Contents

Think Tank Consumer App - Gap Analysis & Implementation Plan

> **Date**: January 19, 2026
> **Status**: CRITICAL - Consumer app is severely underdeveloped
> **Priority**: URGENT

Executive Summary

The Think Tank consumer app (`apps/thinktank/`) is **critically underdeveloped**. It consists of: - **1 page** (page.tsx - 420 lines) - **2 API files** (chat.ts, client.ts) - **Basic chat only** - no advanced features

Meanwhile, the backend has **30 Lambda handlers** and the admin app has **23+ rich pages** with full implementations.

Gap: ~95% of backend capabilities are NOT exposed to consumers.

Current Consumer App State

What Exists

```
apps/thinktank/
  app/
      globals.css
      layout.tsx
      page.tsx          # Basic chat only
      providers.tsx
  lib/
       api/
           chat.ts       # Conversations only
           client.ts     # Basic API client
       auth/
           api-auth.ts
           context.tsx
```

Features in page.tsx

- Basic conversation list
- Send/receive messages with streaming
- New conversation creation
- Basic sidebar with conversation history
- Everything else missing

Backend Capabilities (Lambda Handlers)

Handler	API Endpoints	Consumer Access
`conversations.ts`	CRUD, messages, streaming	Partial
`users.ts`	Profile, preferences	**MISSING**
`models.ts`	Model listing, selection	**MISSING**
`brain-plan.ts`	AI planning visualization	**MISSING**
`user-context.ts`	Persistent context	**MISSING**
`domain-modes.ts`	Domain detection, modes	**MISSING**
`model-categories.ts`	Model categorization	**MISSING**
`ratings.ts`	Message ratings/feedback	**MISSING**
`grimoire.ts`	Learned heuristics	**MISSING**
`economic-governor.ts`	Cost optimization	**MISSING**
`time-travel.ts`	State snapshots	**MISSING**
`council-of-rivals.ts`	Multi-model consensus	**MISSING**
`concurrent-execution.ts`	Parallel tasks	**MISSING**
`structure-from-chaos.ts`	Document synthesis	**MISSING**
`sentinel-agents.ts`	Background agents	**MISSING**
`flash-facts.ts`	Quick facts/memory	**MISSING**
`liquid-interface.ts`	Dynamic UI	**MISSING**
`reality-engine.ts`	Reality simulation	**MISSING**

Handler	API Endpoints	Consumer Access
`security-signals.ts`	Security indicators	**MISSING**
`policy-framework.ts`	Cato policies	**MISSING**
`derivation-history.ts`	Response derivation	**MISSING**
`enhanced-collaboration.ts`	Real-time collab	**MISSING**
`file-conversion.ts`	File processing	**MISSING**
`ideas.ts`	Idea capture	**MISSING**
`artifact-engine.ts`	Code generation	**MISSING**
`analytics.ts`	Usage analytics	**MISSING**
`settings.ts`	User settings	**MISSING**
`my-rules.ts`	Personal rules	**MISSING**
`shadow-testing.ts`	A/B testing	**MISSING**

Summary: 28 of 30 handlers have NO consumer access

Implementation Plan

Architecture: Auto Mode vs Advanced Mode

```
                    CONSUMER EXPERIENCE



                      AUTO MODE (Default)

     • Clean, simple chat interface
     • Cato handles all decisions automatically
     • Model selection: automatic
     • Domain detection: automatic
     • Cost optimization: automatic
     • Hidden complexity, delightful experience



                    [Advanced Mode Toggle]



                     ADVANCED MODE

     • Model selection visible
     • Brain plan viewer
     • Cost/token display
     • Domain mode selection
     • Governor controls
```

- Analytics dashboard
- Time travel controls
- Concurrent execution panel

Phase 1: Core Infrastructure (Week 1)

1.1 Project Structure Expansion

```
apps/thinktank/
  app/
      (chat)/
          layout.tsx          # Chat layout with sidebar
          page.tsx            # Main chat interface
          [conversationId]/
              page.tsx        # Specific conversation
      settings/
          page.tsx            # User settings
      rules/
          page.tsx            # My Rules
      history/
          page.tsx            # Full history
      artifacts/
          page.tsx            # Generated artifacts
      profile/
          page.tsx            # User profile
  components/
      chat/
          ChatInterface.tsx
          MessageBubble.tsx
          InputArea.tsx
          ConversationList.tsx
          StreamingIndicator.tsx
      advanced/
          AdvancedModeToggle.tsx
          BrainPlanViewer.tsx
          ModelSelector.tsx
          DomainModeIndicator.tsx
          CostDisplay.tsx
          GovernorStatus.tsx
      ui/                     # shadcn/ui components
          [all components]
      layout/
          Sidebar.tsx
          Header.tsx
          MobileNav.tsx
  lib/
      api/
          client.ts
          chat.ts
          settings.ts
          rules.ts
          models.ts
          analytics.ts
          artifacts.ts
          brain-plan.ts
          domain-modes.ts
          governor.ts
      hooks/
          useAdvancedMode.ts
          useConversation.ts
          useModels.ts
          useSettings.ts
          useRules.ts
      stores/
          conversation-store.ts
          settings-store.ts
          ui-store.ts
  styles/
      globals.css
```

1.2 API Client Expansion

- Add all 30 API service modules
- Implement proper TypeScript types
- Add caching with TanStack Query
- Add optimistic updates

1.3 State Management

- Zustand stores for UI state
- TanStack Query for server state
- Persist advanced mode preference

Phase 2: Auto Mode Enhancement (Week 2)

