

# IMPLEMENTATION\_GUIDE.md

## 1. 目录结构

ai-dm/ ├── backend/ | | ├── app/ | | | | ├── main.py # FastAPI 路由 | | | | ├── engine.py # 游戏循环 | | | | ├── graph.py # 因果图解析 | | | | ├── agents.py # Classifier / GM | | | | └── schemas.py # Pydantic 模型 | | ├── tests/ | | └── requirements.txt | | ├── frontend/ | | | | ├── src/ | | | | └── vite.config.js | | ├── stories/ # YAML 剧情包 | | ├── .env.example | | └── docker-compose.yml

## 2. 开发里程碑

Day	任务
1	初始化仓库、FastAPI <code>/ping</code> 、Vue3 Hello World
2	实现 <code>graph.py</code> ，解析 YAML & 提供 <code>get_available_actions()</code>
3	<code>ClassifierAgent</code> ：函数调用 + 回退匹配
4	<code>GM_Agent</code> ：Prompt 拼装、SSE 输出
5	前端聊天 UI，拉取流式响应
6	敏感词过滤、异常重试、Docker 化、单元测试
7	参数面板、多剧情包、录制演示视频

## 3. 编码规范

- 语言 / 框架：Python 3.11 + FastAPI；Node 18 + Vue3 + Vite
- 格式：`black --preview`、`ruff`；前端 eslint + prettier
- 类型：`mypy --strict`、TypeScript strict
- 分支：`main` + `feat/*`，PR 通过 CI 后合并
- 测试：后端 pytest；前端 vitest
- 提交：Conventional Commits

## 4. 接口契约

```
POST /play
Body: { "session_id": str, "user_input": str }
Stream(SSE): { "role": "assistant|user", "content": str, "done": bool }

POST /reset
Body: { "session_id": str }
Return: 204 No Content
```

## 5. 安全与配置

.env: OPENAI\_API\_KEY、MODEL\_NAME、TEMPERATURE 所有模型调用封装在 agents.py 日志隐藏用户 PII / token

## 6. 运行方式

```
pnpm i --filter frontend...  
pnpm --filter frontend dev      # 前端  
  
pip install -r backend/requirements.txt  
uvicorn backend.app.main:app --reload # 后端  
  
# 一键 Docker  
docker-compose up --build
```