**Flamey’s Tavern Table**

**Web App Documentation**

**By Toderici Mihai Bogdan**

**Popa Alex**

**Version**

**1.0 — Draft**

**Technologies**

* **Frontend: React (TypeScript)**
* **Backend: ASP.NET Core (.NET 8, C#)**
* **Database: SQL Server or SQLite**
* **AI Integration: OpenAI API / Azure OpenAI (for advisor & assistant features)**

**⚔️ 1. Overview**

**Flamey’s Tavern Table is a full-stack web application that enhances Dungeons & Dragons campaigns for both Players and Dungeon Masters (DMs).**

**The platform allows:**

* **Players to create, manage, and level up their characters.**
* **DMs to design, organize, and run their campaigns with AI-driven help for content generation and worldbuilding.**
* **Both roles to interact through a shared, centralized system that supports storytelling, record-keeping, and creative assistance.**

**The app includes Flamey, the AI advisor — a cheeky, helpful flame spirit who guides users through game strategy, worldbuilding, and campaign management.**

**🧩 2. System Architecture**

**General Architecture**

**Frontend (React)**

**↕**

**REST API (ASP.NET Core)**

**↕**

**Database (SQL / EF Core)**

**↕**

**AI Services (OpenAI / Azure OpenAI)**

* **Frontend (React): Handles all user interactions, dashboards, forms, and displays.**
* **Backend (.NET): Provides APIs for player/DM data, campaign management, authentication, and AI interactions.**
* **Database: Stores user profiles, characters, NPCs, campaign data, and items.**
* **AI Layer: Handles queries to generate advice, NPC dialogue, quests, etc.**

**🎭 3. User Roles**

**3.1 Player**

**Players can:**

* **Create new characters (choose race, class, background, stats, spells, etc.)**
* **Manage inventory, experience points, level ups**
* **Keep session notes and campaign logs**
* **Chat with Flamey, the AI companion, for:**
  + **Combat strategies**
  + **Spell combos**
  + **Character optimization**
  + **Roleplay tips**

**3.2 Dungeon Master**

**DMs can:**

* **Create and manage campaigns**
* **Add and organize:**
  + **NPCs**
  + **Locations**
  + **Items**
  + **Quests**
  + **Events**
* **Use the AI to:**
  + **Generate story hooks or side quests**
  + **Populate random encounters**
  + **Suggest character arcs or twists**
* **Maintain a world map or campaign timeline**

**📂 4. Major Features**

| **Feature** | **Description** | **User Role** |
| --- | --- | --- |
| **Role Selection** | **Choose to sign in as Player or DM** | **Both** |
| **Dashboard** | **Personalized view showing current campaigns, characters, or worlds** | **Both** |
| **Character Builder** | **Step-by-step creation with guided prompts (race, class, stats, etc.)** | **Player** |
| **Inventory Manager** | **Add, edit, and equip items** | **Player** |
| **Level-Up Tracker** | **XP tracking and auto-suggestions for next upgrades** | **Player** |
| **AI Advisor (Flamey)** | **Chat-based assistant for strategy and roleplay ideas** | **Player** |
| **Campaign Manager** | **Tools to create and manage campaign data (NPCs, maps, etc.)** | **DM** |
| **AI Worldbuilder** | **Generate locations, NPCs, encounters, or lore snippets** | **DM** |
| **Session Notes** | **Shared notes for party and DM** | **Both** |
| **Authentication & Profiles** | **Register, log in, manage role and settings** | **Both** |

**🧠 5. AI Integration**

**AI Advisor (Player Side)**

* **Flamey provides in-character tips and context-aware advice.**
* **Can suggest:**
  + **Battle tactics**
  + **Best spell combos**
  + **Dialogue suggestions**
  + **Backstory ideas**
* **Integrated via an /api/advisor endpoint calling a text generation API.**

**AI Campaign Assistant (DM Side)**

* **Helps DMs generate content dynamically:**
  + **NPC backstories**
  + **Town descriptions**
  + **Random encounters**
  + **Quests and story arcs**
* **Supports context persistence — Flamey remembers the campaign state within the session.**

**🧱 6. Technical Implementation**

**Frontend (React)**

* **Routing: React Router (role-based navigation)**
* **State Management: Redux Toolkit / Zustand**
* **UI Library: TailwindCSS + shadcn/ui**
* **API Integration: Axios to communicate with .NET backend**
* **Authentication: JWT tokens stored securely in HttpOnly cookies**
* **Components:**
  + **RoleSelector**
  + **CharacterForm**
  + **CampaignDashboard**
  + **AIChatWindow**
  + **InventoryTable**

**Backend (.NET)**

* **Framework: ASP.NET Core Web API**
* **ORM: Entity Framework Core**
* **Controllers:**
  + **AuthController**
  + **CharacterController**
  + **CampaignController**
  + **AIController**
* **Services:**
  + **CharacterService (CRUD, leveling logic)**
  + **CampaignService (NPCs, items, etc.)**
  + **AIService (communicates with OpenAI API)**
* **Authentication: Identity / JWT**
* **Data Models:**
* **public class Character {**
* **public int Id { get; set; }**
* **public string Name { get; set; }**
* **public string Race { get; set; }**
* **public string Class { get; set; }**
* **public int Level { get; set; }**
* **public int Experience { get; set; }**
* **}**
* **public class Campaign {**
* **public int Id { get; set; }**
* **public string Name { get; set; }**
* **public List<NPC> NPCs { get; set; }**
* **public List<Location> Locations { get; set; }**
* **public List<Item> Items { get; set; }**
* **}**

**🔒 7. Authentication Flow**

1. **User registers → selects role (Player/DM).**
2. **Backend generates JWT.**
3. **React stores token securely and loads role-specific dashboard.**
4. **Access to endpoints gated by role-based authorization attributes.**

**💾 8. Database Schema (simplified)**

**Users**

**├── Id**

**├── Username**

**├── Role**

**├── Email**

**├── PasswordHash**

**Characters**

**├── Id**

**├── UserId (FK)**

**├── Name**

**├── Race**

**├── Class**

**├── Level**

**├── Experience**

**├── Inventory (JSON)**

**Campaigns**

**├── Id**

**├── UserId (FK)**

**├── Name**

**├── Description**

**NPCs**

**├── Id**

**├── CampaignId (FK)**

**├── Name**

**├── Description**

**├── Role**

**Items**

**├── Id**

**├── Campaign Id**

**├── Name**

**├── Type**

**├── Rarity**

**🪄 9. Future Enhancements**

* **Shared campaign boards (DM ↔ Players)**
* **Session logs & dice roller**
* **AI voice integration (Flamey as a talking mascot)**
* **Map integration (draw or auto-generate maps)**
* **Offline mode with local cache**

**🧭 10. Summary**

**Flamey’s Tavern Table combines creativity, structure, and AI to enhance tabletop storytelling.  
It provides tools for both sides of the screen:**

* **Players craft and grow their heroes.**
* **DMs craft and maintain worlds.**
* **Flamey ties it all together with guidance, inspiration, and just a bit of fiery attitude 🔥**