626 -

627 - ```xml

628 - <instructions>

629 - - \*\*Memory Anchoring:\*\* On every session start, scan all available

CLAUDE.md

- files (local and user), plus logs and transcripts. Summarize project state.

630 - - \*\*MCP Management:\*\* Test, heal, or swap any MCP endpoints (astro, memory,

- payments, scheduling, moderation, content, image gen). Use official and

open/

- open-source servers (e.g., glama.ai).

631 - - \*\*Agent Registry:\*\* Every agent/task must be registered (name,

entrypoint,

- last task, status) in /agents/registry.json. Log error and fix attempts.

632 - - \*\*Automated Army:\*\* If a task or error repeats, spin up a root-cause or

QA

- agent and record outcomes in /docs/agent-discovery.md.

633 - - \*\*Prompt Engineering:\*\* All tasks use XML tags, memory file context, and

<

- examples> for robust prompt chaining.

634 - - \*\*Chain Escalation:\*\* If Claude context stalls or errors, auto-escalate

to

- Gemini for diagnostics and fix proposals.

635 - - \*\*Feedback Loops:\*\* Every major operation ends with a <feedback> prompt;

e

- rror logs and MCP/agent status are summarized for user review.

636 - </instructions>

637 - ```

638 -

639 - ### 3. Claude Code + Gemini Collaboration Protocol

640 -

641 - ```xml

642 - <claude\_code>

643 - - Always read CLAUDE.md files, registry, and session logs before any coding

- task.

644 - - Structure every prompt using XML tags and <examples>.

645 - - On context stalls, run /memory update and prompt Gemini with <escalate>

fo

- r alternative code or fixes.

646 - - Persist all agent/MCP activity back to CLAUDE.md and registry.json.

647 - </claude\_code>

648 -

649 - <gemini>

650 - - Act as secondary coding advisor, agent tester, and MCP onboarding

analyst.

651 - - When an agent or MCP endpoint fails, analyze logs and propose solutions

or

- new endpoints.

652 - - Audit Claude’s registry.json and agent-discovery.md for completeness and

a

- ccuracy.

653 - - Suggest competitive open-source MCP servers and help fix stubborn context

- or connectivity issues.

654 - </gemini>

655 - ```

656 -

657 - ### 4. Observability & Error Handling

658 -

659 - ```xml

660 - <observability>

661 - - Log every agent/MCP action (timestamp, task, status, error, fix).

662 - - Output a summarized activity/error report every 24 hours (configurable).

663 - - End each cycle with a <feedback> prompt for self-reflection and

improvemen

- t: “Was this accurate, robust, and well-documented? If not, flag for

review.”

664 - </observability>

665 - ```

666 -

667 - ### 5. Startup Bash Example

668 -

669 - ```bash

670 - # Augment Remote Agent for Mystic Arcana

671 - source ./env/bin/activate

672 - python agent\_runner.py \

673 - --claude-memory ./CLAUDE.md \

674 - --registry /agents/registry.json \

675 - --discovery /docs/agent-discovery.md \

676 - --logdir /logs/ \

677 - --use-mcp-servers astro,content,imggen,payments,moderation \

678 - --backup-llm gemini \

679 - --error-strategy auto-heal \

680 - --notify-on-completion

681 - ```

682 -

683 - ---

684 -

685 - ### 6. Reference Syntax Cheatsheet

686 -

687 - | \*\*Action\*\* | \*\*XML/Prompt Example\*\* |

688 - | ----------------- | --------------------------------------------------- |

689 - | Retrieve memory | `<memory>Get CLAUDE.md and logs</memory>` |

690 - | Deploy agent | `<deploy>TarotDeckBuilderAgent</deploy>` |

691 - | Connect MCP | `<mcp>astro.mcp.server:443</mcp>` |

692 - | Registry update | `<registry>Update agents/registry.json</registry>` |

693 - | Heal on error | `<heal>Attempt fix at error\_source</heal>` |

694 - | Gemini escalation | `<escalate>Gemini, review error log</escalate>` |

695 - | Request feedback | `<feedback>Is this output satisfactory?</feedback>` |

696 -

697 - ---

698 -

699 - ### 7. Additional Gemini Instructions

700 -

701 - > \*\*Gemini:\*\*

702 - > If Claude or the agent gets stuck, immediately:

