VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 014

A Database Management System Project Report On

"FREELANCER HIRING SYSTEM"

Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of

Bachelor of Engineering

In

Computer Science & Engineering

By

RAUNAK CHOUDHARY (4MW17CS064)

Under the guidance of

Ms. SOWMYA S
Assistant Professor

Department of Computer Science and Engineering
SHRI MADHWA VADIRAJA INSTITUTE OF TECHNOLOGY AND MANAGEMENT
Vishwothama Nagar, BANTAKAL – 574 115, Udupi District

NOVEMBER, 2019

SHRI MADHWA VADIRAJA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(A Unit of Shri Sode Vadiraja Mutt Education Trust ®, Udupi)

Vishwothama Nagar, BANTAKAL – 574 115, Udupi District, Karnataka, INDIA

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Database Management System Project Work titled **'Freelancer Hiring System'** has been carried out by **Mr. Raunak Choudhary** (**4MW17CS064)** who is the bonafide student of Shri Madhwa Vadiraja Institute of Technology and Management, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2019-20. This Database Management System Project Report has been approved as it satisfies the academic requirements with respect to the project work guidelines prescribed for the said Degree.

Ms. SOWMYA S	**Dr. VASUDEVA**
Professor & Project Guide	Professor and Head
Dept. of CSE	Dept. of CSE

External Viva

Name of the Examiners:	**Signature with Date**
1. ____________________	____________________
2. ____________________	____________________

Acknowledgements

First and foremost, I express my gratitude to my project guide Prof. **Ms. Sowmya S**, Assistant Lecturer, Dept. of Computer Science and Engineering. Her willingness to motivate me and her guidance contributed tremendously to this project work.

We are indebted to Prof. **Dr. Thirumaleshwara Bhat**, Principal and Prof. **Dr. Vasudeva**, Head of the Department, for their advice and suggestions at various stages of the work.

We are also indebted to the Management of Shri Madhwa Vadiraja Institute of Technology and Management, Bantakal, Udupi for providing us with a good study environment and excellent laboratories facilities. We appreciate the help and support rendered by the teaching and non-teaching staff of Computer Science and Engineering Department.

I also want to thank my Parents for providing me motivation whenever it was needed, without my Parents blessings, this project would not have been completed.

Lastly, I take this opportunity to offer my regards to all of those who have supported me directly or indirectly in completing this project work.

RAUNAK CHOUDHARY

ABSTRACT

Nowadays every company wants their employees to work for long periods and wants to add more responsibility to their work ethic. Freelancing is an excellent and efficient way of working anytime and from almost anywhere whilst earning money as well. It provides flexibility and comfort to the workers. Freelancing provides increased productivity and is highly cost effective.

My aim is to provide an online platform for freelancers to find potential clients, and customers can post to find a developer who fits their requirements to run and grow their business. A Freelance Hiring System, is a system used by companies or individuals to streamline their freelance management. It should not be confused with a freelance marketplace, which focuses on talent sourcing but lacks the features to handle complex operational needs of companies with a large freelancer workforce.

These hybrid sourcing solutions combine the best elements of early Freelancer Hiring Systems with access to robust pools of talent (typically through online Freelance Marketplaces), compliance services, and all the tools needed to facilitate Freelance Engagements.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Purpose of the Project:

This project aims to provide an online platform for freelancers to find potential clients, and customers can post to find a developer who fits their requirements to run and grow their business. It makes it easier for freelancers to look for jobs as they meet companies who can hire them on the same platform.

1.2 Motivation:

The motivation for this project came to me when I saw that how every company wants their employees to work for long periods and wants to add more responsibility to their work ethic. This results in decreased efficiency in work and also reduces flexibility and comfort in work. There I got the implementing the idea of Freelancers and making the work more efficient.

1.3 Existing Procedure:

Currently, there are some websites which provide an opportunity for freelancers to find jobs and also for the companies to look for the employees in the domain they require. I have implemented the project based on a similar ideology so as to make it easier for the job provider and receiver to meet on the same platform.

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes the functional and non-functional requirement for the software to be developed. The functional requirements include what the software should do and non-functional requirement include the constraint on the design or implementation. Requirements must be measurable, testable, related to identified needs or opportunities, and defined to a level of detail sufficient for the system design.

Software requirement specification will contain states what the software will do. When the Software has to do directly perceived by its users – either human users or other software systems. The common understanding between the user and the developer is captured in the requirements document. The writing of software requirement specification reduces development effort, as careful review of the document can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when these problems are easier to correct. The SRS discusses the product but not the project that developed it; hence the SRS serves as a basis for later enhancement of the finished product. The SRS may need to be altered, but it does provide a foundation for continued production evaluation.

2.1 Operating Environment

My proposed system will work on any system that has the ability to run the latest version of the Google Chrome Browser and should have an Internet Connection.

Hardware Requirements

Table 2.1 Hardware Requirements

Hardware	**Description**
Processor	Intel Pentium4 or newer
RAM	2GB
Hard Disk	10GB

Software Requirements

Table 2.2 Software Requirements

Software	**Description**
Operating System	Windows 7 or newer
Web Hosting	Local Host
Programming Language	NODE.JS, HTML
Database	MySQL

2.2 Functional Requirements

These are the statements of services which, system should provide, how the system should react for particular inputs and how the system should behave in particular situations.

They are: -

- System should be accessible from anywhere in the world.
- Should provide easy user interfaces.
- Freelancers and Companies should have separate account with associated functionalities.
- System should provide interface for the Companies to look out for the available Freelancers.

- System should provide interface for the Freelancers to look for the companies and the type of jobs it is providing.

2.3 Non-Functional Requirements

Accessibility

Accessibility can be viewed as the “ability to access” and benefit from a system. Help text for the modules is provided wherever necessary, which guides the user into being able to access the functionalities of the modules.

Availability

These are the statements of services which, system should provide, how the system should react for particular inputs and situations

Compatibility

As the system requires just the latest Google Chrome Browser, so it can be run on any operating system.

