribute("emailmsg");

**if**(em!=**null**)

{

out.print(em);

}

%></p>

</div>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-envelope"*></span>

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data-

validate=*"Phone is required"*>

<inputclass=*"input100"*type=*"text"*name=*"userphoneno"*placeholder=*"Phoneno number"*style="padding-left: *60px*;">

<pstyle="color:*red*"><% String phoneerr=(String)request.getAttribute("phoneerr");

**if**(phoneerr!=**null**)

{

out.print(phoneerr);

}

%>

<% String phderr=(String)request.getAttribute("phderr");

**if**(phderr!=**null**)

{

out.print(phderr);

}

*handset"*></span>

%></p>

</div>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-phone-*

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data-

validate=*"Address is required"*>

<inputclass=*"input100"*type=*"text"*name=*"useraddress"*placeholder=*"Address"*style= "padding-left: *60px*;">

<pstyle="color:*red*"><% String addresserr=(String)request.getAttribute("addresserr");

**if**(addresserr!=**null**)

{

out.print(addresserr);

}

%>

<% String aderr=(String)request.getAttribute("aderr");

**if**(aderr!=**null**)

{

%></p>

}

</div>

out.print(aderr);

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-location"*></span>

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data- validate=*"Valid date-of-birth is required:dd/mm/yyyy"*>

<inputclass=*"input100"*type=*"date"*name=*"userdob"*placeholder=*"Date-of- Birth"*style="padding-left: *60px*;">

<pstyle="color:*red*"><% String doberr=(String)request.getAttribute("doberr");

**if**(doberr!=**null**)

{

out.print(doberr);

}

%>

<% String derr=(String)request.getAttribute("derr");

**if**(derr!=**null**)

{

out.print(derr);

}

%></p>

*full"*></span>

</div>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-calendar-*

</span>

<divclass=*"wrap-input100 validate-input m-b-16"*data-

validate=*"Password is required"*>

<inputclass=*"input100"*type=*"password"*name=*"userpassword"*placeholder=*"Password"* style="padding-left: *60px*;">

<pstyle="color:*red*"><% String ps=(String)request.getAttribute("passmsg");

**if**(ps!=**null**)

{

out.print(ps);

}

%></p>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-lock"*></span>

</span>

</div>

<!-- Testing -->

<divclass=*"wrap-input100 validate-input m-b-16"*data- validate=*"Selection is required"*>

name=*"usertype"*> Work</option> Hire</option>

</div>

<selectstyle="padding-left: *60px*;"

<optionvalue=*"1"*>I want To

<optionvalue=*"0"*>I want To

</select>

<spanclass=*"focus-input100"*></span>

<spanclass=*"symbol-input100"*>

<spanclass=*"lnr lnr-pushpin"*></span>

</span>

<!-- Testing -->

*btn"*name=*"submit"*>

<divclass=*"container-login100-form-btn p-t-25"*>

<buttonclass=*"login100-form-*

Register

</button>

</div>

</div>

</form>

</div>

</div>

<%@includefile=*"includes/footer.jsp"*%>

</body>

</html>

## Edit Profile:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtmlPUBLIC"-//W3C//DTD HTML 4.01

Transitional//EN""<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<title>Edit Profile</title>

<style>

*.award*{

width: *50%*; height: *150px*; float: *left*;

text-align: *center*;

}

*.medal*{

width: *50%*; height: *150px*; float: *left*;

text-align: *center*;

}

*.mkshw*{

font-family: *sans-serif*; font-weight: *bold*;

color: *green*; font-size: *19px*; margin: *5px*; padding: *5px*;

text-align: *center*;

}

</style>

<metahttp-equiv=*"Content-Type"*content=*"text/html; charset=ISO-8859-1"*>

<title>Edit profile</title>

</head>

<body>

<%@includefile=*"includes/header.jsp"*%>

<%-- <%@ include file="includes/banner.jsp" %>--%>

<%@taglibprefix=*"fmt"*uri=*"*[*http://java.sun.com/jsp/jstl/fmt*](http://java.sun.com/jsp/jstl/fmt)*"*%>

<%@includefile=*"checklogin.jsp"*%>

<%@pageimport=*"indianfreelancer.RegisterDao"*%>

<%@pageimport=*"indianfreelancer.Register"*%>

<h4style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;text- decoration:*underline*">Edit profile</h4>

<%

**int** userid=Integer.parseInt((String)session.getAttribute("loginid")); RegisterDao obj2=**new**RegisterDao();

Register obj1=obj2.getUserProfileDetailsById(userid);

String userlogintype1=(String)session.getAttribute("logintype");

**int** logintype1=Integer.parseInt(userlogintype1); String imagepath = obj1.getProfilr\_pic(); **if**(imagepath == **null**)

{

imagepath = ["http://localhost:7080/indianfreelancer/profilepic/user\_default.png"](http://localhost:7080/indianfreelancer/profilepic/user_default.png);

}

%>

<center>

<imgsrc=*"*<%=imagepath %>*"*width=*"200px"*height=*"200px"*style="margin-bottom:

*20px*;" class=*"rounded-circle"*>

<ahref=*"editprofilepic"*class=*"btn"*style="margin-top: *100px*"><iclass=*"fa fa-camera"*> Edit</i></a>

</center>

<%

**if**(logintype1==1)

{

%>

<divstyle="width: *100%*;margin: *0 auto*;text-align: *center*;">

<ahref=*"bidderratinglist"*class=*"btn-success"*style="padding: *8px*;border-radius: *5px*;">View Rating</a></div>

<%**if**(obj1.getAvg\_marks()>0){ %>

<pclass=*"mkshw"*>

<fmt:formatNumbertype=*"number"*groupingUsed=*"false"*value=*"*<%=obj1.getAvg\_marks()

%>*"*/> % Marks Obtained In Quiz Exam

</p>

<%**if**(obj1.getAvg\_marks()>=80)

{ %>

<divclass=*"award"*>

<imgsrc=*"front/images/above80\_award.jpg"*style="height:*150px*;"/>

</div>

<divclass=*"medal"*>

<imgsrc=*"front/images/above80\_medal.png"*style="height:*150px*;"/>

</div>

<%

} %>

{ %>

<%**if**(obj1.getAvg\_marks()>=60 && obj1.getAvg\_marks()<80)

<divclass=*"award"*>

<imgsrc=*"front/images/above60\_award.png"*style="height:*150px*;"/>

</div>

<divclass=*"medal"*>

<imgsrc=*"front/images/above60\_medal.jpg"*style="height:*150px*;"/>

</div>

<%

} %>

<%**if**(obj1.getAvg\_marks()>=30 && obj1.getAvg\_marks()<60)

{ %>

<divclass=*"award"*>

<%

} %>

<%

}

}

%>

<imgsrc=*"front/images/above40\_award.jpg"*style="height:*150px*;"/>

</div>

<divclass=*"medal"*>

<imgsrc=*"front/images/above40\_medal.jpg"*style="height:*150px*;"/>

</div>

<divstyle="clear:*both*"></div>

<divstyle="width:*60%*; margin:*0 auto !important*; margin:*20px*;">

<h5style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;" class=*"commonmsg"*><% String m=request.getParameter("successmsg"); **if**(m!=**null**){ out.print(m); } %></h5>

<spanstyle="color:*green*" class=*"commonmsg"*>

<% String editprofilesuccessmsg=(String)request.getParameter("editprofilesuccessmsg");

**if**(editprofilesuccessmsg!=**null**)

{

out.print(editprofilesuccessmsg); session.setAttribute("loginname",obj1.getName());

}

%>

</span>

<h4style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;">Hello <% String name=(String)session.getAttribute("loginname"); %>

<%= name %>

</h4>

<formclass=*"form-horizontal"*action=*"editprofileaction"*method=*"post"*>

<divclass=*"modal-header"*>

<h5class=*"modal-title"*>Profile Information</h5>

<buttontype=*"button"*class=*"close"*data-dismiss=*"modal"*aria-label=*"Close"*>

<spanaria-hidden=*"true"*>&times;</span>

</button>

</div>

<!--<h3 style="text-decoration:underline; margin:5px; text-align:center; padding-bottom:10px;">Profile Information</h3>

