

WhatsApp Audio 2025-12-17 at 11.05.32 PM

Transcribed by [TurboScribe.ai](#). [Go Unlimited](#) to remove this message.

So, first thing this is just about project creation, what and all a buyer wants to set a project, you have to take the details of that project and create that project, that's first thing, that UI and front end you have to manage. The second part is about whatever projects you have created, you need a place where you manage that projects. You have to see whether that project is an active state, is it created, draft state is in the sense like you have just created that project, but no freelancer has been accepted that project.

Active state, a freelancer has accepted that project and working on it. Completed state, the freelancer has submitted all the credentials and all the things to you, then it is a completed state. All these states can be managed either by the freelancer also and either by you also.

So that is one thing. Then, editing this, whatever the projects you can have before, editing status, the project edit or everything, that edit status will only be valid unless and until, yeah, it's common, like after you assign a project, that editing and all will be personal. But once you, once a freelancer has accepted you to work on the project, you can't edit the project, that one thing.

I think that thing, me and Arush, the thing, once an API you have got, you can manage it, that's a front-end task. Then that's all. The functionality of all the projects, the project management to be specific as a broad spectrum.

Now, the epic four is about, you want to search experts relevant to your problem, like I said was the first thing. The first thing was about project setting and project management. Next is about, like for example, I want to search experts regarding my project.

So just explore what is rule-based. It is not typically AI. It is like, it's like you set up some rules, based on those rules, you will get the respective experts.

You just explore what is rule-based. If you don't understand, I will set up one moment and we can discuss what is rule-based. So based on those rule-based matching, you will suggest some top three experts or top four experts.

That is that the UI, you have to figure out what UI will be there, based on that, the back-end team will help you. Yeah. So you will need to have all these criteria, search by domain, search by availability, search by R-rate.

You can search the experts like this, and you want to get suggestions. Expert search and expert suggestions is another panel. So is this clear? Yeah.

Summary of Understanding: DeepTech Hourly Marketplace MVP (Focus on Epics 3 & 4)

Based on the provided documents—the WhatsApp audio transcript (detailing front-end tasks for project handling and expert discovery) and the MVP Backlog PDF (outlining structured epics and stories)—the core **task** appears to be implementing **front-end functionality** for a buyer-facing platform in a deep-tech freelance marketplace. This aligns with **Phase-1 Credibility MVP**, emphasizing problem-first project intake and expert matching via rule-based systems. The audio positions this as a "front-end task" with backend API support from the team (e.g., Arush), while the backlog provides detailed user stories and acceptance criteria.

I'll break it down into **key components** from the documents, focusing on the discussed epics (3: Problem-First Project Intake; 4: Expert Discovery & Matching). This ensures a clear, actionable understanding for development. The platform involves **buyers** (posting deep-tech problems) and **experts** (vetted freelancers), with states managed collaboratively but locked post-assignment.

1. **Epic 3: Problem-First Project Intake (Project Creation & Management)**

- **Overview**: Buyers define and manage projects in a structured way to attract suitable experts. This is the "first thing" in the audio—capturing project details via UI, then tracking lifecycle states. Editing is flexible pre-assignment but locked afterward to prevent changes once an expert accepts.

- **Key Stories & Requirements** (from Backlog ST-301 & ST-302, aligned with Audio):

- | Story ID | User Story | Acceptance Criteria | Front-End Notes (from Audio/Backlog) |

- |-----|-----|-----|-----|

- | ST-301 | As a Buyer, I want to define my problem clearly so I can get the right expert. | - Domain selection
- Problem description (text field)
- TRL (Technology Readiness Level) selection
- Risk category (tech/regulatory/scale)
- Expected outcome type | - UI for structured form (e.g., dropdowns for domain/TRL/risk, textarea for description).
- Validate inputs before creation.
- Handle file uploads if needed (implied for deep-tech context). |

- | ST-302 | As a Buyer, I want to manage my projects in one place. | - Draft/Active/Completed status
- Editable before hiring
- Archive functionality | - Dashboard/panel to list projects with status badges (e.g., Draft: yellow, Active: green, Completed: blue).
- State transitions: Buyer or Expert can update (e.g., to Active on acceptance; to Completed on submission).
- Edit button (disabled post-acceptance).
- Search/filter by status. |

- **Audio-Specific Insights**:

- **States Defined**:

- **Draft**: Project created but no expert accepted (editable).

