

Chandragupta — 7-Day Integration + Learning Sprint (Day 1 → Day 7)

Goal

Turn the current standalone News AI into a production-ready, integrated, secure, and deployed frontend that plugs into Sankalp (Insight Node), Noopur (backend), and Seeya (orchestration). Learn small, focused topics each day and apply them.

Day 1 — Integration Mapping & Quick Syncs

Tasks

1. Pull latest from blackhole-frontend repo and confirm branch.
2. Create an Integration Map (1 page): list endpoints you must consume (Sankalp, Noopur, Seeya) and map each JSON field to UI components.
3. Quick 30-minute sync calls (or Slack pings) to:
 - Sankalp — confirm `audio_path`, `script`, `priority_score`, `trend_score`
 - Noopur — confirm `/news` and `/processed` endpoints and auth method
 - Seeya — confirm pipeline stage fields for visual timeline
4. Add a local `.env.local.example` with required env keys.

Deliverables

- `/docs/integration_map.md`
- Updated `.env.local.example`

Learning Kit

- Read: “How to read API docs” (10–20 min cheat sheet)
- Short: Postman quick start (run one sample request)

Day 2 — Replace Local Scraper with Unified Backend & Data Binding

Tasks

1. Replace use of `scraped-news.json` as primary source with API calls to Noopur/Sankalp:
 - Implement `lib/api.js` wrappers for `GET /news`, `GET /processed/:id`, `GET /audio/:id`.
2. Implement robust async handling: loading states, retries (3x), and clear error messages.

3. Ensure UI cards render script, tone, priority_score, trend_score, and audio_path.

Deliverables

- lib/api.js with wrapper functions
- UI feed shows real processed items (not local cache)

Learning Kit

- Article: “Async/Await and error handling in React”
- Mini exercise: implement an Axios request with retry logic

Day 3 — TTS Audio Playback + Pipeline Visualization

Tasks

1. Wire the audio player to audio_path from Sankalp; support play/pause, show duration.
2. Build the AI pipeline timeline component showing stages: Fetched → Filtered → Summarized → Verified → Scripted → Voiced. Use status or timestamps from JSON.
3. Add small animations/progress indicators for stage transitions.

Deliverables

- Audio player integrated with live audio
- Pipeline timeline component functioning for each item

Learning Kit

- Watch: “Building audio players in React” (10–20 min)
- Read: “Designing progress & status indicators” (UI pattern)

Day 4 — RL Feedback UI & Client Hooks

Tasks

1. Implement feedback buttons (Approve, Like, Skip, Flag) on each feed item.
2. On click, POST to /feedback (Sankalp) with { id, item, signals }. Add optimistic UI update.
3. Store local actions in localStorage as fallback when offline and sync on reconnect.
4. Add lightweight analytics logging (console + local file link or export).

Deliverables

- Feedback UI working, sends payloads to backend

- Local sync fallback implemented

Learning Kit

- Read: “Designing for feedback collection” (UX)
- Short tutorial: “Using localStorage for offline sync”

Day 5 — Basic SSPL-III Security Primitives (JWT + Nonce + Signature)

Tasks

1. Implement client-side JWT handling:
 - Read JWT from .env.local/login stub
 - Attach Authorization: Bearer <token> to API calls
2. Add a nonce header to requests (X-Client-Nonce) generated per request.
3. Implement simple request signing (HMAC) for calls (shared secret in .env.local) — call lib/security.js to compute signature and attach X-Signature.
 - NOTE: This is client-side helper only; backend must validate (coordinate with Noopur/Sankalp).
4. Add error handling for 401/403 and show login/failure UI.

Deliverables

- lib/security.js with getJWT(), signRequest() helpers
- All API calls include JWT, Nonce, Signature headers

Learning Kit

- Read: “What is JWT?” (10–15 min)
- Read: “Nonce & anti-replay basics” (short primer)
- Mini exercise: implement HMAC-SHA256 signing using Web Crypto or small library

Day 6 — Integration Testing, Edge Cases & Accessibility

Tasks

1. Run 20 test items through the UI pipeline, confirm:
 - audio plays
 - feedback posts recorded

- pipeline visuals match timestamps
- 2. Test edge cases:
 - Missing audio_path
 - Malformed JSON fields
 - Backend 5xx responses
 - Network offline mode (localStorage replay)
- 3. Add ARIA labels and keyboard navigation for feedback buttons and player.

Deliverables

- /tests/integration-checklist.md with pass/fail
- Fixes for top 5 bugs found
- Basic accessibility improvements

Learning Kit

- Read: “Basic accessibility for interactive controls”
- Quick: “How to write integration test checklists”

Day 7 — Final polishing, Deploy to Vercel (staging) & Handover

Tasks

1. Create production build and deploy to Vercel staging (set NEXT_PUBLIC_API_BASE to staging backend).
2. Prepare handover docs:
 - docs/frontend-integration.md (how to run, envs, endpoints, security headers)
 - docs/hand_over_checklist.md
3. Record 2-3 minute demo walkthrough showing:
 - Live feed, pipeline stages, audio playback, feedback submission
4. Tag repo release/frontend-v1 and push.

Deliverables

- Live staging URL on Vercel
- Demo video link and handover docs

- Release tag frontend-v1

Learning Kit

- Read: “Deploying Next.js to Vercel” quick guide
- Watch: “How to record a demo (screen + voice)”

Scoring

Each day graded pass/fail by lead. Final pass requires:

- Live staging working (audio + feedback) — mandatory
- JWT + nonce + signature headers present — mandatory
- Integration docs and demo — mandatory

Final Notes

- Work incrementally: commit early, push often, and tag working milestones.
- Coordinate immediately with Noopur & Sankalp on header signing and JWT scheme — backend must accept your headers.
- Keep UI simple and discoverable; small polished interactions beat flashy but broken ones.