

Micro-Match Marketplace Proposal

Milestone 1

Project Overview:

Title: Micro-Match Marketplace

Purpose:

The purpose of this database-driven platform is to connect students, labs, and startups who post short/long projects to contributors who can complete them. Contributors place bids on specific posted projects, which are later accepted by the client who posted them. These projects are executed through milestones that can easily be controlled by both the contributors and clients. Multiple other features include messaging, attachments, notifications, and a review system to streamline the delivery of the service.

Objectives:

- ★ Provide structured posting, bidding, and assignment workflows.
- ★ Track project progress with milestones and attachments.
- ★ Build trust through mutual reviews and messaging.
- ★ Enable skill-based search and recommendations

User Hierarchy:

There are three types of users in the hierarchy of the marketplace:

Clients: Post projects, review bids, management posted assignments/milestones, and rate contributors

Contributors: Maintain skill profiles, search/bid for projects, deliver milestones, and rate clients

Admins: Moderate content/posts, resolve disputes, and manage catalog data

Key Entities & Attributes:

- 1) **User:** Represents anyone using the platform, regardless of intended usage. This holds logging/identity, permissions, and reputation.

Attributes:

- a) user_id (PK): Unique identifier for each user
- b) name: The display name for users shown in projects, bids, and reviews
- c) email: Primary contact and login credential; must be unique
- d) primary_role: Type of user, controls permissions and UI
- e) rating_avg: Average of stars given to users across completed projects
- f) created_at: Timestamp of account creation

2) **Skill:** Represents the catalog of standardized skills that can be linked to users.

Attributes:

- a) skill_id (PK): Unique identifier for a skill
- b) skill_name: Name for a skill; must be unique

3) **UserSkill:** Links a user to a skill and captures their self-reported proficiency level

Attributes:

- a) user_id (PK; FK): The user who has the skill
- b) skill_id (PK; FK): The skill being referenced
- c) level: Current Proficiency indicated by user

4) **Project:** Defines the project type, deadline, and description of specific projects

Attributes:

- a) project_id (PK): Unique identifier for a project
- b) title: The short name for the project
- c) description: A detailed description of the problem/project with acceptance criteria
- d) budget: Client's target cost range
- e) deadline: Final due date, includes all milestones
- f) status: current stage of delivery of the project
- g) created_at: Timestamp of project creation
- h) client_id (FK): The owner (client) who posts and manages the project

5) **ProjectSkill:** Declares the skills required for a project so contributors can search and match

Attributes:

- a) project_id (PK; FK): The project being tagged
- b) skill_id (PK; FK): The skill being referenced

6) **Bid:** Captures an offer with price, timeline, and pitch. One bid may be accepted per project

Attributes:

- a) bid_id (PK): Unique identifier for a bid
- b) amount: Proposed cost for the entire assignment
- c) timeline_days: Estimated duration of completion after acceptance
- d) proposal_text: Pitch to Client explaining approach and fit
- e) status: Current outcome of the bid
- f) created_at: When the bid was submitted
- g) project_id (FK): The project the bid targets

h) contributor_id (FK): The user submitting the bid

7) **Assignment:** Created automatically when a bid is accepted; anchors milestones and delivery

Attributes:

- a) assignment_id (PK): Unique identifier for the assignment
- b) start_at: Timestamp of acceptance/activation
- c) project_id (FK): The project being executed
- d) bid_id (FK): The accepted bid that produced this assignment

8) **Milestone:** Breaks the execution of an assignment into reviewable chunks with due dates and submissions

Attributes:

- a) milestone_id (PK): Unique identifier for a milestone
- b) title: Specific deliverable or checkpoint name
- c) due_date: When the piece is due
- d) status: Progress state of the milestone towards completion
- e) submission_notes: Context or instructions provided with a submission
- f) submission_url: Link to the files/submission docs
- g) submitted_at: Timestamp when the contributor turned in the assignment
- h) assignment_id (FK): The assignment this milestone belongs to

9) **Attachment:** Stores the evidence and artifacts attached to a Project, Bid, or Milestone

Attributes:

- a) attachment_id (PK): Unique identifier for an attachment
- b) scope_type: What entity the file is attached to
- c) scope_id (FK): The specific record it belongs to
- d) url: Storage location
- e) uploaded_by (FK): Who added the attachment
- f) uploaded_at: Upload timestamp

