# 🦅 Durmah - Voice-Mode Tutor Widget for UK Law Students

**Your compassionate AI companion for Durham Law School success with ChatGPT-grade voice interaction**

Built with love for every Durham law student who needs support, encouragement, and expertise on their journey. Now featuring ultra-low-latency voice interaction that responds faster than ChatGPT’s voice mode.

Durmah Banner Version Voice Mode License

## 🌟 New Voice Features

### Voice-First Interaction

* **Ultra-Low Latency**: First audio response <1200ms (faster than ChatGPT Voice Mode)
* **Instant Barge-In**: Interrupt and pivot conversations <300ms
* **One-Click Voice**: Floating button → instant voice conversation
* **Real-Time Transcription**: Live captions with partial and final transcripts
* **WebRTC Direct Connection**: Bypass servers for minimal latency
* **Smart Fallbacks**: WebSocket backup + SpeechSynthesis TTS fallback

### Drop-In Widget

* **Universal Integration**: Works with any React/Next.js application
* **Floating Interface**: Non-intrusive floating button that expands to voice panel
* **Accessibility First**: ARIA labels, keyboard navigation, screen reader support
* **Mobile Optimized**: Touch-friendly controls for study-on-the-go
* **Status Indicators**: Clear visual feedback (listening/speaking/thinking)

## 🚀 Quick Start

### Live Demo

**🌐 Try Durmah Now:** [**https://pc3qe8ntf0ei.space.minimax.io**](https://pc3qe8ntf0ei.space.minimax.io)

1. Click the floating voice button (bottom-right)
2. Grant microphone permission when prompted
3. Start talking - Durmah responds in real-time!
4. Try interrupting mid-response to test barge-in functionality

### Prerequisites

* Node.js 18+
* Render account (for backend)
* Netlify account (for frontend)
* Supabase project with provided schema
* OpenAI API key with Realtime API access
* ElevenLabs API key (optional, for premium TTS)

## 📦 Deployment

### Backend Deployment (Render)

1. **Create Render Web Service**

* # Connect your GitHub repo to Render  
  # Set build command: cd Server && npm install  
  # Set start command: cd Server && npm start

1. **Configure Environment Variables in Render**

* # Copy all variables from Server/.env.example  
  # Set in Render Dashboard → Environment  
  NODE\_ENV=production  
  OPENAI\_API\_KEY=sk-your-key-here  
  ELEVENLABS\_API\_KEY=your-elevenlabs-key  
  SUPABASE\_URL=https://your-project.supabase.co  
  SUPABASE\_SERVICE\_ROLE=your-service-key  
  ALLOWED\_ORIGINS=https://your-netlify-site.netlify.app  
  # ... (see Server/.env.example for complete list)

1. **Health Check Configuration**
   * Render automatically monitors /api/healthz
   * No additional configuration needed

### Frontend Deployment (Netlify)

1. **Create Netlify Site**

* # Connect your GitHub repo to Netlify  
  # Set build command: cd Client && npm run build  
  # Set publish directory: Client/dist

1. **Configure Environment Variables in Netlify**

* # Set in Netlify Dashboard → Site Settings → Environment Variables  
  VITE\_API\_BASE=https://your-render-service.onrender.com  
  VITE\_SUPABASE\_URL=https://your-project.supabase.co  
  VITE\_SUPABASE\_ANON\_KEY=your-anon-key  
  # ... (see Client/.env.example for complete list)

1. **Build Settings**
   * Build command: cd Client && npm run build
   * Publish directory: Client/dist
   * Node version: 18+

### Database Setup (Supabase)

1. **Import Schema**

* -- Copy contents of database/schema.sql  
  -- Paste into Supabase SQL Editor  
  -- Run to create all tables and RLS policies

1. **Configure RLS Policies**
   * All policies are included in schema.sql
   * No additional configuration needed

## 🛠 Local Development

### One-Line Setup

# Clone and install everything  
git clone https://github.com/mohan0265/DurmahLegalBuddyGPT.git  
cd DurmahLegalBuddyGPT && npm run install:all  
  
# Set up environment files  
cp Server/.env.example Server/.env  
cp Client/.env.example Client/.env  
# Edit .env files with your API keys  
  
# Start development servers  
npm run dev

**Development URLs:** - Frontend: http://localhost:5173 - Backend: http://localhost:3001 - Voice WebSocket: ws://localhost:3001/voice

### Environment Configuration

1. **Server Environment** (Server/.env)

* cp Server/.env.example Server/.env  
  # Edit with your actual API keys

1. **Client Environment** (Client/.env)

* cp Client/.env.example Client/.env  
  # Edit with your local server URL

## 🎯 How It Works

┌─────────────────┐ WebRTC ┌─────────────────┐  
│ React Widget │◄─────────────►│ OpenAI Realtime │  
│ │ │ API │  
└─────────┬───────┘ └─────────────────┘  
 │  
 │ Fallback WebSocket  
 ▼  
┌─────────────────┐ REST/WS ┌─────────────────┐  
│ Render Backend │◄─────────────►│ Supabase │  
│ (Node.js/Express│ │ Database │  
│ + WebSocket) │ │ │  
└─────────────────┘ └─────────────────┘  
 ▲  
 │ TTS Requests  
 ▼  
┌─────────────────┐  
│ ElevenLabs │  
│ Text-to-Speech │  
└─────────────────┘

