U.E.A.B

SCHOOL OF BUSINESS

Department of I.S.C

Technical Documentation and Implementation Guide

Affordable Housing Allocation System

BY: LEONE OKOTH
ID: SLEOOK2111
MAJOR: SOFTWARE ENGINEERING
INSTRUCTOR: MR. OMARI DICKSON

PROPOSAL DOCUMENT

Proposal Document: Affordable Housing Allocation System

Title: Transforming Housing Access Through a Digital Affordable Housing Allocation System

1. Executive Summary

Kenya faces a critical housing shortage, and the current allocation process for affordable housing units is marred by inefficiencies, delays, and lack of transparency. This proposal outlines the development of a centralized, automated Affordable Housing Allocation System to streamline the process, enhance fairness, and restore public confidence in government housing initiatives. A feasibility report outlines the resources, costs, and infrastructure required to successfully complete the project within a 3-month timeline.

2. Introduction

The Kenyan government's Housing Levy Tax initiative aims to fund the construction of affordable housing units. However, manual processes in the allocation phase have created bottlenecks, leaving many eligible citizens underserved. A digital solution can address these issues by automating eligibility checks, application tracking, and allocation processes, leveraging modern technologies to create a robust and secure platform.

3. Problem Statement

The current housing allocation process lacks a centralized, automated system to handle applications, eligibility verification, and tracking. leading to:

- Delayed processing times
- Limited transparency in allocation criteria
- Inadequate applicant notifications
- Susceptibility to bias and human error

These issues have eroded public trust and reduced the overall efficiency of the initiative, making it necessary to adopt a more efficient, tech-driven approach.

4. Objectives

1. Create a centralized platform for citizen registration and housing applications.
2. Automate eligibility verification based on government-defined criteria (e.g., income, dependents).
3. Ensure fairness through a secure, auditable allocation process.
4. Deliver real-time updates and notifications via SMS and email.
5. Provide housing authorities with analytics and reports - These reports help monitor trends, detect inefficiencies, and ensure transparency.

6. Project Scope: Uasin Gishu county (Eldoret)

- The Affordable Housing Allocation System will serve as a comprehensive platform tailored for Uasin Gishu county**: Eldoret**, the county's capital, faces housing shortages due to its growing population and economic activities.

a) **User Registration and Management:** Secure account creation, profile management, and document uploads for eligibility verification.

b) **Eligibility Verification:** Automated checks in the database for income, tax compliance, household details, Prioritization for low-income earners, persons with disabilities, and single-parent households

c) **Application Processing:** End-to-end application management, including submission, report generation, and modification.

d) **Allocation and Ranking:** Allocate houses based on predefined criteria and application timestamps.

e) **Communication Channels:** SMS, email, and in-app notifications to keep applicants updated on their application status.

f) **Administrative Tools:** Dashboards for housing authorities to oversee applications, generate reports, and track allocation metrics.

g) **System Maintenance and Support:** Ongoing technical support, Regular updates based on Eldoret housing policies and performance monitoring.

6. Beneficiaries

a) The implementation of the Affordable Housing Allocation System will benefit multiple stakeholders, including:
b) **Citizens Seeking Affordable Housing:** Faster application processing, real-time updates, and transparent allocation processes.

c) **Housing Authorities:** Streamlined operations, reduced manual work, and access to data-driven insights for policy-making.

d) **Government:** Increased public trust, better housing policy implementation, and improved resource allocation.

e) **Developers and Contractors:** Clearer timelines and faster occupancy, enabling quicker project turnarounds.

f) **Community-Based Organizations (CBOs):** Easier collaboration with authorities to assist vulnerable groups with the application process.

7. Feasibility of proposed solution

The proposed Affordable Housing Allocation System in Kenya aims to address inefficiencies, corruption, and fraudulent activities in the allocation of government-backed housing units. This initiative offers a comprehensive software solution to automate and enhance the housing application and allocation process. Through automated applicant verification, digital identity authentication, secure data management, and online application and payment processing, the system seeks to streamline operations, reduce processing times, and improve security and transparency.

Feasibility Assessment

1. **Technical Feasibility:** The integration of advanced technologies for applicant verification and allocation is technically viable, given the advancements in digital and biometric verification methods. However, the implementation requires substantial upfront investment in technology and infrastructure, including database management and secure payment processing systems.
2. **Operational Feasibility:** Automating the housing allocation process is operationally feasible and can significantly reduce human error, minimize corruption, and enhance efficiency.

