

# Momentum – Product & UX Design (Self-Hosted)

**Momentum** is an AI-driven motivation booster app. Users set their goals and describe what holds them back; the AI then generates daily flashcards and prompts to keep them consistent, resilient, and focused through any type of challenge—language learning, fitness, career growth, or personal development.

---

## 1) Product Goal

Help users stay **consistent and emotionally supported** while working toward any goal. Momentum removes mental barriers like perfectionism, fear, or burnout by providing short daily cards that reframe mindset, prompt action, and spark reflection.

**How:** - Capture **Goal + Blockers** once during onboarding. - Deliver **3 cards per day:** - 🧠 **Mindset:** motivational reframing. - 🖐️ **Action:** one simple task to move forward. - 🙌 **Reflection:** self-awareness or gratitude prompt. - All AI runs **locally or on self-hosted backend** (no subscription fees).

---

## 2) Audience & Scenarios

- **Who:** Anyone pursuing a goal who struggles with consistency or self-criticism.
- **Examples:**
  - A student learning English or coding.
  - A runner training for a race.
  - A professional improving communication skills.
  - A creative fighting perfectionism.

**Core Scenarios:** 1. User sets their goal and defines blockers. 2. Each day Momentum delivers 3 short cards personalized to goal + blocker. 3. User marks cards Done / Skip / Reflect. 4. Weekly review of progress, wins, and obstacles.

---

## 3) Design Overview

### Core Screens

1. **Onboarding** – set goal, blockers, reminders, confirm local AI.
2. **Today** – 3 cards (Mindset, Action, Reflection) + quick record option.
3. **Review / SRS** – revisit earlier cards.
4. **Progress** – streaks, consistency heatmap, reflections summary.
5. **Settings** – AI model config, goal reset, privacy info.

## Style

- Clean, calm visuals: blues, neutrals, minimal UI.
  - Encouraging tone: *"You showed up today. That's progress."*
  - Confetti or gentle animations on streak milestones.
- 

## 4) Expo App Implementation

**Stack:** Expo SDK 52 (React Native + TypeScript) - **Navigation:** Expo Router (file-based tabs) - **State:** Zustand + React Query - **Storage:** AsyncStorage (cards, SRS, settings) - **Audio:** expo-av (TTS / playback / recording) - **Notifications:** Expo Notifications (daily reminders) - **Networking:** Fetch via React Query → self-hosted backend - **Theming:** Tailwind RN or StyleSheet tokens

### Directory Sketch:

```
app/  
  (tabs)/today.tsx  review.tsx  reflect.tsx  progress.tsx  settings.tsx  
components/  
  Card.tsx  Recorder.tsx  ProgressRing.tsx  
lib/  
  api.ts  srs.ts  prompts.ts  
store/  
  usePrefs.ts  useAuth.ts
```

### Sample API Call:

```
const { data: cards } = useQuery({  
  queryKey: ['today'],  
  queryFn: async () => (await fetch(`${BASE_URL}/api/cards/today`)).json(),  
});
```

---

## 5) AI Usage (Self-Hosted, No Subscription)

### Options

- **vLLM or TGI backend** with 7B instruct model (e.g., Mistral, Qwen2.5).
- **Ollama** on local CPU/GPU for small setups.
- **Hybrid:** pre-generate next-day cards nightly; cache results.

## Prompt Template

```
SYSTEM: You generate concise motivational flashcards to help users stay consistent.
USER:
Goal: {{goal}}
Blockers: {{blockers}}
Context: {{yesterday_reflection}}
Rules: 3 cards – Mindset, Action, Reflection. Max 40 words each.
Format JSON: [{type, text}]
```

**Generation Controls:** temperature=0.6, max\_tokens small, validate JSON schema, fallback to local templates on failure.

## Example Output

```
[
  {"type": "Mindset", "text": "Progress isn't perfection – showing up counts."},
  {"type": "Action", "text": "Take 5 minutes today to do one small thing toward your goal."},
  {"type": "Reflection", "text": "What helped you stay on track yesterday?"}
]
```

## 6) Backend Architecture (Self-Hosted)

```
[Expo App] → [API Gateway] → [Generation Service] → [vLLM/TGI/Ollama]
🥱 🥱 [Cache (Redis)] [DB (Postgres)]
```

- **API Gateway:** auth, rate limit, logging. - **Generation Service:** prompt templating, JSON validation, card scheduling. - **Model Runtime:** vLLM/TGI (GPU) or Ollama (CPU) for offline. - **Cache:** Redis keyed by (userId, day, goal, blocker). - **DB:** Postgres for users, cards, reflections, metrics.

**API Endpoints:** - POST /api/cards/generate → {goal, blockers} → [Card] - GET /api/cards/today - POST /api/review - POST /api/reflect - GET /api/progress

### Card JSON:

```
{ "id": "uuid", "type": "mindset|action|reflection", "text": "string", "createdAt": "ISO" }
```

## 7) Roadmap (90 days)

- **Month 1:** MVP — onboarding, daily cards, caching, progress view.
  - **Month 2:** Reflections, streaks, SRS review, export.
  - **Month 3:** Adaptive difficulty, custom themes (fitness, study, confidence), TTS audio, offline sync.
- 

## 8) Metrics

- Activation: first day completed.
  - Engagement: 4+ days/week active.
  - Retention: streak growth, skipped-day decline.
  - Sentiment: user notes reflect higher motivation scores.
- 

## 9) Risks & Mitigations

- **Motivational fatigue:** cap at 3 cards/day, allow snooze.
  - **Model drift:** schema validation + cached fallback.
  - **Latency:** pre-generate nightly; quantized models.
  - **Privacy:** self-host default; no external telemetry.
- 

## 10) Visual Identity

- Calm palette, rounded cards, friendly microcopy.
- Emphasis on *consistency and encouragement* rather than performance.
- Key phrase: “Done imperfectly is still done.”