

Project Title:
Governmental Housing System

Team Members:

Name	ID	Program
Basmala Amr	202202213	SWD
Basmala Alaa	202201138	SWD
Basel Ashraf	202201869	DSAI

Supervisor:
Dr.Yousry AbdulAzeem

Semester/Year:
FALL/2025

Team Number:#57
GitHub Repository Link:

<https://github.com/Basmala27/Housing-system-graduation-project.git>

Initial System Design & Work Division

1. Project Summary & Scope

1.1 Project Summary

The Online Government Housing System is a centralized digital platform for streamlining government housing applications in Egypt.

Replaces manual and fragmented procedures with an integrated online solution for better accessibility, transparency, and efficiency.

Citizen features:

- Account creation and identity verification.
- Browse and filter housing projects.
- Submit applications and upload documents.
- Track application status in real time.
- Receive automated notifications for missing documents, updates, and final decisions.

Administrator features:

- Manage housing projects.
- Review applications and validate documents.
- Update application statuses.
- Monitor demand via an analytics dashboard.

1.2 Scope

- User registration, login, and role-based access (Citizen/Admin).
- Browsing and filtering housing projects.
- Full online application workflow:
 - Form submission
 - Document upload & secure storage
 - Input/file validation

- Real-time application tracking (Pending, Under Review, Accepted, Rejected)
- Notification system (status updates, missing documents alerts)
- Admin dashboard for project management and application review
- Basic analytics (project stats, demand patterns)
- AI recommendation engine (content-based filtering)
- Chatbot for FAQs and user guidance
- Basic audit logs and Help Center

Out of Scope:

- Online payments
- Integration with real government databases
- OCR or advanced automated document validation
- Advanced cybersecurity beyond standard practices

1.3 Progress Since Proposal

- Overall Progress: 40% completed since Week 4, including requirement refinement, system design, repo setup, and initial frontend/backend implementation.

1. Research & Requirement Refinement:

- Analyzed current government housing workflow in Egypt
- Identified citizen/admin pain points
- Finalized functional & non-functional requirements
- Developed complete user journeys for both roles

2. System Design & UI/UX:

- Created high-level system architecture
- Designed Figma wireframes for login, project listing, application submission, admin dashboard
- Built preliminary ERD (Users, Projects, Units, Applications, Documents, Notifications, Logs)

3. Repository & Environment Setup:

- Organized GitHub repo (frontend, backend, docs)

- Configured GitHub Projects board
- Initial commits:
 - React structure with routing
 - Node.js/Express base setup
 - MySQL configuration
 - Authentication & project browsing API skeleton