10) **Message:** Threaded messages tie to a project for coordination and dispute protection

Attributes:

- a) message_id (PK): Unique identifier for a message
- b) project_id (FK): Project the conversation is linked to
- c) sender_id (FK): Author of the message
- d) body: Message content
- e) created_at: Send time

11) **Review:** Mutual ratings/comments once a project is completed to build trust and ranking

Attributes:

- a) review_id (PK): Unique identifier for a review
- b) project_id (FK): The project being reviewed
- c) reviewer_id (FK): Who wrote the review
- d) reviewee_id (FK): Who is being reviewed
- e) role: direction of the review (Client->Contributor/Contributor->Client)
- f) stars: Rating score for aggregation and search rating
- g) comment: extra description on the score
- h) created_at: when the review was submitted

12) **Notification:** Pushes timely updates to users

Attributes:

- a) notification_id (PK): Unique identifier for a notification
- b) user_id (FK): Recipient of the notification
- c) type: Category of the event
- d) payload_json: Event details and message
- e) created_at: When the notification was generated

Relationships & Cardinality:

- **User-Project:** 1 (Client) to N (Projects)
- **User-Bid:** 1 (Contributor) to N (Bids)
- **Project-Bid:** 1 to N; at most one Accepted bid per Project
- **Project-Assignment:** 1 to 0/1 (exactly one created upon acceptance)
- **Assignment-Milestone:** 1 to N
- **Project-Message:** 1 to N
- **User-Skill:** N to N via UserSkill
- **Project-Skill:** N to N via ProjectSkill
- **Project-Review:** 1 to N (two-way, one per reviewer->reviewee per project)
- **Scoped Attachments:** Each Attachment belongs to exactly one scope (Project, Bid, or Milestone)

Business Rules:

1) Users:

- a) Email must be unique per User.
- b) A User may hold multiple roles, but must choose a primary role for permissions
- c) A Contributor must list >1 UserSkill before bidding

2) Projects:

- a) Each Project belongs to exactly one Client (User)
- b) Each Project must declare >1 required Skill (via ProjectSkill) before publication
- c) Only one accepted bid/assignment can be active per Project at a time
- d) A Project cannot be marked **Completed** until all Milestones are **Approved**

3) Bids:

- a) A bid belongs to exactly one Project and one Contributor

- b) A Contributor may not submit more than one Bid per Project
- c) Upon accepting one Bid, all other Bids for that Project auto-change to **Rejected**

4) Assignments & Milestones:

- a) Accepting a Bid creates exactly one Assignment for the Project.
- b) An Assignment must contain >1 Milestone
- c) Each Milestone.due_date < Project.deadline
- d) A Milestone cannot close without at least one Attachment or Explicit Client approval

5) Reviews:

- a) Reviews only for completed Projects.
- b) One review per reviewer->reviewee per Project.
- c) Stars range from {1-5}

6) Skills:

- a) Users must have >1 UserSkill to bid.
- b) Projects must have >1 Project Skill to publish.

7) Attachments:

- a) Each Attachment must be scoped to exactly one of {Project, Bid, Milestone}
- b) Projects must include > Attachment before final completion

