FRONTEND & UI/UX — FINAL INTEGRATION SPRINT

Assignee: Chandragupta Maurya
Duration: 7 Days (Hard Push)
Product: AI Architecture & Design Platform
Role: Frontend Lead (Web + UX Execution)

READ THIS FIRST (Mandatory)

- You are responsible only for frontend and user experience.
- Treat backend APIs as black boxes — do not infer or redesign system logic.
- Focus on clean UI, reliability, and integration correctness.
- Do not expose internal architecture, security, or system rationale.
- Outcome must be a demo-ready, production-quality UI.

INTEGRATION CONTACTS

- Anmol Mishra — Backend APIs & Auth
- Ranjit Patil — Bridge / Health / Status endpoints
- Vinayak Tiwari — Task Bank validation & QA

SCOPE (WHAT YOU ARE BUILDING)

- Web UI (React) for full design flow:

prompt → generate → preview → iterate → evaluate → save → history

- GLB/3D preview integration
- Error handling, loading states, responsiveness
- Final UX polish for demo & go-live readiness

DAY-BY-DAY EXECUTION PLAN

Day 1 — Setup & API Wiring

- Set up React project (Vite preferred).
- Connect to backend:
  - /generate
  - /iterate
  - /evaluate
  - /history
- Implement JWT header handling (env-based).
- Basic layout: header, prompt panel, canvas, JSON sidebar.

Deliverable: Prompt → Generate works with real backend.

Day 2 — 3D Preview & Viewer Stability

- Integrate GLB viewer (Three.js / R3F).
- Add:
  - Orbit controls
  - Lighting presets
  - Zoom/reset
- Handle preview failures gracefully (fallback UI).

Deliverable: Stable GLB preview with real data.

Day 3 — Iteration, Comparison & Errors

- Implement:
  - Iterate button
  - Before/After comparison (modal or split view)
- Handle async states (202 / polling if applicable).
- User-friendly error messages for all failure cases.

Deliverable: Iterate → preview → compare works end-to-end.

Day 4 — Evaluation, Feedback & History

- Add:
  - Rating control (stars)
  - Feedback text input
- Wire to /evaluate.
- Implement history view:
  - Load past designs
  - Restore spec + preview.

Deliverable: Evaluation + history reload functional.

Day 5 — UX Polish & Responsiveness

- Improve layout consistency.
- Add:
  - Loading skeletons
  - Disabled states

    - Mobile responsiveness (web).
- Ensure no UI breaks on slow responses.

Deliverable: Demo-quality UX.

Day 6 — QA, Edge Cases & Cleanup

    - Full end-to-end testing:

generate → iterate → evaluate → save → reload.

- Fix race conditions, preview reload bugs.
- Clean console errors and warnings.

Deliverable: Stable UI with zero critical bugs.

Day 7 — Demo & Handover

    - Record a 3–4 minute demo:

full flow on real backend.

- Prepare FRONTEND_HANDOVER.md:
    - Setup
    - Env vars
    - Known limits
- Submit to Vinayak (Task Bank).

Deliverable: Demo video + clean repo.

LEARNING KIT (Use Only If Needed)

React & Architecture

- “React application architecture best practices”
- “Handling async API states in React”

Three.js / R3F

- “React Three Fiber GLTFLoader tutorial”
- “Three.js camera & lighting basics”

UX

- “Designing async user interfaces”
- “Error-first UX patterns”

(Search by keywords if links change.)

———

DELIVERABLES (NON-NEGOTIABLE)

- Working React frontend repo
- Integrated with real backend APIs
- GLB preview working
- Demo video (3–4 min)
- Frontend handover doc

———