**Product Requirements Document (PRD) for Web Solution for Household Item Rentals (HI Rentals)**

**Project Title: Implementation of a Web Solution for Household Item Rentals**

**Application Name: HI Rentals**

**Terminologies Used:**

 **Owner** – A user who **lists and lends** household items for rent.

 **Renter** – A user who **borrows and rents** household items for temporary use.

**Project Overview:**

The goal of this project is to develop a peer-to-peer (P2P) web-based platform that allows users to rent household items for short-term stays. The platform will connect individuals who need household items (renters) with those who have items available for rent (owners). The system will focus on affordability, convenience, and trust-building, ensuring a seamless experience for both renters and owners.

The platform will be built using **React Vite (Typescript)** and **Firebase** as BAAS, including authentication, database, and storage. The system will also include features like secure payments, item verification, and user reviews to ensure trust and reliability.

**1. Functional Requirements**

**1.1 User Authentication**

* **Account Creation**: Users can create an account using their email or phone number.
* **Password Recovery**: Users can reset their password via email or SMS.
* **Secure Authentication**: Implement Firebase Authentication for secure login and signup.
* **User Roles**:
  + **Renter**: Can search, request, and rent items.
  + **Owner**: Can list items for rent and manage their listings.
  + **Admin**: Can manage users, items, and verify item conditions.
* **Email/Phone Verification**: Users must verify their email to activate their account.

**1.2 Item Listings and Search**

* **Item Listing**: Owners can list items with details such as:
  + Item name
  + Description
  + Category (e.g., furniture, appliances, electronics)
  + Condition (e.g., new, used, good condition)
  + Rental price (per day/week/month)
  + Availability dates
  + Location (address or city)
  + Photos of the item
* **Search and Filter**: Renters can search for items by:
  + Category
  + Location
  + Price range
  + Availability dates
  + Item condition
* **Sorting**: Items can be sorted by price, distance, or user ratings.

**1.3 Rental Transactions**

* **Rental Request**: Renters can request to rent an item by specifying:
  + Rental duration (start and end date)
  + Delivery or pickup preference
* **Rental Agreement**: A digital rental agreement will be generated, including:
  + Rental terms and conditions
  + Security deposit (if applicable)
  + Payment details
* **Payment Processing**: Integrate a secure payment gateway (e.g., Paystack) for:
  + Rental fees
  + Security deposits
  + Refunds (if applicable)
* **Payment Tracking**: Users can track payment status and history.

**1.4 Item Verification**

* **Pre-Rental Verification**:
  + Owners must upload photos or videos of the item’s condition (or a video call schedule with an admin) before listing.
  + Admins will verify the item’s condition before approving the listing.
* **In-Rental Verification**:
  + Owners must upload photos or videos of the item’s condition (or a video call schedule with an admin) when item receives an order (requested by a buyer).
  + Admins will verify the item’s condition before approving to be lent to the renter.
* **Post-Rental Verification**:
  + Renters must upload photos or videos of the item’s condition (or a video call schedule with an admin) before returning it.
  + Admins will verify the item’s condition and approve the return.
* **Verification Reports**: Admins can generate verification reports with comments and status (Approved/Rejected).

**1.5 Ratings and Reviews**

* **Item Ratings**: Renters can rate items based on their experience (1-5 stars).
* **Owner Ratings**: Renters can rate owners based on their responsiveness and item condition.
* **Reviews**: Users can leave detailed reviews for items and owners.
* **Trust Score**: Each user (renter and owner) will have a trust score based on their ratings and reviews.