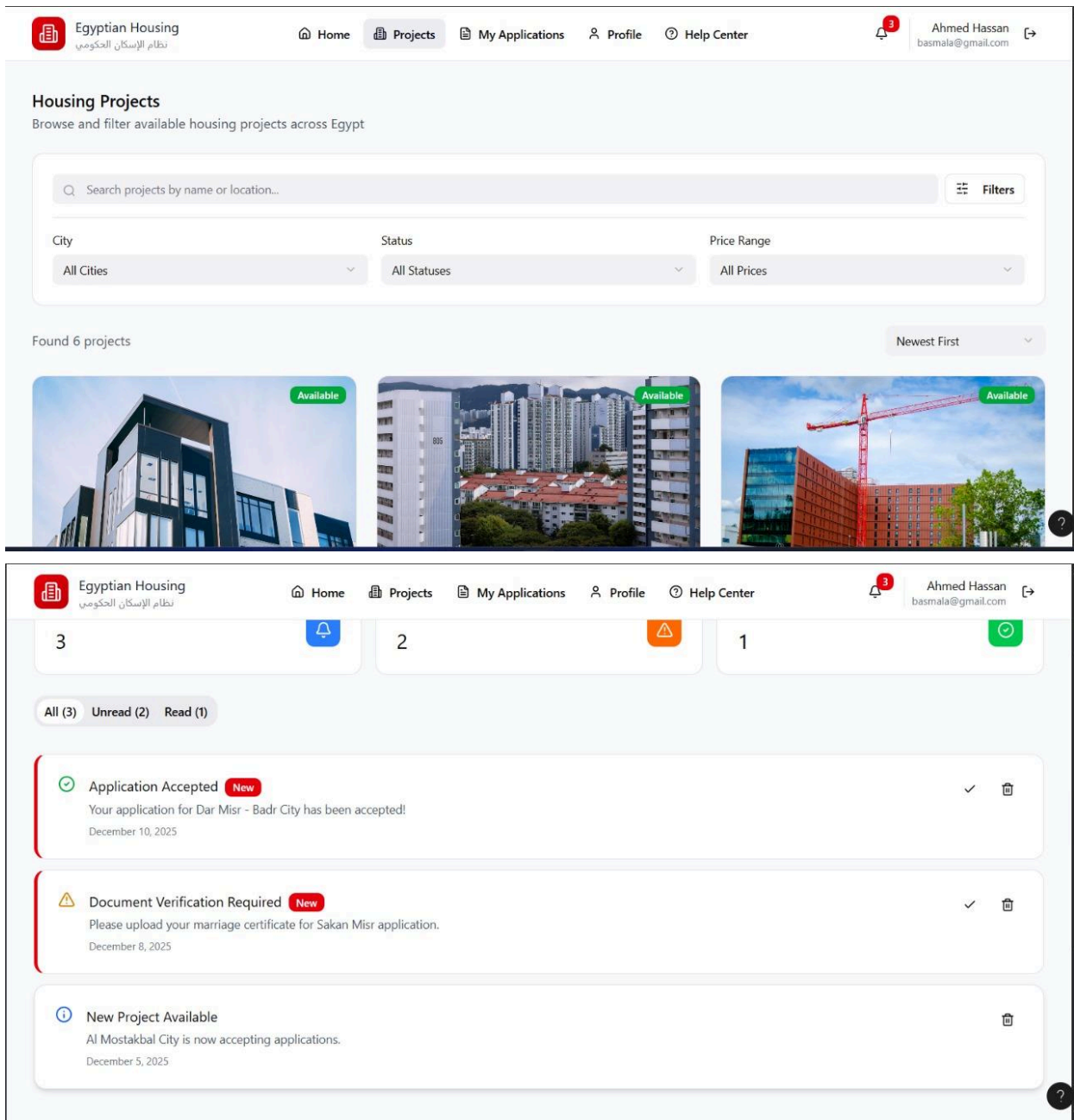
4. Early Implementation Progress:

- Frontend ([React.js](#)):
 - Project structure (components, pages, services)
 - Login & Register pages with validation
 - Housing Projects List + Details pages
 - React Router setup & Axios API integration
- Database:
 - Core tables: Users, Projects, Units, Applications
 - Linked applications with users and units
 - Documents and Notifications tables added

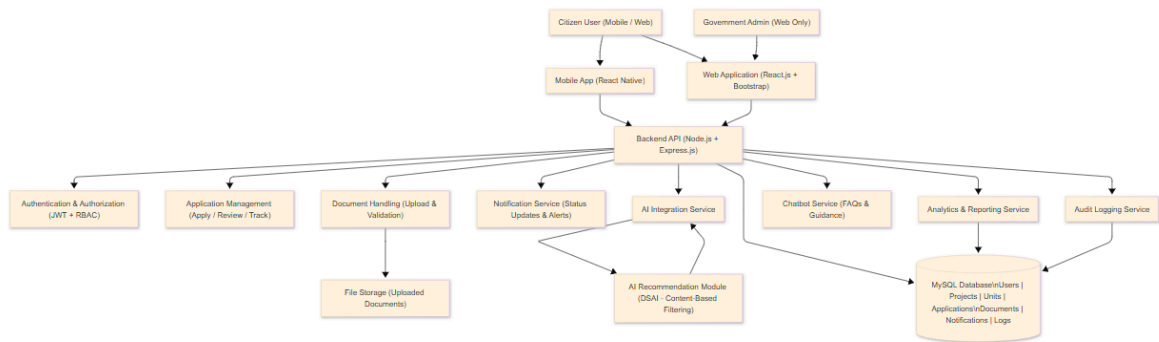
5. Documentation & Reporting:

- Updated SRS
- Initial test cases for authentication & project browsing

Figma Prototype:



2. System Architecture



2. Component Breakdown

1. Citizen User Interface (Mobile & Web)

Functionality: Enables citizens to register, browse housing projects, submit applications, upload documents, and track application status.

