An Undergraduate Internship Project on Real Estate Property Listing System

By

Shahriar Hosen

Student ID: **1910032**

Summer, 2025

Supervisor:

Asif Mahmud

Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

October 11, 2025

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I, Shahriar Hosen, ID: 1910032, affirm that the report named "Real Estate Property Listing System" for IT WAY BD, presented as part of the criteria for the Degree of Computer Science and Engineering at Independent University, Bangladesh (IUB), is the outcome of my personal research efforts.

I additionally declare that the project was completed while I was a student at the university. I want to express my gratitude to my supervisor, Asif Mahmud, Lecturer in the Department of Computer Science and Engineering, for his support during the completion of this project. I also affirm that the work showcased here represents my individual comprehension and insights acquired during my internship at IT WAY BD. The report accurately references and acknowledges all utilized sources. For any additional questions related to this report, please reach out to my internship supervisor at IT WAY BD via the email address

Signature

Date

Shahriar Hosen

Name

Acknowledgement

I want to express my heartfelt thanks to Almighty Allah (SWT) for His infinite mercy and blessings, which allowed me to complete my internship successfully and prepare this report promptly.

I appreciate the Department of Computer Science and Engineering at Independent University, Bangladesh (IUB) for incorporating internships into the curriculum, granting me the important chance to acquire practical experience in an industrial setting.

I wish to express my heartfelt gratitude to my academic supervisor, Mr. Asif Mahmud, Lecturer, Department of Computer Science and Engineering, IUB, for his ongoing support, insightful feedback, and valuable guidance during my internship and report writing process. His support and encouragement have been vital to my development and learning.

I extend my sincere thanks to my technical supervisor, Mr. Rabbi Hossain, Project Manager, and the whole team at IT WAY BD, where I was fortunate to complete my internship. Their collaboration, support, and expert advice significantly enhanced my understanding and abilities. I am particularly appreciative of the time and assistance they devoted to guiding me throughout my internship

Thank you again to Mr. Atik Hasan, Software Business Development, IT WAY BD, for always there supporting and guiding me throughout the internship process. Massive thank you to the entire Software Development team for working with me so well, along with all their input as to how things work and to foster an environment of good faith.

Finally, I owe my deepest appreciation to my parents and family, whose constant support, inspiration and prayers have been the driving force behind my academic journey and personal achievements. Without their unwavering encouragement, none of this would have been possible.

Letter of Transmittal

Asif Mahmud
Lecturer
Department of Computer Science and Engineering
School of Engineering and Computer Science
Independent University, Bangladesh

Subject: Submission of the internship report to fulfill the requirements for graduation

Dear Sir,
I respectfully submit my internship report, which is a prerequisite for the Bachelor of Science program in the Computer Science and Engineering Department. Working under your guidance throughout this process has been a privilege. This document illustrates my internship at IT WAY BD, where I had the opportunity to work for three months guided by a Principal Consultant.
During this internship, I acquired both theoretical and hands-on knowledge, which enabled me to expand my professional network and engage fully with the corporate environment. I have worked diligently to ensure that the information presented in this report is as thorough as possible, drawing from the valuable experiences and insights gained during my tenure at the company. In compiling this report, I adhered to the provided guidelines, ensuring that all relevant sections are clearly articulated.
I truly hope that this report fulfills the internship program requirements and offers a clear summary of my experiences. I would greatly appreciate it if you could accept this report and share your valuable feedback. Sincerely yours,

Shahriar Hosen
ID: 1910032
Department of Computer Science and Engineering
Independent University, Bangladesh

Evaluation Committee

Supervision Panel

.....	.....
Academic Supervisor	Industry Supervisor

Panel Members

.....	.....
Panel Member 1	Panel Member 2
.....	.....
Panel Member 3	Panel Member 4

Office Use

.....	.....
Internship Coordinator	Head of the Department

Abstract

This internship report presents the design and development of a Real Estate Property Listing System undertaken at IT WAY BD as part of the Bachelor of Science in Computer Science and Engineering program at Independent University, Bangladesh. The project was motivated by the challenges in the real estate sector, where buyers, sellers and agents often rely on fragmented and outdated methods such as newspaper ads or scattered online listings. These practices result in inefficiency, poor communication and lack of trust due to incomplete or misleading information.

To address these issues, a centralized, secure and user-friendly platform was developed using the Laravel framework. The system enables sellers and agents to list properties with complete details including images, descriptions, prices and availability, while buyers can search and filter properties by location, budget and type. Role-based dashboards, authentication mechanisms and property approval workflows were integrated to ensure data accuracy, security and transparency. A responsive front-end design was also implemented to provide accessibility across multiple devices, enhancing user experience.

The project applied knowledge from courses such as data structures, database management and system analysis, combining skills in both front-end and back-end development. Agile methodology was followed to allow iterative development, testing and refinement. Challenges included ensuring scalability, optimizing system performance and safeguarding sensitive data, which were overcome through structured problem-solving and guidance from supervisors.

Overall, the Real Estate Property Listing System demonstrates how modern web technologies can improve efficiency, trust and accessibility in property transactions. Beyond technical achievement, the internship provided valuable professional growth by bridging academic knowledge with practical application, preparing me for future contributions in software engineering and system development.

Keywords— Real Estate, Property Management, Laravel Framework, Web Application Development, Online Property Listing, Safe Verification, Role-Driven Access Management, Property Search and Filtering, Dashboard Design, Automation in Real Estate.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The introduction presents the goals and groundwork of the Real Estate Property Listing System created during the internship at IT WAY BD. It stresses how the project integrates academic education with practical implementation, with the intention of developing a scalable, secure and user-friendly platform for property management. The objectives are centered on increasing accessibility, transparency and communication among buyers, sellers and agents, while automating manual processes. The problem statement highlights the inefficiencies, trust concerns and lack of centralized systems in current real estate practices, reinforcing the demand for a modern digital platform that simplifies and secures property transactions.

1.1 Overview/Background of the Work

Internships are essential in shaping students' academic knowledge and facilitating their career development. They offer an opportunity to implement theoretical concepts in real-world situations. The main aim of internships is to assist students in identifying their professional goals. As part of the Computer Science and Engineering curriculum at the School of Engineering and Computer Science at Independent University, Bangladesh (IUB), I was mandated to undertake an internship for hands-on experience. I was fortunate to secure a position at IT WAY BD, a well-regarded software firm that focuses on delivering innovative IT solutions for enterprises.

Throughout my internship, I was tasked with the development of a Real Estate Property Listing System utilizing the Laravel framework. This system was intended to streamline property management and create a centralized platform for buyers, sellers, and agents to interact effortlessly. My responsibilities included contributing to both the front-end and some back-end components of the system. I completed preparatory tasks such as designing landing pages, creating dashboards, and developing various interfaces, which significantly enhanced my skills and understanding of web development. This report presents a detailed account of the work I undertook during my internship, the challenges I encountered, and the lessons I learned, emphasizing how this experience has contributed to my personal and professional growth. It has provided me with a more profound insight into the software industry and practical knowledge

that will be advantageous for my future career.

1.2 Objectives

The fundamental objective of the Real Estate Property Listing System is to develop a scalable, user-friendly, and efficient web-based platform that can innovate property management processes. This system is structured to allow property owners to register and present their properties with complete details, which include descriptions, images, pricing, and availability status. By providing these features, the platform ensures that property owners can showcase their properties to a wider audience in a more organized and professional way. Buyers, on the other hand, can search for properties using advanced filters such as location, budget and property type, which makes it easier for them to find suitable properties that match their requirements. Agents are also provided with tools to manage their property portfolios more effectively, thereby reducing the complexity of handling multiple listings.

Another important objective of this project is to ensure data security, transparency and reliability in the property management process. This is achieved by integrating user authentication mechanisms, secure databases and role-based access control to protect sensitive information. The system seeks to build trust among stakeholders by ensuring that property data is accurate, up to date and protected from unauthorized access.

The project also aims to enhance communication among buyers, sellers and agents through an integrated inquiry and response system, which allows users to interact directly within the platform. This reduces dependency on external communication channels and makes the transaction process smoother.

In addition, the Real Estate Property Listing System is structured to reduce reliance on manual methods, which are typically time-consuming, prone to errors, and inefficient. By automating essential tasks such as property listing, searching, and documentation, the system minimizes human error and increases precision. Overall, the project's objectives revolve around improving accessibility, efficiency and transparency in the real estate industry, ultimately creating a reliable digital solution that can benefit all stakeholders.

1.3 Scopes

The Real Estate Property Listing System, built on the Laravel framework, aims to optimize property management while improving the experiences of buyers, agents and administrators. The project's scope includes the following components and functionalities:

Property Listing and Management: Users create, update and handle property management listings with details such as title, description, price, property type, images and location.

Search and Filtering: Prospective buyers have the ability to explore available properties by utilizing sophisticated search filters, which include criteria such as price range, type of

property, number of rooms, and geographical location.

Authentication and Role Management: Secure registration and login for different roles including Admin, Agent and Buyer. Admin has full control over managing users and listings.

Property Approval Workflow: Admins can review and approve property listings before they are published on the site, ensuring data quality and authenticity.

Dashboard for Users: Separate dashboards for Admins and Sellers and Agents to manage properties, inquiries and user details effectively.

Responsive Frontend Design: The system provides a mobile-friendly, user-friendly interface that ensures accessibility across devices.

Scalability and Extensibility: Built on Laravel, the system can be extended with additional modules such as payment integration, map-based search or rental management.

Knowledge Transfer and Collaboration: The project encourages team members to exchange information and develop by fostering multidisciplinary collaboration. It seeks to establish a learning and problem-solving culture.

1.4 Problem Statement

In the real estate sector, property dealings are often slowed down by outdated processes and poor communication channels. Buyers, sellers and agents typically rely on fragmented systems such as newspaper ads, physical posters, or scattered online listings, which makes the experience inefficient and frustrating. For buyers, it can be difficult to access complete and reliable property information, while sellers often struggle to showcase their listings in a way that reaches a broad and serious audience.

On top of that, many of the existing digital platforms lack robust verification mechanisms. This creates serious issues of trust, as property details may be incomplete, misleading, or even fraudulent. Buyers are left sifting through inconsistent listings without proper filters to match their preferences, while sellers and agents find it challenging to present their properties in a professional and structured manner.

The lack of a centralized, user-friendly and secure system ultimately makes property transactions time-consuming, inefficient, and prone to errors or scams. Without a reliable platform to bridge these gaps, the process of buying, selling or renting real estate remains unnecessarily stressful for all parties involved.

This highlights the urgent need for a modern, scalable and interactive property listing platform one that ensures transparency, builds trust through verification and simplifies the entire transaction process. Such a platform should enable buyers to search and filter properties with ease, give sellers and agents professional tools to manage and market their listings and provide administrators with efficient oversight. By addressing these gaps, the industry can move toward a faster, safer and more transparent real estate experience for everyone.

Chapter 2

Literature Review

The literature review reveals the significance of undergraduate courses in forming the basis for the Real Estate Property Listing System's development, which includes essential skills in programming, database management and system design. Related studies and platforms like Bikroy, Lamudi, Airbnb and Zillow have contributed insights into industry practices, highlighting issues related to scalability, trust and accessibility. Academic research has pointed out the demand for secure, efficient and user-centered real estate systems. This project seeks to bridge these gaps through a Laravel-based solution that incorporates robust authentication, modular design and scalability tailored to the local real estate market.

2.1 Relationship with Undergraduate Studies

The knowledge and skills developed during the undergraduate program were crucial to the success of this project. If the foundational courses in web development and database systems had not been completed before commencing this project, it would have been significantly more arduous. The courses include:

CSE 203 – Data Structures: This course covered organizing and handling complex data structures, including arrays, objects and classes. These concepts were essential in implementing efficient property search and filtering features. Proper use of data structures made the system faster and more reliable in handling large volumes of property listings.

CSE 213 – Object-Oriented Programming: This course offered a broad exploration of classes and objects and emphasized modular programming to reduce duplication and improve code reusability. These principles were critical in modularizing the Laravel-based real estate project, allowing better separation of concerns, scalability and maintainability of the code.

CSE 303 – Database Management: This course formed the foundation upon which the project was planned and designed. Topics such as entity-relationship diagrams, normalization and SQL queries were applied in designing the MySQL database structure for managing users, properties and transactions. These methods helped create, plan and strategize the database design for the system.

CSE 307 – System Analysis and Design: This course introduced tools and methodologies essential for analyzing requirements and designing the system. Concepts such as system modeling, data flow diagrams, UML diagrams, WBS and feasibility studies were applied to design user roles (Admin, Agent, Buyer), workflows and dashboards. This knowledge was critical in bridging client requirements with system functionality.

CSE 309 – Web Applications and Internet: This course focused on web application development using industry-demanded technologies like HTML, CSS, JavaScript and PHP[1]. The knowledge gained was directly applied in building the frontend of the property listing platform and integrating it with Laravel on the backend. It also provided the skills needed to deploy the application in a server environment.

These courses collectively provided the technical and analytical foundation necessary to build the Real Estate Property Listing System. They ensured that the project was both functional and aligned with real-world web application standards.

2.2 Related works

Typically, most real estate websites or property listing platforms are developed using simple technologies such as HTML[2], CSS[3], Bootstrap[4], JavaScript[5]. These approaches allow developers to quickly launch sites but often lack scalability, advanced search features and proper backend integration. Since the goal of this project was to build a more powerful, secure and scalable application, we used the Laravel framework[6], which offers MVC architecture, robust authentication and strong database management. Several notable platforms and studies are related to this project:

Bikroy.com:[1] One of the largest online marketplaces in Bangladesh, Bikroy allows users to post ads for properties, vehicles and products. While its property section provides extensive listings, the platform primarily focuses on classifieds and lacks dedicated dashboards tailored specifically for real estate agents and buyers.

Lamudi: Lamudi is a dedicated real estate website widely used in Bangladesh and other countries. It provides advanced search filters, map integration and verified listings. However, Lamudi is mainly agency-oriented and requires premium packages for better property visibility, making it less accessible for individual sellers [7].

Airbnb:[2] Although designed for short-term rentals, Airbnb is an example of a highly interactive platform with secure booking, user reviews and advanced property search. Its focus on trust-building between users through rating and review mechanisms strongly aligns with the features of this project.

Zillow: A major real estate platform in the United States, Zillow provides property listings

[1] `https://bikroy.com`
[2] `https://www.airbnb.com`

along with price prediction models and mortgage calculators. It highlights how machine learning and data-driven decision-making can enhance real estate systems [8].

Research by Alqaralleh et al. (2021): Proposed a smart real estate system using cloud-based services and AI to improve search efficiency and security in property management. The study emphasized the need for scalable platforms that integrate both buyers and sellers seamlessly [9].

Study by Zhou et al. (2020): Explored the use of recommendation systems in property listing websites. Their findings suggest that intelligent filtering and personalization greatly enhance user experience in browsing large volumes of property data [10].

Rahman and Hossain (2019): Investigated online property platforms in Bangladesh and found that many lacked proper authentication, trust mechanisms and data security. Their research supports the need for a Laravel-based system with verified listings and role-based access control [11].

Hasan and Jahan (2022): Conducted a case study on the use of Laravel in property management systems, demonstrating how Laravel's MVC structure simplifies development while ensuring scalability and maintainability [12].

Ahmed and Chowdhury (2020): Explored the opportunities and barriers of adopting e-property platforms in Bangladesh, concluding that technological awareness and trust are the two most critical factors influencing adoption [13].

Singh and Kumar (2021): Compared real estate portals in emerging markets and found that features such as personalized search, data accuracy and mobile compatibility are key differentiators for user satisfaction [14].

These related works show that while global platforms like Airbnb and Zillow set usability and data-driven benchmarks, local platforms such as Bikroy and Lamudi highlight the real-world challenges of trust, affordability, and accessibility. Academic studies further confirm the importance of security, scalability, and personalization in real estate systems. This project addresses these gaps by developing a Laravel-based real estate property listing system tailored to the local market, with user dashboards, secure authentication and the potential for future extensions such as payment gateways and geolocation-based search.

Chapter 3

Project Management & Financing

The project management and financing of the Real Estate Property Listing System were conducted through a methodical and organized approach. The Work Breakdown Structure (WBS) segmented the project into distinct, manageable phases, which included planning, design, coding, testing and deployment, thereby facilitating a seamless workflow and ensuring accountability. Time and resources were adeptly managed through the use of tools such as the Critical Path Method and Gantt Chart, which helped in pinpointing essential tasks and preventing delays. Resource allocation was focused on both backend and frontend development, while cost estimation reflected realistic expenses, amounting to approximately BDT 516,000, with the majority of the budget allocated to development and testing to ensure quality assurance.

3.1 Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) is an essential project management technique that breaks down a complex system into smaller, more manageable components. For the Real Estate Property Listing System, the WBS was designed to clearly define every phase of development, ensure systematic progression and promote accountability at each step. By organizing the project hierarchically, the WBS helped eliminate duplication of effort, reduced the possibility of overlooked activities and provided clarity for all stakeholders involved in the system development. The application of WBS in this project simplified the overall complexity by dividing the system into seven major phases: Project Initiation and Planning, Requirement Development, Design Development, Coding, Testing, UAT and Training and Deployment and Closure. Each of these phases was then further decomposed into specific, actionable subtasks. This layered approach ensured that the workflow was streamlined, progress was easily monitored and resources were allocated appropriately.

Project Initiation and Planning: This initial stage focused on establishing the foundation of the project. It included project planning, identifying and allocating resources, assigning responsibilities and formally initiating the project through a structured kick-off.

Requirement Development: This phase emphasized gathering and analyzing the require-

ments of all stakeholders, including buyers, agents and administrators. Detailed consultations were carried out to define the system's scope, which was then documented in the baseline requirement document.

Design Development: Once requirements were defined, the design phase prepared both high-level architecture and low-level design specifications. This included UI/UX wireframes, database schema diagrams and system workflow models.

Coding: The development phase was organized into module-wise coding tasks. Each module underwent code reviews to maintain quality, followed by unit testing to detect errors early. Draft releases were generated to validate progress and facilitate incremental development.

Testing: A dedicated phase for quality assurance involved preparing test plans and cases, executing systematic testing cycles, reporting bugs, performing bug fixes and conducting retests. This phase ensured reliability, functionality and security before deployment.

