

Feedback Received (Adrian) & Implementation Plan

Feedback Summary

I sought feedback from Adrian regarding the Entity Relationship Diagram (ERD) for my Freelancer Job Board API. Below are the key areas evaluated and the feedback received: His feedback involved security risks that were put into place following Jesso's feedback, her implementation of feedback was then removed as I couldn't have major security risks in my API

1. Clarity & Understanding

Feedback:

- The ERD correctly represents how freelancers, clients, and jobs interact.
- However, it takes some time to interpret and understand.
- Questioned the necessity of including Payments, considering the number of relationships and CRUD routes already involved.

Implementation:

- Simplified the ERD by **removing the Payments table**, as suggested.
- Enhanced documentation to improve clarity for new users reviewing the ERD.

2. Relationships & Foreign Keys

Feedback:

- Reviewed foreign key constraints and relationships between Users, Jobs, Applications, Contracts, and Payments.
- The **client_id** relationship was missing in the Contracts table.
- **Reviewer_id** and **reviewee_id** in the Reviews table might be redundant since they can be inferred from existing relationships.
- The **contract_id** relationship in Payments should allow multiple payments or no payment at all.
- Suggested nullable foreign keys for relationships that may not always exist.

Implementation:

- **Removed Payments** to reduce complexity.
- **Added client_id to Contracts** to correctly associate clients with their contracts.
- **Reviewed Reviews table**, removing reviewer_id and reviewee_id as they can be inferred from Contracts.

- Adjusted nullable foreign keys where appropriate to allow optional relationships.

3. Normalization & Data Integrity

Feedback:

- Ensure proper normalization (3NF or higher) to avoid redundancy.
- Questioned whether any data is unnecessarily duplicated.
- Ensure all primary keys and foreign keys are correctly referenced.

Implementation:

- Conducted a **final review of data normalization**, ensuring no unnecessary duplication.
- Checked all **foreign key constraints** to ensure proper data integrity enforcement.
- Updated **Contracts** and **Reviews** to minimize redundant fields.

4. Scalability & Future-Proofing

Feedback:

- Users may require additional fields such as **company name** and **employment details**.
- Future expansion may include **reviews and ratings**, which should be properly structured.
- Jobs table may need **duration** or **commitment type (full-time, part-time, etc.)** fields.

Implementation:

- **Did not add company name or employment details** in the Users table, as they were not deemed essential for the current phase of the project.
- **Ensured the Reviews table is structured efficiently** to accommodate future rating and review expansions.

5. Security & Validation

Feedback:

- Payments table contained **ip_address**, which may pose security risks.
- Need to consider proper validation for sensitive data.

Implementation:

- **Removed Payments table** altogether to avoid handling sensitive financial data.
- **Did not implement IP logging** as initially suggested by Jesso, following Adrian's feedback on security risks.

Conclusion

The feedback from Adrian highlighted key areas for improvement, particularly regarding **relationships, data integrity, and scalability**. The removal of the **Payments table** simplifies the system and reduces security risks. By refining relationships, removing unnecessary fields, and improving clarity, the Freelancer Job Board API is now more scalable, easier to maintain, and future-proofed for additional features.