**SMT.TANUBEN AND DR. MANUBHAI TRIVEDI COLLEGE OF INFORMATION SCIENCE [BCA]**

AFFILIATED TO

VEER NARMAD SOUTH GUJARAT UNIVERSITY

SURAT (VNSGU)

**Project report**

**On**

**JOB MATCH**

**As partial requirement for the degree**

**Of**

**Bachelor of Computer Application [BCA]**

**[Academic Year: 2021-2022]**

Guided by:Submitted by:

Prof. BHAUTIKA PATEL PREET S. PATEL (exam no :2019047348)

**Organization**

SMT. TANUBEN & DR. MANUBHAI TRIVEDI COLLEGE OF INFORMATION SCIENCE (B.C.A.)

WADIA WOMEN’S COLLEGE CAMPUS, ATHWALINES, SURAT - 395001

**Personal profile**

|  |  |
| --- | --- |
| Field | Descriptions |
| Name: | PATEL PREETKUMARI SHRIKESHBHAI |
| Address: | 106, Sarvodaya Nagar, Bhimpore, Airport Road, Surat |
| Contact No: | 9558432205 |
| Email ID: | Patelpreet0812@gmail.com |
| Study Center: | **SMT.TANUBEN AND DR. MANUBHAI TRIVEDI COLLEGE OF INFORMATION SCIENCE [BCA]** |
| Exam No: | 2019047348 |
| Academic Year: | 2021-2022 |

**Preface**

With a great pleasure we undertake writing a document or a website to find a match for mutual job transfer. I would be proud, that I am student undertaking an education in the field of computer application with is being offered at graduation level at VNSGU (veer Narmad south Gujarat university) Surat.

As student of computer field, I must have encouraged by the growth and rapid development in software industry. Keeping in mind that ever-increasing demand for s/w and s/w engineer. The university has arranged for third year level.

Thus, it is out moral and compulsory duty to take this as part of my studies with great enthusiasm and seriousness and given it due importance for this i am gone through my development program of third month.

**Acknowledgement**

My self-Ms. PREET PATEL, I have opportunity to express my knowledge. I would like to express my gratitude to all those who gave me the possibility to complete my project. I express my sincere thanks and my heartfelt gratitude to Mr. RAHUL, for their able guidance, inestimable motivation and constant encouragement throughout my project.

I take rather special privilege of thanking my respected **DR. BHARAT PATEL** & all the faculties for making the resource available within the institute for the project work.

I am extremely grateful to **Prof. BHAUTIKA PATEL** project guide, for the tremendous help that she gave us in project.

last but not least genius thanks to all those who have directly or indirectly contributed to our success.

Thanking All,

Preet S. Patel

Index

|  |  |  |
| --- | --- | --- |
| Sr no | Description | Page no |
| 1. | Introduction |  |
|  | Objective |  |
|  | Project category |  |
|  | Scopes |  |
|  | Tools and Equipment’s used |  |
|  | Current System |  |
|  | Drawback of current system |  |
|  | Requirement Specification |  |
|  | Proposed system |  |
|  | Advantages of Proposed System |  |
|  | Use Case Diagram (UML) |  |
|  | Activity Diagram |  |
|  | Sequence Diagram |  |
|  | Table Structure |  |
|  | Security issues |  |
|  | Input screen Layout |  |
|  | Output screen Layout |  |
|  | Test case |  |
|  | Testing issues |  |
|  | Limitations of Proposed System |  |
|  | Future Enhancement of Project |  |
|  | Justification of MIS |  |
|  | References |  |
|  | Thank you |  |

**Introduction**

In this project, bank employee can make their profile for intention of job transfer from one place to another. They find the perfect match for them to transfer. Candidate can search a person who is want to transfer with him/her and make transfer too easy.

They also chat with each other they share their thoughts and opinions making a post. They get a follow request and also get transfer request.

The follow request make allow another user to see the posts and like on it and transfer request is indicate that user (employee) wants to transfer with another user(employee).

**Objective**

* Give a platform to government bank employee to find a person for mutual transfer.
* To provide a platform where they all come together and view each other job profile to find a perfect match for them.
* To make them share their thoughts with each other.
* To make them chat with each other.

**Project category**

* Web based application

a website **provides a quick and easy way of communicating information between buyers and sellers**. You can list your opening hours, contact information, show images of your location or products, and use contact forms to facilitate enquiries from potential customers or feedback from existing ones.