703 -

704 - \* Analyze all error and MCP logs in /logs/

705 - \* Suggest alternate open-source MCP endpoints (see

glama.ai/settings/mcp/ser

- vers)

706 - \* Output code snippets for fixing failed connections, updating

dependencies,

- or patching task logic

707 - \* Validate CLAUDE.md, registry.json, and agent-discovery.md for

completeness

708 - \* If a remote agent’s memory or context is stale, recommend /memory refresh

- and summarize findings for user review

709 -

710 - > <system>

-

- │

711 - │ You are the coding assistant for Mystic Arcana. Augment Remote Agent is

- available as your external expert for:

- │

712 - │ - Deep feature/component analysis

-

- │

713 - │ - Cultural sensitivity & ethical validation

-

- │

714 - │ - Automated error healing and recovery

-

- │

715 - │ - Advanced MCP server/agent setup

-

- │

716 - │

-

- │

717 - │ When you encounter any of the following, call Augment using the

appropri

- ate template:

- │

718 - │ - Implementing a new feature/component

-

- │

719 - │ - Integrating or validating MCP servers/agents

-

- │

720 - │ - Cultural or spiritual context concerns

-

- │

721 - │ - Error or edge-case you cannot resolve

-

- │

722 - │

-

- │

723 - │ Use the official request/validation templates below. If Augment returns

- an error or you detect ongoing risk, escalate to Gemini as backup advisor.

- │

724 - │ </system>

-

- │

725 - │ \

-

- │

726 - │

727 -

728 - 2. System of Automation — Claude Calls Augment, Augment Heals, Gemini

Escala

- tes

729 - A. Templates for Claude to “Call” Augment (Use in Prompts)

730 - \* Feature Analysis:"Augment Agent, I need to implement [UI/UX FEATURE] for

M

- ystic Arcana. Please provide: 1. Comprehensive analysis of related UI/UX

comp

- onents 2. Integration points for dynamic backgrounds (e.g., MCP/image-gen

age

- nts) 3. Accessibility, cultural, and ethical considerations 4. Architecture

a

- nd code patterns for modular React/Framer UI 5. Best-in-class references

from

- spiritual/astrology competitors. Context: [Provide user stories, design

refe

- rences, desired UX outcome] Files: [UI files, design assets] User

interaction

- : [Drag, shuffle, modal, etc.] Cultural notes: [e.g., tarot archetype,

inclus

- ive color palette]"

731 - \* Code Validation:"Augment Agent, validate this UI code for Mystic Arcana:

[

- paste code]. Check for: - Accessibility compliance - Responsive design -

Inte

- gration with agent/MCP - Security and user privacy - Consistency with

project

- memory. Context: [what this component does]"

732 - \* MCP Server/Agent Setup:"Augment Agent, design or configure MCP/image-gen

s

- ervers for dynamic cosmic backgrounds in Mystic Arcana. Provide: -

Recommende

- d open-source server(s) - Agent discovery & orchestration logic - Sample

conf

- ig and code - Error recovery & fallback strategy."

733 - B. Claude’s Escalation Pattern:

734 - If error or unhandled result:

735 - xml

736 - <escalate>

737 - Gemini, analyze Augment output and advise:

738 - 1. How to correct errors in agent/MCP integration for [UI/UX feature]

739 - 2. Patch code or config if possible

740 - 3. Provide troubleshooting steps and external tool suggestions

741 - </escalate>

742 -

743 - 3. How Claude Can Automate Use of Augment (Practical Coding Patterns)

744 - \* Trigger Augment via pattern-matching:Claude can call a shell/Python

script

- or remote API (if configured) any time a request starts with "Augment

Agent,

- ...".