-->

<inputtype=*"hidden"*name=*"userid"*value=*"*<%= userid %>*"*/>

<divclass=*"card"*style="width:*100%*;">

<table>

<tr><td>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email">Name:</label> -->

<divclass=*"col-sm-10"*>

<inputtype=*"text"*class=*"form- control"*placeholder=*"Name\*"*id=*"name"*name=*"username"*value=*"*<%= obj1.getName() %>*"*>

<spanstyle="color:*red*">

<% String nameerr=(String)request.getAttribute("nameerr");

**if**(nameerr!=**null**)

{

out.print(nameerr);

}

%>

</span>

</div>

</div>

</td>

<td>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Email:</label> -->

<divclass=*"col-sm-10"*>

<inputtype=*"text"*name=*"useremail"*class=*"form- control"*placeholder=*"Email"*value=*"*<%=obj1.getEmail() %>*"*readonly=*"readonly"*>

<spanstyle="color:*red*">

<% String emailerr=(String)request.getAttribute("emailerr");

**if**(emailerr!=**null**)

{

out.print(emailerr);

}

%>

<% String emverr=(String)request.getAttribute("emverr");

**if**(emverr!=**null**)

{

out.print(emverr);

}

%>

</span>

</div>

</div>

</td>

</tr>

<tr>

<td>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Phoneno:</label> -->

<divclass=*"col-sm-10"*>

<inputtype=*"text"*name=*"userphoneno"*class=*"form- control"*placeholder=*"Phone Number\*"*value=*"*<%=obj1.getPhoneno() %>*"*>

<spanstyle="color:*red*">

<% String phoneerr=(String)request.getAttribute("phoneerr"); String phderr=(String)request.getAttribute("phderr");

**if**(phoneerr!=**null**)

{

out.print(phoneerr);

}

**if**(phderr!=**null**)

{

out.print(phderr);

}

%>

</span>

</div>

</div>

</td>

<td>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Date-of- birth:</label> -->

<divclass=*"col-sm-10"*>

<inputtype=*"date"*name=*"userdob"*class=*"form-control"*placeholder=*"Date-of- Birth\*"*value=*"*<%=obj1.getDob() %>*"*>

<spanstyle="color:*red*">

<% String doberr=(String)request.getAttribute("doberr");

**if**(doberr!=**null**)

{

out.print(doberr);

}

%>

<% String derr=(String)request.getAttribute("derr");

**if**(derr!=**null**)

{

out.print(derr);

}

%>

<%-- <% String pind6err=(String)request.getAttribute("pind6err");

if(pind6err!=null)

{

out.print(pind6err);

}

%> --%>

</span>

</div>

</div>

</td>

</tr>

<tr>

<tdcolspan=*"2"*>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Address:</label> -->

<divclass=*"col-sm-10"*>

<textarearows=*"2"*cols=*"6"*name=*"useraddress"*class=*"form- control"*placeholder=*"Address\*"*><%=obj1.getAddress() %></textarea>

<spanstyle="color:*red*">

<% String adderr=(String)request.getAttribute("adderr");

**if**(adderr!=**null**)

{

out.print(adderr);

}

%>

</span>

</div>

</div>

</td>

</tr>

<tr>

<td>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Type:</label> -->

<divclass=*"col-sm-10"*>

<selectname=*"usertype"*class=*"form- control"*disabled=*"disabled"*>

<optionvalue=*"0"*<%**if**(obj1.getType()==0){%>selected=*"selected"*<% } %>>I want to Hire</option>

<optionvalue=*"1"*<%**if**(obj1.getType()==1){%>selected=*"selected"*<% } %>>I want to Work</option>

</select>

</div>

</div>

</td>

<td>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Brief introduction:</label> -->

<!--<div class="col-sm-10"> -->

<inputtype=*"text"*name=*"brief\_introduction"*class=*"form- control"*placeholder=*"Bio"*value=*"*<%**if**(obj1.getBrief\_introduction()!=**null**){ out.print(obj1.getBrief\_introduction()); } %>*"*>

<!--</div> -->

</div>

</td>

</tr>

<tr>

<tdcolspan=*"2"*>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="pwd">Summary:</label> -->

<divclass=*"col-sm-10"*>

<textarearows=*"4"*name=*"summary"*class=*"form- control"*placeholder=*"Summary"*><%**if**(obj1.getSummary()!=**null**){ out.print(obj1.getSummary()); } %></textarea>

</div>

</div>

</td>

</tr>

</table>

</div>

<%

String userlogintype=(String)session.getAttribute("logintype");

**int** logintype=Integer.parseInt(userlogintype);

%>

<%

**if**(logintype==1)

{

%>

<h3style="text-decoration:*underline*; margin:*5px*; text- align:*center*">Stripe Payment Gateway Information</h3>

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm-2"*for=*"pwd"*>Stripe PublishableKey:</label>

<divclass=*"col-sm-10"*>

<inputtype=*"text"*name=*"stripe\_publishable\_key"*class=*"form- control"*value=*"*<%**if**(obj1.getStripe\_publishable\_key()!=**null**){ out.print(obj1.getStripe\_publishable\_key()); } %>*"*>

</div>

</div>

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm-2"*for=*"pwd"*>Stripe SecretKey:</label>

<divclass=*"col-sm-10"*>

<inputtype=*"text"*name=*"stripe\_secret\_key"*class=*"form- control"*value=*"*<%**if**(obj1.getStripe\_secret\_key()!=**null**){ out.print(obj1.getStripe\_secret\_key()); } %>*"*>

</div>

</div>

<%

}

%>

<divclass=*"form-group"*>

<divclass=*"col-sm-offset-2 col-sm-10"*>

<buttontype=*"submit"*class=*"btn btn-default"*name=*"submit"*>Submit</button>

</div>

</div>

</form>

</div>

<%@includefile=*"includes/footer.jsp"*%>

</body>

</html>

## Admin add quiz category:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*pageEncoding=*"ISO- 8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Add Quiz Category</title>

</head>

<body>

<bodystyle="background-image: *url('front/images/12.jpg')*;background- size:*cover*;background-position:*top*;box-sizing:*border-box*;">

<%@includefile=*"../frontmodule/includes/header.jsp"*%>

<%@includefile=*"../frontmodule/checklogin.jsp"*%>

<h4style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*blue*;">Add Quiz Category</h4>

<divclass=*"card m"*style="background-color:*rgba(0,0,0,0.1)*;width:*40%*; margin-left:

*30%*;margin-top:*10%*;margin-bottom:*10%*;">

<h5style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;" class=*"commonmsg"*><% String m=(String)request.getAttribute("successmsg"); **if**(m!=**null**){ out.print(m); } %></h5>

<formaction=*"addquizcategory"*method=*"post"*onsubmit="return checkquizform();">

<divclass=*"form-group"*>

<span><labelclass=*"control-label col-sm- 4"*for=*"email"*>Category</label></span>

<span>

<divclass=*"col-sm-8"*>

<inputtype=*"text"*class=*"form- control"*id=*"quizcat"*name=*"quizcat"*placeholder=*"Enter Quiz Category Name"*style="color:*black*";>

<spanid=*"quizcatspanerr"*style="color:*red*"></span>

</span>

</div>

</div>

<divclass=*"form-group"*>

<divclass=*"col-sm-offset-2 col-sm-10"*>

<buttontype=*"submit"*class=*"btn btn- success"*name=*"submit"*style="color:*#fff !important*; margin- top:*10px*;float:*right*;">Submit</button>

</div>

</div>

</form>

</div>

</body>

<%@includefile=*"../frontmodule/includes/footer.jsp"*%>

</html>

<script>

**function** checkquizform()

{

**var** quizcat=$("#quizcat").val();

**if**(quizcat=="")

{

}

</script>

}

**else**

{

}

$("#quizcatspanerr").html("Please give quiz category name");

