**CHAPTER 1**

**INTRODUCTION**

**1.1 Overview/Abstract**

**1.1.1 What is Freelancer**

A freelancer or freelance worker, is a term commonly used for a person

Who is self – employed and is not necessarily committed to a particular

Employer long – term. Freelance labor to clients; others work independently

Or use professional associates or websites to get work.

While the term “Independent contractor” would be used in a higher

Register of English to designate the tax and employment classes of

This type of worker, the term freelancing is common in culture and

Creative industries and this term specifically motions to participation

Therein.

Fields, professions, and industries where freelancing is predominant

Include; music, writing, acting, computer programming, web design

Graphic design, translating and illustrating, films and video produc

Tion and other forms of piece work which some cultural theorists

Consider as central to cognitive – cultural economy.

### 1.2 How does it works?

**1. Clients:**

A Client can post project(s) for free. Then after client will automatically

Begin to receive bids from freelancers (Developers). Alternatively, he can

Browse through the talent available on size, and make a direct offer a

Freelancer instead.

* Choose the perfect freelancer.
* Browse freelancer profiles.
* Chat in real time.
* Compare proposals and select the best one.
* Award the project freelancer and freelancer goes to work.
* Pay when you’re satisified!

### 2. Developer:

A Developer has to complete his profile and has to select skills and

Expertise, should upload a professional profile photo.

* Browse jobs that suits to skills, expertise, price, and schedule
* Put his best foot forward and write the best pitch possible,

Read the project and let the clients know you understand

Their brief.

* Get awarded and earn.

Get ready to work once you get hired. Deliver high quality

Work and earn the agreed amount ,

# TABLE OF CONTENTS

### CHAPTERS PAGE NO.

### 1.INTRODUCTION 01

### 2. SYSTEM REQUIREMENTS 03

### 2.1

* HARDWARE REQUIREMENTS **04**
* SOFTWARE REQUIREMENTS  **05**
* SOFTWARE INTERFACE **06**

2.2 SOFTWARE DESCRIPTION **06**

### 

### 3. SYSTEM ANALYSIS AND 07 DESIGN

3.1 SYSTEM ANALYSIS

3.2 HIGH LEVEL DESIGN

3.2.1 SYSTEM ARCHITECTURE **08**

3.2.2 DATA FLOW DIAGRAM **09**  
 3.2.3 MAJOR MODULES  **12**

### 

3.3 LOW LEVEL DESIGN

3.3.1 SEQUENCE DIAGRAM **13**

3.4 ER – DIAGRAM **15**

3.5 SCHEMA DIAGRAM **16**

3.6 DATABSE TABLE STRUCTURE **17**

### 4. DATABASE TECHNIQUES

### AND RESULTS

4.1 STORED PROCEDURE **18**

4.2 TRIGGER **19**

4.3 TESTING **20**

### 4.2.1 TYPES OF TESTING

4.2.1 UNIT TESTING **21**

4.2.2 INTEGRATED TESTING **21**

4.2.3 SYSTEM TESTING  **21**

4.2.4 TEST CASE **22**

4.2.5 UNIT TESTING **22**

4.2.6 TEST CASES **23**

**5. IMPLEMENTATION**

5.2 SNAPSHOTS **24**

### CONCLUSION 31 REFRENCES 32

**CHAPTER 2**

**SOFTWARE REQUIREMENTSPECIFICATIONS**

**2.1 System Requirements**

•**Hardware Requirements:**

1. PROCESSOR:Pentium 4 and higher (1.2GHZ and higher)
2. RAM:4GB and higher
3. HARD DISK:20GB or higher.
4. Monitor: 15”CRT or LCD monitor
5. Keyboard: normal
6. Mouse: compatible mouse

•**Software Requirements:**

* Operating system: Windows XP/7.

• Coding Language: HTML, PHP, CSS.

• Database : XAAMP

**2.2 SOFTWARE INTERFACE:**

1. Front End Client - The online interface is built using CSS, PHP and HTML.
2. Back End – MySQL database(xampp)

**2.2.1 Software description**

**XAMPP**  is a free and open source cross-platform web server solution stack package developed by Apache Friends,consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.

|  |  |
| --- | --- |
| Developers | Apache Friends |
| Repository | www.apachefriends.org |
| Written in | Various languages |
| Operating System | Cross-platform; Linux; Windows; Solaris; macOS |

**MySQL (XAMPP):**

Is an open source relational database management system (RDBMS) which is integrated in the xamppwebserver.SQL (Structured Query Language) is a standardized programming language used for managing relational databases and performing various operations on the data in them. Initially created in the 1970s, SQL is regularly used by database administrators, as well as by developers writing data integration scripts and data analysts looking to set up and run analytical queries.SQL deviates in several ways from its theoretical foundation, the relational model and its tuple calculus. In that model, a table is a set of tuples, while in SQL, tables and query results are lists of rows: the same row may occur multiple times, and the order of rows can be employed in queries.

