Documentation for Microsoft Teams AI Ticket Response Assistant

This documentation provides a comprehensive guide for setting up, configuring, and using the Microsoft Teams AI Ticket Response Assistant application.

Table of Contents

---

1. Prerequisites

Before setting up the project, ensure that you have the following installed:

- **Node.js** (v16 or above)
- **npm** or **yarn**
- A modern web browser
- A code editor (e.g., Visual Studio Code)
- Access to the OpenWeatherMap API for weather-related queries
- Environment variables for Microsoft Azure Bot Framework and Hugging Face APIs.

---

2. Installation

1. Clone the repository:

   ```
   git clone <repository-url>
   cd <repository-name>
   ```

2. Install the dependencies:

   ```
   npm install
   ```

---

3. Environment Configuration

1. Create a `.env` file in the project root and add the following keys:

   ```
   REACT_APP_MICROSOFT_APP_ID=<Your Microsoft App ID>
   REACT_APP_MICROSOFT_APP_PASSWORD=<Your Microsoft App Password>
   ```

```
BOT_ENDPOINT=<Your Bot's Endpoint URL>
REACT_APP_AI_API_URL=<Your Hugging Face API URL>
REACT_APP_API_KEY=<Your Hugging Face API Key>
```

2. Replace the placeholders with your actual values.

---

4. Running the Application

1. Start the development server:

   ```
   npm start
   ```

2. Open the application in your browser at http://localhost:3000.
3. For a production build:

   ```
   npm run build
   ```

4. Project is deployed at : https://teams-ai-ticket-response-bot.azurewebsites.net/api/messages

---

5. Project Structure

```
src/
├── components/          # UI Components
├── hooks/               # Custom Hooks
├── pages/               # Application Pages
├── services/            # API Service Layer
├── utils/               # Utility Functions
├── App.tsx              # Root Component
├── index.tsx            # Application Entry Point
public/
├── index.html           # HTML Template
.env                     # Environment Variables
package.json             # Dependencies and Scripts
```

---

6. Features

- **Query Input**: Enter a query to receive an AI-generated response.
- **AI Response Card**: Interactive response cards for accepting, editing, or refining responses.
- **Feedback Form**: Collect user feedback on AI responses.
- **Admin Settings**: Configure API keys, endpoints, and toggle features.

---

7. Usage Guide

a. Submitting a Query

1. Navigate to the home page.
2. Enter your query in the input field.
3. Click "Submit Query."
4. View the AI-generated response in the response card.

b. Interactive Response Card

- **Accept**: Confirm the response.
- **Edit**: Modify the response text.
- **Refine**: Generate an improved version of the response.

c. Feedback Form

- Provide a rating (1–5) and optional comments.
- Submit the feedback to improve future AI responses.

d. Admin Settings

- Toggle features like feedback collection and response refinement.
- Update API keys and endpoints.

---

8. API Reference

AI Model API

- **Endpoint**: `https://api-inference.huggingface.co/models/mistralai/Mistral-7B-Instruct-v0.3`
- **Authorization**: Bearer token (`REACT_APP_API_KEY`)
- **Method**: `POST`
- **Request Body**:

  ```
  {
    "inputs": "<query>"
  }
  ```

---

9. Troubleshooting

Common Issues

- **Environment Variables Not Set**: Ensure `.env` file exists and contains valid values.
- **API Errors**: Verify your API keys and endpoints.
- **UI Not Loading**: Check browser console for errors and ensure dependencies are installed.

Debugging Tips

- Use `console.log` or `debug` statements in the code.
- Check API responses with tools like Postman or curl.

---

10. Contributing

1. Fork the repository.
2. Create a new branch:

```
git checkout -b feature/your-feature-name
```

3. Commit your changes:

   ```
   git commit -m "Add your message"
   ```

4. Push the branch:

   ```
   git push origin feature/your-feature-name
   ```

5. Submit a pull request.

For further assistance, please contact the development team or refer to the project README file.