**returnfalse**;

$("#quizcatspanerr").html("");

**returntrue**;

Admin add quiz question:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*pageEncoding=*"ISO- 8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<style>

*.quizoptncls*{ margin-top:*5px*;

}

*.form-group*{ padding-top: *40px*;

}

</style>

<metacharset=*"ISO-8859-1"*>

<title>Add Quiz Question</title>

</head>

<body>

<bodystyle="background-image: *url('front/images/12.jpg')*;background- size:*cover*;background-position:*top*;box-sizing:*border-box*; background-repeat:*no- repeat*; ">

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"java.util.\*"*%>

<%@pageimport=*"myhibernatepersistent.QuizCategory"*%>

<%@includefile=*"../frontmodule/includes/header.jsp"*%>

<%@includefile=*"../frontmodule/checklogin.jsp"*%>

<divstyle="width:*60%*; margin:*0 auto !important*; margin:*20px*; height:*auto*; overflow:*auto*">

<h5style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;" class=*"commonmsg"*><% String m=(String)request.getAttribute("successmsg"); **if**(m!=**null**){ out.print(m); } %></h5>

<h4style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*blue*; text- decoration: *underline*;">Add Quiz Question</h4>

<%

List<QuizCategory> list=(List<QuizCategory>)request.getAttribute("list");

%>

<formaction=*"addquizquestion"*method=*"post"*onsubmit="return checkquizform();">

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm-4"*for=*"email"*style="color:*green*"> Category</label>

<divclass=*"col-sm-8"*>

<selectclass=*"form- control"*id=*"quizcategory"*name=*"quizcategory"*>

Category---</option>

<optionvalue=*""*style="color:*white*">--Please Select

<c:forEachitems=*"*${ list }*"*var=*"u"*>

<optionvalue=*"*${ u.getId() }*"*>${

u.getCategory\_name() }</option>

</c:forEach>

</select>

<spanid=*"quizcategoryspanerr"*style="color:*red*"></span>

</div>

</div>

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm- 4"*for=*"email"*style="color:*green*">Question</label>

<divclass=*"col-sm-8"*>

<inputtype=*"text"*class=*"form- control"*id=*"quizquestion"*name=*"quizquestion"*>

<spanid=*"quizquestionspanerr"*style="color:*red*"></span>

</div>

</div>

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm- 4"*for=*"email"*style="color:*green*">Option Type</label>

<divclass=*"col-sm-8"*>

<selectname=*"optiontype"*class=*"form-control"*id=*"optiontype"*>

<optionvalue=*"radio"*>Radio</option>

</select>

</div>

</div>

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm- 4"*for=*"email"*style="color:*green*">Question Options</label>

<divclass=*"col-sm-8 qs\_opt\_contanr"*>

<inputtype=*"text"*class=*"form-control quizoptncls"*id=*"quizoptions"*name=*"quizoptions"*style="color: *#800000*" >

<ahref=*"javascript:void(0)"*class=*"answanch"*>Set As Answer</a>

<spanid=*"quizoptionspanerr"*style="color:*red*"></span>

</div>

</div>

<divclass=*"form-group"*>

<divclass=*"col-sm-offset-2 col-sm-10"*>

<buttontype=*"button"*class=*"btn btn-success"*name=*"addmore"*style="margin- bottom:*10px*;color:*#fff !important*" onclick="addmore\_question\_options();">Add More</button>

</div>

</div>

<divclass=*"form-group"*>

<labelclass=*"control-label col-sm- 4"*for=*"email"*style="color:*green*">Answer</label>

<divclass=*"col-sm-8"*>

<inputtype=*"text"*class=*"form-control"*id=*"answer"*name=*"answer"*>

<spanid=*"answerspanerr"*style="color:*red*"></span>

</div>

</div>

<divclass=*"form-group"*>

<divclass=*"col-sm-offset-2 col-sm-10"*>

<buttontype=*"submit"*class=*"btn btn- success"*name=*"submit"*style="color:*#fff !important*; margin-top: *10px*;margin- bottom:*10px*;margin-left:*65*

</div>

</form>

</div>

</body>

%;">Submit</button>

</div>

<%@includefile=*"../frontmodule/includes/footer.jsp"*%>

</html>

<script>

**function** checkquizform()

{

**var** quizquestion=$("#quizquestion").val(); **var** quizcategory=$("#quizcategory").val(); **var** answer=$("#answer").val();

**var** quizoptions=$("#quizoptions").val();

**if**(quizcategory=="")

{

name");

$("#quizcategoryspanerr").html("Please select quiz category

**returnfalse**;

}

**elseif**(quizquestion=="")

{

$("#quizcategoryspanerr").html("");

$("#quizquestionspanerr").html("Please give quiz question name");

**returnfalse**;

}

**elseif**(answer=="")

{

$("#quizquestionspanerr").html("");

$("#answerspanerr").html("Please give answer");

**returnfalse**;

}

**elseif**(quizoptions=="")

{

$("#answerspanerr").html("");

$("#quizoptionspanerr").html("Please give question option");

}

**else**

{

}

}

**returnfalse**;

$("#quizoptionspanerr").html("");

$("#answerspanerr").html("");

$("#quizquestionspanerr").html("");

$("#quizquestionspanerr").html("");

**returntrue**;

**function** addmore\_question\_options()

{

$(".qs\_opt\_contanr").append('<input type="text" class="form-control quizoptncls" name="quizoptions" ><a href="javascript:void(0)" class="answanch">Set As Answer</a>');

}

$(document).on("click",".answanch",**function**(){

**var** answer=$(**this**).prev().val();

$("#answer").val(answer);

})

</script>

## Admin view quiz list:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*pageEncoding=*"ISO- 8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Quiz catagory</title>

</head>

<body>

<bodystyle="background-image: *url('front/images/12.jpg')*;background- size:*cover*;background-position:*top*;box-sizing:*border-box*;">

<style>

**th** {

background: *green*; color:*#fff !important*;

}

</style>

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"java.util.\*"*%>

<%@pageimport=*"myhibernatepersistent.QuizCategory"*%>

<%@includefile=*"../frontmodule/includes/header.jsp"*%>

<%@includefile=*"../frontmodule/checklogin.jsp"*%>

<%

ArrayList<QuizCategory> list=(ArrayList<QuizCategory>)request.getAttribute("list"); Iterator<QuizCategory> iter = list.iterator();

%>

<h3style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*blue*;text- decoration:*underline*">Quiz Category List </h3>

<ahref=*"addquizcategory"*class=*"btn btn-success"*style="margin-left:*45%*;">Quiz Category Add</a>

<p></p>

<divstyle="width:*90%*; margin:*0 auto !important*; margin:*20px*;">

<pstyle="color:*green*;margin-bottom:*20px*; text-align:*center*;font-size:*18px*;" >

<%**if**(list.isEmpty()) { out.print( "There is no Quiz Category." ); } %>

</p>

<p></p>

<table>

<tr><th>Id<th>Quiz Category</th><th>Action</th></tr>

<%

**while**(iter.hasNext()){

QuizCategory q = iter.next();

%>

<trstyle="color:*#434241*;"><td><%= q.getId()%></td><td><%=q.getCategory\_name()

%></td><td><ahref=*"viewquizcatquestion?catid=*<%= q.getId()%>*"*> Questions & Options</a></td></tr>

<%

}%>

</table>

</div>

<%@includefile=*"../frontmodule/includes/footer.jsp"*%>

</body>

</html>

## > Hirer Post Project:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Post Project</title>

</head>

<bodystyle="background-image: *url('front/images/3.jpg')*">

<body>

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"indianfreelancer.Skills"*%>

<%@pageimport=*"indianfreelancer.SkillsDao"*%>

<%@includefile=*"includes/header.jsp"*%>

<%-- <%@ include file="includes/banner.jsp" %>--%>

<%@includefile=*"checklogin.jsp"*%>

<%@pageimport=*"java.util.List"*%>

<h3style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*#0b8de3*;text-decoration:*underline*">Post Your Project </h3>

