⚙️ PHASE 1: PROJECT SETUP (WEEK 1)

✅ Step 1: Create GitHub Repository

What to do:

- Make a private repo named track2crack.
- Set up folders: /client (frontend), /server (backend), /docs (for PRD, notes).

What you need:

- GitHub account
- Git + VS Code installed locally

Output:

Repo initialized and project structured for development.

✅ Step 2: Choose the Tech Stack (confirmed)

Layer	Tech
Frontend	React.js (w/ Tailwind)
Backend	Node.js + Express
DB	MongoDB (Atlas)
Auth	Firebase/Auth0
Hosting	Vercel (frontend), Render or Railway (backend)

Output:

Finalized stack. Tools installed locally (Node, npm, MongoDB Compass).

✅ Step 3: Create PRD (Product Requirements Document)

What to do:

- Define core features of MVP:
    1. User login (Firebase)
    2. Add Codeforces handle
    3. Fetch solved problems
    4. Generate quiz from random past problems
    5. Track incorrect/weak topics
    6. Show basic dashboard

I can write this PRD for you if you want.

Output:

Written scope document for MVP development.

💻 PHASE 2: MVP DEVELOPMENT (WEEK 2–5)

✅ Step 4: Build Backend (API Server)

Tasks:

- Set up Node.js + Express in /server
- Connect to MongoDB Atlas
- Create models: User, Quiz, WeakTopic
- Create APIs:
    - POST /register
    - POST /connect-codeforces
    - GET /fetch-codeforces-data
    - POST /submit-quiz

What you need:

- Postman (for testing API)
- MongoDB Atlas setup

Output:

Functional API server + database setup.

✅ Step 5: Build Frontend

Tasks:

- React setup in /client
- Pages:
    - Login/Register
    - Dashboard
    - Quiz page
    - Weak topics view

Use Tailwind CSS for speed.

Output:

Working UI + connected to backend.

✅ Step 6: Platform Integration — Codeforces First

What to do:

- Use Codeforces public API:
  https://codeforces.com/api/user.status?handle=your_handle
- Extract problems → filter by verdict: OK
- Map by tags → store in MongoDB

Output:

Codeforces progress shown on dashboard + used for quiz generation.

✅ Step 7: Basic Quiz Engine

What to do:

- Randomly select 3–5 previously solved problems
- Generate MCQ-style quiz or problem-based questions
- If user skips/answers wrong → mark tag as weak

Output:

Working revision quiz + weak topic tracker

🧪 PHASE 3: POLISH & TESTING (WEEK 6–7)

✅ Step 8: Add Dashboard

What to do:

- Track:
  - Total problems solved
  - Weekly quiz stats
  - Weak topic graph

Use Chart.js or Recharts.

Output:

Real-time, interactive dashboard

✅ Step 9: User Testing & Debugging

What to do:

- Deploy MVP to test link
- Share to 5–10 friends/classmates
- Collect feedback → iterate

Output:

MVP validation from real users

🚀 PHASE 4: LAUNCH & GROWTH (WEEK 8+)

✅ Step 10: Deploy

Frontend: Vercel
Backend: Render/Railway
DB: MongoDB Atlas
Domain: track2crack.in (GoDaddy)

✅ Step 11: Market & Grow

- Reach out to coding communities
- Make Instagram, LinkedIn page
- Add beta tester leaderboard
- Run college-level contests using your portal