Performance

Since the web application does not require any critical procedure, the system performance only depends on the server’s capacity to serve the client(s).

Reliability

Reliability is the ability of the system to perform and maintain its functions in routine, hostile and unexpected circumstances.

Portability

Since the app just requires latest Chrome browser so portability is not an issue

Usability

A system like the one proposed in this project is very helpful for people looking to work for a shorter time and gaining experience.

Security

Since the web app can be accessed by only the authorized users and since each of the user type has a different level of access, the system is secure.

2.4 User Characteristics

Company

- Need to have a system connected to the Internet
- Can add jobs according to the requirement.
- Can directly contact the freelancer with the mobile number provided.

Freelancer

- Need to have a system connected to the Internet.
- Each freelancer has a profile to which he/she has to login.
- Can view the details in the database i.e. the jobs available and stipend provided.

Chapter 3

LITERATURE SURVEY

3.1 Current Situation:

Currently, there are websites which provide Freelancer hiring. They provide a platform for freelancers to apply for jobs based on their skill set and work domain.

3.2 Front-end Technology

- **HTML – Hypertext Markup Language (HTML)**

  HTML is the standard markup language for creating web pages and web applications. It forms the skeletal framework for creating a webpage.

- **CSS – Cascading Style Sheets (CSS)**

  CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

- **NODE.JS –**

  (Node) is an open source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications

3.3 Database Technology

- **MySQL –**

  MySQL is an open-source relational database management system (RDBMS). Here SQL is the abbreviation for Structured Query Language.

Chapter 4

SYSTEM DESIGN

4.1 Database Design

The data required for this project is organized and store as tables in MYSQL database.

The lists of tables in this project are:

- **Buyerinfo** : Information about freelancer including his personal details to apply for jobs
- **Joblisting** : Shows the information about the type of job application. Lists out the job description in general.
- **Login** : User login information. Used while logging into user account. Helps in authorization.
- **Ssign** : Sign-in information about the company that makes a new account in order to hire people.
- **Binfo** : Includes extra information about worker’s hobbies and interests.

