**Ideation Phase**

**Brainstorm & Idea Prioritization**

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

**Brainstorm & Idea Prioritization :**

The ideation phase lays the foundation for the success of the HOUSE HUNT project. It involves gathering the right team, identifying a real-world problem to solve, listing down ideas to address it, and prioritizing features to deliver a functional and user-friendly solution within a limited timeframe.

**Goal of the Project**

Create a **House Rent Application (HOUSE HUNT)** using the **MERN Stack** (MongoDB, Express.js, React.js, Node.js) that allows users to:

* Search for rental properties based on filters (location, price, amenities, etc.)
* View detailed property information
* Post their property for rent (for owners)
* Contact property owners or agents
* Manage their account (sign up, login, update profile)

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

**Objective:**

To bring together individuals with diverse skill sets, align on a shared vision, and finalize a real-world problem to address using the MERN stack.

**Actions Taken:**

**Team Formation**: A balanced team was formed consisting of frontend developers, backend developers, and database designers—all passionate about solving real-life problems through web development.

**Skill Mapping**:

* Frontend: React.js, Bootstrap, HTML/CSS
* Backend: Node.js, Express.js
* Database: MongoDB

**Collaboration Tools Used**:

* Communication: WhatsApp, Google Meet
* Project Planning: Trello, Google Docs

**Problem Identification:**

After several rounds of discussion and idea pitching, the team unanimously agreed on solving a common issue: the difficulty in finding and renting houses conveniently and transparently.

**1. Team Formation and Role Distribution**

The foundation of any successful project lies in assembling a well-balanced and dedicated team. For our project titled **HOUSE HUNT – A House Rent Application Using MERN Stack**, we began by identifying team members with strong enthusiasm for web development, practical problem-solving, and interest in building a real-world application from scratch.

Our team comprises four committed members. Each member was assigned a specific role based on their individual skill set, domain knowledge, and preferences. This structured approach ensured a clear division of responsibilities, accountability, and efficient collaboration throughout the development lifecycle.

| **Team Member** | **Assigned Role** | **Key Responsibilities** |
| --- | --- | --- |
| Member 1 | **Project Lead & Full Stack Developer** | Coordinating project timelines, managing GitHub repository, integrating frontend and backend, deployment. |
| Member 2 | **Frontend Developer** | Designing responsive user interfaces using React.js, managing UI/UX flow, integrating API data, using Tailwind CSS for styling |
| Member 3 | **Backend Developer** | Developing RESTful APIs using Node.js and Express.js, implementing authentication and authorization (JWT), business logic |
| Member 4 | **Database Manager & QA Tester** | Designing MongoDB database schemas, managing data models, writing queries, performing manual testing and reporting bugs |

We adopted a collaborative model that allowed flexibility and learning opportunities. Members often cross-collaborated across modules to support one another, strengthening teamwork and overall output.

**2. Collaboration Tools and Workflow**

To ensure seamless communication and progress tracking, especially in a hybrid working environment, we utilized a range of collaboration tools. These helped us stay aligned with project goals, deadlines, and updates:

**Git & GitHub**: For version control, collaborative code reviews, branching strategies, and pull requests.

**Google Meet / Whatsapp** : For regular virtual meetings, daily/weekly stand-ups, and technical discussions.

**Google Drive / Docs**: For storing and sharing important project-related documents, reports, and references.

We followed a simplified agile approach with iterative development and continuous integration. Weekly check-ins helped us identify blockers early, track progress, and maintain momentum.

**3. Brainstorming and Problem Selection**

Our initial goal was to identify a real-world problem that could be solved effectively using the MERN stack. We conducted several brainstorming sessions where each team member proposed different ideas from areas such as:

* Healthcare management systems
* Online education platforms
* E-commerce stores for small businesses
* Rental property listing and management platforms
* Local job portals and community service apps

After evaluating these ideas based on feasibility, scope, relevance, and innovation, we collectively agreed upon the **Online House Rental Management System** as the most promising and impactful choice.

**4. Why We Chose This Problem**

The decision to work on a house rent application was driven by the following key factors:

**High Practical Relevance**: With increasing urban migration and digitization, there is a growing demand for online rental platforms, especially among students and working professionals relocating to new cities.

