



# FIND JOB PROJECT REPORT

A PROJECT SUBMITTED TO

# Noble University

Submitted In Partial Fulfillment Of The Requirements For The Degree Of

MASTER OF COMPUTER APPLICATION

Sem – 2<sup>nd</sup>

Submitted By

ABHI SHETH

(Enrolment Number Is :: 220441020)

Under The Esteemed Guidance Of

“Mr. Ezaz Shaikh”

Faculty Of MCA

Noble University

# Certificate

This is to certify that the Project report entitled **FIND JOB** is a bonafide record of research work done by **SHETH ABHI BHAVINBHAI** Enrollment No.: **220441020** under my supervision and submitted to **Noble University** in partial fulfilment for **MASTER OF COMPUTER APPLICATION (MCA) SEMESTER 2.**

**Signature of the Guide**

**Designation:**

## PREFACE

There is a wide difference between theory and practical. If one has only theoretical background of any subject, one would not succeed in own aim therefore it is necessary for any person to have acceptable practical knowledge of the concerned subject. As I know MCA is a course based on “Information Technology” and it is totally practical field. With only theoretical knowledge one can't be succeeded or one can't be on the peak position.

In the course of MCA designed by the “Noble University” they have taken full care of these things and designed the course in such a manor with which student can get theoretical and practical both type of knowledge perfectly. According to the rules & regulation of “Web Development”, I have a subject named “Find Job”. In which we have to create a web project that provide information about any institute or industry.

As a MCA student, I have gathered general information about Numbers of Jobs from Difference Company. Then I decided to develop the site for that. In this site you can start Finding Best Jobs. You can also put the inquiry for any Jobs in which you are interested to get Job, on this site.

In this project report I have covered all the information, which is required for the web project of MCA student.

I have tried as my best present this project report in such a way that it makes easy to understand the project work.

## FIND JOB

### ACKNOWLEDGEMENT

I am thankful to all, who have helped me in preparing this project. I am very much happy to present this “Project Report”. Before you, expecting that you will acknowledgement it. It is a matter of great pleasure for me that I had an opportunity to express my view on the same.

As a part of my academic study as the student of 2<sup>nd</sup> semester of MCA I am required to experienced training software project an institute or industry in order to obtain practical knowledge and broaden me regarding the same.

At first, I would like to express my& humble thanks & gratitude to the who has provided me such a great, Co-operative & progressive environment.

Secondly at this moment, I would like to express my deepest sense of gratitude to my professor as well as project guides and who have contributed their precious time for the purpose of giving me the correct information with special interest & guidance throughout my project work.

I am also thankful to my classmate and few others who helped me directly or indirectly in solving problem & in making my software project more efficient & good working.

## FIND JOB

## INDEX

Ch. No.	Title	Page No.
1	Introduction and Objects of Project	1,2
2	Tools or Platform	3
3	System Analysis: 3.1 Identification of needs 3.2 Preliminary investigation 3.3 Feasibility study 3.4 Project planning 3.5 Project scheduling 3.6 Software requirement specification 3.7 Data models 3.8 Normalization	4 to 37 4 to 7 8,9 10 to 12 13 14,15 16 to 21 22 to 35 36,37
4	System Design: 4.1 Tables details 4.2 Database structure 4.3 User interface	38 to 59 38 39 to 41 42 to 59
5	Coding	60 to 71
6	Testing: 6.1 Techniques and strategies 6.2 Cost estimation model and SDLC Models 6.3 Future scope and further Enhancement of the project 6.4 Bibliography 6.5 Appendices 6.6 Glossary	72 to 84 72 to 75 76 to 84 85 85 86 86

# FIND JOB

## Chapter – 1

### Introduction and Object of Project

#### INTRODUCTION

This project is specially designed for Job Seeker And Company because in this project Patients Can Find And Make online Jobs , Apply Jobs ,Create Jobs and many more and also can registered as a User And Company in this project “**Job Management System**”.

Job Seeker can find their best Jobs , Apply Jobs And take information about Companies through our project “**Job Management System**”.

The “**Job Management System**” project is a multi-user system.

It has been developed in a way that allows user to perform the function smoothly and with proper accuracy.

The system is developed in **HTML, CSS, PHP, Bootstrap** and some other software .

So, this system is very useful for all types of Users And Companies.

# FIND JOB

## HOW IT WORKS

In this project first of all a homepage is displayed for all visitor, Seeker as well as registered Seeker .

This page will give information about **Job Management System** details etc.

If user wants to Apply, then they have to Login first. If the user is not a Registered User, He / She will have to click on the Signup Button and fill all fields in the registration form.

After registering in “**Job Management System**” there you can see login button.

Users will have to login with registered username and password.

After login, user will see User-Panel of “**Job Management System**” Website.

There is various information about Profile , Jobs etc.

# FIND JOB

## Chapter – 2

### Tools or Platform

#### Requirements Details

❖ SOFTWARE: -

- ♣ XAMPP Server
- ♣ Any Browser Like (Google Chrome, Internet Explorer, Mozilla etc.)

❖ HARDWARE: -

- ♣ Processor: P3 or higher
- ♣ RAM: 512 MB or higher
- ♣ Hard Disk: 50 GB or higher

❖ TOOLS USED: -

- ♣ Front End: HTML, CSS, JavaScript, Bootstrap
- ♣ Back End: PHP
- ♣ Operating System: Windows xp or higher

## Chapter – 3

### System Analysis

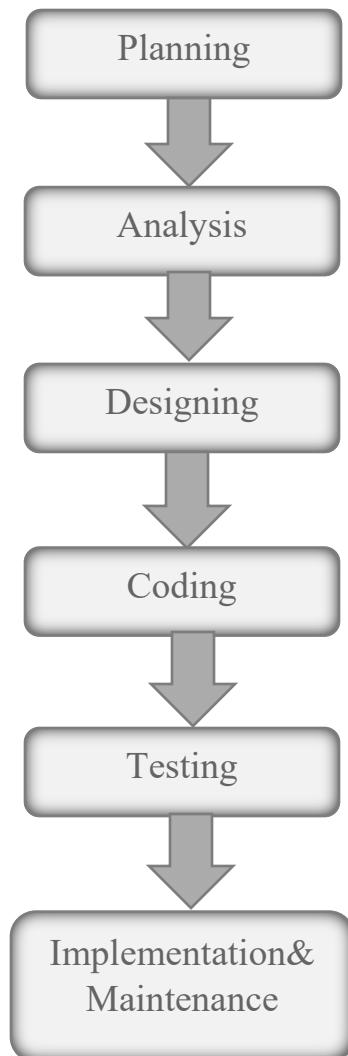
- 3.1 Identification of needs
- 3.2 Preliminary investigation
- 3.3 Feasibility Study
- 3.4 Project Planning
- 3.5 Project Scheduling (PERT Chart and Gantt chart)
- 3.6 Software Requirement Specification
- 3.7 Data Models

#### 3.1 - Identification of Needs

- The viewer of our system does not need to login as long as User wants to get Information only. But User wants to view health record and many more things with Sunshine Hospital they must be registered to our website.
- User need to provides user name, e-mail and password when creating an account.

# FIND JOB

## SDLC MODEL



# FIND JOB

## SDLC STEPS

- Requirement Gathering
- Requirement Specification and Analysis
- Design
- Coding
- Maintenance

### REQUIREMENT GATHERING: -

In this phase of SDLC necessary information are collected. We collect the information in this phase through questionnaire and online websites also. We are also note down the requirements of the User.

### REQUIREMENT SPECIFICATION AND ANALYSIS: -

The requirements are being specify in this phase. Analyst checks how the current system is working and then plans how to develop the proposed system and implement it.

# FIND JOB

## DESIGN: -

In this phase the design of both front-end and back-end is decided. In this phase we using HTML, CSS, BOOTSTRAP for designing, and designing of tables are also done properly in this phase.

## CODING: -

After the completion of designing there is a turn for coding. Coding is completed in various files. In most case coding takes the more time than the design.

## TESTING: -

In testing phase first unit level testing is done. In unit level testing the software is tested in an individual file. After the unit testing of each file the integration testing done. In integration testing the files are tested into their own package. At last, in the system testing the whole system is run on the INTERNET ENVIRONMENT.

## MAINTENANCE: -

The proper maintenance is given as indicate in the SRS (Software Requirement Specification). It is a responsibility of Developer to maintain the software during its criteria mentioned in SRS document.

# FIND JOB

## 3.2 - PRELIMINARY INVESTIGATION

### OUR OBJECTIVES: -

- Admin only add Company, Delete and Edit records.
- User can browse category wise.
- User can handle all type of information easily
- Easy updating, time saving, quick process.
- User can update their registration details.
- User can get guidance of whole Find Job different steps.

## PLATFORM SPECIFICATION: -

- We have used windows based developing graphical based applications our software will provide support GUI efficiently. It must for our system because for our project.
- It is very useful and support of PHP and Java Script control and component.
- Use of Table for menu create and display Form in it.
- It is easy to integrity database report use with Data Report.
- PHP platform is mostly use for development Web Based Application
- Regular version of operating system use with it.
- Always need to Back end tool for developed web Application with front tool.
- Starting a new business all of the organization have some requirement for it.
- There are many reasons for starting a new business or modify the old system.
- Business require set is depending on top level organization that is manager, administrator etc.

## 3.3 - FEASIBILITY STUDY

- A feasibility analysis usually involves a thorough assessment of the operational (need), financial and technical aspects of a proposal.
- Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints.
- A feasibility study should be relatively cheap and done at the earliest possible time.
- Depending on the study, the decision is made whether to go ahead with a more detailed analysis.
- When a new project is proposed, it normally goes through feasibility assessment.
- Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.
- Facts considered in the feasibility analysis were.

# FIND JOB

## 1) TECHNICAL FEASIBILITY: -

- Technical feasibility is considered. In terms of technical requirements and their availability in the market. It determines whether the current level of technology supports the proposed system or not. The technical possibility of proposed system is as follows.
- The unit does possess the hardware as well as related software for the project.
- The proposed system does not require much technical detail.
  - ♣ It just requires window operating system.
  - ♣ The organization has already purchased all the enough devices for latest technical.
  - ♣ These technical specifications are easily available in the market.
- Hence, the proposed system is technically feasible.

## 2) ECONOMICAL FEASIBILITY STUDY: -

- Economic justification is generally the “Bottom Line” consideration for most systems.
- Economic justification includes a broad range of concerns that includes cost benefit analysis.
- In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e. profit making, the project is making to the analysis and design phase.
- The financial and the economic questions during the preliminary investigation are.

### **Verified to Estimate the Following:**

- ♣ The cost to conduct a full system investigation.
- ♣ The cost of hardware and software for the class of application being considered.
- ♣ The benefits in the form of reduced cost.
- ♣ The proposed system will give the minute information; as a result, the performance is improved which in turn may be expected to provide increased profits.
- This feasibility checks whether the system can be developed with the available funds. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of man hours required.

### **3) OPERATIONAL FEASIBILITY STUDY: -**

- It is mainly related to human organizations and political aspects. The points to be considered are:
  - ♣ What changes will be brought with the system?
  - ♣ What organization structures are disturbed?
  - ♣ What new skills will be required?
  - ♣ Do the existing staff members have these skills? If not, can they be trained in due course of time?
- The system is operationally feasible as it very easy for the End users to operate it. It only needs basic information about Windows platform.

# FIND JOB

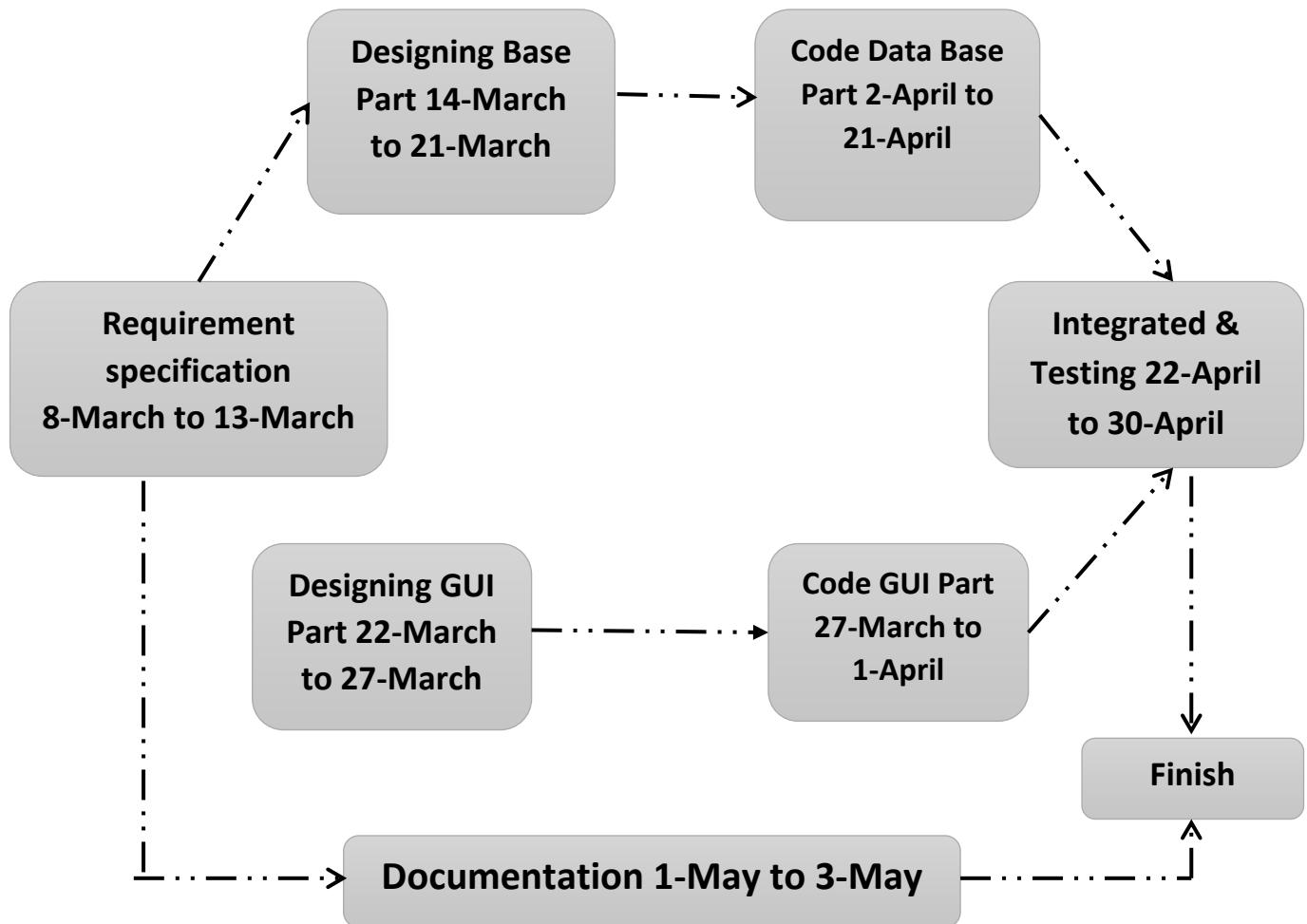
## 3.4 - PROJECT PLANNING

No.	Task Name	Start	Finish	Duration
1.	Planning	08-March	13- March	6 Days
2	Analysis	14- March	21- March	8 Days
3	Designing	22- March	01- April	11 Days
4	Coding	02- April	21- April	20 Days
5	Implementation	22- April	26- April	5 Days
6	Testing	27- April	30- April	4 Days
TOTAL				54 DAYS

# FIND JOB

## 3.5 - PROJECT SCHEDULING

**PERT CHART:** - This chart represents the development of our system Date wise



# FIND JOB

## ■ GANTT CHART: -

This is a graphical representation of the date wise development of our project system. It is very similar to the PERT chart except that it is represented graphically.

ID	TASK NAME	March 23	March 23	March/April 23	April/May 23
1	Planning	08-March to 13- March			
2	Analysis		14- March to 21- March		
3	Designing			22- March to 01- April	
4	Coding			02- April to 21- April	
5	Implementation				22- April to 26-April
6	Testing				27-April to 30-April

## 3.6 - SOFTWARE REQUIREMENT

■ PHP: -



- PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.
- PHP generally runs on a web server, taking PHP code as its input and creating Web pages as output.
- However, it can also be used for command-line scripting and client-side GUI applications.
- PHP can be deployed on most web servers and on almost every operating system and platform free of charge.
- The PHP Group also provides the complete source code for users to build, customize and extend for their own use.
- PHP primarily acts as a filter. The PHP program takes input from a file or stream containing text and special PHP instructions and outputs another stream of data for display.
- From PHP 4, the PHP parser compiles input to produce byte code for processing by the Send Engine, giving improved performance over its interpreter predecessor. PHP 5 uses the Send Engine II.
- Originally designed to create dynamic web pages, PHP's principal focus is server-side scripting.

# FIND JOB

■ My SQL: -



- MySQL is a database management system.
  - ♣ A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.
- MySQL databases are relational.
  - ♣ A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

## FIND JOB

- ♣ The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.
- ♣ SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist.
- *MySQL software is Open Source.*
  - Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), <http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us.
- *The MySQL Database Server is very fast, reliable, scalable, and easy to use.*
- *MySQL Server works in client/server or embedded systems*

## JAVA Script: -

### • JAVA SCRIPT ORIGINS: -

- JavaScript was released by Netscape and Sun Microsystems in 1995.
- However, JavaScript is not the same thing as Java.
  - It is a programming language.
  - It is an interpreted language.
  - It is object-based programming.
  - It is widely used and supported
  - It is accessible to the beginner.

### • USES OF JAVA SCRIPT: -

- Use it to add multimedia elements with JavaScript you can show, hide, change, resize images, and create image rollovers. You can create scrolling text across the status bar.
- Create pages dynamically Based on the user's choices, the date, or other external data, JavaScript can produce pages that are customized to the user.

# FIND JOB

## HTML: -

- **HTML** is the standard Mark-up language for creating web pages and CSS and JavaScript it forms a triad of cornerstone technologies for the WWW.
- Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages.
- HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
- **Hyper Text Mark-up Language (HTML)** can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages.
- Inclusion of CSS defines the look and layout of content. The World Wide Web consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## CSS: -

- **Cascading Style Sheet (CSS)** is a Style sheet language used for describing the presentation of a document written in a Mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.
- CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts.