**website is accessible across platforms and can be easily shared among users, as well as search engines, it has far greater reach capability than a native app**.

**Tools / Environment used (Hardware and Software used along with purpose)**

* **Developer side**

|  |  |
| --- | --- |
| **Hardware configuration** | |
| **Hard disk** | 500 GB |
| **RAM** | 4 GB |
| **Processor** | Intel®CoreTM-i3-4 |
| **Software configuration** | |
| **Operating system** | Microsoft windows 10 |
| **Front-end** | PHP (Laravel framework) Bootstrap, HTML, CSS, Ajax. |
| **Back-end** | MYSQL, Socket |
| **Web server** | XAMP |

* **Server side**

|  |  |
| --- | --- |
| **Hardware configuration** | |
| **Hard disk** | 40 GB |
| **RAM** | 1 GB |
| **Processor** | Intel Premium 4 and above |
| **Software configuration** | |
| **Operating system** | Microsoft windows 10 and above |
| **Front-end** | PHP (Laravel framework) Bootstrap, HTML, CSS, Ajax. |
| **Back-end** | MYSQL, Socket |

* **Client side**

|  |  |
| --- | --- |
| **Hardware configuration** | |
| **Hard disk** | 50 GB |
| **RAM** | 512 MB |
| **Processor** | Intel Premium 4 and above |
| **Software configuration** | |
| **Operating system** | Microsoft windows 10 and above |
| **Browser** | 100 MB |
| **Web server** | XAMP |

**Technologies:**

1. **PHP**

* **What is PHP?**
* PHP is an acronym for "PHP: Hypertext Pre-processor"
* PHP is a widely-used, open-source scripting language
* PHP scripts are executed on the server
* PHP is free to download and use
* **PHP is an amazing and popular language!**
* It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!  
  It is deep enough to run large social networks!  
  It is also easy enough to be a beginner's first server-side language!

## What is a PHP File?

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code is executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".**php**"

## What Can PHP Do?

* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can be used to control user-access
* PHP can encrypt data

1. **Laravel framework:**

* Laravel is a PHP based web framework for building high-end web applications using its significant and graceful syntaxes. It comes with a robust collection of tools and provides application architecture. Moreover, it includes various characteristics of technologies like ASP.NET MVC, codeigniter, Ruby on Rails, and many more. This framework is an open-source framework. It facilitates developers by saving huge time and helps reduce thinking and planning to develop the entire website from scratch.
* Along with that, the security of the application is also taken care of by Laravel. So all its features can increase the speed of web development for you. If you are familiar with intermediate PHP scripting and PHP basics, Laravel can prepare your work more efficiently.

1. **MYSQL:**

## What is MySQL?

* MySQL is a relational database management system
* MySQL is open-source
* MySQL is free
* MySQL is ideal for both small and large applications
* MySQL is very fast, reliable, scalable, and easy to use
* MySQL is cross-platform
* MySQL is compliant with the ANSI SQL standard
* MySQL was first released in 1995
* MySQL is developed, distributed, and supported by Oracle Corporation
* MySQL is named after co-founder Monty Wideness’s daughter: My

## Who Uses MySQL?

* Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
* Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
* A very large number of web developers around the world

1. **Socket:**

* Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard Unix file descriptors. In Unix, every I/O action is done by writing or reading a file descriptor. A file descriptor is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else.
* To a programmer, a socket looks and behaves much like a low-level file descriptor. This is because commands such as read() and write() work with sockets in the same way they do with files and pipes.
* Sockets were first introduced in 2.1BSD and subsequently refined into their current form with 4.2BSD. The sockets feature is now available with most current UNIX system releases.

## Where is Socket Used?

* A Unix Socket is used in a client-server application framework. A server is a process that performs some functions on request from a client. Most of the application-level protocols like FTP, SMTP, and POP3 make use of sockets to establish connection between client and server and then for exchanging data.

1. **AJAX:**

## What is AJAX?

* AJAX = **A**synchronous **J**avaScript **A**nd **X**ML.
* AJAX is not a programming language.
* AJAX just uses a combination of:
* A browser built-in XMLHttpRequest object (to request data from a web server)
* JavaScript and HTML DOM (to display or use the data)
* **AJAX is a developer's dream, because you can:**
* Update a web page without reloading the page
* Request data from a server - after the page has loaded
* Receive data from a server - after the page has loaded
* Send data to a server - in the background

1. **BOOTSTRAP:**

* Bootstrap is **a free and open-source tool collection for creating responsive websites and web applications**. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue.

