Phase 1&2 Project Documents

Ideation Phase
Define the Problem Statements

Date	22 June 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	3 Marks
Team Members	● Madhavv Viswanath K ● Mohnishwar D ● Hiruthick SM ● Ganesh S

Customer Problem Statement:

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Tenent/Renter	Rent a house posted on the site using mobile/laptop	It takes much time to load the available houses and Images	The site's server is slow and it takes time to fetch the image and load in the UI	Frustrated and May does not visit the site again
PS-2	Owner	To Post a House in the site for sale using phone/laptop by filling a Form and uploading Documents	Difficult to upload the necessary documents and doesn't support all image formats.	The website may only support specific file formats. If you are trying to upload a file format that	Confused and Looks for Alternative site if the issue persists

				is not supported, the upload will fail.	

Project Design Phase-I
Proposed Solution

Date	22 June 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	3 Marks

Proposed Solution:

Project team shall fill the following information in proposed solution template.

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	Develop a user-friendly house rental platform to streamline property searches, automate tenant verification, manage leases, ensure timely payments, and efficiently handle maintenance requests, enhancing the rental experience for all stakeholders.
2.	Idea / Solution description	Create an integrated rental platform featuring advanced search filters, automated tenant screening, digital lease management, secure online payment systems, and efficient maintenance request handling to simplify and enhance the rental process.
3.	Novelty / Uniqueness	Combines advanced search algorithms, automated tenant verification and seamless rental experience.

4.	Social Impact / Customer Satisfaction	1. Increasing Access to Housing: By providing a platform for landlords to list them properties, the app can increase the availability of rental housing options, especially in areas where housing is scarce. 2. Empowering Tenants: Tenants can use the app to find suitable housing that meets their needs and budget, giving them more control over their living arrangements.
5.	Business Model (Revenue Model)	● Listing Fees ● Subscription Plans ● Market Expansion
6.	Scalability of the Solution	The platform can easily scale to accommodate a growing number of users, properties, and locations, with robust cloud infrastructure ensuring reliable performance and efficient handling of increased demand and data.

Project Planning Phase
Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	22 June 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Authentication	USN-1.1	As an Owner or Renter, I can register for the application by	3	Low	Madhavv, Hiruthick

			entering my email, password, and confirming my password. In the Registration page			
Sprint-1		USN-1.2	Implement separate registration API’s for handling Owner and Renter Data	5	Medium	Monish, Ganesh
Sprint-1		USN-1.3	Integrate frontend with backend for registration	8	High	Monish, Ganesh
Sprint-1		USN-1.4	As an Owner or Renter, I can login from any device through Login page by entering my registered email, password	2	Low	Madhavv, Hiruthick
Sprint-2	Property Listing	USN-2.1	Renter Should be able to See the List of properties in the property Listing page	3	Low	Madhavv, Hiruthick
		USN-2.2	Implement property listing API	5	Medium	Madhavv, Hiruthick
		USN-2.3	Integrate frontend with backend for listing the properties posted by	8	High	Monish, Ganesh

			owners			
Sprint-3	Property Submission	USN-3.1	Owner Should be able to submit the Properties details in a form in the property Submission page	3	Low	Monish, Ganesh
		USN-3.2	Implement property submission API	5	Medium	Monish, Ganesh
		USN-3.3	Integrate frontend with backend for submission of property by Owner which will be posted in property listing page of Renters.	8	High	Madhavv, Hiruthick
Sprint-4	Property Details	USN-4.1	Renters Should be able to see the details of each listed property in The Details Page	3	Low	Madhavv, Hiruthick
		USN-4.2	Implement property details API by fetching the data posted by the Owners in the Property Submission Forms	5	Medium	Madhavv, Hiruthick
		USN-4.3	Integrate	8	High	Monish,

			frontend with backend for details			Ganesh
Sprint-5	Booking Property	USN-5.1	Renter should be able to request Booking by filling form details, in the Property Details page and, the form is sent to the property Owner	5	Medium	Monish, Ganesh
		USN-5.2	Implement booking API	5	Medium	Ganesh, Hiruthick
		USN-5.3	Integrate frontend with backend for booking status update	8	High	Monish, Madhavv
Sprint-6	Admin Dashboard	USN-6.1	Admin should be able to Handle Users, Owners and the Booking Status of Property	5	Medium	Monish, Ganesh
		USN-6.2	Implement admin monitoring API	8	High	Madhavv, Hiruthick
		USN-6.3	Integrate frontend with backend for monitoring tools	8	High	Madhavv, Hiruthick
Sprint-7	Property Management	USN-7.1	Owner should be able to Manage	5	Medium	Ganesh, Hiruthick