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 System analysis**

System analysis is a detailed study of the various operations performed by a system and their relationships within and outside the system. It is a systematic technique that defines goals and objectives. The goal of system development is to deliver the system in line with the user’s requirement, and analysis of the system plays an important role. One of the main aspects of analysis is defining the boundaries of the system. System study has been conducted with the following objectives:

(1)Identify user’s need.

(2) Evaluate the system concept for feasibility.

(3)Perform economical and technical analysis.

(4)Allocate functions to hardware, software, people, database, etc.

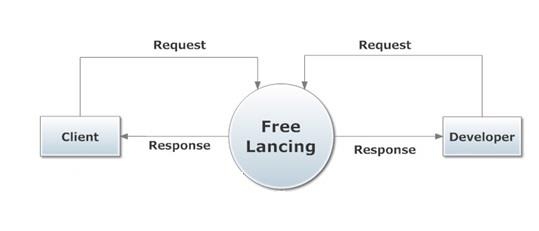
The various tools of structured analysis are**:**

* Entity-Relationship Diagram
* Data Flow Diagram
* Sequence Diagram
* Data flow Diagram
* Major Modules

**High Level Design**

**3.2.1 System Architecture**

System architecture is the conceptual model that defines the structure, behavior, and views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationship (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work to get implement the overall system.

The figure depicts the system architecture of our project. The farmer requests for the service from the server through the android phone . The admin at the server end receives the request . The request is stored in the database .the processed request sent to farmer’s phone in the form of text. Notifications are also sent via mail. After the completion of harvesting of crops farmer uploads the picture of the product which can be viewed by the client.

**Figure 3.2.1**: System Architecture

### 3.2.2 MAJOR MODULES

**CLIENT MODULE:**

**Post a project:** Clients are always free to post their project. They automatically begin to receive  
 bids from our freelancers. Alternatively, they can browse through the talent available on  
 our site, and make a direct offer to a freelancer instead

**Choose the perfect freelancer:**•Browse freelancer profiles  
•Chat in real-time  
•Compare proposals and select the best one  
•Award your project and your freelancer goes to work

**Pay when you are satisfied:** Pay safely using chat attribute by exchanging their accounts.

**DEVELOPER MODULE:  
 Complete your profile:**•Select your skills and expertise  
•Upload a professional profile photo  
•Go through the Verification Center checklist

**Browse jobs that suit your skills, expertise, price, and schedule:**

We have jobs available for all skills. Maximize your job opportunities by  
 optimizing your filters. Save your search and get alerted when relevant jobs are available.

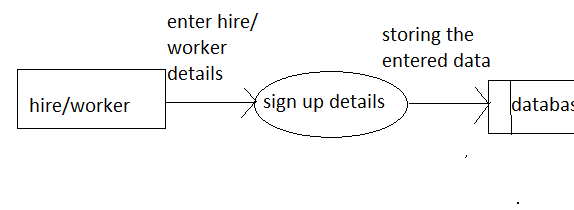
**Write your best bid:** Put your best foot forward and write the best pitch possible. Read the project and  
 let the clients know you understand their brief. Tell them why you're the best person for  
 this job. Writing a new brief for each project is more effective than using the same one!

