Name: Samiha Akter Maisha

Roll: 0112231010

Course code: CSE 3522

Course name: Database Management System

Section: (D)

## Introduction/Overview:
This project, 'RuralLink,' is a platform designed to connect agents, clients, and workers efficiently. The primary goal is to streamline the process of hiring skilled workers for various tasks while ensuring seamless financial transactions and referral incentives. Workers register under agents, listing their professions, skills, and necessary identification details. Agents, in turn, manage teams of workers specializing in various fields such as electrical work, construction, and more. Clients can then hire these teams for their projects at competitive prices.

## Motivation:
The need for an organized and efficient labor hiring system in rural and urban settings motivated the development of 'RuralLink.' By creating a structured platform, this project aims to enhance job accessibility, provide financial transparency, and establish a seamless process for both workers and clients.

## Similar Projects:
Existing platforms offer worker-client connectivity; however, most lack a structured agent system with direct payment integration. 'RuralLink' differentiates itself by incorporating agents as intermediaries, ensuring job accountability and efficient workforce distribution.

## Benchmark Analysis:
Compared to conventional hiring methods, 'RuralLink' provides advantages such as direct digital payments, structured referral programs, and integrated transportation services. Unlike traditional labor hiring, this system ensures transparency and fair compensation for all parties involved.

## Complete Feature List:

- Worker registration under agents with profession, skills, and identification details
- Job postings by agents showcasing available workforce teams
- Direct payments to workers via bKash, bank accounts, or Nagad
- Agents receiving payments from clients for workforce services
- Clients hiring teams at their preferred price points
- Referral system with a 5% commission for agents, workers, and clients
- Integrated transportation service for workers to job locations

## Database Design Approach:
The database was designed following normalization techniques to ensure efficiency, scalability, and data integrity. Key entities include users (clients, workers, agents), job postings, payments, referrals, and transportation logistics.

Queries for Feature Implementation:

## 1. User Role Management

- **Add a New User with Role**:

```
INSERT INTO `users` (`name`, `email`, `phone`, `password_hash`, `role`,
`created_at`)
VALUES ('John Doe', 'john.doe@example.com', '01712345678', 'john123',
'agent', current_timestamp());
```

- **Update a User’s Role**:

```
UPDATE `users`
SET `role` = 'client'
WHERE `user_id` = 3;
```

---

## 2. Permission Management for Roles

- **Add Permissions for a Role**:

```
INSERT INTO `role_permissions` (`role`, `permission_name`,
`permission_value`, `created_at`)
VALUES ('admin', 'view_dashboard', 'allowed', current_timestamp());
```

- **Update Permissions for a Role**:

```
UPDATE `role_permissions`
SET `permission_value` = 'denied'
WHERE `role` = 'agent' AND `permission_name` = 'view_dashboard';
```

- **Fetch All Permissions for a Role**:

```
SELECT `permission_name`, `permission_value`
FROM `role_permissions`
WHERE `role` = 'admin';
```

---

## 3. Searching Users (Search Index)

- **Add a Search Index for a User**:

```
INSERT INTO `searchindex` (`user_id`, `name`, `type`)
VALUES (1, 'Rahim', 'worker');
```

- **Search for a User by Name and Type**:

```
SELECT `user_id`, `name`, `type`
FROM `searchindex`
```

```
WHERE `name` LIKE '%Rahim%' AND `type` = 'worker';
```

---

## 4. Worker Management (Assigning Workers to Agents)

- **Assign a Worker to an Agent**:

```
UPDATE `workers`
SET `agent_id` = 2
WHERE `worker_id` = 1;
```

- **Fetch Workers Assigned to an Agent**:

```
SELECT `worker_id`, `name`, `contact_number`
FROM `workers`
WHERE `agent_id` = 2;
```

---

## 5. Client Profile Management

- **Add a New Client Profile**:

```
INSERT INTO `client_profiles` (`client_id`, `name`, `email`,
`contact_number`)
VALUES (1, 'Client Name', 'client@example.com', '01798765432');
```

- **Update a Client's Profile**:

```
UPDATE `client_profiles`
SET `email` = 'newemail@example.com', `contact_number` = '01712345678'
WHERE `client_id` = 1;
```

---

## 6. Role-Based Access Control

- **Check If a User Has Permission to Perform an Action**:

```
SELECT `permission_value`
FROM `role_permissions`
WHERE `role` = 'admin' AND `permission_name` = 'delete_user';
```

- **Check User's Role Before Performing an Action**:

```
SELECT `role`
FROM `users`
WHERE `user_id` = 3;
```

---

7. Auditing User Actions (Activity Log)

- **Insert a Log Entry**:

```
INSERT INTO `activity_logs` (`user_id`, `action`, `created_at`)
VALUES (2, 'Logged in', current_timestamp());
```

- **Fetch User Activity Logs**:

```
SELECT `action`, `created_at`
FROM `activity_logs`
WHERE `user_id` = 2;
```

---

8. List Users by Role

- **List All Agents**:

```
SELECT `user_id`, `name`, `email`
FROM `users`
WHERE `role` = 'agent';
```

---

### 1.1.1 9. Password Management

- **Change a User's Password**:

```
UPDATE `users`
SET `password_hash` = 'newpasswordhash'
WHERE `user_id` = 1;
```

- **Reset a User's Password**:

```
UPDATE `users`
SET `password_hash` = 'resetpassword'
WHERE `email` = 'client@example.com';
```

---

10. Deleting Users

- **Delete a User**:

```
DELETE FROM `users`
WHERE `user_id` = 3;
```

- **Delete a Worker Profile**:

```
DELETE FROM `workers`
WHERE `worker_id` = 1;
```