			Property by approving or rejecting the Renter and updating the Booking Status.			
		USN-7.2	Implement property CRUD API	8	High	Ganesh, Hiruthick
		USN-7.3	Integrate frontend with backend for property CRUD	8	High	Monish, Madhavv

Project Tracker, Velocity & Burndown Chart: (2 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	3 Days	5 Jul 2024	7 Jul 2024	10	8 Jul 2023
Sprint-2	16	3 Days	7 Jul 2024	10 Jul 2024		
Sprint-3	16	5 Days	5 Jul 2024	10 Jul 2024		
Sprint-4	16	6 Days	5 Jul 2024	11 Jul 2024		
Sprint-5	18	7Days	5 Jul 2024	12 Jul 2024		
Sprint-6	19	7 Days	5 Jul 2024	12Jul 2024		
Sprint-7	21	7 Days	5 Jul 2024	12 Jul 2024		

Velocity:

Sprint	**Average Velocity (story points/Duration)**
Sprint-1	**18/3=6**
Sprint-2	**16/3=5.33**
Sprint-3	**16/5=3.2**
Sprint-4	**16/6=2.66**
Sprint-5	**18/7=2.57**
Sprint-6	**19/7=2.71**
Sprint-7	**19/7=2.71**

Requirement Gathering and Analysis Phase
Solution Requirements (Functional & Non-functional)

Date	06 July 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	**Functional Requirement**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	● Registration through Form

FR-2	Property Listing	● Type of Property ● Price and Availability ● Location of Property
FR-3	Property Submission	● Registration of Owner Details through Form ● Property Details and Proof Document ● Uploading Property Images and Features
FR-4	Property Details	● Detailed Description of Property ● Image Carousel ● Status of Property (Booked or Requested)
FR-5	Booking Property	● Requesting Property through Form ● Filling The User Details to Request to Owner ● Cancel Booking
FR-6	Admin Dashboard	● Handle The User and Owner List ● Approve The Owner to post Property in the site ● Monitor the Property Booking Status
FR-7	Property Management	● Reject Rental Request by User ● Approve Rental and Change Status by Owner ● Add Property to the site by Owner ● Delete the posted Property ● Update the Property Details

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

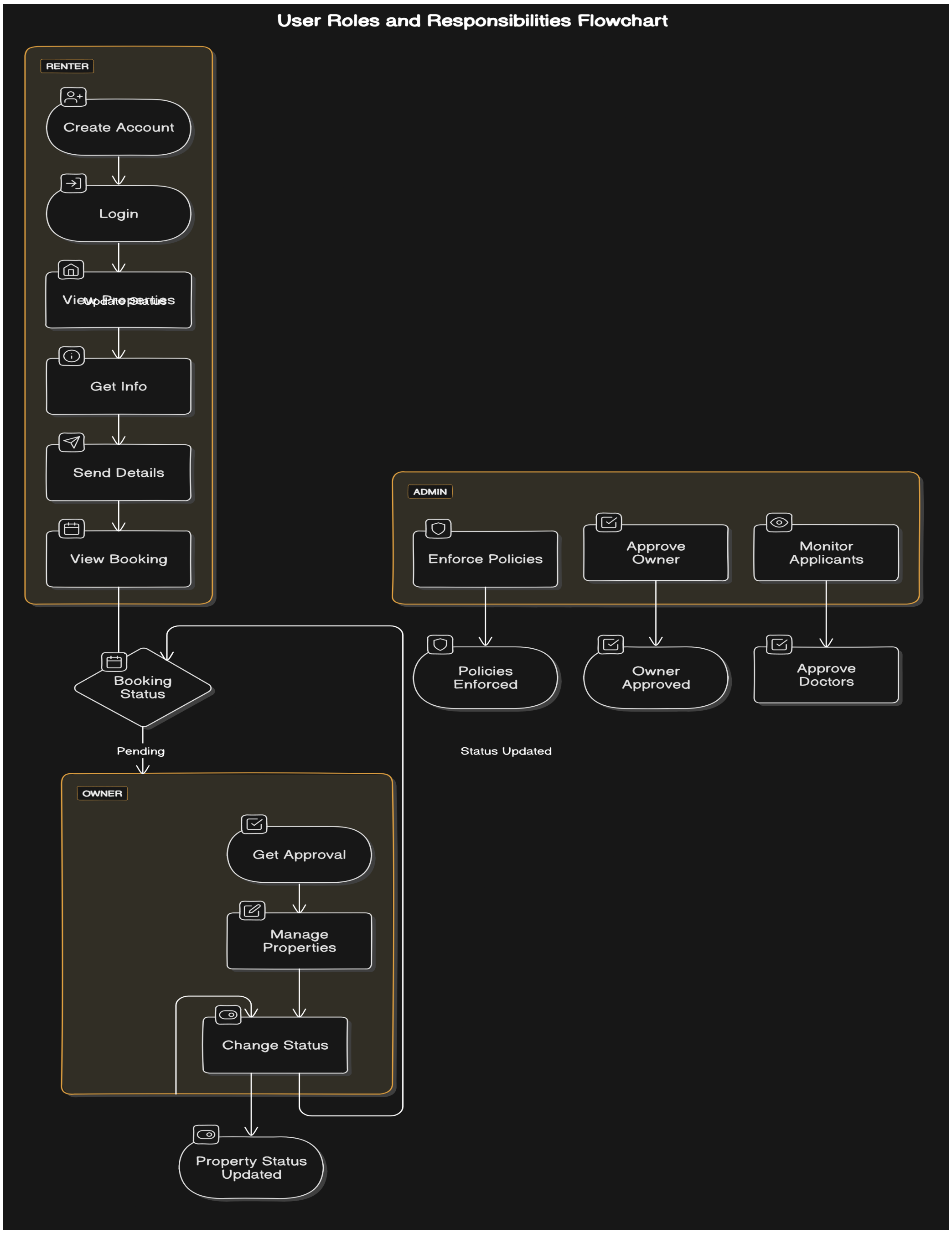
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	● The user interface should be intuitive and easy to navigate for users of all technical backgrounds. ● The application should provide a clear and responsive List of Property Cards for users of all kinds of device. ● The property viewing and renting experience should be easy and user-