Inputs / Outputs: User credentials and application data → application status, notifications, and recommendations.

Technology Stack:

- Flutter (Mobile) for cross-platform development.
- React.js (Web) for scalable and interactive user interfaces.

2. Government Admin Interface (Web)

Functionality: Allows administrators to manage projects, review applications, verify documents, and generate reports.

Inputs / Outputs: Admin actions → approval/rejection decisions and reports.

Technology Stack:

- React.js with Bootstrap for responsive administrative dashboards.
-

3. Backend API Layer

Functionality: Central layer that processes client requests, enforces business rules, and coordinates communication between services.

Inputs / Outputs: HTTP requests → JSON responses.

Technology Stack:

- Node.js with Express.js for scalable RESTful APIs.
-

4. Authentication & Authorization Service

Functionality: Ensures secure access control for citizens and administrators.

Inputs / Outputs: Credentials → JWT tokens and access decisions.

Technology Stack:

- JWT for stateless authentication.
 - Role-Based Access Control (RBAC).
-

5. Application Management Service

Functionality: Manages the full lifecycle of housing applications, including submission, review, and status updates.

Technology Stack:

- Node.js services with database integration.
-

6. Document Handling Service

Functionality: Handles secure upload, validation, and storage of user documents.

Technology Stack:

- Multer for file uploads.
 - Secure file storage system.
-

7. Notification Service

Functionality: Notifies users about application progress and system events.

Technology Stack:

- Event-driven Node.js service.
 - Email/SMS gateways.
-

8. AI Integration & Recommendation Module

Functionality: Provides personalized housing recommendations using content-based filtering.

Technology Stack:

- Python-based ML services.
 - Scikit-learn for recommendation algorithms.
-

9. Chatbot Service

Functionality: Offers automated guidance and answers to frequently asked questions.

Technology Stack:

- NLP frameworks integrated via REST APIs.
-

10. Analytics & Reporting Service

Functionality: Generates system usage statistics and administrative reports.

Technology Stack:

- SQL-based analytics.
- Data visualization tools.

11. Audit Logging Service

Functionality: Records all critical user and system actions for compliance and security.

Technology Stack:

- Centralized logging frameworks.
-

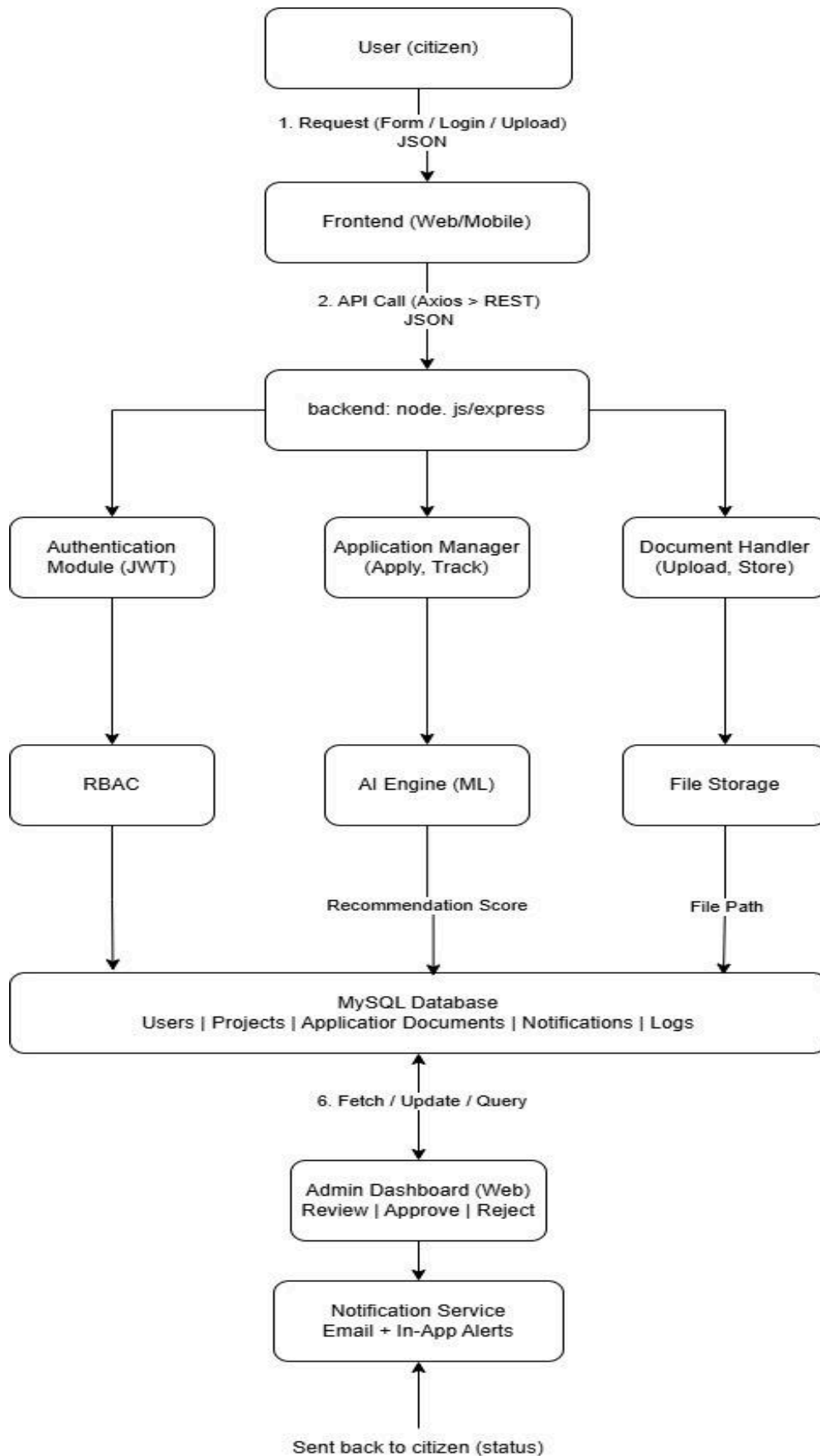
12. Database Layer (MySQL)

Functionality: Stores persistent data such as users, applications, documents, and logs.

Technology Stack:

- MySQL for reliable relational data management.
-

4.1 Data Flow(SWD):



Data flow contain :

