

[BUILD THEORY FELLOWSHIP · CASE STUDY / 003]

From student projects to a shipped product.

Two third-year engineering students who already knew how to build projects. This is the record of the 32 days in which they learned to build Monetiq, a real product, and put it in front of real users.

DAY 0

A broad idea about personal finance. No product yet.

DAY 32

Monetiq, deployed and live, used by friends and professors.



≡ Ch Srikar

FELLOW 008



≡ Paarthu B

FELLOW 009

They knew how to build. They didn't yet know how to ship.

Srikar and Paarthu entered Build Theory as third-year engineering students, **not starting from zero**, but without a structured process behind their work.

[BEFORE BUILD THEORY]

- 01 Srikar from K LH University, Paarthu from GITAM University, both with minor college and hackathon projects behind them.
- 02 They knew how to open a dev environment and start building. The gap was process, not technical skill.
- 03 Their past projects stayed projects. None had reached real users.

[WHY THEY JOINED]

- 01 A referral from Ch Sreemayee, a Build Theory fellow, started the conversation.
- 02 Srikar didn't want to build alone, so he brought in his friend Paarthu.
- 03 The goal at the start was practical: a complete project for their resumes.

2

UNIVERSITIES, ONE SHARED
DECISION

0

PRODUCTS EITHER HAD
DEPLOYED BEFORE

1

REFERRAL THAT STARTED
THE CONVERSATION

“We had done projects before, but this was the first time we understood the process we can actually follow.”

Srikar & Paarthu • FELLOWS 008 • 009

A big idea about finance became one specific problem.

The first intervention was not technical. It was strategic: **deciding exactly what they were building, and for whom.** That meant choosing an ICP, an ideal customer profile: the specific type of user a product is built for.

STAGE	WHAT CHANGED
IDEATION	Arrived thinking about finance as an entire sector: expenses, taxes, loans, investments.
INTERVENTION	Introduced to starting with an ICP instead of attempting to solve an entire market.
ICP	Chose students as their initial audience.
REFINEMENT	Even inside “students,” taxes was knowledge to have, not an urgent problem to solve.
DIRECTION	Narrowed to managing expenses and understanding financial behaviour.

The problem itself was already familiar to them, from overspending and subscription overload to poor budgeting and limited investment knowledge. The lesson was not listing every problem finance could touch. It was choosing **one specific problem worth building around.**

“We learned that we should not directly jump into building. First we need clarity on the idea, the problem, the users, and what exactly we are going to build.”

The idea became a system, then a map of the whole product.

After the Ideation Document, they moved into the Product Requirements Document, then the MVP.

[PRD]

Product requirements document. Defines what the product must do, in detail, before anyone starts building it.

[MVP]

Minimum viable product. The smallest complete version of the product worth putting in front of real users.

MVP CAPABILITY

BROKEN INTO

AUTHENTICATION

Registration, login, password recovery, email verification, logout.

OCR SCANNER

Upload a receipt, extract merchant, amount and date, categorise, save.

BANK STATEMENTS

PDF/CSV upload, transaction analysis, income vs expense, reports.

DEBT & ANALYTICS

EMI and interest calculators, loan-closure suggestions, budget and savings charts.

AI CHATBOT

Budget advice, savings tips, cost cutting, loan suggestions.

The hardest part was not writing the document. It was thinking clearly enough about the product to write it down.

The MVP became their turning point: a visual and logical map of the product before they designed a single screen.

The product found its name. AI became a design collaborator.

The project went through naming changes before the team settled on Monetiq. The earlier working name was Finly AI, “finance made friendly.”

[FINAL IDENTITY]

MONETIQ

A personal finance platform designed to help users understand expenses, manage financial health, and make better spending decisions.

[WORKING WITH CLAUDE DESIGN]

- 01 Once the system, name, type and colours were set, they wrote a structured prompt instead of asking AI to “make an app.”
- 02 Claude Design generated the first screens. They reviewed, changed and iterated, never treating it as final.
- 03 Build Theory pushed a harder lesson: judge the design as the user, not as the person who built it.

[WHY A SYSTEM FIRST]

A design system, fixed colours, type, spacing and components, turns a blank prompt into a set of rules. Give an AI generator a system to follow and every screen comes back speaking the same visual language, instead of reinventing itself each time.

“You should see each and every screen and judge the design according to the user, not according to yourself.”

Connected layers, run through a structured loop.

