Voice Translation Application Manual

1. Application Overview

The Voice Translation Application is a sophisticated web-based tool that combines voice recording, speech recognition, and language translation capabilities. It's designed to facilitate real-time voice recording, transcription, and translation across multiple languages.

Key Features:

- Real-time voice recording with waveform visualization
- Instant speech-to-text transcription
- AI-powered translation using Google's Gemini model
- Support for 13 different languages
- Secure authentication system
- Local storage for recordings and translations

2. System Architecture

2.1 Core Technologies

Frontend Framework

- Next.js 14.2.16
- React 18
- TypeScript for type safety

AI and Speech Processing

- Google Gemini AI (version 0.22.0) for translation
- Web Speech API for voice recognition
- Web Audio API for recording

Storage

- Local storage for recordings and user preferences
- HTTP-only cookies for session management

2.2 Key Components

Authentication System

```
// Authentication Provider
export function AuthProvider({ children }: { children: ReactNode }) {
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  // ... authentication logic
}
```

Recording System

```
interface Recording {
  id: string
  blob: Blob
  timestamp: Date
  transcript?: string
  sourceLanguage?: string
  translation?: {
    text: string
    targetLanguage: string
  }
}
```

3. Features

3.1 Authentication System

Implementation Details

```
// Session Management
const cookieStore = cookies();
cookieStore.set('session', JSON.stringify({
  email: user.email,
  name: user.name,
  isAdmin: user.isAdmin
}), {
  maxAge: 60 * 60 * 24 * 7, // 1 week
  path: '/'
});
```

User Access Levels

- Regular Users: Basic recording and translation features
- Admin Users: Additional management capabilities
- Guest Users: Limited to public pages

Security Features

- HTTP-only cookies
- Session expiration
- Protected route middleware

3.2 Voice Recording System

Recording Features

```
const startRecording = async () => {
  const stream = await navigator.mediaDevices.getUserMedia({ audio: true });
  const mediaRecorder = new MediaRecorder(stream, {
    mimeType: MediaRecorder.isTypeSupported('audio/webm') ? 'audio/webm' : 'audio/ogg'
  });
}
```

Supported Audio Formats

- WebM (preferred)
- OGG (fallback)
- WAV (export)

Audio Processing

- Real-time waveform visualization
- Noise reduction
- Auto-gain control

3.3 Language Support

```
const LANGUAGE_NAMES: Record<string, string> = {
  'eng': 'English',
  'can': 'Canadian English',
  'ind': 'Indonesian',
  'spa': 'Spanish',
```

```
  'mex': 'Mexican Spanish',
  'deu': 'German',
  'ita': 'Italian',
  'por': 'Portuguese',
  'jpn': 'Japanese',
  'kor': 'Korean',
  'hin': 'Hindi',
  'gle': 'Irish',
  'gre': 'Greek'
};
```

Translation Features

- Real-time translation
- Language auto-detection
- Dialect support
- Regional variations

3.4 User Interface

Components

Audio Player

```
export function AudioPlayer({ audioBlob, onPlaybackStart, onPlaybackEnd }: AudioPlayerProps) {
  // Playback controls
  // Progress tracking
  // Time formatting
}
```

Language Selector

```
export function LanguageSelector({ value, onChange, disabled }: LanguageSelectorProps) {
  // Language selection dropdown
  // Flag icons
  // Language codes
}
```

Text Actions

```
export function TextActions({ text }: TextActionsProps) {
  // Copy to clipboard
  // Share functionality
  // Export options
}
```

4. Technical Documentation

4.1 System Requirements

Development Environment

```
{
  "node": ">=14.x",
  "npm": ">=6.x",
  "dependencies": {
    "next": "14.2.16",
    "react": "^18",
    "react-dom": "^18",
    "@google/generative-ai": "^0.22.0"
  }
}
```

Browser Support

- Chrome (latest)
- Firefox (latest)
- Safari (latest)
- Edge (latest)

```
# Clone repository
git clone [repository-url]

# Install dependencies
npm install

# Set up environment variables
cp .env.example .env.local

# Start development server
npm run dev
```

4.3 Configuration

Environment Variables

NEXT_PUBLIC_GEMINI_API_KEY=your_api_key

NEXTAUTH_SECRET=your_secret

NEXTAUTH_URL=http://localhost:3000

5. User Guide

5.1 Getting Started

1. **Account Creation**
    - Navigate to `/auth/signup`
    - Enter required information
    - Verify email (if enabled)
2. **Recording Audio**
    - Click microphone button
    - Speak clearly
    - Click stop when finished
3. **Translation**
    - Select source language
    - Select target language
    - View translation results

5.2 Advanced Features

1. **Managing Recordings**
    - View recording history
    - Delete recordings
    - Share translations
2. **Custom Settings**
    - Language preferences
    - Audio quality
    - Interface customization

6. Security Considerations

6.1 Data Protection

1. **Audio Data**
    - Local storage encryption
    - Secure transmission
    - Automatic cleanup
2. **User Data**
    - Password hashing
    - Session management
    - Data retention policies

6.2 API Security

1. **Rate Limiting**
2. **Input Validation**
3. **Error Handling**

7. Troubleshooting

7.1 Common Issues

1. **Recording Problems**
    - Microphone permissions
    - Browser compatibility
    - Audio format support
2. **Translation Issues**
    - API connectivity
    - Language support
    - Text formatting

7.2 Error Messages

```
// Example error handling
try {
  // Operation
} catch (error) {
  console.error('Operation failed:', error);
  toast({
    title: "Error",
    description: "Specific error message",
    variant: "destructive",
  });
}
```

8. Development Guide

8.1 Code Structure

```
text

Apply to route.ts
```

8.2 Contributing

1. **Code Style**
    - TypeScript standards
    - Component patterns
    - Testing requirements
2. **Pull Request Process**

- Branch naming
- Commit messages
- Review process