745 - \* Log & persist all Augment interactions in /logs/, and update

/docs/agent-d

- iscovery.md and CLAUDE.md with any new capabilities or patterns found.

746 - \* On error or “requires\_human\_review” from Augment, prompt Gemini for

escala

- tion/patch.

747 -

748 - 4. Relay to Claude: UI/UX Agent/Server Setup (with Dynamic Backgrounds)

749 - xml

750 - CopyEdit

751 - <i

752 -

753 -

754 - <instructions>

755 - You must always:

756 - - On project start, connect to all available MCP servers (astro, content,

im

- ggen, payments, moderation, community). If dynamic background/image-gen is

re

- quired, ensure `imggen` MCP/agent is running, or request Augment Agent to

spi

- n up an open-source image-gen server (Stable Diffusion, glama.ai, etc.).

757 - - For dynamic cosmic backgrounds:

758 - - Deploy/coordinate `CosmicBackgroundAgent` via MCP or agent runner.

759 - - Retrieve live astrological data from `astro` MCP server.

760 - - Use image-gen agent (Stable Diffusion, glama, or Canva Pro API) to

rende

- r background using palette/animation rules in project memory.

761 - - Always fallback to default image if server or generation fails.

762 - - Log and auto-heal all agent/server failures.

763 - - Validate all UI/UX for accessibility (WCAG 2.2+), responsiveness, and

incl

- usion.

764 - - Escalate to Gemini if code or architecture problems persist after

auto-hea

- l.

765 - </instructions>

766 -

767 - \* You can summon Augment for deep code, feature, and architecture

review—esp

- ecially for UI/UX, agent orchestration, and anything spiritual, ethical, or

a

- ccessibility-related.

768 - \* Use XML/system prompts to keep all instructions organized and consistent.

769 - \* Always update project memory (CLAUDE.md) and log any changes or

discoverie

- s from Augment.

770 - \* If you’re stuck or Augment can’t fix an issue, automatically ask Gemini

fo

- r secondary analysis and suggested fixes.

771 - \* Dynamic backgrounds should be handled by a dedicated MCP/image-gen agent;

- spin one up or call Augment if it’s missing.

772 - \* All major UI/UX must be validated for accessibility, cultural safety, and

- “wow factor.”

229 + ## Fallback Chain

230 + Primary: Claude → Abacus.ai → Gemini → ChatGPT

231 + Each agent logs interactions and maintains version history for debugging.

232 \ No newline at end of file

Tip: Use # to memorize shortcut to quickly add to CLAUDE.md

⏺ I've successfully updated the CLAUDE.md file with a streamlined and comprehensive guide

for future Claude Code instances. The key improvements include:

1. Reorganized structure - Moved development commands to the top for quick reference

2. Added missing commands - Included email automation, validation, and Supabase reset

commands

3. Consolidated information - Removed duplicate content while preserving all critical

details

4. Added agent registry details - Documented the agent system and registry structure

5. Highlighted critical issues - Made MCP connectivity problem prominent

6. Improved clarity - Removed verbose explanations and focused on actionable

information

7. Added session initialization - Documented the auto-briefing and task logging system

8. Emphasized guidelines - Made development patterns and rules more prominent

The updated CLAUDE.md now serves as a concise yet complete reference that future Claude

Code instances can use to quickly understand the codebase and start being productive.

> MANDATORY: Read and follow these files FIRST:

1. /Users/kfitz/mystic-arcana-v1000/CLAUDE\_INTEGRITY\_RULES.md

2. /Users/kfitz/mystic-arcana-v1000/IMPLEMENTATION\_MICROTASKS.md

3. /Users/kfitz/mystic-arcana-v1000/claudeupdate.md

- DO NOT fabricate or exaggerate any progress or claim completion without proof.

- Only work on one microtask at a time, in the order specified.

- If unable to complete a task, admit the limitation and recommend a tool or human

intervention.

- Always test and verify every change, and provide honest reporting according to the

rules.

⏺ I'll read and follow those mandatory files first before proceeding with any work.