<divstyle="width:*60%*; margin:*0 auto !important*; margin:*20px*;">

<pstyle="color:*green*;margin-bottom:*20px*" class=*"commonmsg"*>

<% String prosuccessmsg=(String)request.getAttribute("prosuccessmsg");

**if**(prosuccessmsg!=**null**)

{

out.print(prosuccessmsg);

}

%>

</p>

<formclass=*"form-horizontal"*action=*"projectaddaction"*method=*"post"*>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email">Project name

:</label> -->

<divclass=*"col-sm-10"*>

<inputtype=*"text"*class=*"form-*

*control"*id=*"title"*name=*"title"*value=*""*placeholder=*"Project Name\*"*>

<spanstyle="color:*red*">

<% String titleerr=(String)request.getAttribute("titleerr");

**if**(titleerr!=**null**)

{

out.print(titleerr);

}

%>

</div>

</span>

</div>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email"> Description

:</label> -->

<divclass=*"col-sm-10"*>

<textarearows=*"10"*cols=*"10"*name=*"description"*class=*"form- control"*placeholder=*"Description\*"*></textarea>

<spanstyle="color:*red*">

<% String descerr=(String)request.getAttribute("descerr");

**if**(descerr!=**null**)

{

out.print(descerr);

}

%>

</div>

</span>

</div>

<%

List<Skills> list =SkillsDao.getAllSkills();

request.setAttribute("list",list);

%>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email"> Skills :</label>

-->

</div>

<divclass=*"col-sm-10"*>

<selectmultiple=*"multiple"*style="height:*130px*" name=*"projectskills"*>

<c:forEachitems=*"*${list}*"*var=*"u"*>

<optionvalue=*"*${ u.getId() }*"*>${ u.getSkills\_name() }</option>

</c:forEach>

</select>

</div>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email"> How do you want to pay? </label> -->

<divclass=*"col-sm-10"*>

<selectclass=*"form- control"*id=*"payment\_type"*name=*"payment\_type"*placeholder=*"Payment Type\*"*required>

<optionvalue=*"0"*>Pay fixed price</option>

<optionvalue=*"1"*>Pay by hour</option>

</select>

</div>

</div>

<divclass=*"form-group"*id=*"paymenttype\_container"*>

<!--<label class="control-label col-sm-2" for="email"></label> --

>

<divclass=*"col-sm-10"*> Rupees:<inputtype=*"number"*name=*"minbudget"*class=*"form-*

*control"*placeholder=*"Minimum budget in Rupees\*"*style="margin-bottom:*10px*;">

<spanstyle="color:*red*">

<% String minpriceerr=(String)request.getAttribute("minpriceerr");

**if**(minpriceerr!=**null**)

{

out.print(minpriceerr);

}

%>

</span> Rupees:<inputtype=*"number"*name=*"maxbudget"*class=*"form-*

*control"*placeholder=*"Maximum budget in Rupees\*"*>

<spanstyle="color:*red*">

<% String maximumpriceerr=(String)request.getAttribute("maximumpriceerr");

**if**(maximumpriceerr!=**null**)

{

out.print(maximumpriceerr);

}

%>

</span>

<spanstyle="color:*red*">

<% String maximum\_min\_priceerr=(String)request.getAttribute("maximum\_min\_priceerr");

**if**(maximum\_min\_priceerr!=**null**)

{

out.print(maximum\_min\_priceerr);

}

%>

</div>

</div>

</span>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email"> When you will start project</label> -->

<divclass=*"col-sm-10"*>

<selectclass=*"form- control"*id=*"project\_start"*name=*"project\_start"*placeholder=*"Project Start Date\*"*>

<option>Between 1 Week</option>

<option> Between 2 Week</option>

<option> Between 3 Week</option>

<option>Between 4 Week</option>

<option>Between 5 Week</option>

<option>More than 5 Week</option>

<option>Immediately</option>

</select>

</div>

</div>

<divclass=*"form-group"*>

<!--<label class="control-label col-sm-2" for="email"> Project type</label> -->

<divclass=*"col-sm-10"*>

<selectclass=*"form- control"*name=*"project\_type"*placeholder=*"Project Type\*"*>

<optionvalue=*"0"*>Development</option>

<optionvalue=*"1"*>Maintenance</option>

</select>

</div>

</div>

<divclass=*"form-group"*>

<divclass=*"col-sm-offset-2 col-sm-10"*>

<buttontype=*"submit"*class=*"btn btn-default"*name=*"submit"*>Submit</button>

</div>

</div>

</form>

</div>

<%@includefile=*"includes/footer.jsp"*%>

</body>

</html>

## Hirer Leads Project:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Hirer Leads Projects</title>

</head>

<bodystyle="background-image: *url('front/images/3.jpg')*">

<body>

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"indianfreelancer.Project"*%>

<%@pageimport=*"indianfreelancer.ProjectDao"*%>

<%@pageimport=*"java.util.\*"*%>

<%@includefile=*"includes/header.jsp"*%>

<%-- <%@ include file="includes/banner.jsp" %>--%>

<%@includefile=*"checklogin.jsp"*%>

<%

List<Project> list=ProjectDao.myLeadsList(); request.setAttribute("list",list);

%>

<h3style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*#fc03a9*; text-decoration:*underline*">My Leads </h3>

<divstyle="width:*90%*; margin:*0 auto !important*; margin:*20px*;">

<pstyle="color:*white*;margin-bottom:*20px*; text-align:*center*;font-size:*18px*;" >

<%**if**(list.isEmpty()) { out.print( "Currently You have no project. Please post your project." ); } %>

</p>

<c:forEachitems=*"*${ list }*"*var=*"u"*>

<divclass=*"card"*style="background-color:*rgba(0,0,0,0.5)*;width: *24rem*;float: *left*;margin-left: *20px*;">

<divclass=*"card-wrapper text-center"*style="border: *2px solid #ccc*;border- radius: *5px*;border-color:*#0b8de3*;padding: *10px*;">

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Project Name :</span> ${ u.getTitle() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Project Description

:</span> ${ u.getDescription() }</p>

<pstyle="color:*white*">

<spanstyle="font-weight:*bold*">Payment Type :</span>

<c:iftest=*"*${u.getPayment\_type()==0 }*"*>

<c:outvalue=*"Pay Fixed Price"*/>

</c:if>

<c:iftest=*"*${u.getPayment\_type()==1 }*"*>

<c:outvalue=*"Pay By Hour"*/>

</c:if>

<spanstyle="font-weight:*bold*">Min Price :</span>Rs ${ u.getMin\_price() }

<spanstyle="font-weight:*bold*"> Max Price :</span>Rs ${ u.getMax\_price() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Project Will Start

:</span> ${ u.getStartdate\_of\_project() }

<spanstyle="font-weight:*bold*"> Project Type :</span>

<c:iftest=*"*${u.getProject\_type()==0 }*"*>

<c:outvalue=*"Development"*/>

</c:if>

<c:iftest=*"*${u.getProject\_type()==1 }*"*>

<c:outvalue=*"Maintenance"*/>

</c:if>

</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Post Date :</span> ${ u.getCreated\_at() }

<spanstyle="font-weight:*bold*">Expiry Date :</span> ${ u.getExpiry\_date() }</p>

<pstyle="color:*white*"><spanstyle="font-weight:*bold*">Required Skills :</span>

${ u.getSkills\_name() }</p>

<pstyle="margin-top: *10px*;">

<ahref=*"placebid?pinfo=*${ u.getId() }*&pmtype=*${u.getPayment\_type()}*"*class=*"btn btn-outline-warning"*style="padding: *8px*;border-radius: *5px*;"> Bid on This Project</a>

<%-- <a href="sendmessage?rid=${ u.getUser\_id() }" class="btn-primary" style="padding: 8px;border-radius: 5px;"> Send Message</a></p> --%>