## BOOTSTRAP: -

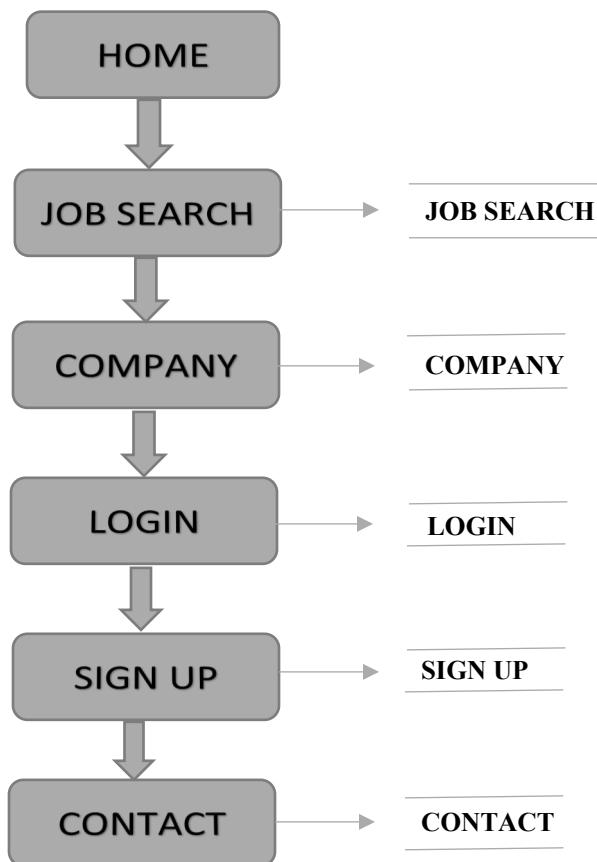
- Originally created by a designer and a developer at Twitter, Bootstrap has become one of the most popular front-end frameworks and open source projects in the world.
- Bootstrap was created at Twitter in mid-2010 by @mdo and @fat. Prior to being an open-sourced framework, Bootstrap was known as Twitter Blueprint. A few months into development, Twitter held its first Hack Week and the project exploded as developers of all skill levels jumped in without any external guidance. It served as the style guide for internal tools development at the company for over a year before its public release, and continues to do so today.
- Originally released on Friday, August 19, 2011, we've since had over twenty releases, including two major rewrites with v2 and v3. With Bootstrap 2, we added responsive functionality to the entire framework as an optional stylesheet. Building on that with Bootstrap 3, we rewrote the library once more to make it responsive by default with a mobile first approach.
- With Bootstrap 4, we once again rewrote the project to account for two key architectural changes: a migration to Sass and the move to CSS's flexbox. Our intention is to help in a small way to move the web development community forward by pushing for newer CSS properties, fewer dependencies, and new technologies across more modern browsers.

# FIND JOB

## 3.7 – DATA MODEL

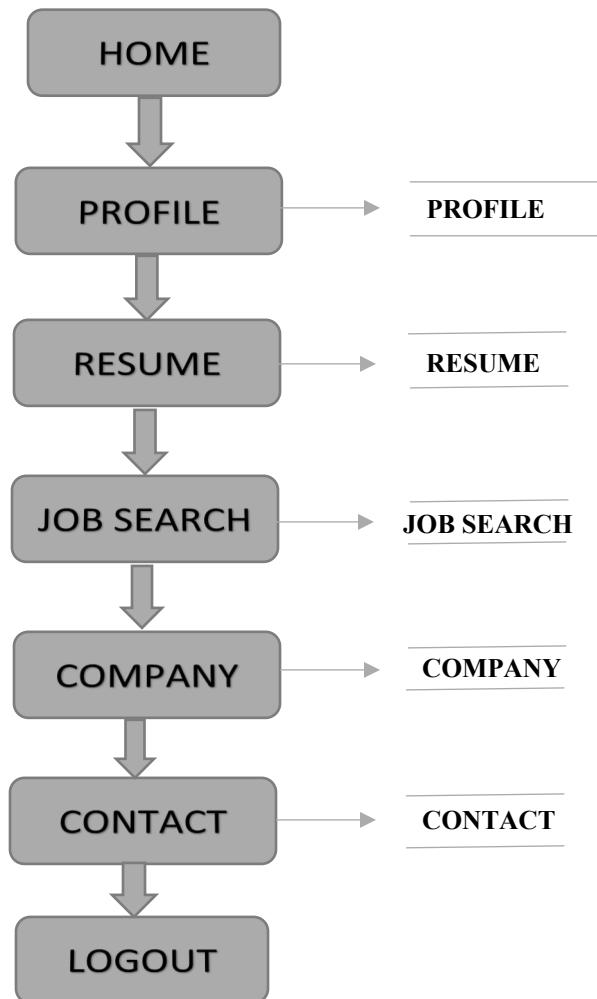
### Data Flow Diagram: -

#### 1) Visitor Data Flow Diagram: -



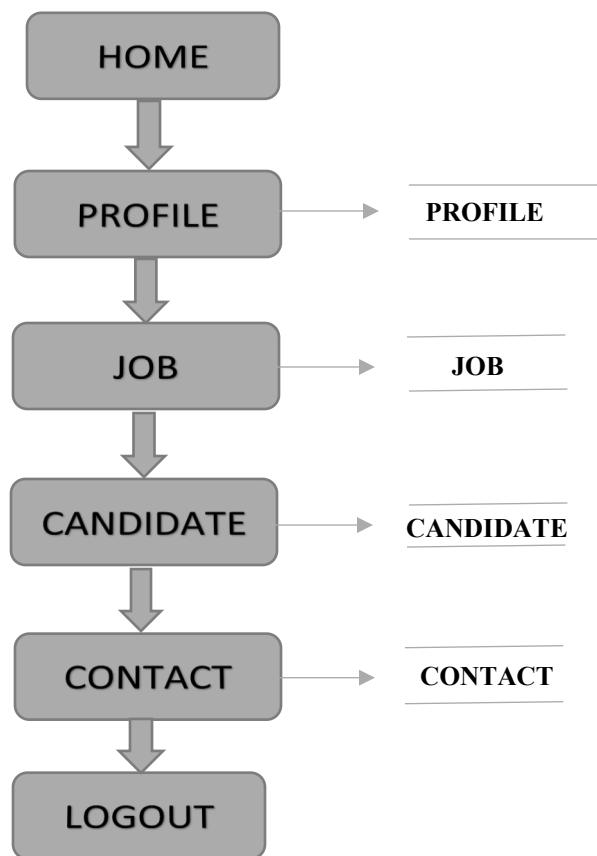
# FIND JOB

## 2) User Data Flow Diagram:-



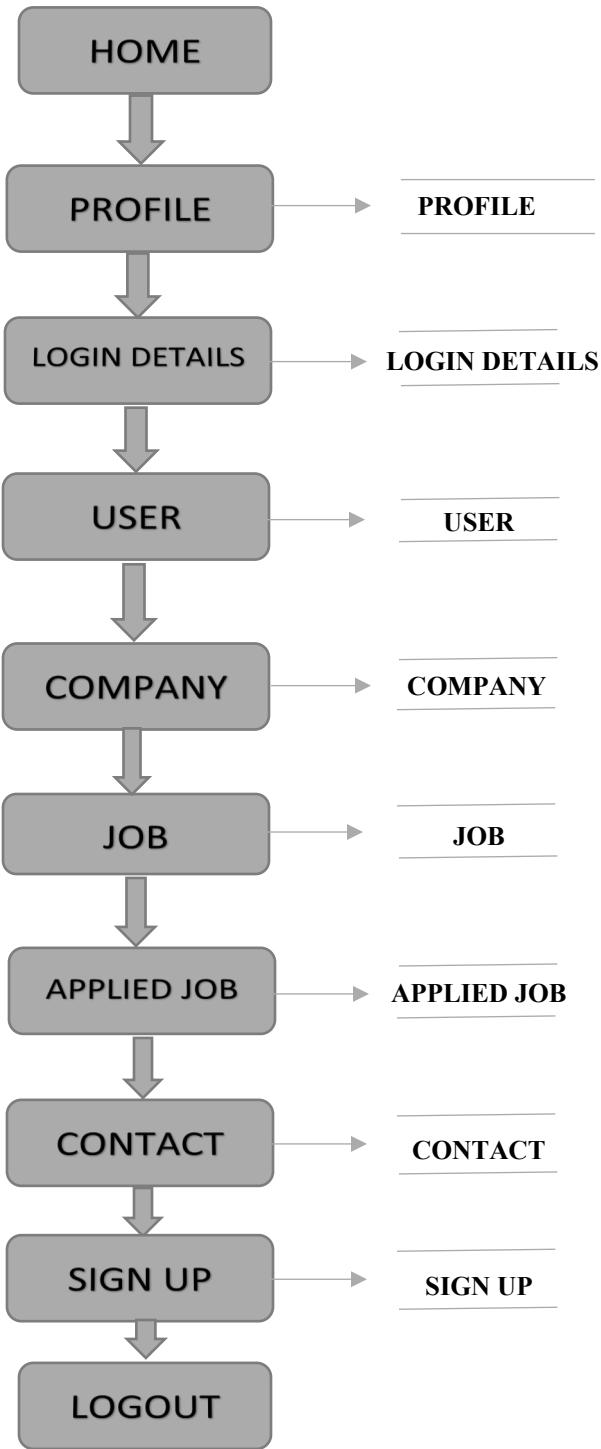
# FIND JOB

### 3) Company Data Flow Diagram :-



# FIND JOB

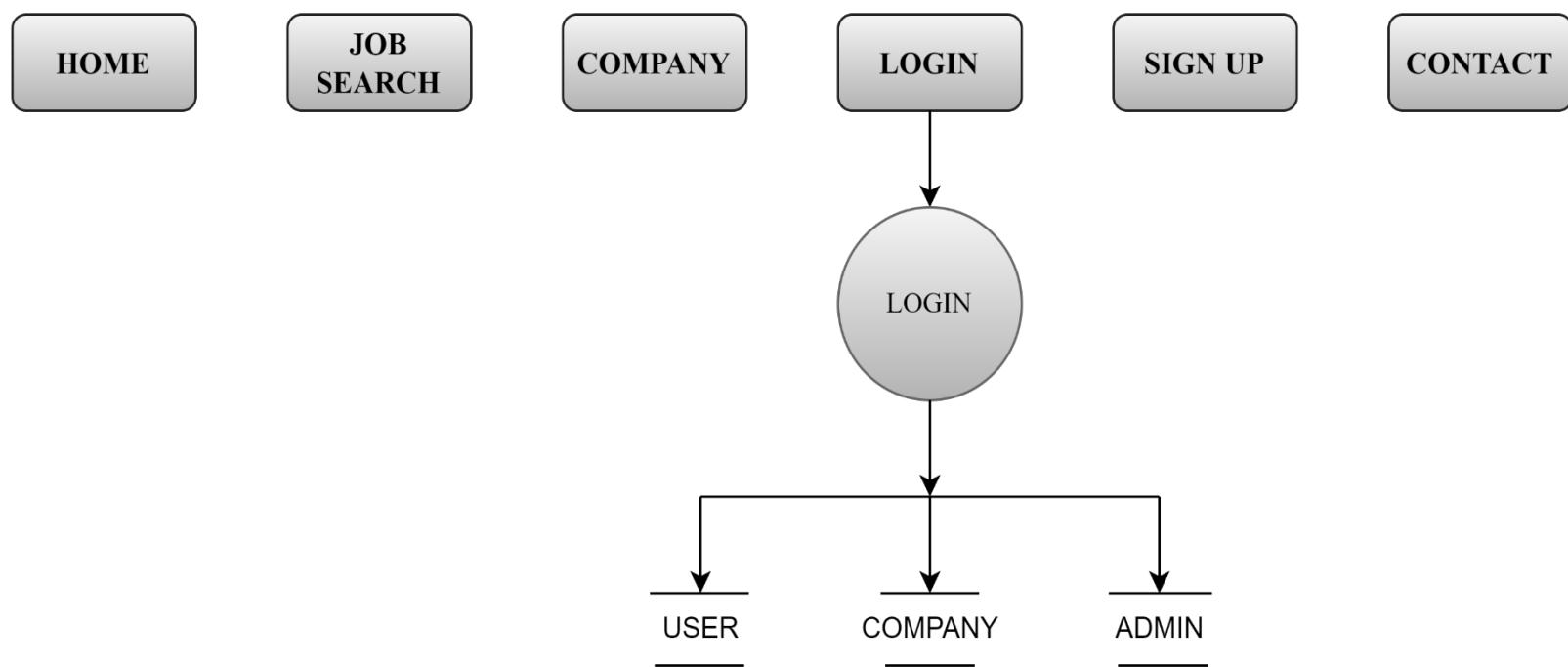
## 4) Admin Data Flow Diagram:-



# FIND JOB

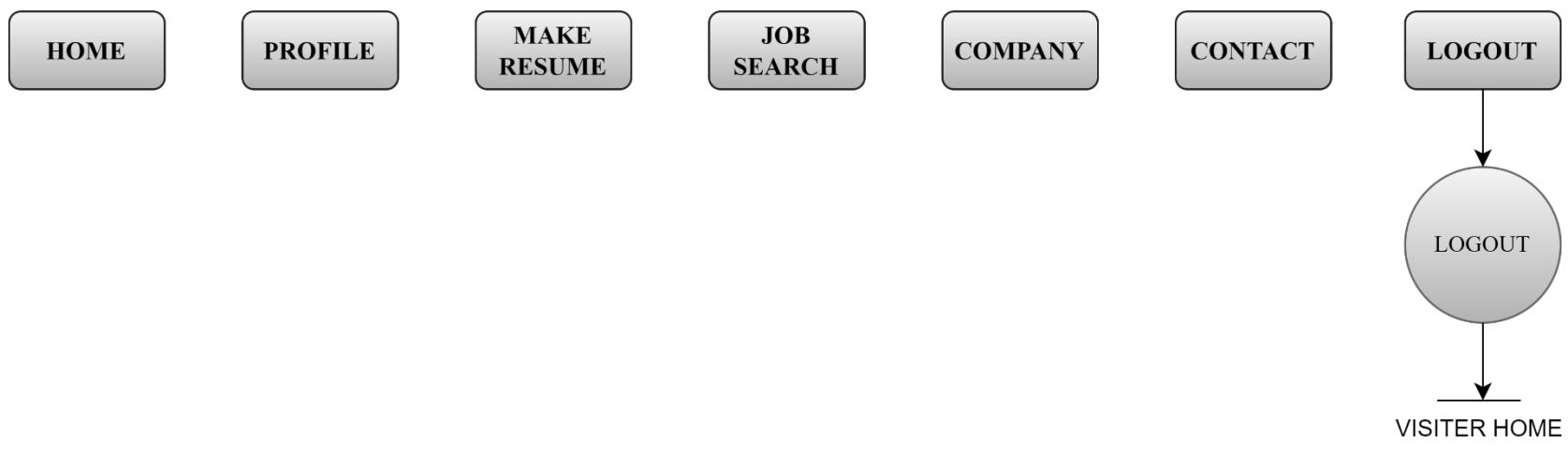
## ■ Data Flow Diagrams:

### 1) Visitor Data Flow Diagram:-



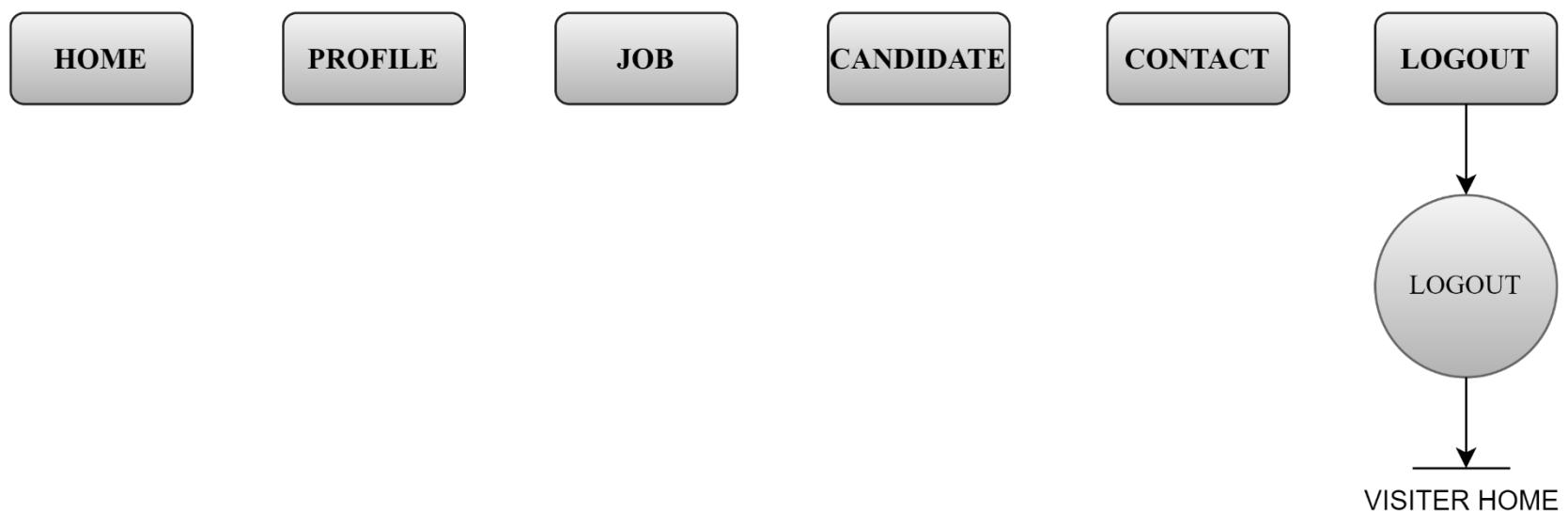
# FIND JOB

## 2) User Data Flow Diagram:



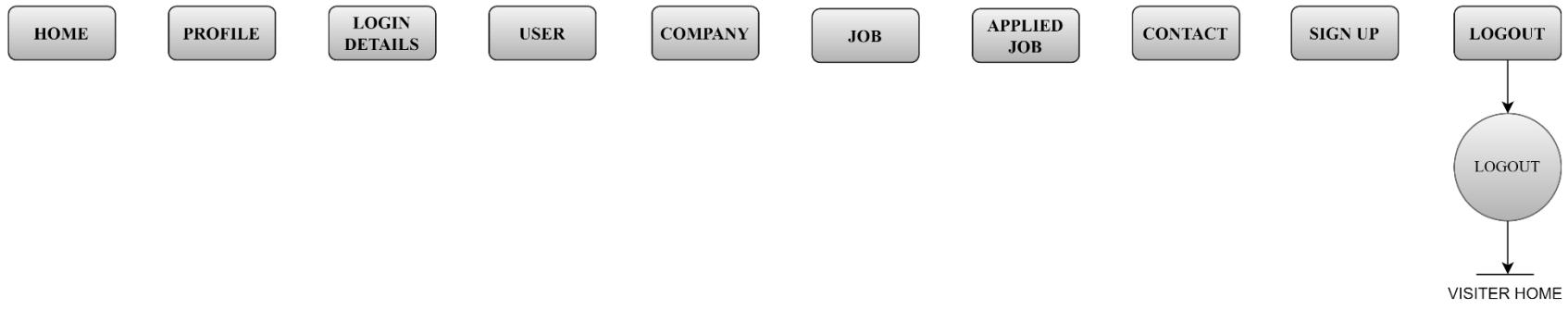
# FIND JOB

### 3) Company Data Flow Diagram:



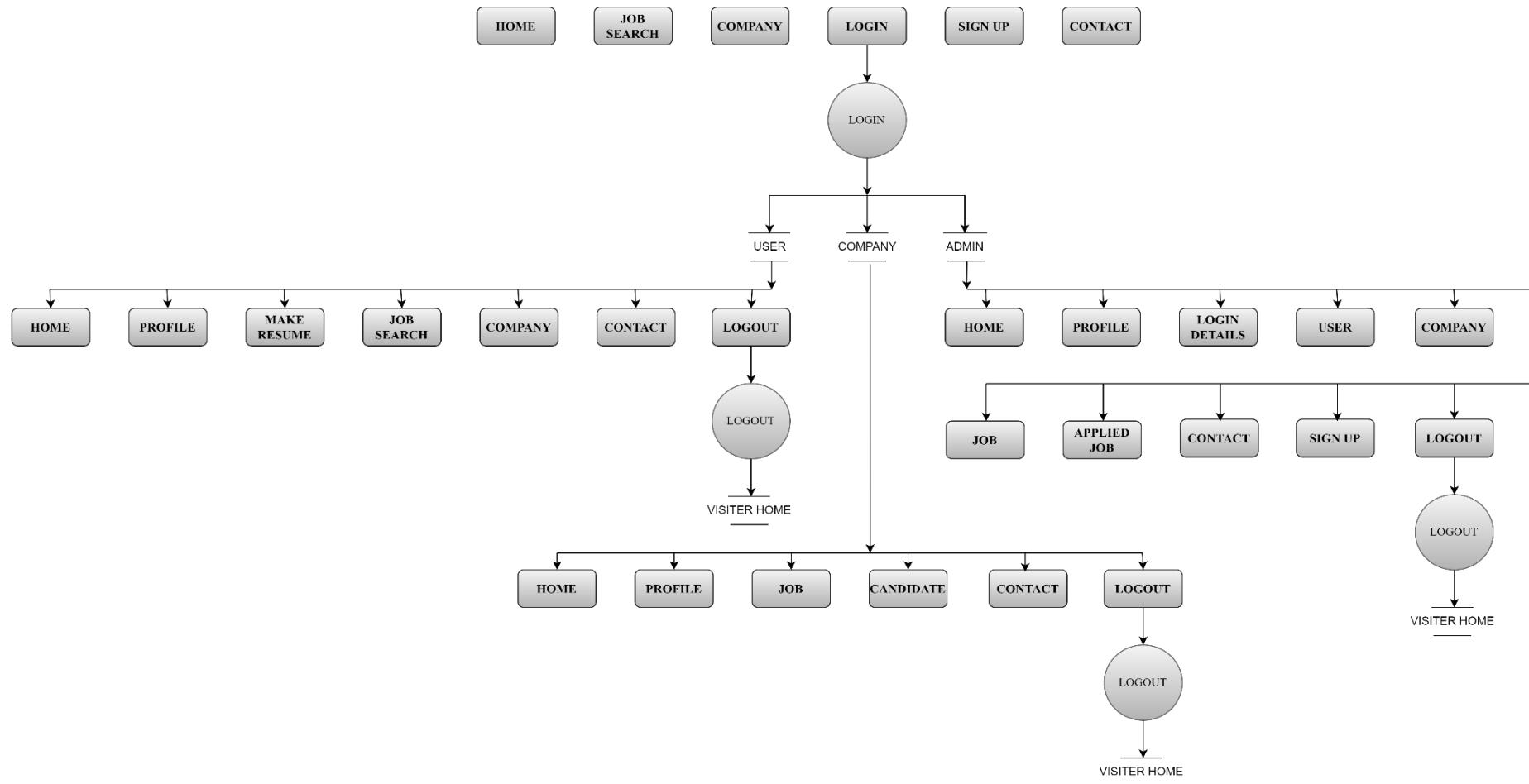
# FIND JOB

## 4) Admin Data Flow Diagram:



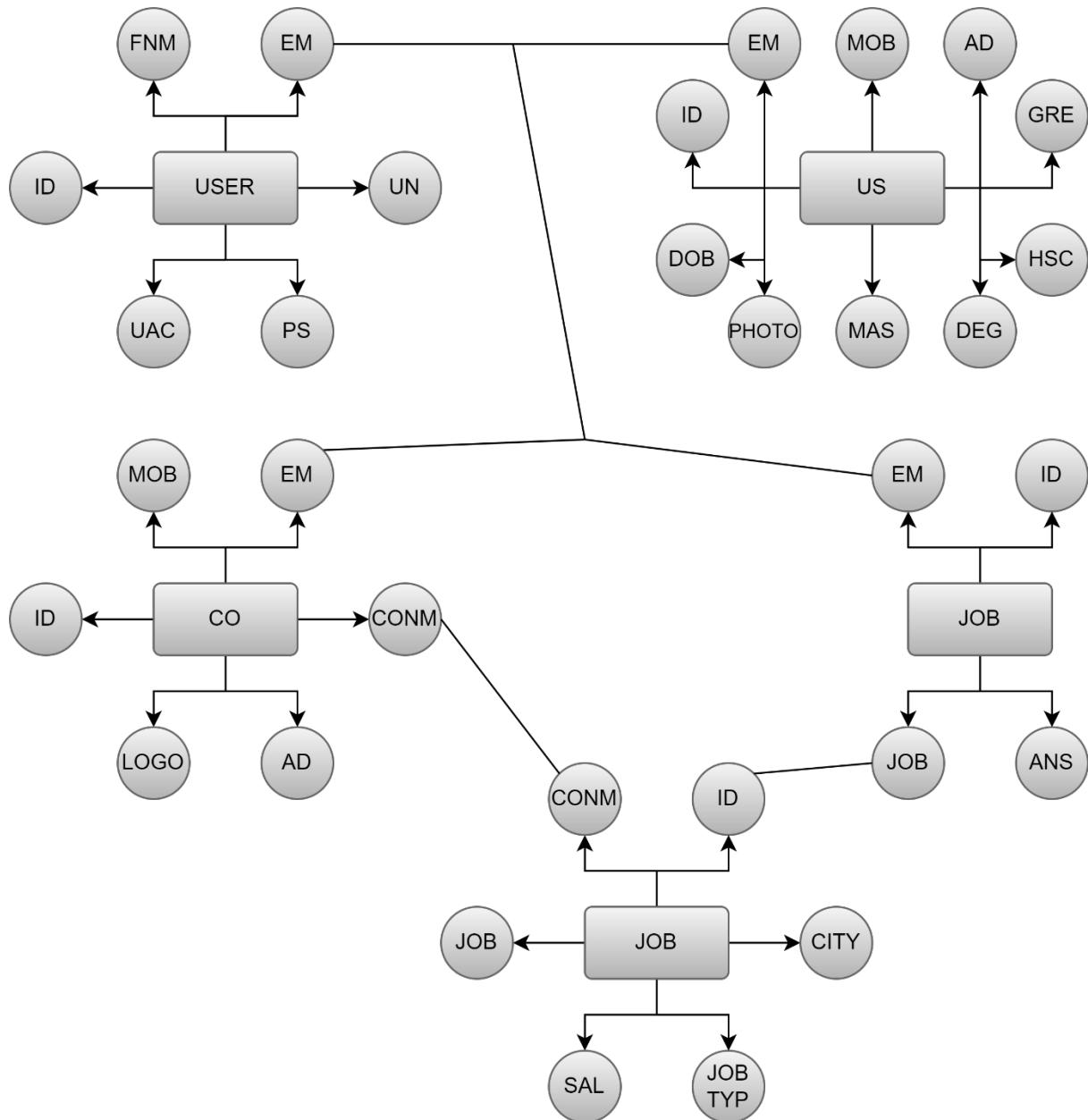
# FIND JOB

## 5) Visitor, User and Admin Data Flow Diagram:



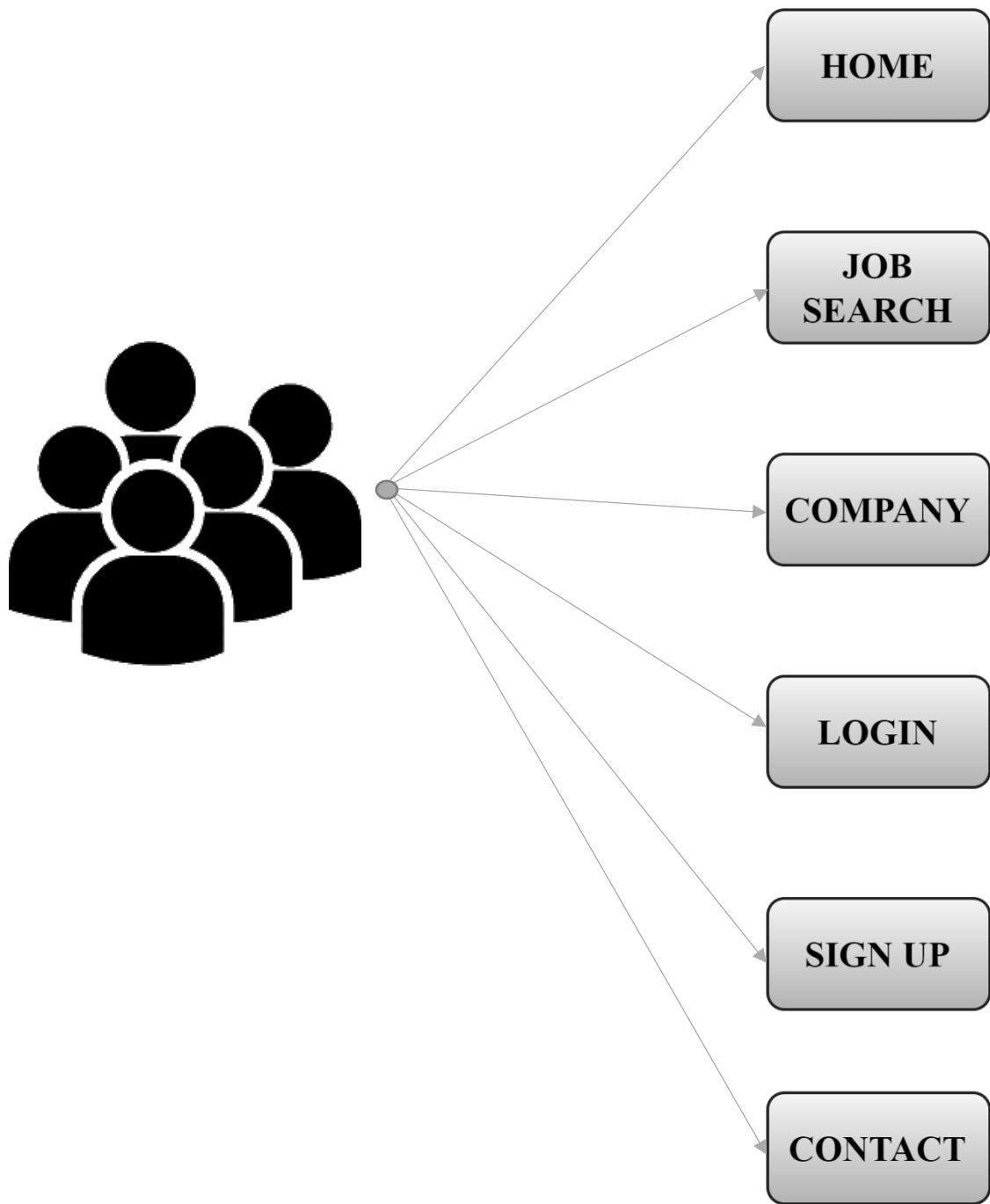
# FIND JOB

## ER Diagrams:



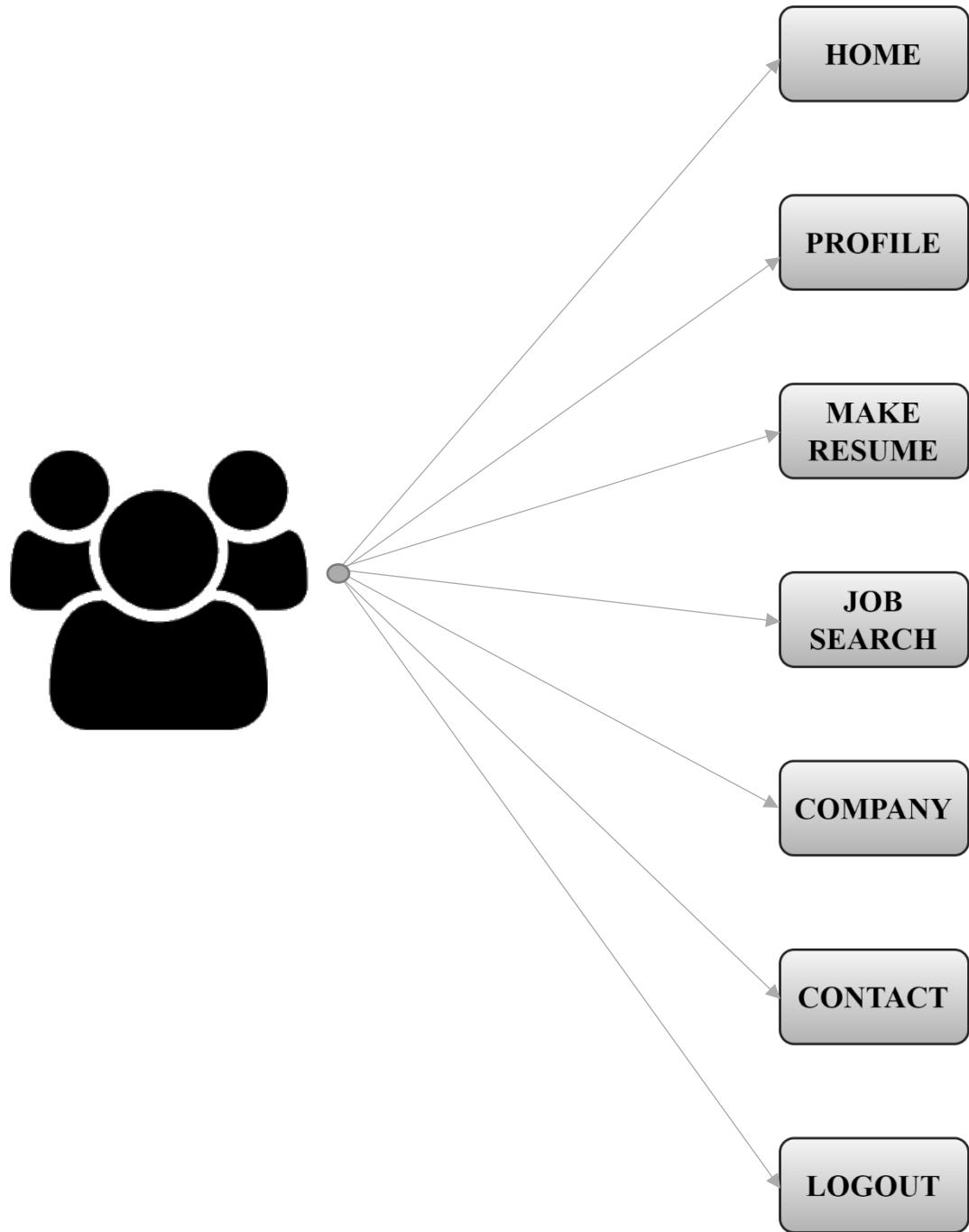
# FIND JOB

Visitor Use Case Diagrams:



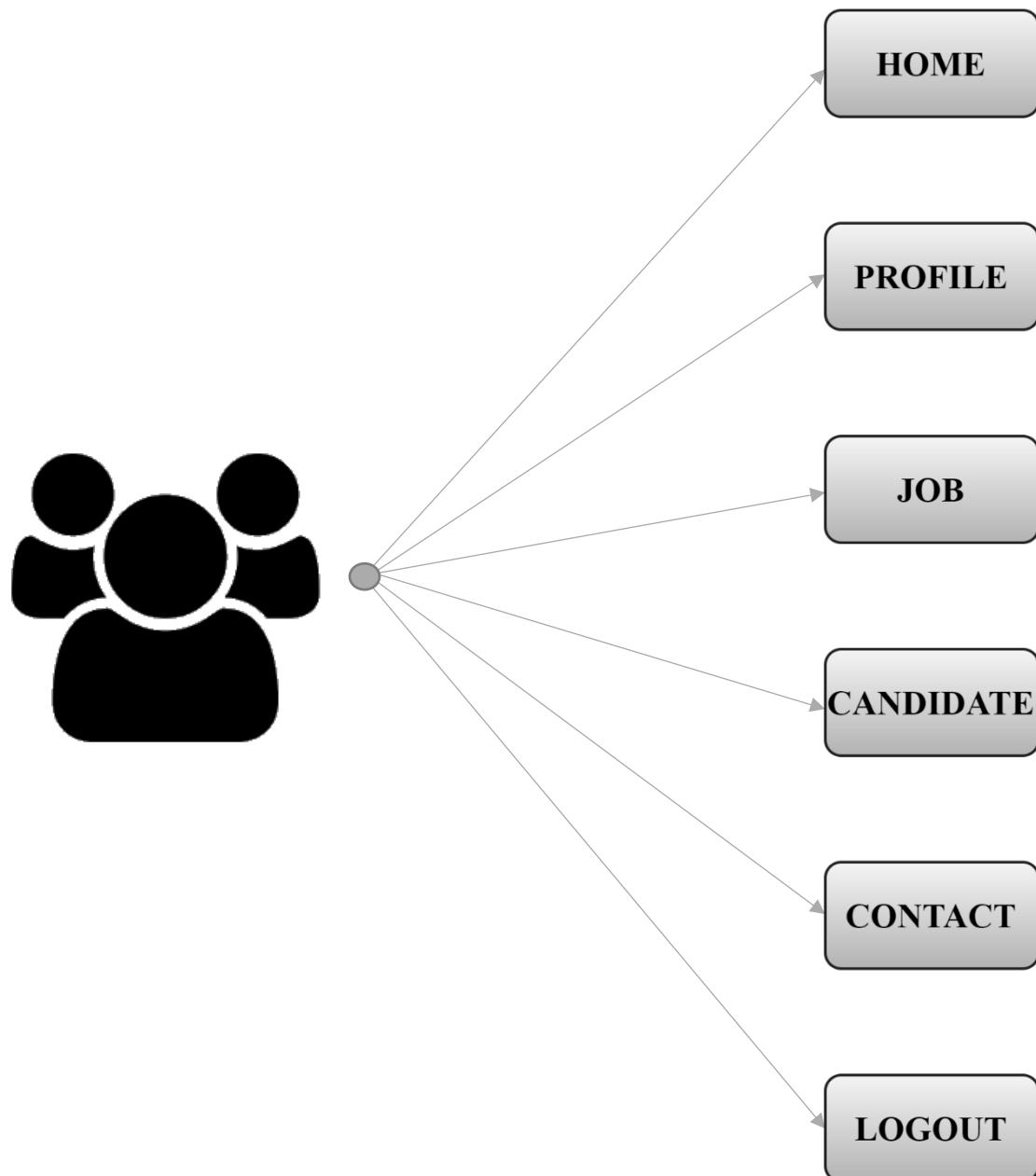
# FIND JOB

## User Use Case Diagrams:



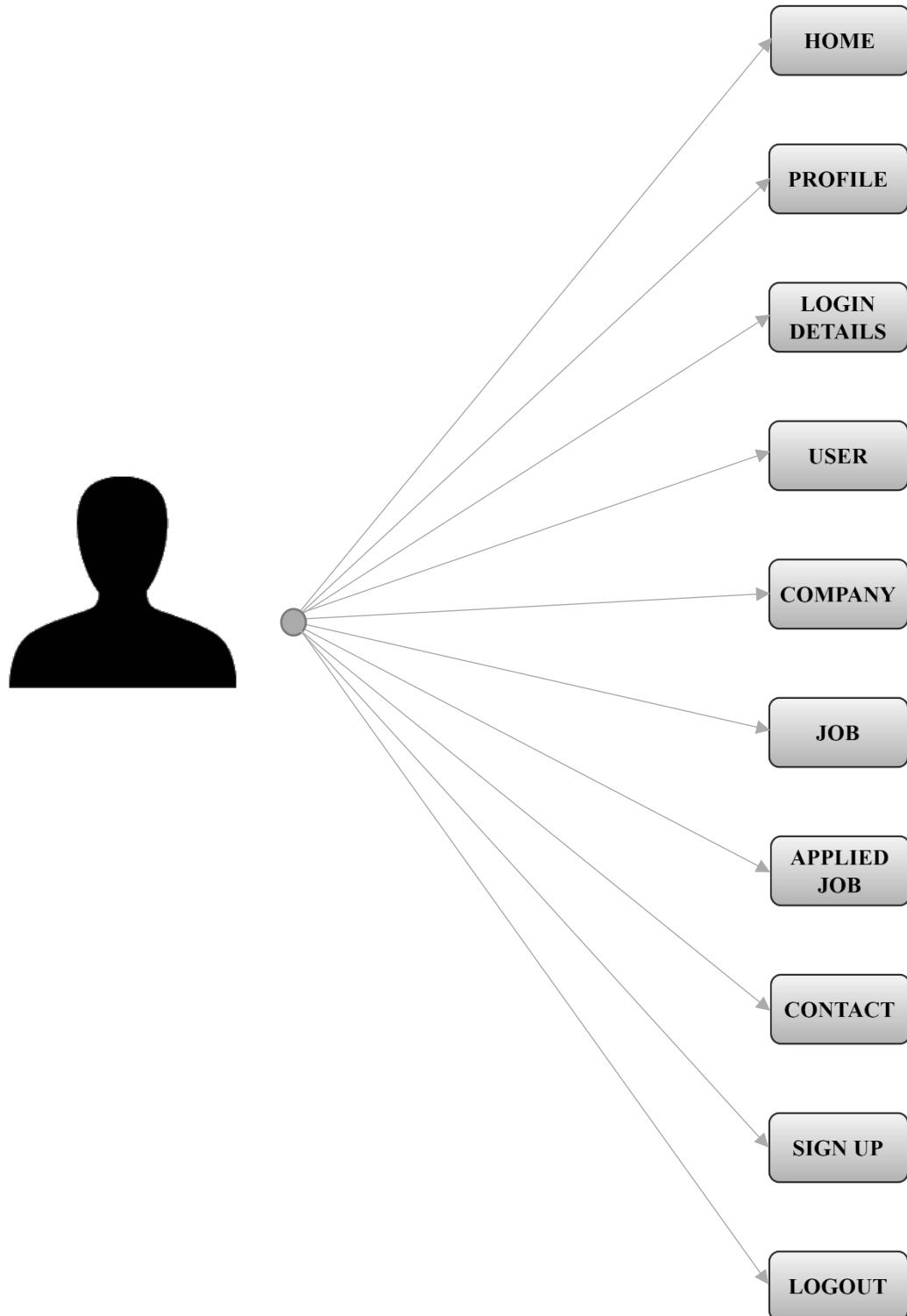
# FIND JOB

Company Use Case Diagrams:



# FIND JOB

## Admin Use Case Diagrams:



## 3.8 - NORMALIZATION

✚ **Normalization rules are divided into following normal form.**

1. First Normal Form
  2. Second Normal Form
  3. Third Normal Form
- 1) **First Normal Form (1NF):** - As per First Normal Form, no two Rows of data must contain repeating group of information id each set of columns must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique. The Primary key is usually a single column, but sometimes more than one column can be combined to create a single primary key. For example, consider a table which is not in First normal form.
- 2) **Second Normal Form (2NF):** - As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.  
In User Table the candidate key will be column. Now, both the above tables qualify for Second Normal Form and will never suffer from Update Anomalies. Although there are a few complex cases in which table in Second Normal Form suffers Update Anomalies, and to handle those scenarios Third Normal Form is there.

## FIND JOB

3) **Third Normal Form (3NF):** - Third Normal form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So, this transitive functional dependency should be removed from the table and also the table must be in Second Normal form. For example, consider a table with following fields.

In this table User id is Primary key, but street, city and state depend upon Zip. The dependency between zip and other fields is called transitive dependency. Hence to apply 3NF, we need to move the street, city and state to new table, with Zip as primary key.

→ The advantage of removing transitive dependency is,

- Amount of data duplication is reduced.
- Data integrity achieved.

# FIND JOB

## Chapter - 4

### SYSTEM DESIGN

4.1 Tables

4.2 Database Structure

4.3 User Interface

#### 4.1 - TABLES

Table	Action	Rows	Type	Collation	Size	Overhead
appjob	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	2	InnoDB	utf8mb4_general_ci	<a href="#">16.0 KiB</a>	-
co	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	2	InnoDB	utf8mb4_general_ci	<a href="#">32.0 KiB</a>	-
job	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	2	InnoDB	utf8mb4_general_ci	<a href="#">16.0 KiB</a>	-
us	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1	InnoDB	utf8mb4_general_ci	<a href="#">48.0 KiB</a>	-
user	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	4	InnoDB	utf8mb4_general_ci	<a href="#">48.0 KiB</a>	-
5 tables	Sum	11	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

# FIND JOB

## 4.2 – DATABASE STRUCTURE

1) login/Sign Up:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 	int(3)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	2 <b>fnm</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	3 <b>em</b> 	varchar(30)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	4 <b>un</b> 	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	5 <b>ps</b>	varchar(15)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	6 <b>uac</b>	varchar(1)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾

2) User :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 	int(3)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	2 <b>em</b> 	varchar(30)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	3 <b>mob</b>	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	4 <b>ad</b>	varchar(250)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	5 <b>graduation</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	6 <b>hsc</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	7 <b>deg</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	8 <b>mas</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	9 <b>photo</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
<input type="checkbox"/>	10 <b>dob</b>	date			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾

# FIND JOB

3) Company:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□ 1	<b>id</b> ↗	int(3)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 2	<b>conm</b>	varchar(40)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 3	<b>em</b> ↗	varchar(30)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 4	<b>mob</b>	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 5	<b>ad</b>	varchar(250)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 6	<b>logo</b>	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾

4) Job:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□ 1	<b>id</b> ↗	int(3)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 2	<b>conm</b>	varchar(40)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 3	<b>job</b>	varchar(40)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 4	<b>city</b>	varchar(40)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 5	<b>jobtyp</b>	varchar(100)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾
□ 6	<b>sal</b>	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> ▾

# FIND JOB

5) Apply Job

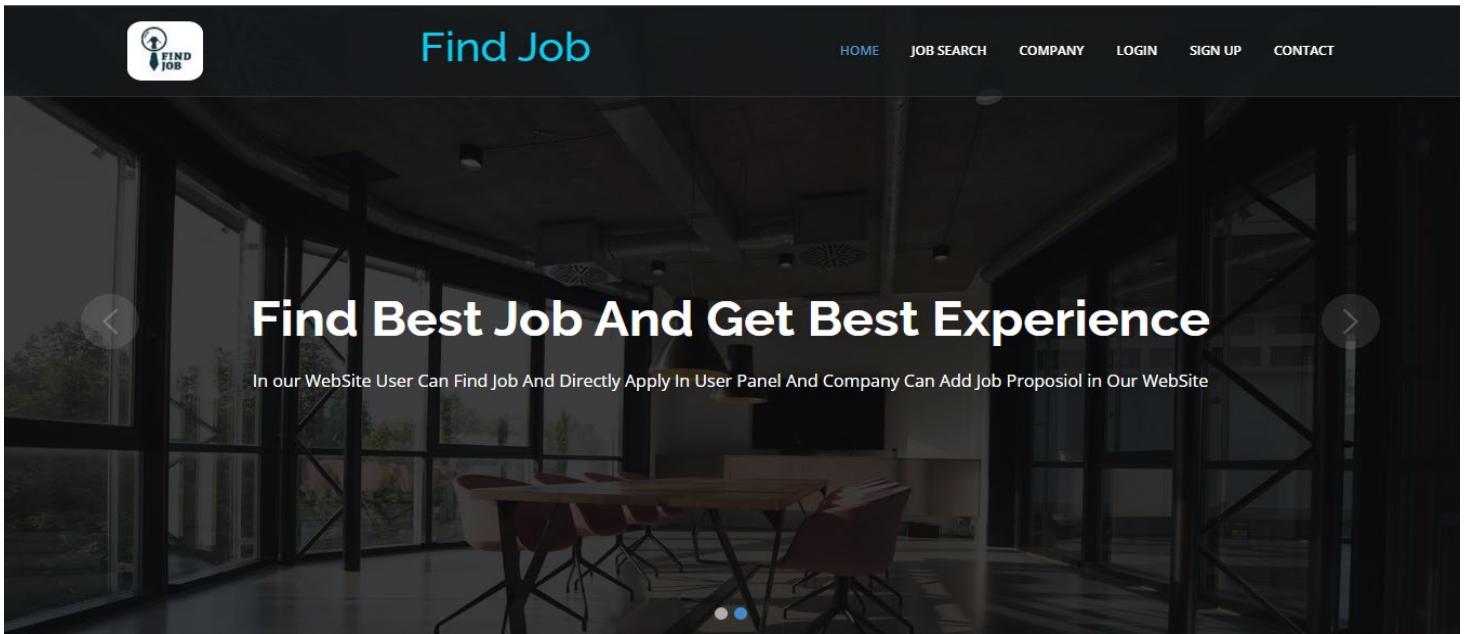
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<b>id</b> 	int(3)			No	None		AUTO_INCREMENT	 <a href="#">Change</a>  <a href="#">Drop</a> <a href="#">More ▾</a>
2	<b>em</b>	varchar(50)	utf8mb4_general_ci		No	None			 <a href="#">Change</a>  <a href="#">Drop</a> <a href="#">More ▾</a>
3	<b>job</b>	varchar(20)	utf8mb4_general_ci		No	None			 <a href="#">Change</a>  <a href="#">Drop</a> <a href="#">More ▾</a>
4	<b>ans</b>	varchar(20)	utf8mb4_general_ci		No	None			 <a href="#">Change</a>  <a href="#">Drop</a> <a href="#">More ▾</a>

# FIND JOB

## 4.3 – USER INTERFACE

### 1) Visitor Side Screenshots: -

#### ➤ Index Page:



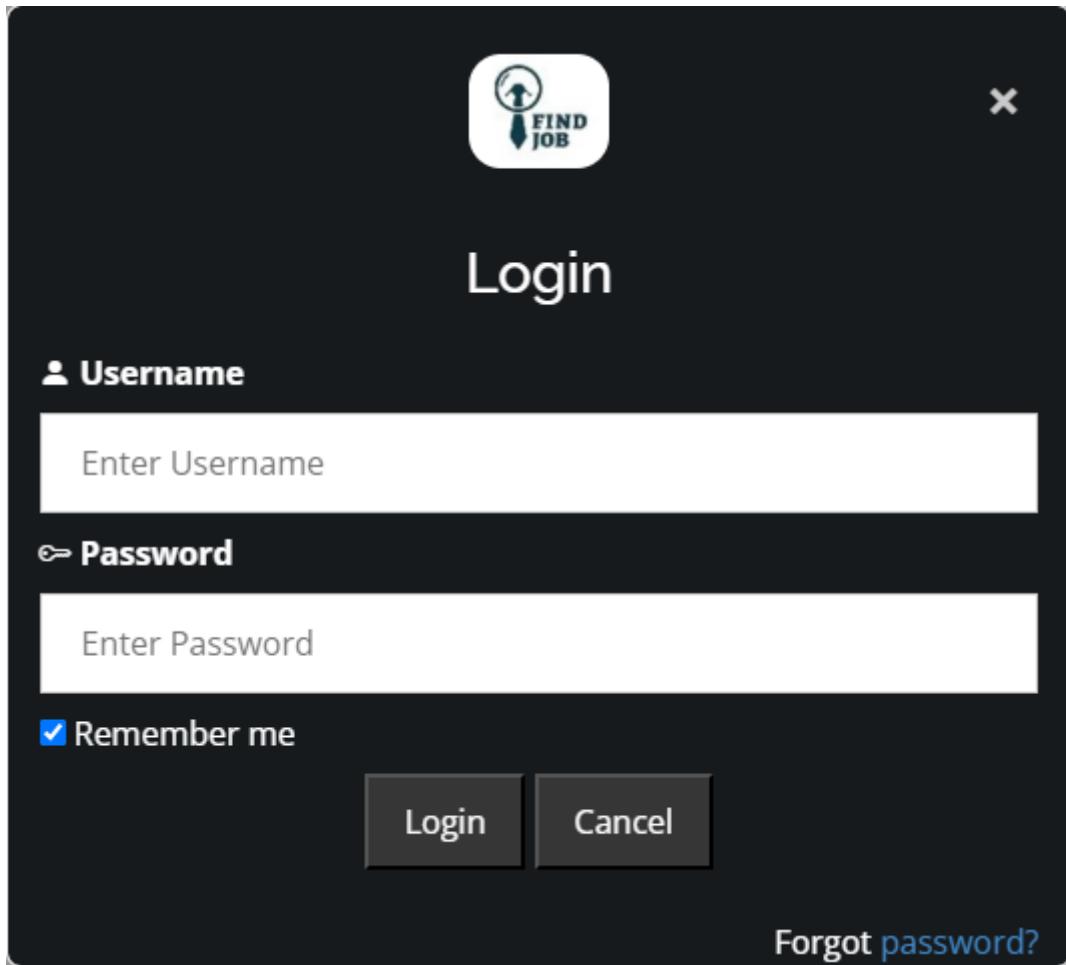
#### ➤ Job Search Page:

JOB SEARCH

COMPANY	Enter Comp.	JOB TYPE	Enter Jobtyp	CITY	Enter City	SALARY	Enter Salary
<input type="button" value="Search"/> <input type="button" value="Cancel"/>							
Company	JOB TYPE	CITY	JOB REQUIREMENT			SALARY	
TCS	PL/SQL DEVLOPER	BENGALURU	Regular Employee Years of Experience 6 to 10+ years			15000	
ORACLE	FULL STACK DEVLOPER	JUNAGADH	Regular Employee Years of Experience 6 to 10+ year			35000	

# FIND JOB

➤ Login Form:



The image shows a dark-themed login form titled "Login". At the top center is a logo featuring a stylized key or lock icon above the text "FIND JOB". In the top right corner is a white "X" button. Below the title, there are two input fields: one for "Username" and one for "Password", both with placeholder text "Enter Username" and "Enter Password" respectively. A "Remember me" checkbox is checked. At the bottom are "Login" and "Cancel" buttons, and a "Forgot password?" link.

