High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **AsyncFetch - API Data Fetcher**, a frontend educational application. The app is built with React, JavaScript, and Tailwind CSS, and is designed to fetch and display data asynchronously from a public API.

---

1. System Architecture Overview

Architecture Description:
AsyncFetch is a single-page application (SPA) structured into modular React components. It interacts with a public API via asynchronous JavaScript calls, processes the response, and renders the data using Tailwind CSS for styling.

Main System Modules

Module	Description
UI Layer	React components for layout, user input, and data display
API Service Layer	Handles asynchronous API requests and error management
State Management	Manages application state (e.g., fetched data, loading status)
Styling Layer	Applies responsive design using Tailwind CSS

---

2. Component Interactions

Interaction Step	Description
1. User Action	User initiates data fetch (e.g., button click)
2. API Request	API Service Layer sends async request to public API
3. Data Processing	Response is parsed and stored in State Management
4. UI Update	UI Layer re-renders to display fetched data or errors

---

3. Data Flow Overview

Source	Flow Description	Destination
User	Initiates fetch action	UI Layer
UI Layer	Triggers async API call	API Service
API Service	Receives data, handles errors	State Management

State Mgmt	Updates state with new data or error	UI Layer
UI Layer	Renders updated data or error messages	User

---

4. Technology Stack

Layer	Technology
Framework	React
Language	JavaScript (ES6+)
Styling	Tailwind CSS
API	Public REST API
Tooling	npm, Webpack/Vite

---

5. Scalability & Reliability

- **Scalability:** Designed for educational/demo use; modular React components allow easy extension. For larger datasets, implement pagination or lazy loading.
- **Reliability:** Handles API errors gracefully; displays user-friendly error messages. Uses React state to ensure UI consistency.
- **Security:** No sensitive data handled; follows best practices for API consumption in frontend apps.

End of Document