مشروع **المحكمة الافتراضية (Virtual Court)**

النسخة الكاملة من منصة **Pal-Law-AI** مع الملفات التالية:

* **Frontend**:
  + صفحات المصادقة (Login.jsx, Register.jsx, ForgotPassword.jsx)
  + لوحة التحكم (Dashboard.jsx, Sidebar.jsx, Header.jsx, UserManagement.jsx, Settings.jsx, Analytics.jsx, Profile.jsx)
  + صفحات الهبوط (LandingPage.jsx, HeroSection.jsx, Features.jsx, Footer.jsx)
  + المحكمة الافتراضية (VirtualCourt.jsx) باستخدام Three.js، @react-three/fiber، @react-three/drei، مع نماذج GLB
  + سياقات وإعدادات (AuthContext.jsx, ThemeContext.jsx)
  + الخدمات (api.js, auth.js, users.js)
  + الـ hooks (useAuth.js, useTheme.js)
  + CSS وجافاسكربت المساند
* **Backend (Flask)**:
  + نماذج البيانات (User.py, Role.py, Permission.py)
  + Routes (auth.py, users.py, settings.py)
  + أدوات مساعدة (auth\_utils.py, validators.py)
  + ملف الإعداد config.py
  + app.py لتشغيل السيرفر
* **ملفات GLB افتراضية**:
  + courtroom.glb
  + avatar\_judge.glb
  + avatar\_lawyer.glb
  + avatar\_defendant.glb
* **ملف README.md** مفصل كما صممناه سابقًا

اريد عمل كود لصفحة المحكمة الافتراضية وتفرعاتها وارتباطاتها واستخدام الواقع الافتراضي المعزز والذكاء الاصطناعي في عمل المحكمة الافتراضية الشبيهة بالمحاكم النظامية من حيث العناصر والادوات والمكونات داخل المحكمة من عنصر بشري ومادي والبروتوكولات التي تسير عليها في عقد الجلسات والمحاكمات والشخوص الذين يديرون المحكمة من قضاة ومحامين وكتبة وادعاء عام ومتهمين وحضور وامكانية الاستفادة من تطبيقات الالعاب وخاصة لعبة counterstrike في اختيار الشخصيات والتحكم فيها وفي حركتها واختياراتها وامكانية مشاركة مستخدمين اخرين من اجهزة اخرى في الشبكة للتفاعل والمشاركة في الجلسات والمحاكمات والمداولات التي تعقدها المحكمة الافتراضية وامكانية تغيير الادوار من قاض الى محامي الى مدعي عام او متهم او شاهد او من الحضور والباقي الغير مختارين يكونون باستخدام الذكاء الاصطناعي روبوت منافس او مشارك والمطلوب تصميم مشاهد ثلاثية الابعاد بالواقع الافتراضي والذكاء الاصطناعي كل ذلك بالكود المساند لمنصة pal-law-ai

مشروع **المحكمة الافتراضية (Virtual Court)** الذي تصفه هو مشروع ضخم ومتقدم يجمع بين **الواقع الافتراضي (VR/AR)، الذكاء الاصطناعي (AI)، والألعاب التفاعلية متعددة اللاعبين**. سأضع لك خطة واضحة ومخطط كود **متكامل المراحل** يمكنك البدء به على منصة pal-law-ai مع أمثلة عملية للكود لدعم هذا المشروع.

**1️⃣ الهيكل العام للمشروع**

**التقسيم الأساسي للـ Virtual Court:**

1. **الواجهة الرئيسية للمحكمة**
   * دخول المستخدم
   * اختيار الدور: قاضي، محامي، مدعي عام، شاهد، متهم، حضور
   * اختيار غرفة المحكمة الافتراضية
2. **المحكمة الافتراضية ثلاثية الأبعاد**
   * القاعة (Courtroom) بتقسيمها إلى:
     + منصة القاضي
     + منصة الدفاع والادعاء
     + أماكن الشهود والمتهمين والحضور
   * أدوات المحاكمة:
     + دفتر المحكمة / الكتيبات القانونية
     + أدوات التصويت والمناقشة
     + إمكانية تسجيل الجلسات صوتياً ومرئياً
   * شخصيات:
     + بشرية (مستخدمين متصلين عبر الشبكة)
     + روبوتات AI (ذكاء اصطناعي لتغطية الأدوار الغير موجودة)
3. **البروتوكولات**
   * فتح الجلسة، تقديم الأدلة، الاستجواب، المرافعات، الحكم
   * إدارة أدوار اللاعبين AI و Users
   * التفاعل المباشر مع الأدوات والشخصيات
4. **التفاعل الشبكي**
   * مشاركة مستخدمين آخرين عبر WebSockets أو WebRTC
   * تغييرات ديناميكية في الدور (Role switching)

**2️⃣ التكنولوجيا المقترحة**

| **الغرض** | **التقنية** |
| --- | --- |
| الواجهة ثلاثية الأبعاد | **Three.js** أو **Babylon.js** (Web-based VR/3D) |
| VR/AR | **WebXR API** |
| محاكاة AI للشخصيات | **OpenAI GPT API** (للسلوك والتفاعل النصي) + الذكاء الاصطناعي للقرارات |
| التواصل الشبكي | **Socket.io** أو **WebRTC** لمزامنة اللاعبين |
| محاكاة الألعاب | استلهام حركة الشخصيات من **Unity** أو تصميم مشابه للـ Counter-Strike في Three.js |