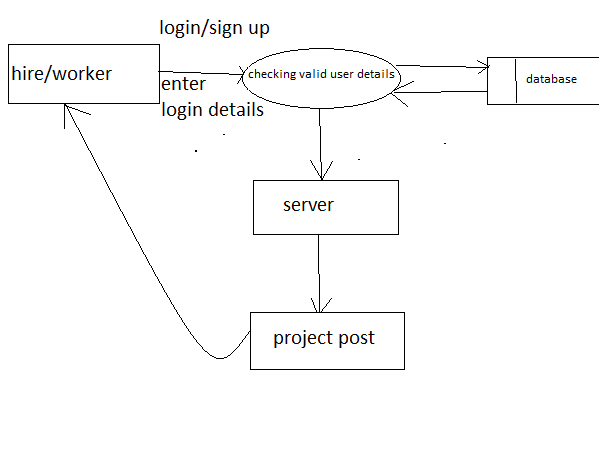
**Get awarded and earn:** Get ready to work once you get hired. Deliver high quality work and earn the  
 agreed amount

**3.2.2 Data flow diagram (DFD)**

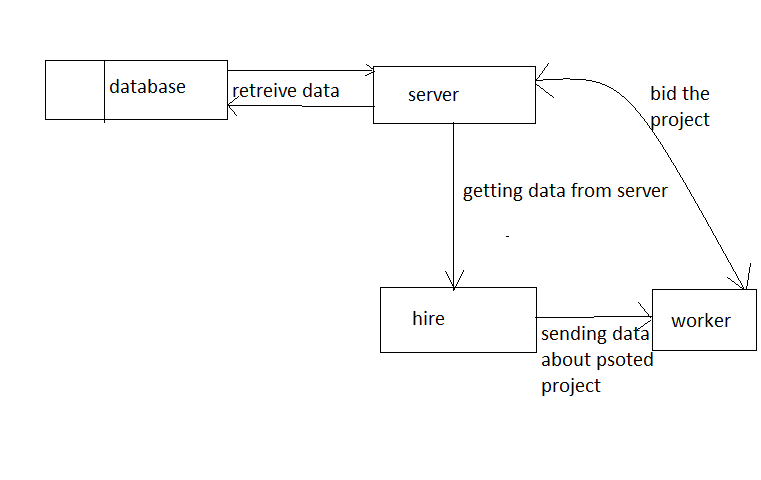
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

It is common practice to draw the context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram the system's interactions with the outside world are modeled purely in terms of data flows across the system bound.

**Figure 4.2.2(a) :** Level 1 DFD



**Figure 4.2.2(b**) : Level 2 DFD



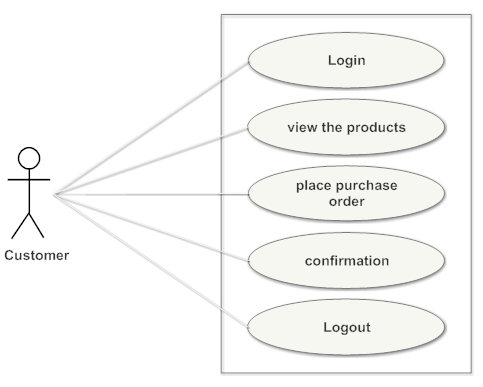
**Figure 4.2.2(c**) : Level 3 DFD

**3.3 Low Level Design**

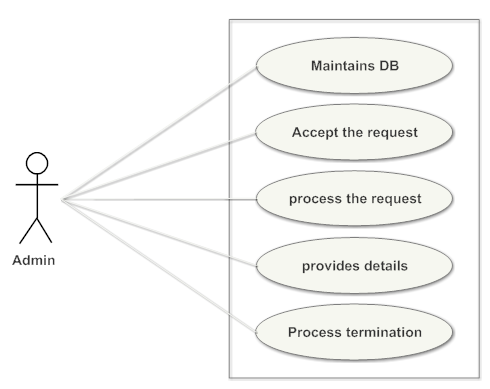
**3.3.1 Use case Diagram**

Use-case diagram is a coherent piece of functionality that a system can provide by interacting with actors. In our system, all the modules are going to interact with one or the other actors (different actors could be, registered users, new users, database admin). The use case technique is used to capture a system behavioral by dealing scenario drives threads through the functional requirements. The components in a use case diagram include:

* **Use cases:** A use case describes a sequence of actions that provides something of measurable value to an actor and is drawn as a horizontal ellipse.
* **Actors:** An actor is a person, organization, or an external system that plays a role in one or more interactions with the system.
* **Associations:** Associations are shown between actors and use cases by drawing a solid line between them.