Before development, they learned the role each technology plays, and integrated Google OAuth for the first time.

TECHNOLOGY	ROLE IN THE PROJECT
TYPESCRIPT	Application development with typed code.
REACT & NEXT.JS	Frontend application and framework.
SUPABASE & POSTGRESQL	Backend platform and database.
NETLIFY	Deployment.
ZOD	Validation.
VITEST & PLAYWRIGHT	Testing and end-to-end testing.
TANSTACK QUERY	Data fetching and server-state management.
GOOGLE OAUTH	Authentication integration, their first time implementing OAuth.
GITHUB & VS CODE	Source control and local development workflow.

[THE AI-NATIVE LOOP]

The build ran as a repeatable loop across backend, frontend, integration and QA, instead of one vague instruction: understand the phase, prompt Claude Code, let it implement, then carry the handoff into the next phase.

4

DEVELOPMENT PHASES

2

DAYS TO COMPLETE THEM

“If you ask for the UI, it builds the UI. If you ask for the connections, it works on the connections. We learned how to divide the work.”

They learned to navigate the codebase, then to ship it safely.

The project was cloned locally through VS Code. They walked through the structure: where the frontend lived, where backend logic sat, where tests belonged.

Monetiq has two user types, a normal user and a super admin behind a separate `/admin` route, a detail they had to relearn to test properly.

When a bug appears, they now have a starting point: enter the project, identify the relevant layer, inspect what is happening, and use AI with the right context to fix it.

[GETTING TO PRODUCTION]

- 01 Ran security checks and fixed what surfaced.
- 02 Deployed through Netlify, read the errors it threw, and fixed them without waiting for help.
- 03 Retested, removed seeded data, and created fresh super-admin accounts.

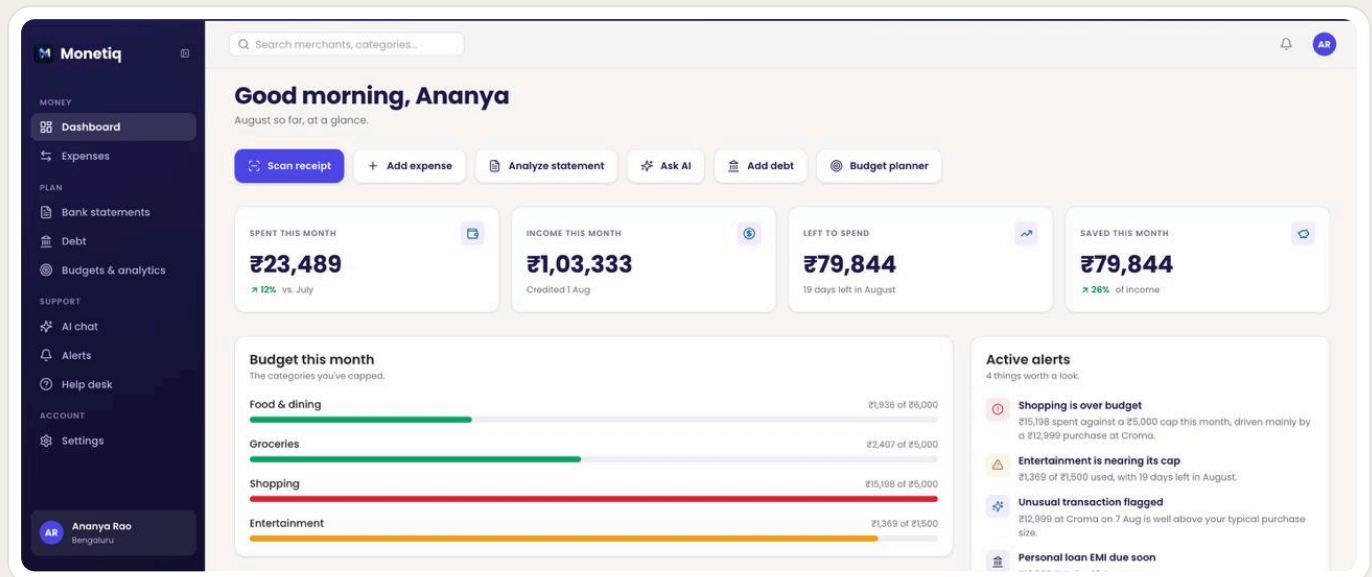
[WHY DEPLOYMENT MATTERS]

A project that only runs on a laptop is a demo. A deployed product lives on the open internet, reachable by a URL, usable by anyone, whether or not the person who built it is in the room.

