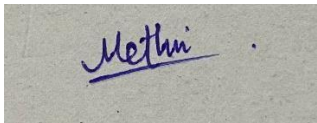
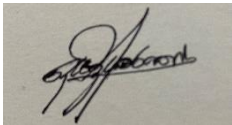
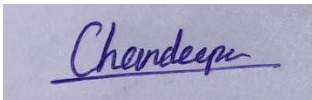
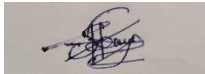



# Information Systems and Data Modeling – IT1090

## Assignment



Title: (Recruitment Company System)		
Batch Number: (CSNE.01.02)		Group Number: (MLB_CSNE_01.02_01)
<p>Declaration:</p> <p>We hold a copy of this assignment that we can produce if the original is lost or damaged.</p> <p>We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.</p>		
Group Members:		
IT23366336	A.M.N.S. WEERARATHNE	
IT23363670	E.D.O. SAMARAKOON	
IT23362994	H.M.C.M.B. HERATH	
IT23363052	K.H.S. DAMSILUNI	
IT22246028	D.S. EGALAHEWA	
Submitted on: 04/05/2024.		

## Table of Contents

<b>1.Introduction.....</b>	<b>3</b>
<b>2.Hypothetical Scenario.....</b>	<b>4</b>
<b>3. Requirement Analysis.....</b>	<b>5</b>
<b>3.1 Main Requirements .....</b>	<b>5</b>
<b>3.1.1 Functional Requirements .....</b>	<b>5</b>
<b>3.1.2 Non-Functional Requirements.....</b>	<b>6</b>
<b>3.2 Data Requirements .....</b>	<b>7</b>
<b>4.ER Diagram .....</b>	<b>10</b>
<b>5.Relational Schema.....</b>	<b>11</b>
<b>6. SQL Queries .....</b>	<b>12</b>
<b>7. Security Requirements .....</b>	<b>29</b>
<b>8. Performance Requirement.....</b>	<b>29</b>

# 1.Introduction

In today's competitive world, undergraduates and different job seekers move here and there seeking various types of jobs they like to do. In addition to that people interact with the recruitment companies to find new job vacancies in different companies in the business world. Therefore, the recruitment industry plays a crucial role for the qualified candidates to find appropriate job opportunities for them. Recruitment companies focus on connecting companies who find employees and qualified candidates who have the need. But now, the process of finding a job for an individual is quite difficult with the manual systems in the recruitment companies.

To overcome these difficulties, computer-based online recruitment company systems are introduced to the world by the recruitment industry. These days, most of the young laborers are searching for their jobs via online platforms. In fact, it has become a believable source of information for them. Through these recruitment company management systems, people and companies can interact very easily and they can achieve their job seeking and recruitment process. We have also created a computer-based recruitment company system.

When designing a recruitment company system, various tools are used. HTML, CSS, and JavaScript are among some of them. To build up this process, a well-designed database is a must. In this project, we have focused on the database of the recruitment company system. For this purpose, MySQL is used as a tool for creating the database.

A database must be able to store a large amount of data of the users in the system. A computer-based database is very easy to manage when compared to managing a manual written or typed documents. Therefore, the users and the administrators can easily use the system. In addition to that, as the data is stored in the database, the security of the data in the database is ensured because the administrators and the users of the system are the only ones who have the possibility to control and manage the database.

## **2.Hypothetical Scenario**

“ABC job solutions” is a recruitment company which provides the facility to job seekers to find their jobs and the business companies which have job vacancies to find new employees to the company. The mainly focused areas of this system are Client companies, job posting, candidates, applications, interviews, and job offerings. The companies who have joined this system can enter the system and send their job vacancies to be posted by the admin of the recruitment company system and also, they can check whether there are any pending applications for their job posts. Then the administrator receives the job vacancies of the relevant companies and approves the posts of those companies.

The job seekers can enter the system and seek for the job vacancies published by various companies, and they can apply for those job vacancies by submitting their applications. Once the application is submitted, it will be stored in the database for further processing. The system allows the administrator to schedule the interviews between the companies and candidates. After the interview, if the candidate is selected to the job by the company, the system facilitates the process of sending the job to the candidate.

In addition to this process, this system helps to provide data to generate reports on recruitment metrics like the number of job postings, number of job seekers in a certain area and the success rate of interviews.

Overall, ABC job solutions provides a comprehensive platform for the job seekers and the companies to connect mutually and fulfill their recruitment needs. This system brings a new face to the recruitment process, making it efficient and easy for all the parties involved in this process.

