# Voice Shopping Assistant — Technical Documentation

## 1. Introduction

1. The Voice Shopping Assistant is a multilingual, voice-driven shopping list manager with smart suggestions and search features.
2. It was developed as part of a Software Engineering technical assessment within a time limit.
3. The goal is to demonstrate the ability to design, build, and deploy a production-quality application using modern web technologies, voice recognition APIs, and clean coding practices.

## 2. System Architecture

1. Frontend: React.js (component-based architecture)
2. Voice Recognition: Web Speech API (browser-native speech recognition)
3. Parser & NLP: Custom utility (utils/parser.js) for extracting item name, quantity, unit, and category.

* Data:

1. data/products.js → sample product dataset for search results.
2. data/suggestions.js → predefined seasonal/frequent/substitute suggestions.
3. Styling: CSS (dark/light themes, glassmorphism, responsive layout).
4. Hosting: Vercel (static deployment for React app).
5. **How to use it:**
6. Tap the mic
7. Speak in for the things you want to add in your list.
8. Tap the mic again so that the item gets listed in the shopping list.
9. If you want to remove item, just ask for “ remove \_\_ item”.
10. If you want to search for an item then just ask for “ search \_\_ item”.
11. **High-Level Flow:**
12. User taps mic → Web Speech API captures speech.
13. Transcript passed to parser.js → identifies item, quantity, unit.
14. Item stored in ShoppingList.js state with category.
15. Smart suggestions and search results displayed dynamically.
16. UI provides real-time transcript + shopping list with category icons.
17. Then the mic is again tapped and then the captured voice stores the items in the shopping list.