### Voice Flow

1. **Session Start**: Widget requests ephemeral key from backend
2. **WebRTC Connection**: Direct connection to OpenAI Realtime API
3. **Audio Streaming**: Real-time bidirectional audio over WebRTC
4. **Barge-In Detection**: Client detects user speech, interrupts current playback
5. **Transcription**: Live transcripts saved to Supabase for context
6. **Fallback**: WebSocket backup if WebRTC fails

## 🔧 Troubleshooting

### Microphone Issues

* **Chrome/Edge**: Ensure HTTPS for microphone access
* **Safari**: Requires user gesture (click) before audio access
* **Firefox**: Check media.navigator.permission.disabled setting
* **Mobile**: Test with headphones to avoid echo cancellation issues

### WebRTC Connection Issues

* Check TURN server configuration for NAT traversal
* Verify CORS settings for cross-origin requests
* Test with different networks (corporate firewalls may block WebRTC)
* Use browser dev tools Network tab to check WebSocket fallback

### Safari Quirks

* Safari requires AudioContext to be created on user interaction
* WebRTC may need additional polyfills on older Safari versions
* Test with latest Safari version for best compatibility

### Performance Issues

* **High Latency**: Check network connection and geographic distance to servers
* **Audio Dropouts**: Verify sample rate settings (24kHz recommended)
* **Memory Issues**: Monitor for WebRTC connection leaks in dev tools

### Common Error Messages

* Microphone permission denied: User must grant permission
* WebRTC connection failed: Check TURN server or use WebSocket fallback
* Session expired: Backend will automatically refresh ephemeral keys
* Supabase connection error: Check database credentials and RLS policies

## 🧪 Testing

### Voice Quality Tests

# Test ultra-low latency (should be <1200ms)  
1. Click voice button  
2. Say "Hello Durmah" and measure time to first audio response  
3. Verify response is <1200ms  
  
# Test barge-in functionality (should be <300ms)  
1. Ask a long question to get extended response  
2. Interrupt mid-response with new question  
3. Verify interruption happens <300ms  
4. Verify new response addresses interruption

### Session Stability

# 10-minute continuous session test  
1. Start voice session  
2. Maintain conversation for 10+ minutes  
3. Monitor browser console for errors  
4. Verify no connection drops or memory leaks

### Cross-Browser Testing

* ✅ Chrome 120+ (Recommended)
* ✅ Edge 120+
* ✅ Safari 16+ (with user gesture requirements)
* ✅ Firefox 120+ (may require WebRTC polyfill)
* ⚠️ Mobile browsers (test with headphones)

## 🎓 Academic Integrity Features

### Built-in Guardrails

* **No Ghostwriting**: Durmah provides guidance, not completed work
* **OSCOLA Citations**: Helps with proper legal citation format
* **Integrity Reminders**: Every response includes academic integrity context
* **Educational Scaffolding**: Breaks down complex topics for learning

### Student-Aware Memory

* **Progress Tracking**: Remembers topics studied and areas of struggle
* **Wellbeing Monitoring**: Detects stress patterns and suggests breaks
* **Personalized Support**: Adapts responses based on individual learning style
* **Context Retention**: Maintains conversation history for coherent support

## 📱 Widget Integration

### Drop-in Component

import { DurmahWidget } from './components/DurmahWidget';  
  
function App() {  
 return (  
 <div>  
 {/\* Your existing application \*/}  
 <DurmahWidget   
 apiBase="https://your-backend.onrender.com"  
 supabaseUrl="https://your-project.supabase.co"  
 supabaseAnonKey="your-anon-key"  
 />  
 </div>  
 );  
}

### Customization Options

* **Theme**: Light/dark mode support
* **Position**: Configurable floating button position
* **Size**: Compact/expanded widget modes
* **Voice**: Multiple voice options for TTS
* **Language**: Multi-language support (English default)

## 📊 Monitoring & Analytics

### Performance Metrics

* **First Audio Response Time**: Target <1200ms
* **Barge-in Response Time**: Target <300ms
* **Session Duration**: Average conversation length
* **Connection Success Rate**: WebRTC vs WebSocket usage
* **Error Rates**: Failed connections, dropped sessions

### Student Success Metrics

* **Engagement Time**: Daily/weekly usage patterns
* **Topic Coverage**: Areas of focus and improvement
* **Wellbeing Indicators**: Stress levels and support needs
* **Academic Progress**: Learning trajectory and achievements

## 🤝 Contributing

Durmah is built for the Durham Law community. Contributions welcome!

1. Fork the repository
2. Create a feature branch
3. Follow existing code style
4. Add tests for new features
5. Submit a pull request

## 📄 License

MIT License - Built with love for Durham Law students 💜

## 🆘 Support

For technical issues: - Check the troubleshooting guide above - Review browser console for error messages - Test with different browsers/devices - Verify environment variable configuration

For academic support: - Durmah is designed to guide, not replace learning - Always maintain academic integrity - Use as a supplementary learning tool - Seek human support for complex personal issues

**Built with ❤️ for Durham Law students by MiniMax Agent** *“Every great lawyer started as a student who needed support. Durmah is here to be that support.”*