## 1. Scope, Objectives, Approach, and Expected Impact

### Scope

Habitat-AI is a hyper-local audit and transparency platform designed exclusively for students residing in Mathikere, Gokula, and New BEL Road (MSRIT vicinity).

- **In-Scope:** The platform focuses on auditing accommodation safety, verifying food hygiene in messes/hotels, and utilizing AI to summarize qualitative reviews.
- **Out-of-Scope:** The system explicitly excludes booking features, rent processing, or real-estate brokerage services to maintain neutrality as an audit tool.

### Objectives

- **Establish a Trusted Audit System:** To create a student-driven, data-verified repository for PG and Mess standards.
- **Eliminate Information Asymmetry:** To combat misleading advertisements and fake photos by prioritizing peer-verified content.
- **Enhance Decision Efficiency:** To replace the need for reading hundreds of long reviews with instant, AI-generated "Pros & Cons" summaries.
- **Drive Accountability:** To force service providers to maintain hygiene and safety standards through visible, public performance metrics.

### Approach

Our solution utilizes the MERN Stack (MongoDB, Express, React, Node.js) integrated with Generative AI to deliver three core modules:

1. **Accommodation Audit Module:** A structured rating system for PGs where users verify specific amenities (Wi-Fi speed, Water backup, Security) rather than just giving generic star ratings.
2. **Mess Module:** A visual-first feed where students upload photos of meals and kitchen conditions.
3. **AI-Powered Sentiment Summarization:** The backend integrates the **Google Gemini API** to process unstructured text reviews. It filters out emotional noise and extracts actionable insights (e.g., "Consistently salty food," "Rude warden"), presenting them as a concise summary.

### Expected Impact

- **Student Welfare:** Significant reduction in health risks (food poisoning) and financial loss (deposit scams) due to informed decision-making.
- **Market Correction:** Service providers (PG/Mess owners) will be compelled to improve infrastructure and hygiene to avoid negative AI summaries, creating a competitive, quality-driven market.
- **Community Data:** Long-term aggregation of data on "unsafe zones" (dark streets, waterlogging) that can be shared with local authorities for urban maintenance.

## 2. Team Roles & Responsibilities

- **Project Lead & Full Stack Architect:**
    - Defining the system architecture and database schema.
    - Managing the GitHub repository and integrating Frontend with Backend.
- **Backend & AI Developer:**
    - Developing RESTful APIs using **Node.js** and **Express**.
    - Integrating the **Google Gemini API** for the review summarization logic.
    - Managing database connections with **MongoDB Atlas**.
- **Frontend Developer (UI/UX):**
    - Building the responsive client interface using **React.js** and **Tailwind CSS**.
    - Designing intuitive "Audit Forms" for uploading reviews and photos.
- **Data & Quality Assurance Specialist:**
    - Collecting initial "seed data" (real reviews from MSRIT students) to populate the demo.
    - Implementing content moderation guidelines to prevent spam.
    - Testing the application across desktop and mobile.
    - Ensuring the project aligns with the "Urban Maintenance" and "Public Services" themes.

## 3. Requirements / Resources Needed

### Technical Requirements

- **Frontend:** React.js (Vite), Tailwind CSS.
- **Backend:** Node.js, Express.js.
- **Database:** MongoDB Atlas (NoSQL) for storing user profiles, PG details, and reviews.
- **AI/ML Integration: Google Gemini API** for Natural Language Processing.
- **Cloud Storage: Cloudinary** for hosting user-uploaded images (Mess food/Room conditions).
- **Version Control:** Git & GitHub.

### Non-Technical Resources

- **Dataset:** Initial collection of details for 5-10 popular PGs and Messes in Mathikere for the prototype.
- **Access:** Participation of MSRIT students for beta testing and feedback.

## 4. Sources / References

### Technical Documentation

1. **Google AI Studio Docs:** For implementing the Gemini API and prompt engineering for summarization. (*ai.google.dev*)
2. **React.js Documentation:** For component lifecycle and state management. (*react.dev*)
3. **MongoDB Atlas Documentation:** For schema design and cloud database deployment. (*mongodb.com/docs*)

### Contextual References

4.**Local Problem Statement**: Informal surveys conducted with MSRIT hostel/PG students regarding food safety and rent transparency.

5.**Benchmarking:** Analysis of existing platforms (e.g., Stanza Living, Zomato) to identify gaps in the unorganized, hyper-local market sector.