UAT and Training: UAT an analysis was performed to validate whether the system met stakeholder expectations. The UAT environment was prepared, tests were executed with end users and feedback was collected. Alongside testing, training sessions were provided to sellers, agents and administrators, equipping them with the knowledge to operate the system effectively.

Deployment and Closure: The final phase dealt with the system's transition from development to a live environment. Activities included preparing the hosting environment, completing the system handover checklist, taking formal sign-off from stakeholders and issuing a project closure note.

By employing this structured WBS, the Real Estate Property Listing System was able to progress logically through each stage, ensuring smooth coordination among team members. The WBS not only improved efficiency and reduced risks but also established a framework for accountability, monitoring and control. Ultimately, it served as the backbone of the project's management strategy, ensuring that the system was successfully completed and aligned with stakeholder expectations.

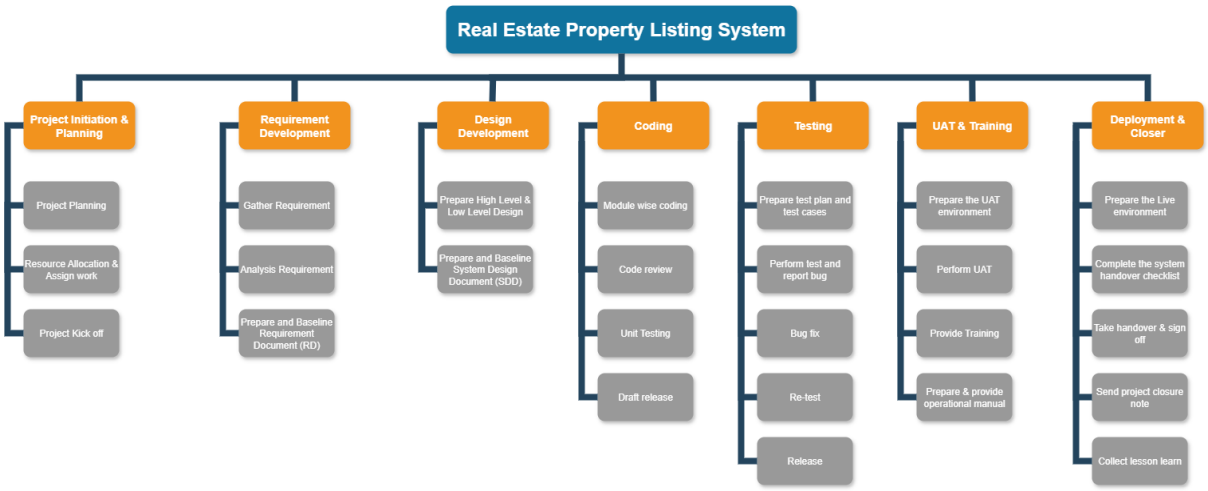


Figure 3.1: Work Breakdown Structure for Real Estate Property Listing System

3.2 Process/Activity-wise Time Distribution (Critical Path Method)

The allocation of time to each activity strongly influenced the pace and success of the Real Estate Property Listing System. Because the project had to be completed within a limited timeframe, identifying the critical path was vital. The critical path signifies the series of interdependent activities that dictate the overall duration of a project; thus, any postponement in these tasks would consequently lead to a delay in the project's completion. For this project, the critical path began with Requirement Gathering, which laid the foundation for all subsequent work. It then advanced into System Architecture and UI/UX Design, where the structure, workflows and user experience were defined. The next major stage was Backend and Frontend Development, which provided both the system's core functionality and its interface. Once development was completed, the focus shifted to Testing and UAT to verify system performance and ensure stakeholder needs were met. The path concluded with Deployment and Documentation, marking system delivery, knowledge transfer and project closure.

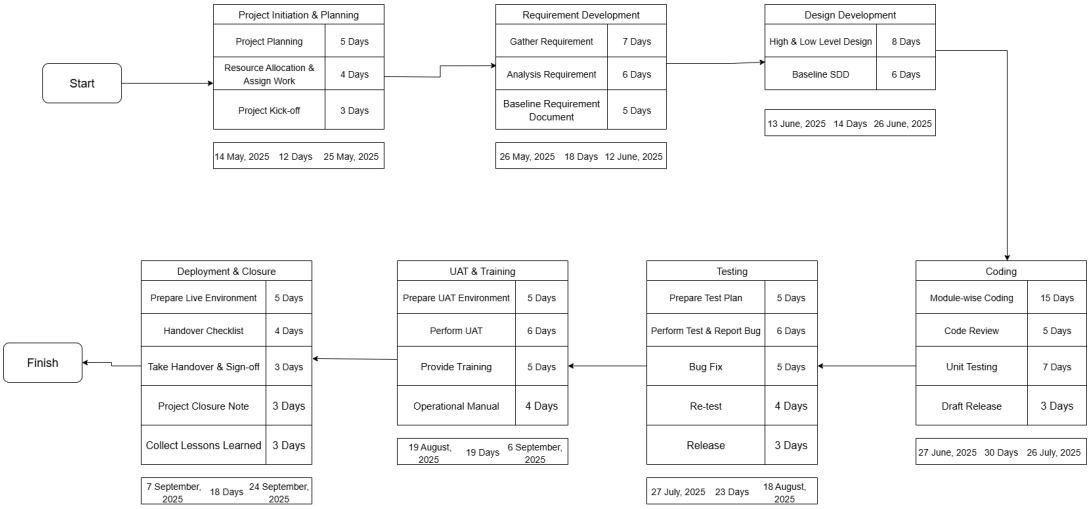
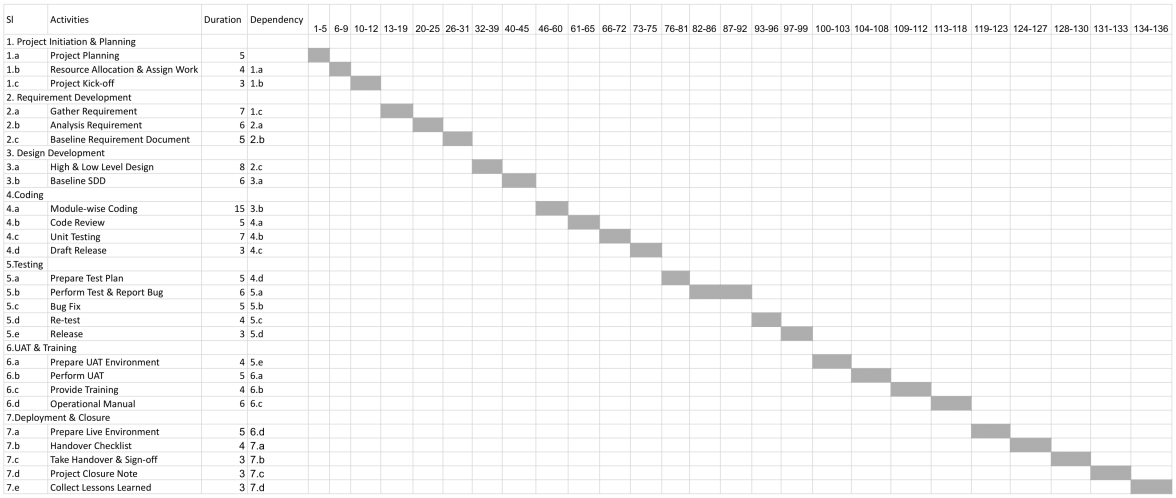


Figure 3.2: Critical Path Method

By analyzing the critical path, project managers were able to prioritize the most time-sensitive tasks, allocate resources accordingly and avoid unnecessary delays. This ensured that milestones were achieved efficiently and the Real Estate Property Listing System was delivered successfully within the planned short timeframe.

3.3 Gantt Chart

Throughout the development phase the Real Estate Property Listing System, I used a Gantt chart as my main project management tool. It allowed me to visualize all the tasks, their durations and how each activity depended on the previous one. This helped me plan my time effectively and follow a clear sequence of work without delays. The project began with Initiation and Requirement Development, where I planned, gathered and analyzed requirements. It then moved into the Design Phase to create system designs and documentation, followed by the Coding Phase, which covered module development, reviews and unit testing. After coding, the Testing Phase ensured quality through planning, bug fixing and re-testing. This was followed by UAT and Training, where I prepared the environment, performed acceptance testing and created the operational manual. The project concluded with Deployment and Closure, including live setup, handover, sign-off and documentation of lessons learned.



Sl	Activities	Duration	Dependency
1. Project Initiation & Planning			
1.a	Project Planning	5	
1.b	Resource Allocation & Assign Work	4	1.a
1.c	Project Kick-off	3	1.b
2. Requirement Development			
2.a	Gather Requirement	7	1.c
2.b	Analysis Requirement	6	2.a
2.c	Baseline Requirement Document	5	2.b
3. Design Development			
3.a	High & Low Level Design	8	2.c
3.b	Baseline SDD	6	3.a
4.Coding			
4.a	Module-wise Coding	15	3.b
4.b	Code Review	5	4.a
4.c	Unit Testing	7	4.b
4.d	Draft Release	3	4.c
5.Testing			
5.a	Prepare Test Plan	5	4.d
5.b	Perform Test & Report Bug	6	5.a
5.c	Bug Fix	5	5.b
5.d	Re-test	4	5.c
5.e	Release	3	5.d
6.UAT & Training			
6.a	Prepare UAT Environment	4	5.e
6.b	Perform UAT	5	6.a
6.c	Provide Training	4	6.b
6.d	Operational Manual	6	6.c
7.Deployment & Closure			
7.a	Prepare Live Environment	5	6.d
7.b	Handover Checklist	4	7.a
7.c	Take Handover & Sign-off	3	7.b
7.d	Project Closure Note	3	7.c
7.e	Collect Lessons Learned	3	7.d

Figure 3.3: Gantt Chart for the Real Estate Property Listing System

3.4 Process/Activity-wise Resource Allocation

Resource allocation for the Real Estate Property Listing System was carefully planned to maximize productivity and ensure smooth progress across all phases. Backend development accounted for the largest share of resources, as it involved implementing domain logic, authentication mechanisms, property listing workflows and role management. Alongside this, frontend development focused on building a user-friendly interface with responsive layouts and intuitive navigation to improve the overall usability of the system. Database design was also prioritized to maintain data consistency, reliability and efficient retrieval across different modules. In addition, testing activities were allocated significant attention to guarantee functionality, security, and stability before deployment. Finally, documentation played a critical role not only for academic submission but also for handover, knowledge transfer, and future system maintenance. By distributing resources across these core areas, I was able to ensure that each stage contributed effectively to the successful completion of the project.

3.5 Estimated Costing

Even though the project was academic, a costing analysis was prepared to simulate real-world deployment. The initiation and planning phase, which included project planning, resource allocation and kickoff activities, required about BDT 53,000. The requirement development stage, involving requirement gathering, analysis and baseline documentation, accounted for approximately BDT 67,000. The design phase, where high- and low-level designs and the system design document were prepared, required around BDT 52,000. The coding and development phase represented the largest share, with module-wise coding, code reviews, unit testing and draft release costing nearly BDT 88,000 in total. The testing phase, which included test planning, execution, bug fixing, re-testing and release, consumed about BDT 95,000, reflecting the importance of ensuring quality and reliability. The UAT and training phase, which covered preparing the UAT environment, conducting acceptance testing, providing training and preparing the operational manual, required approximately BDT 78,000. Finally, the deployment and closure phase, which involved preparing the live environment, completing the handover checklist, obtaining sign-off, issuing closure notes and documenting lessons learned, amounted to around BDT 83,000.

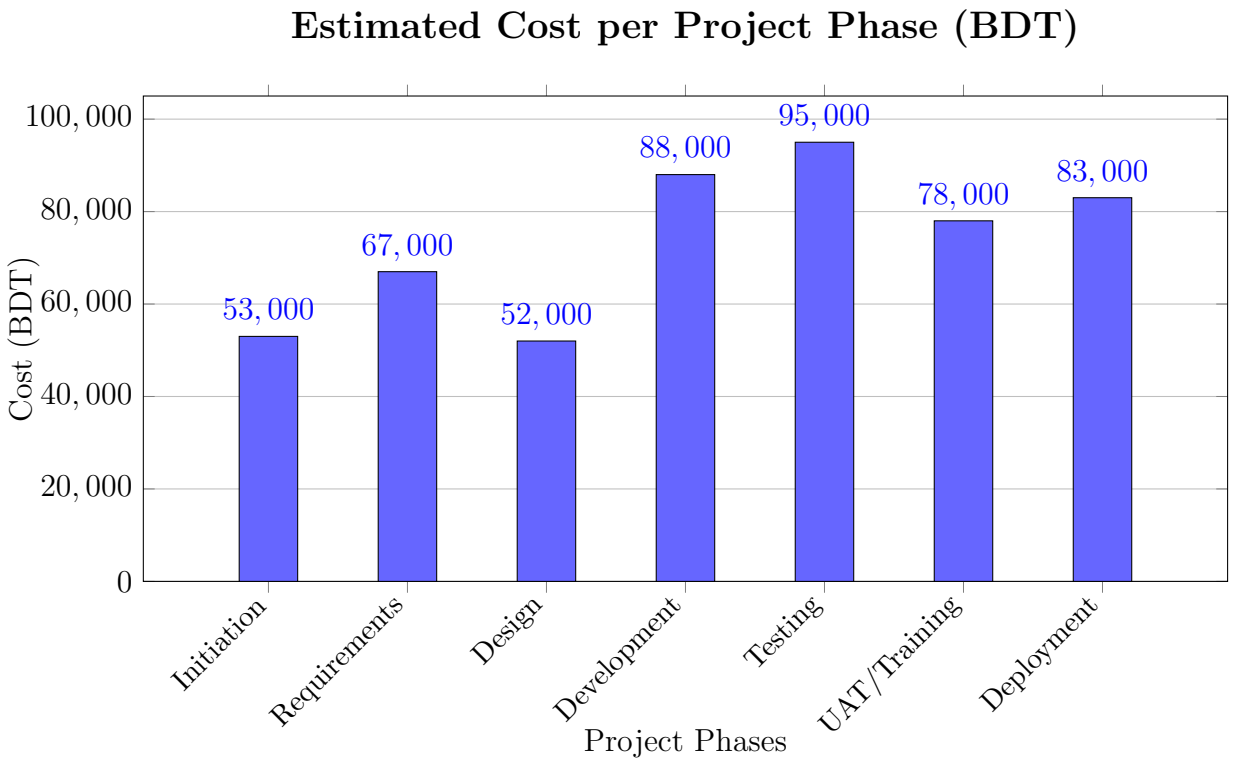


Figure 3.4: Estimated Costing Breakdown for the Project

Together, the estimated cost of the project was approximately BDT 516,000, with the majority of resources allocated to development and testing to ensure that the system was functional and reliable.

Chapter 4

Methodology

The methodology chapter explains the organized approach employed in the development of the Real Estate Property Listing System. The project adhered to the Agile model, which facilitated flexibility and iterative enhancements, allowing for the effective integration of new requirements. The development process encompassed meticulous planning, responsive design, and execution utilizing Laravel, MySQL, HTML, CSS, JavaScript and Bootstrap within a local XAMPP environment. Comprehensive testing, which included unit, integration, system, responsiveness and security assessments, guaranteed both reliability and usability. In summary, the methodology offered a systematic, collaborative and adaptive framework that culminated in the successful development of a secure, scalable and user-friendly web application.

4.1 Project Overview

A methodology refers to a set of principles, guidelines and procedures designed to operate systematically. It provides a structured framework to achieve objectives, solve problems and carry out specific activities. These methods help standardize processes, promote consistency and encourage collaboration between team members. For this project, the focus was on developing a Real Estate Property Listing System that is secure, scalable and user-friendly. To achieve this, different Software Development Life Cycle (SDLC) models were studied and evaluated before selecting the most appropriate approach. The system was envisioned as a web-based platform that simplifies property transactions by bringing buyers, sellers, agents and administrators together on a centralized interface. Built using the Laravel framework, the system integrates both frontend and backend components to ensure efficiency, trust and accessibility. Development and testing were carried out in a localhost environment (XAMPP[15]), which provided a controlled and cost-effective setup for experimentation, debugging and demonstration. The technology stack consisted of HTML, CSS, JavaScript and Bootstrap for the frontend, Laravel (PHP) for the backend and MySQL[16] for database management. This combination ensured that the system adhered to modern web standards while being practical for academic use and adaptable for future deployment on live servers.

4.2 Planning

The planning stage provided the foundation for the system development. The first step was to define goals that aligned with the needs of the stakeholders. The main goals of the project were to create a secure and centralized property listing platform, implement role-based dashboards for agents and administrators, and allow buyers to easily filter properties based on preferences. Key stakeholders identified for the project included:

Property Owners and Sellers: who could upload and manage property listings.

Buyers: who could search and filter properties according to their requirements.

Agents: who could oversee multiple properties and manage their portfolios.

Administrators: who were responsible for reviewing and approving listings before publication.

To better understand the existing landscape, a small market study was conducted by reviewing platforms such as Bikroy, Lamudi and Zillow. These platforms provided useful insights but also revealed limitations in areas like verification, scalability and user experience. Based on this analysis, the scope of the project was defined to include features such as:

- Real estate property listing and management.
- Advanced filtering by location, property type, and budget.
- Role-based authentication with secure login.
- An approval workflow for property verification.
- A basic inquiry system to facilitate buyer-seller communication.
- Responsive design for cross-device accessibility.

The project followed a structured timeline beginning with requirement gathering and database design, moving through frontend and backend development, integration and testing and finally preparation of the report. While the project was hosted locally and did not require a financial budget, resources were allocated for software tools, documentation and system testing. Risks were also considered during this phase. Common risks included dependency on a local environment, data security concerns and scalability limitations. These were mitigated through careful database design, the use of role-based access control and the adoption of a modular system architecture that could support future extensions.

4.3 Design

The design phase focused on building interfaces that were both responsive and user-friendly. Wireframes were created to visualize the flow of dashboards for sellers, buyers and administrators. Bootstrap was used to ensure consistency across devices, while the database schema was carefully designed in MySQL to maintain efficiency and avoid redundancy. Special attention was

given to role-based access during design. Administrators were provided with oversight features, sellers and agents were given portfolio management tools and buyers had simplified navigation with search and filtering. This design ensured that each user type interacted only with relevant parts of the system.