3. **Legal and Regulatory Feasibility:** Implementing such a system must comply with national and international data protection and privacy regulation, Robust data encryption and access control measures will be critical to ensure compliance and safeguard personal information. Furthermore, adherence to housing allocation policies and fairness principles is essential to maintain public trust and credibility.

8. Project Plan
8.1 Project Timeline
Month 1: Planning & Setup

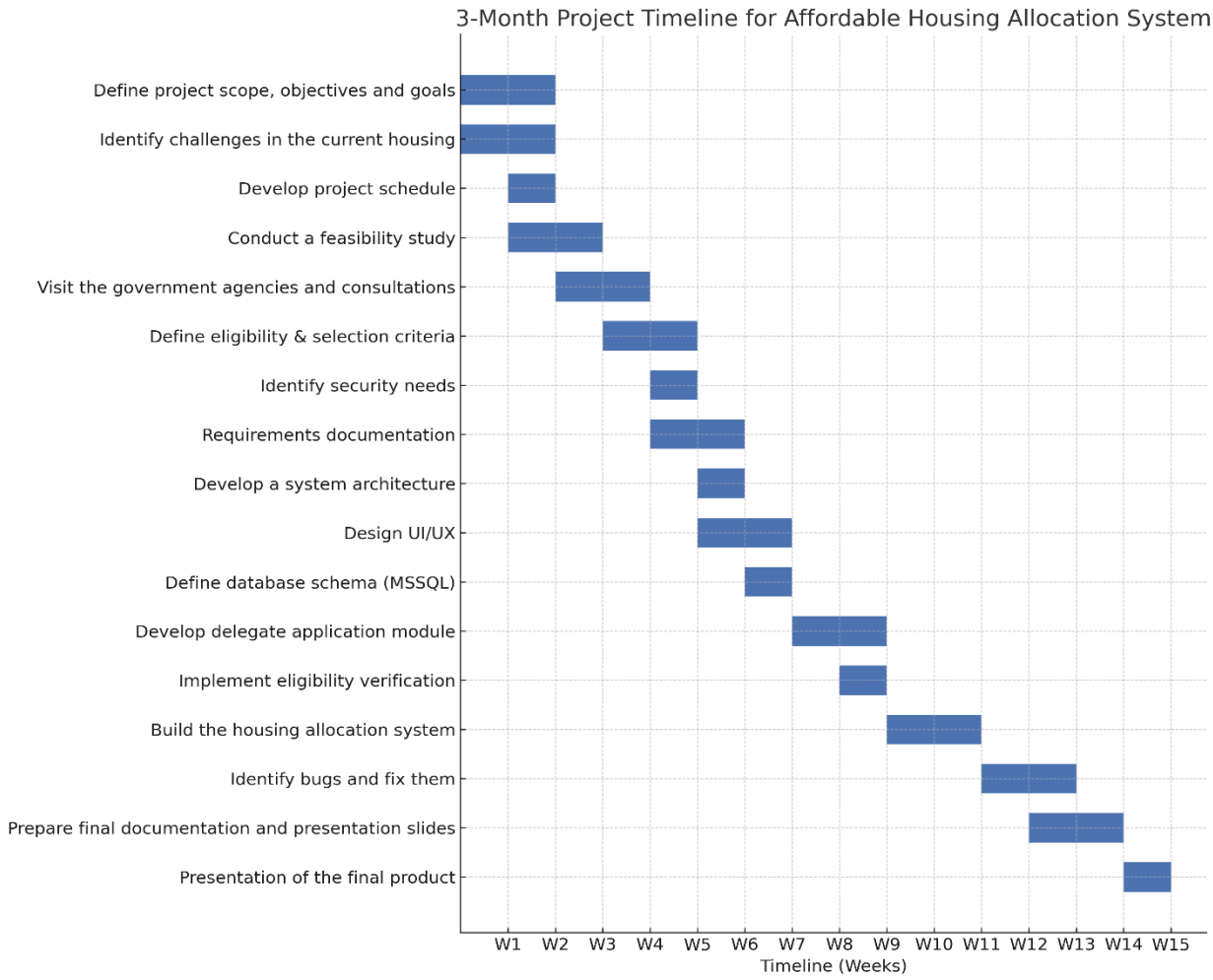
- Define requirements & eligibility rules.
- Design MSSQL database & system architecture.
- Set up backend structure.

Month 2: Development

- Build backend (user auth, allocation logic, security).
- Develop frontend (user page, admin panel, Intuitive UI).

Month 3: Testing & Deployment

- Test allocation logic, security, and UI.
- Deploy on cloud & monitor performance.
- Launch system & onboard users.