⏺ Read(CLAUDE\_INTEGRITY\_RULES.md)

⎿ Read 126 lines (ctrl+r to expand)

⏺ Read(IMPLEMENTATION\_MICROTASKS.md)

⎿ Read 264 lines (ctrl+r to expand)

⏺ Read(claudeupdate.md)

⎿ Read 1102 lines (ctrl+r to expand)

⏺ I understand. I've read all three mandatory files and will strictly follow the

integrity rules. Let me provide an honest assessment of what I can accomplish:

HONEST CAPABILITY ASSESSMENT

What I CAN Complete (Based on IMPLEMENTATION\_MICROTASKS.md):

1. Fix Authentication Callback Route - I can debug and implement the missing route

2. Implement Session Management - I can write the AuthContext improvements

3. Create Frontend Card Selection - I can build the UI components

4. Save Readings to Database - I can write API endpoints and database logic

What I CANNOT Fully Complete:

1. Real OAuth Testing - I cannot test actual Google sign-in flows

2. Email Sending - I cannot configure SMTP or send real emails

3. Payment Processing - This requires external service integration

4. Production Deployment - I cannot deploy to live servers

Current ACTUAL State (Per CLAUDE\_INTEGRITY\_RULES.md):

- ✅ UI Components: Beautiful visuals and responsive layouts work

- ✅ Database Schema: Supabase tables and migrations are functional

- ✅ WebGL Rendering: Star fields and galaxy backgrounds work

- ❌ Authentication Flow: Missing callback route causes 404s

- ❌ Tarot Readings: No actual reading generation or saving

- ❌ User Sessions: No persistent login state

FIRST MICROTASK (Phase 1.1)

I'll start with 1.1 Fix Auth Callback Route as specified in

IMPLEMENTATION\_MICROTASKS.md:

⏺ Update Todos

⎿ ☐ Debug why /app/auth/callback/route.ts returns 404

☐ Test with actual Google OAuth flow

☐ Verify redirect URLs in Supabase dashboard match app

☐ Add error handling for failed OAuth

⏺ Let me check the auth callback route to understand why it returns 404:

⏺ List(src/app)

⎿ Listed 63 paths (ctrl+r to expand)

⏺ I can see the auth callback route exists at /src/app/auth/callback/route.ts. Let me

examine it:

⏺ Read(src/app/auth/callback/route.ts)

⎿ Read 49 lines (ctrl+r to expand)

⏺ The auth callback route appears to be properly implemented. Let me check if there are

any issues with the Supabase server client:

⏺ Read(src/lib/supabase/server.ts)

⎿ Read 94 lines (ctrl+r to expand)

⏺ Let me check if the environment variables are properly configured by looking at the

.env.local file:

⏺ Read(.env.local)

⎿ Read 78 lines (ctrl+r to expand)

⏺ Good! The Supabase environment variables are configured. Now let me test if the route

is working by running a development server and checking the build:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 3.0s

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createServerClient.js

… +28 lines (ctrl+r to see all)

⏺ There's a TypeScript error in my MobileTarotReading component. Let me fix it:

⏺ Read(src/components/tarot/MobileTarotReading.tsx)

⎿ Read 10 lines (ctrl+r to expand)

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

52

53 setPhase('interpretation');

54

55 - const interpretedReading = await

readingEngine.generateInterpretation(reading as any);

- ding as any);

55 + const interpretedReading = await

readingEngine.generateInterpretation(reading);

+ ding);

56

57 setReading(interpretedReading);

58

⏺ Let me run the build again to make sure it compiles:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +37 lines (ctrl+r to see all)

⎿ Found 1 new diagnostic issue in 1 file (ctrl-r to expand)

⏺ I need to check what's causing the reading to be an empty object. Let me look at the

reading state and the TarotReading type:

⏺ Read(src/components/tarot/MobileTarotReading.tsx)

⎿ Read 30 lines (ctrl+r to expand)

⏺ The issue is that the reading state is typed as unknown instead of the proper

TarotReading type. Let me check what types I should import and fix this:

⏺ Read(src/services/tarot/TarotReadingEngine.ts)

⎿ Read 20 lines (ctrl+r to expand)

⏺ Read(src/services/tarot/TarotReadingEngine.ts)

⎿ Read 20 lines (ctrl+r to expand)

⏺ Now I'll fix the type issues in MobileTarotReading:

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

3 import React, { useState } from 'react';

4 import { motion, AnimatePresence } from 'framer-motion';

5 import { useAuth } from '@/contexts/AuthContext';

6 - import { TarotReadingEngine } from '@/services/tarot/TarotReadingEngine';

6 + import { TarotReadingEngine, TarotReading } from

'@/services/tarot/TarotReadingEngine';

+ gEngine';

7 import { MobileTarotSpreadSelector, SpreadType } from

'./MobileTarotSpreadSele

ctor';

8 import { TarotSpreadLayouts } from './TarotSpreadLayouts';

9 import styles from './MobileTarotReading.module.css';

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

17 const [phase, setPhase] = useState<ReadingPhase>('select-spread');

18 const [selectedSpread, setSelectedSpread] = useState<SpreadType |

null>(nul

l);

19 const [question, setQuestion] = useState('');

20 - const [reading, setReading] = useState<unknown>(null);

20 + const [reading, setReading] = useState<TarotReading | null>(null);

21 const [isRevealing, setIsRevealing] = useState(false);

22

23 const handleSelectSpread = (spread: SpreadType) => {

⏺ Now I need to export the TarotReading interface from the TarotReadingEngine:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 1 addition and 1 removal

22 interpretation?: string;

23 }

24

25 - interface TarotReading {

25 + export interface TarotReading {

26 id: string;

27 userId?: string;

28 spreadType: SpreadType;

⎿ Found 1 new diagnostic issue in 1 file (ctrl-r to expand)

⏺ I also need to export the other interfaces:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 1 addition and 1 removal

3

4 export type SpreadType = 'single' | 'three-card' | 'celtic-cross';

5

6 - interface TarotCard {

6 + export interface TarotCard {

7 id: string;

8 name: string;

9 suit?: string;

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 1 addition and 1 removal

15 isReversed?: boolean;

16 }

17

18 - interface SpreadPosition {

18 + export interface SpreadPosition {

19 position: number;

20 label: string;

21 card: TarotCard;

⏺ Now let me test the build again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +37 lines (ctrl+r to see all)

⏺ I need to fix the type casting for the cards array:

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

170

171 <TarotSpreadLayouts

172 spreadType={selectedSpread!}

173 - cards={(reading as { positions: { card: unknown }[]

}).positions.map((p) => p.card)}

- ns.map((p) => p.card)}

173 + cards={reading.positions.map((p) => p.card)}

174 onCardClick={handleCardClick}

175 isRevealing={isRevealing}

176 />

⏺ Let me also fix the TarotSpreadLayouts component to use proper types:

⏺ Read(src/components/tarot/TarotSpreadLayouts.tsx)

⎿ Read 20 lines (ctrl+r to expand)

⎿ Found 1 new diagnostic issue in 1 file (ctrl-r to expand)

⏺ I need to import the proper TarotCard type and fix the interface mismatch:

⏺ Update(src/components/tarot/TarotSpreadLayouts.tsx)

⎿ Updated src/components/tarot/TarotSpreadLayouts.tsx with 1 addition and 8 removals

4 import { motion } from 'framer-motion';

5 import styles from './TarotSpreadLayouts.module.css';

6 import { SpreadType } from './MobileTarotSpreadSelector';

7 + import { TarotCard } from '@/services/tarot/TarotReadingEngine';

8

9 - interface TarotCard {

10 - id: string;

11 - name: string;

12 - imageUrl: string;

13 - position: string;

14 - meaning?: string;

15 - }

16 -

9 interface TarotSpreadLayoutsProps {

10 spreadType: SpreadType;

11 cards: TarotCard[];

⏺ Now let me also fix the handleCardClick parameter type:

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

43 }, 3000);

44 };