4.4 Implementation

The implementation phase translated the designs transformed into a functional web system. The frontend was built using HTML, CSS, JavaScript and Bootstrap ensuring responsive layouts and smooth navigation. The backend was developed in Laravel, following the MVC architecture to maintain separation of concerns and scalability. The MySQL database was integrated with Laravel to manage property details, user accounts and inquiries efficiently. Authentication and role management were implemented using Laravel's built-in authentication system, which ensured secure access for different types of users. The entire project was deployed and tested locally on XAMPP, which simulated a web server environment with Apache, PHP and MySQL.

4.5 Testing

The testing was performed in different stages to ensure the functionality, reliability, and security of the system. The following methods were implemented:

Unit Testing: Authenticated individual elements like login, property creation, and approval workflows.

Integration Testing: Checked whether the client-side and server-side modules communicated correctly, particularly in property submissions and database retrievals.

System Testing: Ensured the overall workflow, from property submission to approval and listing, worked seamlessly.

Responsiveness Testing: Confirmed that the system displayed correctly on multiple devices and screen sizes.

User Acceptance Testing (UAT): Conducted with peers to collect feedback on usability and performance. **Security Testing:** Implemented form validation and tested against common vulnerabilities such as SQL injection.

4.6 Tools and Libraries

A variety of tools and libraries were utilized throughout the project:

Frontend: HTML, CSS, JavaScript, Bootstrap
Backend: PHP with Laravel framework
Database: MySQL
Hosting: Localhost

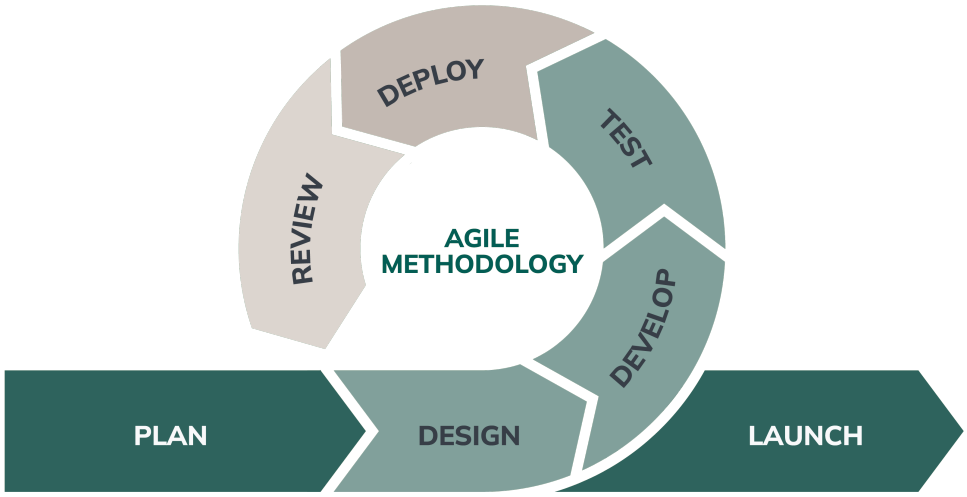


Figure 4.1: Agile Methodology

4.7 Selected Software Development Life Cycle for the Project

Although multiple methodologies were reviewed, Agile [17] methodology was chosen for this project. The iterative and collaborative nature of Agile ensured that evolving user requirements such as property search filters, image uploads and dashboard functionalities could be incorporated effectively. This made the development process flexible, efficient and aligned with real-world expectations for a real estate property listing system.

Chapter 5

Body of the Project

This chapter presents the comprehensive details of the project, where the reader can gain a full understanding of its scope and execution. It provides structured documentation covering all essential aspects of the system. Within this chapter, a number of key components are included, such as a comprehensive description of the work performed, requirement analysis, system analysis, as well as the design, development and testing processes. These elements collectively explain how the project was conceptualized, structured and implemented, ensuring clarity of both objectives and outcomes.

5.1 Work Description

Throughout my internship at IT WAY BD, I was engaged in the design and development of a Real Estate Property Listing System. The goal of this project was to develop a centralized and up-to-date platform for buyers, sellers, and agents to interact in an efficient manner. My contribution included both frontend development, where I created responsive user interfaces and backend development, where I implemented features such as authentication, property management, and database integration using Laravel and MySQL. Through this experience, I gained practical knowledge of professional web application development, improved my teamwork skills and learned how to solve real-world software challenges. The key activities and components I worked on are described below:

Frontend Development: I designed user-friendly landing pages, dashboards and navigation menus using HTML, CSS, JavaScript and Bootstrap. Special attention was given to ensuring was that the platform visually appealing, responsive and simple to utilize across all devices. The layouts were structured to guide users smoothly through property listings, search filters and account management features.

Backend Development: On the server side, I worked alongside the Laravel framework to implement secure login and user authentication. I also developed modules for managing user accounts and property data, integrating them with the MySQL database. CRUD operations were applied to ensure smooth and reliable data handling. These implementations helped

maintain both system security and consistency of property information.

Property Management: I contributed to building modules that allowed one to create, update, and manage property listings. Each property entry included details such as descriptions, pricing, images, type of property and availability status. The module was designed to be flexible and scalable, making it easier for users to maintain accurate property records.

Search and Filtering: I assisted in implementing advanced filtering features that enabled buyers to explore properties according to various criteria, including location, budget, number of rooms, and type of property. This feature was critical for improving the buyer's experience, as it reduced the time spent browsing irrelevant listings and allowed users to quickly find suitable options.

Role-Based Access Control: I worked on defining and managing user roles, including Admin, Agent and Buyer. Each role was provided with access rights appropriate to its responsibilities. For example, Admins had full control over listings and user management, while Buyers only had access to browsing and inquiry functions. This ensured a secure and structured workflow within the system.

Approval Workflow: I supported the creation of a property approval system where Admins could review and verify property listings before they were published on the platform. This feature was introduced to maintain data accuracy, reduce fraudulent postings and build trust among users.

Responsive Design: I ensured that the entire system was fully responsive and worked seamlessly across desktops, tablets and smartphones. Bootstrap was used extensively to maintain design consistency, while testing was performed on different devices to confirm usability and accessibility.

Communication System: I contributed to features that enabled buyers to send inquiries directly to agents within the system. This reduced reliance on external communication methods and streamlined the property negotiation process, making transactions faster and more efficient.

Testing and Validation: To ensure system reliability and security, I engaged in multiple testing phases, including unit testing, responsiveness testing, as well as security testing. I implemented validation checks for input fields, secured database queries against SQL injection and tested workflows such as property submission, approval and search.

In summary, the tasks I undertook throughout this internship contributed to the development of a secure, scalable, and user-friendly property management system that modernizes the real estate transaction process. This experience not only improved my technical skills but also provided me with the chance to engage in practical software solutions that tackle genuine industry challenges.

5.2 Requirement Analysis

Rich Picture

The rich picture of the Real Estate Property Listing Platform shows how different stakeholders interact with the system. Buyers search, view listings and send inquiries, while Agents submit properties and manage their portfolios. The Admin oversees the platform by approving or rejecting listings, managing access and reviewing reports. At the core, the system connects with a database that stores property details, user information and inquiries, ensuring accurate and updated data. This representation highlights the overall flow of information and responsibilities, making the system transparent and efficient.

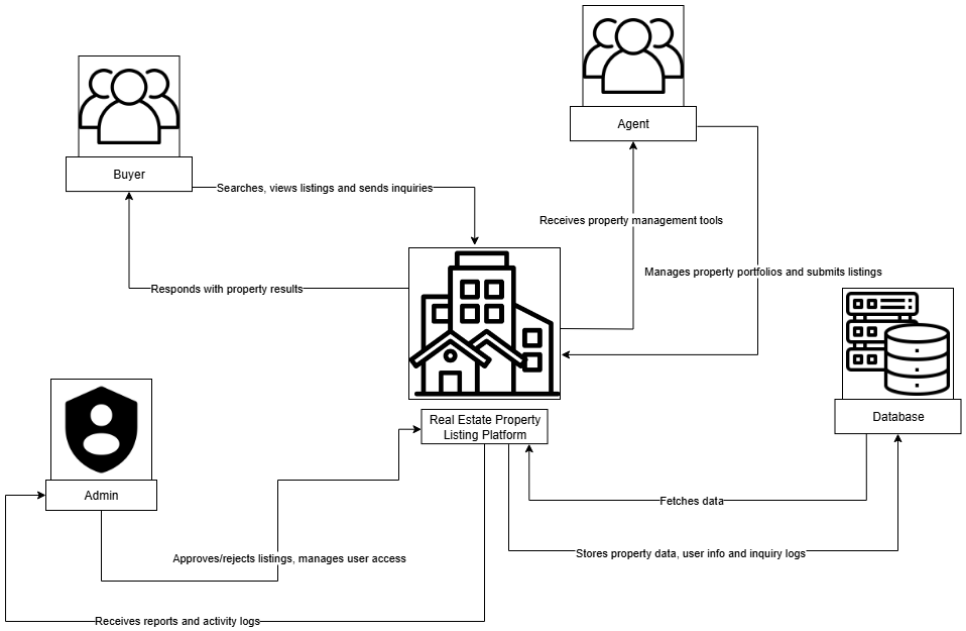


Figure 5.1: Rich Picture

Functional Requirements

The core features of the system are encapsulated in the functional requirements, which directly address the specific issues faced by the real estate sector. These requirements ensure that the system facilitates a smooth interaction platform for property agents, buyers, and administrators. Below are the main functional requirements:

User Registration and Authentication: The system must allow different types of users (Admin, Buyer, Agent) to register and log in securely. A system of verification must be established to guarantee that only authorized individuals can set up accounts.

Role-Based Entrance Control: Each user role should have different levels of access. Admin Can approve or reject property listings, manage users and monitor overall activity. Agent Can create, update and manage property listings. Buyer Can browse and search for properties, send inquiries and save favorites.

Property Listing Management: Agents must be able to add properties with all relevant details such as property title, description, price, images, size, location and availability. They should also be able to update or delete listings when needed.

Property Approval Workflow: Every property listing must go through an admin approval process before being published. This ensures authenticity, prevents fraudulent activities and maintains trust in the platform.

Search and Filtering Mechanism: Buyers ought to have the capability to explore properties by utilizing sophisticated filters, including:

- Range of prices
- Type of property
- Location
- Count of rooms and dimensions of the property

Inquiry and Communication System: A built-in communication system must allow buyers to send inquiries about specific properties directly to agents. Agents should receive notifications and be able to reply within the platform, reducing dependency on external communication methods.

User Dashboards: Each role must have a dedicated dashboard for managing their activities. Admin Dashboard Manage users, approve/reject listings, oversee transactions. Agent Dashboard Manage personal listings, inquiries and track engagement with properties. Buyer Dashboard Manage saved properties, track inquiries and view property details.

Responsive Website: The system should be reachable from various devices, including desktops, tablets, and smartphones. An adaptable design guarantees that users can engage with the system at any time and from any location.

Non-Functional Requirements

Non-functional requirements establish the quality benchmarks and limitations within which the system is required to function. These specifications guarantee that the system is not merely operational but also dependable, secure, and user-friendly. The non-functional requirements for the Real Estate Property Listing System are detailed below:

Performance: The system must possess the ability to manage multiple concurrent users without noticeable delays. Search queries must return results within two seconds to ensure efficiency.

Scalability: The system should be structured to accommodate a growing number of users and property listings. Built on the Laravel framework, it should allow for future extensions such as:

- Payment gateway integration

- Rental management features
- Map-based property search

Security: Security is a major concern in property transactions. The system must:

- Store passwords in encrypted format
- Protect against SQL injection, cross-site scripting and CSRF attacks
- Enforce role-based authentication to prevent unauthorized access

Reliability and Availability: The platform must ensure high availability, with a target uptime of 99 percent once deployed on a live server. Regular data backups should be scheduled to avert the loss of data in the event of system malfunctions.

Usability: The system should be designed to be intuitive and user-friendly. Interfaces need to feature clear navigation, consistent layouts, and user-friendly forms, enabling individuals with limited technical skills to utilize the system efficiently.

Maintainability: The codebase should follow Laravel's MVC architecture for better separation of concerns. Proper documentation, code comments and version control (Git) must be maintained to ensure future developers can easily update and extend the system.

Portability: The application should run seamlessly in both local (XAMPP) and cloud-hosted environments.

Ethical and Sustainability Considerations: The platform must ensure ethical handling of user data, protecting privacy at all times. Only verified properties should be listed to maintain trust. Furthermore, the system should be optimized to minimize server overhead, supporting sustainable computing practices.

5.3 System Analysis

The System Analysis of the Real Estate Property Listing System delivers an extensive review of its design, feasibility and effects. The Six-Element Analysis specifies the essential framework, addressing key inputs like property and user data, crucial processes such as authentication and property management, outputs that include listings and reports, control mechanisms for access and validation, feedback through alerts, and a technical environment developed with Laravel, SQL and a responsive design. The Feasibility Analysis affirms the system's technical reliability, operational feasibility, economic practicality and achievable timeline, bolstered by open-source tools and established milestones. The Problem–Solution Analysis confronts challenges in traditional real estate management such as inefficiency, data fragmentation and communication delays—through centralized data management, role-based approvals, advanced search capabilities and automated notifications. The Effect and Constraints Analysis underscores improved transparency, efficiency and decision-making while recognizing limitations like

resource demands, scalability and data accuracy. Overall, the system is a trustworthy and scalable platform designed to enhance property management and user experience in the real estate field.

5.3.1 Six Element Analysis

The table provides a Six-Element Analysis of a Real Estate Property Listing System. It details the essential components of the system: inputs that include property and user data; processes such as authentication, listing management and interactions with buyer agents; and outputs that feature approved listings, search results and dashboards. Control mechanisms are implemented to ensure data security and suitable role-based access, while feedback consists of system notifications and alerts for both users and administrators. In conclusion, it identifies the environment as a web-based responsive platform developed using Laravel and SQL, designed for the functioning of an online real estate marketplace.

Element	Description
Input	Property details (title, description, location, price, size, images, availability); user credentials (Admin, Agent, Buyer); buyer search/filter queries; and inquiry messages.
Process	Role-based login and authentication; property listing creation and management by Agents; admin approval workflow; buyers searching and filtering properties; inquiry and response handling between buyers and agents.
Output	Approved property listings visible to buyers; filtered property search results; user dashboards (listed properties, saved properties, inquiries); and admin reports on overall system activity.
Control	Role-based access control (Admin, Agent, Buyer); property approval process; input validation on listing forms; backend rules to ensure consistency and security of stored data.
Feedback	System notifications on property submission, approval and updates; error messages for invalid input; email/onsite alerts for inquiries and responses; admin alerts for pending approvals.
Environment	Web-based responsive system accessible via desktop, tablet, and mobile; implemented with Laravel framework and SQL database; designed for online real estate marketplace usage.

Table 5.1: Six Element Analysis of Real Estate Property Listing System

5.3.2 Feasibility Analysis

A feasibility study was carried out to assess the viability of creating and executing the Real Estate Property Listing System during the internship period. The study considered technical, operational, economic and scheduling perspectives to ensure the system could be delivered effectively and meet its objectives.

5.3.3 Technical Feasibility

Technical viability is central to the Real Estate Property Listing System, which employs a Laravel backend, an SQL-based relational database for structured storage, and a Bootstrap-enabled responsive frontend. This evaluation verifies whether the current technology stack and infrastructure are adequate for a successful implementation. It considers hardware capacity ,

software compatibility, and integration capabilities . To ensure reliable performance, we confirm that servers and client devices meet the framework requirements; that the OS, runtime, and database versions are supported; and that data models, migrations, and indexing strategies can sustain listing volume, search, and concurrent sessions. Moreover, security controls and backup procedures are validated. This review establishes the system's technical feasibility, ensuring stability, scalability, and secure data handling for a production-ready real estate platform.

5.3.4 Operational Feasibility

Operational feasibility emphasizes the extent to which the platform integrates into the daily workflows of stakeholders within the real estate sector. Agents are required to generate and oversee listings that include images, pricing, and availability; Buyers need to conduct searches, apply filters, and make inquiries effortlessly; and Admins are tasked with the approval or rejection of listings while monitoring user activities. This assessment examines user readiness, ease of adoption and minimal training needs through a clean, consistent UI and intuitive dashboard flows. It also evaluates whether the system improves decision-making and efficiency—faster listing approval cycles, reduced back-and-forth via in-platform inquiries and clearer visibility into listing status and performance. Documentation, quick-start guides and role-specific help content support smooth onboarding. Ultimately, operational feasibility confirms that the platform integrates with existing practices, reduces manual steps, enhances transparency and promotes reliable, data-informed actions across roles.

5.3.5 Economic Feasibility

Economic viability guarantees that the solution is both financially responsible and sustainable. The project emphasizes open-source technologies to avoid licensing fees. Expected costs include hosting, storage for images, domain and modest maintenance; during the internship phase, human-resource costs are minimal. A simple cost–benefit review weighs these predictable expenses against tangible gains: streamlined listing operations, faster time-to-market for properties, improved lead handling via inquiries, and reduced overhead from manual coordination. Over time, operational efficiencies and higher listing throughput translate to favorable ROI. This analysis supports the conclusion that the platform is financially viable for both initial deployment and incremental growth.

5.3.6 Schedule Feasibility

The assessment of schedule feasibility determines if the project can be finalized within the duration of the internship by employing a systematic, step-by-step approach. Milestones are defined for requirements and schema design, authentication and role management, listing CRUD with image handling, admin approval workflow, search and filtering, inquiry messaging, integrated notifications, testing and deployment. Each milestone includes clear deliverables and review criteria to manage scope and reduce rework. Resource availability is checked early

to mitigate risks. By executing in short iterations with continuous testing and feedback, the project remains on track and achieves stable, demonstrable functionality within the allotted period. This validates the schedule as realistic and achievable for the Real Estate Property Listing System.