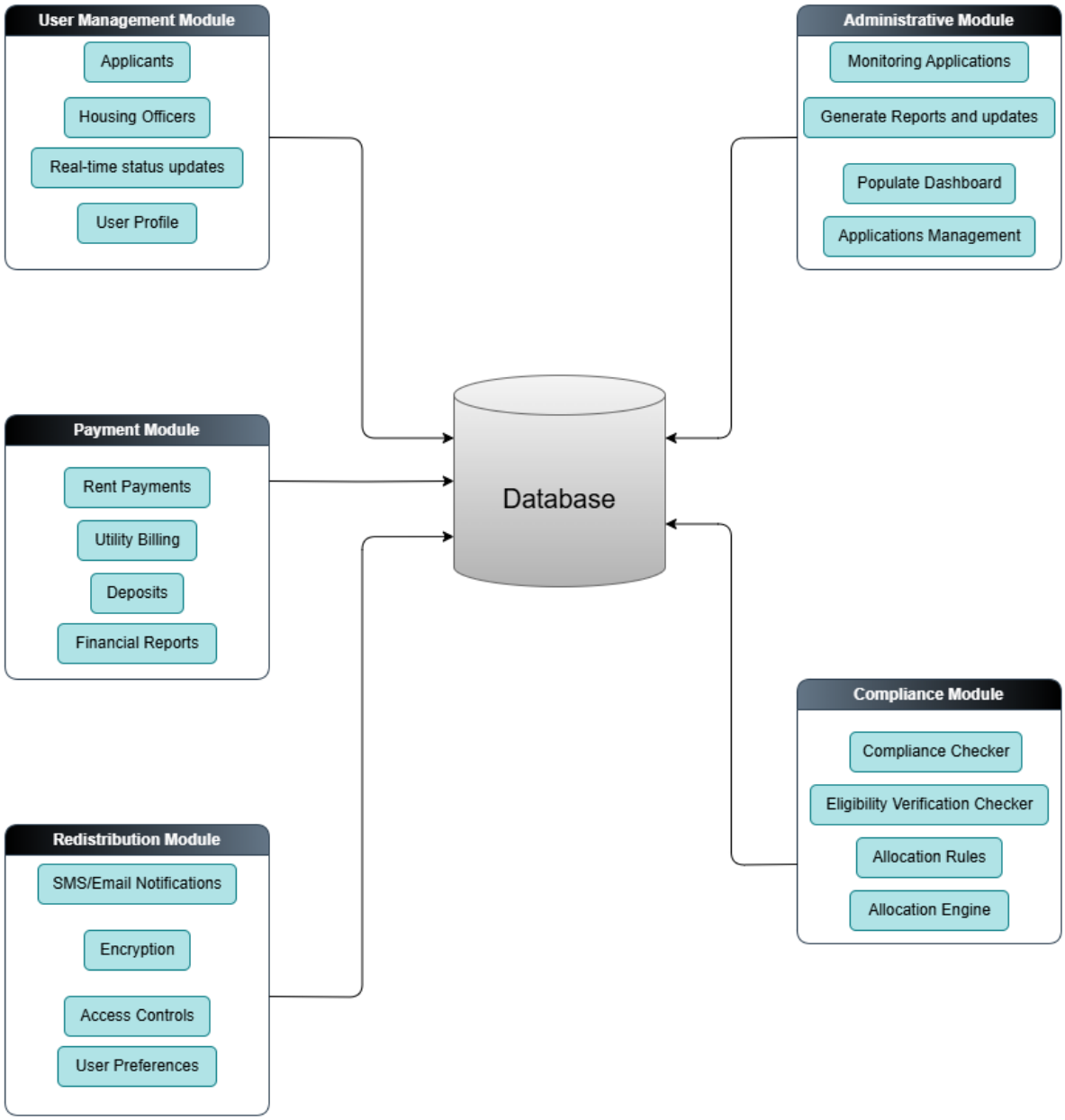
9. Expected Outcomes

- Faster, more efficient allocation of housing units
- Increased transparency and public trust
- Reduced administrative overhead
- Streamlined communication with applicants
- Scalable infrastructure to accommodate future growth

10. Budget

Item	Description	Cost in KSH
Printing papers and binding	For proposal, progress documents and final documentation	1000
Requirements Gathering	Visiting government agencies	3000
Unexpected costs	Budget overrun	2000
TOTAL	COST	6000

11.Conceptual Diagram



12. Conclusion

- By adopting a digital allocation system, Kenya can revolutionize its affordable housing program, ensuring equitable access to housing while boosting public confidence in government initiatives. The feasibility report confirms that this solution is practical, sustainable, and achievable within the outlined budget and timeline. This proposal marks the first step toward a fairer, more efficient housing future for all Kenyans.