</div>

</div>

</c:forEach>

</div>

<divstyle="clear:*left*;margin-top: *50px*;">

%@includefile=*"includes/footer.jsp"*%

</div>

</body>

</html>

## Hirer Rating List:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Rating</title>

<style>

*.rateicon*{ height: *40px*;

}

*.rateicon1*{

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"java.util.\*"*%>

<%@pageimport=*"myhibernatepersistent.Ratingdetails"*%>

<%@includefile=*"includes/header.jsp"*%>

<%-- <%@ include file="includes/banner.jsp" %>--%>

<%@includefile=*"checklogin.jsp"*%>

<%

List<Ratingdetails> list=(List<Ratingdetails>)request.getAttribute("list"); request.setAttribute("list",list);

%>

<h3style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;text- decoration:*underline*">View Rating And Review </h3>

<divstyle="width:*90%*; margin:*0 auto !important*; margin:*20px*;">

<pstyle="color:*green*;margin-bottom:*20px*; text-align:*center*;font-size:*18px*;" >

<%**if**(list.isEmpty()) { out.print( "There is no rating and review." ); } %>

</p>

<%**int** k=0; %>

<c:forEachitems=*"*${ list }*"*var=*"u"*>

<% k++; %>

<%**if**(k==1) { %>

<h5style="font-weight:*bold*;margin:*10px*;color:*green*"><span>Hired Person :</span> ${ u.getBiddername() }

~ ${ u.getAvg\_rating\_no() }

<c:iftest=*"*${u.getAvg\_rating\_no()>=1 && u.getAvg\_rating\_no()< 1.5}*"*>

<%

**for**(**int** i=1;i<=1;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()>=1.5 && u.getAvg\_rating\_no()< 2}*"*>

<%

**for**(**int** i=1;i<=1;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

<imgsrc=*"front/images/Star-Half-Full.png"*class=*"rateicon1"*/>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()>=2 && u.getAvg\_rating\_no()< 2.5}*"*>

<%

**for**(**int** i=1;i<=2;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()>=2.5 && u.getAvg\_rating\_no()< 3}*"*>

<%

**for**(**int** i=1;i<=2;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

<imgsrc=*"front/images/Star-Half-Full.png"*class=*"rateicon1"*/>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()>=3 && u.getAvg\_rating\_no()< 3.5}*"*>

<%

**for**(**int** i=1;i<=3;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()>=3.5 && u.getAvg\_rating\_no()< 4}*"*>

<%

**for**(**int** i=1;i<=3;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

<imgsrc=*"front/images/Star-Half-Full.png"*class=*"rateicon1"*/>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()>=4 && u.getAvg\_rating\_no()< 4.5}*"*>

<%

**for**(**int** i=1;i<=4;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${ u.getAvg\_rating\_no()>=4.5 && u.getAvg\_rating\_no()< 5 }*"*>

<%

**for**(**int** i=1;i<=4;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

<imgsrc=*"front/images/Star-Half-Full.png"*class=*"rateicon1"*/>

</c:if>

<c:iftest=*"*${u.getAvg\_rating\_no()==5}*"*>

<%

**for**(**int** i=1;i<=5;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

</h5>

<% } %>

<divclass=*"row"*style="background-color: *#f5dada*;min-height: *128px*;margin: *10px*;width: *100%*;">

<divclass=*"col-sm-12 biddercontent"*style="border: *2px solid #ccc*;border-radius: *5px*;border-color: *green*;padding: *10px*;">

<pstyle="text-transform: *capitalize*;"><spanstyle="font-weight:*bold*">Project Poster

:</span> ${ u.getPostername() } </p>

<pstyle="font-weight:*bold*"><spanstyle="font-weight:*bold*;">Review

:</span><spanstyle="color:*#125ade*;">${ u.getReview() } </span></p>

<p><spanstyle="font-weight:*bold*">Rating :</span>

<c:iftest=*"*${u.getRating()==4 }*"*>

<%

**for**(**int** i=1;i<=4;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getRating()==5 }*"*>

<%

**for**(**int** i=1;i<=5;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getRating()==3 }*"*>

<%

**for**(**int** i=1;i<=3;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getRating()==2 }*"*>

<%

**for**(**int** i=1;i<=2;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

<c:iftest=*"*${u.getRating()==1 }*"*>

<%

**for**(**int** i=1;i<=1;i++){

%>

<imgsrc=*"front/images/Actions-rating-icon.png"*class=*"rateicon"*/>

<%

}

%>

</c:if>

</p>

<pstyle="font-weight:*bold*"><spanstyle="font-weight:*bold*;">Created Date

:</span><spanstyle="color:*#125ade*;">${ u.getCreated\_date() } </span></p>

</div>

</div>

</c:forEach>

</div>

<%@includefile=*"includes/footer.jsp"*%>

</body>

</html>

## Bidder Online Quiz Exam:

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*pageEncoding=*"ISO- 8859-1"*%>

<!DOCTYPEhtml>

<html>

<head>

<style>

<**title**>**OnlineQuizExam**2</**title**>

*#timer*{

width: *10%*; height: *10px*;

font-family: *sans-serif*; font-weight: *bold*; color: *black*;

font-size: *21px*; margin-bottom: *23px*; text-align: *center*; margin: *0 auto*; padding-bottom: *10px*;

}

*#timelft*{

width: *10%*; height: *10px*;

font-family: *sans-serif*; font-weight: *bold*; color: *black*;

font-size: *21px*; margin-bottom: *23px*; text-align: *center*; margin: *0 auto*; padding-bottom: *10px*; margin-bottom:*15px*;

}

</style>

<metacharset=*"ISO-8859-1"*>

<title>Online Quiz Exam</title>

<%

ArrayList<QuizQuestion> list=(ArrayList<QuizQuestion>)request.getAttribute("list"); Iterator<QuizQuestion> iter = list.iterator();

%>

<%**if**(!list.isEmpty()) { %>

<script>

**function** formatTime(seconds) {

**var** h = Math.floor(seconds / 3600),

m = Math.floor(seconds / 60) % 60, s = seconds % 60;

**if** (h < 10) h = "0" + h;

**if** (m < 10) m = "0" + m;

**if** (s < 10) s = "0" + s;

**return** h + ":" + m + ":" + s;

}

//var count = 62;

**var** count = 300;// set 5 min

**var** counter = setInterval(timer, 1000);

**function** timer() { count--;

**if** (count < 0)

{

document.quizform.submit();

**return** clearInterval(counter);

}

document.getElementById('timer').innerHTML = formatTime(count);

}

</script>

<% } %>

</head>

<body>

<style> **input**[type=radio] { border: *0px*; width: *100%*; height: *2em*;

}

**th** {

background: *green*; color:*#fff !important*;

}

*.qustcls*{

font-weight: *bold*; font-family: *sans-serif*; font-size: *18px*;

text-transform: *capitalize*;

}

*.maincontain*{

padding: *10px*; margin: *10px*;

border: *2px solid #ccc*;

}

</style>

<%@tagliburi=*"*[*http://java.sun.com/jsp/jstl/core*](http://java.sun.com/jsp/jstl/core)*"*prefix=*"c"*%>

<%@pageimport=*"java.util.\*"*%>

<%@pageimport=*"myhibernatepersistent.QuizQuestion"*%>

<%@pageimport=*"myhibernatepersistent.QuizOption"*%>

<%@pageimport=*"myhibernatedao.QuizCategoryDao"*%>

<%@includefile=*"../frontmodule/includes/header.jsp"*%>

<%-- <%@ include file="../frontmodule/includes/banner.jsp" %>--%>

<%@includefile=*"../frontmodule/checklogin.jsp"*%>

<%**if**(!list.isEmpty()) { %>

<h3style="text-transform:*capitalize*; margin:*20px*;text-align:*center*; color:*green*;text- decoration:*underline*">Online Quiz Test : Number of questions : 10 | Time : 5 minutes