1. **CSS:**

* CSS stands for Cascading Style Sheets.
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
* CSS saves a lot of work. It can control the layout of multiple web pages all at once.
* External stylesheets are stored in CSS files.

1. **HTML:**

* HTML stands for Hyper Text Markup Language.
* HTML is the standard markup language for creating Web pages.
* HTML describes the structure of a Web page.
* HTML consists of a series of elements.
* HTML elements tell the browser how to display the content.

**Current (Existing System)**

There is manual system for Mutual Transfer in government bank employees. They have to find a person manually for transfer.

**Drawback Or Limitation of Current (Existing System)**

* Manual search.
* Take a too much time and in some case, it is waste of time.
* Long process for searching.
* In some case if a person finds a person and for transfer and that two-person post in bank are not matched then they can’t transfer. Even they find a same post person but not same but not they want to go at each other place, means if two person a want to get mutual transfer and for that they find a person, one person is working on Surat branch and he/she wants to go at Navsari and another person is working in Baroda city branch and wants to go at Gandhinagar then that person post are match but transfer place are not matched.

**Requirement Specification**

* As we have a see the drawbacks of current system, means we have to make system which remove a current system drawbacks and limitations.
* There should a system which provide a platform which is not manual, means for transfer a searching should not mutual, they get a searching result easily on the click of button.
* There should a platform that provide them a searching according to their requirements, means a platform where they can fill their searching detail and find a perfect match according to their requirements. with help of this feature, they get searching result which they wants means they get the candidate’s which is in that place, which place they select while filing searching form and same as post in bank.
* There should a system which provide a feature with help of employee are talk with each and clear there all the doubts.
* There should a system which provide them to share their each other thoughts with each other.
* There should a platform which provide them a each and every notification of each other activities. Which make them more and more interactive which each other, Reactions and comments on their shared posts.
* There should a platform that makes them do changes in their job profile, means after some time employees’ position, placed state, placed city, and also branch is changes then they can update their profile easily.

**Proposed System**

**Users of proposed system:**

* Admin
* Bank Employee
* Registered user
* Guest User

**Admin**:

* login
* Authenticate User
* Verify documentation
* Put recommendation
* Change password
* Forgot password
* Block or unblock bank employee
* Logout

**Bank Employee:**

* login
* Make New Job Profile
* View profile each other’s
* Search match for transfer
* Accept transfer request
* Send follow request
* Accept follow request
* Share post
* Like post
* Comment on post
* Delete own post
* Get notification of each other activities
* Edit profile
* Edit Job profile
* Change password
* Forgot password
* Change profile photo
* Dislike post
* Chat with each other
* Logout

**Registered user**:

* Login
* Make job profile
* Edit profile
* Change password
* Forgot password
* Logout

**Guest User:**

* Visit website
* Register on website

**Advantages of proposed system**

* Easy search.
* Less time consuming.
* Find perfect match.
* So many choices for transfer.
* Share their thoughts.
* Chat with each other.
* Get the notification of each other activities.

**Table Structure**

* **User Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | This stores a uses id | Primary key |
| firstname | varchar | 255 | It stores first name | NOT NULL |
| lastname | Varchar | 255 | It stores last name | NOT NULL |
| username | Varchar | 255 | It stores username name | NOT NULL |
| email | varchar | 255 | It stores user email | unique |
| Date\_of\_birth | date |  | It will store a birth date | Nullable |
| State\_id | bigint | 20 | This stores a state id | NOT NULL |
| City\_name | varchar | 255 | It will store city name | NOT NULL |
| Is\_admin | tinyinteger | 4 | It will store that user is admin in 0 and 1 | Default(0) |
| password | varchar | 255 | It stores password in encrypted form | NOT NULL |
| Profile\_image | varchar | 255 | It will store profile image | Nullable |
| userstatus | tinyint | 4 | It stores user status | Default (0) |
| device\_token | Varchar | 255 | It stores device token of user | Nullable |
| token | varchar | 255 | It stores authentication token | Nullable |
| Is\_deleted | tinyint | 4 | It stores user delete status | Default (0) |
| email\_verified\_at | timestamp |  | It stores a time where user email verified | Nullable |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **State**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | inetger | 110 | It stores state unique id | Primary key |
| name | varchar | 500 | It stores state name | NOT NULL |