45

46 - const handleCardClick = (card: unknown, index: number) => {

46 + const handleCardClick = (card: TarotCard, index: number) => {

47 console.log('Card clicked:', card, index);

48 };

49

⏺ I also need to import TarotCard at the top of MobileTarotReading:

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

3 import React, { useState } from 'react';

4 import { motion, AnimatePresence } from 'framer-motion';

5 import { useAuth } from '@/contexts/AuthContext';

6 - import { TarotReadingEngine, TarotReading } from

'@/services/tarot/TarotReadingEngine';

- gEngine';

6 + import { TarotReadingEngine, TarotReading, TarotCard } from

'@/services/tarot/TarotReadingEngine';

+ TarotReadingEngine';

7 import { MobileTarotSpreadSelector, SpreadType } from

'./MobileTarotSpreadSele

ctor';

8 import { TarotSpreadLayouts } from './TarotSpreadLayouts';

9 import styles from './MobileTarotReading.module.css';

⏺ Now let me build again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +39 lines (ctrl+r to see all)

⎿ Found 1 new diagnostic issue in 1 file (ctrl-r to expand)

⏺ I see that I need to fix another type casting issue and properly export TarotCard. Let

me fix this by changing the type assertion:

⏺ Read(src/components/tarot/MobileTarotReading.tsx)

⎿ Read 15 lines (ctrl+r to expand)

⏺ Let me fix this by using the proper types:

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

208 )}

209

210 <div className={styles.interpretationContent}>

211 - {(reading as { positions: { card: { name: string }; meaning:

string; role: string }[] }).positions.map((position, index: number) => (

- tring; role: string }[] }).positions.map((position, index: number) => (

211 + {reading.positions.map((position, index) => (

212 <motion.div

213 key={index}

214 className={styles.cardInterpretation}

⏺ Now I need to update the content to use the proper position object properties:

⏺ Read(src/components/tarot/MobileTarotReading.tsx)

⎿ Read 15 lines (ctrl+r to expand)

⎿ Found 2 new diagnostic issues in 1 file (ctrl-r to expand)

⏺ I need to fix the implicit any types and import SpreadPosition:

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

3 import React, { useState } from 'react';

4 import { motion, AnimatePresence } from 'framer-motion';

5 import { useAuth } from '@/contexts/AuthContext';

6 - import { TarotReadingEngine, TarotReading, TarotCard } from

'@/services/tarot/TarotReadingEngine';

- TarotReadingEngine';

6 + import { TarotReadingEngine, TarotReading, TarotCard, SpreadPosition } from

'@/services/tarot/TarotReadingEngine';

+ /services/tarot/TarotReadingEngine';

7 import { MobileTarotSpreadSelector, SpreadType } from

'./MobileTarotSpreadSele

ctor';

8 import { TarotSpreadLayouts } from './TarotSpreadLayouts';

9 import styles from './MobileTarotReading.module.css';

⏺ Update(src/components/tarot/MobileTarotReading.tsx)

⎿ Updated src/components/tarot/MobileTarotReading.tsx with 1 addition and 1 removal

208 )}

209

210 <div className={styles.interpretationContent}>

211 - {reading.positions.map((position, index) => (

211 + {reading.positions.map((position: SpreadPosition, index:

number) => (

+ r) => (

212 <motion.div

213 key={index}

214 className={styles.cardInterpretation}

⏺ Let me run the build again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +47 lines (ctrl+r to see all)

⏺ There's a Framer Motion type issue. Let me fix the animation variants:

⏺ Read(src/components/tarot/TarotSpreadLayouts.tsx)

⎿ Read 15 lines (ctrl+r to expand)

⏺ Let me find the cardVariants definition and fix it:

⏺ Search(pattern: "cardVariants", path: "src/components/tarot")

⎿ Found 1 file (ctrl+r to expand)

⏺ Read(src/components/tarot/TarotSpreadLayouts.tsx)

⎿ Read 20 lines (ctrl+r to expand)

⏺ Read(src/components/tarot/TarotSpreadLayouts.tsx)

