ROOM BRIDGE

A Room Finder with Chat & AI-based Ranking

TEAM MEMBERS

- AKSHAT JAIN (00115607223)
- KHUSHAL (00715607223)
- PANKAJ SINGH (04315602722)
- RAGHAV DWIVEDI (01015602722)

PROBLEM STATEMENT

- Difficulty in finding suitable roommates and rental spaces
- Lack of verification, personalization, and safety
- Tenants face compatibility issues
- Landlords struggle to find genuine tenants
- Current platforms focus only on availability

WHY THIS TOPIC?

- Rising demand for trusted roommate and rental platforms
- Increased student and professional migration to urban areas
- Need for secure, verified systems to avoid fraud
- AI-driven personalization and compatibility recommendations
- Improved landlord–tenant communication

OBJECTIVES

- Build a secure and verified roommate & rental platform
- Integrate AI-based ranking & compatibility recommendations
- Provide mutual-consent chat system
- Enable landlord–tenant communication & review system
- Ensure admin-level document verification

SCOPE

- Students and professionals in urban and semi-urban areas
- Landlords and tenants with transparent communication
- Verified users through admin-level document verification
- AI-enhanced experience with ranked suggestions

PRELIMINARY ANALYSIS

- Problem: Safety, unverified users, lack of personalization
- Solution: Unified platform with verification, AI-ranking, privacy chats
- Costs: Development, hosting, admin verification
- Benefits: Increased trust, better matches, reduced fraud
- Recommendations: Hybrid backend, AI/ML models, encrypted chat

METHODOLOGY

- User Registration & Verification
- Room/Roommate Listing
- AI Recommendation Engine
- Mutual Consent Chat
- Landlord–Tenant Interaction

HARDWARE & SOFTWARE REQUIREMENTS

- Hardware: Intel i5/Ryzen 5+, 8–16 GB RAM, 500 GB SSD
- Backend: Spring Boot, Node.js, MySQL
- Frontend: React.js (Vite), TailwindCSS/Material UI
- AI/ML: Python, scikit-learn, TensorFlow/PyTorch
- Tools: GitHub, Docker, Postman

FRONTEND (REACT)

- We built the frontend using React for its fast rendering, reusable components, and responsive UI, making it ideal for a smooth user experience in room listing, profile verification, and rent management.

BACKEND (SPRING BOOT, NODE.JS)

- We chose Spring Boot for handling core business logic, authentication, and secure APIs due to its robustness and scalability. Node.js powers the real-time chat functionality, ensuring low-latency communication between tenants and landlords.

DATABASE (MYSQL)

- We used MySQL for structured data storage such as user profiles, room details, and transactions because of its reliability, ACID compliance, and ease of integration with Spring Boot.

AI INTEGRATION

- We integrated an ML model that ranks properties by combining star ratings with sentiment analysis of user comments. This ensures listings are ordered not just by static factors but by real tenant feedback, making recommendations more accurate, trustworthy, and user-centric.

LIMITATIONS

- Manual verification may slow onboarding
- AI accuracy depends on data availability
- Scaling requires significant server resources

TESTING TECHNOLOGIES

- Unit Testing: JUnit, Mocha/Chai
- Integration Testing: Postman, REST API
- Database Testing: MySQL Workbench
- Performance Testing: JMeter
- Security Testing: JWT authentication, penetration testing
- User Acceptance Testing with pilot group

CONTRIBUTION OF THE PROJECT

- Trustworthy platform with verified users
- AI-driven personalization and compatibility checks
- Secure, consent-based communication
- Direct landlord–tenant reviews and interaction
- Reduced fraud and improved roommate satisfaction

CONCLUSION & VISION

- Room Bridge = AI-enhanced housing & roommate ecosystem
- Ensures safety, trust, and convenience
- Bridges gap between landlords and tenants
- Aims to expand from urban pilots to national scale

THANKYOU