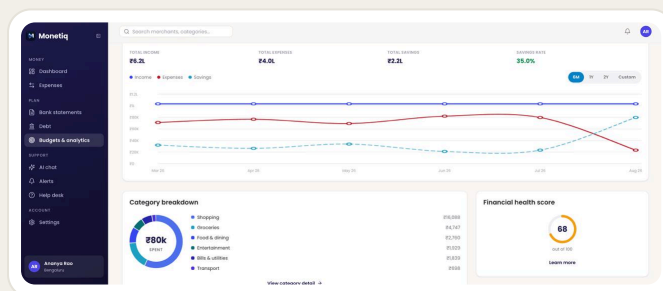
“This was the first project we made that actually got deployed and was fully working for users.”

Monetiq.

Designed and built by Srikar and Paarthu. Deployed, and used by real people.

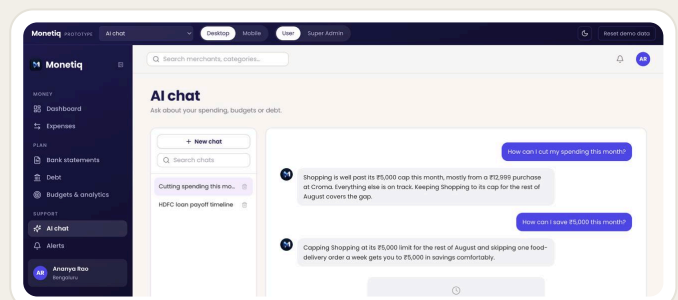


DASHBOARD Spend, income, and active alerts at a glance, with quick actions for scanning receipts and asking the AI.



ANALYTICS

Category breakdown and a financial health score, built on top of the analysis engine.



AI CHAT

Ask about spending, budgets or debt, and get a direct answer.

[LIVE TODAY]

“Even though we are the super admins, we would use this application personally as users too.”

monetiq-app.netlify.app



What changed in 32 days.

The output was a product. **The deeper output was a new way of thinking.**

BEFORE BUILD THEORY	AFTER 32 DAYS
Idea → start building	Problem → ICP → product definition → build
Broad ideas, features considered at a high level	Narrower problem, features broken into sub-features and flows
Design while building	Design system before screen generation
AI as a tool for prompts and code	AI used through structured, phase-specific workflows
Code as something to write	Codebase as something to understand and navigate
Projects remained undeployed	Production deployment, fixed by them when it broke
Project to put on a resume	Deployed product to demonstrate and discuss

Two months on, the column on the right is simply how they default to working now.

Twelve skills. Two students, one product.

Named by Srikar and Paarthu themselves, in their own interview.

01 Ideation

Turning an interest into one buildable idea.

02 Problem clarity

Naming the one problem worth solving.

03 Prompting

Instructions specific enough for AI to act on.

04 Product explanation

Explaining the product to someone new.

05 Understanding project structure

Knowing where frontend, backend and tests live.

06 Technical awareness

Knowing what each layer of the stack does.

07 Git and GitHub workflow

Committing and branching like a real team.

08 Deployment

Taking a product from a laptop to a public URL.

09 Testing and debugging

Tracing an error back to its cause.

10 Communication

Daily updates that kept mentors in the loop.

11 Confidence

Presenting the same product to anyone.

12 Progress discipline

Shipping something small, even during exams.

They worked through all of it together, not in split roles. The hardest part wasn't the code: it was time management during exams.

“Building is the easier part. Ideation and feature selection are harder.”

The resume changed. So did the story.

Instead of a repository holding a college assignment, they now have a deployed product: a defined problem, an MVP, a design system, a backend, AI functionality, testing, and a live production deployment. More importantly, they can **explain how it was built**, not just that it exists.

“If you have an idea, take suggestions and feedback, get some senior guidance, and then build it.”

SRIKAR & PAARTHU, FELLOWS 008 · 009

32

DAYS, IDEA TO DEPLOYED
PRODUCT

11

MVP CAPABILITY AREAS

12

TRANSFERABLE SKILLS
NAMED

[APPLY]

edupodx.com/buildtheory

+91 88860 93609 · WhatsApp

buildtheory@edupodx.com · Instagram @buildtheoryhq

[MENTORS FROM]



Bower
School of Entrepreneurship



Founders First Network



ISB



Poditivity