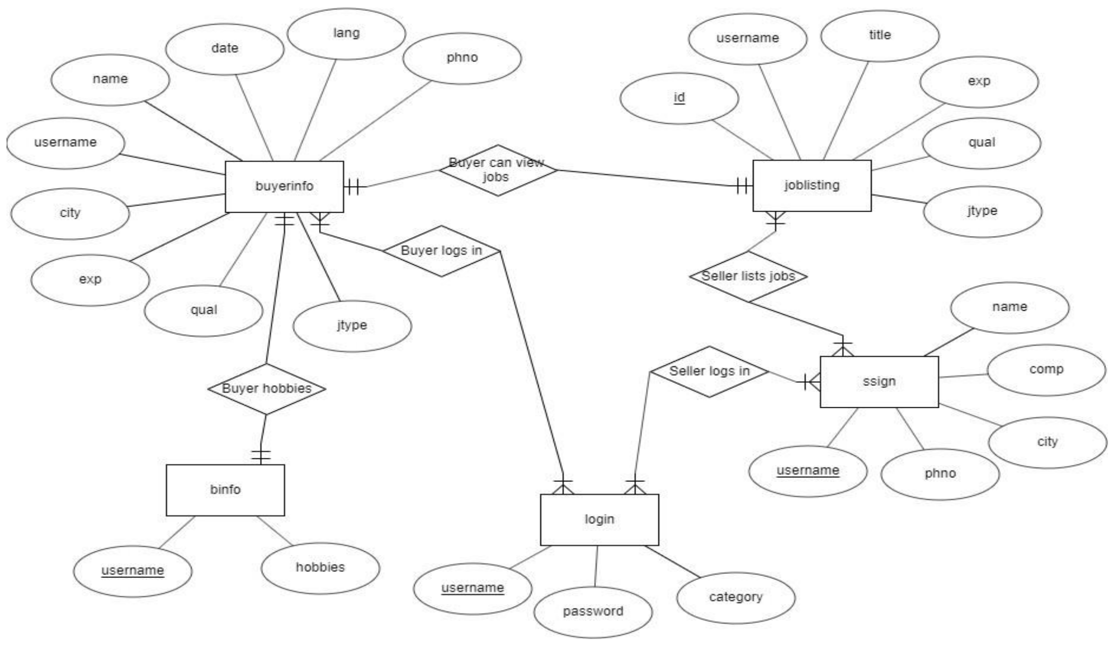


Fig 4.1: Entity Relationship diagram

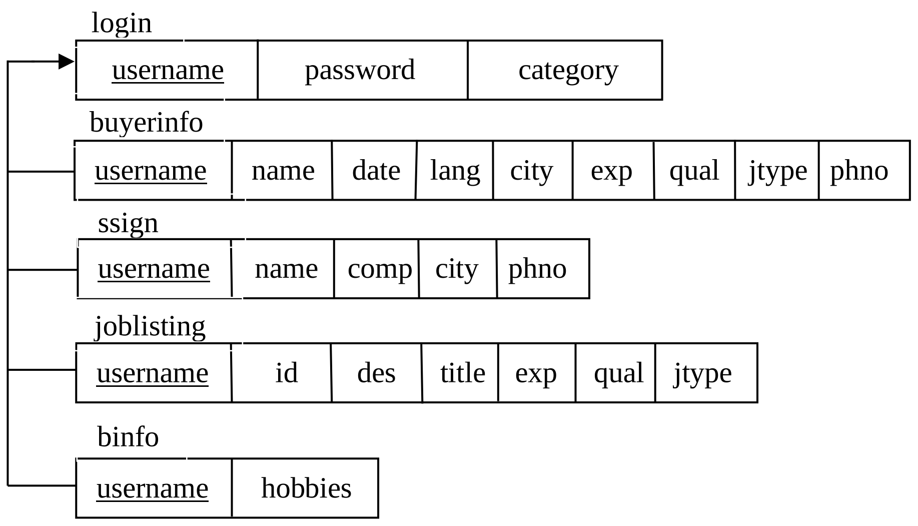
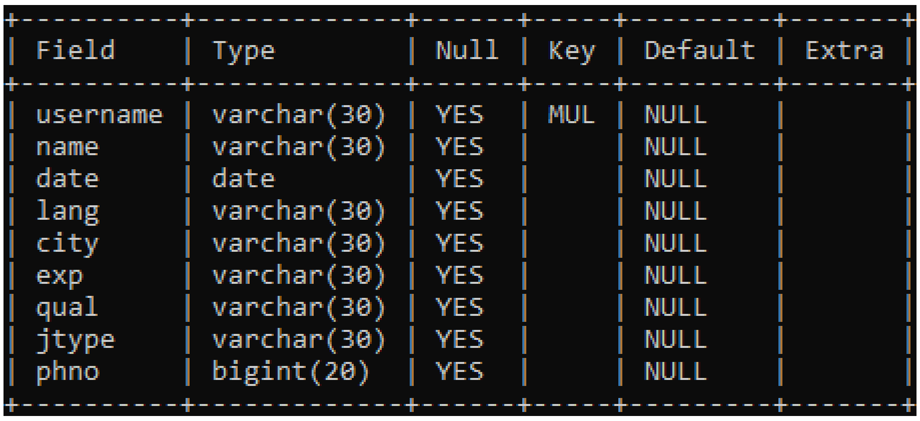


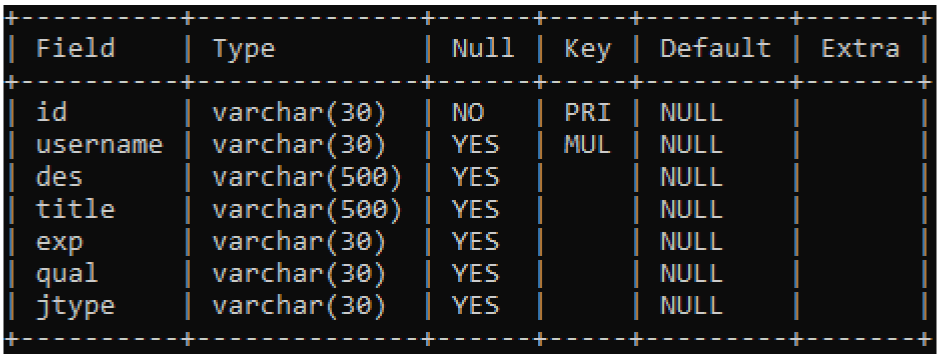
Fig 4.2: Schema diagram

Table 4.1 buyerinfo



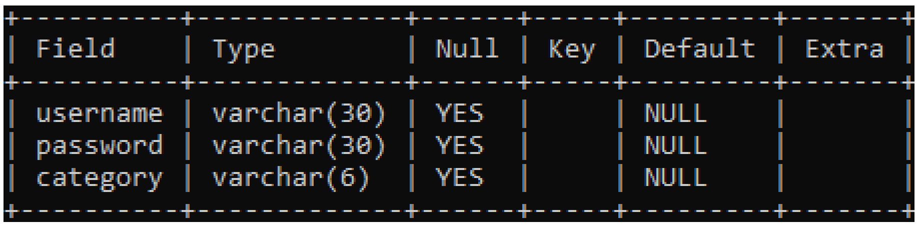
```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(30) | YES  | MUL | NULL    |       |
| name     | varchar(30) | YES  |     | NULL    |       |
| date     | date        | YES  |     | NULL    |       |
| lang     | varchar(30) | YES  |     | NULL    |       |
| city     | varchar(30) | YES  |     | NULL    |       |
| exp      | varchar(30) | YES  |     | NULL    |       |
| qual     | varchar(30) | YES  |     | NULL    |       |
| jtype    | varchar(30) | YES  |     | NULL    |       |
| phno     | bigint(20)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Table 4.2 joblisting



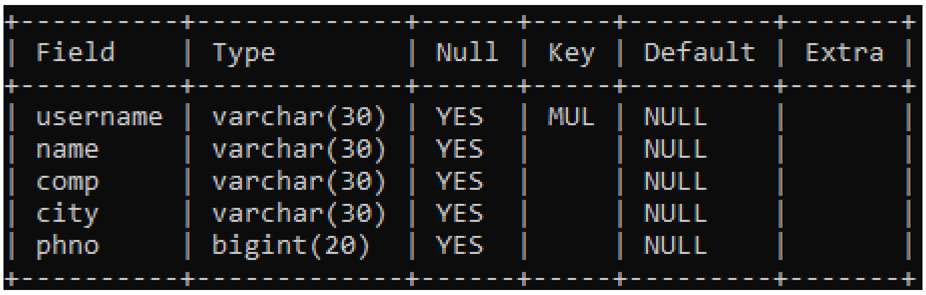
```
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| id       | varchar(30)  | NO   | PRI | NULL    |       |
| username | varchar(30)  | YES  | MUL | NULL    |       |
| des      | varchar(500) | YES  |     | NULL    |       |
| title    | varchar(500) | YES  |     | NULL    |       |
| exp      | varchar(30)  | YES  |     | NULL    |       |
| qual     | varchar(30)  | YES  |     | NULL    |       |
| jtype    | varchar(30)  | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

Table 4.3 login



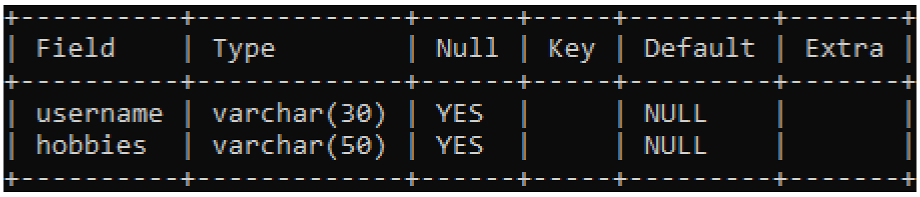
```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(30) | YES  |     | NULL    |       |
| password | varchar(30) | YES  |     | NULL    |       |
| category | varchar(6)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Table 4.4 ssign