5.3.7 Problem Solution Analysis

The analysis of problem solutions involves a comprehensive investigation into the issues that the Real Estate Property Listing System aims to resolve, as well as the methods and strategies employed to tackle these challenges. The analysis focuses on the unique issues in traditional real estate management, particularly inefficiencies in property listing, search, communication and approval processes. The Real Estate Property Listing System tackles several significant challenges associated with conventional real estate management. In the past, property information was recorded manually using spreadsheets or paper-based methods, leading to inefficiencies, data redundancy, and delays in updates and communication. The lack of a centralized platform resulted in disjointed operations among buyers, agents and administrators, which in turn limited visibility regarding property availability. Furthermore, the absence of a formal approval process heightened the risk of inaccurate or fraudulent listings, while communication remained sluggish due to dependence on phone calls and emails. Additionally, the use of non-standardized listing formats and unresponsive interfaces adversely affected user experience, particularly on mobile devices. To address these challenges, the system implements a centralized SQL-based database for uniform data management, a role-based access and administrative approval workflow to ensure accuracy and reliability and sophisticated search and filtering tools for expedited property discovery. It also features an integrated inquiry and notification system to facilitate timely communication among users, along with a responsive, user-friendly interface developed with Bootstrap to improve accessibility across all devices.

5.3.8 Effect and Constraints Analysis

Impact and Limitations of the Real Estate Property Listing System provide a balanced understanding of the benefits the system delivers as well as the constraints that may affect its use. This analysis highlights the positive effects achieved through digitization of the property listing process, alongside the limitations that must be addressed for long-term sustainability.

Positive Effects:
The Real Estate Property Listing System delivers several positive effects that enhance overall efficiency, transparency and user satisfaction within the real estate domain:

Improved Transparency and Trust: By centralizing listings and requiring admin approval, the system reduces fraudulent entries and enhances credibility in property transactions.

Efficient Property Management: Agents can easily create, update and manage listings, while buyers benefit from advanced search and filter features, leading to streamlined workflows.

Faster Communication: Built-in inquiry and notification mechanisms ensure timely interaction between buyers and agents, reducing delays caused by manual communication.

Better Decision-Making: Dashboards and role-based access provide stakeholders with real-time visibility into listings and inquiries, enabling informed decisions about pricing, availability, and approvals.

Accessibility and User Experience: A responsive design guarantees that users can utilize the platform on various devices, thereby enhancing both adoption and convenience.

Constraints:
The Real Estate Property Listing System also faces certain constraints that must be addressed to ensure long-term efficiency and sustainability:

Resource Limitations: While open-source frameworks minimize costs, the platform still requires hosting, storage, and regular maintenance, which could be challenging for small agencies with limited resources.

Integration Complexity: Future integration with APIs may involve technical challenges and additional development effort.

Data Quality and Verification: The system's reliability depends on accurate data entry by sellers and agents. Incomplete or incorrect listings can undermine user trust.

User Training and Adoption: Some users may resist adopting digital tools if they are accustomed to manual processes. Basic training and awareness programs may be required.

Scalability: As the number of properties and users grows, database performance and search speed could become a limitation, requiring optimization or scaling of infrastructure.

The Real Estate Property Listing System brings significant improvements in efficiency, transparency, and decision-making for all stakeholders, while its long-term success depends on addressing integration, resource and scalability constraints.

5.4 System Design

UML Class Diagrams

Based on the Unified Modeling Language (UML), the class diagram of the Real Estate Property Listing System illustrates the structural design of the application by defining its classes, attributes, methods, and their interrelationships.

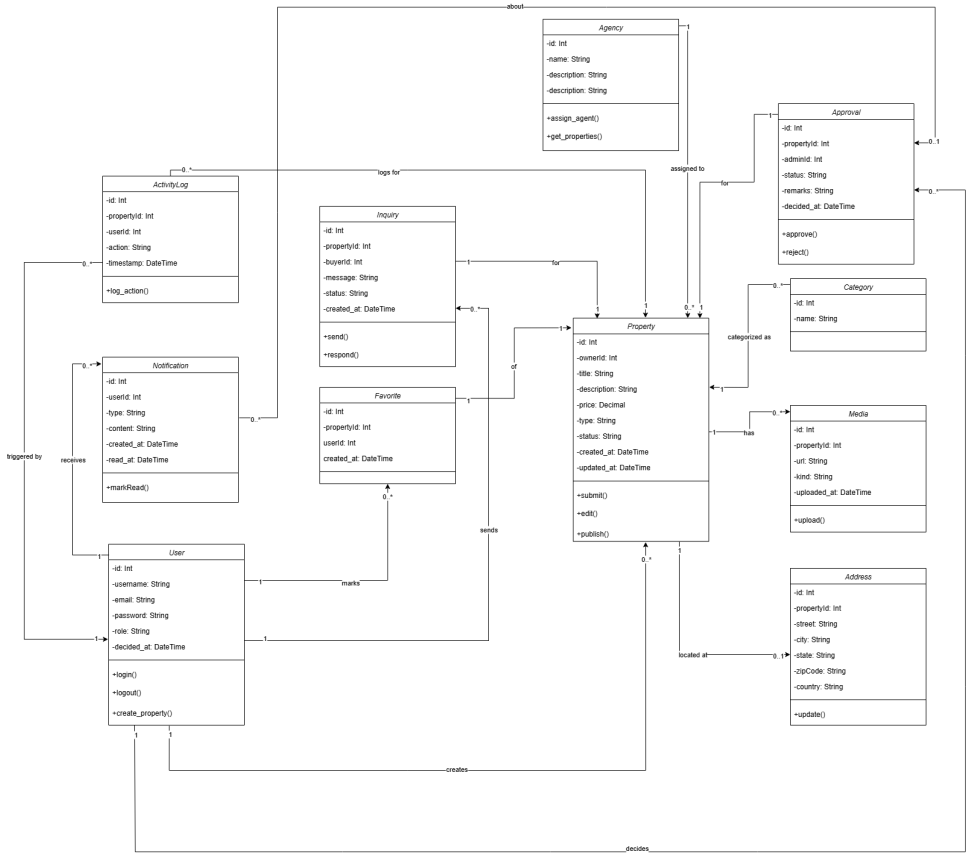


Figure 5.2: UML Class Diagrams

The diagram highlights the main entities such as User, Property, Agency, Category, Media, Address, Inquiry, Approval, Favorite, Notification and Activity Log, along with their respective roles within the system. Each class specifies its data fields and functions, while associations, dependencies and multiplicities show how these components interact—for instance, users create and manage properties, buyers send inquiries, admins approve listings, and notifications are triggered for system events. By capturing these relationships visually, the class diagram provides a clear and standardized blueprint of the system's architecture. This representation not only simplifies communication between stakeholders, developers and designers but also assists in understanding, maintaining, and extending the software in a structured and efficient manner.

Use Case Diagram

This UML Use Case Diagram of the Real Estate Property Listing System illustrates how different user roles interact with the application.

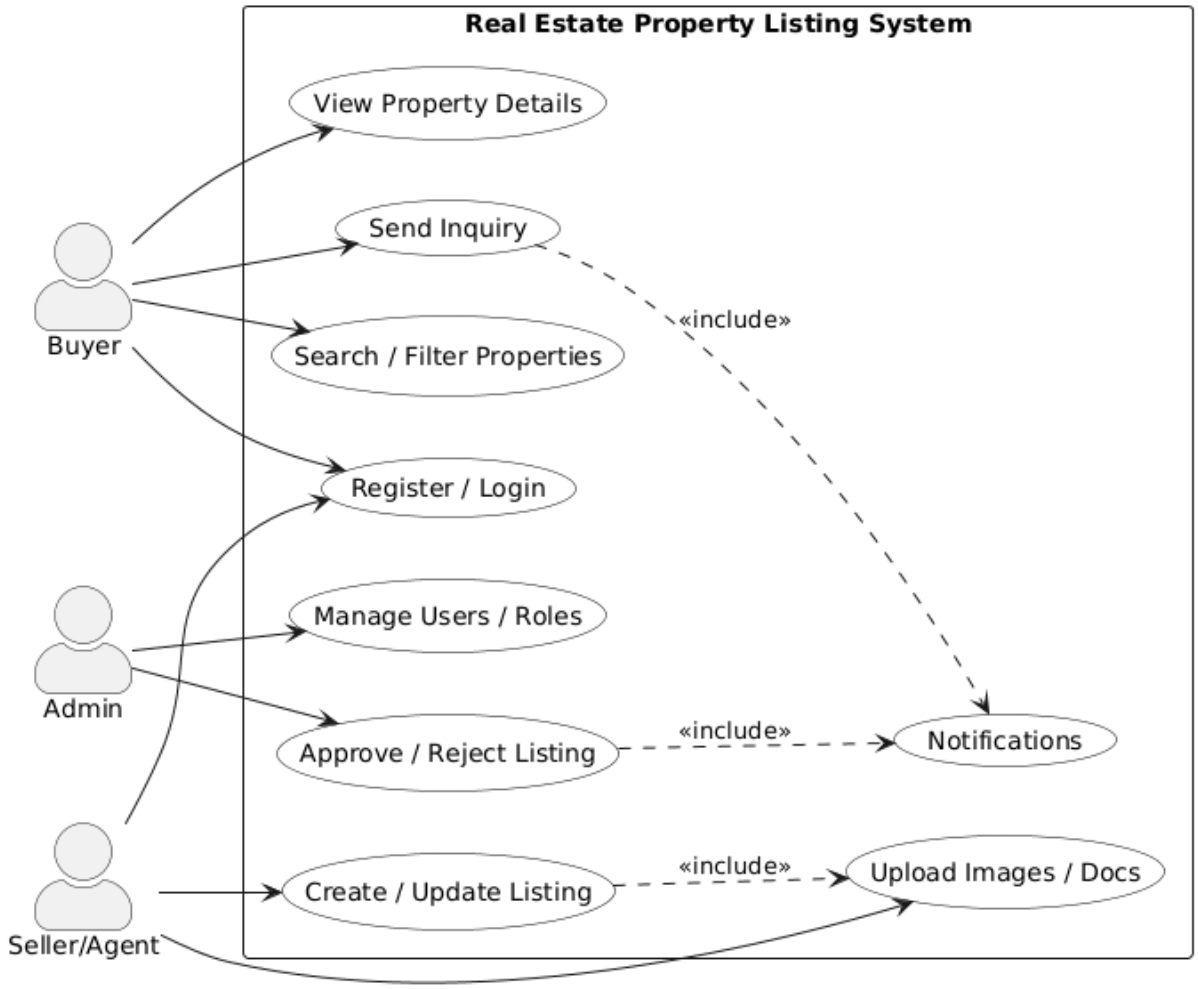


Figure 5.3: UML Use Case Diagram

Buyers are able to register or log in, search and filter properties, view property details and send inquiries, which include generating notifications. Agents create and update property listings, along with uploading images or documents. Admin manage users and roles, as well as approve or reject listings, which also trigger notifications. The diagram provides a clear representation of the system's functional requirements and the interactions between actors and use cases.

Architecture

The architecture forms the backbone of any system, defining its core structure and functionality. In the context of a website, the architecture outlines how the site operates, including the flow of data and how it will be stored and accessed.

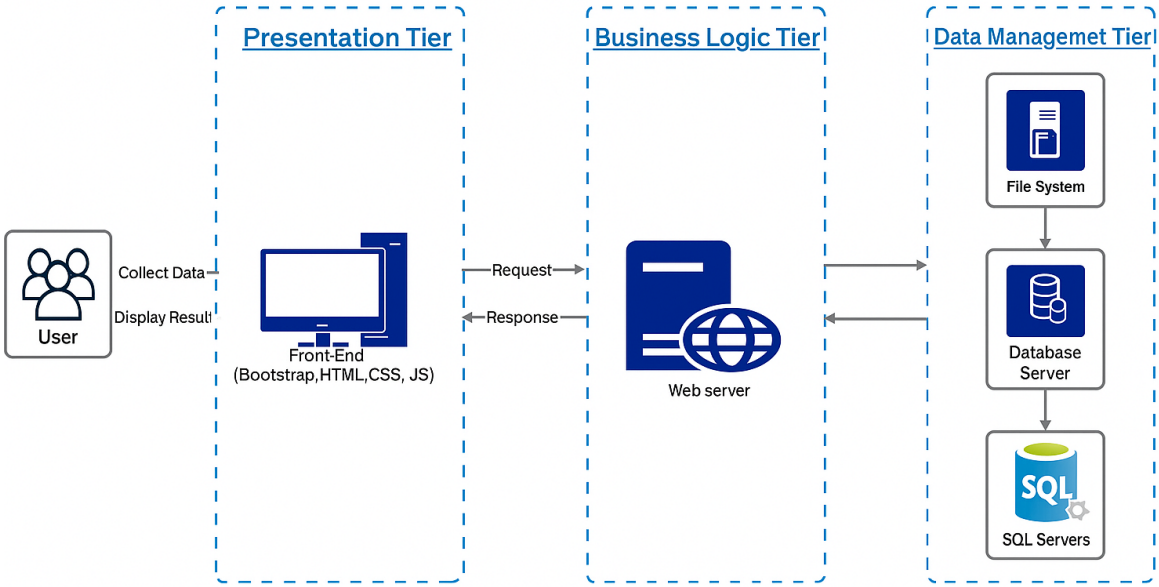


Figure 5.4: Architecture of Real Estate Property Listing Platform

The architecture of our project highlights the exclusive interaction of users with the front-end interface. The front end processes user commands and sends them to the web server, which retrieves data from the file system or database. The server then processes this data and returns the suitable response to the front end for the user to see.

5.5 Implementation

Implementation is the phase of a project in which a planned system or solution is created and implemented. It involves coding, configuring, and integrating components according to project specific requirements .Turning requirements into a working product or system. The following are the pictures of the implementation phase:

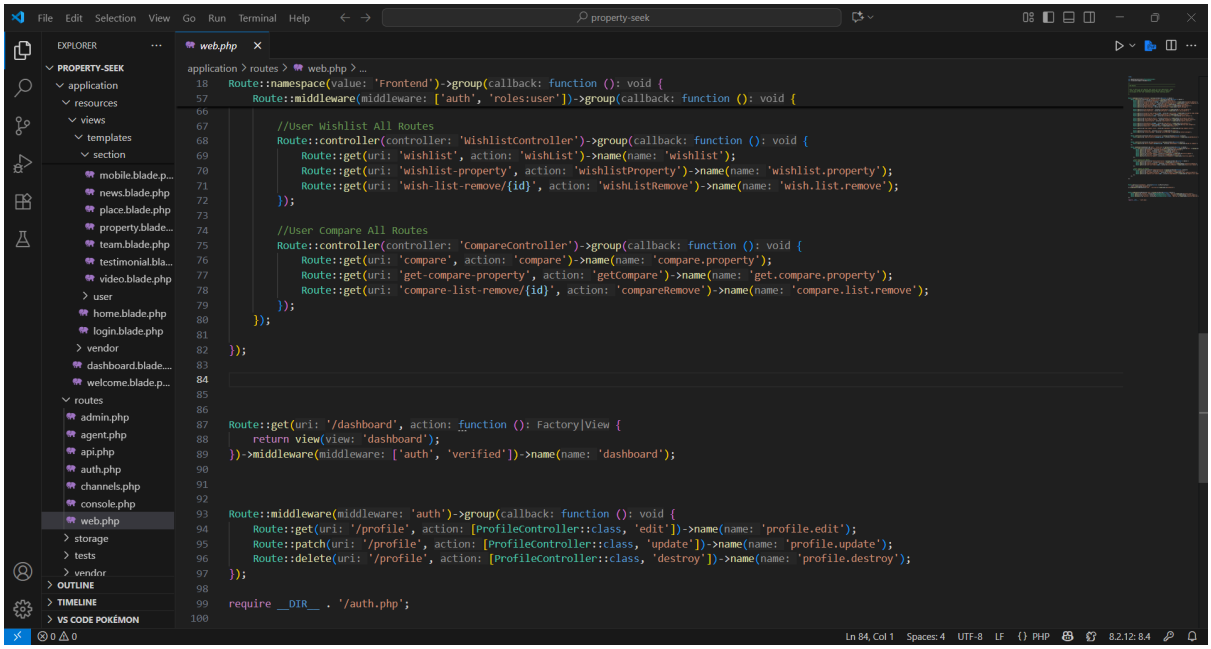


Figure 5.5: Coding Implementation

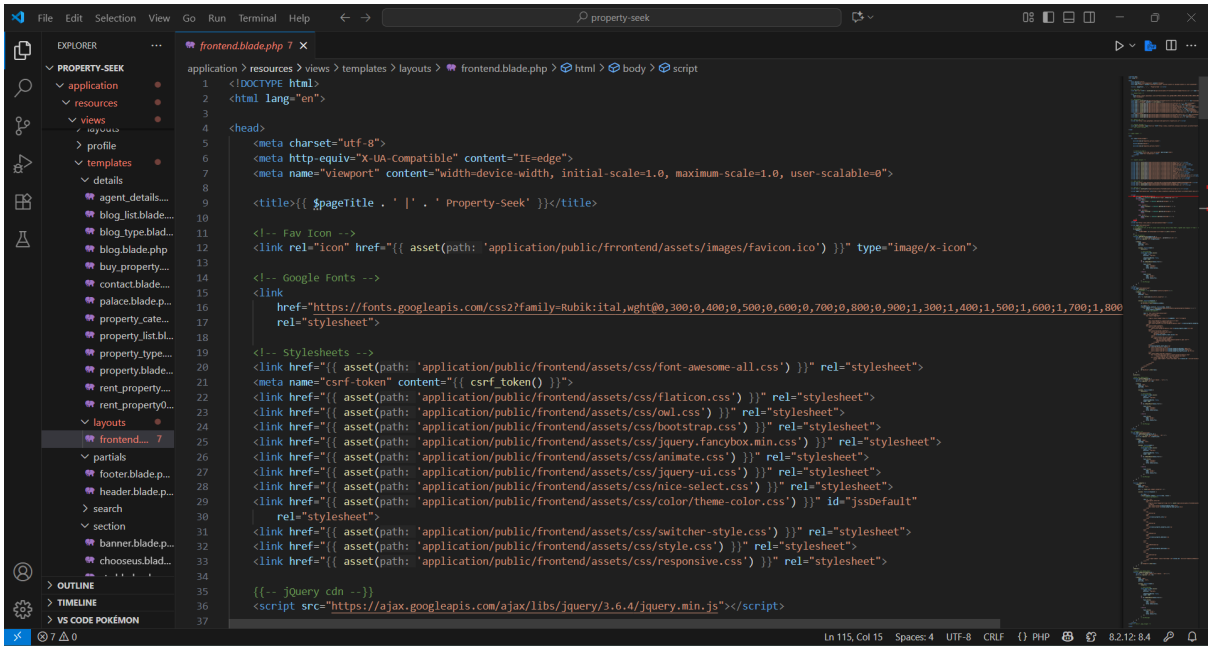


Figure 5.6: Coding Implementation

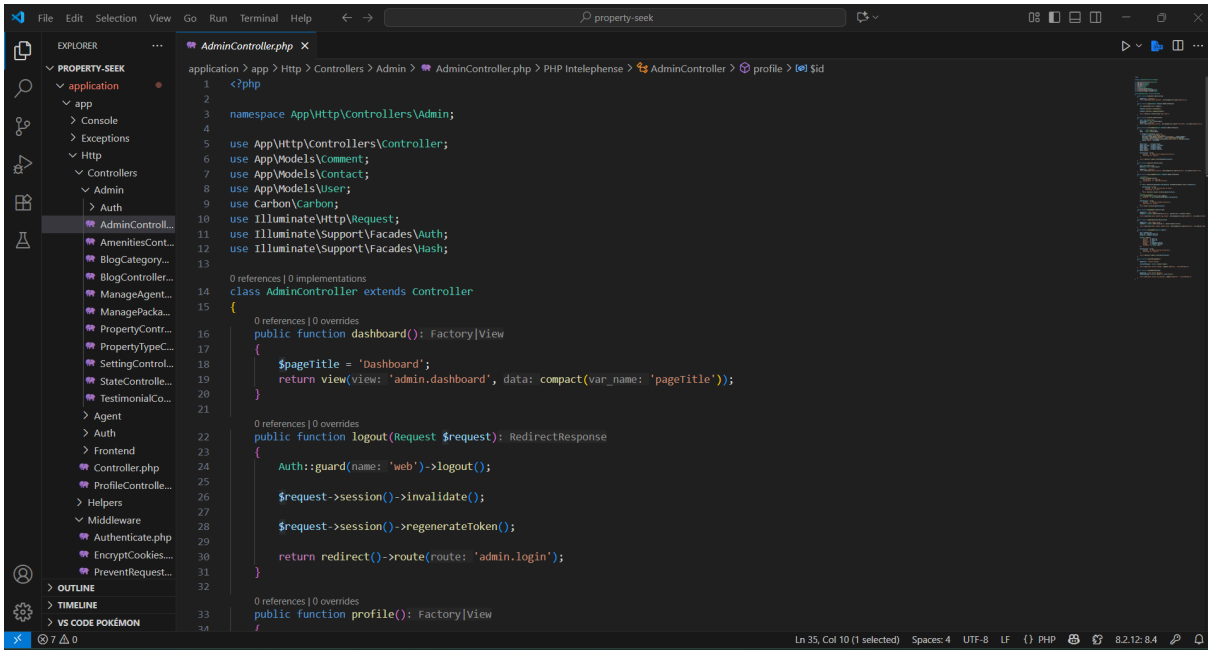


Figure 5.7: Coding Implementation

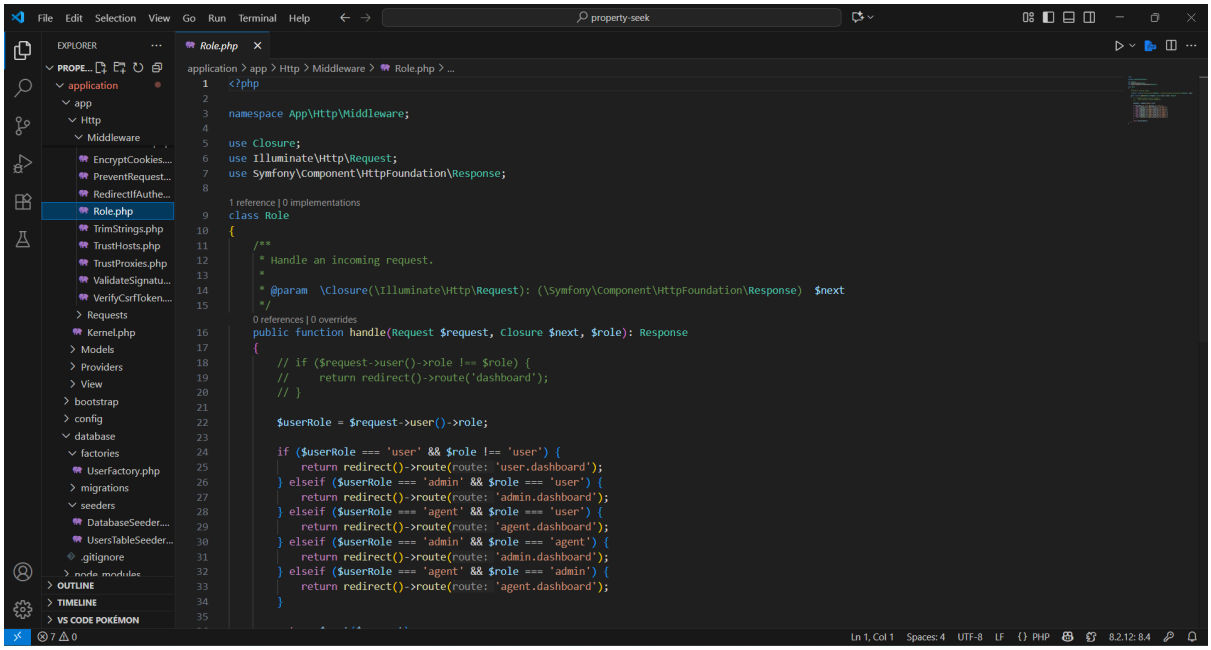


Figure 5.8: Coding Implementation

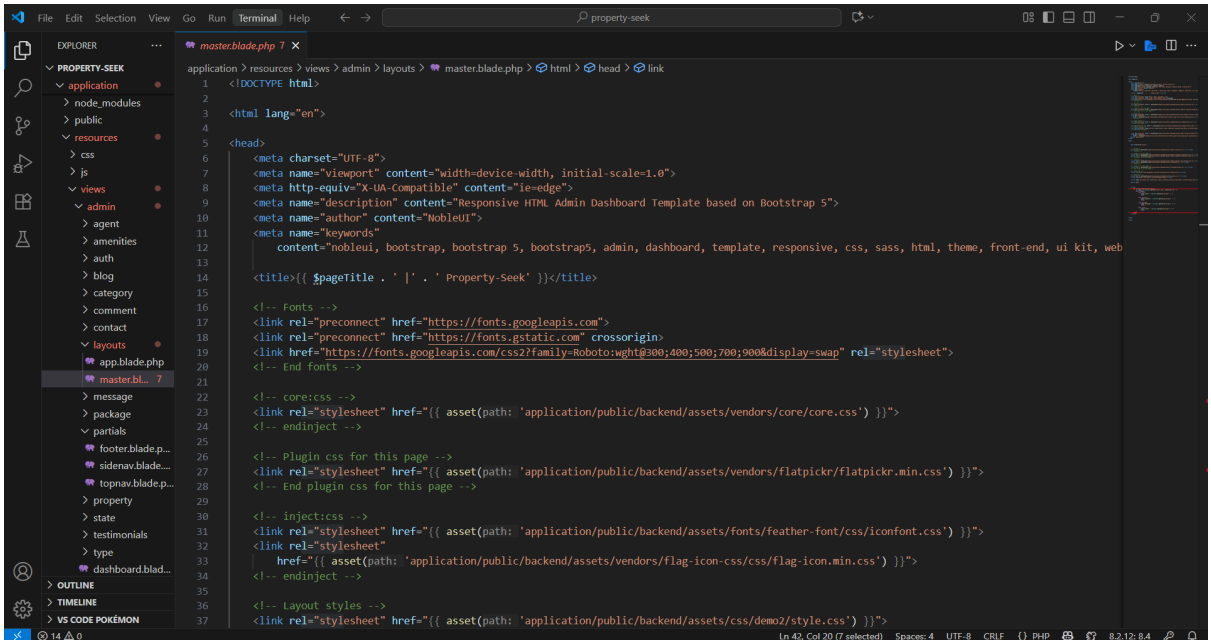


Figure 5.9: Coding Implementation

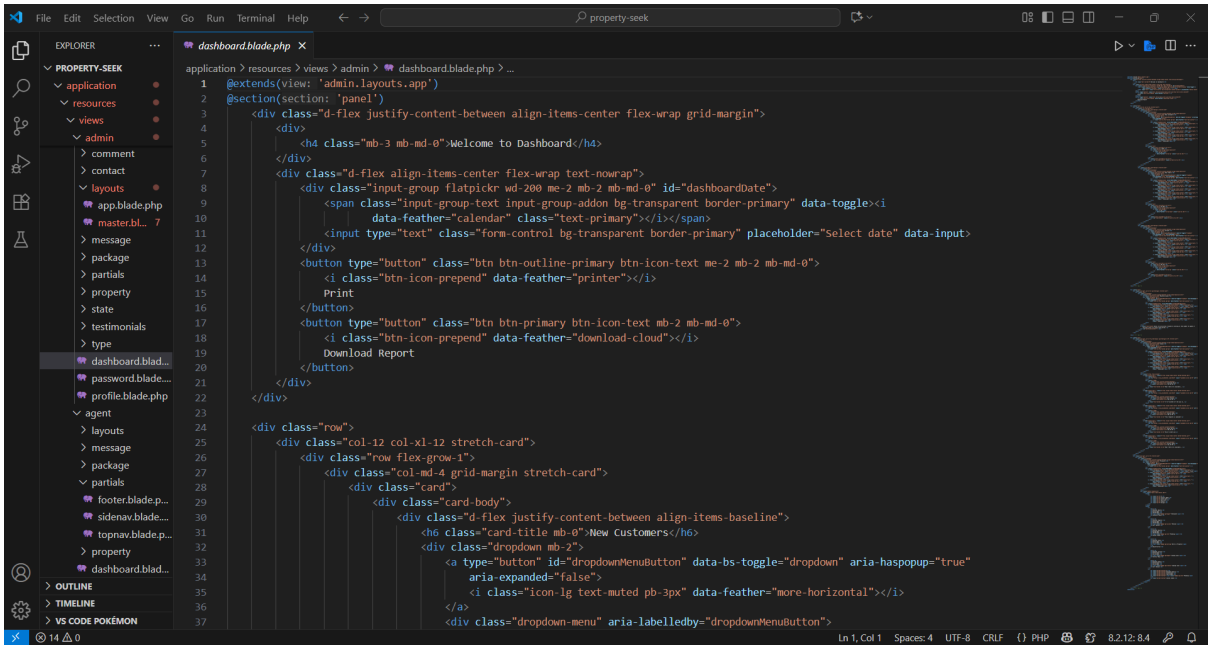


Figure 5.10: Coding Implementation

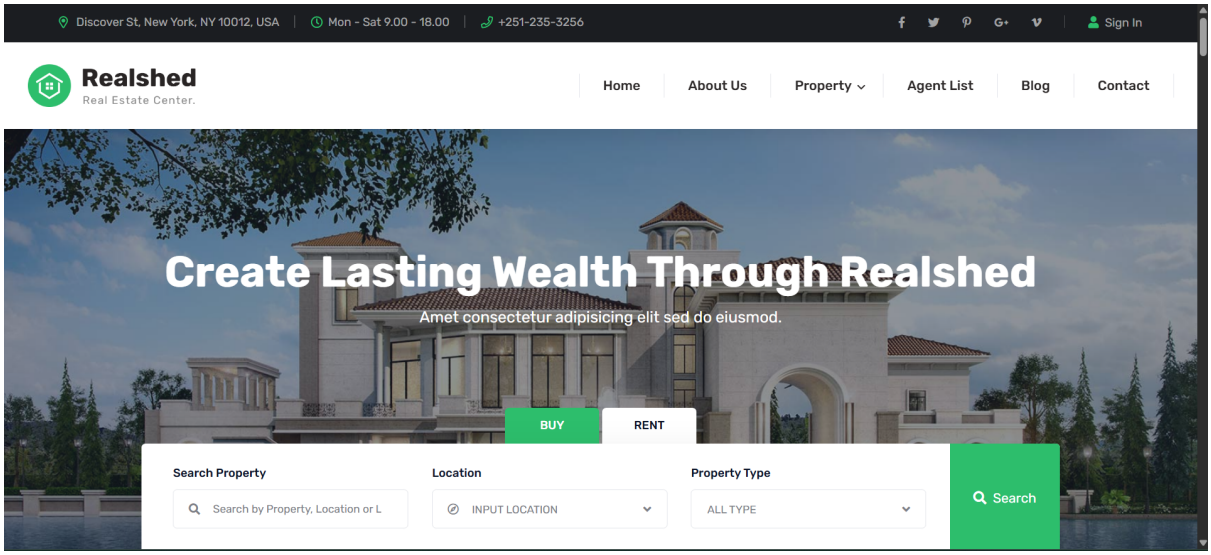


Figure 5.11: Website Implementation

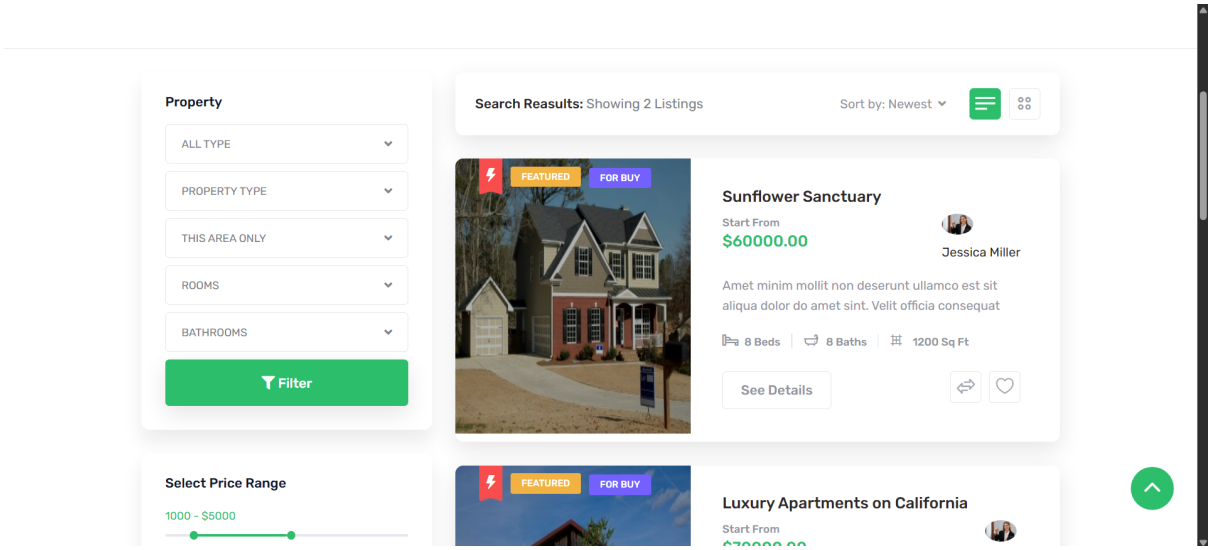


Figure 5.12: Website Implementation

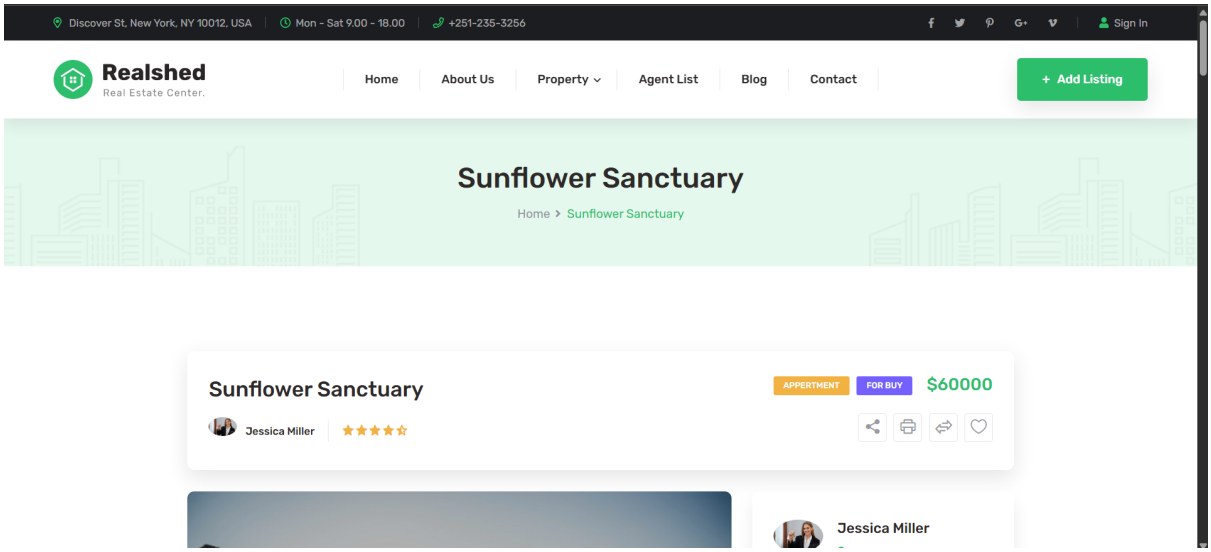


Figure 5.13: Website Implementation

Figure 5.14: Admin Dashboard Implementation

Figure 5.15: Agent Dashboard Implementation

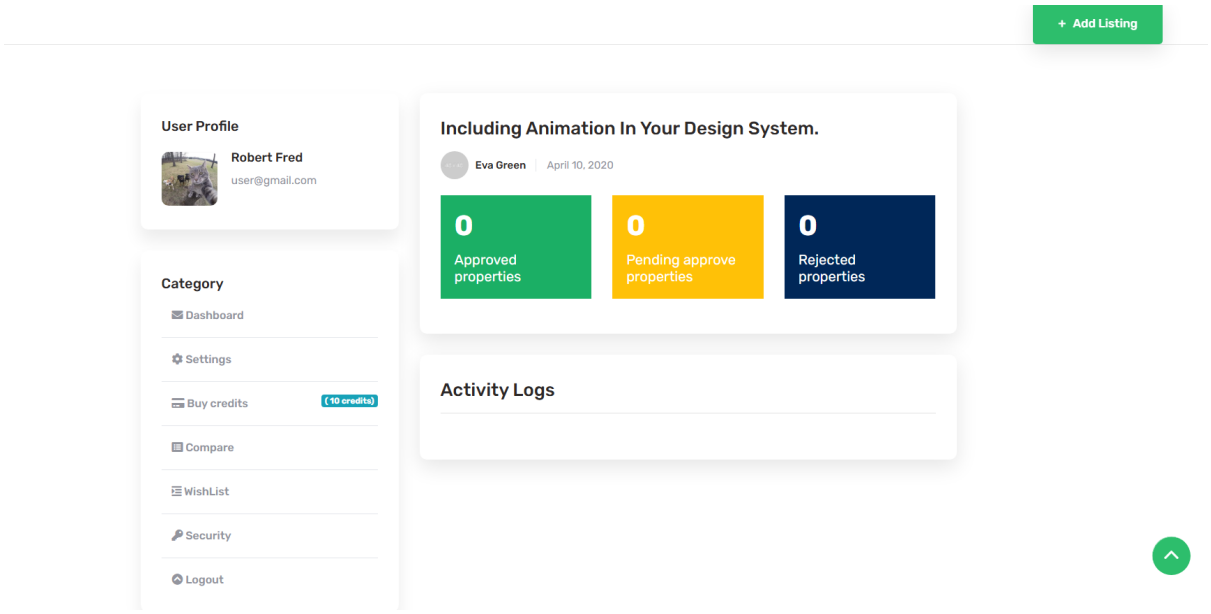


Figure 5.16: User Dashboard Implementation

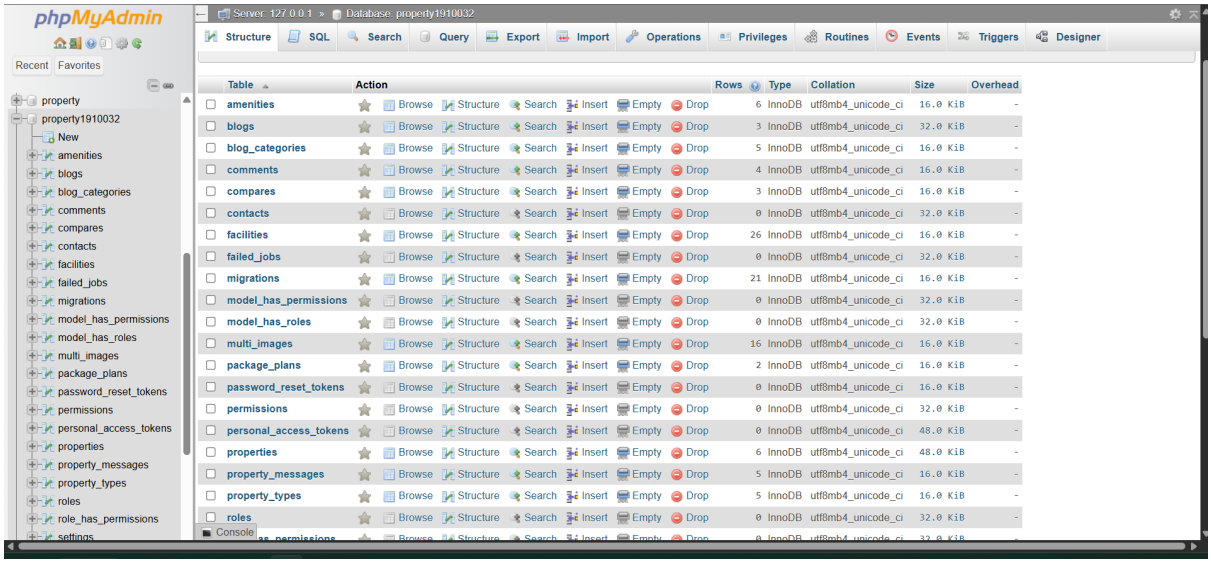


Figure 5.17: Database Implementation

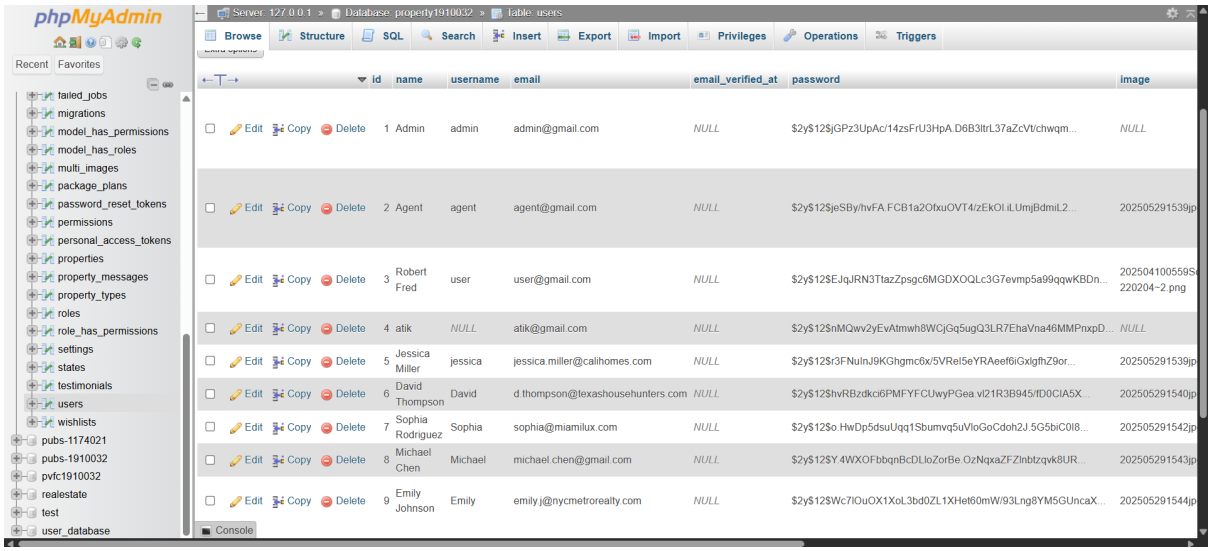


Figure 5.18: Database Implementation

5.6 Testing

Testing is an important step to ensure that the Real Estate Property Listing System satisfies all functional and non-functional requirements. This section details the test inputs, expected outputs, the design of test cases, and the results that were obtained.

Input

The inputs for testing included:

- User credentials for login (admin, agent, buyer).
- Property listing details (title, description, price, images, location, size).
- Search and filter parameters (price range, property type, location, number of rooms).
- Inquiry messages sent by buyers to agents.

Output

The expected outputs from the system were:

- Successful authentication for valid users; rejection for invalid credentials.
- Property listings saved, displayed, and updated correctly in the database.
- Accurate search/filter results based on provided parameters.
- Inquiries delivered to the correct agent and confirmation notifications to the buyer.

Designing Test Cases

Test cases were formulated to validate both the functional and non-functional dimensions of the system:

Login Test: Input valid/invalid credentials and verify authentication handling.

Property Listing Test: Add, update and delete a property record; verify database consistency.

Search and Filter Test: Input multiple filter combinations and confirm correct property results.

Inquiry Test: Send an inquiry as a buyer; ensure the seller receives it and the buyer gets confirmation.

Role-Based Access Test: Verify that only admins can approve/reject listings and unauthorized users are restricted.

Test Results

The outcomes of the executed tests included:

Login Functionality: All valid users logged in successfully; invalid inputs were rejected as expected.

Property Listings: Listings were added, displayed and updated accurately; deletion removed records from the system.

Search and Filtering: Queries returned correct and relevant results within acceptable response times.

Inquiries: Buyer inquiries reached intended agents and confirmation messages were successfully generated.

Role-Based Access: Access restrictions worked correctly, ensuring only authorized users performed sensitive actions.

Test Case Table

