That sounds like an incredible and meaningful application! A tool like this can have a huge impact on representation, literacy, and a love for reading. To help you build this AI-powered React app effectively, here's a breakdown of the **requirements**—technical, functional, and data-related—along with ideas on how AI can be integrated.

## **🧩 Core Functionality Requirements**

### **1. User Profiles and Inputs**

Users (parents or children) should be able to specify:

* **Reader’s age** or **grade level**
* **Reading ability** (struggling, on grade level, advanced)
* **Book format preferences** (e.g., picture book, early reader, middle grade, graphic novel, chapter book, audiobook, etc.)
* **Interests or genres** (e.g., adventure, fantasy, real-life stories, STEM, sports, history)
* **Representation preferences**:  
  + Black/African American protagonists
  + Family or cultural themes
  + Historical or contemporary settings
* **Preferred themes or values** (e.g., empowerment, friendship, overcoming obstacles, STEM focus)
* Whether the child is reading independently or the parent is reading with them

### **2. Book Recommendations Engine**

This will use AI to match the user’s profile with curated book options. The backend system should:

* Use NLP or a fine-tuned recommendation model trained on:  
  + Summaries/descriptions of books
  + Reader reviews and reading level metadata
  + Thematic keywords
* Include or integrate with a **book metadata API**, like:  
  + Open Library
  + Google Books API
  + Bookshop.org affiliate API
  + Possibly scraped or curated datasets from Black-owned bookstores or diversity-focused collections

### **3. User Experience and Front-End Features**

* Search bar with AI-enhanced autocomplete (e.g., “Books like *Hair Love*”)
* Interactive quiz for new users
* Book cards with:  
  + Cover image
  + Age range
  + Reading level
  + Short summary
  + Buy or borrow links
  + AI-generated personalized note (e.g., "Perfect for a curious 2nd grader who loves science!")
* Save to "My Library"
* Share with others
* Track reading progress or books read

## **🧠 AI and ML Integration**

### **1. Recommendation System**

* Use a **content-based filtering model** (like cosine similarity with TF-IDF vectors on book descriptions) or **collaborative filtering** with a dataset of user preferences if available.
* **Fine-tune a language model** to rank or suggest books based on user profiles (e.g., using GPT-4 or similar).
* Use GPT for generating custom summaries, parent guides, or “why this book?” blurbs.

### **2. Chatbot / Book Concierge**

* Use a GPT-powered chatbot to interact with kids/parents:  
  + “Hi there! How old is your reader?”
  + “Do they like fantasy adventures or real-life heroes?”
  + Then suggest books dynamically.

## **📚 Book Data Sources and Curation**

You’ll need a reliable and diverse database. Consider:

* Partnering with **We Need Diverse Books**, **Lee & Low Books**, or **The Brown Bookshelf**
* Scraping public lists from:  
  + GoodReads (e.g., “Books with Black Protagonists for Kids”)
  + Common Sense Media
  + The Conscious Kid
  + Library systems with APIs (like the NYPL or WorldCat)

Include metadata:

* Title, Author
* Cover image
* ISBN
* Reading level (e.g., Lexile, F&P, DRA)
* Summary
* Keywords/tags (e.g., family, friendship, hair love, identity)
* Representation tags (Black girl protagonist, African American family, African culture, etc.)

## **🛠️ Technical Stack (Suggested)**

### **Frontend (React)**

* React + TypeScript
* TailwindCSS or Chakra UI for styling
* Redux or Context API for state management
* Axios for API calls
* Auth: Firebase Auth or Auth0

### **Backend (AI + APIs)**

* Node.js + Express OR Python (Flask/FastAPI)
* AI engine:  
  + GPT-4 via OpenAI API (for summarization, recommendations)
  + Scikit-learn for simple similarity algorithms
* Database:  
  + PostgreSQL or MongoDB for storing user data and book info
  + Redis (optional) for caching

### **Hosting**

* Frontend: Vercel or Netlify
* Backend: Render, Railway, or AWS Lambda
* Database: Supabase, MongoDB Atlas, or Firebase

## **📈 Future Features (Optional but Cool)**

* Parental controls / child-safe mode
* Integration with local library catalogs (e.g., OverDrive or Libby)
* Reading challenges and badges
* Book clubs or community features
* Personalized book box subscription integrations