13. References

a) Republic of Kenya. (2023). *Affordable Housing Program: Policy Framework and Implementation Strategy.* Nairobi: Ministry of Lands and Physical Planning.

b) World Bank. (2022). *Kenya Urbanization Review: Housing and Infrastructure Challenges.* Washington, DC: The World Bank Group.

c) Smith, J. A., & Kimani, P. (2021). Digital solutions for public sector housing allocation: Lessons from sub-Saharan Africa. *Journal of Housing Policy and Management, 15*(3), 245–267. https://doi.org/10.1080/jhpm.2021.134567

d) United Nations Habitat. (2020). *The State of African Cities: Affordable Housing and Inclusive Growth.* Nairobi: UN-Habitat.

FEATURES OF THE SYSTEM

1.0 System Features

This section outlines the core components of the Affordable Housing Allocation System designed to promote efficiency, transparency, and accessibility.

1.1 Core Functional Features

- I enabled property browsing, allowing users to view housing units with details like location, price, unit type, and availability.
- I built a buyer application form, submitted via modal, capturing the applicant's full name, contact details, and selected property.
- I enforced eligibility verification, where admins review and approve applications based on criteria like employment status.
- I implemented dynamic property allocation, assigning units on a first-come, first-verified basis and locking a unit once payment is confirmed.

- I added a user profile page, showing personal details, application history, real-time allocation status, and a downloadable PDF of allocations.
- I ensured users can edit profile info, such as phone number or email, directly from their profile dashboard.

1.2 Admin Dashboard Features

- I developed a property management module, allowing admins to add, edit, and remove listings and track their real-time status.
- I added a purchase request module, enabling admins to view buyer applications and approve or reject them accordingly.
- I included a customer feedback system, where admins can review feedback and identify improvement areas.
- I integrated PDF report generation, with downloadable reports for:
    - Listed properties
    - Pending cash-based purchase requests
    - Customer feedback
- I added session management, to auto-logout users after 20 minutes of inactivity for improved security.
- I integrated SweetAlert2, to handle confirmations, deletions, and logout prompts with modern popup alerts.

1.3 Payment Integration

- I implemented M-PESA STK Push via IntaSend API, enabling real-time payment prompts to buyer phones.
- I automated payment status updates, switching transactions from "Pending" to "Paid" upon successful callback confirmation.
- I generated PDF receipts using mPDF, including buyer name, property, amount, and transaction ID.
- I added SMS notifications, so buyers receive confirmation once payment is complete.
- I enabled manual payment handling, allowing admin approval for cash or bank payments before allocation proceeds.

1.4 Users

- Applicant (Buyer): Browses properties, applies, and makes payments.
- Admin: Manages listings, reviews applications, and verifies payments.
- Verifier: Reviews eligibility criteria before approval.
- Customer Support: Responds to feedback and user concerns.

- Auditor: Generates reports and reviews logs for accountability.
- System Developer (Internal Admin): Manages technical aspects and system maintenance.

IMPLEMENTATION

2.0 System Implementation

- I followed a modular, milestone-based SDLC to develop and deploy the Affordable Housing Allocation System, covering frontend, backend, database, and integrations.

2.1 System Architecture

- I used HTML5, CSS3 (Bootstrap 5), and JavaScript to build a responsive, mobile-first frontend.
- I developed the backend using PHP, handling sessions, form submissions, and integrating third-party services (IntaSend, mPDF, PHPMailer).
- I implemented the database using Microsoft SQL Server, and connected via SQLSRV drivers for PHP.

2.2 Implementation Phases

2.2.1 Project Initialization

- I defined the project scope and a 3-month timeline.
- I conducted feasibility studies via stakeholder consultations and online research.
- I identified key risks, including payment gateway complexity and SQLSRV compatibility.

2.2.2 Requirements Gathering and Analysis

- I gathered functional requirements like registration, listings, payments, and admin access.
- I captured non-functional needs, including responsiveness, session timeouts, and security.
- I outlined eligibility criteria after interviews with housing policy stakeholders.

2.2.3 System Design

- I designed wireframes and UX flows using Bootstrap with custom frosted-glass styling.

- I structured the database schema, including tables for users, properties, payments, applications, etc.
- I mapped critical system logic using flowcharts and pseudocode for application, payment, and admin review flows.

