SOHAM KOTKAR — GURUKUL EXPERIENCE OWNER
Frontend • UX • Stability • Integrity

You now own the face, experience, and emotional truth of Gurukul.

Yashika is stabilizing the backend.
You must ensure that what reaches the student, parent, teacher, and admin feels:
 • disciplined
 • calm
 • meaningful
 • purposeful
 • production-grade
 • human, not mechanical edtech

This system must not "look cool".
It must guide, hold, teach, and sustain learning.

You are responsible for that.

────

PRIMARY OBJECTIVE

Transform Gurukul from:
"Functional UI + Features"
to
"A living learning environment with soul, structure, and emotional intelligence"

Your job:
 • Make the platform usable
 • Make it stable
 • Make it calm

- Make it meaningful
- Make it production ready
- Ensure nothing breaks
- Ensure backend stability issues do not leak to UX
- Align UI to Gurukul philosophy

This is no longer a student demo.

This is an institutional product.

———

YOU WORK WITH

Yashika — Backend Owner
- Provides stable APIs
- Ensures correctness
- Ensures contract discipline

You must:
- consume APIs gracefully
- never break frontend because backend changed
- push back if backend change breaks UX
- close the loop with her daily

Rishabh + Nikhil — EMS / Governance
- Provide Admin / Teacher / Parent structure
- Provide Tenant + Role logic
- Ensure institutional discipline

You must:
- integrate their governance layer
- reflect roles in UI
- ensure visibility discipline
- ensure clarity in user journeys

No improvisation here.
Use their structure.

————

CORE RESPONSIBILITY AREAS
    1. Experience Architecture (Not Just Screens)

Gurukul must feel like a guided journey, not a menu of tools.

Student flow must feel like:
Enter → Learn → Practice → Reflect → Improve

Not:
Click → Feature → Done

You must design UX to:
- lead students
- reduce overwhelm
- encourage continuity
- gently keep them engaged

Nothing flashy.
Everything meaningful.

————

    2. Stability and Production Discipline

You must remove:
- fragile layouts
- random crashes
- UI inconsistencies
- dependency issues
- routing glitches
- rough edges

Frontend must:
- load reliably
- behave consistently
- degrade gracefully when backend/API fails
- never freeze
- never embarrass the system in a demo

You must treat this like a real product.

────

3. Soul Layer Implementation (Student Side)

Ensure UI supports:
- structured learning experience
- visible learning progress
- space for reflection elements when backend adds them
- flashcards + quizzes feel part of learning, not add-ons
- emotional calm tone
- language friendliness (Arabic readiness respected)

No noise.
No chaos.
No distraction.

────

4. EMS + Governance UI Layer

With Rishabh and Nikhil:

Implement UI foundations for:
- Admin view
- Teacher view
- Parent view

You do not build full analytics systems now.

For this sprint:
- role-based entry structure
- clear dashboards
- safe visibility
- basic monitoring scaffolding
- student journey visibility railings

Keep it light.
Keep it disciplined.

Do not invent governance logic.
Use EMS governance structure.

─────

5. Integration Discipline with Backend

With Yashika:
- lock API consumption properly
- handle failures politely
- never require frontend users to refresh or "try again randomly"
- remove any "we need to change API frequently" experience pain

Frontend must:
- expect backend errors
- not collapse
- show graceful fallback states

If backend breaks → UX must still hold.

─────

TIMELINE FRAME

Same sprint as Yashika.
You move in parallel.
Your work must always stay aligned with backend maturity.

No divergence.
No ego builds.
Real product discipline.

DELIVERABLES
1. Stable, calm, unified Gurukul UI
2. Student experience that feels guided, not fragmented
3. EMS-enabled Admin / Teacher / Parent layer foundation
4. Production ready frontend
5. Integration tested with backend
6. Documentation of final UX behavior
7. Demo ready experience
8. No regressions
9. No cosmetic lies
10. Truthful system

NON-NEGOTIABLES

Do not break what exists
Do not overdesign
Do not rush UI shortcuts
Do not ignore soul
Do not ignore governance
Do not ignore stability