- **Active**: Expert accepted and working (no edits allowed).

- **Completed**: Expert submitted deliverables (e.g., credentials/insights).

- **Management Flow**: Centralized "projects" view for visibility. Use APIs for state updates (e.g., once freelancer accepts, lock edits).

- **Edge Cases**: Common edits pre-assignment; post-assignment, changes require admin intervention (not in MVP scope yet).

- **UI/UX Considerations**: Simple, intuitive forms and dashboards. Ensure mobile-responsive (WhatsApp context suggests quick access). Broad spectrum: "Project management functionality" as a holistic panel.

2. **Epic 4: Expert Discovery & Matching (Search & Recommendations)**

- **Overview**: After project creation, buyers search or get auto-suggestions for experts. This is "rule-based" (not AI-driven)—pre-defined criteria match projects to vetted experts (from Epic 2). Audio emphasizes exploring rule-based logic (e.g., set rules like domain match + availability score) to suggest top 3-4 experts.

- **Key Stories & Requirements** (from Backlog ST-401 & ST-402, aligned with Audio):

- | Story ID | User Story | Acceptance Criteria | Front-End Notes (from Audio/Backlog) |

- |-----|-----|-----|-----|

- | ST-401 | As a Buyer, I want to search experts relevant to my problem. | - Search by domain
- Filter by availability
- Filter by hourly rate | - Search bar with autocomplete (domains).
- Filters as sidebar toggles/sliders (e.g., rate range: \$50-200/hr; availability: calendar preview).
- Results grid/list with expert cards (profile pic, bio, rate, availability). |

- | ST-402 | As a Buyer, I want expert suggestions based on my project intake. | - Rule-based matching
- Shows top 3 experts
- Explainable recommendations | - Auto-generate suggestions post-project creation (e.g., modal/panel: "Top Matches").
- Display "why" (e.g., "Matched 90% on AI domain + available next week").
- Rules to explore: Domain priority > Availability > Rate (discuss with team for exact logic). |

- **Audio-Specific Insights**:

- **Rule-Based Matching**: Set criteria (e.g., if project domain = "AI" and risk = "tech", prioritize experts with matching domain + high availability + rate under budget). Suggest top 3-4; not AI, so backend provides filtered list via API.

- **Separate Panels**: One for manual search (criteria-based), another for auto-suggestions (project-linked).

- **Criteria Examples**: Search by domain, availability (e.g., calendar slots), R-rate (hourly rate).

- **UI Collaboration**: Design first (e.g., wireframes for search results), then backend builds API. "Expert search and expert suggestions is another panel."

Overall MVP Context & Dependencies

- **Sprint Structure**: 2-week sprints; target end-to-end flow (buyer hires expert, expert logs hours with value tags, payments via escrow).
- **Dependencies**:
 - Epic 1 (User Auth/Profile) for login roles.
 - Epic 2 (Expert Onboarding) for vetted pool.
 - Backend APIs for states, searches (audio confirms: "once an API you have got, you can manage it").
- **Out of Scope (for Now)**: Social login, AI matching, full disputes—focus on credibility via basic flows.
- **Exit Criteria**: 10 real engagements; ensure NDA/IP (Epics 8) and hour logging (Epic 6) tie back to project states.

This captures the **front-end focus** (UI management, panels/dashboards) while noting backend handoffs. If the task is to proceed (e.g., wireframes, code stubs, or clarify rules), let me know specifics—like sketching a rule-based example or prioritizing stories!