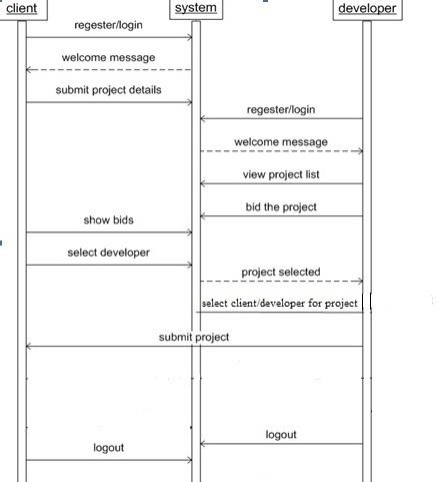
****

**Figure 3.3.1(a) :** Use case diagram of customer module

****

**Figure 3.3.1(b**) : Use case diagram of Admin Module

### Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

**Figure 3.3.2:** Sequence diagram of web page

**Figure 3.3.2(c**) : Use case diagram of Admin Module

**3.4 ER Diagram**

Entity Relationship model (ER model for short) is an abstract way to describe a database design. It usually starts with a relational database, which stores data in tables. Some of the data in these tables point to data in other tables.Data in the table is retrieved using primary key. The primary key is the one which uniquely identifies each record in the table in relational database management system.In ER Diagram an entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified. An entity is an abstraction from the complexities of a domain.A relationship captures how entities are related to one another. Entity–relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets and relationship sets.In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.Entity–relationship modeling was developed for database design by Peter Chen and published in a 1976 paper.[[1]](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model#cite_note-Chen-1) However, variants of the idea existed previously.Some ER modelers show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontologies.

POSTS

PROJECT

HIRE

BIDS

HIRES

BIDDING

WORK

JOB

BIDDING

**Figure 3.4**: ER Diagram of whole system

### 3.5 SCHEMA DIAGRAM

|  |
| --- |
| project\_name category technologies amount hire\_name details |

### PROJECT:

### WORK:

|  |
| --- |
| email-id password name phone\_no skills |

### HIRE:

|  |
| --- |
| email\_id password name phone\_no |

### JOB:

|  |
| --- |
| project\_name hire\_name work\_name bid\_amount hire\_feedback work\_feedback progress amount |

**s**

### BIDDING:

|  |
| --- |
| hire\_name work\_name project\_nam amount |

**Figure 3.5:** Schema diagram for freelancer

**3.6 Database Table Structure**

|  |  |  |
| --- | --- | --- |
| **Table name** | **Attributes** | **Primary keys** |
| Table Hire | Email\_id(PK varchar(50), Not null)  Password(varchar(20), null)  Name(FK char(30),null)  Phone\_no(real,null) | Email\_id |
| Table Project | Project\_name(PK varchar(20), not null)  Hire\_name (varchar(20), null)  details(varchar(20),null)  technologiesl(varchar(20),null)  catagories(varchar(50), null) | Project\_name |
| Table work | Email\_id(PK varchar(50),not null)  password(varchar(20), null)  name(varchar(50),null)  phone\_no(real,null)  skills(varchar(15),not null) | Email\_id |
| Table Job | Project\_name(FK varchar(50),not null)  Hire\_name(varchar(50), null)  Work\_name(varchar(50),null)  Hire\_feedback(varchar(50),not null)  Work\_feedback(varchar(50),not null)  Bid\_amount(real,not null)  Amount(real,not null)  Progress(varchar(20),null) |  |
| Table Bidding | Hire\_name(FK varchar(50), null)  Work\_name(varchar(50), null)  Project\_name(varchar(30),null)  bid\_amount(real,not null) |  |

**Figure 3.5** : Database table Structure

# CHAPTER 4

**DATABASE TECHNIQUES & RESULTS**

**4.1Stored Procedure:-**

Stored procedure is a procedure written in SQL and other control statement stored in database which can be called by the database engine and connected programming languages.

In this e-farming system there is a stored procedure to be executed whenever a new login is to be done when a user enters details in the page.

$(document).ready(function(){

$('#characterLeft').text('140 characters left');

$('#message').keydown(function () {

var max = 140

varlen = $(this).val().length

if (len>= max)

$('#characterLeft').text('You have reached the limit')

$('#characterLeft').addClass('red')

$('#btnSubmit').addClass('disabled');

}

Else

$('#characterLeft').text(ch + ' characters left');

$('#btnSubmit').removeClass('disabled');

$('#characterLeft').removeClass('red');

}

});

});

**4.2 Trigger:-**

The MySQL trigger is a database object that is associated with a table it will be activated when a defined action is executed for the table the trigger can be executed when you run one of the following MySQL statements on the table that is INSERT, UPDATE and DELETE before or after the event.