## **3. Requirement Analysis**

### **3.1 Main Requirements**

#### **3.1.1 Functional Requirements**

1. User Management:
  - Registration for job seekers and companies.
  - Login and authentication for registered users.
2. Company Management:
  - Ability for companies to post job vacancies.
  - View and manage their posted job vacancies.
  - Check pending applications for their job posts.
  - Receive notifications/alerts for new applications.
3. Job Seeker Management:
  - Search and view job vacancies posted by companies.
  - Apply for job vacancies by submitting applications.
  - View status of their applications.
  - Receive notifications/alerts for interview schedules and job offers.
4. Admin Management:
  - Approve job postings submitted by companies.
  - Schedule interviews between companies and candidates.
  - Manage interview schedules and notifications.
  - Process and send job offers to selected candidates.
5. Database Management:
  - Store user information (job seekers, companies, admins).
  - Store job postings, applications, and interview details.
  - Generate reports on recruitment metrics (number of job postings, number of job seekers, success rate of interviews).
6. Communication and Notification:
  - Send notifications/alerts to users for interview schedules, job offers, and application status updates.
  - Allow communication between companies, candidates, and administrators (e.g., messaging system).
7. Security and Privacy:
  - Implement secure login/authentication mechanisms.
  - Ensure data privacy and compliance with regulations.
8. Reporting and Analytics:

- Generate reports on recruitment metrics for analysis and decision-making.
- Provide insights into job market trends, candidate preferences, and company hiring patterns.

9. Feedback and Rating System:

- Allow companies and candidates to provide feedback on the recruitment process.
- Implement a rating system for companies based on candidate experiences.

### **3.1.2 Non-Functional Requirements**

Also known as quality attributes, determine how effectively the recruitment system operates beyond its core features. These aspects are equally vital, if not more so, than the system's functionalities. Failure to meet these requirements could lead to frustration and impede the recruitment process. Let's delve into some essential non-functional requirements for a recruitment company system.

#### **Responsiveness**

- The system must respond promptly to user interactions, ensuring a better experience for candidates, recruiters, and employers.

#### **Security**

- Protect sensitive data from unauthorized access. Implement encryption, authentication, and authorization.

#### **Scalability**

- The system should be able to control increased user load without compromising performance.

#### **Usability**

- Design an intuitive user interface that allows users to navigate effortlessly. Consider accessibility features for diverse users.

#### **Speed**

- The system can access more users at the same time without any errors and it must have excellent speed.

#### **User friendly**

- The system should be accessible to users with low IT literacy.

## 3.2 Data Requirements

Admin Table:

- User\_ID (Primary Key)
- Username
- Role
- Password
- Email
- Contact\_Number
- Linkedin

Application Table:

- Application\_ID (Primary Key)
- Job\_ID (Foreign Key)
- User\_ID (Foreign Key)
- Candidate\_ID (Foreign Key)
- Posting\_date
- Applied\_date

Candidate Table:

- Candidate\_ID (Primary Key)
- User\_ID (Foreign Key)
- First\_Name
- Last\_Name
- Resume
- Email
- Phone\_Number
- Linkedin
- College
- University
- Others
- Placement\_ID (Foreign Key)
- Client\_ID (Foreign Key)
- Interview\_ID (Foreign Key)

Candidate\_Experience Table:

- Candidate\_ID (Primary Key)
- Experience

Candidate\_Information Table:

- Candidate\_ID (Primary Key)
- Candidate\_Information

Candidate\_Skills Table:

- Candidate\_ID (Primary Key)
- Skill

Client Table:

- Client\_ID (Primary Key)
- Company\_Name
- Industry
- Email
- Contact\_Number
- User\_ID (Foreign Key)

Client\_Location Table:

- Client\_ID (Primary Key)
- Location

Feedback Table:

- Feedback\_ID (Primary Key)
- Rating
- User\_ID (Foreign Key)
- Candidate\_ID (Foreign Key)
- Client\_ID (Foreign Key)

Feedback\_Content Table:

- Feedback\_ID (Primary Key)
- Content

Interview Table:

- Interview\_ID (Primary Key)
- Interview\_Category
- Physical
- Online
- Client\_ID (Foreign Key)

Job\_Offer\_Details Table:

- Placement\_ID (Primary Key)
- Job\_Details



Job\_Posting Table:

- Job\_ID (Primary Key)
- Job\_Title
- Salary\_Range
- Opened
- Closed
- User\_ID (Foreign Key)
- Candidate\_ID (Foreign Key)

Job\_Posting\_Description Table:

- Job\_ID (Primary Key)
- Description

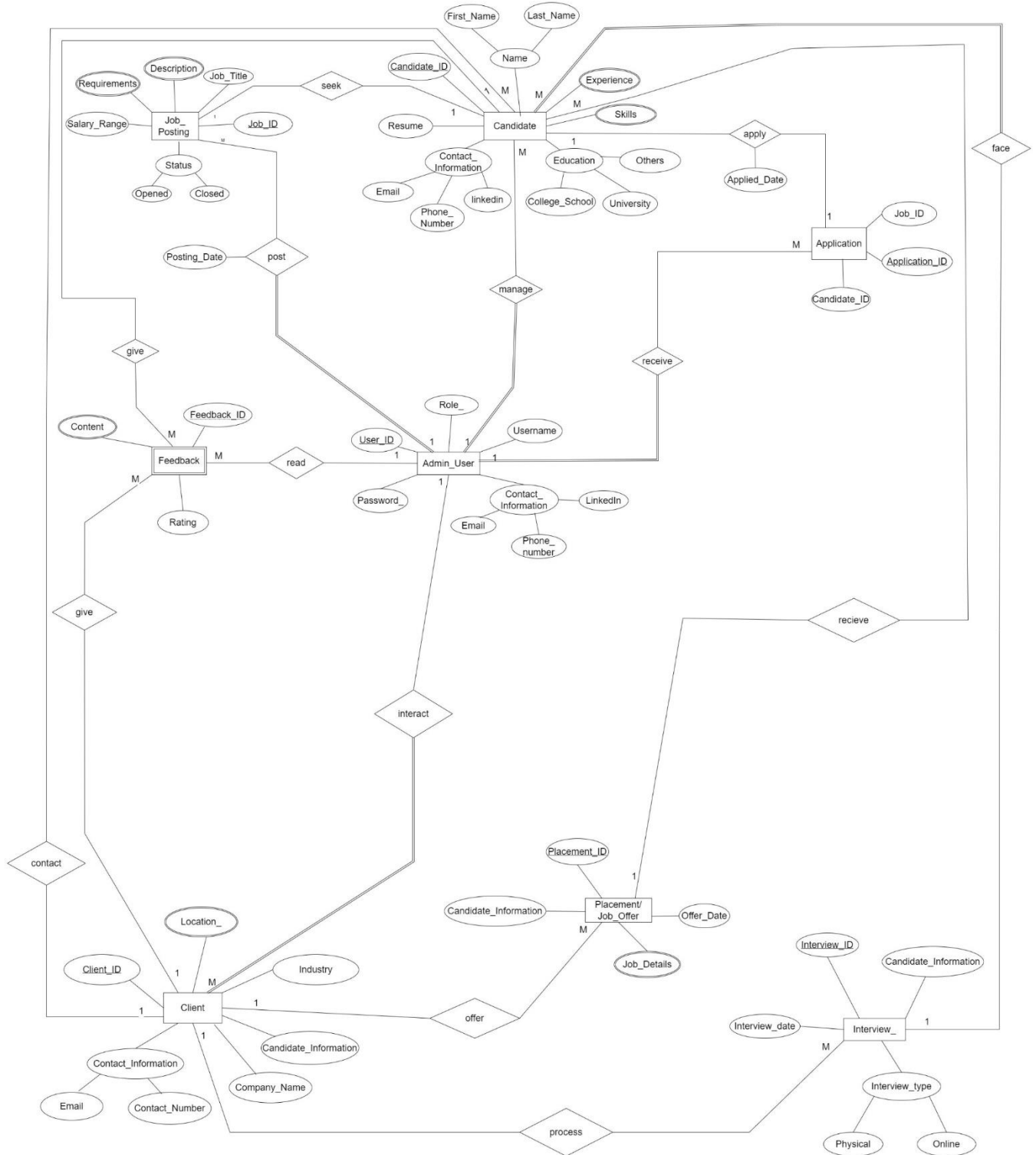
Job\_Posting\_Requirement Table:

- Job\_ID (Primary Key)
- Description

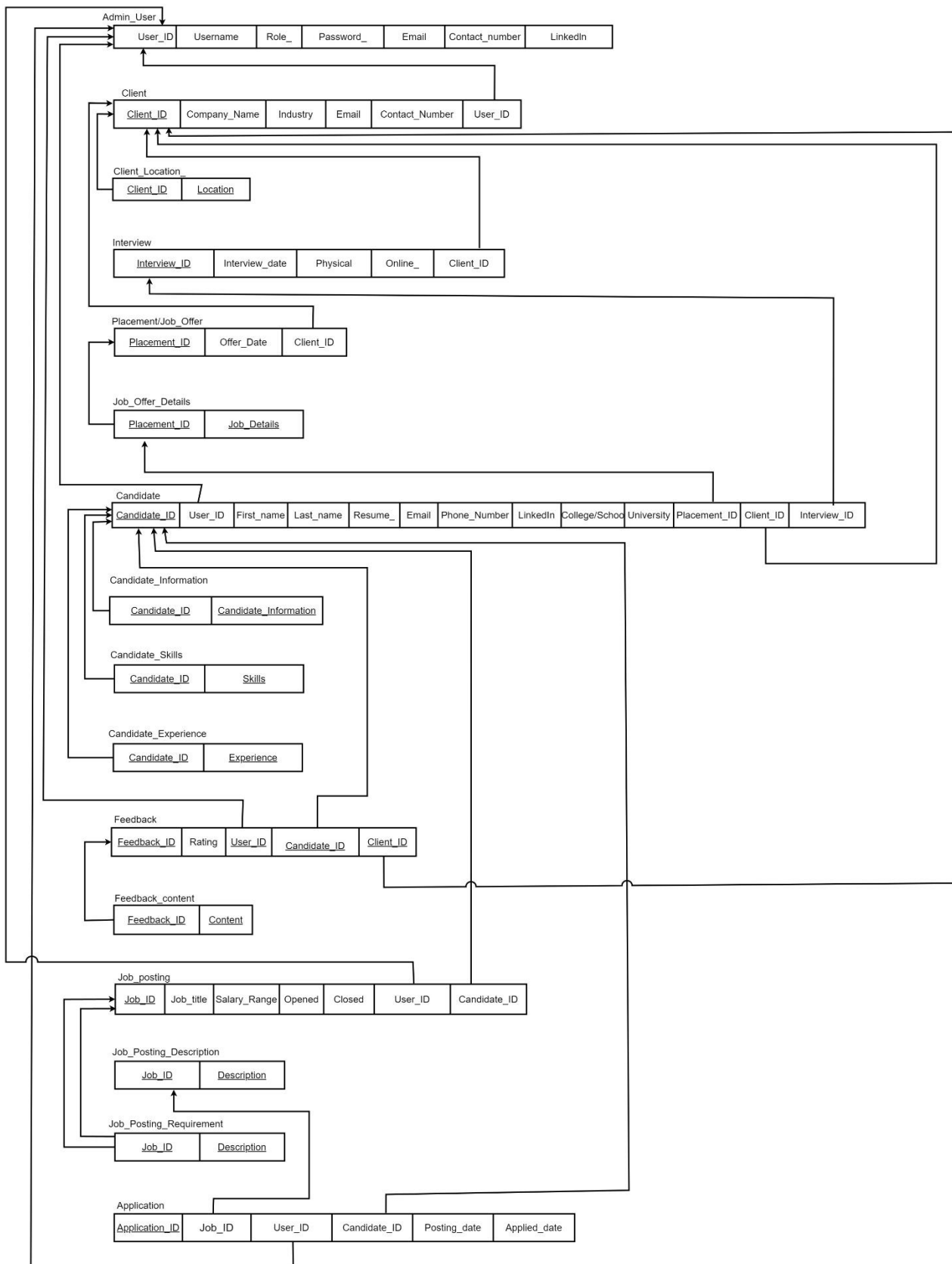
Placement\_Job\_Offer Table:

- Placement\_ID (Primary Key)
- Offer\_Date
- Client\_ID (Foreign Key)

## 4.ER Diagram



## 5.Relational Schema



## 6. SQL Queries

```
--  
-- Database: `companyrecruitment`  
--  
--  
-- Table structure for table `Admin`  
--  
  
CREATE TABLE `Admin` (  
  `User_ID` int(11) NOT NULL,  
  `Username` varchar(50) NOT NULL,  
  `Role` varchar(50) NOT NULL,  
  `Password` varchar(255) NOT NULL,  
  `Email` varchar(100) NOT NULL,  
  `Contact_Number` varchar(20) NOT NULL,  
  `Linkedin` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
--  
-- Dumping data for table `Admin`  
--  
  
INSERT INTO `Admin` (`User_ID`, `Username`, `Role`, `Password`, `Email`,  
  `Contact_Number`, `Linkedin`) VALUES  
(1, 'admin', 'Admin', 'password123', 'admin@company.com', '123-456-7890', NULL),  
(2, 'john.doe', 'Recruiter', 'secretpass', 'john.doe@company.com', '987-654-3210',  
  NULL),  
(3, 'admin1', 'Administrator', 'admin@123', 'admin@example.com', '123-456-7890',  
  'linkedin.com/admin1'),  
(4, 'admin2', 'HR Manager', 'hr@123', 'hr@example.com', '987-654-3210',  
  'linkedin.com/hrmanager');  
  
--
```

```

-- Table structure for table `Application`
--

CREATE TABLE `Application` (
  `Application_ID` int(11) NOT NULL,
  `Job_ID` int(11) DEFAULT NULL,
  `User_ID` int(11) DEFAULT NULL,
  `Candidate_ID` int(11) DEFAULT NULL,
  `Posting_date` date DEFAULT NULL,
  `Applied_date` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--

-- Table structure for table `Candidate`
--

CREATE TABLE `Candidate` (
  `Candidate_ID` int(11) NOT NULL,
  `User_ID` int(11) DEFAULT NULL,
  `First_Name` varchar(50) NOT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Resume` text NOT NULL,
  `Email` varchar(100) NOT NULL,
  `Phone_Number` varchar(20) NOT NULL,
  `Linkedin` varchar(255) DEFAULT NULL,
  `College` varchar(100) DEFAULT NULL,
  `University` varchar(100) DEFAULT NULL,
  `Others` varchar(100) DEFAULT NULL,
  `Placement_ID` int(11) DEFAULT NULL,
  `Client_ID` int(11) DEFAULT NULL,
  `Interview_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--

-- Dumping data for table `Candidate`
--

