# Phase 1: Entity-Relationship Diagram (ERD)

**User**

| | |
|---|---|
| _id 🔑 | ObjectId |
| name | String |
| email | String |
| password | String |
| user_type | Enum(Freelancer,Client,Employer) |
| profile | JSON |
| is_verified | Boolean |
| created_at | Date |

**Company**

| | |
|---|---|
| _id 🔑 | ObjectId |
| employer_id | ObjectId |
| name | String |
| industry | String |
| location | String |
| website | String |
| logo | String |

**Message**

| | |
|---|---|
| _id 🔑 | ObjectId |
| sender_id | ObjectId |
| receiver_id | ObjectId |
| message | Text |
| sent_at | Date |

**UserTag**

| | |
|---|---|
| user_id | ObjectId |
| tag_id | ObjectId |
| primary | key(user_id,tag_id) |

**Tag**

| | |
|---|---|
| _id 🔑 | ObjectId |
| name | String |

**HiringRequest**

| | |
|---|---|
| _id 🔑 | ObjectId |
| job_id | ObjectId |
| freelancer_id | ObjectId |
| client_id | ObjectId |
| status | Enum(Pending,Accepted,Rejected) |
| message | Text |
| sent_at | Date |

**UserReview**

| | |
|---|---|
| reviewer_id | ObjectId |
| reviewed_id | ObjectId |
| review_id | ObjectId |
| primary | key(reviewer_id,reviewed_id,review_id) |

**Review**

| | |
|---|---|
| _id 🔑 | ObjectId |
| client_id | ObjectId |
| freelancer_id | ObjectId |
| rating | Number |
| comment | Text |
| created_at | Date |

**UserSkill**

| | |
|---|---|
| user_id | ObjectId |
| skill_id | ObjectId |
| primary | key(user_id,skill_id) |

**Skill**

| | |
|---|---|
| _id 🔑 | ObjectId |
| name | String |

# One-to-Many Relationships

## 1. User → HiringRequest (as Client)

- One client can send many hiring requests to freelancers.

## 2. User → HiringRequest (as Freelancer)

- One freelancer can receive many hiring requests from different clients.

## 3. User → Message (as Sender)

- One user can send many messages.

## 4. User → Message (as Receiver)

- One freelancer can receive many messages.

# Many-to-Many Relationships

## 1. Users ⇄ Skills

- A user can have multiple skills.
- A skill can belong to multiple users.

## 2. Users ⇄ Reviews

- A user can give multiple reviews to different professionals.
- A user can receive multiple reviews from different clients.

## 3. Users ⇄ Tags (for filtering and searchability)

- A user can have multiple tags (e.g., "Web Developer", "UI/UX", "SEO Specialist").
- A tag can be linked to multiple users.