		friendly
NFR-2	**Security**	● The application must implement secure user authentication and authorization mechanisms. ● All the Booking and User Details filling must be secure, and the API request and response data must be encrypted. ● The app must run in https protocol and be hosted in a genuine hosting site.
NFR-3	**Reliability**	● The application should be highly reliable with minimal downtime or disruptions.
NFR-4	**Performance**	● The application should provide high quality images of the property and the booking status must reflect changes without any delay ● It must be responsive in all devices with all sizes.
NFR-5	**Availability**	● The application should be available to users 24/7 with minimal downtime for maintenance. ● The application should be able to handle high volumes of concurrent users without performance degradation.
NFR-6	**Scalability**	● The platform must easily scale to accommodate a growing number of users, properties, and locations. ● With robust cloud infrastructure it must ensure reliable performance and efficient handling of increased demand and data

Requirement Gathering and Analysis Phase
Solution Architecture

Date	06 July 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	

Solution Architecture:



EXPLANATION:

1. Renter/Tenent:
    - Create an account and log in to the system using their email and password.
    - They will be shown automatically all the properties in their dashboard.
    - After clicking on the Get Info, all the information of the property and owner will come, and small form will generate in which the renter needs to send his\her details.
    - After that they can see their booking in booking section where the status of booking will be showing “pending”. It will be change by owner of the property.

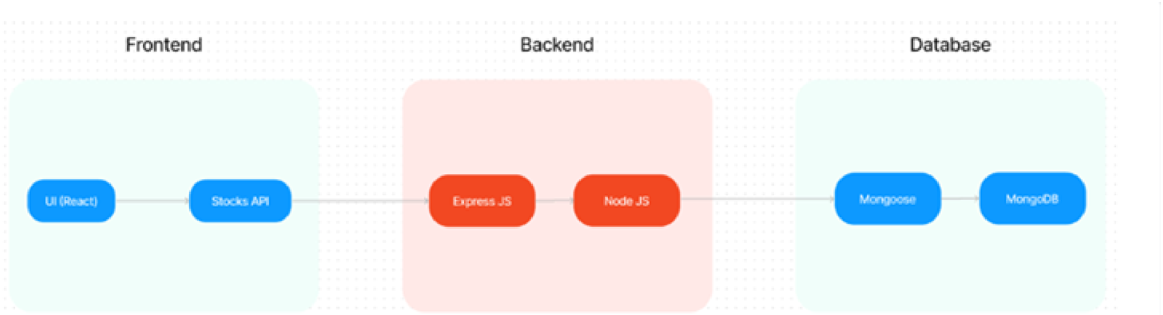
2. **Admin:**
    - He/she can approve the user as “owner” for the legit user to add properties in his app
    - He monitors the applicant of all doctors and approve them and then doctors are registered in the app.
    - Implement and enforce platform policies, terms of service, and privacy regulations.