In this Online E-farmigthere is a trigger to be executed to set the vegetable to be ready to be marketedwhenever the farmer is getting is yeild.

CREATE TRIGGER ‘ftrigger’ BEFORE INSERT ON ‘job’

FOR EACH ROW BEGIN

delete from bidding where project\_name=new.project\_name;

delete from project where project\_name=new.project\_name;

END

CREATE TRIGGER ‘ftrigger’ BEFORE INSERT ON ‘job’

FOR EACH ROW BEGIN

delete from bidding where project\_name=new.project\_name;

delete from project where project\_name=new.project\_name;

END

**Figure** **4.2** : Trigger

**4.3 Testing :**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product .It provides a way to check the functionality of components, sub-assemblies and or a finished product. It is the process of exercising software with the intent of ensuring that a software system meets its requirements and user expectation does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**4.3.1 Types of Testing**

**4.3.2 Unit Testing**

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

**4.3.3 Integrated Testing**

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up a process are tested together. Beyond that, if the program is composed of more than one process, they should be tested in pairs rather than all at once.

**4.3.4 System Testing**

The process of performing a variety of tests on a system to explore functionality or to identify problems is called System Testing. System testing is usually required before and after a system is put in place. A series of systematic procedures are referred to while testing is being performed. These procedures tell the tester how the system should perform and where common mistakes may be found. Testers usually try to "break the system" by entering data that may cause the system to malfunction or return incorrect information. For example, a tester may put in a city in a search engine designed to only accept states, to see how the system respond to the incorrect inputs.

**4.3.5 Test Case**

Test case is a set of test inputs, executions and expected results developed for a particular objective.

An excellent test case satisfies the following criteria:

* Reasonable probability of catching errors
* Does interesting things
* Doesn’t do unnecessary things
* Neither too simple nor too complex
* Allows isolation and identification of errors

**4.3.6 Unit Testing**

**Admin Module**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TEST**  **CASE**  **ID** | **TEST**  **CASE** scenario | **TEST CASE DESCRIPTION** | **INPUT** | **EXPECTED RESULT** | **ACTUAL RESULT** | **RESULT** |
| TC 01 | Check user login | Go to site Enter uid and Password | Enter valid  Uid and Password  Click submit | User should login to site | As expected | PASS |
| TC 02 | Check clients login with invalid data | Validate Uid id and Password | Enter  Invalid U id and Password  Clock submit | User should not login to site | As expected | PASS |
| TC 03 | Check post project updated  To each developers profile | Fill the project details | Then click post project | project details should upload to each developer profile | As expected | PASS |
| TC 04 | Check weather all bided developers are displayed for particular project | Bid amount to project posted by client | Client post the project | Bided developers should display | As expected | PASS |
| TC 05 | Check the project assign particular developer, project details should be deleted in other developer profile | Assign project to particular developer | Developer design the aasigned job | Project details should be deleted in other developer profile | As expected | PASS |
| TC 06 | Check if finish job assigned by client/developer it need to updated in their past project detail | Click finish button | Enter the feedback | Should update in their past project detail | As expected | PASS |
| TC 07 | Check if unauthorized access is disabled for people who try to access site through URL after logout | Tale URL displays after valid user login. | Paste it in to new tab after logout. | Should bring in to login page | As expected | PASS |

**Table 4.3.6:** whole module Test cases (Unit Testing)

# 

4.3 MAJOR MODULES

**CLIENT MODULE:**

Post a project:  
 Clients are always free to post their project. They automatically begin to receive  
bids from our freelancers. Alternatively, they can browse through the talent available on  
our site, and make a direct offer to a freelancer instead

Choose the perfect freelancer:  
•Browse freelancer profiles  
•Chat in real-time  
•Compare proposals and select the best one  
•Award your project and your freelancer goes to work

Pay when you are satisfied:

Pay safely using chat attribute by exchanging their accounts.

DEVELOPER MODULE:

Complete your profile:  
•Select your skills and expertise  
•Upload a professional profile photo  
•Go through the Verification Center checklist  
 **Browse jobs that suit your skills, expertise, price, and schedule:** We have jobs available for all skills. Maximize your job opportunities by  
optimizing your filters. Save your search and get alerted when relevant jobs are available.

Write your best bid:

Put your best foot forward and write the best pitch possible. Read the project and  
let the clients know you understand their brief. Tell them why you're the best person for  
this job. Writing a new brief for each project is more effective than using the same one!