```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(30) | YES  | MUL | NULL    |       |
| name     | varchar(30) | YES  |     | NULL    |       |
| comp     | varchar(30) | YES  |     | NULL    |       |
| city     | varchar(30) | YES  |     | NULL    |       |
| phno     | bigint(20)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Table 4.5 binfo



```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(30) | YES  |     | NULL    |       |
| hobbies  | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

4.2 Normalization

Normalization is the process of organizing data into a related table; it also eliminates redundancy and increases the integrity which improves performance of the query. To normalize a database, we divide the database into tables and establish relationships between the tables. Database normalization can essentially be defined as the practice of optimizing table structures. Optimization is accomplished as a result of a thorough investigation of the various pieces of data that will be stored within the database, in particular concentrating upon how this data is interrelated. Optimization of database is very important.

To decide a suitable logical structure for given database design the concept of normalization, which are briefly described below.

1. **1st Normal Form (1 NF):** A relation is said to be in 1 NF if and only if all unaligned domains contain one value only. That is the fields of an n-set should have no group items and no repeating groups.

2. **2nd Normal Form (2 NF):** A relation is said to be in 2 NF if and only if it is in 1 NF and every non key attribute is fully dependent on primary key. This normal takes care of functional dependencies on non-key attributes.
3. **3rd Normal Form (3 NF):** A relation is said to be in 3 NF if and only if it is in 2 NF and every non key attribute is non transitively dependent on the primary key. This normal form avoids the transitive dependencies on the primary key.
4. **Boyce code Normal Form (BCNF):** This is a stronger definition than that of NF. A relation is said to be in BCNF if and only if every determinant is a Candidate key.
5. **4th Normal Form (4 NF):** A relation is said to be in 4 NF if and only if whenever there exists a multi valued dependency in a relation say A- >>B then all of the relation is also functionally dependent on A (i.e. A- >X for all attributes x of the relation.).
6. **5th Normal Form (5 NF) OR Projection Join Normal Form (PJNF):** A relation R is in 5 NF if and only if every join dependency in R is implied by the candidate key on R . A relation can’t be non-loss split into two tables but can be split into three tables. This is called Join Dependency.

Chapter 5

IMPLEMENTATION

5.1 Queries

SQL Table Creation Queries (in MySQL)

- create table buyerinfo(username varchar(30) primary key,name varchar(30),dob date,lang varchar(30),city varchar(30),exp varchar(30),qual varchar(30),jtype varchar(30),phno int);
- create table joblisting(id varchar(30) primary key,username varchar(30),title varchar(30),exp varchar(30),qual varchar(30),jtype varchar(30));
- create table login(username varchar(30),password varchar(30),category varchar(6));
- create table ssign(username varchar(30) primary key,name varchar(30),comp varchar(30),city varchar(30),phno int);
- create table binfo(username varchar(30),hobbies varchar(50));
- alter table login add foreign key(username) references buyerinfo(username) on delete cascade;
- alter table ssign add foreign key(username) references buyerinfo(username) on delete cascade;
- alter table joblisting add foreign key(username) references buyerinfo(username) on delete cascade;
- alter table binfo add foreign key(username) references buyerinfo(username) on delete cascade;

5.2 Connecting to database

Establish MySQL Connection (in node.js):

```
var express = require('express');

var app = express();

var moment = require('moment');

var session = require('express-session');

var bodyParser = require('body-parser');

var mysql = require('mysql');

var urlencoderParser = bodyParser.urlencoded({ extended: false });


//----------------------------------------

//Init

//----------------------------------------

app.use(express.static('views'));

app.use(express.static('css'));

app.set('view engine', 'ejs');

app.use(

  session({

    name: 'sid',

    secret: '124343',

    resave: true,

    saveUninitialized: true
  })
);
app.use(bodyParser.json());


var con = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  insecureAuth: true,
  database: 'nodelogin'
});


const redirectLogin = (req, res, next) => {
  if (!req.session.username) {
    res.redirect('/login');
  } else next();
};
//----------------------------------------------
//Render routes
// --------------------------------------------
app.get('/login', function(req, res) {
  res.render('Login.ejs', { state: 'hidden' });
});
```

```
app.get('/sellerSignup', function(req, res) {

  res.render('SellerSignup.ejs');

});

app.get('/buyerSignup', function(req, res) {

  res.render('BuyerSignup.ejs');

});

app.get('/shome', redirectLogin, function(req, res) {

  if (req.session.cat == 'S')

    con.query(

      'SELECT * FROM joblisting WHERE username = ?',

      [req.session.username],

      function(err, result, fields) {

        res.render('Company View.ejs', { jobs: result });

      }

    );

  else res.redirect('/sellerSignup');

});



app.get('/bhome', redirectLogin, function(req, res) {

  if (req.session.cat == 'S')

    con.query('SELECT * FROM joblisting', function(err, result,
fields) {

      res.render('BuyerView.ejs', { jobs: result });
    });

  else res.redirect('/sellerSignup');

});

app.get('/addjob', redirectLogin, function(req, res) {

  if (req.session.cat == 'S') res.render('AddJob.ejs');

  else res.redirect('/login');

});



//----------------------------------------------

//Post Request

//----------------------------------------------



//Login

app.post("/O'auth", urlencoderParser, function(req, res) {

  var username = req.body.username;

  var password = req.body.password;

  con.query(

    'SELECT * FROM login WHERE username = ? AND password = ?;',

    [username, password],

    function(err, result, fields) {

      console.log(result);

      if (result.length > 0) {

        var cat = result[0].category;
        req.session.loggedin = true;

        req.session.username = username;

        req.session.cat = cat;

        if (cat == 'S') res.redirect('/shome');

        if (cat == 'B') res.redirect('/bhome');