3. **Owner:**
    - Gets the approval from the admin for his Owner account.
    - After approval, he/she can do all CRUD operation of the property in his/her account
    - He/she can change the status and availability of the property.

Requirement Gathering and Analysis Phase
Technology Stack (Architecture & Stack)

Date	06 July 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	

Technical Architecture



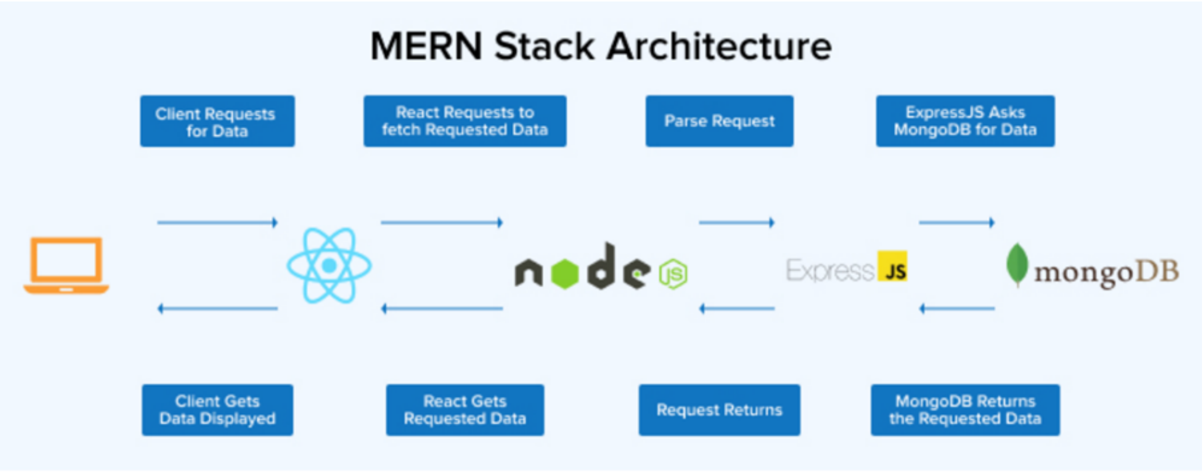

MERN Stack Architecture
Client Requests for Data
React Requests to fetch Requested Data
Parse Request
ExpressJS Asks MongoDB for Data
node JS
Express JS
mongoDB
Client Gets Data Displayed
React Gets Requested Data
Request Returns
MongoDB Returns the Requested Data

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	How users interact with the application (Web UI, Mobile)	HTML, CSS, JavaScript, React.js
2	Application Logic-1	Logic for user authentication and authorization	Node.js, Express.js

3	Application Logic-2	Logic for property management (CRUD operations)	Node.js, Express.js
4	Application Logic-3	Logic for booking management (request and status update)	Node.js, Express.js
5	Database	Data storage and configurations	MongoDB, Mongoose
6	Cloud Database	Database service on cloud	MongoDB Atlas
7	File Storage	File storage requirements (property images, documents)	Local Filesystem
8	Infrastructure (Server/Cloud)	Application deployment on local system/cloud	Local

Table-2: Application Characteristics:

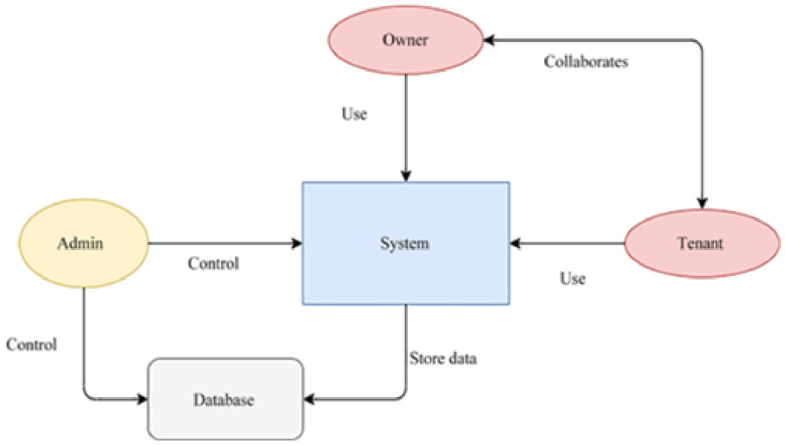
S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used	React.js, Node.js, Express.js, Mongoose
2	Security Implementations	List all the security/access controls implemented, use of firewalls, etc.	HTTPS, BSON
3	Scalable Architecture	Justify the scalability of architecture (3-tier,Client-Server)	MERN