Enter Username

Enter Password

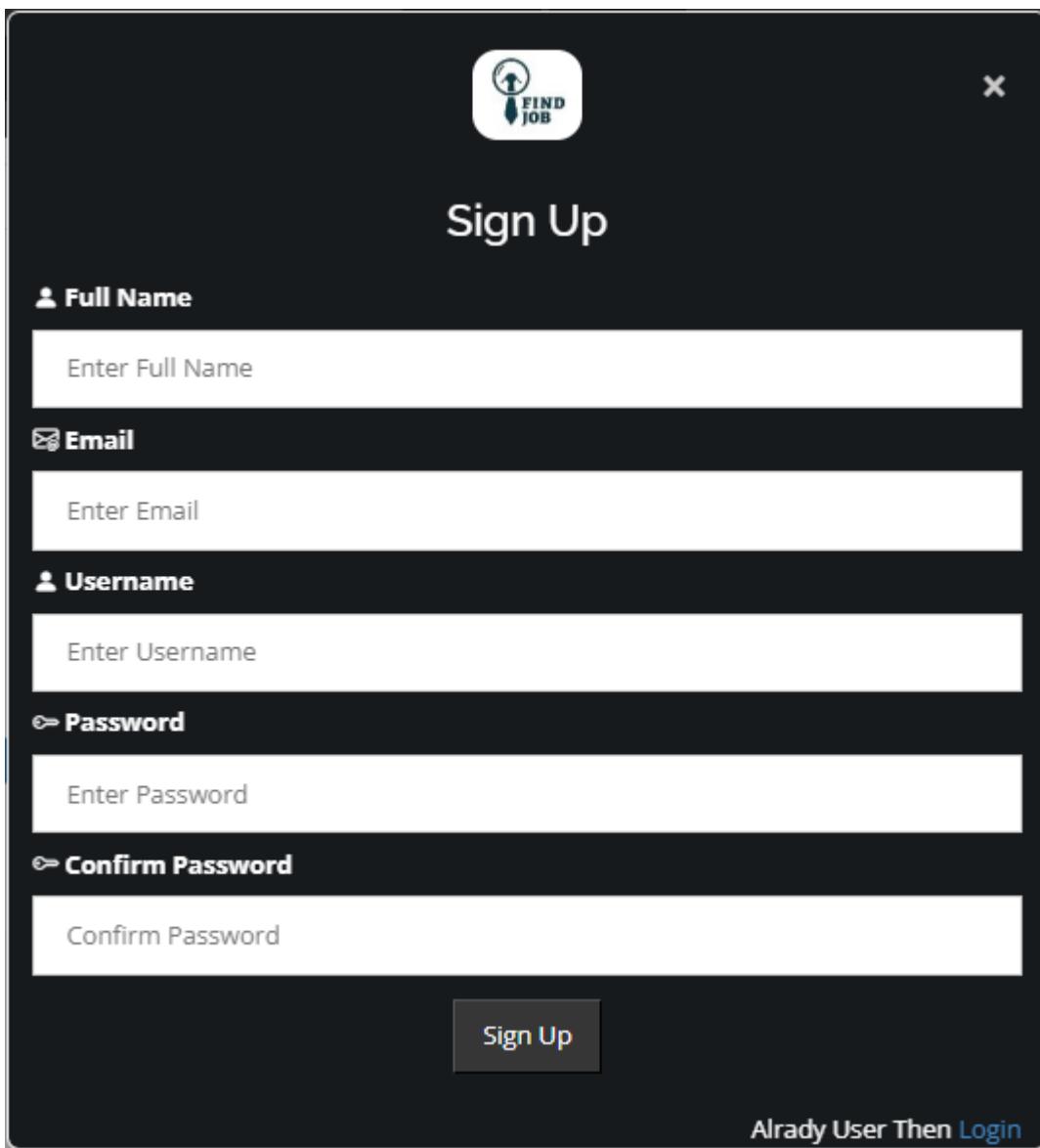
Remember me

Login Cancel

[Forgot password?](#)

# FIND JOB

## ➤ Sign Up Form:



The image shows a mobile application's sign-up screen. At the top center is a white circular logo containing a stylized 'F' and the words 'FIND JOB'. In the top right corner is a small 'X' icon. Below the logo, the word 'Sign Up' is centered in a large, bold, white font. The form consists of six input fields, each preceded by a small icon and a label: 'Full Name' (person icon), 'Email' (envelope icon), 'Username' (person icon), 'Password' (keyhole icon), 'Confirm Password' (keyhole icon), and 'Sign Up' (a large button). Below the 'Sign Up' button is a link 'Alrady User Then Login'.

Full Name

Email

Username

Password

Confirm Password

Sign Up

Alrady User Then Login

# FIND JOB

## ➤ Contact:

**CONTACT**

**Get In Touch With Us**

 <b>ADDRESS</b> Address Line 1, Address Line 2	 <b>CALL US</b> +91 9999999999	<b>Your Name</b> <input type="text"/> <b>Subject</b> <input type="text"/> <b>Message</b> <input type="text"/>
 <b>EMAIL US</b> contact@example.com test@example.com	 <b>WORKING HOURS</b> Mon to Sat 9:30 AM to 6:00 PM	<b>Send Message</b>

**Subscribe For New Job Notification**

Enter Email And Subscribe Our New Job Related Information

Enter Email  **Subscribe**

**USEFUL LINKS**

- Home
- Job Search
- Company
- Contact

**CONTACT US**

Address  
Address Line 1  
Address Line 2

**Phone:** +91 9999999999  
**Email:** test@example.com

**Follow US**

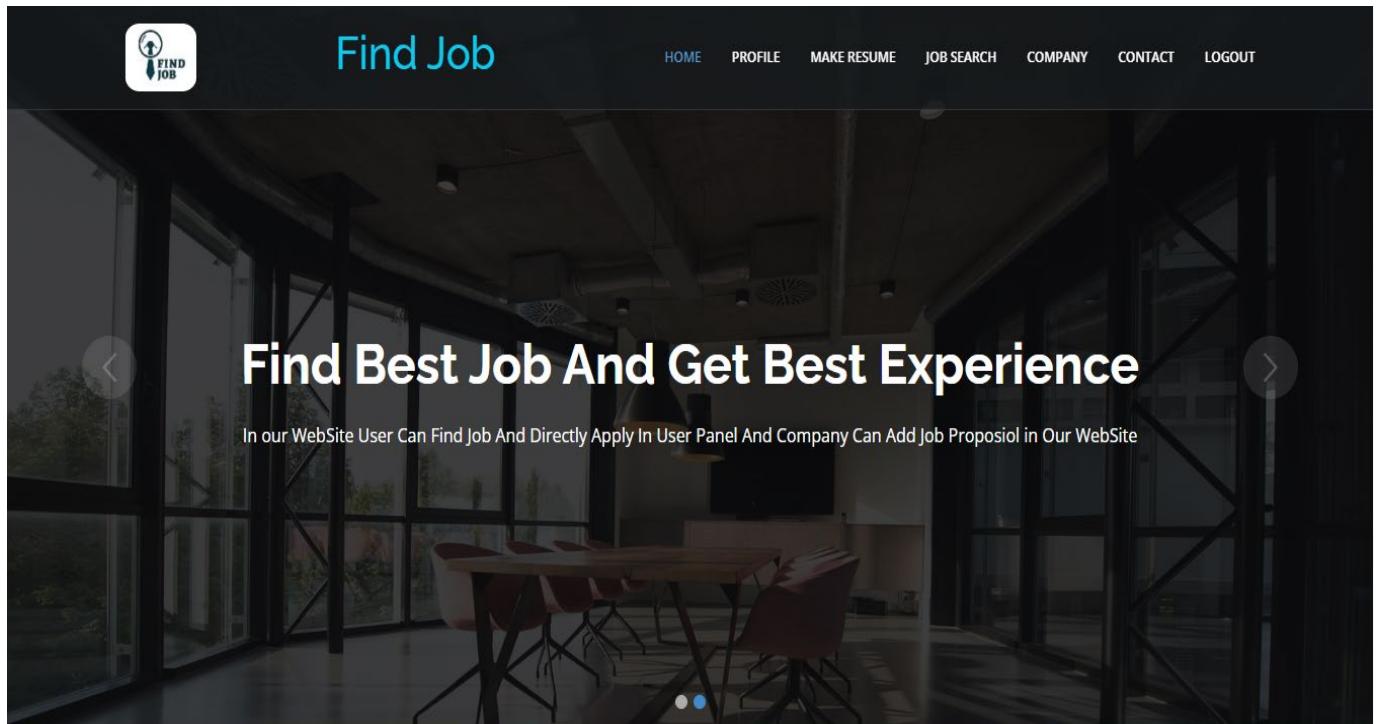
    

© Copyright Find Job. 2023 All Rights Reserved  
Designed by Abhi Sheth

# FIND JOB

## 2) User Side Screenshots: -

### ➤ User Home Page:



# FIND JOB

## ➤ Profile Page:

×

  
**Profile**

<b>Full Name</b>	USER
<b>Email</b>	user@gmail.com
<b>Mobile</b>	9999999999
<b>Address</b>	asdfsdfdfdsfdsfsdf
<b>Graduation</b>	M.C.A
<b>H.S.C</b>	B.O
<b>Degree</b>	B
<b>Master Degree</b>	B.I
<b>Date Of Birth</b>	10-05-2000
<input type="button" value="Update"/> <input type="button" value="Cancel"/>	

# FIND JOB

## ➤ Job Search Page:

JOB SEARCH

COMPANY  JOB TYPE  CITY  SALARY

Company	JOB TYPE	CITY	JOB REQUIREMENT	SALARY	APPLY
TCS	PL/SQL DEVELOPER	BENGALURU	Regular Employee Years of Experience 6 to 10+ years	15000	<input type="checkbox"/>
ORACLE	FULL STACK DEVELOPER	JUNAGADH	Regular Employee Years of Experience 6 to 10+ year	35000	<input type="checkbox"/>

JOB

ID	USER EMAIL	NAME	COMPANY NAME	JOB ID	JOB TYPE	APPROVE OR REJECT	DELETE
1	user@gmail.com	USER	TCS	1	PL/SQL DEVELOPER	APPROVE	<input type="checkbox"/>
2	user@gmail.com	USER	ORACLE	2	FULL STACK DEVELOPER	APPROVE	<input type="checkbox"/>

## ➤ Company Page:



# FIND JOB

## ➤ Contact Page:

**CONTACT**

**Get In Touch With Us**

 **ADDRESS**  
Address Line 1,  
Address Line 2

 **EMAIL US**  
contact@example.com  
test@example.com

 **CALL US**  
+91 9999999999

 **WORKING HOURS**  
Mon to Sat  
9:30 AM to 6:00 PM

Your Name

Your Email

Subject

Message

**Send Message**

**Subscribe For New Job Notification**  
Enter Email And Subscribe Our New Job Related Information

Enter Email **Subscribe**

**USEFUL LINKS**

- Home
- Job Search
- Company
- Contact

**CONTACT US**

Address  
Address Line 1  
Address Line 2

Phone: +91 9999999999  
Email: test@example.com

**Follow US**

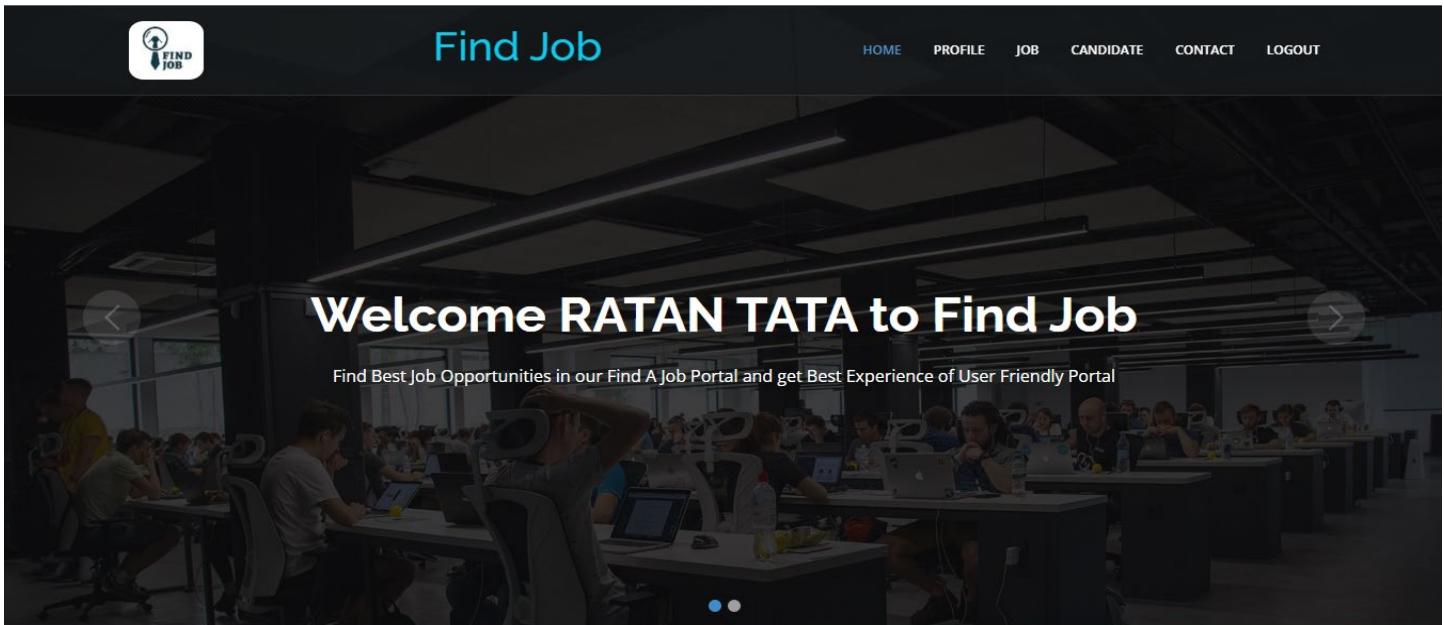
    

© Copyright Find Job. 2023 All Rights Reserved  
Designed by Abhi Sheth

# FIND JOB

### 3) Company Side Screenshots :-

- Home Page:

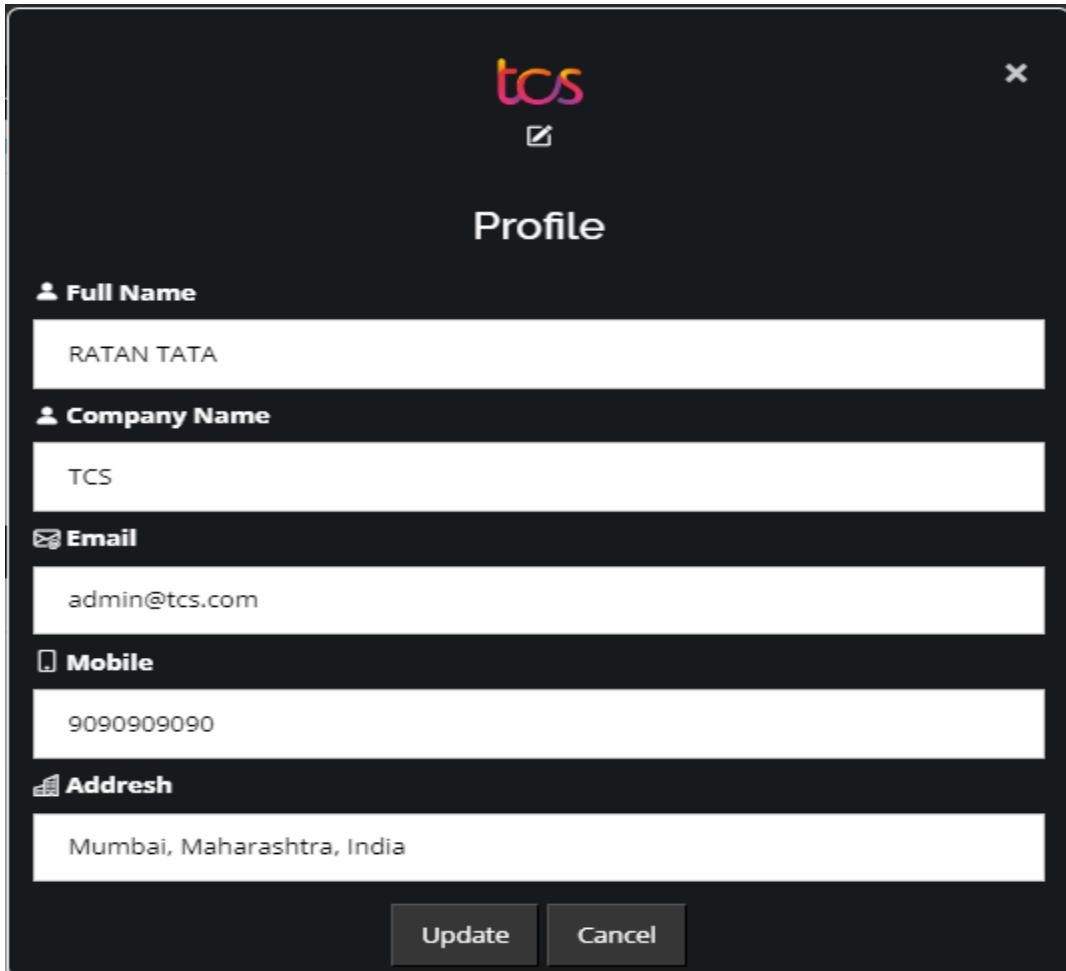


- Job Page:

ID	Company	JOB TYPE	CITY	JOB REQUIREMENT	salary	DELETE	UPDATE
1	TCS	PL/SQL DEVELOPER	BENGALURU	Regular Employee Years of Experience 6 to 10+ years	15000		

# FIND JOB

➤ Profile Page:



The image shows a profile page for a user named Ratan Tata, associated with TCS. The page has a dark background with white text fields. It includes fields for Full Name, Company Name, Email, Mobile, and Address, each with a corresponding icon. There are 'Update' and 'Cancel' buttons at the bottom.