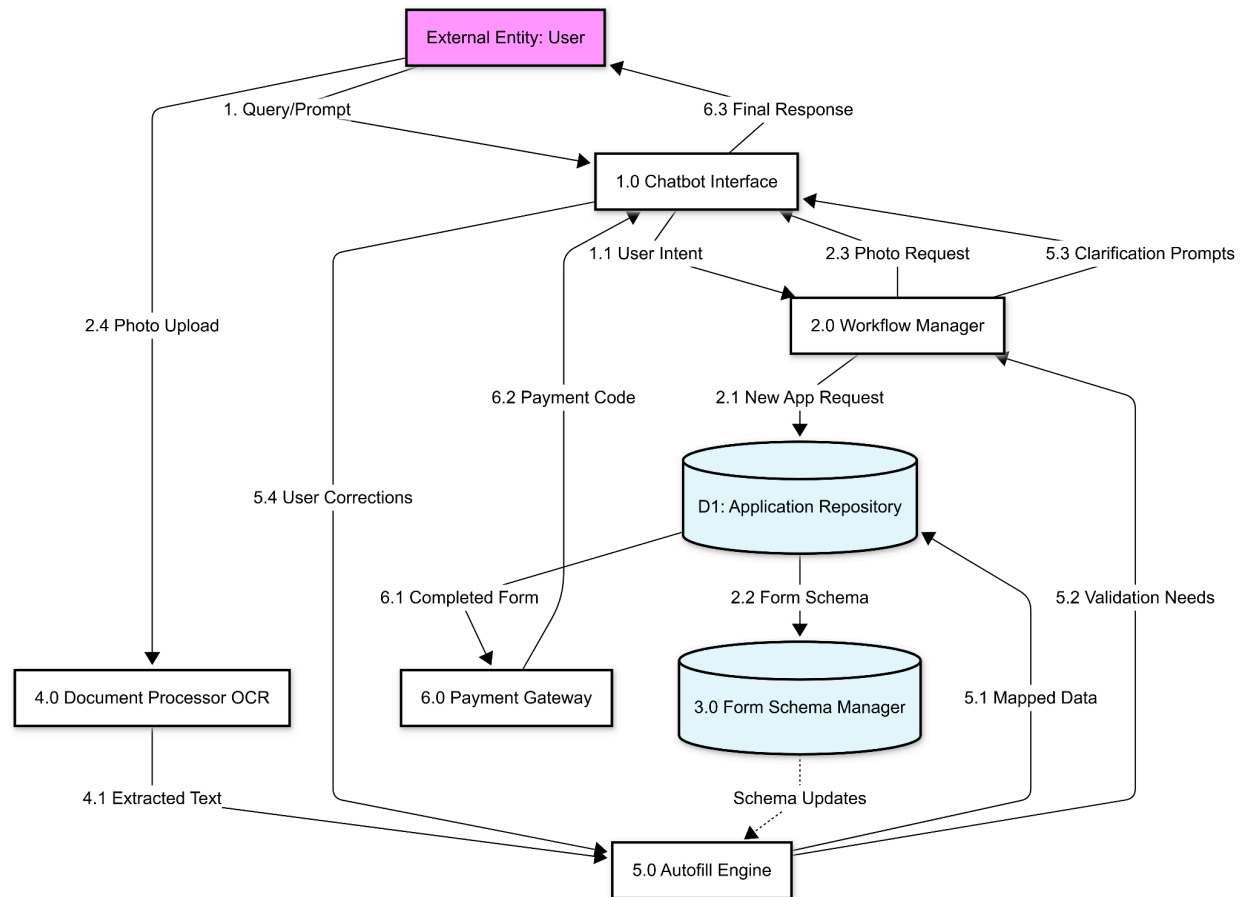
The user sends a request from the web or mobile app, and the frontend forwards it to the backend through REST APIs. The backend routes the request to the correct module (authentication, application handling, or document upload).

Data is then validated and stored or fetched from the MySQL database.

The admin dashboard reads the same data to review and update applications.

Any change triggers the notification service, which sends updates back to the user.

4.2 Data Flow(DSAD):



Data flow contain :

The user initiates a conversation via the chatbot on a web or mobile app, where the interface captures prompts and documents photos before forwarding them to the backend via WebSocket or REST APIs. The backend routes the request to the appropriate module (workflow orchestration, OCR processing, or autofill engine), extracting and mapping data from uploads. Validated information is then stored or retrieved from the MySQL database, maintaining application sessions and form states. The admin dashboard accesses this data to monitor, review, and approve applications in real-time. Any status update or completion triggers the notification service, delivering payment codes and confirmations directly to the user through the chatbot.

5. Work Breakdown Structure (WBS)

Task	Responsible Team Member	Start Date	End Date	Status
Requirement Gathering & Stakeholder Interviews	Basmala Amr, Basmala Alaa, Basel	Week 3 October	Week 4 October	Done
User Personas Creation (Citizen, Admin, Officer)	Basmala Alaa	Week 5	Week 6	Done
Functional & Non-Functional Requirements Documentation	Basmala Amr	Week 5	Week 6	Done
AI Model Data Collection	Basel	Week 5	Week 6	Done
Data Cleaning & Feature Engineering	Basel	Week 5	Week 6	Done
System Architecture Diagram Design	Basmala Alaa	Week 6	Week 7	Done
Data Flow Diagram Creation	Basmala Amr	Week 6	Week 7	Done
Database Schema Design (Citizens, Applications, Units)	Basmala Amr	Week 6	Week 7	Done
Midterm	All Team Member	Week 8	Week 9	Done
Website UI/UX Design(Figma)	Basmala Alaa	Week 9	Week 10	Done
Backend API Design/ Authentication	Basmala Amr, Basmala Alaa	Week 9	Week 12	In progress
Backend API Design –	Basmala Amr	Week 12	Week 13	In progress