```

```

INSERT INTO `Candidate` (`Candidate_ID`, `User_ID`, `First_Name`, `Last_Name`,
`Resume`, `Email`, `Phone_Number`, `Linkedin`, `College_University_Other`,
`Placement_ID`, `Client_ID`, `Interview_ID`) VALUES
(1, 2, 'John', 'Doe', 'Path/to/resume.pdf', 'john.doe@example.com', '777-888-9999',
'linkedin.com/johndoe', 'University X', 1, 1, 1),
(2, 2, 'Jane', 'Smith', 'Path/to/resume.pdf', 'jane.smith@example.com', '333-444-
5555', 'linkedin.com/janesmith', 'College Y', 2, 2, NULL);

```

```

--
-- Table structure for table `Candidate_Experience`
--

```

```

CREATE TABLE `Candidate_Experience` (
  `Candidate_ID` int(11) NOT NULL,
  `Experience` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `Candidate_Experience`
--

```

```

INSERT INTO `Candidate_Experience` (`Candidate_ID`, `Experience`) VALUES
(1, 'Worked on multiple Java projects in previous roles.'),
(2, 'Completed internships in financial institutions during college.');
```

```

--
-- Table structure for table `Candidate_Information`
--

```

```

CREATE TABLE `Candidate_Information` (
  `Candidate_ID` int(11) NOT NULL,
  `Candidate_Information` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `Candidate_Information`

```

--

```
INSERT INTO `Candidate_Information` (`Candidate_ID`, `Candidate_Information`)
VALUES
```

```
(1, 'John Doe has 5 years of experience in software development.'),
(2, 'Jane Smith is a recent graduate with a degree in finance.');
```

--

-- Table structure for table `Candidate\_Skills`

--

```
CREATE TABLE `Candidate_Skills` (
  `Candidate_ID` int(11) NOT NULL,
  `Skill` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `Candidate\_Skills`

--

```
INSERT INTO `Candidate_Skills` (`Candidate_ID`, `Skill`) VALUES
```

```
(1, 'Java'),
(2, 'Financial Analysis');
```

--

-- Table structure for table `Client`

--

```
CREATE TABLE `Client` (
  `Client_ID` int(11) NOT NULL,
  `Company_Name` varchar(255) NOT NULL,
  `Industry` varchar(100) NOT NULL,
  `Email` varchar(100) NOT NULL,
  `Contact_Number` varchar(20) NOT NULL,
```

```

    `User_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `Client`
--

INSERT INTO `Client` (`Client_ID`, `Company_Name`, `Industry`, `Email`,
`Contact_Number`, `User_ID`) VALUES
(1, 'Company A', 'Technology', 'info@companya.com', '111-222-3333', 1),
(2, 'Company B', 'Finance', 'info@companyb.com', '444-555-6666', 1);

--
-- Table structure for table `Client_Location`
--

CREATE TABLE `Client_Location` (
  `Client_ID` int(11) NOT NULL,
  `Location` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `Client_Location`
--

INSERT INTO `Client_Location` (`Client_ID`, `Location`) VALUES
(1, 'New York, USA'),
(2, 'London, UK');

--
-- Table structure for table `Feedback`
--

CREATE TABLE `Feedback` (
  `Feedback_ID` int(11) NOT NULL,
  `Rating` int(11) NOT NULL,

```



```

`User_ID` int(11) DEFAULT NULL,
`Candidate_ID` int(11) DEFAULT NULL,
`Client_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `Feedback`
--

INSERT INTO `Feedback` (`Feedback_ID`, `Rating`, `User_ID`, `Candidate_ID`,
`Client_ID`) VALUES
(1, 4, 1, 1, 1),
(2, 5, 1, 2, 2);

--
-- Table structure for table `Feedback_Content`
--

CREATE TABLE `Feedback_Content` (
  `Feedback_ID` int(11) NOT NULL,
  `Content` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `Feedback_Content`
--

INSERT INTO `Feedback_Content` (`Feedback_ID`, `Content`) VALUES
(1, 'John Doe performed well in the technical interview.'),
(2, 'Jane Smith demonstrated strong analytical skills during the interview.');
```