ID	USER EMAIL	NAME	JOB ID	JOB TYPE	APPROVE OR REJECT	DELETE	UPDATE
1	user@gmail.com	USER	1	PL/SQL DEVELOPER	APPROVE	<input type="button" value="Delete"/>	<input checked="" type="checkbox"/>

➤ Candidate Page:

# FIND JOB

## ➤ Job Update Page:

X

TCS

**JOB TYPE**

PL/SQL DEVELOPER

**CITY**

BENGALURU

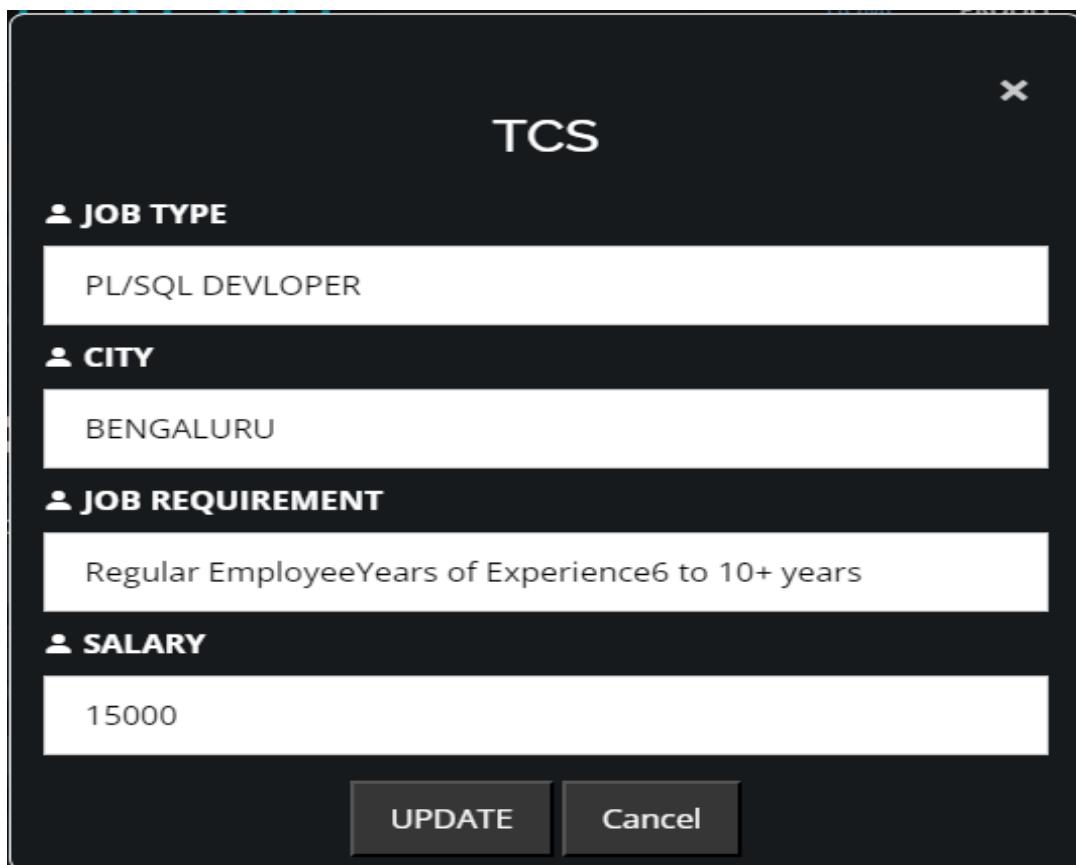
**JOB REQUIREMENT**

Regular Employee Years of Experience 6 to 10+ years

**SALARY**

15000

UPDATE Cancel

A screenshot of a mobile application showing a job update form. The title bar says "FIND JOB". Below it, a section header "➤ Job Update Page:" is followed by a large black rectangular dialog box. The dialog has a close button "X" at the top right. Inside, the word "TCS" is centered above the form fields. The form consists of several sections: "JOB TYPE" with a dropdown menu containing "PL/SQL DEVELOPER"; "CITY" with a dropdown menu containing "BENGALURU"; "JOB REQUIREMENT" with a dropdown menu containing "Regular Employee Years of Experience 6 to 10+ years"; and "SALARY" with a dropdown menu containing "15000". At the bottom of the dialog are two buttons: "UPDATE" and "Cancel".

# FIND JOB

## ➤ Candidate Update Page:

 X

### DETAILS

**Full Name**  
USER

**Email**  
user@gmail.com

**Mobile**  
9999999999

**Address**  
asdfsdfdfdsfdsfsdf

**Greduation**  
M.C.A

**H.S.C**  
80

**Degree**  
8

**Master Degree**  
8.1

**Date Of Birth**  
2000-05-10

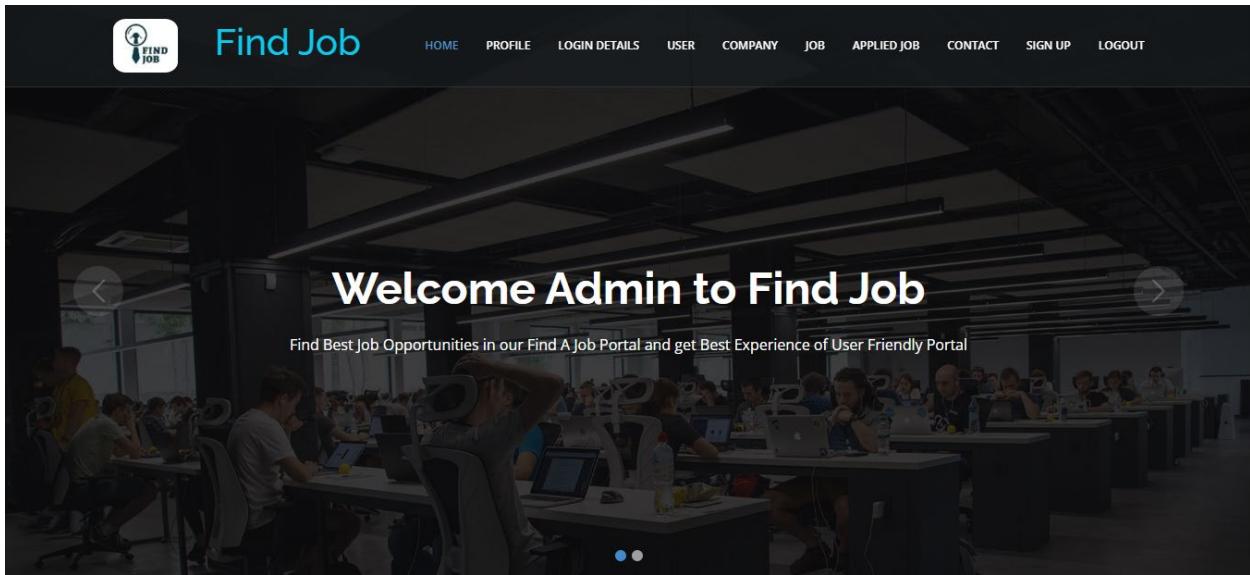
**APPROVE OR REJECT**  
 APPROVE  REJECT

UPDATE Cancel

# FIND JOB

## 4) Admin Side Screenshots :-

➤ Home page:



➤ Login Details page:

ID	FULL NAME	EMAIL ID	USER NAME	USER TYPE	DELETE	UPDATE
1	Admin	admin@findjob.com	admin	A	<a href="#"></a>	<a href="#"></a>
2	USER	user@gmail.com	user	U	<a href="#"></a>	<a href="#"></a>
3	RATAN TATA	admin@tcs.com	tcs	C	<a href="#"></a>	<a href="#"></a>
4	LARRY ELLISON	admin@oracle.com	oracle	C	<a href="#"></a>	<a href="#"></a>

# FIND JOB

## ➤ User page:

USER DETAILS						
ID	FULL NAME	EMAIL ID	MOBILE	DATE OF BIRTH	DELETE	UPDATE
1	USER	user@gmail.com	9999999999	2000-05-10	<a href="#">Delete</a>	<a href="#">Edit</a>

## ➤ Company page:

COMPANY DETAILS							
ID	FULL NAME	COMPANY	EMAIL ID	MOBILE	ADDRESS	DELETE	UPDATE
1	RATAN TATA	TCS	admin@tcs.com	9090909090	Mumbai, Maharashtra, India	<a href="#">Delete</a>	<a href="#">Edit</a>
2	LARRY ELLISON	ORACLE	admin@oracle.com	9999999999	2300 Oracle Way, Austin, Texas	<a href="#">Delete</a>	<a href="#">Edit</a>

## ➤ Job page:

JOB							
ID	Company	JOB TYPE	CITY	JOB REQUIREMENT	SALARY	DELETE	UPDATE
1	TCS	PL/SQL DEVELOPER	BENGALURU	Regular Employee Years of Experience 6 to 10+ years	15000	<a href="#">Delete</a>	<a href="#">Edit</a>
2	ORACLE	FULL STACK DEVELOPER	JUNAGADH	Regular Employee Years of Experience 6 to 10+ year	35000	<a href="#">Delete</a>	<a href="#">Edit</a>

## ➤ Apply Job page:

JOB								
ID	USER EMAIL	NAME	COMPANY NAME	JOB ID	JOB TYPE	APPROVE OR REJECT	DELETE	UPDATE
1	user@gmail.com	USER	TCS	1	PL/SQL DEVELOPER	APPROVE	<a href="#">Delete</a>	<a href="#">Edit</a>
2	user@gmail.com	USER	ORACLE	2	FULL STACK DEVELOPER	APPROVE	<a href="#">Delete</a>	<a href="#">Edit</a>

# FIND JOB

➤ Contact page:

**CONTACT**

**Get In Touch With Us**

 **ADDRESS**  
Address Line 1,  
Address Line 2

 **EMAIL US**  
contact@example.com  
test@example.com

 **CALL US**  
+91 9999999999

 **WORKING HOURS**  
Mon to Sat  
9:30 AM to 6:00 PM

Your Name

Your Email

Subject

Message

**Send Message**

**Subscribe For New Job Notification**

Enter Email And Subscribe Our New Job Related Information

**USEFUL LINKS**

- > [Home](#)
- > [User](#)
- > [Company](#)
- > [Job](#)
- > [Contact](#)

**CONTACT US**

Address  
Address Line 1  
Address Line 2

Phone: +91 9999999999  
Email: test@example.com

**Follow US**

© Copyright Find Job. 2023 All Rights Reserved  
Designed by [Abhi Sheth](#)

[↑](#)

# FIND JOB

➤ Login Update page:



**Login**

**Full Name**  
RATAN TATA

**Email**  
admin@tcs.com

**Username**  
tcs

**Password**  
...

**Confirm Password**  
...

Choose Role:  User  Company  Admin

**UPDATE** **Cancel**

# FIND JOB

➤ User Update page:



Full Name  
USER

Email  
user@gmail.com

Mobile  
9999999999

Address  
asdfsdfdsfdsfsdf

Graduation  
M.C.A

H.S.C  
B0

Degree  
B

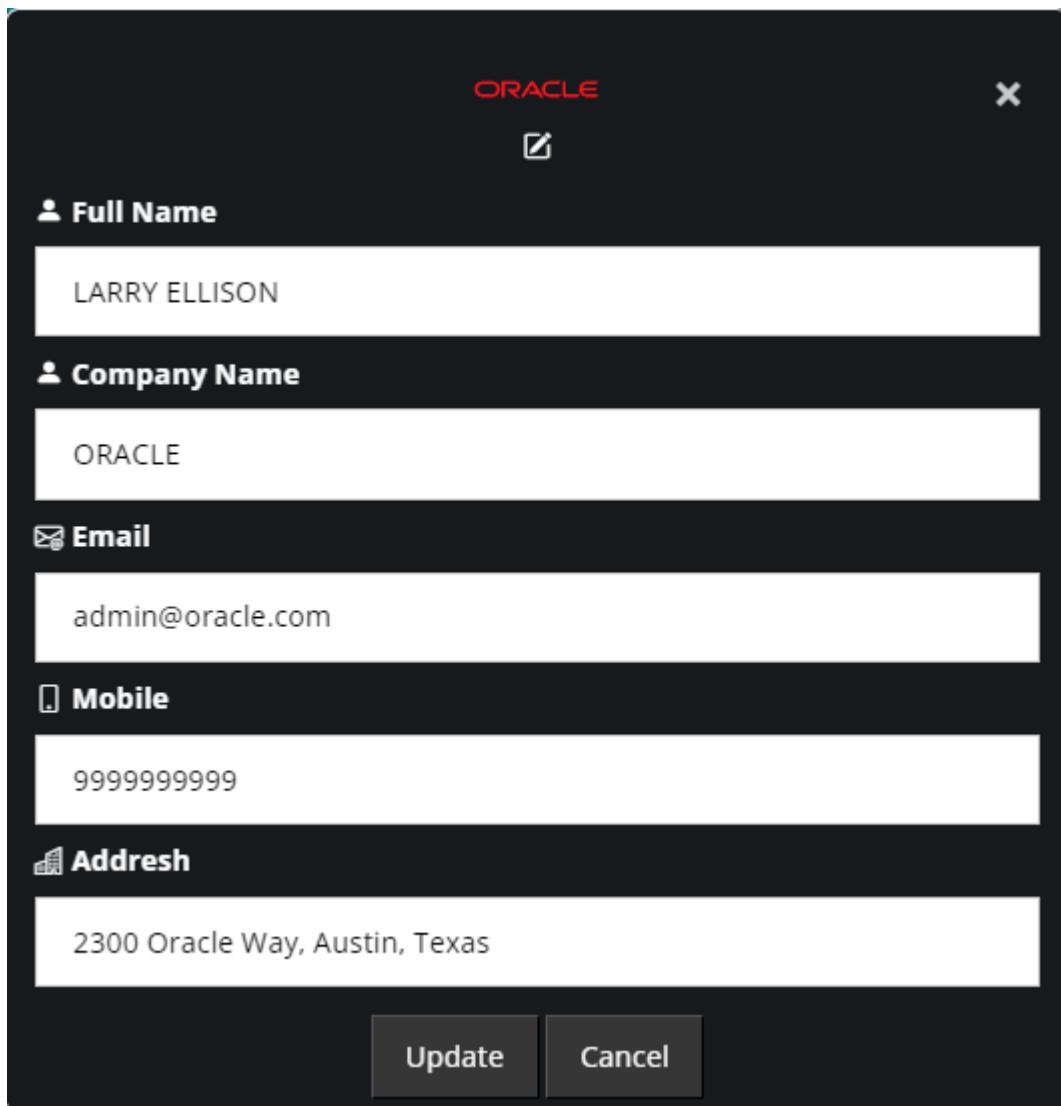
Master Degree  
B.1

Date Of Birth  
10-05-2000

Update Cancel

# FIND JOB

➤ Company Update page:



A screenshot of a mobile application's "Company Update" screen. The background is black. At the top center is the word "ORACLE" in red. To its right is a white "X" icon. Below this is a white edit icon. The screen contains five input fields with labels and placeholder text:

- Full Name**: LARRY ELLISON
- Company Name**: ORACLE
- Email**: admin@oracle.com
- Mobile**: 9999999999
- Address**: 2300 Oracle Way, Austin, Texas

At the bottom are two buttons: "Update" on the left and "Cancel" on the right.

## Chapter – 5

### CODING SCREENSHOTS

- **Insert And Update Data :-**

```
<?php
$cn = mysqli_connect("localhost", "root", "", "fj");
if (isset($_POST['ok']) || isset($_POST['adok'])) {
    if ($_POST['ps'] == $_POST['rps']) {

        $fnm = strtoupper($_POST['fnm']);
        $em = strtolower($_POST['em']);
        $un = $_POST['un'];
        $ps = $_POST['ps'];
        $rps = $_POST['rps'];
        $uac = $_POST['uac'];

        $query = "INSERT INTO user (id, fnm, em, un, ps, uac) VALUES ('', '$fnm',
'$em', '$un', '$ps', '$uac')";
        $result = mysqli_query($cn, $query);
        $row = mysqli_affected_rows($cn);
        $fe = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM user WHERE
un='$un' AND ps='$ps'"));
        if ($row > 0) {
            session_start();
            if (isset($_POST['adok'])) {
                echo "<script>alert('Record Update Sucessfully...')";
                window.location.assign('admin.php');</script>";
            }else{
                $_SESSION['lo'] = 'lo';
                $_SESSION['usr'] = $fe['un'];
                echo "<script>alert('Record Update
Sucessfully...');window.location.assign('user.php');</script>";
            }
        } else {
            $msg = "Something Wrong.";
        }
    } else {
        if (isset($_POST['adok'])) {
```

# FIND JOB

```
echo "<script>alert('Password Must Be Same.');
```

```
window.location.assign('admin.php');
```

```
</script>";
```

```
    } else {
```

```
        echo "<script>alert('Password Must Be Same.');
```

```
window.location.assign('index.php');
```

```
</script>";
```

```
    }
```

```
}
```

```
}
```

```
if (isset($_POST['adup'])) {
```

```
    if ($_POST['ps'] == $_POST['rps']) {
```

```
        $id = $_POST['id'];
```

```
        $fnm = strtoupper($_POST['fnm']);
```

```
        $em = strtolower($_POST['em']);
```

```
        $un = $_POST['un'];
```

```
        $ps = $_POST['ps'];
```

```
        $rps = $_POST['rps'];
```

```
        $uac = $_POST['uac'];
```

```
        $emq = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `user` WHERE
```

```
`id` ='$id'"));
```

```
        $uem = mysqli_query($cn, "SELECT * FROM `us`");
```

```
        while ($uem1 = mysqli_fetch_array($uem)) {
```

```
            if ($uem1['em'] === $emq['em']) {
```

```
                $uquery = "UPDATE us SET em ='$em' WHERE id = '$uem1[id]'" ;
```

```
                $result = mysqli_query($cn, $uquery);
```

```
            }}
```

```
            $cem = mysqli_query($cn, "SELECT * FROM `co`");
```

```
            while ($cem1 = mysqli_fetch_array($cem)) {
```

```
                if ($cem1['em'] === $emq['em']) {
```

```
                    $cquery = "UPDATE co SET em ='$em' WHERE id = '$cem1[id]'" ;
```

```
                    $result = mysqli_query($cn, $cquery);
```

```
                }}
```

```
                $update = "UPDATE `user` SET
```

```
                `fnm`='$fnm',`em`='$em',`un`='$un',`ps`='$ps',`uac`='$uac' WHERE id ='$id'" ;
```

```
                $result = mysqli_query($cn, $update);
```

```
                $row = mysqli_affected_rows($cn);
```

```
                if ($row > 0) {
```

```
                    session_start();
```

```
                    echo "<script>alert('Record Update Sucessfully...');
```

```
window.location.assign('admin.php');
```

```
</script>";
```

```
                } else {
```

```
                    $msg = "Something Wrong.";
```

```
                }
```

```
            } else {
```

