`Let's Write An Email - Technical Implementation Guide

Architecture Overview

Frontend Stack

- **Framework**: Next.js with React and TypeScript
- **Styling**: Tailwind CSS for utility classes
- **Game Engine**: Plain JavaScript (intentionally janky) OR Typescript OR Unity
- **Terminal UI**: Custom components with retro styling
- **Sadcoin.net:** Raw HTML with minimal CSS.
    - /buy page to purchase Sadcoin
    - /games page to link to letswritean.email game

Backend Services

- **Email Generation**: Multiple LLM integration via AWS Bedrock
    - Officer: Claude/GPT-4 (smart model)
    - Agent: Mid-tier model
    - Monkey: Cheapest/dumbest model available
    - Intern: Balanced model / no model (user plays as Intern)
- **Email Delivery**: Eliza agent integration
- **State Management**: React state (no persistence needed for MVP)

Web3 Integration (Post-MVP but Pre-Hackathon Deadline)

- **Wallet Connect**: Multiple chain support
- **Chainlink Functions**: Email reply → NFT minting
- **Smart Contracts with Chainlink Pricefeeds**: SADCoin and FEELS tokens

---

Component Breakdown

1. Terminal Container (`/components/Terminal`)

```
interface TerminalProps {
  aspectRatio: "4:3"  // Core viewport
```

```
  backgroundColor: "#000000"
  textColor: "#00FF00"
  borderStyle: "retro"
}
```

Key features:

- Fixed 4:3 aspect ratio that scales
- Green monospace text
- Retro border styling
- Scrollable text areas where needed

2. Game State Manager (`/components/GameStateManager`)

States to track:

- `BOOT_SCREEN`
- `WALLET_CONNECT`
- `CHARACTER_SELECT`
- `EMAIL_INBOX`
- `EMAIL_READING`
- `MINI_GAME_[TYPE]`
- `EMAIL_WRITING`
- `EMAIL_SENT`

3. Email System (`/components/EmailSystem`)

```
interface Email {
  from: "officer" | "agent" | "monkey" | "yourself"
  subject: string  // Pre-written prefix
  preview: string  // First 5 words
  fullContent?: string  // AI generated on click
}
```

4. Mini-Games (Intentionally Buggy)

Banana Game (`/games/BananaGame`)

- **Controls**: A/D (move), W (reach up), S (pick from floor), Space (jump)
- **Bug Features**:
    - Inconsistent hitboxes
    - Random control lag

    - Bananas occasionally phase through player
    - Score randomly resets

Sniper Game (`/games/SniperGame`)

- **Controls**: Mouse to aim, click to "deliver severance"
- **Bug Features**:
    - Scope drift
    - Random zoom glitches
    - Targets occasionally invincible

Golf Game (`/games/GolfGame`)

- **Controls**: Position player, use power meter
- **Bug Features**:
    - Power meter moves at random speeds
    - Ball physics occasionally moon gravity
    - Hole sometimes moves

AI Prompt Architecture

Base System Context

```
const BASE_CONTEXT = {
  game: "Let's Write an Email",
  setting: "SadCoin corporate office",
  tone: "Corporate absurdist humor",
  userRole: "intern",
  task: "Help user write actual email while playing"
}
```

Character Prompts

Officer

```
const OFFICER_PROMPT = {
  personality: "Corporate buzzword enthusiast, deadline obsessed",
  outputFormat: "3 high-level thoughts + call to action",
  vocabulary: ["synergies", "leverage", "mission-critical", "circle back"],
  emailLength: "3-4 paragraphs"
}
```

Agent

```
const AGENT_PROMPT = {
  personality: "Everything is code, speaks in double meanings",
  outputFormat: "3 specific details + incomprehensible attachment",
  hiddenMeanings: true,
  attachmentType: "fake dossier or SWOT analysis"
}
```

Monkey

```
const MONKEY_PROMPT = {
  personality: "Chaotic, banana-obsessed, random CAPS",
  outputFormat: "Stream of consciousness with tangents",
  mustInclude: ["banana references", "GM", "fire emoji"],
  coherence: 0.3  // Low coherence score
}
```

Implementation Phases

Phase 1: Core Terminal UI (Day 1)

1. Terminal component with green-on-black styling
2. Basic navigation (keyboard + mouse)
3. Wallet/Google connect flow
4. Character selection (intern only for MVP)

Phase 2: Email System (Day 2)

1. Inbox display with 4 emails
2. Dynamic email generation on click
3. AI integration for each character
4. Scrollable email reader

Phase 3: Mini-Games (Day 3-4)

1. Banana game (simplest)
2. Sniper game (if time)
3. Golf game (stretch goal)
4. Intentional bugs for each

Phase 4: Email Assembly (Day 5)

1. Combine inputs from all interactions
2. Generate final email
3. Send to user's real email
4. Optional: Reply triggers NFT

Phase 5: SAD Economy (Post-MVP)

1. SADCoin integration
2. FEELS generation from gameplay
3. Staking mechanism
4. Cross-chain via CCIP

Key Technical Decisions

Why Next.js?

- Server-side rendering for SEO (letswritean.email)
- API routes for LLM calls
- Easy deployment to Vercel
- TypeScript for type safety

Why Multiple LLMs?

- Creates personality differences
- Cost optimization (monkey = cheap)
- Comedic effect from quality variance
- AWS Bedrock provides easy access

Why Intentionally Buggy?

- We don’t have time to make it perfect
- Creates frustration → productivity loop
- Authentic "sad" experience
- Makes email writing seem appealing
- Memorable user experience

Team Responsibilities

Tippi (Product/Integration)

- AI prompt engineering
- Chainlink integration research
- Documentation and team coordination
- Playtesting and feedback

Jason (Creative/Design)

- Game design and flow
- Asset creation
- Dialogue and humor writing
- Visual language consistency

Crome/Vasiliy (Backend/Architecture)

- Smart contract development
- Backend architecture
- Reluctant frontend assistance
- Code quality (intentionally variable)

Will (Frontend/Music?)

- UI implementation
- Email system integration
- Potential: Sad piano soundtrack
- Good Rabbit experience applicable

Eman (Smart Contracts)?

- SAD coin contract
- FEELS token mechanics
- Cross-chain architecture
- Staking implementation

Critical Path Items

1. **Terminal UI with basic navigation** (blocks everything)
2. **Email generation with AI** (core feature)
3. **At least ONE mini-game** (proves concept)
4. **Email delivery system** (completes loop)
5. **Basic wallet integration** (for Web3 requirement)

Stretch Goals

- All three mini-games
- Full SAD/FEELS economy
- Cross-chain support
- Leaderboard system
- Chat room at water cooler
- Custom sad music soundtrack