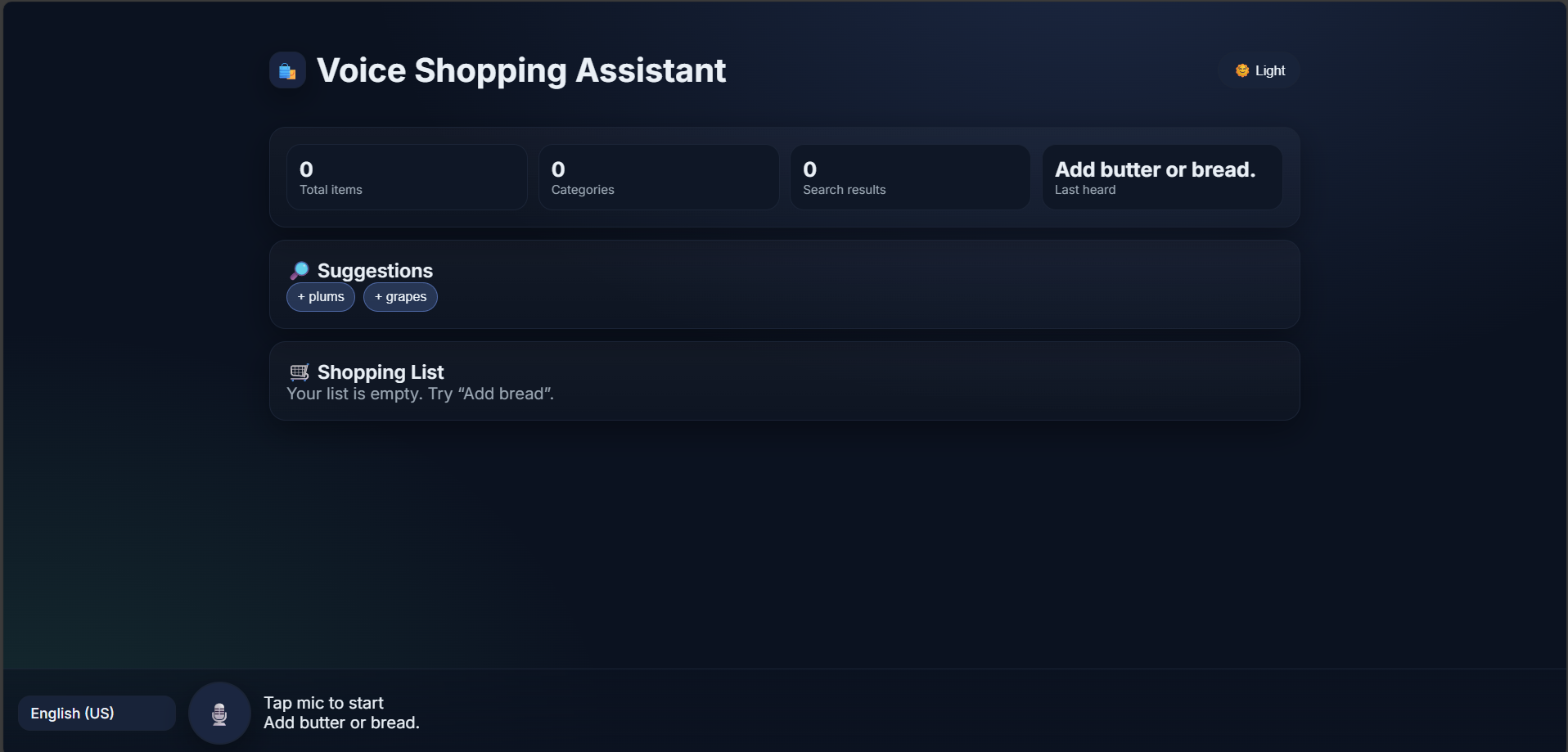
## 3. Key Features

1. Voice Input & NLP

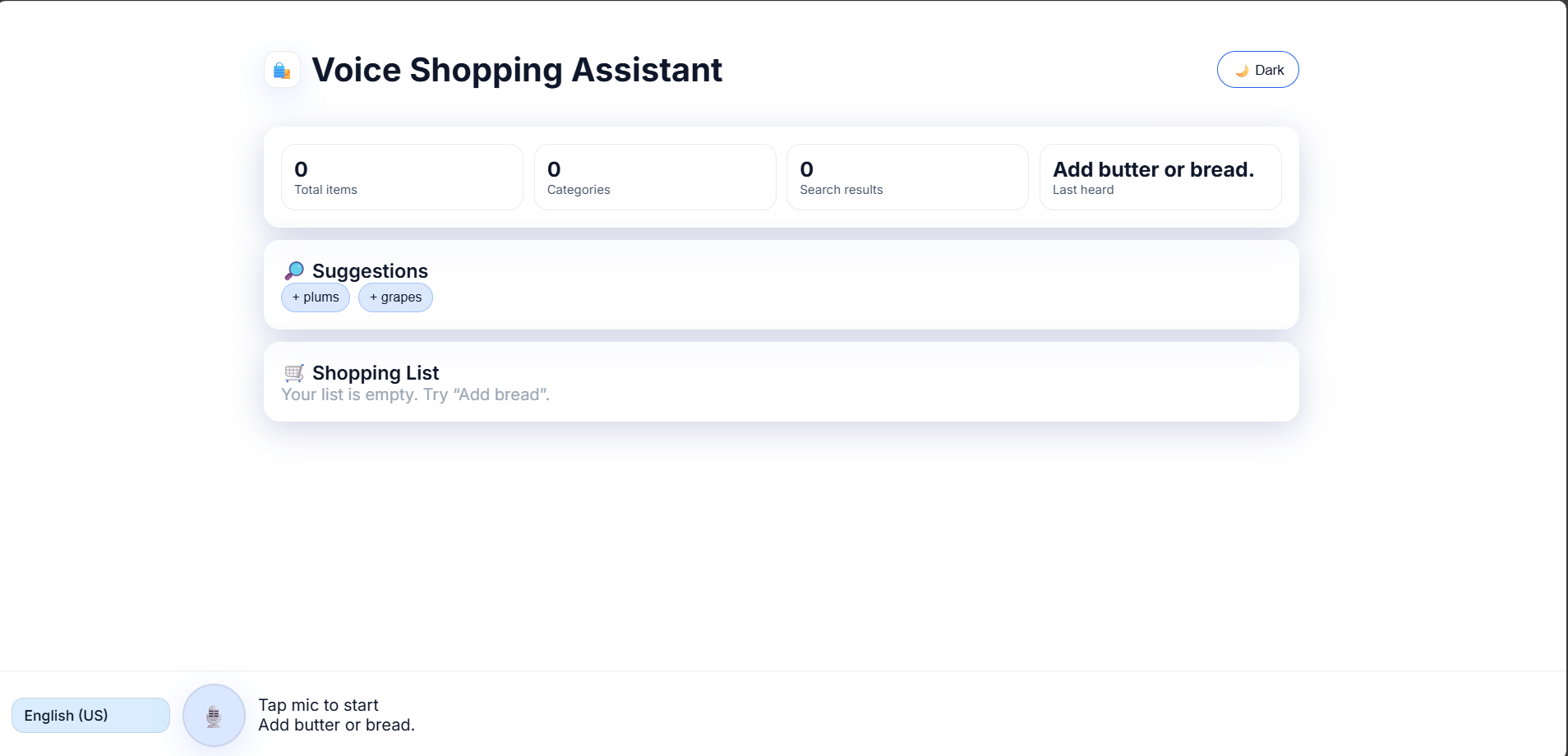
* Web Speech API integrated with support for 5 languages: English (US, IN), Hindi (India), Spanish (Spain), French (France), German (Germany).
* Flexible parsing to handle natural phrases: “Add 2 kg potatoes”, “दो किलो प्याज चाहिए”, “Acheter du lait”.
* Extracts { name, qty, unit, category } from transcript.