</h3>

<divid=*"timelft"*>Time Left</div>

<divid=*"timer"*></div>

<% } %>

<formaction=*"userquizexamsubmitfinal"*method=*"post"*name=*"quizform"*>

<divstyle="width:*60%*; margin:*0 auto !important*; margin:*20px*;" class=*"topcontain"*>

<pstyle="color:*green*;margin-bottom:*20px*; text-align:*center*;font-size:*18px*;" >

<%**if**(list.isEmpty()) { out.print( "There is no Quiz Questions." ); } %>

</p>

<%

**int** i=0;

**while**(iter.hasNext()){ i++;

QuizQuestion q = iter.next();

**int** question\_id=q.getId();

**int** cat\_id=q.getQuizcategory\_id(); ArrayList<QuizOption>

list1=QuizCategoryDao.user\_quiz\_options\_list(question\_id); Iterator<QuizOption> iter1 = list1.iterator(); **if**(i==1){

%>

<inputtype=*"hidden"*name=*"catid"*value=*"*<%=cat\_id %>*"*/>

<%

}

%>

<divclass=*"maincontain"*<%**if**(i>1){ %>style="display:*none*" <% } %>>

<pclass=*"qustcls"*><%=i %> .<%=q.getQuestion() %></p>

<inputtype=*"hidden"*name=*"question\_id*<%=i %>*"*value=*"*<%=question\_id %>*"*/>

<inputtype=*"hidden"*name=*"question\_value*<%=i %>*"*value=*"*<%=q.getQuestion() %>*"*/>

<%

**int** j=0;

**while**(iter1.hasNext()){ j++;

QuizOption op = iter1.next();

%>

<p>

<divclass=*"col-8 col-12-small"*>

<inputtype=*"radio"*id=*"radio-beta*<%=i%><%=j %>*"*name=*"option*<%=i

%>*"*value=*"*<%=op.getOption() %>*"*>

<labelfor=*"radio-beta*<%=i%><%=j %>*"*><%=op.getOption() %></label>

</div>

</p>

<%

}

%>

<inputtype=*"button"*name=*"nextbutton"*class=*"nextbutton btn btn-primary"*value=*"Next"*/>

</div>

<%

}%>

</div>

<inputtype=*"hidden"*name=*"hdvalue"*value=*"*<%=i %>*"*/>

</form>

<%@includefile=*"../frontmodule/includes/footer.jsp"*%>

</body>

</html>

<script>

$(document).on("click",".nextbutton",**function**(){

$(**this**).closest(".maincontain").hide();

$(**this**).closest(".maincontain").next().show();

**if**($(**this**).closest(".maincontain").next().next().length==0)

{

$(".nextbutton").hide();

$(".topcontain").append('<input type="submit" name="submit" class="nextbutton btn btn-success" value="Submit" style="margin-bottom:10px;"/>');

}

})