2.1 Enhanced Chat Interface

- Polished message bubbles with markdown
- Code syntax highlighting
- Image/file preview
- Typing indicators with personality
- Smooth animations (Framer Motion)

2.2 Smart Sidebar

- Conversation grouping (Today, Yesterday, Last Week, etc.)
- Search conversations
- Favorites
- Quick actions

2.3 Delight Integration

- Personality messages from Cato
- Achievement notifications
- Easter egg triggers
- Sound effects (optional)

2.4 Auto-Behaviors

- Automatic domain detection (visual indicator)
- Automatic model selection (subtle badge)
- Cost optimization running silently
- Governor decisions invisible

Phase 3: Advanced Mode (Week 3)

3.1 Advanced Mode Toggle

- Toggle button in header
- Smooth transition animations
- Persist preference
- Keyboard shortcut (+Shift+A)

3.2 Brain Plan Viewer

- Show AI's execution plan
- Step-by-step progress
- Model selection reasoning
- Time estimates

3.3 Model Selector

- Model picker dropdown
- Model capabilities display
- Cost comparison
- Performance metrics

3.4 Cost/Governor Panel

- Real-time cost tracking
- Token usage
- Governor decisions
- Savings visualization

3.5 Domain Mode Controls

- Manual domain selection
- Domain confidence display
- Subspecialty drilling
- Model proficiency view

Phase 4: Feature Pages (Week 4)

4.1 Settings Page

- Profile settings
- Notification preferences
- Personality mode selection
- Feature toggles
- Privacy controls
- Data export

4.2 My Rules Page

- View active rules
- Add custom rules
- Browse presets
- Toggle rules on/off
- Rule effectiveness stats

4.3 History Page

- Full conversation history
- Advanced search/filter
- Date range picker
- Export conversations
- Bulk actions

4.4 Artifacts Page

- Generated code/documents
- Preview artifacts
- Download/export
- Version history

4.5 Profile Page

- Account details
- Subscription tier
- Usage statistics
- API keys (if applicable)

Phase 5: Magic Carpet Integration (Week 5)

5.1 Spatial Glass Effects

- Glassmorphism cards
- Depth layers
- Glow effects

5.2 Reality Scrubber

- Timeline navigation
- State snapshots
- Playback controls

5.3 Pre-Cognition Suggestions

- Predictive actions
- Quick suggestions
- Confidence indicators

5.4 AI Presence Indicator

- Emotional state
- Thinking animation
- Confidence display

Phase 6: Polish & Performance (Week 6)

6.1 Performance

- Code splitting
- Lazy loading
- Image optimization
- Service worker

6.2 Accessibility

- Keyboard navigation
- Screen reader support
- Focus management
- ARIA labels

6.3 Mobile Experience

- Responsive design
- Touch gestures
- Mobile-first components
- PWA support

6.4 Error Handling

- Graceful degradation
- Retry mechanisms
- Offline support
- Error boundaries

UI/UX Requirements

Design Principles

1. **Simplicity First**: Auto mode is clean, uncluttered
2. **Progressive Disclosure**: Advanced features revealed on demand
3. **Delightful**: Personality, animations, surprises
4. **Fast**: Instant feedback, optimistic updates
5. **Accessible**: Works for everyone

Key Interactions

Action	Auto Mode	Advanced Mode
Send message	Just type and send	Same + model selection
New conversation	One click	Same + domain preset
View history	Basic list	Full history page
Settings	Minimal	Full settings page
See costs	Hidden	Real-time display
Model selection	Cato decides	User can override

Visual Language

- **Colors**: Purple gradient (primary), Dark mode default
- **Typography**: Inter (body), JetBrains Mono (code)
- **Animations**: Subtle, purposeful, 200-300ms
- **Icons**: Lucide icons, consistent sizing
- **Spacing**: 8px grid system

API Integration Priority

Tier 1 - Must Have (Week 1-2)

1. `conversations.ts` - Full integration
2. `settings.ts` - User preferences
3. `my-rules.ts` - Personal rules
4. `models.ts` - Model listing
5. `domain-modes.ts` - Domain detection

Tier 2 - Should Have (Week 3-4)

6. `brain-plan.ts` - Plan visualization
7. `economic-governor.ts` - Cost tracking
8. `analytics.ts` - Usage stats
9. `ratings.ts` - Message feedback
10. `artifact-engine.ts` - Artifacts

Tier 3 - Nice to Have (Week 5-6)

11. `time-travel.ts` - State snapshots
12. `concurrent-execution.ts` - Parallel tasks
13. `enhanced-collaboration.ts` - Collab features
14. `flash-facts.ts` - Quick memory
15. Remaining handlers

Success Metrics

Metric	Target
Time to first message	< 1 second
Message send latency	< 500ms perceived
Page load time	< 2 seconds
Lighthouse score	> 90
User satisfaction	> 4.5/5

Risk Assessment

Risk	Mitigation
Scope creep	Strict phase boundaries
Performance issues	Incremental optimization
API inconsistencies	Strong typing, error handling
Mobile experience	Mobile-first development

Immediate Next Steps

1. **Set up project structure** - Create all directories and base files
2. **Install dependencies** - Add required packages
3. **Create UI components** - shadcn/ui setup
4. **Implement API services** - All 30 handlers
5. **Build Auto Mode chat** - Enhanced chat interface
6. **Add Advanced Mode toggle** - Feature flag system
7. **Implement settings** - User preferences
8. **Add My Rules** - Personal customization
9. **Polish and test** - QA everything
10. **Deploy** - Production release

This document is the source of truth for Think Tank consumer app implementation.