**Market Gap**: Most existing rental portals are either paid, limited in scope, or difficult to use for individual landlords. Our platform aims to bridge this gap by offering a simple, user-friendly, and accessible solution for both tenants and property owners.

**Technology Fit**: The problem is ideal for implementing CRUD operations, user authentication, RESTful API design, and MongoDB-based data management — perfectly aligning with the MERN stack.

**Scalability and Innovation Potential**: This system can be expanded in the future with features such as:

* Online payment integration (e.g., Razorpay, Stripe)
* Google Maps API for property location tracking
* Messaging/chat system between tenants and owners
* Admin analytics and property verification dashboard
* Mobile app version using React Native

**5. Final Problem Statement**

**"To design and develop a responsive, secure, and scalable web application using the MERN stack that enables house owners to list rental properties and allows tenants to search, filter, view, and inquire about properties, thereby simplifying the rental process and minimizing dependency on third-party agents."**

This problem not only allowed us to apply our technical knowledge in full-stack development but also inspired us to build something that could be expanded and deployed in a real-world setting.

**Step-2: Brainstorm, Idea Listing and Grouping**

Once the team was formed and the problem statement was finalized, the next crucial step in the project development process was to brainstorm different ideas related to the solution, list those ideas collaboratively, and group them into meaningful categories. This step helped us visualize the complete scope of the application, prioritize features, and streamline the development process.

**1. Purpose of the Brainstorming Session**

The goal of the brainstorming session was to:

* Understand what functionalities users would expect from a house rental platform.
* Identify all possible features that can enhance user experience and platform efficiency.
* Break down the problem into smaller manageable modules for better planning and development.
* Align everyone's vision regarding the project scope and deliverables.

**2. Brainstorming Methodology**

We conducted a series of brainstorming sessions over virtual meetings using tools like:

* **Miro** for real-time collaborative whiteboarding and mind mapping.
* **Google Docs** for live note-taking and idea capturing.
* **Trello** for organizing features into categories and prioritizing tasks.

Each team member contributed ideas based on:

* Personal experiences
* Existing popular rental apps (like NoBroker, Housing.com, MagicBricks, etc.)
* User expectations and pain points

We used a **Round-Robin format** where every team member was encouraged to present one or more ideas at a time. No idea was considered irrelevant or dismissed immediately — everything was listed for review.

**3. Idea Listing**

Below is the comprehensive list of ideas/features that emerged during the brainstorming phase:

**Core Features:**

* Property listing by owners
* Tenant registration/login
* Property search with filters (location, price, number of rooms)
* Property detail view with photos, amenities, and descriptions
* Owner contact or inquiry form
* User dashboard for both tenants and owners
* Property status (available/booked)

**Extended Features:**

* User reviews and ratings on properties
* Save/favorite properties for later
* Admin panel to manage listings and users
* Push notifications (e.g., new listings in user’s area)
* Email confirmation on inquiry or registration
* OTP-based login or multi-factor authentication

**Future Enhancement Ideas:**

* Online rent agreement generation
* Integrated payment gateway (Razorpay, Stripe)
* Chat system between tenant and owner
* Google Maps integration for viewing location
* Recommendation system based on user search history
* Mobile app version (React Native or Flutter)

**4. Grouping of Ideas**

After listing all possible ideas, we organized them into logical **feature groups** for better clarity and to ease project execution. This step was critical for defining the Minimum Viable Product (MVP) and planning development sprints.

| **Category** | **Features Grouped** |
| --- | --- |
| **User Management** | Registration, Login, Logout, Dashboard, Profile Management |
| **Property Listings** | Add Property, View Property, Edit/Delete Listings, Add Photos, Property Status |
| **Search & Filters** | Search by location, price range, bedrooms, availability |
| **Inquiries & Contact** | Inquiry form, Contact owner, Email confirmation, Inquiry history |
| **Admin Operations** | User monitoring, Property moderation, Feedback & review management |
| **Security** | JWT authentication, Password encryption, Role-based access |
| **Advanced Features** | Favorites, Reviews, Ratings, Notifications, Chat system (Future), Payment gateway |

**5. Key Takeaways from This Phase**

We successfully transformed a broad problem statement into well-defined, actionable modules.