Requirement Gathering and Analysis Phase
Data Flow Diagram & User Stories

Date	06 July 2024
Team ID	SWTID1720019632
Project Name	HOUSE RENT APP USING MERN
Maximum Marks	

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Renter, Owner	User Authentication	USN-1.1	As an Owner or Renter, I can register for the	Form fields: email, password, confirm password, role	Low	Sprint-1

			application by entering my email, password, and confirming my password. In the Registration page	selection (renter/owner). Successful registration should navigate to dashboard.		
Renter, Owner		USN-1.2	Implement separate registration API's for handling Owner and Renter Data	API should create user in the database, handle errors, and return success/failure response	Medium	Sprint-1
Renter, Owner		USN-1.3	Integrate frontend with backend for registration	Successful integration should allow new users to register and log in immediately.	High	Sprint-1
Renter, Owner		USN-1.4	As an Owner or Renter, I can login from any device through Login page by entering my registered email, password	API should authenticate user, handle errors, and return user-specific token	Low	Sprint-1

Renter	Property Listing	USN-2.1	Renter Should be able to See the List of properties in the property Listing page	Page should display list of available properties with summary information	Low	Sprint-2
Renter		USN-2.2	Implement property listing API	API should retrieve list of properties from the database, handle errors, and return properties	Medium	Sprint-2
Renter		USN-2.3	Integrate frontend with backend for listing the properties posted by owners	Should display properties on the listing page posted by owners	High	Sprint-2
Owner	Property Submission	USN-3.1	Owner Should be able to submit the Properties details in a form in the property Submission page	Form should contain property name, description, address, price, availability. Successful submission should add property to the database	Low	Sprint-3
Owner		USN-3.2	Implement property submission	API should add new property to the	Medium	Sprint-3

			API	database, handle errors, and return success/failure response.		
Owner		USN-3.3	Integrate frontend with backend for submission of property by Owner which will be posted in property listing page of Renters.	On Successful integration should allow owners to submit new properties.	High	Sprint-3
Renter	Property Details	USN-4.1	Renters Should be able to see the details of each listed property in The Details Page	Page should display detailed information about a selected property, including owner contact form.	Low	Sprint-4
Renter		USN-4.2	Implement property details API by fetching the data posted by the Owners in the Property Submission Forms	API should retrieve detailed information of a specific property from the database	Medium	Sprint-4

Renter		USN-4.3	Integrate frontend with backend for details	Successful integration should display detailed property information on the details	High	Sprint-4
Renter	Booking Property	USN-5.1	Renter should be able to request Booking by filling form details, in the Property Details page and, the form is sent to the property Owner	Forms with renter details, booking dates. Successful submission should create a booking request with status "pending".	Medium	Sprint-5
Renter		USN-5.2	Implement booking API	API should create a new booking in the database, handle errors, and return success/failure response.	Medium	Sprint-5
Renter		USN-5.3	Integrate frontend with backend for booking status update	Successful integration should allow renters to submit booking	High	Sprint-5

Admin	Admin Dashboard	USN-6.1	Admin should be able to Handle Users, Owners and the Booking Status of Property	Dashboard should display pending user approval requests with approve/reject buttons.	Medium	Sprint-6
Admin		USN-6.2	Implement admin monitoring API	API should update user approval status in the database	High	Sprint-6
Admin		USN-6.3	Integrate frontend with backend for monitoring tools	successful integration should allow admin to approve/reject user requests.	High	Sprint-6
Owner	Property Management	USN-7.1	Owner should be able to Manage Property by approving or rejecting the Renter and updating the Booking Status.	Owners should be able to create, read, update, and delete properties.	Medium	Sprint-7
Owner		USN-7.2	Implement property CRUD API	API should handle property CRUD operations.	High	Sprint-7

Owner		USN-7.3	Integrate frontend with backend for property CRUD	Successful integration should allow owners to manage their properties.	High	Sprint-7