**1.6 Notifications and Alerts**

* **Rental Alerts**: Notify renters and owners about:
  + Rental due dates
  + Upcoming payments
  + Item return reminders
* **New Listings**: Notify renters when new items matching their preferences are listed.
* **System Notifications**: Notify users about:
  + Account updates (e.g., password changes)
  + Verification status (e.g., item approved/rejected)
  + Payment confirmations

**1.7 Admin Dashboard**

* **User Management**: Admins can view, edit, or delete user accounts.
* **Item Management**: Admins can view, approve, or remove item listings.
* **Verification Management**: Admins can verify items before and after rentals.
* **Transaction Monitoring**: Admins can view all transactions and payment statuses.
* **Reports**: Admins can generate reports on:
  + User activity
  + Item listings
  + Rental transactions
  + Revenue (if applicable)

**2. Non-Functional Requirements**

**2.1 Security**

* **Data Protection**: Ensure all user data is encrypted and stored securely in Firebase.
* **Secure Payments**: Use a trusted payment gateway (e.g., Paystack) for secure transactions.
* **Authentication**: Implement Firebase Authentication with email/phone verification and password hashing.
* **Role-Based Access Control**: Ensure only authorized users can access certain features (e.g., only admins can verify items).

**2.2 Scalability**

* **Database Scalability**: Use Firebase Firestore for scalable and real-time data storage.
* **Performance Optimization**: Ensure the platform can handle increasing users, listings, and transactions without performance degradation.
* **Load Balancing**: Implement caching and load balancing for high traffic.

**2.3 User-Friendliness**

* **Responsive Design**: Ensure the platform is fully responsive and works on both desktop and mobile devices.
* **Intuitive UI/UX**: Design a clean and intuitive user interface with easy navigation.
* **Accessibility**: Ensure the platform is accessible to users with disabilities (e.g., screen reader support).

**2.4 Performance**

* **Fast Load Times**: Optimize the platform for fast page load times, especially for search results and item listings.

**2.5 Reliability and Availability**

* **Minimal Downtime**: Ensure the platform has high uptime (99.9%).
* **Error Handling**: Implement proper error handling and user-friendly error messages.
* **Backup and Recovery**: Regularly back up data and implement recovery mechanisms in case of failures.

**2.6 Legal Compliance**

* **Data Privacy**: Comply with data protection laws (e.g., GDPR, CCPA).
* **Rental Agreements**: Ensure rental agreements comply with local laws and regulations.
* **Security Deposits**: Implement clear policies for security deposits and refunds.

**3. Additional Features**

**3.1 Chat System**

* **In-App Messaging**: Renters and owners can communicate directly within the platform to discuss rental details or admin to verify item condition.
* **Notification for Messages**: Users receive notifications for new messages.

**3.2 Social Login**

* **Google Login**: Allow users to sign up and log in using their Google accounts.

**3.3 Wishlist**

* **Save Items**: Renters can save items to a wishlist for future rentals.
* **Wishlist Notifications**: Notify renters when items in their wishlist become available or drop in price.

**3.4 Analytics Dashboard**

* **User Analytics**: Provide owners with insights into their item performance (e.g., views, rentals, earnings).
* **Platform Analytics**: Provide admins with insights into platform usage (e.g., active users, popular items).

**4. Technical Stack**

**Frontend:**

* **Framework**: React Vite
* **State Management**: Redux
* **UI Library**: Shadcn / Chakra UI
* **Routing**: React Router

**Backend (BAAS):**

* **Database**: Firebase Firestore
* **Authentication**: Firebase Authentication
* **Storage**: Firebase Storage (for item photos/videos)
* **Hosting**: Vercel

**Payment Integration:**

* **Payment Gateway**: Paystack

**5. Implementation Plan**

**Phase 1: Core Features**

* User Authentication (Signup, Login, Password Recovery)
* Item Listing and Search
* Rental Transactions (Request, Payment, Agreement)
* Basic Admin Dashboard (User and Item Management)

**Phase 2: Advanced Features**

* Item Verification (Pre and Post Rental)
* Ratings and Reviews
* Notifications and Alerts
* Chat System

**Phase 3: Additional Features**

* Wishlist
* Analytics Dashboard

**6. Conclusion**

This document outlines the detailed product requirements for the **Web Solution for Household Item Rentals**. The platform will be built using **React Vite** and **Firebase**, ensuring a scalable, secure, and user-friendly experience. The system will include core features like user authentication, item listings, rental transactions, and item verification, along with advanced features like ratings, reviews, and a chat system. Additional features like a wishlist, referral system, and multi-language support will enhance the user experience and make the platform more competitive.