---

11. Worker-Client Communication History

- **Log a Communication Event**:

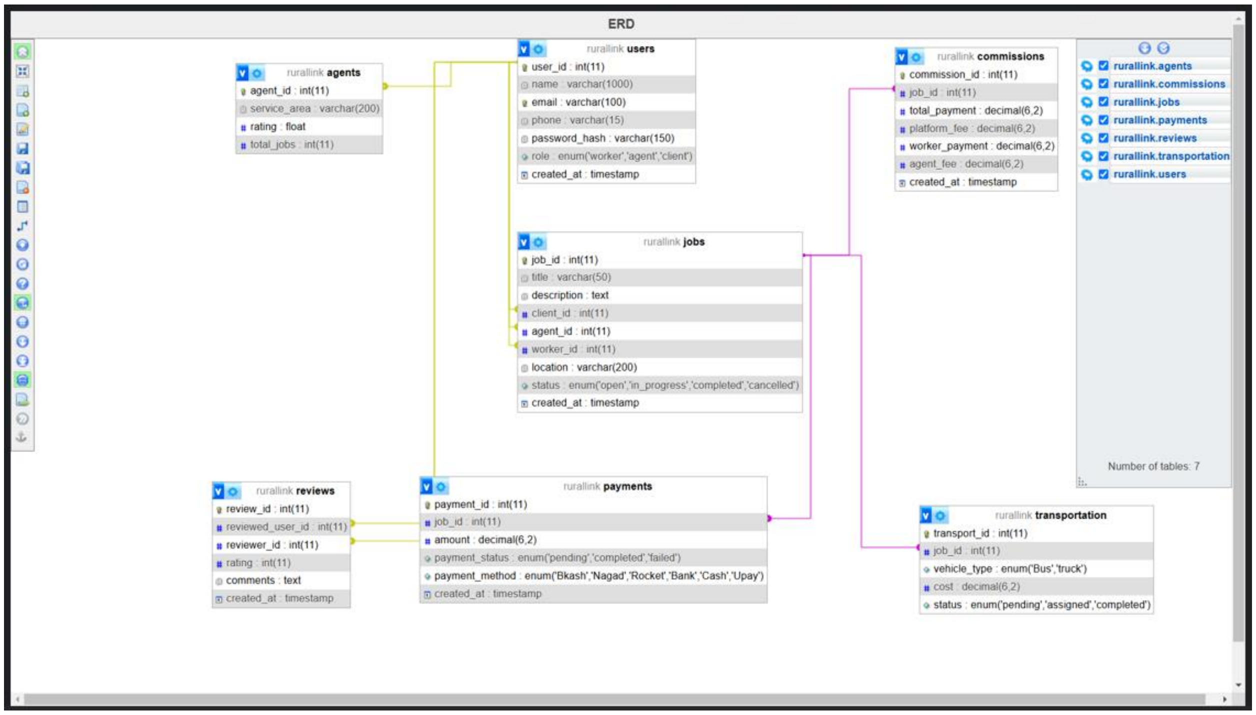
```
INSERT INTO `communication_logs` (`worker_id`, `client_id`, `message`,
`timestamp`)
VALUES (1, 3, 'Completed task successfully', current_timestamp());
```

---

12. Time-Based Access (Audit by Date)

- **Fetch Logs Within a Date Range**:

```
SELECT `user_id`, `action`, `created_at`
FROM `activity_logs`
WHERE `created_at` BETWEEN '2025-01-01' AND '2025-01-31';
```


ERD
rurallink agents
agent_id : int(11)
service_area : varchar(200)
rating : float
total_jobs : int(11)
rurallink users
user_id : int(11)
name : varchar(1000)
email : varchar(100)
phone : varchar(15)
password_hash : varchar(150)
role : enum('worker','agent','client')
created_at : timestamp
rurallink commissions
commission_id : int(11)
job_id : int(11)
total_payment : decimal(6,2)
platform_fee : decimal(6,2)
worker_payment : decimal(6,2)
agent_fee : decimal(6,2)
created_at : timestamp
rurallink jobs
job_id : int(11)
title : varchar(50)
description : text
client_id : int(11)
agent_id : int(11)
worker_id : int(11)
location : varchar(200)
status : enum('open','in_progress','completed','cancelled')
created_at : timestamp
rurallink reviews
review_id : int(11)
reviewed_user_id : int(11)
reviewer_id : int(11)
rating : int(11)
comments : text
created_at : timestamp
rurallink payments
payment_id : int(11)
job_id : int(11)
amount : decimal(6,2)
payment_status : enum('pending','completed','failed')
payment_method : enum('Bkash','Nagad','Rocket','Bank','Cash','Upay')
created_at : timestamp
rurallink transportation
transport_id : int(11)
job_id : int(11)
vehicle_type : enum('Bus','truck')
cost : decimal(6,2)
status : enum('pending','assigned','completed')
rurallink.agents
rurallink.commissions
rurallink.jobs
rurallink.payments
rurallink.reviews
rurallink.transportation
rurallink.users
Number of tables: 7

**Limitations:**
Some limitations of the current system include:

- Lack of real-time tracking for job completion
- Dependency on digital payment gateways
- Limited scalability in remote regions with low internet connectivity

**Future Work:**
Potential enhancements for future development include:

- Implementing AI-driven job matching for efficiency
- Enhancing UI/UX for better user interaction
- Expanding transportation services to include real-time tracking

**Conclusion:**
'RuralLink' successfully establishes a structured platform for agents, workers, and clients, providing an efficient hiring and payment system. The integrated referral and transportation features further enhance the usability of the platform. Future improvements will focus on scalability, automation, and enhanced user experience.