</script> Bidder

### Appendix II

#### Home

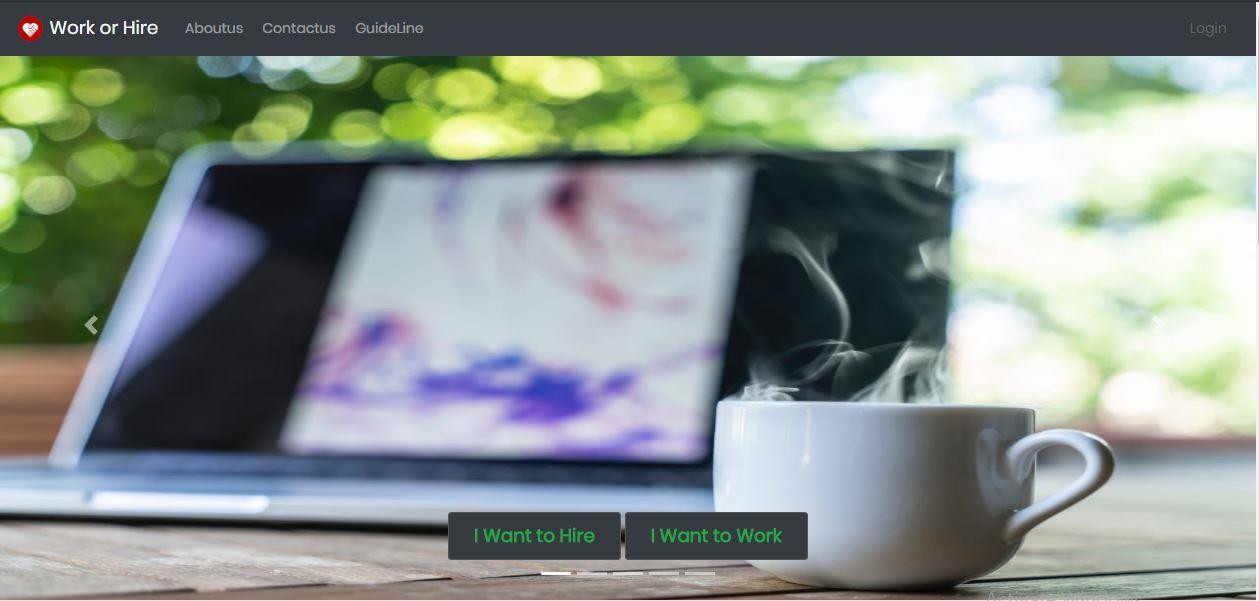


IMAGE 31

#### Registration

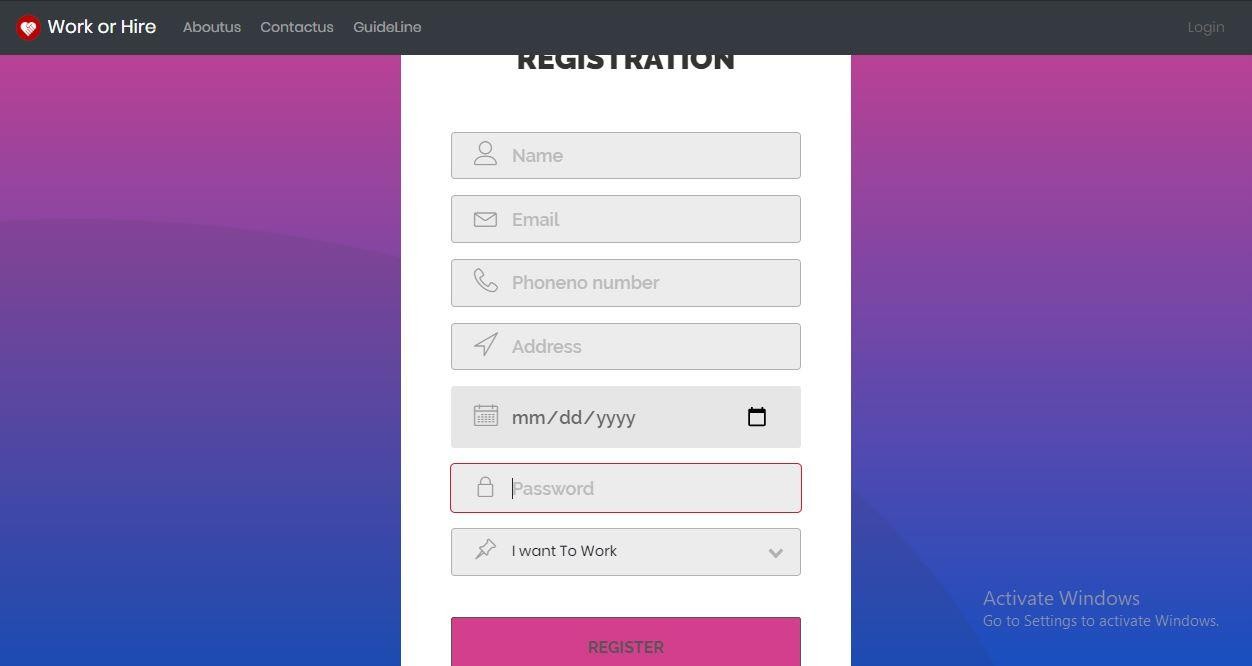


IMAGE 32

#### Log In

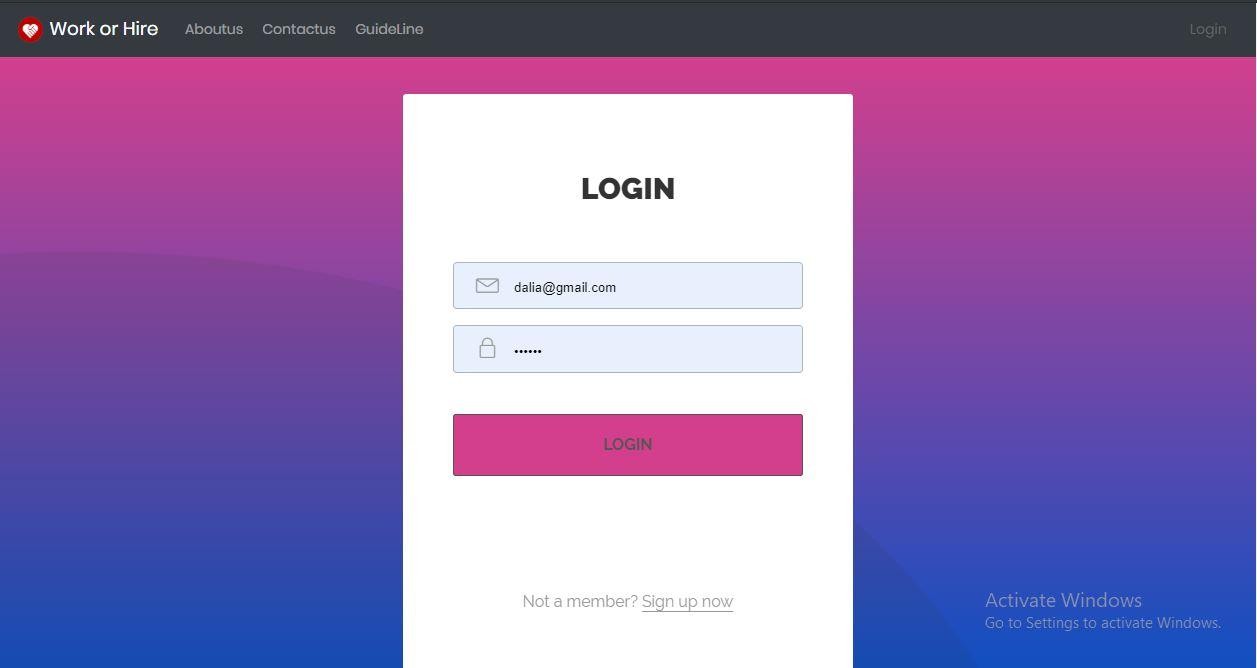


IMAGE 33

#### Edit Profile

****

IMAGE 34

#### Quiz Category

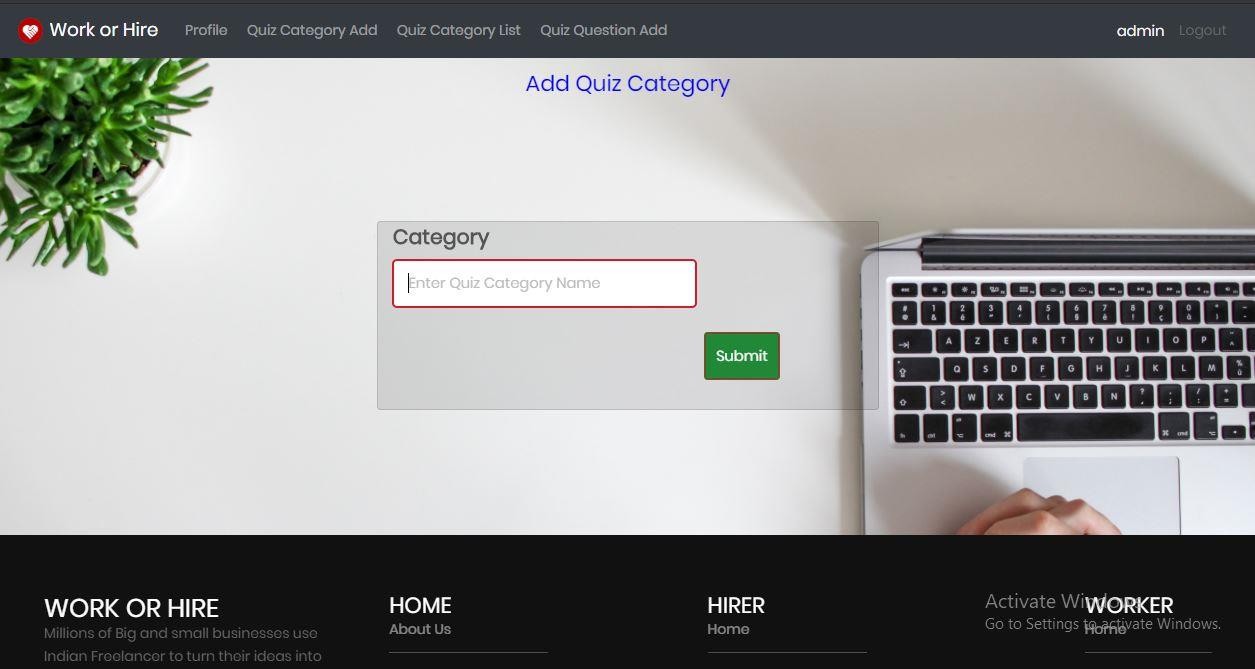


IMAGE 35

#### Add quiz Category

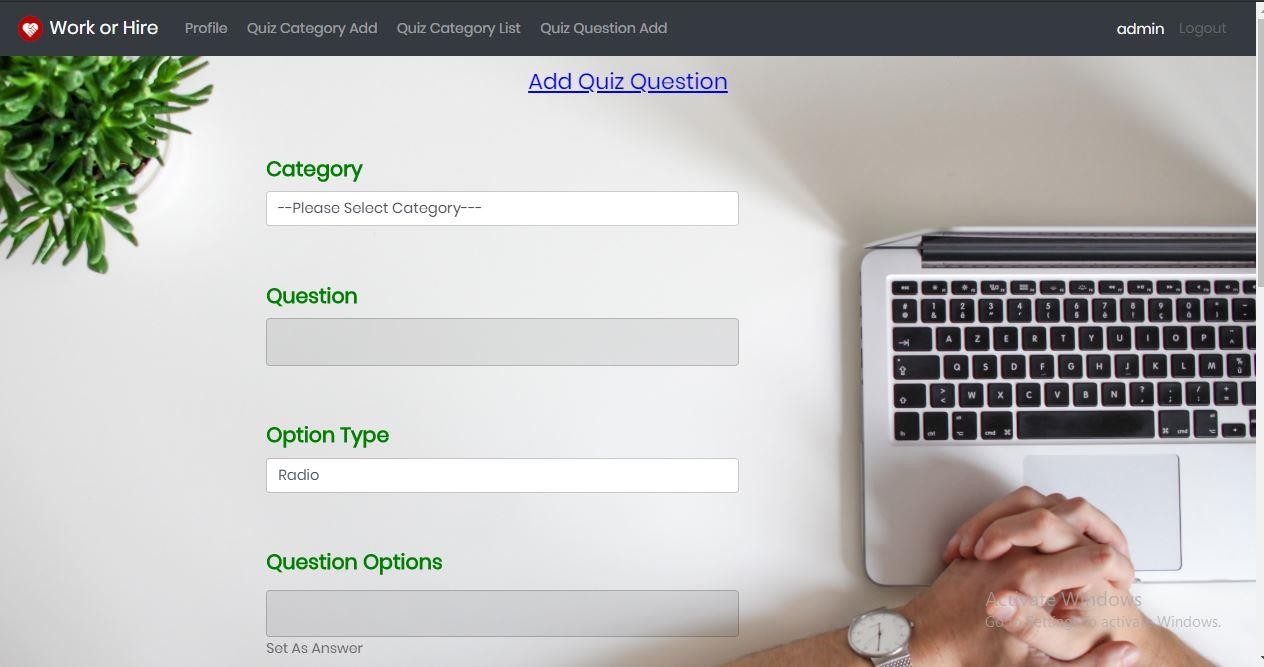


IMAGE 36

#### Quiz category list

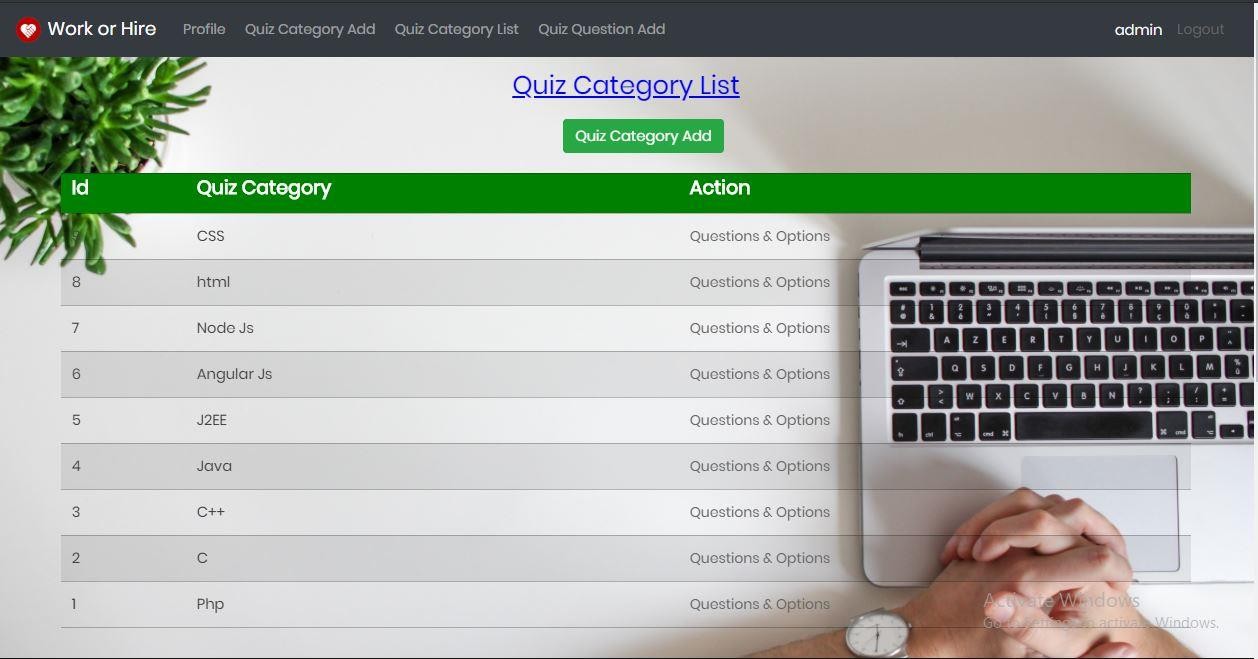


IMAGE 37

