

Phase 2: Idea Confirmation & Planning:

A. Final Idea Confirmation

1. **Reconfirm the Chosen Idea:** Quickly double-check with everyone via your team chat (Slack, Discord, WhatsApp etc.). "Just confirming, are we all 100% decided on proceeding with [Chosen Project Idea Name]?" Get explicit agreement (👍 reactions work).
2. **Write Down Problem Statement:** In a shared document (like Notion, Google Docs, or a pinned message), clearly write 1-2 sentences defining the exact *problem* your idea solves. *Example: "Finding available and suitable study spots on campus during peak hours is frustrating and time-consuming for students."*
3. **Write Down Solution Statement:** Right below the problem, write 1-2 sentences defining your *solution*. *Example: "We are building a web app that shows real-time availability of campus study spots and allows students to filter by noise level and amenities."*
4. **Final Team Read & Confirm:** Ask everyone to read these two statements and give a final confirmation that this accurately reflects the agreed-upon project. This ensures absolute clarity before diving into planning.

B. Planning & Setup

1. **Define the Minimum Viable Product (MVP):**
 - **Brainstorm Features:** List all features you *could* build for this idea.
 - **Prioritize Ruthlessly:** Ask: "What is the absolute minimum set of features needed to *demonstrate the core value* of our solution by the submission

deadline (Sunday)?" Be realistic, especially with the 4 PM weekday start. Think core functionality, not nice-to-haves.

- **List MVP Features:** Clearly list the 3-5 core features that make up your MVP. *Example MVP Features: User Login, View Map of Study Locations, See Basic Availability Status (Available/Full), Filter by Building.* Document this list prominently!

2. Outline Key Tasks:

- Break down each MVP feature into smaller, actionable technical tasks.
- *Example for "View Map":* Setup map library (Leaflet/Mapbox), Create API endpoint to fetch location data, Display location markers on map UI, Style map markers based on availability.
- Try to estimate roughly how long each task might take (even a guess is helpful).

3. Assign Roles/Responsibilities (Initial):

- Who is strong in frontend? Backend? UI/UX design? Database? Deployment?
- Assign initial ownership for the major components or features based on skills/interest (e.g., Person A - Backend API, Person B - Frontend Map UI, Person C - Database & Auth).
- Document who is taking the lead on what, even if people collaborate.

4. Choose Preliminary Tech Stack:







- Based on the MVP and your team's collective skills/comfort level, make quick decisions. Don't get stuck here!
- *Example:* Frontend: React, Backend: Node.js/Express, Database: PostgreSQL, Map Library: Leaflet.

- Document the chosen stack.

5. Set up Development Environment:

- **Code Repository:** Create a GitHub/GitLab/Bitbucket repository. Invite all team members. Initialize it (git init, add a README).
- **Communication:** Confirm your main channel (Slack/Discord?). Create specific channels if helpful (#dev, #design, #general).
- **Project Management:** Set up a simple board (Trello, Notion Kanban, GitHub Projects). Create columns like To Do, In Progress, Done. Add the tasks identified in step B.2 to the To Do column. Assign owners based on step B.3.
- **Shared Resources:** Create a shared document (if not already done) or pinned message with links to the repo, project board, and any other key resources.

Goal Achieved When:

-  Everyone knows the exact problem & solution statement.
-  The core MVP features are clearly listed and agreed upon.
-  Major tasks are outlined and visible on a project board.
-  Initial roles are assigned.
-  Tech stack is chosen.
-  Code repository and communication channels are set up.