2.2.4 System Development

- I built responsive UI layouts and modals for interactive forms and confirmations.
- I implemented backend scripts in PHP for form handling, validation, and dynamic content loading.
- I developed an admin dashboard with secured access to property and buyer management.
- I integrated IntaSend STK Push, using `process_payment.php` to trigger M-PESA, save transactions, and update statuses.
- I used mPDF to generate receipts.
- I added SMS alerts through a third-party API to confirm payment success.

2.2.5 Deployment

- I deployed the system using Ngrok with MSSQL support, and IntaSend on production.

2.2.6 Testing & Debugging

- I performed unit and end-to-end testing on all modules.
- I fixed issues like mobile redirection, email SMTP failures, and form blocking without payment.

2.3 Technologies Used

Component	Technology/Tool	
Frontend	HTML5, CSS3, JavaScript, Bootstrap 5	
Backend	PHP 8.5	
Database	Microsoft SQL Server (MSSQL)	
PDF Generation	mPDF	
Emailing	PHPMailer	
SMS Notifications	SMS API (3rd Party)	
Payment Gateway	IntaSend (M-PESA STK Push)	
Alerts/Popups	SweetAlert2	
Version Control	Git	

CHALLENGES FACED

3.0 Challenges Faced

- **I encountered challenges integrating the IntaSend API for M-PESA STK Push**, especially with handling asynchronous callbacks and maintaining payment status accuracy.
- **I resolved SQL Server integration issues with PHP**, by using the official SQLSRV driver and optimizing for MSSQL-specific syntax in both local and remote setups.
- **I faced mobile UI bugs**, like search boxes redirecting to notifications and filter modals overlapping elements on small screens.
- **I dealt with tracking manual (bank) payments**, since they couldn't be auto-verified like M-PESA, requiring admin-side verification features.
- **I addressed security concerns due to inactive sessions**, by implementing auto-logout after 20 minutes of user inactivity, with SweetAlert2 for warning users.
- **I experienced scaling issues on mobile screens**, where elements like modals, buttons, and input fields were misaligned or overflowing due to poor responsiveness