# FIND JOB

```
echo "<script>alert('Password Must Be Same.');
```

```
window.location.assign('admin.php');</script>";
```

```
}
```

```
}
```

```
if (isset($_POST['upc'])) {
```

```
$folder = "logo/" . $_FILES["logo"]["name"];
```

```
$id = $_POST['id'];
```

```
$fnm = strtoupper($_POST['fnm']);
```

```
$conm = strtoupper($_POST['conm']);
```

```
$em = strtolower($_POST['em']);
```

```
$mob = $_POST['mob'];
```

```
$ad = $_POST['ad'];
```

```
$logo = $_FILES['logo']['name'];
```

```
move_uploaded_file($_FILES['logo']['tmp_name'], $folder);
```

```
$emq = mysqli_fetch_array(mysqli_query($cn, "SELECT `em` FROM `user` WHERE id = '$id'"));
```

```
$uquery = "UPDATE user SET fnm ='$fnm',em ='$em' WHERE id = '$id'";
```

```
$result = mysqli_query($cn, $uquery);
```

```
$emqc = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `co` WHERE em = '$emq[em]'"));
```

```
$job = mysqli_query($cn, "SELECT * FROM `job`");
```

```
while ($job1 = mysqli_fetch_array($job)) {
```

```
if ($job1['conm'] === $emqc['conm']) {
```

```
$uquery = "UPDATE job SET conm ='$conm' WHERE id = '$job1[id]'" ;
```

```
$result = mysqli_query($cn, $uquery);
```

```
}
```

```
if ($emq['em'] == $emqc['em']) {
```

```
$i = "UPDATE co SET conm='$conm',em ='$em',mob='$mob',ad='$ad',logo='$logo' WHERE id='$emqc[id]'";
```

```
$ir = mysqli_query($cn, $i);
```

```
} else {
```

```
$q = "INSERT INTO `co`(`id`, `conm`, `em`, `mob`, `ad`, `logo`) VALUES ('','$conm','$em','$mob','$ad','$log')";
```

```
$r = mysqli_query($cn, $q);
```

```
}
```

```
$row = mysqli_affected_rows($cn);
```

```
if ($row > 0) {
```

```
mysqli_close($cn);
```

```
echo "<script>alert('Record Added
```

```
Successfully.');?>window.location.assign('admin.php');</script>";
```

```
} else {
```

```
mysqli_close($cn);
```

# FIND JOB

```
echo "<script>alert('Something  
Wrong.');?>window.location.assign('admin.php');</script>";  
}  
}  
if (isset($_POST['upu'])) {  
  
$folder = "photo/" . $_FILES["photo"]["name"];  
$id = $_POST['id'];  
$fnm = strtoupper($_POST['fnm']);  
$em = strtolower($_POST['em']);  
$mob = $_POST['mob'];  
$ad = $_POST['ad'];  
$greduation = strtoupper($_POST['greduation']);  
$hsc = $_POST['hsc'];  
$deg = $_POST['deg'];  
$mas = $_POST['mas'];  
$photo = $_FILES['photo']['name'];  
$dob = $_POST['dob'];  
move_uploaded_file($_FILES['photo']['tmp_name'], $folder);  
$emq = mysqli_fetch_array(mysqli_query($cn, "SELECT `em` FROM `user` WHERE id  
='$id'"));  
$uquery = "UPDATE user SET fnm ='$fnm',em ='$em' WHERE id = '$id'";  
$result = mysqli_query($cn, $uquery);  
$emqc = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `us` WHERE em  
='$emq[em]'"));  
if ($emq['em'] == $emqc['em']) {  
    $i = "UPDATE us SET em  
='$em',mob='$mob',ad='$ad',greduation='$greduation',hsc='$hsc',deg='$deg',mas='$m  
as',photo='$photo',dob='$dob' WHERE id='$emqc[id]'";  
    $ir = mysqli_query($cn, $i);  
} else {  
    $q = "INSERT INTO `us`(`em`, `mob`, `ad`, `greduation`, `hsc`, `deg`,  
`mas`, `photo`, `dob`) VALUES  
('$em','$mob','$ad','$greduation','$hsc','$deg','$mas','$photo','$dob')";  
    $r = mysqli_query($cn, $q);  
}  
$row = mysqli_affected_rows($cn);  
if ($row > 0) {  
    mysqli_close($cn);  
    echo "<script>alert('Record Added  
Successfully.');?>window.location.assign('admin.php');</script>";  
} else {  
    mysqli_close($cn);
```

# FIND JOB

```
echo "<script>alert('Something  
Wrong.');?>window.location.assign('admin.php');</script>";  
}  
}  
}  
if (isset($_POST['aju'])) {  
    $id = $_POST['id'];  
    $ans = $_POST['ans'];  
    $i = "UPDATE appjob SET ans='$ans' WHERE id='$id'";  
    $ir = mysqli_query($cn, $i);  
    mysqli_close($cn);  
    echo "<script>alert('Record Added  
Successfully.');?>window.location.assign('admin.php');</script>";  
}
```

- **Login :-**

```
<?php  
  
if (isset($_POST['ok'])) {  
    $cn = mysqli_connect("localhost", "root", "", "fj");  
    if ($cn) {  
        echo "";  
    } else {  
        die("Connaction Faild Because" . mysqli_connect_error());  
    }  
  
    $un = $_POST['un'];  
    $ps = $_POST['ps'];  
  
    $rec = mysqli_query($cn, "SELECT * FROM user WHERE un='$un' AND  
    ps='$ps'");  
    $res = mysqli_num_rows($rec);  
    $fe = mysqli_fetch_array($rec);  
    if ($res == 1) {  
        session_start();  
        $_SESSION['lo'] = 'lo';  
        if ($fe['uac'] === 'A') {  
            $_SESSION['adm'] = $fe['un'];  
            echo "<script>window.location.assign('admin.php');</script>";  
        }  
        if ($fe['uac'] === 'C') {  
            $_SESSION['com'] = $fe['un'];  
            echo "<script>window.location.assign('company.php');</script>";  
        }  
    }  
}
```

# FIND JOB

```
if ($fe['uac'] === 'U') {
    $_SESSION['usr'] = $fe['un'];
    echo "<script>window.location.assign('user.php');</script>";
}
} else {
    echo "<script> alert('Username or Password Not Valid');
        window.location.assign('index.php');</script>";
}
?>
```

- **Logout :-**

```
<?php
session_start();
unset($_SESSION['adm']);
unset($_SESSION['usr']);
unset($_SESSION['com']);
session_destroy();
session_start();
$_SESSION['lo'] = 'lo';
header("location:index.php");
?>
```

## Admin Side Screenshots: -

- **Insert , Delete And Update:**

```
<?php
session_start();
if (isset($_SESSION['lo'])) {
    echo "<audio id='bgAudio' src='assets/admin.mp3' autoplay ></audio>
<script>
    var audio = document.getElementById('bgAudio');
    audio.volume = 0.3;
</script>";
    unset($_SESSION['lo']);
}
if (!isset($_SESSION['adm'])) {
    header("location:index.php");
```

# FIND JOB

```
}

$cn = mysqli_connect("localhost", "root", "", "fj");
if (isset($_GET['key']) && $_GET['key'] == "d") {
    if ($_GET['id'] == 1) {
        echo "<script>alert('Admin Can Not
Deleted.');?>window.location.assign('admin.php');</script>";
    } else {
        $d_query = "DELETE FROM `user` WHERE id=" . $_GET['id'];
        $d_result = mysqli_query($cn, $d_query);
        $d_row = mysqli_affected_rows($cn);
        if ($d_row > 0) {
            echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
        }
    }
}
if (isset($_GET['key']) && $_GET['key'] == "du") {
    $d_query = "DELETE FROM `us` WHERE id=" . $_GET['id'];
    $d_result = mysqli_query($cn, $d_query);
    $d_row = mysqli_affected_rows($cn);
    if ($d_row > 0) {
        echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
    }
}
if (isset($_GET['key']) && $_GET['key'] == "dc") {
    $d_query = "DELETE FROM `co` WHERE id=" . $_GET['id'];
    $d_result = mysqli_query($cn, $d_query);
    $d_row = mysqli_affected_rows($cn);
    if ($d_row > 0) {
        echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
    }
}
if (isset($_GET['key']) && $_GET['key'] == "dj") {
    $d_query = "DELETE FROM `job` WHERE id=" . $_GET['id'];
    $d_result = mysqli_query($cn, $d_query);
    $d_row = mysqli_affected_rows($cn);
    if ($d_row > 0) {
        echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
    }
}
if (isset($_GET['key']) && $_GET['key'] == "daj") {
```

# FIND JOB

```
$d_query = "DELETE FROM `appjob` WHERE id=" . $_GET['id'];
$d_result = mysqli_query($cn, $d_query);
$d_row = mysqli_affected_rows($cn);
if ($d_row > 0) {
    echo "<script>alert('Record Deleted
Successfully.');?>
window.location.assign('admin.php');
window.location.assign('admin.php');
```

# FIND JOB

```
}

?>
<?php
if (isset($_GET['key'])) && $_GET['key'] == "ej") {
    $jid = $_GET['id'];
?>

<body onload="document.getElementById('job1').style.display = 'block'"></body>
<?php
}
?>
<?php
if (isset($_GET['key'])) && $_GET['key'] == "eaj") {
    $ajid = $_GET['id'];
?>

<body onload="document.getElementById('ajob1').style.display = 'block'"></body>
<?php
}
?>
```

## User Side Screenshots: -

- Insert , Delete And Update:

```
<?php
session_start();
if(!isset($_SESSION['usr'])){
    header("location:index.php");
}
if(isset($_SESSION['lo'])){
    echo "<audio id='bgAudio' src='assets/user.mp3' autoplay ></audio>
<script>
    var audio = document.getElementById('bgAudio');
    audio.volume = 0.3;
</script>";
unset($_SESSION['lo']);
}
$cn = mysqli_connect("localhost", "root", "", "fj");
$u11 = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `user` WHERE un =
'$_SESSION[usr]'"));
$u1 = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `us` WHERE em =
'$u11[em]'"));
if (isset($_POST['search'])) {
```

# FIND JOB

```
$company = strtoupper($_POST['company']);
$jobtype = strtoupper($_POST['jobtype']);
$city = strtoupper($_POST['city']);
$salary = $_POST['salary'];
$log = mysqli_query($cn, "SELECT * FROM `job` WHERE conm like '$company' OR job
like '$jobtype' OR city like '$city' OR sal like '$salary'");
}
else{
    $log = mysqli_query($cn, "SELECT * FROM `job`");
}
if (isset($_GET['key']) && $_GET['key'] == "daj") {
    $d_query = "DELETE FROM `appjob` WHERE id=" . $_GET['id'];
    $d_result = mysqli_query($cn, $d_query);
    $d_row = mysqli_affected_rows($cn);
    if ($d_row > 0) {
        echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
    }
}
if (isset($_GET['key']) && $_GET['key'] == "aaj") {
    $id = $_GET['id'];
    $insert = mysqli_query($cn , "INSERT INTO `appjob`(`em`, `job`) VALUES
('$_11[em]','$id')");
    echo "<script>alert('Applye
Successfully.');?>window.location.assign('admin.php');</script>";
}
?>
```

## Company Side Screenshots: -

- Insert , Delete And Update:

```
<?php
session_start();
if (isset($_SESSION['lo'])) {
    echo "<audio id='bgAudio' src='assets/company.mp3' autoplay ></audio>
<script>
    var audio = document.getElementById('bgAudio');
    audio.volume = 0.3;
</script>";
    unset($_SESSION['lo']);
}
if (!isset($_SESSION['com'])) {
```

# FIND JOB

```
header("location:index.php");
}
$cn = mysqli_connect("localhost", "root", "", "fj");
$c11 = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `user` WHERE un =
'$_SESSION[com]'"));
$c1 = mysqli_fetch_array(mysqli_query($cn, "SELECT * FROM `co` WHERE em =
'$c11[em]'"));
if (isset($_GET['key']) && $_GET['key'] == "daj") {
    $d_query = "DELETE FROM `appjob` WHERE id=" . $_GET['id'];
    $d_result = mysqli_query($cn, $d_query);
    $d_row = mysqli_affected_rows($cn);
    if ($d_row > 0) {
        echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
    }
}
if (isset($_GET['key']) && $_GET['key'] == "dj") {
    $d_query = "DELETE FROM `job` WHERE id=" . $_GET['id'];
    $d_result = mysqli_query($cn, $d_query);
    $d_row = mysqli_affected_rows($cn);
    if ($d_row > 0) {
        echo "<script>alert('Record Deleted
Successfully.');?>window.location.assign('admin.php');</script>";
    }
}
if (isset($_GET['key']) && $_GET['key'] == "ej") {
    $jid = $_GET['id'];
?>

<body onload="document.getElementById('job1').style.display = 'block'"></body>
<?php
}
?>
<?php
if (isset($_GET['key']) && $_GET['key'] == "eaj") {
    $ajid = $_GET['id'];
?>

<body onload="document.getElementById('ajob1').style.display = 'block'"></body>
<?php
}
if (isset($_POST['ju'])) {
    $conm = $c1['conm'];
    $job = $_POST['job'];
}
```

# FIND JOB

```
$city = $_POST['city'];
$jobtyp = $_POST['jobtyp'];
$sal = $_POST['sal'];
$insert = mysqli_query($cn , "INSERT INTO `job` ( `conm` , `job` , `city` ,
`jobtyp` , `sal` ) VALUES ('$c1[conm]','$job','$city','$jobtyp','$sal')");
echo "<script>alert('Applye
Successfully.');?>
window.location.assign('admin.php');
```

## Vesitor Side Screenshots: -

```
<?php
session_start();
if (isset($_SESSION['usr'])) {
    header("location:user.php");}
if (isset($_SESSION['com'])) {
    header("location:company.php");}
if (isset($_SESSION['adm'])) {
    header("location:admin.php");}
if (isset($_SESSION['lo'])) {
    echo "<audio id='bgAudio' src='assets/logout.mp3' autoplay ></audio>
<script>
    var audio = document.getElementById('bgAudio');
    audio.volume = 0.3;
</script>";
    unset($_SESSION['lo']);
    session_destroy();}
$cn = mysqli_connect("localhost", "root", "", "fj");
if (isset($_POST['search'])) {
    $company = strtoupper($_POST['company']);
    $jobtype = strtoupper($_POST['jobtype']);
    $city = strtoupper($_POST['city']);
    $salary = $_POST['salary'];
    $log = mysqli_query($cn, "SELECT * FROM `job` WHERE conm like '$company' OR job
like '$jobtype' OR city like '$city' OR sal like '$salary'");
}
else{
    $log = mysqli_query($cn, "SELECT * FROM `job`");
}
?>
```

## Chapter – 6

### Testing

- 6.1 Techniques and strategies
- 6.2 Cost estimation model
- 6.3 Future scope and further Enhancement of the project
- 6.4 Bibliography
- 6.5 Appendices
- 6.6 Glossary

#### 6.1 – TECHNIQUES AND STRATEGIES

Testing is a process to show the corrections of the program. Testing is needed to show completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary reduce the testing costs and operation time.

Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. Based on the way the software reacts to these testing.

We can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

## ❖ White Box testing: -

White Box (or glass box) testing is the process of giving input to the system and checking how the system processes input to generate output.

It refers to the testing a system with full knowledge and access to all source code and other architecture documents. This testing enables to reveal bugs and vulnerabilities quickly in comparison with trial and error method. More complete testing coverage is ensured by exactly knowing what to test.

White box testing involves thorough testing of the application. It requires knowledge of code and the test cases chosen verifies if the system is implemented as expected. It typically includes checking with the data flow, exceptions, and errors, how they are handled, comparing if the code produces the expected results.

## ❖ Black Box testing: -

Black Box testing is the process of giving input to the system and checking the output of the system without bothering how the output is generated.

It refers to testing a system without knowledge of specification to the internal workings of the system, access to the source code, and knowledge of the architecture.

Essentially this approach mimics in a close approach, how an attacker typically follows approach to the application. However, the uncovering of issues or vulnerabilities could be further longer, because of lacking internal application knowledge.

Black box testing is done at an outer level of the system. Test cases merely check if the output is correct for the given input. User is not expected to the internal flow or design of the system.

## Grey Box testing: -

Grey Box testing is a combination of White Box and Glass Box Testing. In this, the tester has little knowledge about the internal working of the software.

So, he tests the output as well as process carried out to generate the output.

It refers to a testing system by knowing limited information about the internals of the system. The knowledge is always limited for detailed design documents and architecture diagrams. In concise, it is a good blend of black and white box testing, which leverages the strengths of each of the testing.

Grey box testing is a combination of both black box and white box testing. This is because it involves access to the system; however, at an outer level. A little knowledge of the system is expected in Grey box testing.

## Non-functional testing: -

Non-functional testing is the testing of a software application or system for its non-functional Requirements: the way a system operates, rather than specific behaviours of that system. This is contrast to functional testing, which tests against functional requirements that describe the functions of a system and its components. The names of many non-functional tests are often used interchangeably because of the overlap in scope between various non-functional requirements. For example, software performance is a broad term that includes many specific requirements like reliability and scalability.

# FIND JOB

## ⊕ Software testing strategies: -

Testing Involves:

- A. Unit Testing
- B. Integration Testing
- C. Acceptance testing

### **A. Unit Testing: -**

The unit testing is purpose of unit testing is to ensure that each program is fully tested.

### **B. Integration Testing: -**

The integration testing is individual program units or programs are integrated and tested as a complete system to ensure that the software.

### **C. Acceptance Testing: -**

This testing involves planning and the execution of various types of test in order to demonstrate that the implemented software system satisfied the requirements. Finally, our project meets the requirements after going through all the levels of testing.

## 6.2 – COST ESTIMATION MODEL

### ⊕ Costing Objectives: -

- ✓ **To ensure viability:**
  - ♣ Feasibility study
  - ♣ Resource planning
  - ♣ Cost/benefit analysis
- ✓ **Provide input for pricing (including bidding negotiations etc.)**
- ✓ **To serve as a management tool:**
  - ♣ Cost Control and Management
  - ♣ Risk management
  - ♣ Budget planning
- ✓ **Criteria for good project costing:**
  - ♣ Accurate
  - ♣ Realistic (good procurement and engineering practice)
  - ♣ Consistent
  - ♣ Transparent
  - ♣ self-effective
  - ♣ Good Documentation.

# FIND JOB

Many looks upon project testing as a cost. While it is true that software testing does cost money, in many cases significant amounts of money, it is also an activity that an organization to avoid costly failures further on in the development process.

Most understand this relationship project testing is spending money to save money. What many do not also realize is that software testing also produces valuable assets for the organization. This article will discuss those assets of software testing.

