This document outlines three multilingual solutions for a **Laravel Inertia.js React.js** web application. Each solution is explained in terms of **how it works, features, integration, cost, and whether it requires manual translation work**.

1. React-i18next (Predefined Translations - Open Source)

Overview:

react-i18next is a popular open-source library that provides structured multilingual support using JSON-based translation files. It allows developers to store predefined translations and switch languages dynamically within a React.js web application.

How It Works:

- Each language has a corresponding JSON file containing translated content.
- The React application loads the appropriate language file based on user selection or browser preferences.
- Translations are manually created and stored in files, requiring human effort for content translation.

Features:

✔️ Supports **language detection** and automatic switching.
✔️ Allows **lazy loading** of translations for better performance.
✔️ Works well with **server-side rendering (SSR)** and **Inertia.js**.
✔️ Community-driven and **completely free**.

Integration & Effort Required:

- **Manual translation** work is required since all content needs to be added to JSON files manually.
- Developers need to configure the **i18n instance** in the React.js app and ensure translation files are correctly structured.

Cost:

- **Software Cost**: **$0** (Open-source)
- **Translation Cost**: **Depends on hiring a translator** (or in-house effort)
- **Development Time**: Medium, as it requires manually adding translation keys and content.

2. Google Translate API (Automated Translations - Paid)

Overview:

Google Translate API provides **real-time automated translations**, making it ideal for applications that require dynamic multilingual support **without manual translation work**.

How It Works:

- Text content is sent to Google's servers, which return the translated text in the selected language.
- No need for manually maintaining translation files—Google handles everything dynamically.
- Useful for applications where content is frequently updated and needs automatic translation.

Features:

✔️ **Fully automated** translation with no manual effort required.
✔️ Supports **over 130 languages**.
✔️ **Fast and scalable**, ideal for large applications.
✔️ Can translate **user-generated content dynamically**.

Integration & Effort Required:

- Developers need to **integrate the Google Translate API** into the React.js web application.
- A **Google Cloud account** is required to obtain an API key.
- Google's API provides **both text-based** and **website translation** options.

Cost:

- **Software Cost**: Paid API (Usage-based pricing)
- **Pricing (as of March 2025)**:
    - First **500,000 characters/month → Free**
    - After that: **$20 per 1 million characters**
    - Example: Translating a **500-word page (~3,000 characters)** would cost **$0.06** beyond the free tier.
- **Development Time**: Low, as it **automates translation work**.

3. LinguiJS (Optimized Translation - Open Source)

Overview:

LinguiJS is an **efficient translation management tool** that extracts translatable text from the code and stores it in files for translation. Unlike react-i18next, it automates some of the translation process by identifying text directly from the source code.

How It Works:

- Developers mark translatable text in the codebase using **LinguiJS macros**.
- The system extracts these texts and generates translation files.
- These files are then manually translated and compiled for use.

Features:

✔️ **Extracts translations automatically** (less manual effort than react-i18next).
✔️ Supports **pluralization and rich text formatting**.
✔️ Works well with **server-side rendering and Inertia.js**.
✔️ Free and open-source, but **requires manual translation work**.

Integration & Effort Required:

- Developers need to **run an extraction command** to collect all translatable text.
- Translators need to **manually translate extracted content** and compile it.
- Works best for projects where structured content is used.

Cost:

- **Software Cost**: (Open-source)
- **Translation Cost**: **Manual translation required**
- **Development Time**: Medium (requires extraction, translation, and compilation)