The table displays sample test cases formulated to assess the functionality and dependability of the Real Estate Property Listing System. Each test case specifies particular inputs, anticipated outcomes, actual results and their status to guarantee that the system operates as intended. The tests encompass essential features such as user authentication, property addition, search and filtering, inquiry management and role-based access control. Security and performance are also scrutinized through scenarios like SQL injection prevention and concurrent user searches. All test results indicate that the system fulfills expectations, thereby affirming its stability, security and efficient performance.

Test ID	Input	Expected Output	Actual Output	Status
TC-01	Legitimate and illegitimate login credentials	Successful authentication for legitimate users; error notification for unauthorized users.	Matches expected behavior	Pass
TC-02	Add a new property with all required fields	Property saved in database and displayed in listings	Property saved and displayed correctly	Pass
TC-03	Search by location and price filter	Display only properties matching criteria	Correct results returned	Pass
TC-04	Buyer sends inquiry to seller	Seller receives inquiry; buyer receives confirmation	Both notifications delivered successfully	Pass
TC-05	Agents attempts to approve property (restricted role)	Access denied; only admin can approve	Access correctly restricted	Pass
TC-06	Multiple users perform searches simultaneously	System responds within acceptable time	No delay in response time	Pass
TC-07	Attempt SQL injection in login form	System rejects malicious input	Input validation prevents attack	Pass

Table 5.2: Sample Test Cases for Real Estate Property Listing System

Test Result Summary

Metric	Value
Number of Test Cases	7
Successful	7
Unsuccessful	0
All Status	All functionalities passed successfully

Table 5.3: Test Result Summary for Real Estate Property Listing System

The testing phase demonstrated that the Real Estate Property Listing System is not only functional and stable but also adheres to essential security standards. It performed effectively across all defined user roles: buyers, agents and admin ensuring that each role could carry out its intended tasks without errors. Overall, the results confirm that the system meets its requirements and provides a reliable, user-friendly experience.

Chapter 6

Results & Analysis

During testing of the Real Estate Property Listing System, several issues were identified. These were mostly minor bugs related to data validation and user navigation, and they were fixed during the debugging process. Test cases were prepared in detail, and once the problems were resolved, the system worked as expected. All test cases were supported by standard testing methodologies. Testing was initially performed on a local server. Once the project is hosted online, further testing will be conducted on the live server to verify stability under real usage. Some additional integrations will be included later, so the testing process is not yet fully complete. However, the features implemented so far are fully functional and stable. User-based live testing with a wider group of real estate agents and buyers has not yet been completed. This will be done in the deployment phase, as it requires actual user onboarding and feedback.

6.1 Graphical User Interface Result

The graphical user interface (GUI) plays a central role in system usability. The Real Estate Property Listing System provides a responsive and modern design that adapts to desktops, tablets, and mobile devices. The GUI ensures a user-friendly experience for Admins, Agents, and Buyers with role-specific dashboards. Below are sample interfaces from the system:

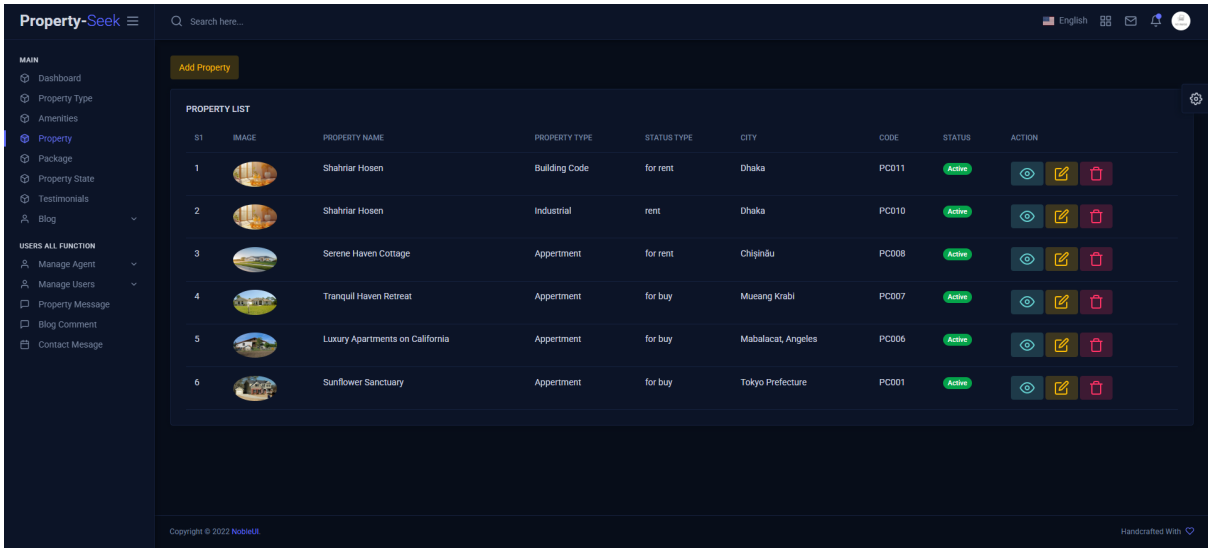


Figure 6.1: Admin Add Property Dashboard

Figure 6.2: Admin Create Property

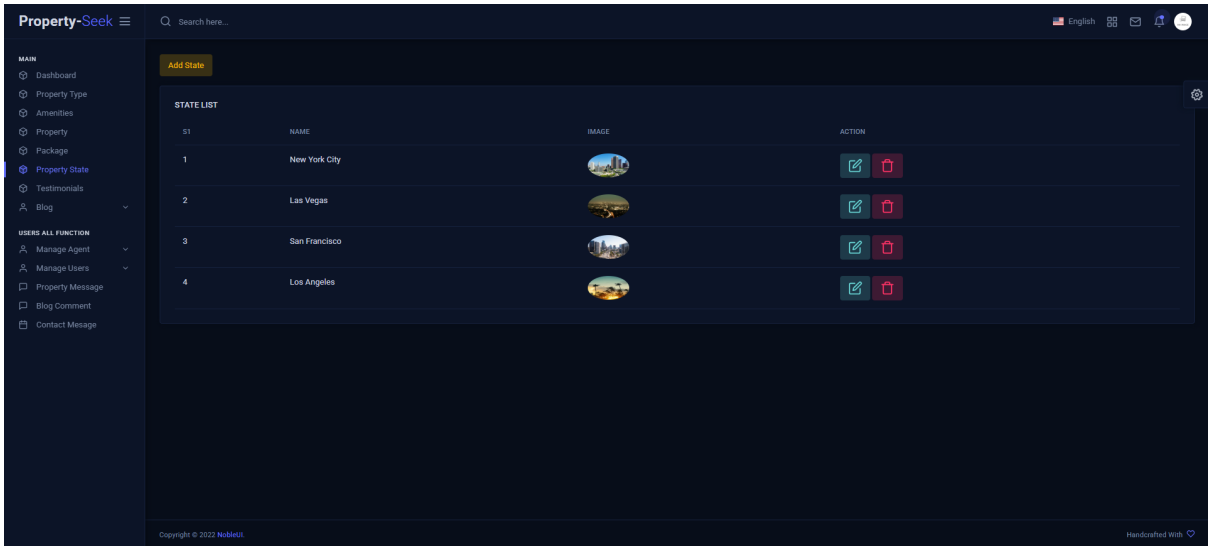


Figure 6.3: Admin State Viewing Dashboard

Figure 6.4: Creating New State

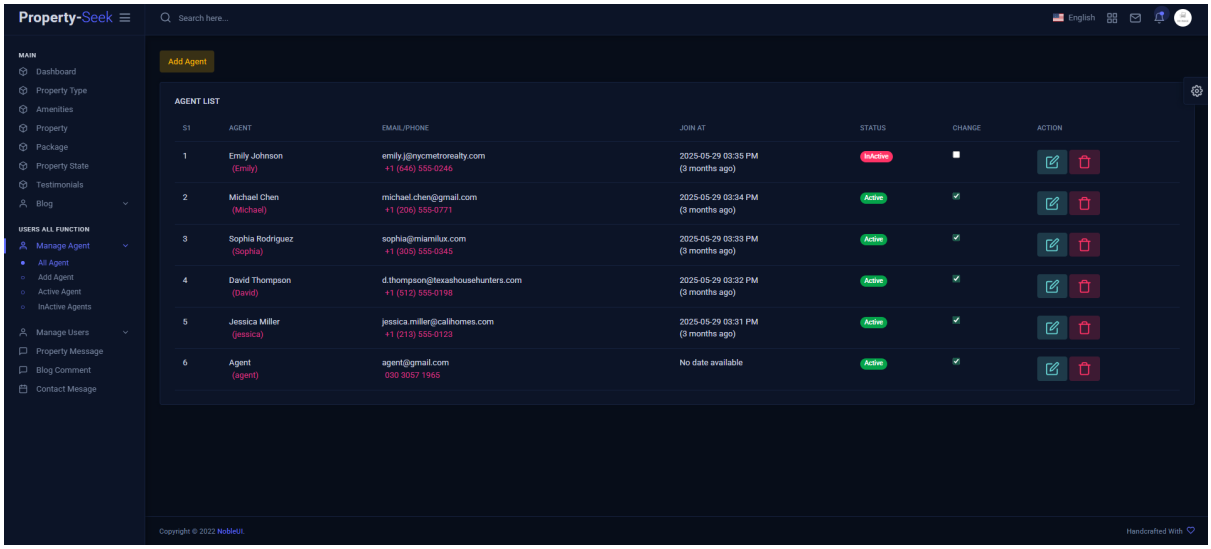


Figure 6.5: Admin Agent Details View Dashboard

Figure 6.6: Create New Agent

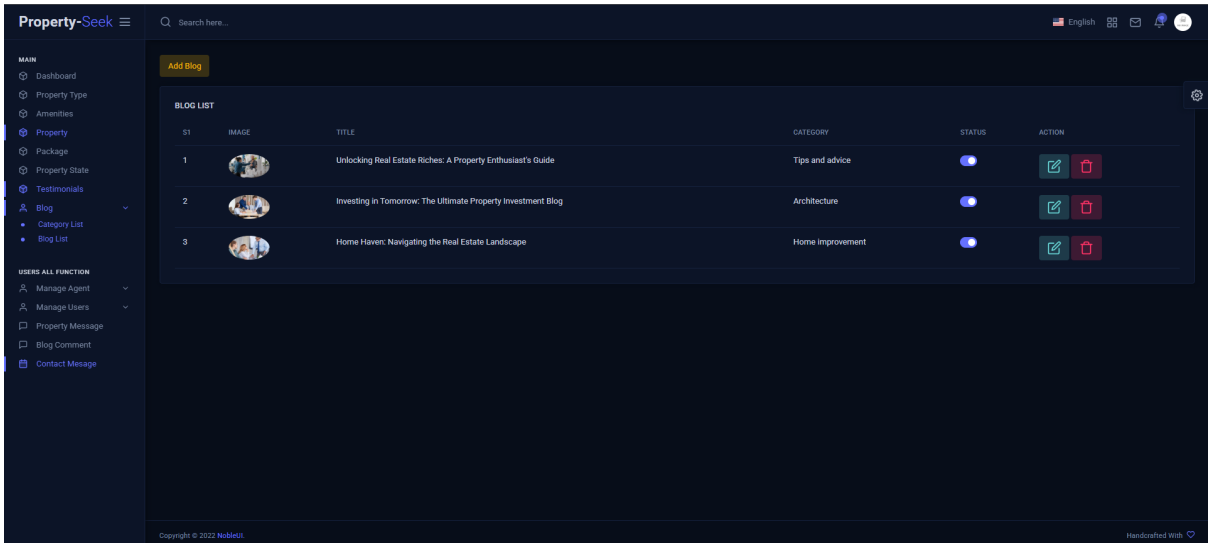


Figure 6.7: Admin Blog Dashboard

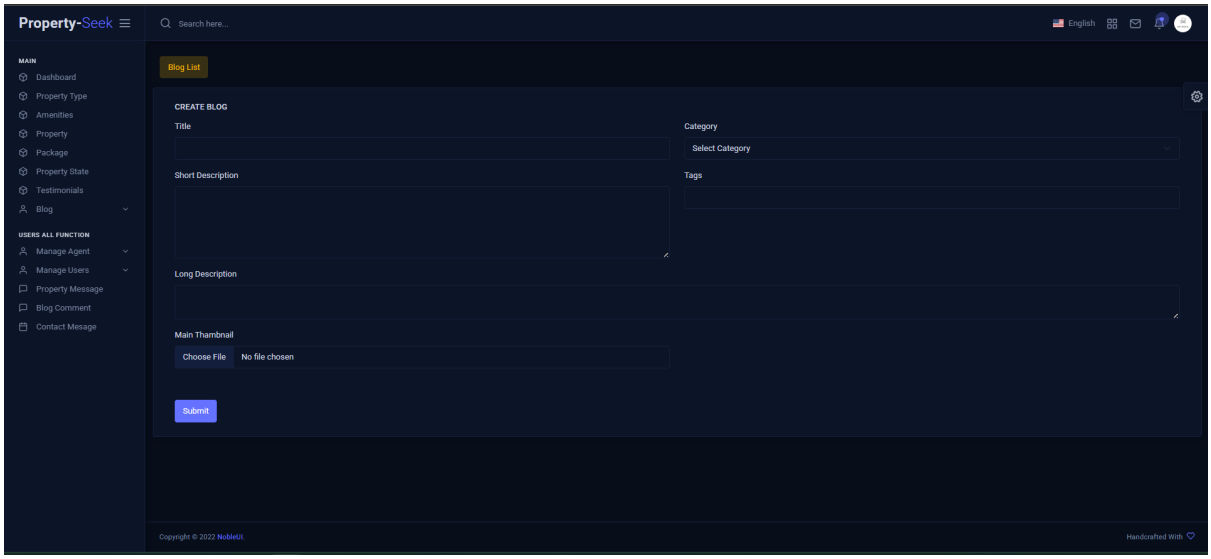


Figure 6.8: Create New Blog

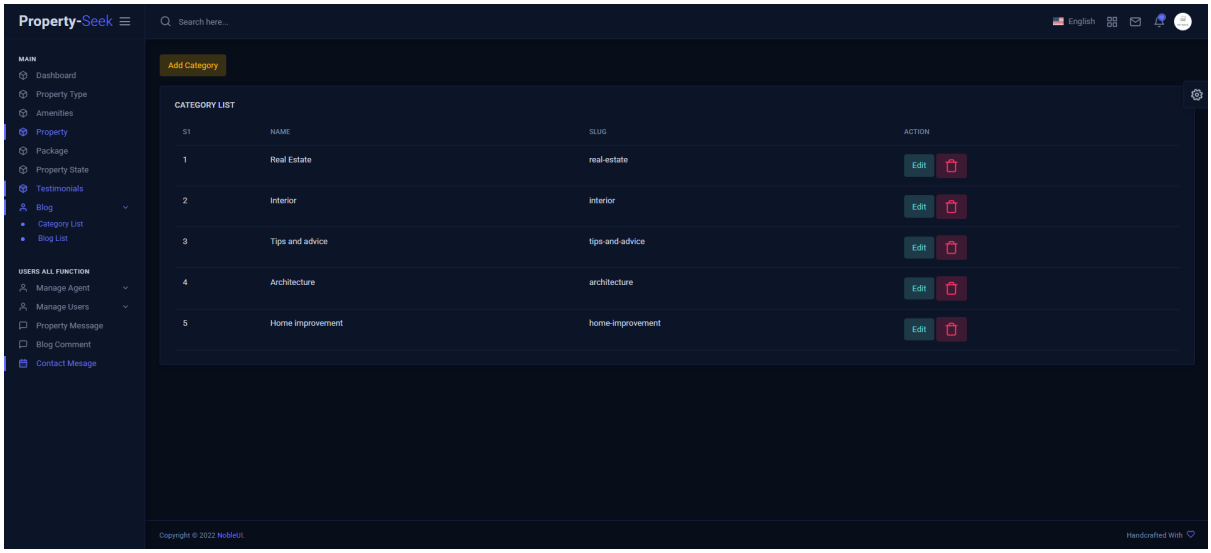


Figure 6.9: Blog Category

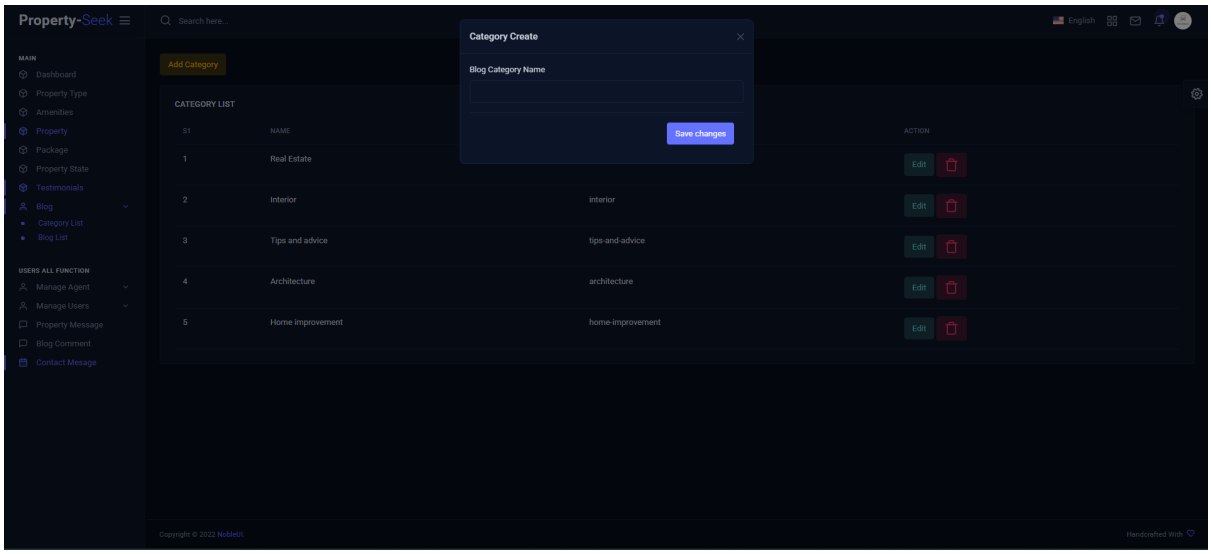


Figure 6.10: Create Blog Categorty

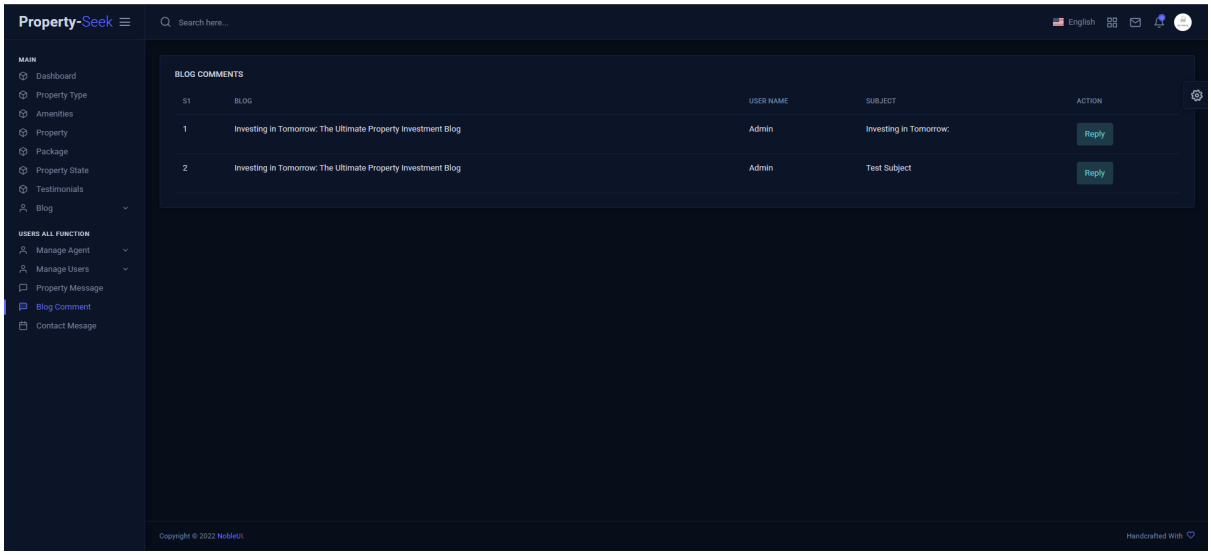


Figure 6.11: Blog Comment View

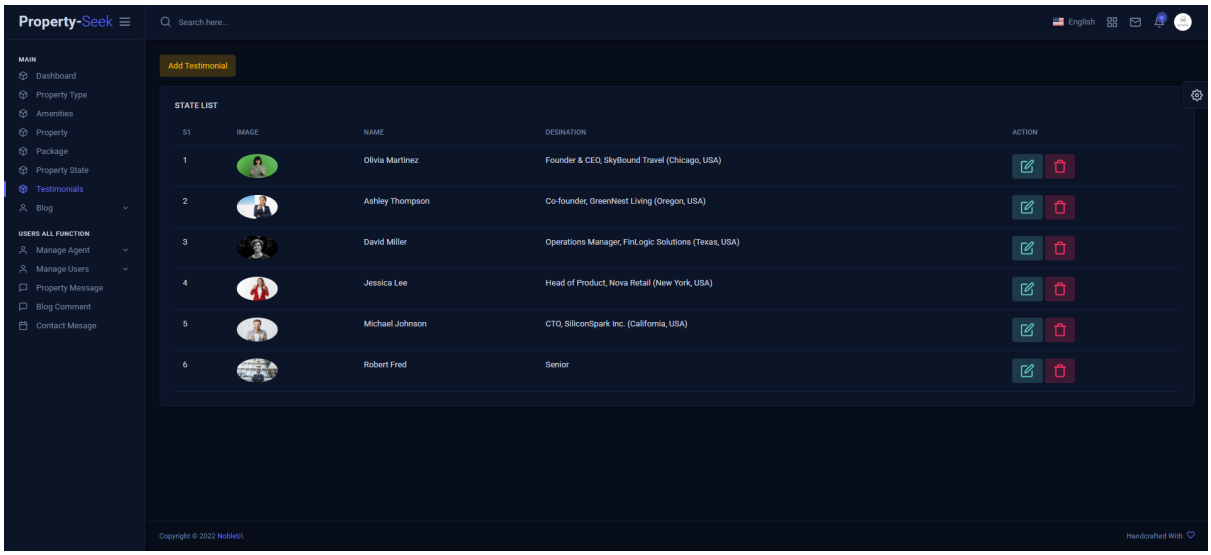


Figure 6.12: View Testimonial

Figure 6.13: Create New Testimonial

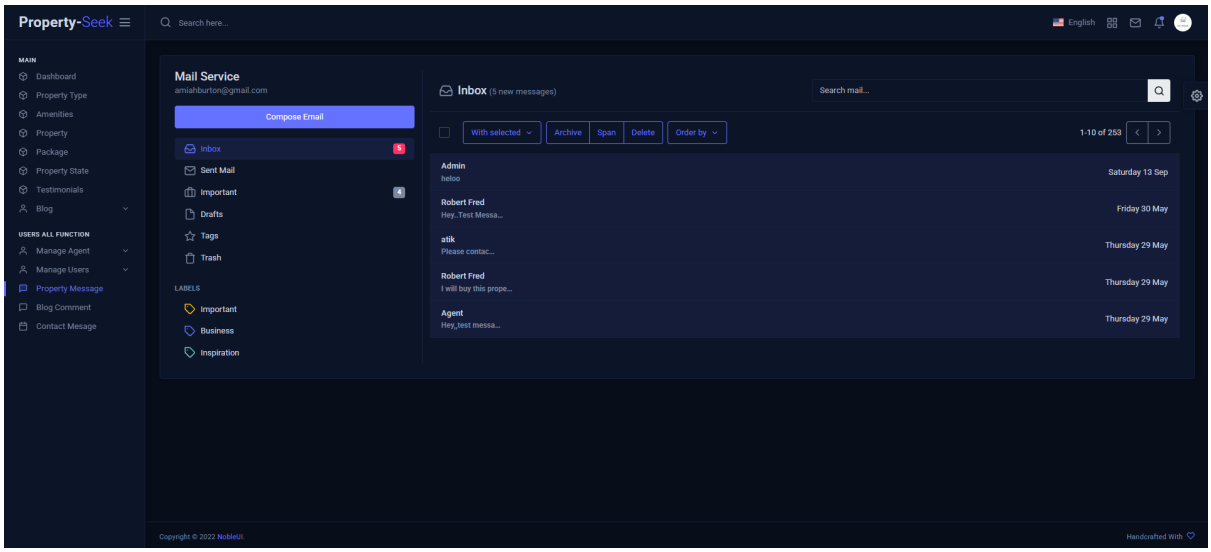


Figure 6.14: View Message Dashboard

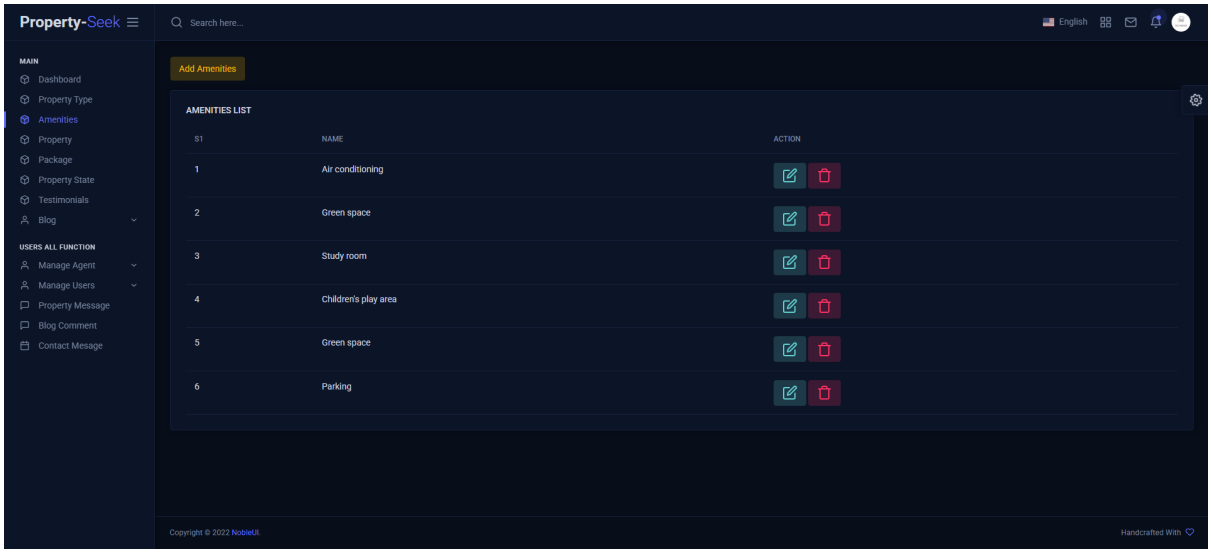


Figure 6.15: View Amenities

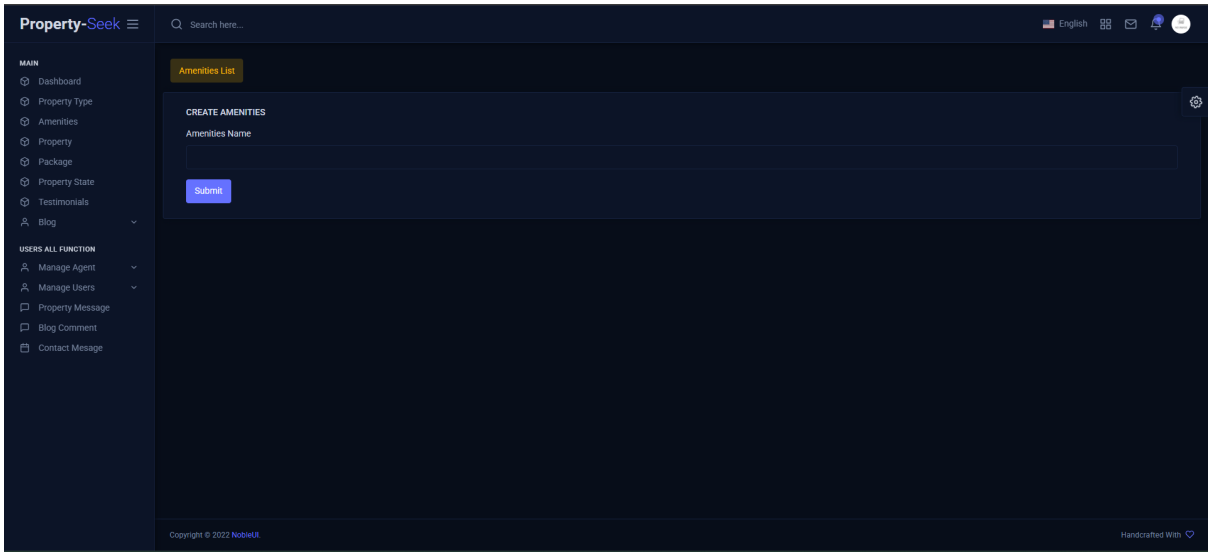


Figure 6.16: Create New Amenities

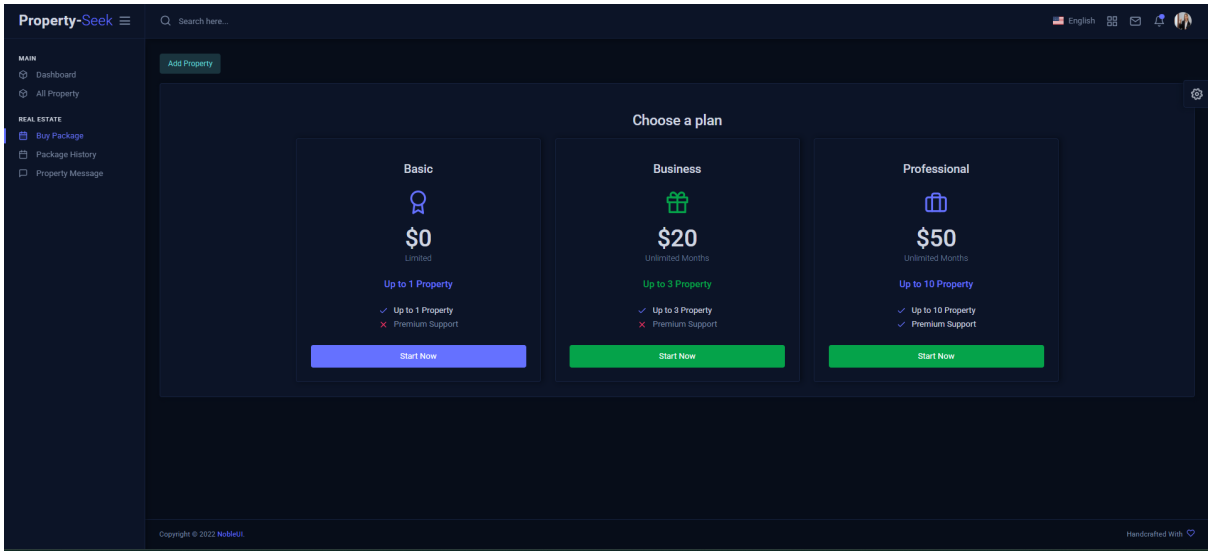


Figure 6.17: Agent Buy Package

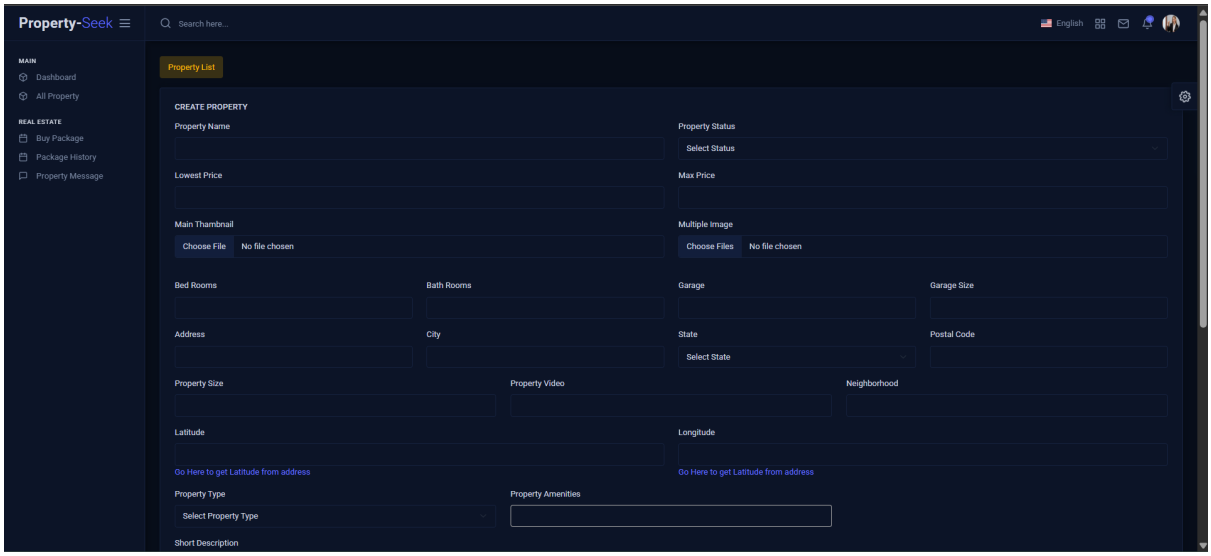


Figure 6.18: Create New Property By Agent

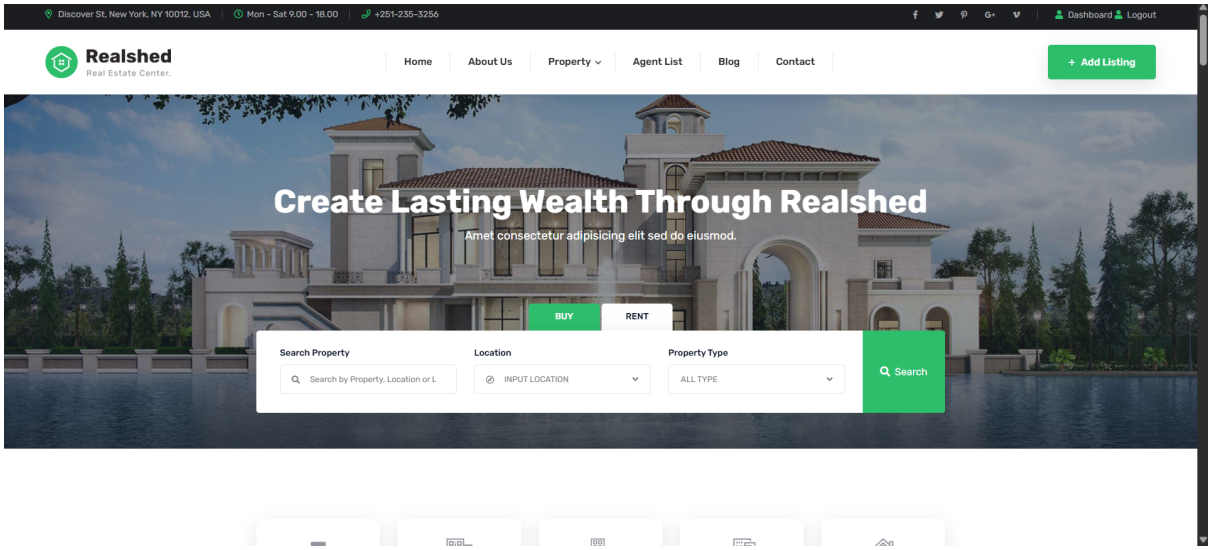


Figure 6.19: User Search Filter Section

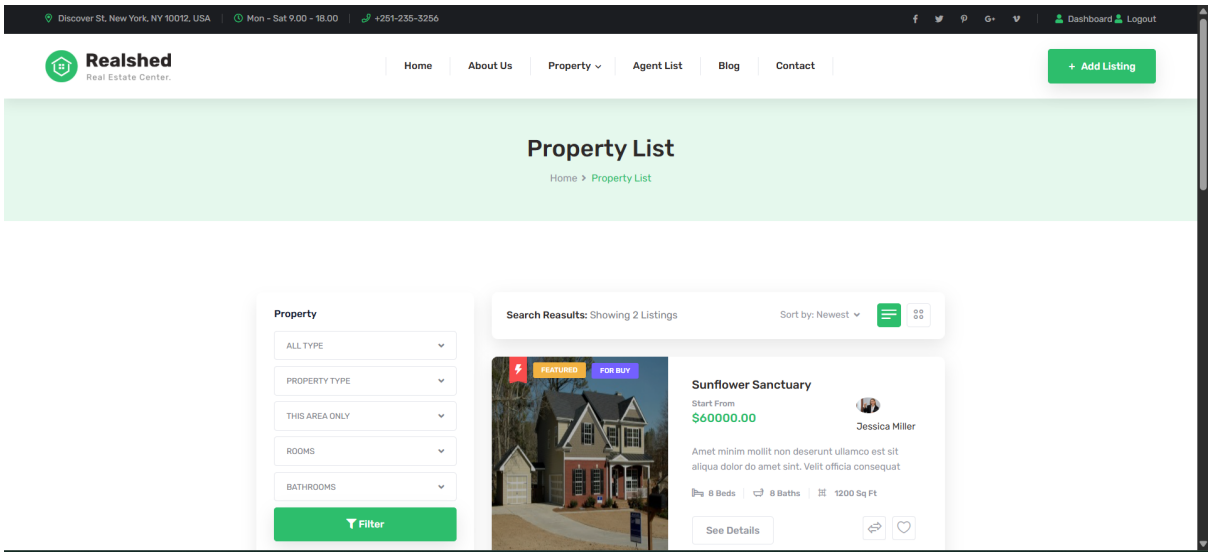


Figure 6.20: User Property View Page

6.2 Analysis

The Real Estate Property Listing System demonstrated strong results during its development and testing phases, successfully meeting its primary objectives. The platform's responsive design ensured seamless performance across desktops, tablets, and smartphones, providing a consistent and user-friendly experience. Features such as role-based dashboards, secure authentication, and approval workflows functioned effectively, streamlining interactions between buyers, agents, and administrators. The search and filtering mechanisms operated efficiently,

delivering accurate results with minimal latency even under multiple concurrent requests. Testing confirmed that data handling and CRUD operations were reliable, while security measures such as password encryption and SQL injection prevention maintained system integrity. Usability analysis showed that the intuitive interface reduced the learning curve for first-time users and encouraged engagement. Although advanced features like map-based search and integrated payment gateways were identified as future enhancements, the current implementation successfully modernizes property transactions and establishes a scalable foundation for further development.

Chapter 7

Project as Engineering Problem Analysis

The analysis of the project emphasizes the development of the Real Estate Property Listing System with a focus on sustainability, social responsibility and ethical integrity. From a technical perspective, it employs stable, open-source technologies alongside a modular Laravel architecture to guarantee scalability and long-term maintenance. On a social level, the system fosters transparency, trust and digital literacy, while simultaneously reducing bias and enhancing efficiency in property transactions. From an environmental standpoint, it significantly decreases paper usage, travel and the consumption of physical resources through digitization. Ethically, it protects user privacy, maintains data integrity, enforces accountability and encourages fair usage through mechanisms such as encryption, audit trails, consent policies and transparent workflows thereby positioning the platform as a secure, responsible and sustainable solution for contemporary real estate management.

7.1 Sustainability of the Project/Work

The Real Estate Property Listing System has been developed with an emphasis on long-term sustainability. The underlying vision for this system is to create a centralized, automated, and secure platform for the listing and management of properties, which effectively serves both buyers and agents in a seamless manner. By reducing reliance on fragmented, outdated methods this system ensures efficiency, accountability and scalability. The sustainability of the project can be understood through the following dimensions:

Technical Sustainability

The system has been constructed utilizing Laravel for the backend, while the frontend is developed with HTML, CSS, Bootstrap, and JavaScript, all underpinned by a MySQL database. These are widely adopted, stable, and open-source technologies, ensuring maintainability in the long term. A modular MVC architecture allows future developers to add new features such

as payment integration, map-based search, or rental modules without redesigning the entire system. This makes debugging, upgrading, and scaling easier over time.

Operational Sustainability

Once implemented, the platform eliminates inefficiencies of manual property promotion, scattered listings and unreliable communication. Buyers benefit from advanced filtering options; agents manage properties in a structured way; and administrators oversee approval workflows. The user-friendly, mobile-responsive design makes the platform accessible to both technical and non-technical users, ensuring adoption and regular use in the long term.

Team Administrative Flexibility

The system has been developed with detailed technical documentation, user manuals and training support to ensure smooth handover. This enables administrators, future developers and agents to adapt easily, continue maintenance, and integrate new features sustainably. Knowledge transfer ensures that the system is not dependent on any one developer.

Cost Effectiveness

By automating listings, inquiries and approvals, the platform reduces marketing and administrative overhead. Since the system relies on open-source tools, there are no recurring license costs. Additionally, digitizing property management saves money otherwise spent on advertisements, manual errors and third-party platforms.

Future-Proofing

The architecture of the system is structured to support future enhancements, which include:

- Payment gateway integration for online transactions.
- AI-driven property recommendation engine.
- Integration with government verification databases.
- Mobile application support.
- Sophisticated analysis of market trends and consumer behavior.

These upgrades can be added incrementally, ensuring the platform remains relevant and sustainable in the evolving real estate sector.

