**Architecture Document**

**Hestia-Verified Real Estate Marketplace**

**1. Introduction**

**Hestia Verified Real Estate** is a **secure, transparent, and verified property marketplace** designed for buyers, sellers, and renters. The platform ensures fraud-free real estate transactions through multi-level verification involving admin checks and agent site visits.

This document outlines the **architecture methodology**, **justification**, and **key diagrams** supporting the system's design.

## ****2.**** Chosen Architecture: Microservices Architecture

Justification for Microservices Architecture: Justification for Microservices Architecture:

1. **Scalability**: Each service (user management, property listing, verification, notifications, payments) can scale independently.

2.**Flexibility**: Different teams can work on independent services (e.g., authentication, verification, and search functionalities).

3. **Reliability**: Failure in one service does not impact the entire system, ensuring high availability.

4. **Technology Agnostic**: Different microservices can use different technologies best suited for their functionality.

5. **Security & Compliance**: Sensitive services like authentication and verification can be isolated for better security.

## ****3****. System Components

|  |  |
| --- | --- |
| Component | Description |
| User Service | **Manages authentication, role-based access (buyer, seller, agent, admin).** |
| Property Service | Handles property listings, image uploads, and document storage. |
| |  | | --- | | **Verification Service** |  |  | | --- | |  | | Processes admin and agent verification, real-time property visits. |
| Payment Service | Handles secure payments for premium listings, booking fees, etc. |
| Search & Filter Service | Enables buyers and renters to search for properties with filters. |
| |  | | --- | | **Notification Service** |  |  | | --- | |  | | Sends real-time updates to users regarding approvals, rejections, and inquiries. |
| Admin Panel Service | Provides admins with property approval workflows and analytics. |

## The system consists of the following microservices:

1.**Authentication Service** – Manages user authentication, registration, and role-based access control.

2.**User Management Service** – Handles user profiles, roles, and account settings.

3.**Property Listing Service** – Manages property submissions, updates, and listing statuses.

4.**Verification Service** – Facilitates admin and agent verification for property approvals.

5.**Search & Filter Service** – Enables buyers and renters to search and filter properties efficiently.

6.**Notification Service** – Sends real-time updates via email, SMS, and push notifications.

7.**Transaction & Payment Service** – Manages secure payments for premium listings and transactions.

8.**Admin Panel Service** – Provides admins with tools to manage listings, verifications, and analytics.

9.**File & Media Service** – Handles image and document uploads for property verification.

**4. Architecture Components & Justification**

|  |  |  |
| --- | --- | --- |
| **Component** | **Description** | **Justification** |
| **API Gateway** | Acts as a single entry point, routing requests to appropriate microservices. | Enhances security, load balancing, and request management. |
| **Authentication Service** | Handles user login, registration, and role-based access control using JWT. | Ensures secure authentication and authorization. |
| **User Management Service** | Manages user profiles, roles (buyer, seller, agent, admin), and account settings. | Provides structured user management and access control. |
| **Property Listing Service** | Handles property submissions, updates, and listing statuses. | Enables seamless property management for sellers. |
| **Verification Service** | Manages admin and agent-based property verification processes. | Ensures property authenticity and fraud prevention. |
| **Search & Filter Service** | Allows buyers and renters to search and filter properties efficiently. | Enhances user experience with quick property discovery. |
| **Notification Service** | Sends email, SMS, and push notifications for system updates. | Keeps users informed about listing status, verification, and approvals. |
| **Admin Panel Service** | Provides admin tools for managing listings, verifications, and analytics. | Enables platform moderation and data-driven decision-making. |
| **File & Media Service** | Manages image and document uploads for property verification. | Ensures secure and scalable media storage. |
| **Messaging & Communication Service** | Facilitates communication between buyers, sellers, and agents. | Enhances collaboration and transparency in transactions. |
| **Logging & Monitoring Service** | Tracks system events, errors, and performance metrics. | Ensures system reliability, security, and debugging capabilities. |

## ****5. Detailed Architecture Explanation - Data Flow****

1. **User logs in** → API Gateway forwards request to **User Service**.
2. **Battery data ingestion** → Collected from EV logs and stored in **Battery Data Service**.
3. **Prediction request** → API Gateway routes the request to **Machine Learning Service**.
4. **ML Model processes data** → Sends prediction to **Battery Data Service**.
5. **User Dashboard fetches battery health** from Battery Data Service via the API.
6. **Notification Service triggers alerts** if degradation is detected.