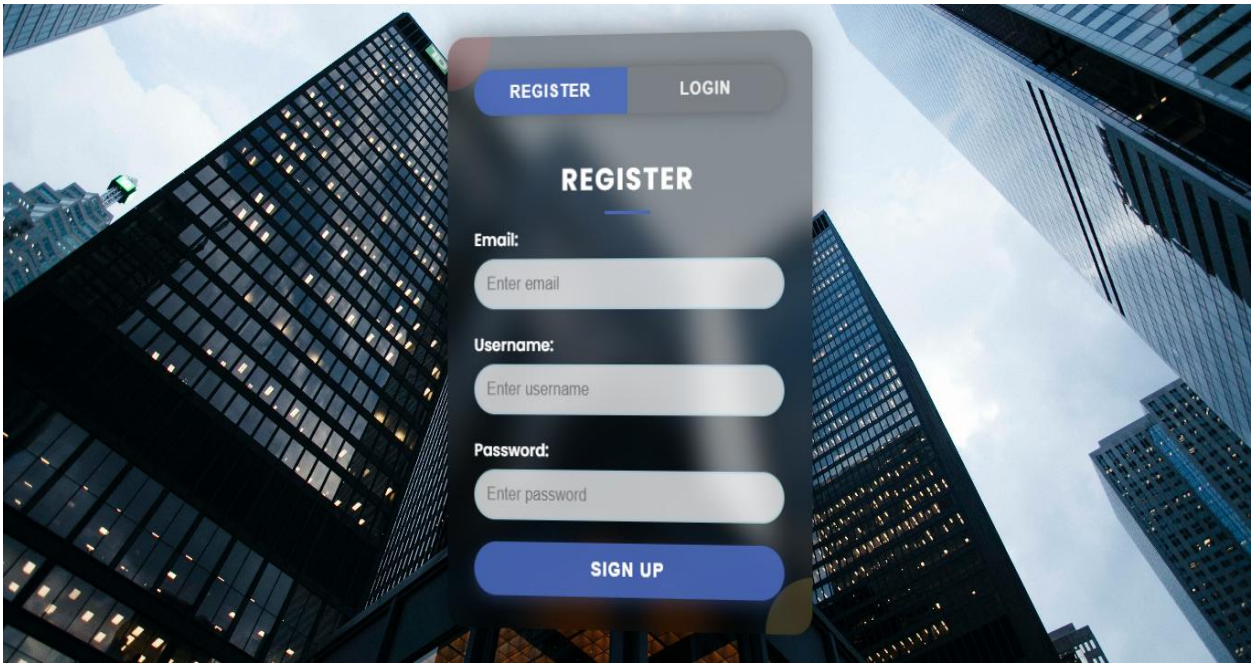
FUTURE RECOMMENDATIONS

4.0 Future Recommendations

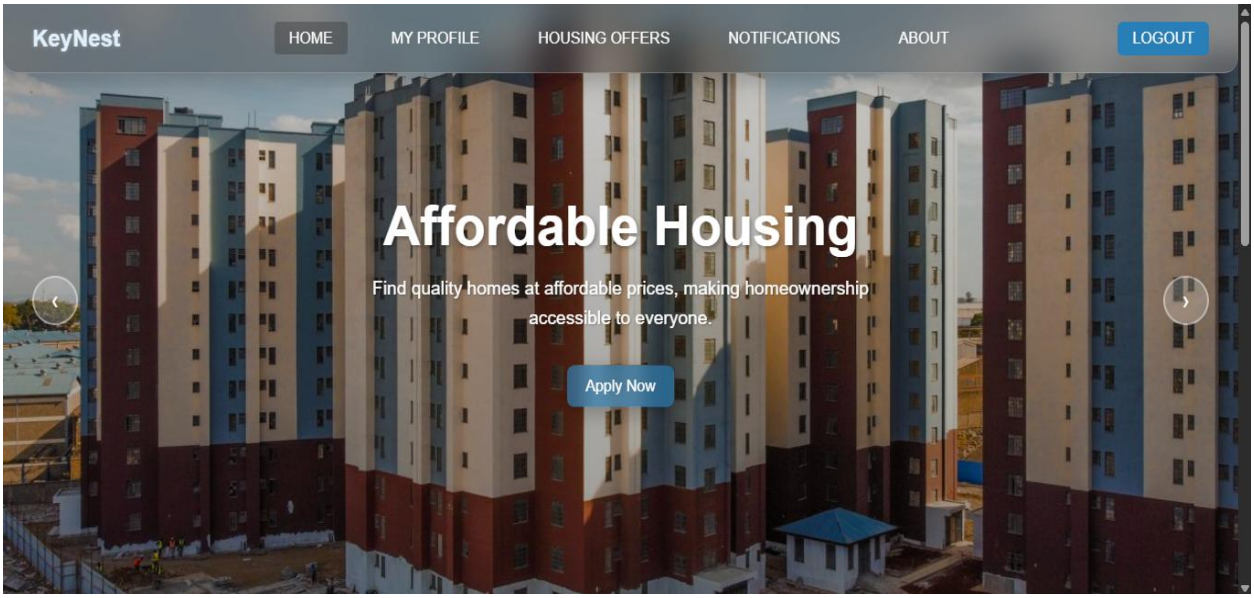
- **I recommend migrating the system to Laravel**, to improve maintainability with MVC structure, built-in security, and efficient API integration.
- **I recommend implementing user authentication with role-based access**, to securely manage different user types like Admins, Applicants, and Verifiers.
- **I recommend integrating National ID and KRA PIN verification APIs**, to enable real-time identity checks and prevent fraudulent applications.
- **I recommend adding analytics and dashboard visualizations**, to track metrics like total applicants per property, payment methods, and allocation speed.
- **I recommend deploying the system to a cloud platform like AWS or Azure**, to ensure better performance, uptime, and scalability as the system grows.