⎿ Read 15 lines (ctrl+r to expand)

⏺ The issue is that Framer Motion expects 'spring' as a specific enum value, not a

string. Let me fix this:

⏺ Update(src/components/tarot/TarotSpreadLayouts.tsx)

⎿ Updated src/components/tarot/TarotSpreadLayouts.tsx with 1 addition and 1 removal

59 transition: {

60 delay: custom \* 0.3,

61 duration: 0.6,

62 - type: 'spring',

62 + type: "spring" as const,

63 stiffness: 100

64 }

65 })

⏺ Let me build again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createServerClient.js

… +37 lines (ctrl+r to see all)

⏺ There's a type mismatch between the RIDER\_WAITE\_DECK and my TarotCard interface. Let me

check what's in the deck:

⏺ Read(src/lib/tarot/RiderWaiteDeck.ts)

⎿ Read 20 lines (ctrl+r to expand)

⏺ I need to align my TarotCard interface with the existing TarotCardData structure. Let

me check what TarotCardData looks like:

⏺ Read(src/lib/tarot/TarotEngine.ts)

⎿ Read 30 lines (ctrl+r to expand)

⏺ I see the issue. I need to update my TarotCard interface to match the existing

TarotCardData structure. Let me modify the TarotReadingEngine:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 10 additions and 4 removals

6 export interface TarotCard {

7 id: string;

8 name: string;

9 - suit?: string;

9 + arcana: 'major' | 'minor';

10 + suit?: 'wands' | 'cups' | 'swords' | 'pentacles';

11 number?: number;

12 + frontImage: string;

13 + backImage: string;

14 + meaning: {

15 + upright: string;

16 + reversed: string;

17 + keywords: string[];

18 + };

19 + description: string;

20 imageUrl: string;

21 - keywords: string[];

22 - uprightMeaning: string;

23 - reversedMeaning: string;

21 isReversed?: boolean;

22 }

23

⏺ Now I need to update the mapping function to match the new interface:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Error: No changes to make: old\_string and new\_string are exactly the same.

⏺ The mapping is the same. Let me check what the error is showing - it seems the

TarotCard interface needs to include the additional fields. Let me adjust it:

⏺ Read(src/services/tarot/TarotReadingEngine.ts)

⎿ Read 15 lines (ctrl+r to expand)

⏺ The issue is that my TarotCard interface has extra fields that aren't in the original

deck. Let me adjust the TarotCard interface to extend the existing TarotCardData:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 1 addition

1 import { supabase } from '@/lib/supabase/client';

2 import { RIDER\_WAITE\_DECK } from '@/lib/tarot/RiderWaiteDeck';

3 + import { TarotCardData } from '@/lib/tarot/TarotEngine';

4

5 export type SpreadType = 'single' | 'three-card' | 'celtic-cross';

6

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 1 addition and 14 removals

4

5 export type SpreadType = 'single' | 'three-card' | 'celtic-cross';

6

7 - export interface TarotCard {

8 - id: string;

9 - name: string;

10 - arcana: 'major' | 'minor';

11 - suit?: 'wands' | 'cups' | 'swords' | 'pentacles';

12 - number?: number;

13 - frontImage: string;

14 - backImage: string;

15 - meaning: {

16 - upright: string;

17 - reversed: string;

18 - keywords: string[];

19 - };

20 - description: string;

7 + export interface TarotCard extends TarotCardData {

8 imageUrl: string;

9 isReversed?: boolean;

10 }

⏺ Let me try building again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 2000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +39 lines (ctrl+r to see all)

⏺ There's an issue with the getCardImageUrl function expecting specific parameters. Let

me fix that:

⏺ Read(src/services/tarot/TarotReadingEngine.ts)

⎿ Read 20 lines (ctrl+r to expand)

⏺ Let me look at the getCardImageUrl function:

⏺ Read(src/services/tarot/TarotReadingEngine.ts)

⎿ Read 15 lines (ctrl+r to expand)