```

--
-- Table structure for table `Interview`
--

CREATE TABLE `Interview` (

```

```

`Interview_ID` int(11) NOT NULL,
`Interview_Category` varchar(100) NOT NULL,
`Physical` bit(1) NOT NULL,
`Online` bit(1) NOT NULL,
`Client_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `Interview`
--

```

```

INSERT INTO `Interview` (`Interview_ID`, `Interview_Category`, `Physical`, `Online`,
`Client_ID`) VALUES
(1, 'Technical Interview', b'1', b'0', 1),
(2, 'Behavioral Interview', b'0', b'1', 2);

```

```

--
-- Table structure for table `Job_Offer_Details`
--

```

```

CREATE TABLE `Job_Offer_Details` (
  `Placement_ID` int(11) NOT NULL,
  `Job_Details` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `Job_Offer_Details`
--

```

```

INSERT INTO `Job_Offer_Details` (`Placement_ID`, `Job_Details`) VALUES
(1, 'Software Engineer position with a focus on Java development'),
(2, 'Financial Analyst position with a focus on risk assessment');

```

```

--
-- Table structure for table `Job_Posting`
--

CREATE TABLE `Job_Posting` (
  `Job_ID` int(11) NOT NULL,
  `Job_Title` varchar(255) NOT NULL,
  `Salary_Range` varchar(50) NOT NULL,
  `Opened` datetime NOT NULL,
  `Closed` datetime DEFAULT NULL,
  `User_ID` int(11) DEFAULT NULL,
  `Candidate_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `Job_Posting`
--

INSERT INTO `Job_Posting` (`Job_ID`, `Job_Title`, `Salary_Range`, `Opened`,
`Closed`, `User_ID`, `Candidate_ID`) VALUES
(1, 'Software Engineer', '$80,000 - $100,000', '2024-04-15 09:00:00', NULL, 1,
NULL),
(2, 'Financial Analyst', '$70,000 - $90,000', '2024-04-20 09:00:00', '2024-05-01
17:00:00', 1, NULL);

--
-- Table structure for table `Job_Posting_Description`
--

CREATE TABLE `Job_Posting_Description` (
  `Job_ID` int(11) NOT NULL,
  `Description` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Job_Posting_Description` (`Job_ID`, `Description`)
VALUES

```

```

(1, 'Looking for a skilled software engineer with experience in Java and Spring
framework. '),
(2, 'Seeking a creative graphic designer proficient in Adobe Creative Suite for our
marketing team. ');

--
-- Table structure for table `Job_Posting_Requirement`
--

CREATE TABLE `Job_Posting_Requirement` (
  `Job_ID` int(11) DEFAULT NULL,
  `Description` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Job_Posting_Requirement` (`Job_ID`, `Description`)
VALUES
(1, 'Bachelor's degree in Computer Science or related field, experience with Python
and Django. '),
(2, 'Minimum of 3 years of experience in marketing, strong analytical skills, and
proficiency in SEO techniques. ');

--
-- Table structure for table `Placement_Job_Offer`
--

CREATE TABLE `Placement_Job_Offer` (
  `Placement_ID` int(11) NOT NULL,
  `Offer_Date` date NOT NULL,
  `Client_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `Placement_Job_Offer`
--

```

```
INSERT INTO `Placement_Job_Offer` (`Placement_ID`, `Offer_Date`, `Client_ID`)
VALUES
```

```
(1, '2024-04-20', 1),
```

```
(2, '2024-04-25', 2);
```

```
--
```

```
-- Indexes for dumped tables
```

```
--
```

```
--
```

```
-- Indexes for table `Admin`
```

```
--
```

```
ALTER TABLE `Admin`
```

```
  ADD PRIMARY KEY (`User_ID`);
```

```
--
```

```
-- Indexes for table `Application`
```

```
--
```

```
ALTER TABLE `Application`
```

```
  ADD PRIMARY KEY (`Application_ID`),
```

```
  ADD KEY `Job_ID` (`Job_ID`),
```

```
  ADD KEY `User_ID` (`User_ID`),
```

```
  ADD KEY `Candidate_ID` (`Candidate_ID`);
```

```
--
```

```
-- Indexes for table `Candidate`
```

```
--
```

```
ALTER TABLE `Candidate`
```

```
  ADD PRIMARY KEY (`Candidate_ID`),
```

```
  ADD KEY `User_ID` (`User_ID`),
```

```
  ADD KEY `Placement_ID` (`Placement_ID`),
```

```
  ADD KEY `Client_ID` (`Client_ID`),
```

```
  ADD KEY `Interview_ID` (`Interview_ID`);
```

```
--
```

```
-- Indexes for table `Candidate_Experience`
```

```
--
```

```

ALTER TABLE `Candidate_Experience`
  ADD PRIMARY KEY (`Candidate_ID`);