* **city**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | inetger | 110 | It stores city unique id | Primary key |
| name | varchar | 500 | It stores city name | NOT NULL |
| state\_id | integer | 110 | It stores state id of state table | NOT NULL |

* **Job\_Profile**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It stores user job profile unique id | Primary key |
| email | varchar | 255 | It stores user email | unique |
| First\_name | varchar | 255 | It stores first name | NOT NULL |
| Last\_name | varchar | 255 | It stores last name | NOT NULL |
| User\_id | bigint | 20 | It stores user id of users table | NOT NULL |
| post\_name | varchar | 255 | It stores bank employee post | NOT NULL |
| Date\_of\_join | date |  | It stores a joining date of employee | NOT NULL |
| State\_id | bigint | 20 | It stores a state id from states table | NOT NULL |
| City\_name | varchar | 255 | It will store city name from cities name | NOT NULL |
| Profile\_img | varchar | 255 | It stores a profile image file | NOT NULL |
| Salary\_slip | varchar | 255 | It stores a salary slip file | NOT NULL |
| Appointment\_later | varchar | 255 | It stores appointment letter file | NOT NULL |
| Bank\_id | varchar | 255 | It stores IFSC code from banks table | NOT NULL |
| salary | double | (8,2) | It will store Employee salary | Nullable |
| Is\_deleted | tinyinteger | 4 | It stores user delete status | Default (0) |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Transfer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id of transfer table | Primary key |
| job\_profile\_id | bigint | 20 | It will store job profile id of employee from job profiles table | NOT NULL |
| reference\_job\_profile\_id | bigint | 20 | It will store job profile id of employee from job profiles table who request for transfer | NOT NULL |
| status | enum |  | It will store a transfer request status in (‘accept’,’pending’,’reject’) | NOT NULL |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Post**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for every post | Primary key |
| User\_id | bigint | 20 | It stores user id of users table | NOT NULL |
| job\_profile\_id | bigint | 20 | It will store job profile id of employee from job profiles table | NOT NULL |
| title | varchar | 225 | It will store title of post | Nullable |
| File\_name | Varchar | 255 | It will store user post file name | NOT NULL |
| description | varchar | 255 | It will store a description about post | Nullable |
| Is\_deleted | tinyinteger | 4 | It stores post delete status | Default (0) |
| View\_count | int | 11 | It will store a count of view in post | Nullable |
| Like\_count | int | 11 | It will store a count of like in post | Nullable |
| Comment\_count | int | 11 | It will store a count of comment in post | Nullable |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Post\_like**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| Id | bigint | 20 | It will store unique id for every post like | Primary key |
| User\_id | bigint | 20 | It stores user id of users table | NOT NULL |
| Post\_id | bigint | 20 | It will store post id from post table | NOT NULL |
| Is\_like | tinyinteger | 4 | It will store user is liked or not in 1 and 0 | Default (1) |
| count | int | 11 | It will store like count | Nullable |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Post\_comment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for every post comment | Primary key |
| User\_id | bigint | 20 | It stores user id of users table | NOT NULL |
| Post\_id | bigint | 20 | It will store post id from post table | NOT NULL |
| comment | varchar | 255 | It will store user comment on post | NOT NULL |
| count | int | 4 | It will store count of post comment | Nullable |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Connections**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for every connection request | Primary key |
| User\_id | bigint | 20 | It stores user id of users table | NOT NULL |
| Followed\_id | bigint | 20 | It will store followed user id | NOT NULL |
| Is\_follow | tinyinteger | 4 | It will store user is follow or not in 1 and 0 | Default(0) |
| status | enum |  | It will store a follow request status in (‘active,’pending’) | NOT NULL |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Job\_transfer\_search**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It stores user job transfer search unique id | Primary key |
| User\_id | bigint | 20 | It stores user id from users | NOT NULL |
| Job\_profile\_id | bigint | 20 | It stores user job profile id of job profile table | NOT NULL |
| State\_id | int | 110 | It stores a state id from states table | NOT NULL |
| City\_name | varchar | 255 | It will store city name from cities name | NOT NULL |
| Bank\_id | varchar | 255 | It stores IFSC code from banks table | NOT NULL |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Bank**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for banks | Primary key |
| ifsc | varchar | 20 | It will store IFSC code of bank |  |
| Branch\_name | varchar | 255 | It will store branch name |  |
| state | varchar | 255 | It will store state name of bank |  |
| city | varchar | 255 | It will store city name of bank |  |
| phone | varchar | 20 | It will store phone number of banks |  |