      } else {

        res.render('login.ejs', { state: 'text' });

      }

    }

  );

});


//Buyer Signup

app.post('/newBSignup', urlencoderParser, function(req, res) {

  var name = req.body.name;

  var username = req.body.username;

  var date = req.body.date;

  var password = req.body.password;

  var lang = req.body.lang;

  var city = req.body.city;

  var exp = req.body.exp;

  var qual = req.body.qual;

  var jtype = req.body.jtype;
  var phno = req.body.phno;


  con.query(
    'INSERT INTO buyerinfo VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)',
    [username, name, date, lang, city, exp, qual, jtype, phno],
    function(err, result) {
      if (err) throw err;
      else {
        console.log('Record inserted');
        con.query(
          'INSERT INTO login VALUES (?, ?, ?)',
          [username, password, 'B'],
          function(err, result) {
            if (err) throw err;
            else {
              console.log('User registered');
              res.redirect('/bhome');
            }
          }
        );
      }
    }
  );
```

```
});


//Seller Signup
app.post('/newSSignup', urlencoderParser, function(req, res) {
  var name = req.body.name;
  var comp = req.body.cname;
  var username = req.body.username;
  var password = req.body.password;
  var city = req.body.city;
  var phno = req.body.phno;

  con.query(
    'INSERT INTO ssign VALUES (?, ?, ?, ?, ?)',
    [username, name, comp, city, phno],
    function(err, result) {
      if (err) throw err;
      else {
        console.log('Record inserted');
        con.query(
          'INSERT INTO login VALUES (?, ?, ?)',
          [username, password, 'S'],
          function(err, result) {
            if (err) throw err;
            else {

              console.log('User registered');

              res.redirect('/shome');

            }

          }

        );

      }

    }

  );

});



//Seller info ===> Add Button ===> Seller info

app.post('/addjob', redirectLogin, urlencoderParser, function(req,
res) {

  console.log(

    'User:' + req.session.username + ' has request Add Page' +
req.session.cat

  );

  if (req.session.cat == 'S') var title = req.body.title;

  var username = req.session.username;

  var des = req.body.des;

  var exp = req.body.exp;

  var qual = req.body.qual;

  var jtype = req.body.jtype;
  console.log('----------------------->' + req.session.username);


  con.query(

    'INSERT INTO joblisting VALUES (?,?, ?, ?, ?, ?, ?)',

    ['hello', username, title, des, exp, qual, jtype],

    function(err, result) {

      if (err) throw err;

      else {

        console.log('Add Job');

        res.redirect('/shome');

      }

    }

  );

});


//Seller info ==> delete ==> Seller info

app.post('/deletepost', urlencoderParser, function(req, res) {

  var postid = req.body.id;

  console.log(postid);

  con.query('DELETE FROM joblisting WHERE id = ?;', [postid], 
function(

    err,

    result
  ) {

    if (err) throw err;

    else {

      console.log('Deleted Job');

      res.redirect('/shome');

    }

  });

});

//-------------------------------------------------

//Server config

//-------------------------------------------------

var server = app.listen(8081, function() {

  var host =  server.address().address;

  var port = server.address().address;



  console.log('Example app listening at http://%s:%s', host,
port);

