**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 26 -052025 |
| Team ID | LTVIP2025TMID55949 |
| Project Name | House Hunt: Find your perfect rental home |
| Maximum Marks | 4 Marks |

Here’s an expanded theoretical explanation for the **Solution Architecture** section of your *House Hunt* project report. This version adds depth to the original content and is structured to suit documentation or academic submission:

**Solution Architecture (Enhanced)**

The *House Hunt* application is a full-fledged MERN-stack-based solution designed to simplify the house renting process by digitizing the interactions between renters and property owners. The architecture of this system is strategically designed to ensure modularity, scalability, security, and performance while maintaining a user-centric approach.

**1. Objective of the Solution Architecture**

The architecture aims to:

* Facilitate **seamless communication** and interaction between tenants and property owners.
* Provide a **centralized platform** for managing property listings, bookings, and transactions.
* Ensure **high availability**, **performance**, and **data integrity** throughout the system.

**2. Key Components and Features**

**a) Effortless Property Search and Booking**

* An intuitive frontend interface built with **React.js** allows users to easily browse, filter, and view verified property listings.
* Integrated **search filters** based on location, price, property type, and availability.
* Real-time availability status and **booking confirmation system** using backend APIs.

**b) End-to-End Rental Management**

* **Owner dashboards** for managing property details, rent status, tenant profiles, and availability.
* **Renter dashboards** to view current bookings, payment history, and chat with owners.
* Maintenance request module for ongoing communication.

**c) Scalable User Authentication and Authorization**

* **JWT-based authentication** ensures secure and scalable login/session handling.
* **Role-based access control (RBAC)** is implemented to distinguish between Admins, Owners, and Renters.
* Support for **user registration, verification (via email/SMS),** and secure login.

**d) Secure and Trackable Rent Payment Transactions**

* Integration with **payment gateways (e.g., Razorpay or Stripe)** for secure transactions.
* Auto-generated **payment receipts and transaction history** for each user.
* SSL encryption and backend data hashing for transaction and account security.

**e) Reliable Real-Time Chat and Notifications**

* Chat system implemented using **WebSockets or third-party APIs** like Firebase for instant messaging.
* **Push notifications** and alerts for new messages, rent due dates, and booking updates.
* Ensures seamless communication and reduces miscommunication between parties.

**3. Technical Architecture Layers**

| **Layer** | **Description** |
| --- | --- |
| **Presentation Layer** | Frontend (React.js) UI for users to interact with the application. |
| **Application Layer** | Node.js and Express.js APIs to manage business logic and API routing. |
| **Data Layer** | MongoDB used for storing users, properties, transactions, and messages. |
| **Integration Layer** | External APIs such as Payment Gateway and Notification services. |

**4. Deployment and Scalability**

* **Containerization using Docker** for scalable and consistent deployment.
* Hosted on **cloud services** like AWS/GCP/Vercel for frontend and backend.
* **Horizontal scaling** to manage increasing number of users and requests.

**5. Security Considerations**

* **Input validation and sanitization** to prevent SQL/NoSQL injection.
* **HTTPS protocols** used throughout the app to ensure data privacy.
* **User sessions are securely stored and managed** using tokens and cookies.

**6. Future Enhancements**

* AI-based **property recommendation system** based on user behavior.
* Integration of **voice search or chatbot assistant** for an advanced experience.
* Blockchain-based **smart contracts** for automated agreement processing.

Let me know if you’d like this formatted into a Word document or enhanced with diagrams like system architecture, sequence diagrams, or DFDs.

**Solution Architecture Diagram:** 