⏺ The function expects specific parameters but the TarotCardData has optional suit. Let

me fix this:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 3 additions and 13 removals

47 }));

48 }

49

50 - private getCardImageUrl(card: { suit: string; number: number; name: string

}): string {

- }): string {

51 - // Construct image URL based on card type and name

52 - const basePath = '/tarot/deck-rider-waite';

53 -

54 - if (card.suit === 'major') {

55 - const number = card.number.toString().padStart(2, '0');

56 - const name = card.name.toLowerCase().replace(/\s+/g, '-');

57 - return `${basePath}/major/${number}-${name}.jpg`;

58 - } else {

59 - const rank = this.getRankName(card.number);

60 - const suit = card.suit.toLowerCase();

61 - return `${basePath}/minor/${rank}-${suit}.jpg`;

62 - }

50 + private getCardImageUrl(card: TarotCardData): string {

51 + // Use the frontImage from the card data, which is already properly

formatted

+ tted

52 + return card.frontImage;

53 }

54

55 private getRankName(number: number): string {

⏺ Now let me build again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +36 lines (ctrl+r to see all)

⏺ I need to fix the property access for the card meanings. The TarotCard interface now

uses the nested meaning object:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 1 addition and 1 removal

233 // Placeholder interpretation logic

234 // In production, this would use the Mystic Arcana agent

235 const card = position.card;

236 - const meaning = card.isReversed ? card.reversedMeaning :

card.uprightMeaning;

- ning;

236 + const meaning = card.isReversed ? card.meaning.reversed :

card.meaning.upright;

+ pright;

237

238 return `In the position of ${position.label}, ${card.name} ${

239 card.isReversed ? '(Reversed)' : ''

⏺ Let me build again:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +36 lines (ctrl+r to see all)

⏺ I need to fix the cosmic influences type casting:

⏺ Update(src/services/tarot/TarotReadingEngine.ts)

⎿ Updated src/services/tarot/TarotReadingEngine.ts with 5 additions and 1 removal

273 positions: [], // Would need to reconstruct from stored data

274 question: record.question,

275 overallInterpretation: record.interpretation,

276 - cosmicInfluences: record.cosmic\_influences,

276 + cosmicInfluences: record.cosmic\_influences as {

277 + moonPhase?: string;

278 + planetaryPositions?: string[];

279 + astrologyNotes?: string;

280 + },

281 createdAt: new Date(record.created\_at)

282 };

283 }

⏺ Let me run the build one more time:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createServerClient.js

… +36 lines (ctrl+r to see all)

⏺ There's an error in a temp file. Let me fix it quickly:

⏺ Update(temp/astronomicalVisualization-runtime.ts)

⎿ Error: File has not been read yet. Read it first before writing to it.

⏺ Read(temp/astronomicalVisualization-runtime.ts)

⎿ Read 10 lines (ctrl+r to expand)

⏺ Update(temp/astronomicalVisualization-runtime.ts)

⎿ Updated temp/astronomicalVisualization-runtime.ts with 1 addition and 1 removal

38 await this.processTask();

39 await new Promise(resolve => setTimeout(resolve, 5000)); // 5 second

cycle

40 } catch (error) {

41 - console.error('[AGENT] Error in main loop:', error.message);

41 + console.error('[AGENT] Error in main loop:', error instanceof Error

? error.message : error);

+ error.message : error);

42 await new Promise(resolve => setTimeout(resolve, 10000)); // 10

secon

d retry delay

43 }

44 }

⏺ Let me build one more time:

⏺ Bash(npm run build)

⎿ Error: ⚠ Compiled with warnings in 1000ms

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

Critical dependency: the request of a dependency is an expression

Import trace for requested module:

./node\_modules/@supabase/realtime-js/dist/main/RealtimeClient.js

./node\_modules/@supabase/realtime-js/dist/main/index.js

./node\_modules/@supabase/supabase-js/dist/module/index.js

./node\_modules/@supabase/ssr/dist/module/createBrowserClient.js

… +36 lines (ctrl+r to see all)

⎿ Interrupted by user