});
```

Chapter 6

RESULTS

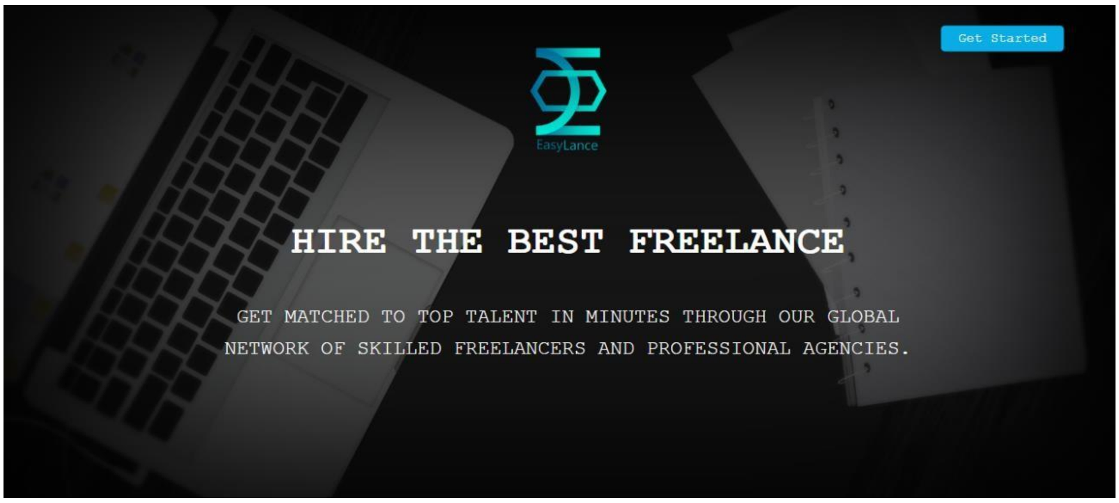

Get Started
EasyLance
HIRE THE BEST FREELANCE
GET MATCHED TO TOP TALENT IN MINUTES THROUGH OUR GLOBAL NETWORK OF SKILLED FREELANCERS AND PROFESSIONAL AGENCIES.

Fig 6.1: Homepage

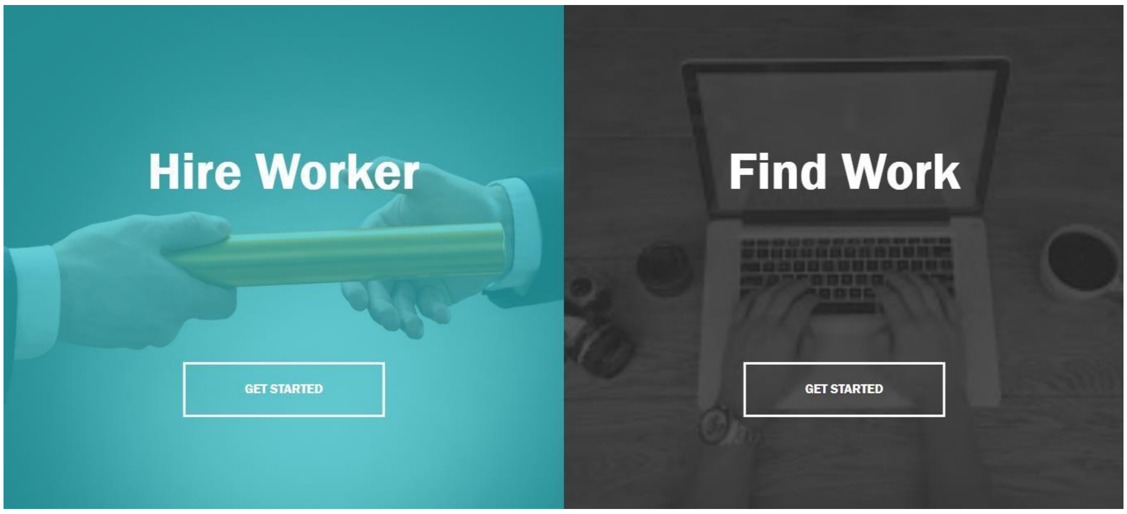


Fig 6.2: Selecting Option

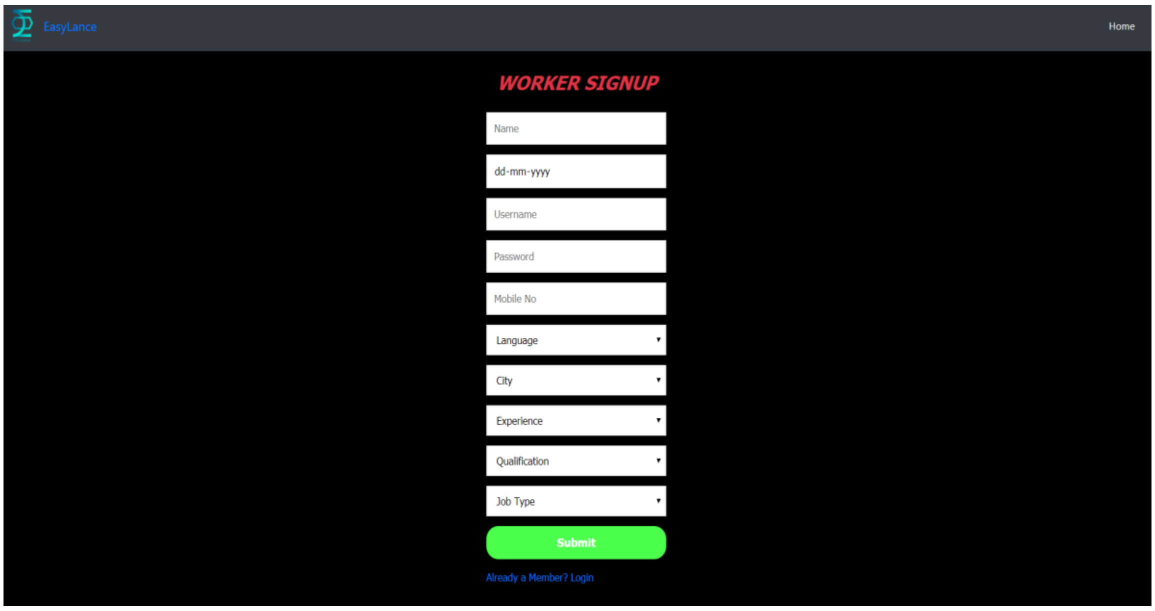


Fig 6.3: Worker Signup Page

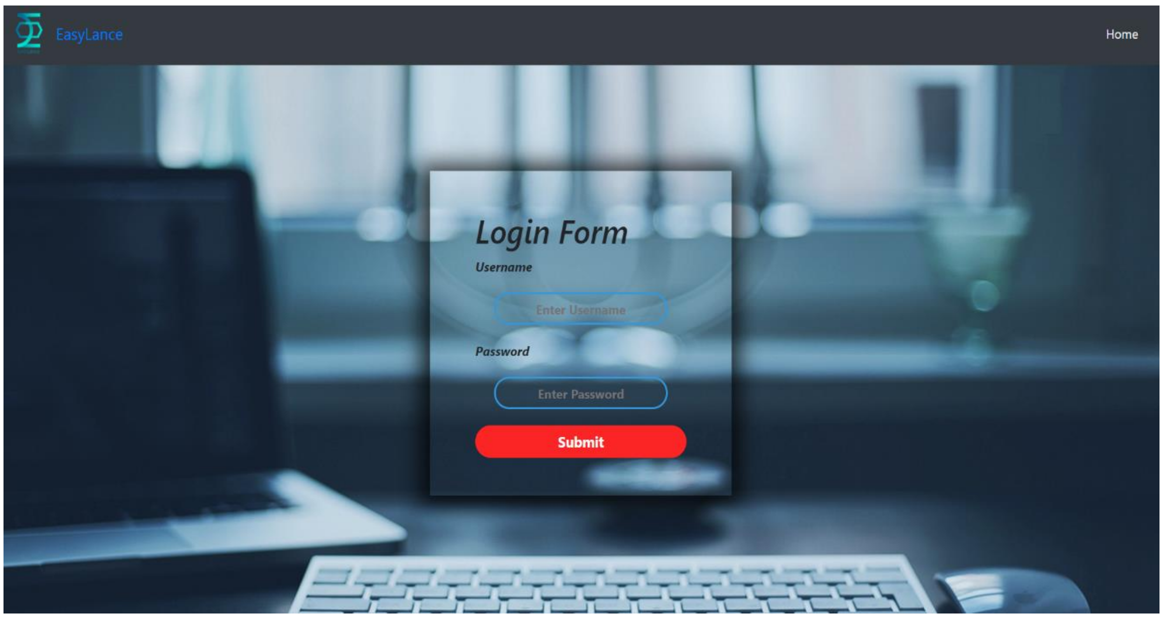
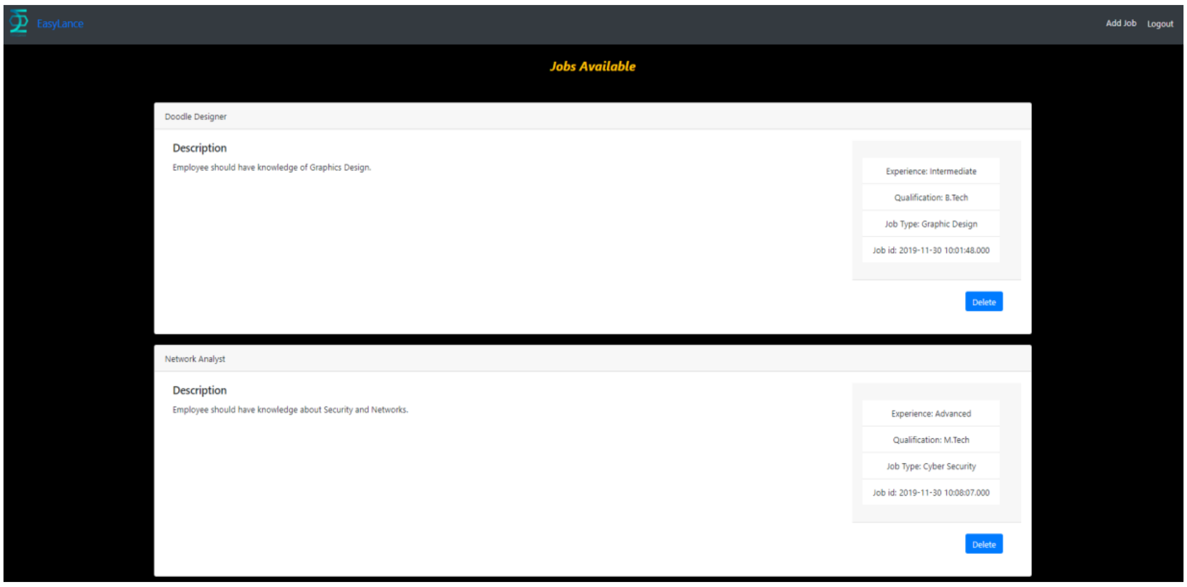


Fig 6.4: Login Page



Jobs Available

Doodle Designer

Description

Employee should have knowledge of Graphics Design.

Experience: Intermediate

Qualification: B.Tech

Job Type: Graphic Design

Job id: 2019-11-30 10:01:48.000

Network Analyst

Description

Employee should have knowledge about Security and Networks.

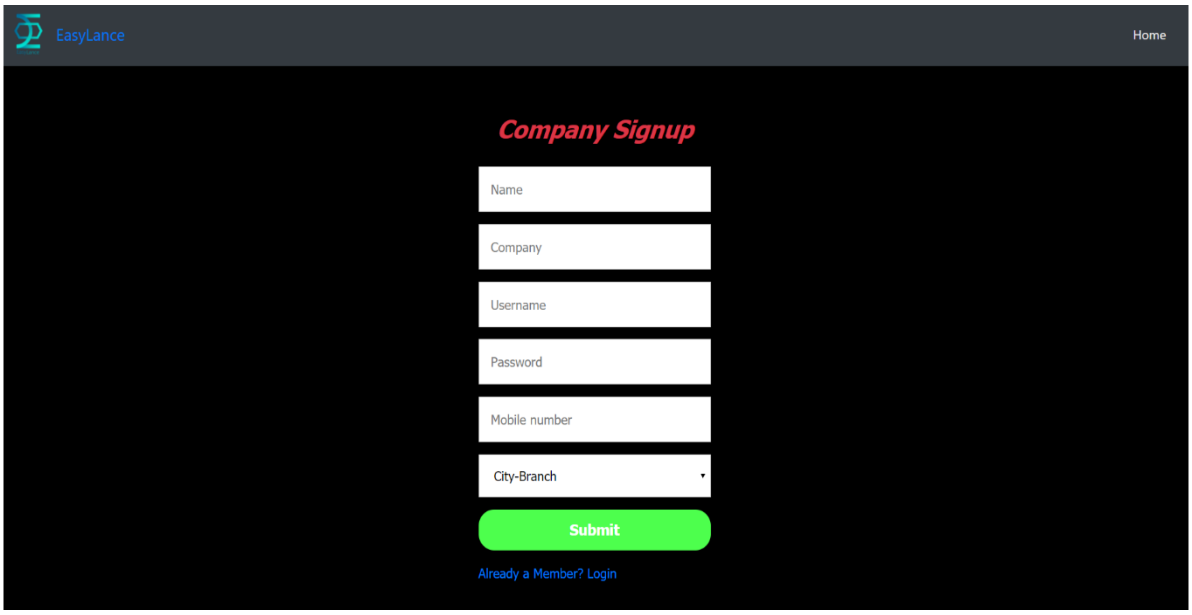
Experience: Advanced