--
-- Indexes for table `Candidate_Information`
--
ALTER TABLE `Candidate_Information`
  ADD PRIMARY KEY (`Candidate_ID`);

--
-- Indexes for table `Candidate_Skills`
--
ALTER TABLE `Candidate_Skills`
  ADD PRIMARY KEY (`Candidate_ID`);

--
-- Indexes for table `Client`
--
ALTER TABLE `Client`
  ADD PRIMARY KEY (`Client_ID`),
  ADD KEY `User_ID` (`User_ID`);

--
-- Indexes for table `Client_Location`
--
ALTER TABLE `Client_Location`
  ADD PRIMARY KEY (`Client_ID`);

--
-- Indexes for table `Feedback`
--
ALTER TABLE `Feedback`
  ADD PRIMARY KEY (`Feedback_ID`),
  ADD KEY `User_ID` (`User_ID`),
  ADD KEY `Candidate_ID` (`Candidate_ID`),
  ADD KEY `Client_ID` (`Client_ID`);

```

```

--
-- Indexes for table `Feedback_Content`
--
ALTER TABLE `Feedback_Content`
  ADD PRIMARY KEY (`Feedback_ID`);

--
-- Indexes for table `Interview`
--
ALTER TABLE `Interview`
  ADD PRIMARY KEY (`Interview_ID`),
  ADD KEY `Client_ID` (`Client_ID`);

--
-- Indexes for table `Job_Offer_Details`
--
ALTER TABLE `Job_Offer_Details`
  ADD PRIMARY KEY (`Placement_ID`);

--
-- Indexes for table `Job_Posting`
--
ALTER TABLE `Job_Posting`
  ADD PRIMARY KEY (`Job_ID`),
  ADD KEY `User_ID` (`User_ID`),
  ADD KEY `Candidate_ID` (`Candidate_ID`);

--
-- Indexes for table `Job_Posting_Description`
--
ALTER TABLE `Job_Posting_Description`
  ADD PRIMARY KEY (`Job_ID`);

--
-- Indexes for table `Job_Posting_Requirement`
--
ALTER TABLE `Job_Posting_Requirement`

```

```

ADD KEY `Job_ID` (`Job_ID`);

--
-- Indexes for table `Placement_Job_Offer`
--
ALTER TABLE `Placement_Job_Offer`
  ADD PRIMARY KEY (`Placement_ID`),
  ADD KEY `Client_ID` (`Client_ID`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `Admin`
--
ALTER TABLE `Admin`
  MODIFY `User_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `Candidate`
--
ALTER TABLE `Candidate`
  MODIFY `Candidate_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `Client`
--
ALTER TABLE `Client`
  MODIFY `Client_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `Feedback`
--

```



```

ALTER TABLE `Feedback`
  MODIFY `Feedback_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `Interview`
--
ALTER TABLE `Interview`
  MODIFY `Interview_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `Job_Posting`
--
ALTER TABLE `Job_Posting`
  MODIFY `Job_ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `Placement_Job_Offer`
--
ALTER TABLE `Placement_Job_Offer`
  MODIFY `Placement_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=3;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `Application`
--
ALTER TABLE `Application`
  ADD CONSTRAINT `application_ibfk_1` FOREIGN KEY (`Job_ID`) REFERENCES
`Job_Posting_Description` (`Job_ID`),
  ADD CONSTRAINT `application_ibfk_2` FOREIGN KEY (`User_ID`)
REFERENCES `Admin` (`User_ID`),
  ADD CONSTRAINT `application_ibfk_3` FOREIGN KEY (`Candidate_ID`)

```

```

REFERENCES `Candidate` (`Candidate_ID`);

--
-- Constraints for table `Candidate`
--
ALTER TABLE `Candidate`
  ADD CONSTRAINT `candidate_ibfk_1` FOREIGN KEY (`User_ID`) REFERENCES
`Admin` (`User_ID`),
  ADD CONSTRAINT `candidate_ibfk_2` FOREIGN KEY (`Placement_ID`)
REFERENCES `Placement_Job_Offer` (`Placement_ID`),
  ADD CONSTRAINT `candidate_ibfk_3` FOREIGN KEY (`Client_ID`)
REFERENCES `Client` (`Client_ID`),
  ADD CONSTRAINT `candidate_ibfk_4` FOREIGN KEY (`Interview_ID`)
REFERENCES `Interview` (`Interview_ID`);

--
-- Constraints for table `Candidate_Experience`
--
ALTER TABLE `Candidate_Experience`
  ADD CONSTRAINT `candidate_experience_ibfk_1` FOREIGN KEY
(`Candidate_ID`) REFERENCES `Candidate` (`Candidate_ID`);