Get awarded and earn:

Get ready to work once you get hired. Deliver high quality work and earn the  
agreed amount

# CHAPTER 5

# 5.1 SNAP SHOTS

**Homepage of the web portal :**

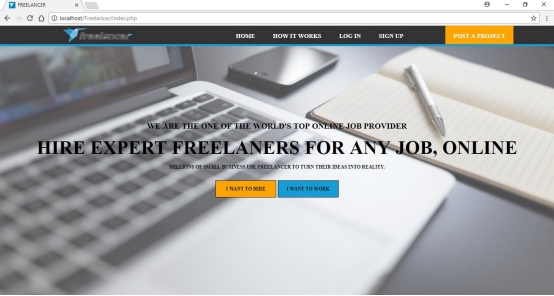
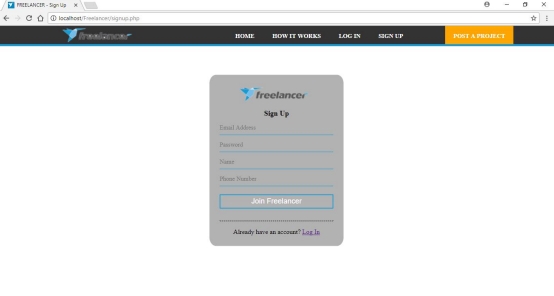


Figure : home page of freelancer web portal

This is the main page of the web page, which contains different modules and activities to perform respective tasks.

### SIGN UP PAGE:

****

**Figure :** User interface page

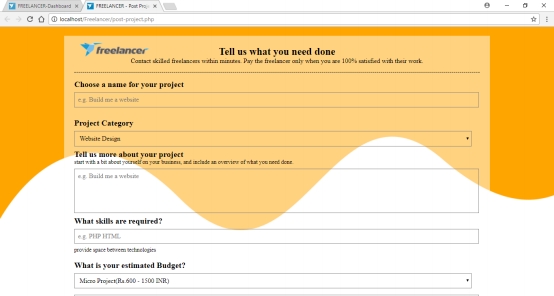
Clients and Developer registration pages, which need to be filled there appropriate details for the registration process

# CHECKING VALIDITY:

**Figure** : Checking validity

# POSTING PROJECTS:

Figure : admin page to add a projects details

****

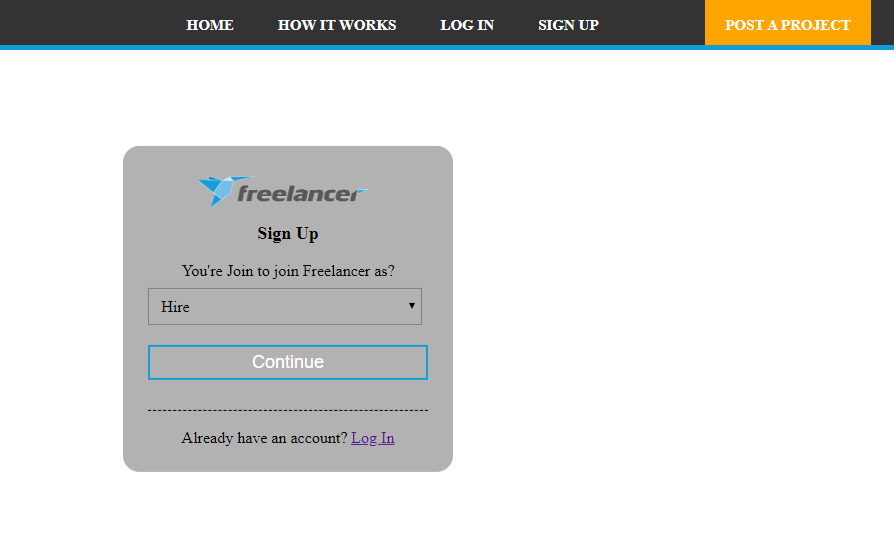
**Figure :** Admin page to post project detail