Qualification: M.Tech

Job Type: Cyber Security

Job id: 2019-11-30 10:08:07.000

Fig 6.5: Jobs Available



Company Signup

Name

Company

Username

Password

Mobile number

City-Branch

Fig 6.6: Company Signup Page

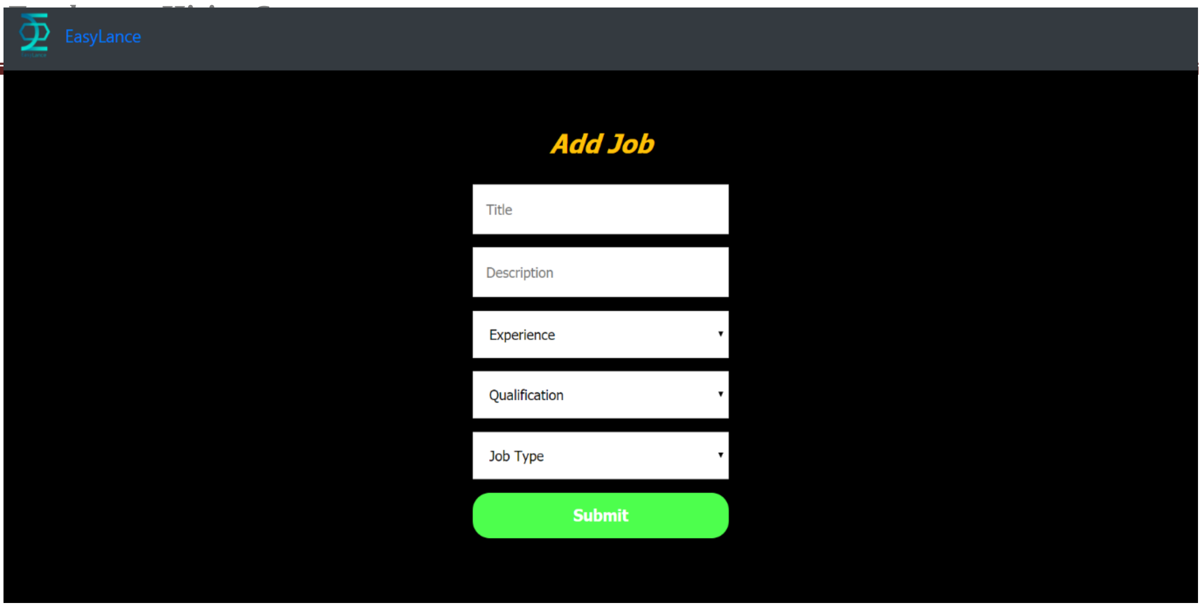
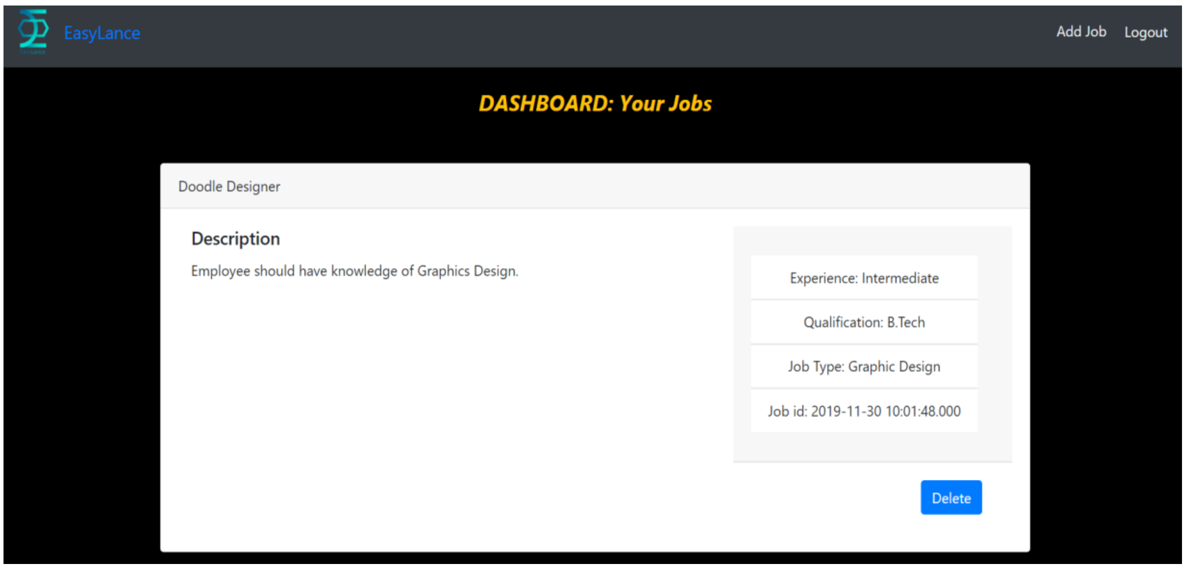


Fig 6.7: Adding Job Page



Doodle Designer

Description

Employee should have knowledge of Graphics Design.

Experience: Intermediate

Qualification: B.Tech

Job Type: Graphic Design

Job id: 2019-11-30 10:01:48.000

Fig 6.8: Dashboard

Chapter 7

CONCLUSION

We have successfully implemented the proposed model. The project is designed keeping in view the day to day problems faced freelancers.

Deployment of our system will certainly help users the most.

7.1 Limitations of the project

Regarding the design constraints of all the changes that can be brought to the system can be done only through the admin. Hence admin is the backbone of the system. Some key limitations are:

User should have internet access.

REFERENCES

1. MySQL

- http://php.net/manual/en/book.mysql.php
- https://stackoverflow.com

2. HTML

- https://www.w3schools.com/html/default.asp
- https://stackoverflow.com

3. CSS

- https://www.w3schools.com/css/default.asp
- https://stackoverflow.com

4. Project References

- https://www.hellobonsai.com/blog/whats-a-freelance-management-system
- https://www.forbes.com/sites/jonyounger/2018/10/11/fifteen-important-trends-in-freelancing-why-this-matters-now/#64c159343c10
- https://www.hcmworks.com/blog/freelance-management-systems-explained-what-they-do-and-how-they-work
- https://www.upwork.com/hiring/enterprise/freelancer-management-system-care/
- https://shortlist.co/what-is-a-freelancer-management-system-fms/