* **Ads**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for every ad | Primary key |
| Client\_name | varchar | 255 | It stores client name | NOT NULL |
| email | varchar | 255 | It will store client email | NOT NULL |
| title | varchar | 255 | It will store title of ads | Nullable |
| Ads\_template | varchar | 255 | It will store a template of ads | NOT NULL |
| View\_count | int | 4 | It will store count ads | Nullable |
| description | varchar | 255 | It will store description of ads | Nullable |
| Per\_view\_amount | double | 4 | It will store per view amount | NOT NULL |
| Is\_deleted | tinyinteger | 2 | It will store given ad is deleted or not in 0 and 1 | Default(0) |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Recommendation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for every recommendation | Primary key |
| Client\_name | bigint | 20 | It stores client name | NOT NULL |
| email | bigint | 20 | It will store client email | NOT NULL |
| title | varchar | 255 | It will store title of recommendation | Nullable |
| Ads\_template | varchar | 255 | It will store a template of recommendation | NOT NULL |
| View\_count | int | 4 | It will store count recommendation | Nullable |
| description | varchar | 255 | It will store description of ads | Nullable |
| Per\_view\_amount | double | 4 | It will store per view amount | NOT NULL |
| Is\_deleted | tinyinteger | 2 | It will store given recommendation is deleted or not in 0 and 1 | Default(0) |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Chat**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | biginy | 20 | It stores chat unique id | Primary key |
| Sender\_id | bigint | 20 | It stores job\_profile\_id of sender | NOT NULL |
| Receiver\_id | bigint | 20 | It stores job\_profile\_id of receiver | NOT NULL |
| Is\_read | tinyinteger | 4 | It will store message is read or not in 0 and 1 | Default (0) |
| type | enum |  | It will store which type of chat in(‘chat’,’lbl’) | NOT NULL |
| message | varchar | 255 | It will store a message of chat | NOT NULL |
| Is\_deleted | tinyinteger | 4 | It will store that message is deleted or not in 0 and 1 | Default (0) |
| Is\_deleted\_for\_all | tinyinteger | 4 | It will store that message is deleted for all or not in 0 and 1 | Default (0) |
| Deleted\_by | bigint | 20 | It stores job\_profile\_id of message deleted employee | NOT NULL |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Contact\_us**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It stores conact us unique id | Primary key |
| Full\_name | varchar | 255 | It stores full name | NOT NULL |
| Email | varchar | 255 | It will store email of user | NOT NULL |
| message | varchar | 255 | It will store a message | NOT NULL |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Feedback**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It stores feedback unique id | Primary key |
| Email | varchar | 255 | It will store email of user | NOT NULL |
| subject | varchar | 255 | It will store subject | Nullable |
| msg | varchar | 255 | It will store a message | NOT NULL |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **Notifications**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It stores Notification table unique id | Primary key |
| User\_id | bigint | 20 | It will job profile id of user who get notification | NOT NULL |
| reference\_id | bigint | 20 | It will job profile id of user from notification is goes | NOT NULL |
| Is\_read | tinyinteger | 4 | It will store notification is read or not in 0 and 1 | Default(0) |
| type | enum |  | It will store which type of notification in ('chat','follow\_request',  'request\_accept','help','reminder') |  |
| Unread\_count | int | 11 | It will store unread notification count |  |
| Is\_send | tinyinteger | 4 | It will store notification is send or not in 0 and 1 |  |
| Created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |

* **notification**
* **Recommendation\_view\_profit\_count**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | Description | Constraint |
| id | bigint | 20 | It will store unique id for every recommendation\_view\_profit\_count | Primary key |
| Recommendation\_id | bigint | 20 | It will store recommendation id from recommendation table | NOT NULL |
| Job\_profile\_id | bigint | 20 | It will store job profile id of employee who view recommendation | NOT NULL |
| View\_count | int | 4 | It will store count recommendation | Nullable |
| profit | double |  | It will store a profit in percentage | Nullable |
| Total\_profit\_amount | double |  | It will store total profit amount in rupees | Nullable |
| created\_at | timestamp |  | It stores create time |  |
| updated\_at | timestamp |  | It stores update time |  |