### VIEW PROJECT BIDING:

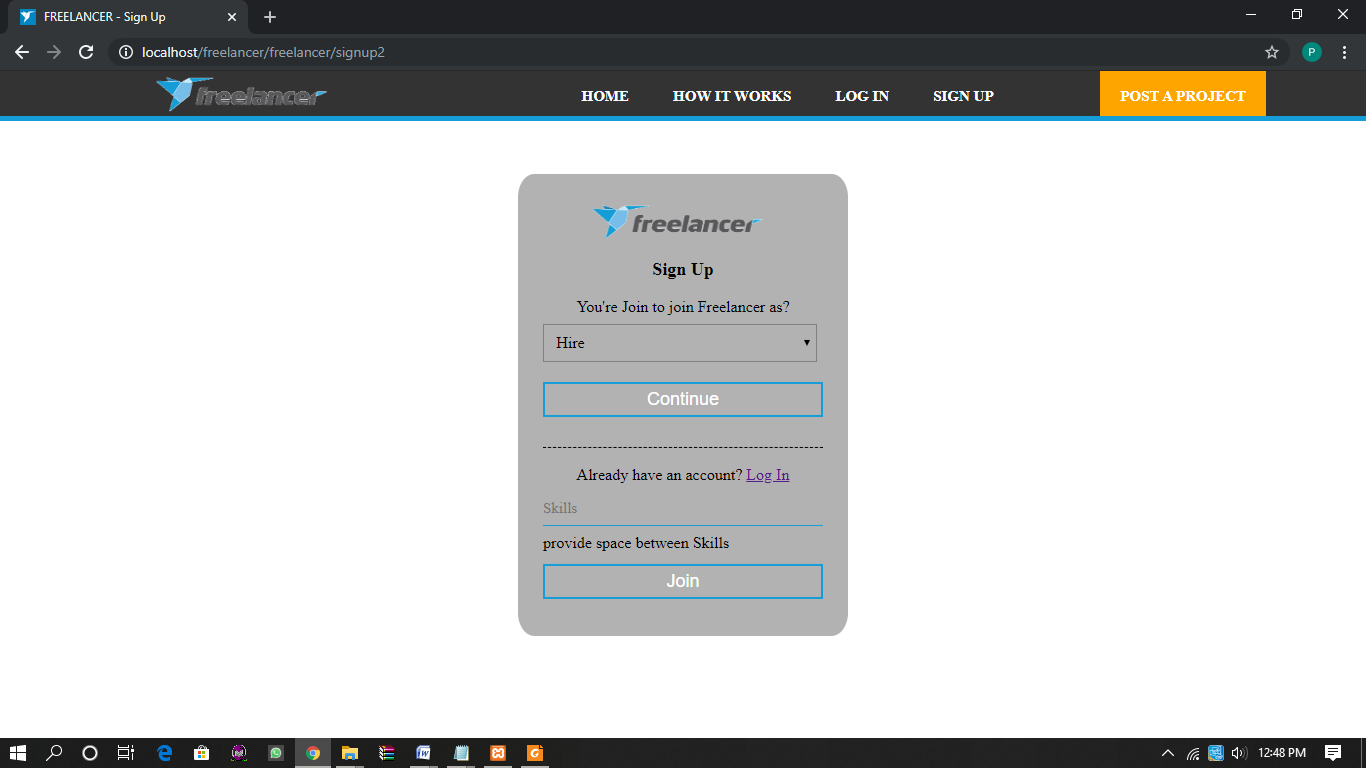
****

**Figure**: viewing the project biding

### SELECTING THE ROLE:



**Figure** :Selecting the role page



**Figure** 5.1: Woker details

**CONCLUSION**

The application “veggie bowl” would be a boon to Indian farmers as well as common people in the remote areas. Those who are already using an phone, can register with their phone number and get an account. The data sent by them will be stored in the corresponding account. The experts may provide their advice to that particular phone number. By this application, the customer’s may be able to buy agriculture yields directly from the farmer profiting them better through the online web portal which is dedicated to Indian farmer’s. Using this application, farmers may manage the crop loss and prevent the food wastage and can get proper price for their produce. Special features which would popularize the application are Simple Graphical User Interface (GUI) that can be used by everybody. One Stop Solution” to all kinds of issues to the farmers.The main objective of this project is build a website which will help farmers from Indian villages to sell their products to different cities and obtain real benefits of online marketing which eliminates the middle men who snatch away the profits from farmers which is satisfied in this project.

# References

1.Learning PHP, MySQL & JavaScript

* Robin Nixon

2.Learn HTML and CSS with W3 Schools

* HegeRefsnes
* Stale Refsnes
* Jim Refsnes
* Jan EgilRefsnes
* C. Michael Woodward

3.W3 Schools - https://www.w3schools.com

4.Stack overflow – Http://stackoverflow.com