Application Submission & Status				
Priority Scoring Model Development (AI / Rules)	Basel	Week 9	Week 13	In progress
Frontend UI/UX Design (Portal + Dashboard)	Basmala Alaa Basmala Amr	Week 10	Week 13	In progress
Frontend Implementation (Forms, Authentication, Dashboard)	Basmala Alaa	Week 13	Week 15	Not Started
Spring				
Backend Development – Admin Functions (Approval, Review)	Basmala Amr, Basmala Alaa	Week 3	Week 5	Not Started
Integration of AI Model with Backend	Basel	Week 4	Week 5	Not Started
Security Layer Implementation (Encryption, Auth, Roles)	Basel	Week 6	Week 8	Not Started
UnitTesting(Fronted + Backend)	Basmala Amr ,Basmala Alaa	Week 7	Week 8	Not Started
Testing – Integration Tests	Basmala Amr ,Basmala Alaa	Week 8	Week 9	Not Started
Performance and Load Testing	Basmala Alaa,Basmala Amr,Basel	Week 9	Week 10	Not Started
Deployment (Server setup, DB migration, Hosting)	Basmala Alaa ,Basmala Amr ,basel	Week 9	Week 11	Not Started

Final Documentation & Presentation Preparation	Basel ,Basmala Amr ,Basmala Alaa	Week 13	Week 14	Not Started
--	--	---------	---------	-------------

6. Teamwork and Equity

6.1 Teamwork Approach

Distributed Collaboration

Tasks are split between frontend, backend, AI, and documentation teams, with clear interfaces to avoid conflicts.

Weekly Coordination Meetings

Teams share progress, discuss blockers, and update priorities.

Version Control and Shared Repositories

Ensures consistent code updates and equal contribution visibility.

6.2 Equity Principles

Equal Contribution and Accountability

Each member receives tasks based on skill level but with balanced workload.

Transparent Task Assignment

Everyone can see who is responsible for what.

Inclusive Decision-Making

All team members provide input during design and planning stages.

Skill Development

Tasks are given to help members improve, not only based on what they already know.

7. Risk Analysis and Mitigation

7.1 Technical Risks

Risk	Impact	Mitigation
High traffic overload during project release	System crash or slow response	Use load balancing, caching, scalable backend, stress testing before launch
Integration issues with ministry systems	Delays in syncing housing data	Early API coordination, parallel testing environments
Security vulnerabilities	Data breaches, system downtime	Implement encryption, secure APIs, regular penetration testing
Payment integration difficulties (if approved)	Failed transactions	Partner with secure payment providers, sandbox testing

7.2 Operational Risks

Risk	Impact	Mitigation
Delay in receiving housing data/content	Project timeline disruption	Establish fixed delivery deadlines, maintain constant communication
Limited infrastructure resources from ministry	Poor performance	Suggest cloud-based solutions, adjust architecture accordingly
User resistance to new digital system	Low adoption rates	Provide help guides, tutorials, and simple UX

7.3 Team and Management Risks

Risk	Impact	Mitigation
Uneven work distribution	Conflict or delays	Regular workload evaluation, redistribute tasks early
Communication gaps	Misaligned implementation	Weekly sync meetings, shared documentation
Skill gaps in team members	Slow development	Internal workshops, pairing junior and senior members

8. Professional Communication

8.1 Internal Communication

Formal Weekly Reports

Each team sends progress summaries, blockers, and next week's plan.

Communication Tools

- Microsoft Teams: Quick updates
- Jira: Task tracking
- GitHub: Code collaboration

Meeting Standards

Agendas prepared before meetings, minutes shared after, decisions documented.

8.2 External Communication

- Professional Email Structure
 - Clear subject line, formal tone, concise message, action points.
 - Documentation Delivery
 - All user guides, technical manuals, and designs are stored and shared formally.
-