1. Shopping List Management
   * Add/remove/modify items via voice commands or buttons.
   * Auto-categorization: Dairy, Bakery, Vegetables, etc.
   * Items displayed with icons + category pills.
2. Smart Suggestions
   * recommendations.js provides: Frequently bought items, Seasonal suggestions, Substitutes (e.g., milk → almond milk).
3. Voice Search
   * Natural voice search for brand, type, and price filtering.
   * Example: “Find organic apples under ₹100”.
   * Displays mock results from products.js.
4. UI/UX

* Minimalist, glassmorphism design with dark/light themes.
* Real-time “Last Heard” transcript panel for user feedback.
* Responsive for desktop/mobile.



Dark Mode



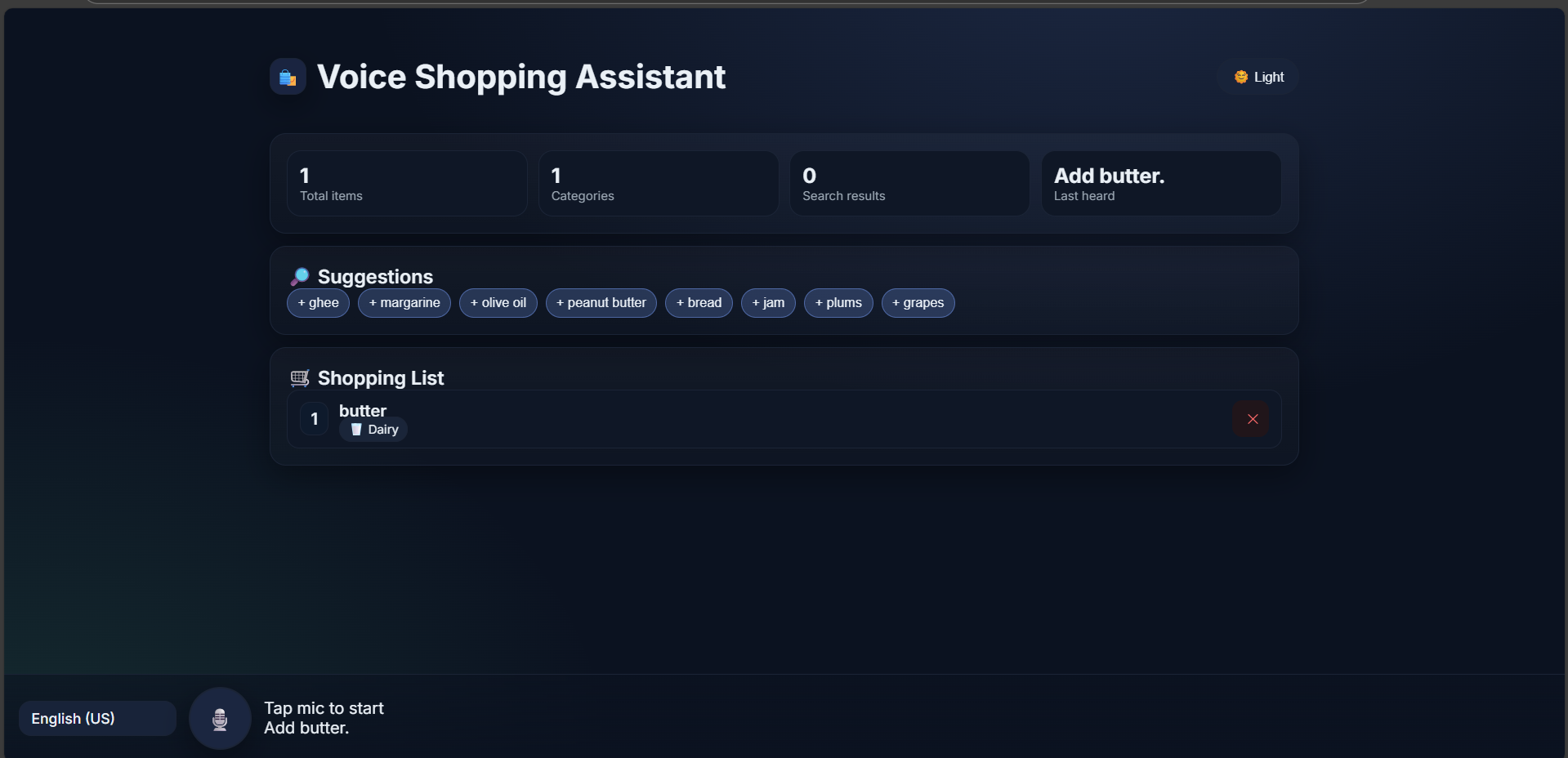
Light Mode

## 4. Implementation Details

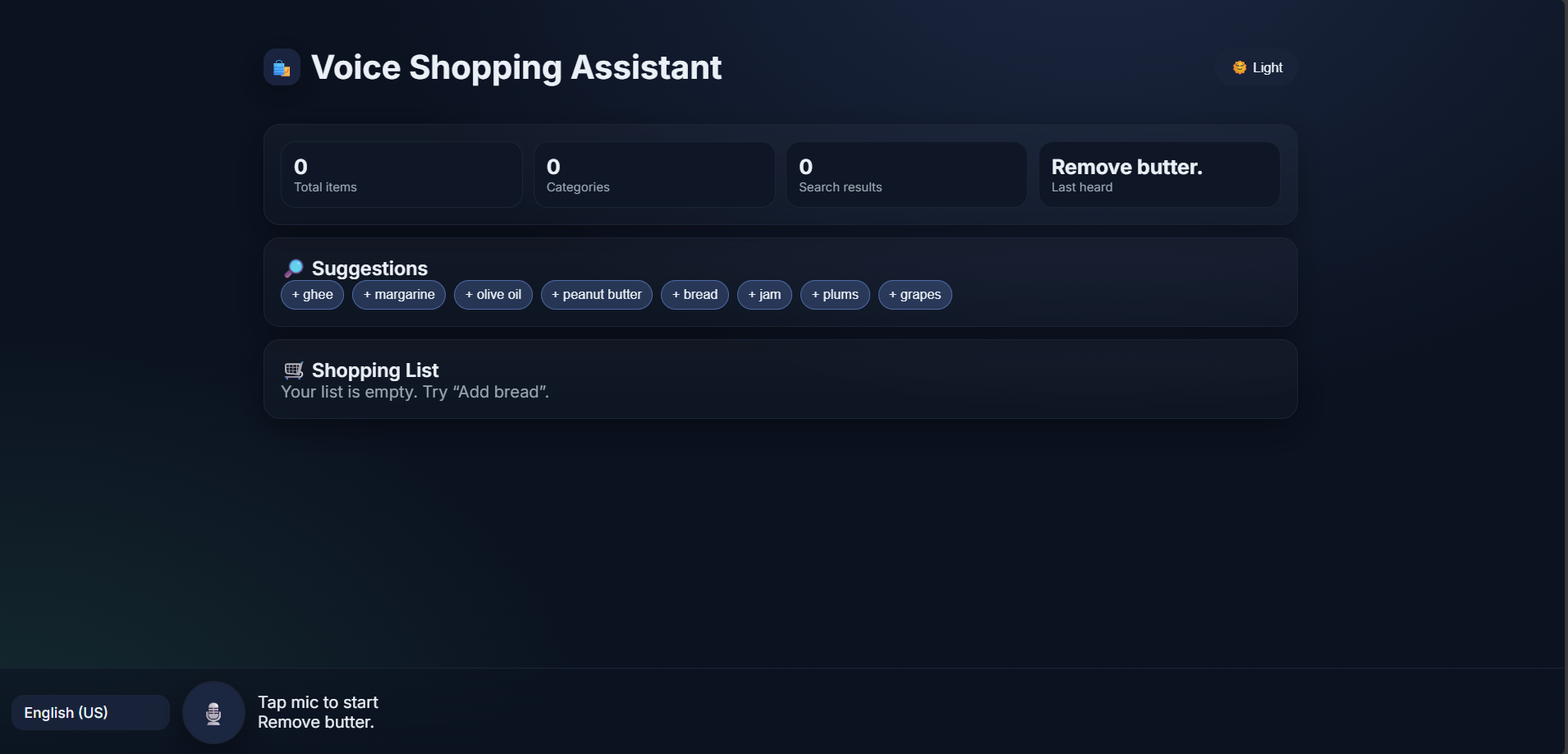
1. Core Component Breakdown:

* App.js → Main layout, handles theme & state.
* VoiceInput.js → Mic button, language dropdown, transcript.
* ShoppingList.js → Displays categorized list with remove buttons.
* SearchResults.js → Displays search results grid.
* StatusToast.js → Shows feedback (info/success/error).
* parser.js → NLP parser for extracting structured data.