SCREENSHOTS

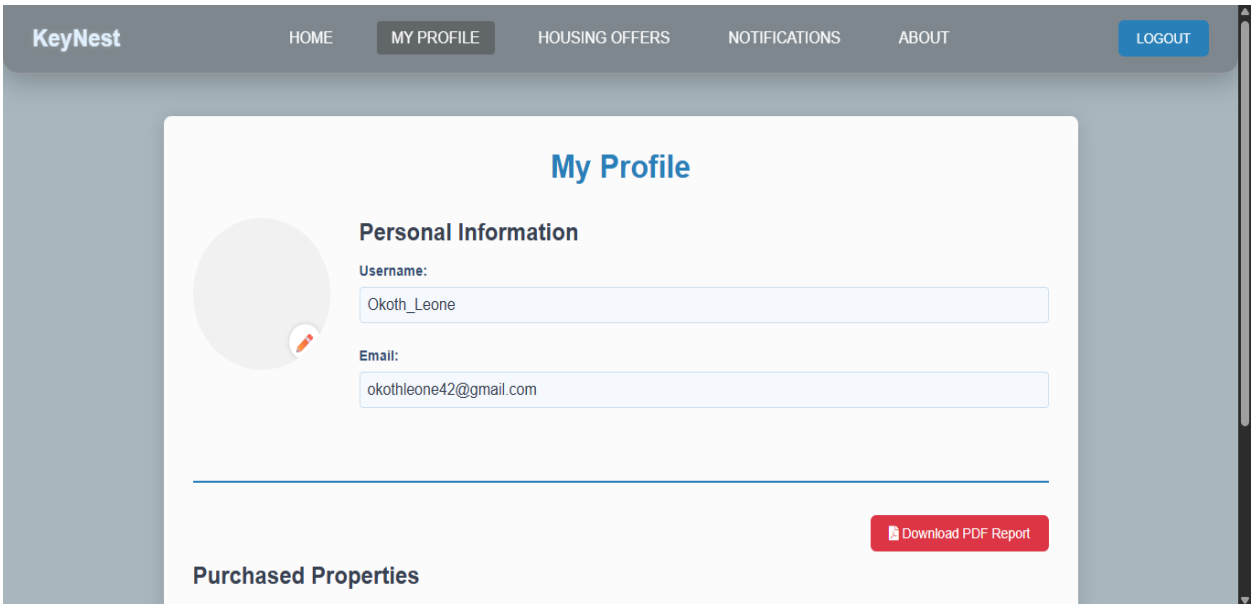
1. Login Page



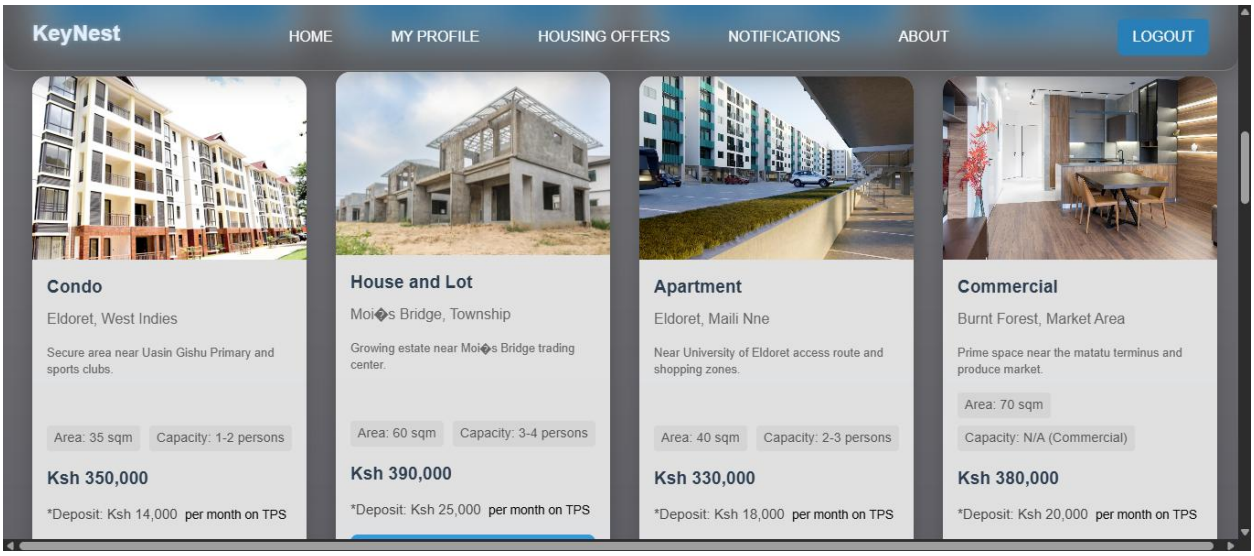
2. Homepage.



