**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 14 April 2025 |
| Team ID | SWTID1743354369 |
| Project Name | House Rent App Using MERN – House Hunt |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

### ****Solution Architecture Overview****

**Solution Architecture** is a strategic process that ensures your technology choices align effectively with business needs. For the HOUSE HUNT project, this approach helps deliver a scalable, secure, and user-friendly rental management system.

#### ****Purpose of Solution Architecture in HOUSE HUNT****

* **Identify Best Tech Solution:** Leverage the MERN stack (MongoDB, Express.js, React.js, Node.js) to build a responsive, real-time web application tailored to house rentals.
* **Communicate with Stakeholders:** Visualize and communicate how users interact with the system, what data is processed, and how the backend supports these operations.
* **Define Features & Phases:** Clearly structure the development timeline through sprints—starting from registration, login, and listing to admin panel and search filters.
* **Deliver Specifications:** Provide technical documentation including APIs, data models, authentication mechanisms, and UI/UX workflows.

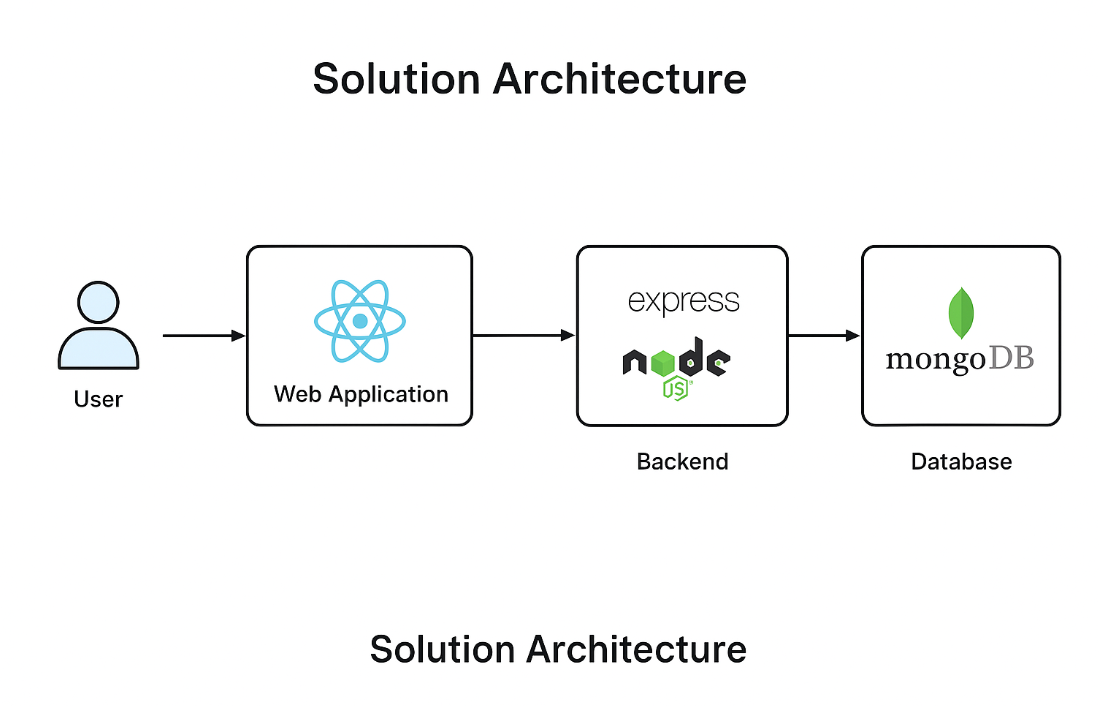
### ****Key Components of the Solution Architecture****

| **Component** | **Description** |
| --- | --- |
| **Frontend (React.js)** | Delivers a dynamic and responsive UI for tenants and landlords. |
| **Backend (Node.js + Express.js)** | Handles API routing, business logic, and user authentication. |
| **Database (MongoDB)** | Stores user data, property listings, bookings, chat history, etc. |
| **Authentication** | JWT-based secure login and signup for users (email, Gmail, Facebook). |
| **Admin Panel** | Enables admin to verify listings and manage reported properties. |
| **Search & Filters** | Allows users to find homes based on price, location, and preferences. |
| **Chat Feature** | Real-time chat between tenants and landlords (optional WebSocket integration). |
| **Deployment** | Hosted on cloud platforms (like Vercel/Netlify for frontend, and Render/Heroku for backend). |

### ****Development Phases****

1. **Sprint 1:** Registration, login, email verification.
2. **Sprint 2:** Property listing, homepage UI, integration with MongoDB.
3. **Sprint 3:** Advanced search filters, admin panel setup.
4. **Sprint 4:** Final integration, testing, chat feature (if feasible), and deployment.

**Example - Solution Architecture Diagram:**

****

*Figure 1: Architecture and data flow of the voice patient diary sample application*

**Reference:** [**https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/**](https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/)