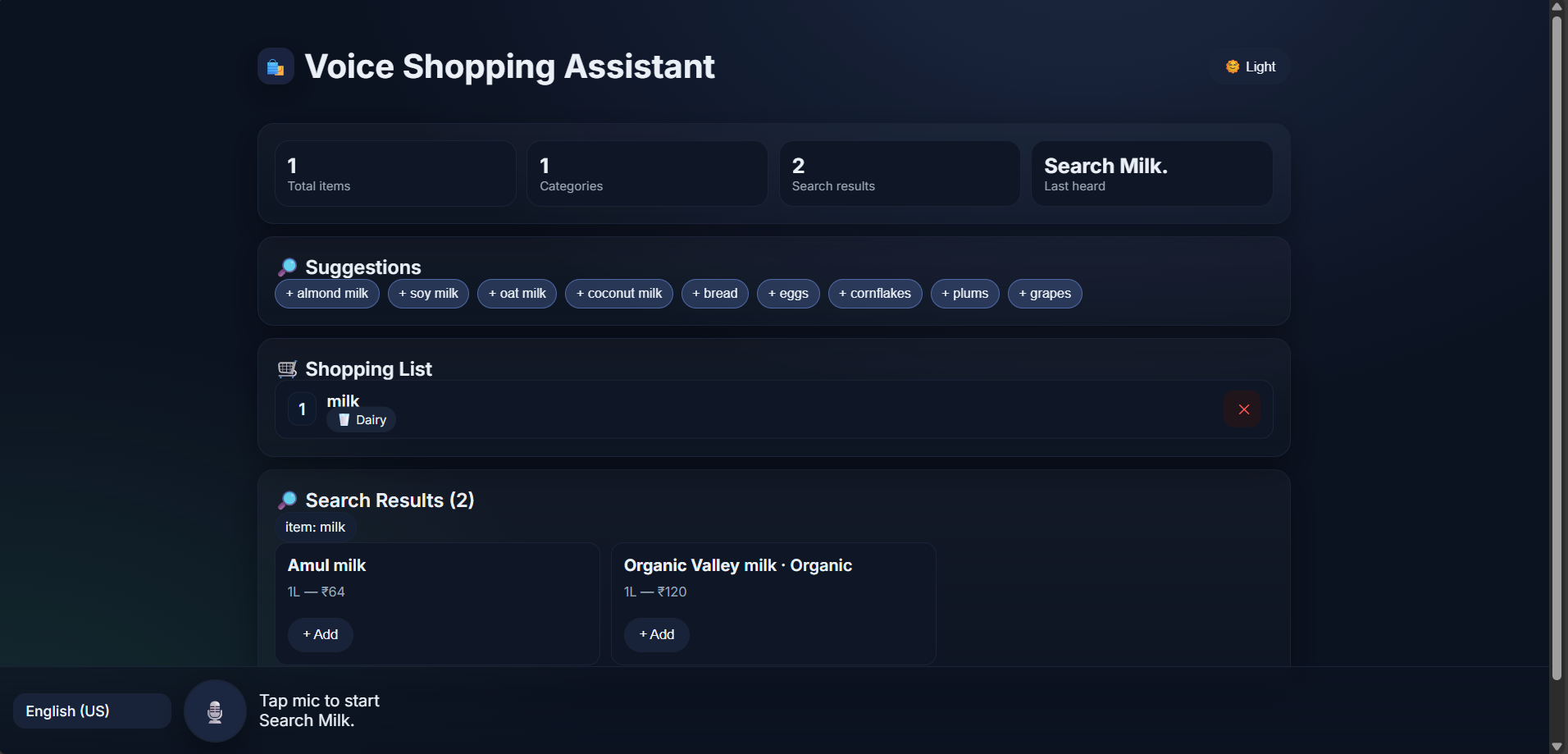
1. Item Added Command –“Add Butter “ also shows combinations of butter .



1. Item removed , Command “Remove butter”



1. Searches for items , Command “ Search Milk"



1. Error Handling:

* Warns if SpeechRecognition API unsupported.
* Empty transcripts ignored.
* ESLint used to catch unused variables & errors.

1. Performance:
   * Lightweight with local in-memory state (no DB required).
   * Parser optimized to handle multilingual regex efficiently.

## 5. Challenges & Solutions

1. Multilingual Parsing: Different syntax in Hindi, French, German required flexible regex + translation mapping.
2. Time Constraint: Focused on core features (voice input, NLP, smart suggestions) .
3. Cross-Browser Compatibility: Tested mainly on Chrome (best Web Speech API support).

## 6. Future Improvements

1. Backend with database (Firebase/AWS) for persistent shopping history.
2. AI/ML-based recommendation engine (personalized suggestions).
3. Offline voice support (using TensorFlow.js or Vosk).
4. Push notifications for “low stock” reminders.

## 7. Deliverables

1. Working URL: https://voice-shopping-assistant.vercel.app
2. GitHub Repo: https://github.com/ArjitaSahu123/Voice-Shopping-Assistance
3. Documentation: This technical doc + README