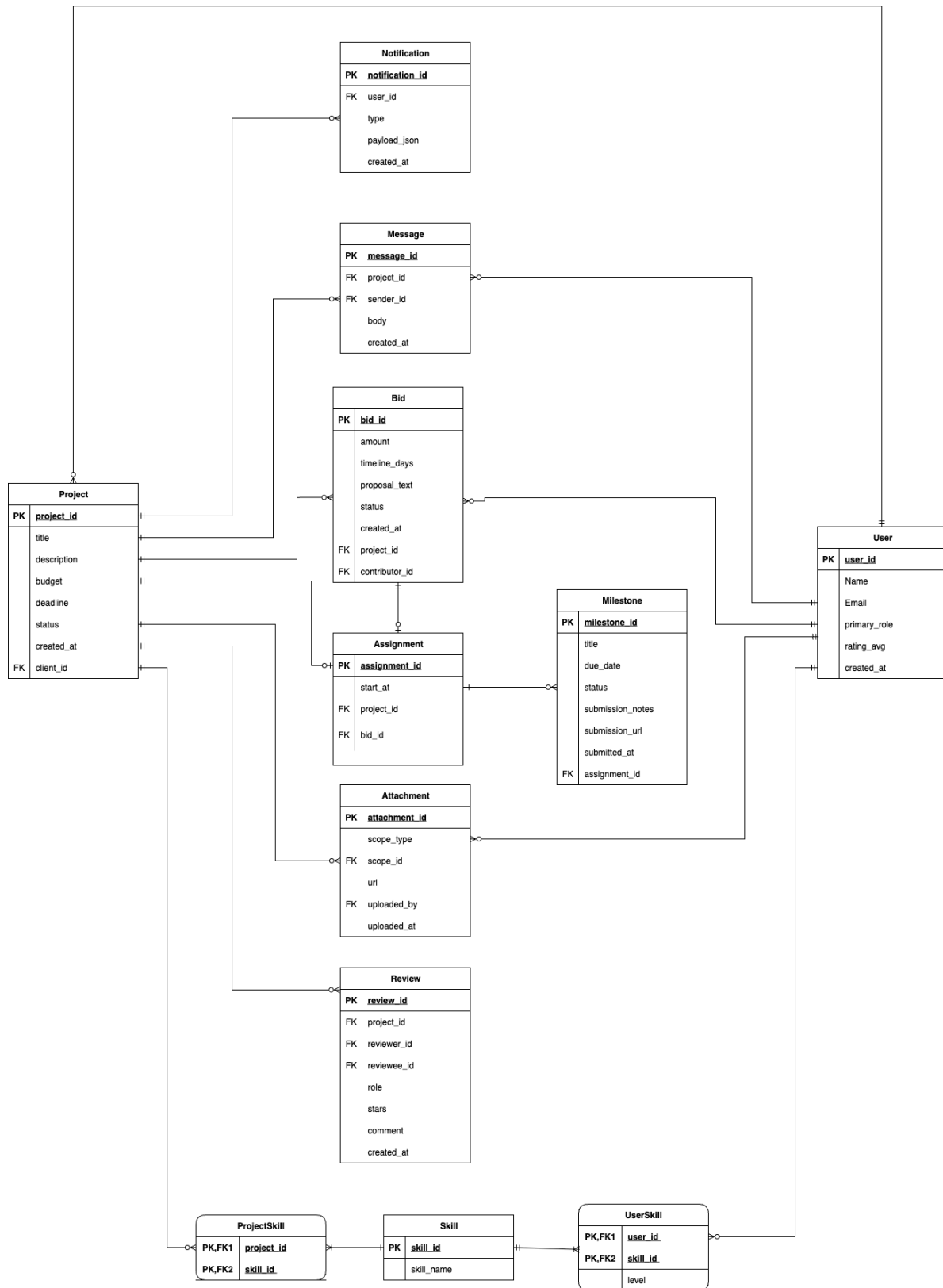
8) Messages:

- a) Each Message has a sender (User) and a receiver (User) or a system flag
- b) Messages must be tied to a Project context (and optionally to the active Assignment)
- c) Messages are not hard-deleted to preserve auditability

9) Notifications:

- a) Each Notification is tied to a triggering event
- b) Notifications belong to exactly one recipient User
- c) Deadline reminders are dispatched >12 hours before any Milestone/Project deadline

ERD:



Schema Statements:

User(user_id, name, email, primary_role, rating_avg, created_at)

Project(project_id, title, description, budget, deadline, status, created_at, @client_id)

Skill(skill_id, skill_name)

UserSkill(@user_id, @skill_id, level)

ProjectSkill(@project_id, @skill_id)

Bid(bid_id, amount, timeline_days, proposal_text, status, created_at, @project_id, @contributor_id)

Milestone(milestone_id, title, due_date, status, submission_notes, submission_url, submitted_at, @assignment_id)

Message(message_id, body, created_at, @project_id, @sender_id)

Attachment(attachment_id, scope_type, scope_id, url, uploaded_at, @uploaded_by)

Review(review_id, role, stars, comment, created_at, @project_id, @reviewer_id, @reviewee_id)

Assignment(assignment_id, start_at, @project_id, @bid_id)

Notification(notification_id, @user_id, type, payload_json, created_at)