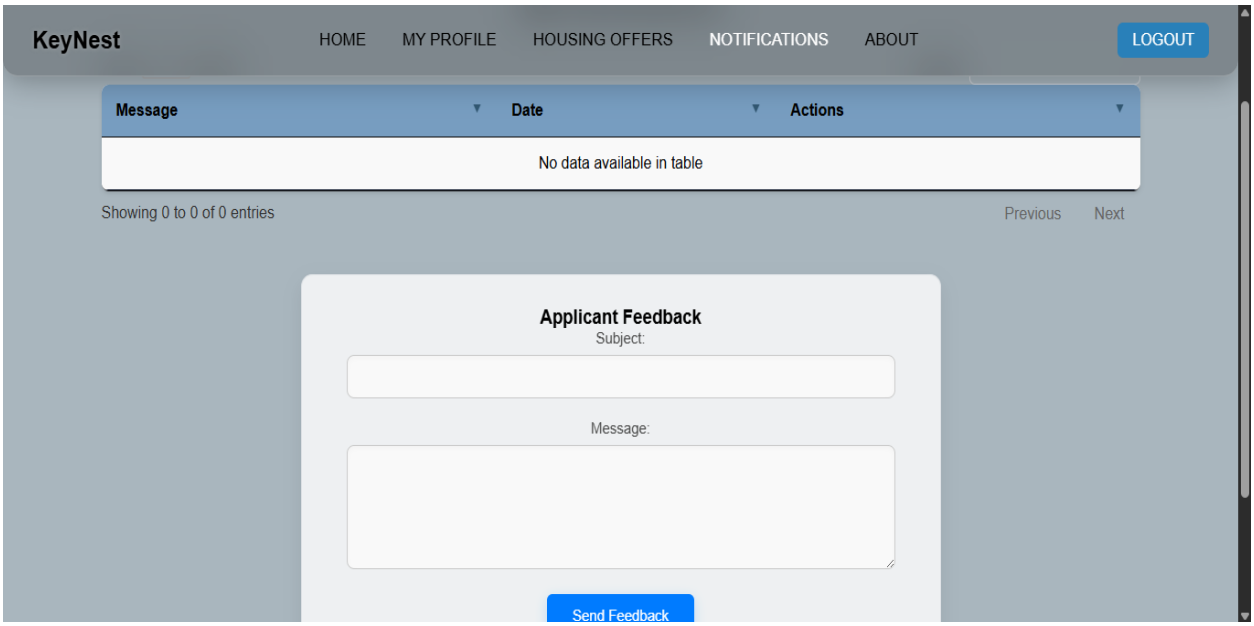
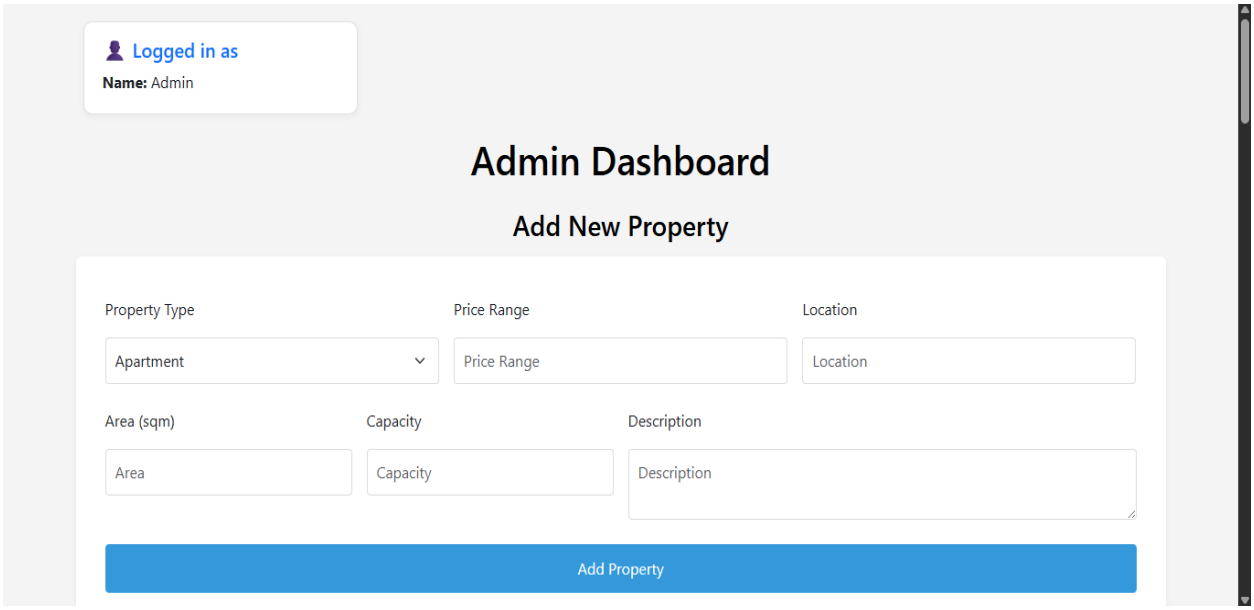
3. Profile Page



4. Searching page



5. User Feedback



6. Admin Dashboard