

Project Kickoff Document

Project Title: Love & Dice RPG AI Companion

Goal

To create an immersive AI-powered roleplaying interface for the dating simulation RPG *Love & Dice*. The system will include a frontend chat UI, backend API for handling gameplay mechanics (Geist Rolls, character creation, etc.), and integration with Hugging Face models for roleplay and image generation. The platform is intended to serve both players and Game Masters (GMs), blending structured dice-based mechanics with AI-driven improvisational storytelling.

System Overview

The system is composed of the following layers:

1. Frontend Chat UI

- Built using React or similar framework.
- Allows player to engage in dialogue with AI-powered NPCs.
- Sends structured requests to backend API (e.g., perform a Geist Roll, generate an image, continue a story segment).

2. Backend REST API (Node.js or Python FastAPI)

- Endpoints for:
 - Character creation
 - Geist Roll resolution
 - AP tracking
 - Action table management
- Hugging Face inference for AI dialogue & image gen
- Stores and updates the probability table of known social actions
- Enables both static (rules-based) and generative (AI-based) interactions

3. AI Integration

- **Text Generation:**
 - Hugging Face models (e.g., roleplay-tuned LLMs like *MythoMax*, *Puffin*, etc.)
 - Prompts conditioned on character sheet, context, and recent interactions
- **Image Generation:**

- Optionally leverage a free model on Hugging Face (like `stablediffusion` via inference API)
- Generates profile pics, scene illustrations, emotional reactions, etc.

4. Geist Rolls System

- Core mechanic modeled after saving throws/checks in TTRPGs
 - Named "Geist Rolls" to represent inner spirit, emotional stakes
 - Each action is rolled against a Target Number (TN)
 - System includes:
 - Known Probability Table (JSON-driven)
 - GM-defined probabilities for new/undefined actions
 - Auto-save of new actions to extend the table
-

Phased Build Plan

Phase 1: Core Backend Scaffold

- REST API with Node.js or FastAPI (Python)
- Routes for character creation, Geist Rolls, and table management
- Mock integration with Hugging Face APIs
- In-memory or flat-file storage for initial dev

Phase 2: Frontend UI (Chat Focused)

- Simple chat interface (React)
- History view, basic emotion tagging
- Buttons for action proposals / roll attempts

Phase 3: AI Integration

- Connect to Hugging Face API (roleplay LLM)
- Send persona sheet + chat logs as context
- Connect to image model (optional user input, scene suggestions)

🎮 Phase 4: Game State + Narrative Tracking

- Store AP and relationship stages
- Log major decisions, narrative branches
- Enable saved sessions for replay or multi-day stories

🔧 Phase 5: Polishing + Deployment

- Secure API, add rate limiting, model safety
 - Finalize UI polish and deploy
 - Add onboarding walkthrough, rulebook links
-

Example Folder Structure

```
love-and-dice-rpg/
├── backend/
│   ├── app.py (or index.js)
│   ├── routes/
│   │   ├── character.py
│   │   ├── geist.py
│   │   ├── image.py
│   │   └── story.py
│   ├── models/
│   │   └── probability_table.json
│   ├── services/
│   │   └── huggingface_api.py
│   └── utils/
│       └── modifiers.py
├── frontend/
│   └── [React app code]
├── docs/
│   └── rulebook.md
├── requirements.txt / package.json
└── README.md
```

Next Step

Scaffold the backend with: - `/create-character` - `/geist-roll` - `/add-action` - `/list-actions`
- `/generate-text` - `/generate-image`

Then zip it up for distribution and local testing.