## Cost Estimation:

Working time estimation is as given,

$$1 \text{ months} + 24 \text{ days} = 54 \text{ days}$$

$$5 \text{ hr / day} = 270 \text{ hours}$$

Now, the expenses & cost estimation are given below:

Computer rent	= 10 ,000 /-
---------------	--------------

+ Light Bill Rs. 7 / unit	
---------------------------	--

Worth 400 units	= 2,800 /-
-----------------	------------

+ Database design & creation	= 5,500 /-
------------------------------	------------

+ Coding& Validation	= 9000 /-
----------------------	-----------

Amount	= 27,300 /-
--------	-------------

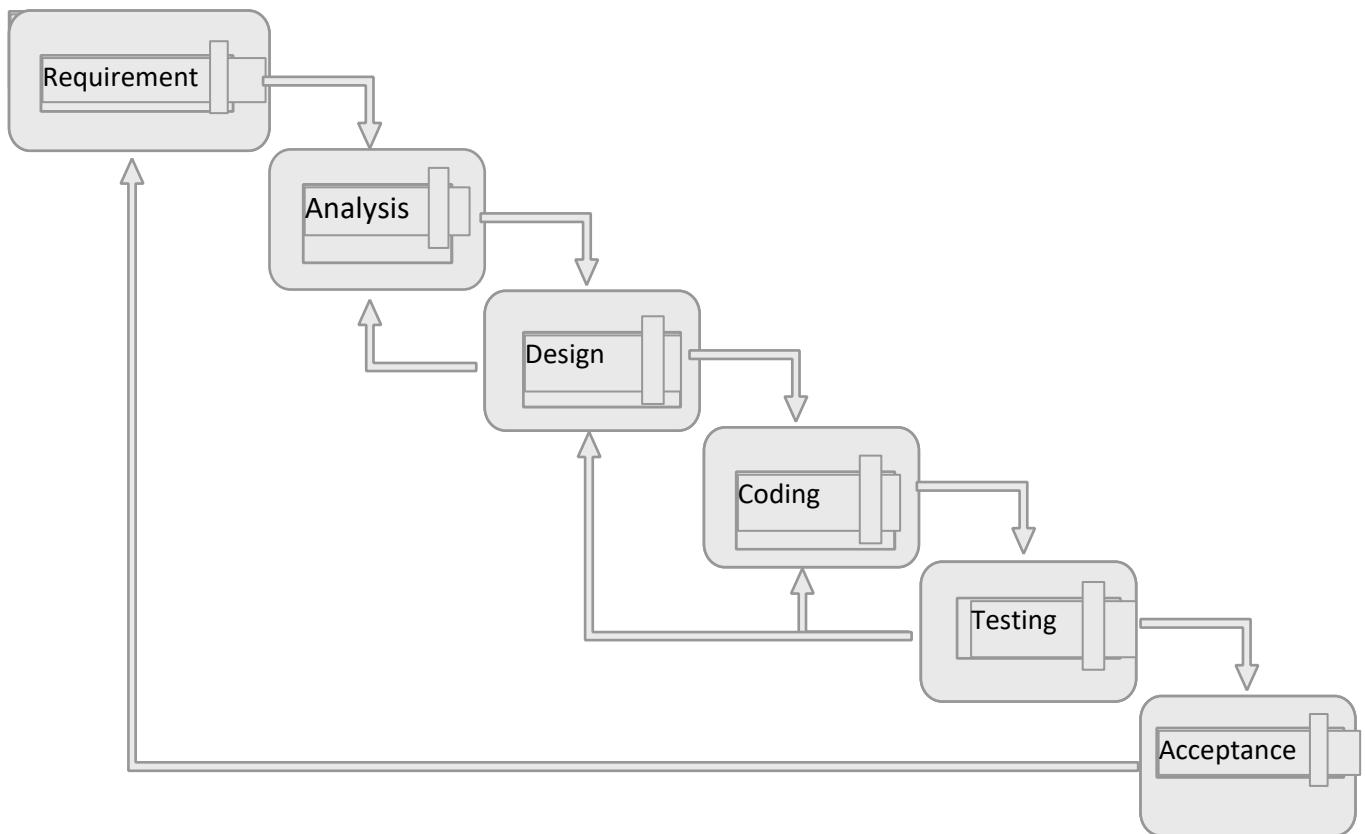
---

Total	= 27,300 /-
-------	-------------

## SDLC MODEL

### Waterfall Model: -

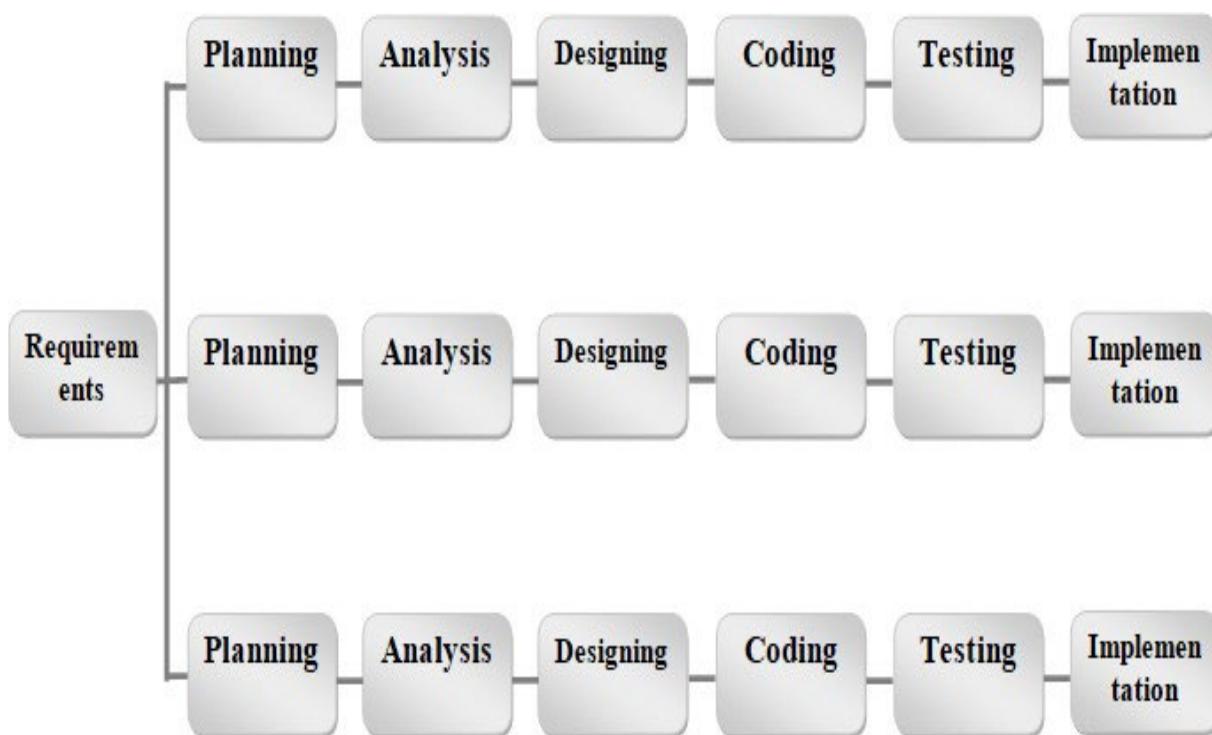
Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.



# FIND JOB

## ■ Iterative Model: -

In Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.



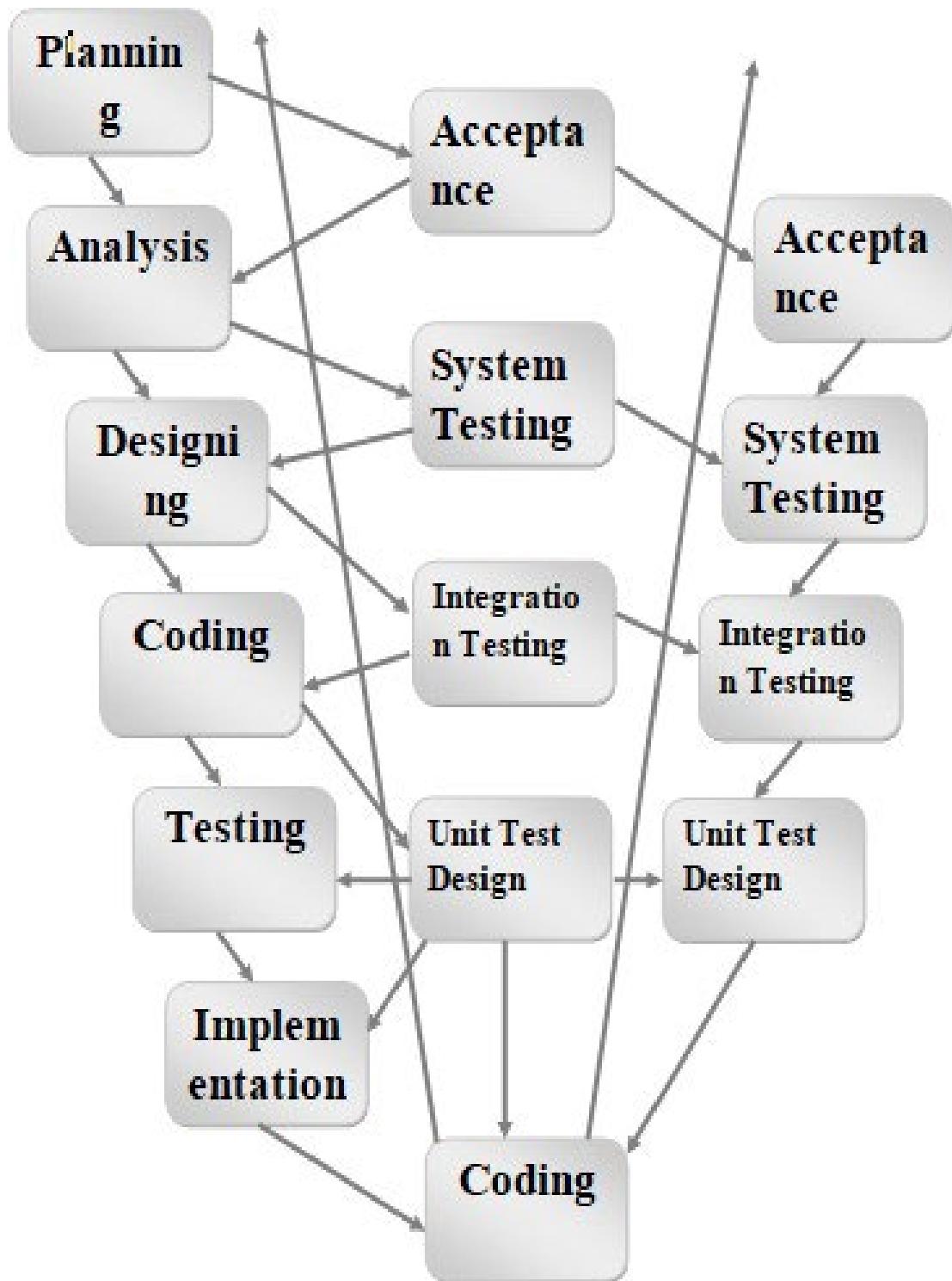
## V Model: -

The V - model is SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model. V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.

## V Model Design: -

Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the V. and Validation phases on the other side. Coding phase joins the two sides of the V-Model. The below figure illustrates the different phases in V-Model of SDLC.

# FIND JOB





## Spiral Model: -

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. Spiral model is a combination of iterative development process model and sequential linear development model i.e. waterfall model with very high emphasis on risk analysis. It allows for incremental releases of the product, or incremental refinement through each iteration around the spiral.



## Spiral Model Design: -

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

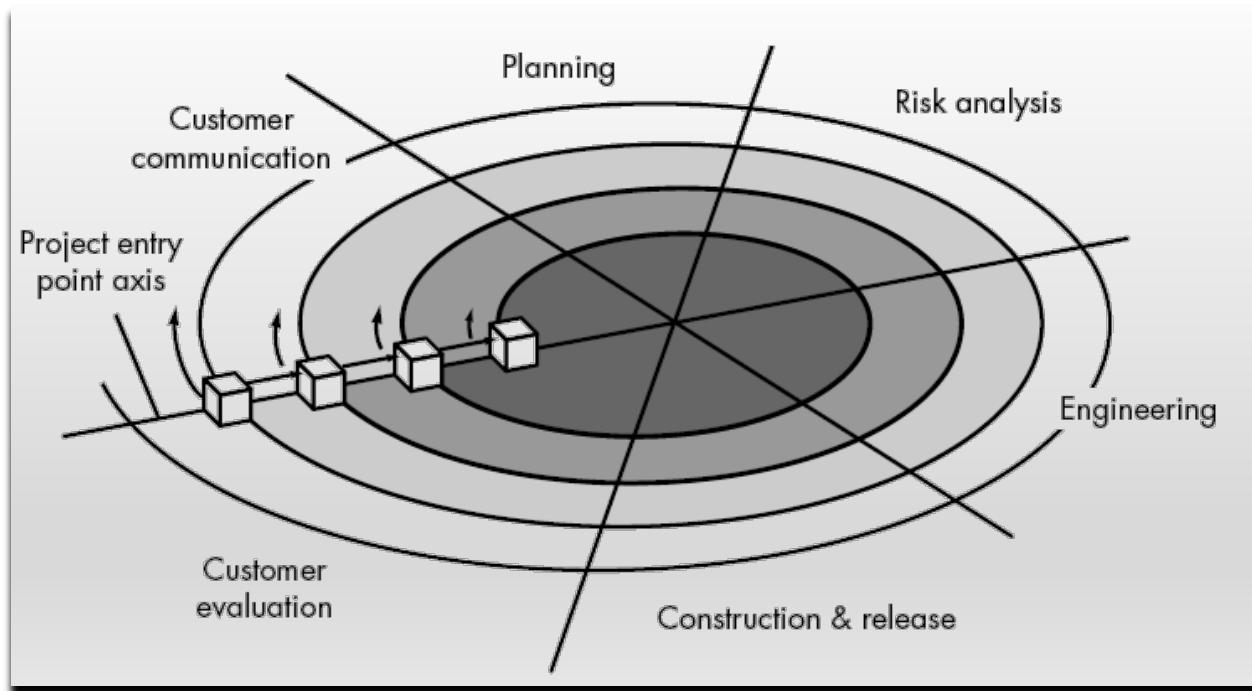
### **Identification:**

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in. This phase.

This also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral the product is deployed in the identified market.



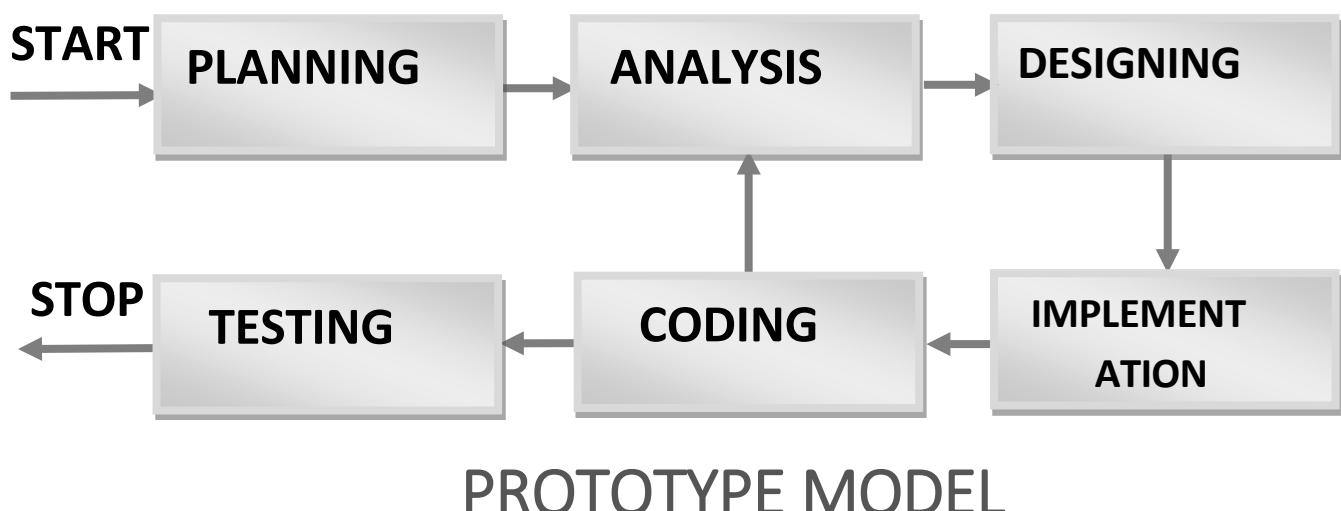
## Spiral Model Design: -



# FIND JOB

## Prototype Model: -

The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.



# FIND JOB

## 6.3 – FUTURE ENHANCEMENT

A good project is one which never stops developing according to the changing situations and technologies there is a lot of scope of future enhancements.

- Make it Fully Dynamic Website.
- Online Payment.
- User Search History.
- Update Their Profile.
- Notifications to users.
- Change password option for users.
- Alert Notification message to user on email or SMS.

## 6.4 – BIBLIOGRAPHY

For the successful working of my project I have referred many sources for the code snippets, logic and tips and tricks from the various books as well as web sites. Most I searched for the required possessions on the google.com search engine.

Web Links:

- W3school
- Bootstrap.com
- Geeksforgeeks

# FIND JOB

## 6.5 – APPENDICES

- The project Find Job is the customize working on Jobs Information Provider.
- The software takes care of all requirements and helps to the user for choosing good job.
- We have successfully designed, coded and implemented our project with a lot of Hard work.
- Finally, I would like to thank our project guide “Mr. Ezaz Shaikh” helpful guidelines for our project. Also given suggestion when difficulties raised in our project.

## 6.6 – GLOSSARY

### Full Form:-

<b>D.F.D.</b>	Data Flow Diagram
<b>S.R.S.</b>	Software Requirement Specification
<b>C.S.S</b>	Cascading Style Sheet
<b>P.E.R.T.</b>	Program Evaluation and Review Technique
<b>S.D.L.C</b>	Software Development Life Cycle
<b>P.H.P.</b>	Personal Home Page / Hypertext Pre-processor
<b>E.R</b>	Entity Relationship
<b>My S.Q.L.</b>	My Structure Query Language
<b>H.T.M.L.</b>	Hyper Text Mark-up Language
<b>XML</b>	Extensible Mark-up Language.

FIND JOB

**THANK  
YOU**