#### Post project

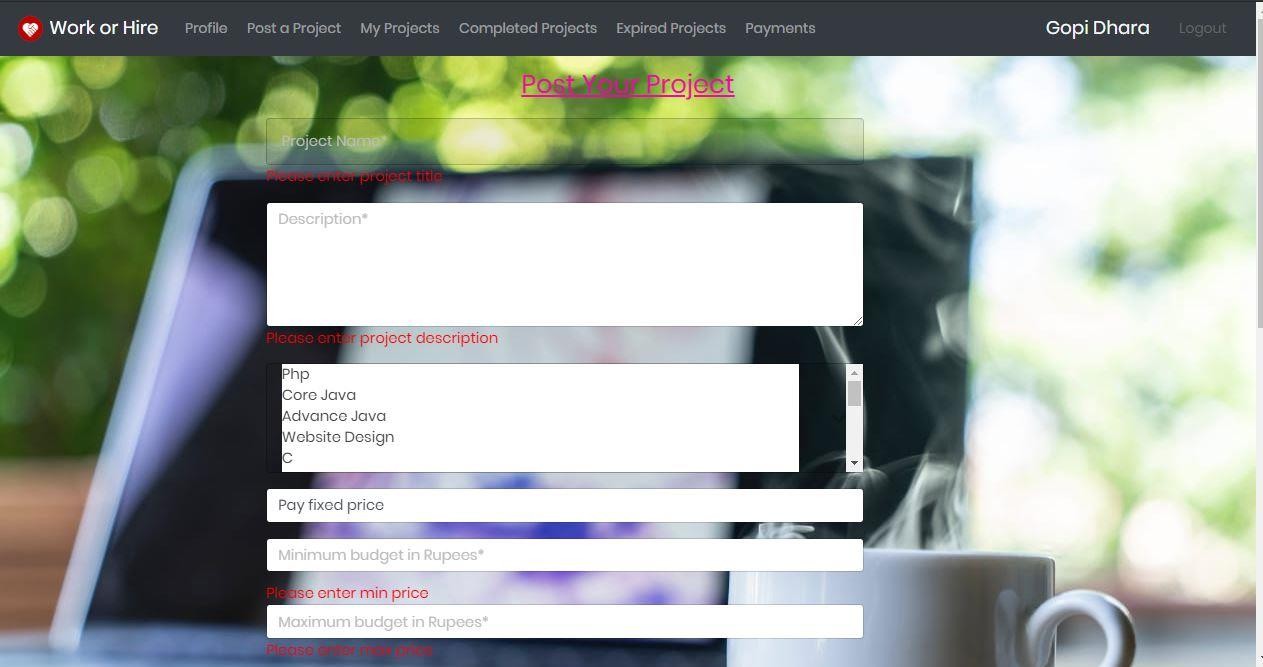


IMAGE 38

#### My leads project

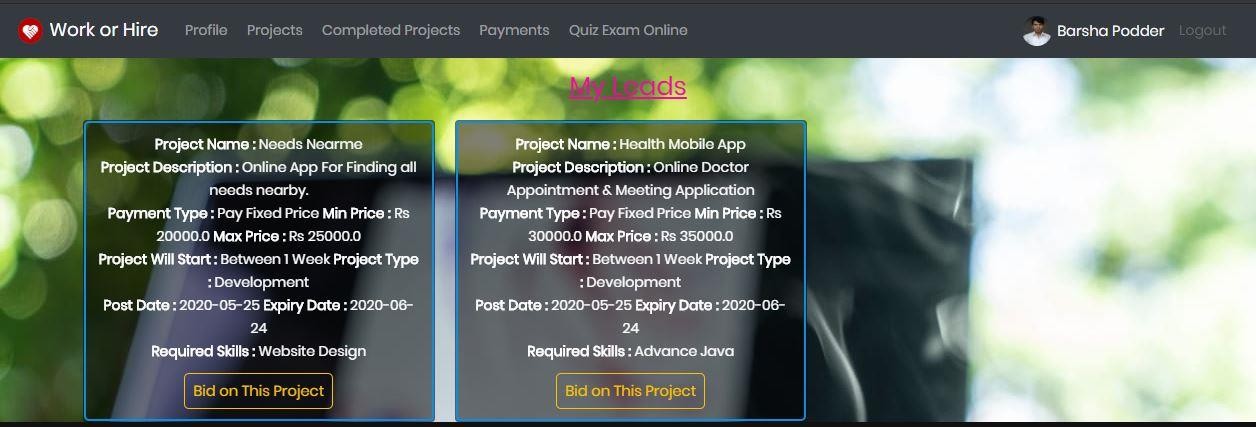


IMAGE 39

### Appendix III

|  |  |
| --- | --- |
| Validator names | The field checks whether each field is filled or not. No important field can  be blank. |
| Regular Expression Validator | The field validator checks pattern like  ‘@gmail.com’ should be in email-id |
| Compare Validator | Compares password and retype password for matching values. |

TABLE 11