Prioritized **core functionalities** that are essential for launch, and documented **advanced features** for future expansion.

Ensured everyone in the team had a shared understanding of what needs to be built and how we’ll divide it across upcoming development phases.

Established the foundation for **Requirement Analysis**, **System Architecture Design**, and **Sprint Planning** in the next phases.

**Step-3: Idea Prioritization**

After brainstorming and organizing a comprehensive list of ideas and features in Step 2, the next logical step in our development process was **prioritizing these ideas**. This stage was essential to identify which features to develop first (Minimum Viable Product – MVP), which ones to add later (Post-MVP or V2), and which ones to consider as long-term enhancements or stretch goals.

* Effective idea prioritization ensured that:
* The project remained manageable within the given timeline and resources.
* Core user needs were addressed from the start.
* The development process followed a clear, goal-driven roadmap.

**1. Prioritization Approach**

To determine what features should be prioritized for development, we adopted a combination of two popular techniques:

**a. MoSCoW Method**

We classified each idea into four categories:

**Must Have** – Critical features required for the app to function.

**Should Have** – Important features that enhance user experience but are not critical for MVP.

**Could Have** – Nice-to-have features that can be added if time/resources allow.

**Won’t Have (for now)** – Features we decided to postpone or not include in this version.

**b. Value vs. Effort Matrix**

Each feature was analyzed based on:

**Value** to the end-users (usability, necessity, impact)

**Effort** required to implement (time, complexity, team skill)

This allowed us to balance our work between quick wins, high-impact features, and manageable complexities.

**2. Prioritized Feature List (with Justifications)**

| **Feature** | **Priority** | **Justification** |
| --- | --- | --- |
| User Registration and Login | Must Have | Basic entry point for all users; essential for access control |
| JWT Authentication | Must Have | Secures APIs and protects user data |
| Owner Dashboard | Must Have | Allows property owners to add/manage listings |
| Add/View/Edit/Delete Property | Must Have | Core CRUD functionality; enables main feature of the platform |
| Property Detail Page | Must Have | Users must see full details before making inquiries |
| Search and Filters | Must Have | Helps users find relevant listings quickly |
| Inquiry/Contact Form | Must Have | Enables tenant-owner communication |
| Email Confirmation for Inquiry | Should Have | Adds professionalism and ensures communication reliability |
| Favorites (Save Property) | Should Have | Enhances user convenience |
| Admin Dashboard | Should Have | Helps in content moderation and managing abuse |
| Reviews and Ratings | Could Have | Adds value but can be added post-MVP |
| Chat System | Could Have | Requires WebSockets or Firebase; good for future versions |
| OTP Login / Multi-Factor Auth | Could Have | Adds extra security; useful in real-world scenarios |
| Online Rent Agreement | Won’t Have | Requires legal template integration; out of scope for current version |
| Payment Gateway Integration | Won’t Have | Complex to implement and test within current timeline |

**3. Final MVP Feature Set**

Based on the prioritization, the following features were **locked in as MVP (Minimum Viable Product)**:

* User Registration & Login (Tenant/Owner)
* JWT Authentication
* Owner Property Listing (Add/Edit/Delete)
* Property Search with Filters
* Property Details Page
* Inquiry Form (with optional email notification)
* Tenant Dashboard to view inquiries and saved properties

This MVP would allow us to deliver a fully functional product that solves the primary pain points of both property owners and tenants.

**4. Post-MVP Planning**

We also created a **Phase 2 backlog** to record "Should Have" and "Could Have" features, which could be implemented after the initial launch if time permits. This list was added to Trello for tracking and sprint planning purposes.

These features include:

* Admin Panel
* Save/Favorite Properties
* Google Maps Integration
* Property Reviews
* Chat Functionality

**5. Outcome of the Prioritization Phase**

We defined a **clear roadmap** that separates core from auxiliary features.

The team could now focus on the most **impactful and feasible deliverables**.

Helped manage scope and timeline, avoiding overengineering or feature creep.

Set the stage for **requirement analysis, UI wireframing, and architecture design**.