--
-- Constraints for table `Candidate_Information`
--
ALTER TABLE `Candidate_Information`
  ADD CONSTRAINT `candidate_information_ibfk_1` FOREIGN KEY
(`Candidate_ID`) REFERENCES `Candidate` (`Candidate_ID`);

--
-- Constraints for table `Candidate_Skills`
--
ALTER TABLE `Candidate_Skills`
  ADD CONSTRAINT `candidate_skills_ibfk_1` FOREIGN KEY (`Candidate_ID`)
REFERENCES `Candidate` (`Candidate_ID`);

--

```

```

-- Constraints for table `Client`
--
ALTER TABLE `Client`
  ADD CONSTRAINT `client_ibfk_1` FOREIGN KEY (`User_ID`) REFERENCES
`Admin` (`User_ID`);

--
-- Constraints for table `Client_Location`
--
ALTER TABLE `Client_Location`
  ADD CONSTRAINT `client_location_ibfk_1` FOREIGN KEY (`Client_ID`)
REFERENCES `Client` (`Client_ID`);

--
-- Constraints for table `Feedback`
--
ALTER TABLE `Feedback`
  ADD CONSTRAINT `feedback_ibfk_1` FOREIGN KEY (`User_ID`) REFERENCES
`Admin` (`User_ID`),
  ADD CONSTRAINT `feedback_ibfk_2` FOREIGN KEY (`Candidate_ID`)
REFERENCES `Candidate` (`Candidate_ID`),
  ADD CONSTRAINT `feedback_ibfk_3` FOREIGN KEY (`Client_ID`) REFERENCES
`Client` (`Client_ID`);

--
-- Constraints for table `Feedback_Content`
--
ALTER TABLE `Feedback_Content`
  ADD CONSTRAINT `feedback_content_ibfk_1` FOREIGN KEY (`Feedback_ID`)
REFERENCES `Feedback` (`Feedback_ID`);

--
-- Constraints for table `Interview`
--
ALTER TABLE `Interview`
  ADD CONSTRAINT `interview_ibfk_1` FOREIGN KEY (`Client_ID`) REFERENCES
`Client` (`Client_ID`);

```

```

--
-- Constraints for table `Job_Offer_Details`
--
ALTER TABLE `Job_Offer_Details`
  ADD CONSTRAINT `job_offer_details_ibfk_1` FOREIGN KEY (`Placement_ID`)
REFERENCES `Placement_Job_Offer` (`Placement_ID`);

--
-- Constraints for table `Job_Posting`
--
ALTER TABLE `Job_Posting`
  ADD CONSTRAINT `job_posting_ibfk_1` FOREIGN KEY (`User_ID`)
REFERENCES `Admin` (`User_ID`),
  ADD CONSTRAINT `job_posting_ibfk_2` FOREIGN KEY (`Candidate_ID`)
REFERENCES `Candidate` (`Candidate_ID`);

--
-- Constraints for table `Job_Posting_Requirement`
--
ALTER TABLE `Job_Posting_Requirement`
  ADD CONSTRAINT `job_posting_requirement_ibfk_1` FOREIGN KEY (`Job_ID`)
REFERENCES `Job_Posting_Description` (`Job_ID`);

--
-- Constraints for table `Placement_Job_Offer`
--
ALTER TABLE `Placement_Job_Offer`
  ADD CONSTRAINT `placement_job_offer_ibfk_1` FOREIGN KEY (`Client_ID`)
REFERENCES `Client` (`Client_ID`);

```

## **7. Security Requirements**

- Strong password policy for all users.
- Minimum necessary permissions (SELECT, INSERT, UPDATE, DELETE).
- Multi Factor Authentication , if possible.
- Sensitive data at rest and in transit should be encrypted.
- User roles with specific permissions.
- Restrictions on network access and usage of firewalls.
- Audit tracking to follow user actions and modification activities.
- Regular review of audit logs for unusual activity.
- Security patches to MySQL software should be up to date.
- Unnecessary features disabled in MySQL setup.
- Review of MySQL settings and regulation measures.
- Backups on a regular basis and offsite backups.
- Periodic penetration testing.
- User training on security awareness.

## **8. Performance Requirement**

- Decide your database's maximum transactions and queries during peak and nonpeak periods.
- How many maximum connections or active users cause system slowdown .
- Decide to increase the server power or add more power (vertical or horizontal scaling) to your server based on current usage and future usage requirements
- Estimate how much data is stored in the current database? And how much it grows every year?.
- How frequently our data is backed up, and backup data will take how long to back up and how long terms to restore .
- Aim for uptime that hovers around 99.9%. How much downtime is acceptable?
- Simplify complex queries and optimize WHERE clause.
- Periodic possible slow-performing query review/improvement.
- Monitor CPU, memory, disk, and query performance and solve the problem.
- Aim for sub-second response times (e.g., less than 100 milliseconds) for frequently used queries that retrieve a small amount of data.