7.2 Social and Environmental Effects and Analysis

Real Estate Property Listing System is not just a technical solution. It also delivers positive social and environmental impacts, contributing to both digital transformation and sustainability in the property market.

Social Impacts

The system improves transparency, efficiency, and equity by validating listings, automating procedures, and ensuring equal visibility for all properties. Additionally, it fosters digital literacy, skill enhancement, and the adoption of technology among both users and developers.

Improved Transparency and Trust: Through admin approval workflows and verified listings, the platform ensures that buyers interact with authentic and accurate property data. This reduces fraud, misinformation and builds trust among stakeholders.

Enhanced Productivity and Efficiency: Automated processes (inquiries, approvals, notifications) reduce workload for agents. Buyers save time with powerful search tools, while administrators monitor property transactions seamlessly. This boosts organizational and individual efficiency.

Technology Adoption and Skill Development: By using the system, agents and buyers are exposed to digital literacy and modern property management practices, encouraging adoption of technology in real estate. The development team also gains valuable experience in scalable web application deployment.

Equal Access and Reduced Bias: The system provides equal visibility to all listings. Properties are displayed based on search criteria and filters, not favoritism or personal bias. This ensures fairness for both small sellers and large real estate agencies.

Environmental Impacts

The system mitigates environmental impact by decreasing paper usage, substituting physical storage with digital records and reducing energy consumption via online property filtering and streamlined manual processes. It fosters environmentally sustainable and resource-efficient real estate practices.

Reduced Paper Usage: Traditional real estate processes rely heavily on physical brochures, posters and paperwork. With digital records and listings, the system significantly reduces paper consumption, contributing to environmental conservation.

Lower Physical Space Needs: Digital storage replaces the need for physical file cabinets and advertisement boards, freeing up office space and reducing waste of physical resources.

Energy Optimization through Digitization Although servers consume power, the centralized digital solution eliminates repetitive manual tasks, reduces travel for in-person property

checks (buyers can pre-filter online) and minimizes dependence on energy-heavy practices like printing.

Overall, socially, the project enhances trust, inclusivity and efficiency in property transactions. Environmentally, it fosters green operations by digitizing records and reducing carbon footprint.

7.3 Discussing Ethics and Ethical Concerns

Managing sensitive user information necessitates thorough ethical deliberation. The Real Estate Property Listing System tackles these issues by promoting transparency, equity, and robust data security. Given that property transactions encompass significant financial amounts and personal information, it is essential to establish trust and ethical practices within the system to ensure its sustained acceptance and the platform's reputation.

Data Privacy and Confidentiality

Issue: Storing sensitive information such as property ownership documents, pricing details and buyer inquiries raises the risk of misuse, unauthorized leaks or identity theft. Breaches could result in financial loss or reputational damage.

Solution: To safeguard data, the system implements role-based access, ensuring that only authorized users have the ability to view or modify records. All passwords are stored in encrypted form, while databases are secured against SQL injection and cross-site scripting. Two-factor authentication adds another security layer. Data in transit is encrypted using SSL, ensuring safety of communication between users and servers.

Data Integrity and Manipulation

Issue: Unauthorized modifications to property details, approval workflows or transaction history could mislead buyers or harm sellers. Even accidental changes can compromise trust and reduce confidence in the platform.

Solution: To ensure integrity, audit trails automatically log all activities, including who made changes and when. Permissions are restricted so only authenticated users can modify critical records. Regular backups are scheduled to prevent data loss and allow recovery from corruption. This makes the system reliable for all stakeholders.

Accountability and Fair Use

Issue: Automated workflows such as property approvals or recommendation engines could unintentionally disadvantage certain users, leading to unfair visibility of listings or biased results. If not monitored, this could reduce user confidence in the fairness of the platform.

Solution: Approval and recommendation processes are made transparent, following established business rules instead of personal bias. Administrators can monitor system outputs and override decisions if necessary. Feedback mechanisms allow agents and sellers to challenge errors, creating accountability and reducing misuse of automation.

User Awareness and Consent

Issue: Users may not fully understand how their personal data are collected, stored, and used within the platform. Lack of awareness can lead to mistrust and legal disputes.

Solution: Clear consent forms are required at registration, and the platform publishes a data use and privacy policy that is easy to read. Disclaimers are displayed during login to remind users of monitoring practices. Additionally, user awareness sessions and help documentation are provided to educate buyers, sellers, and agents on data handling practices.

Responsible Use of Analytics and Reporting

Issue: Advanced analytics, if misused, could create unfair advantages by highlighting certain agencies or by misinterpreting market data. This can generate distrust and accusations of bias.

Solution: Analytics and reporting are strictly used to improve market transparency, not discriminate among users. Dashboards present the information in full context, allowing administrators and sellers to verify before conclusions are drawn. Regular audits of the reporting modules are conducted to prevent manipulation and ensure fairness.

By incorporating these ethical safeguards, the Real Estate Property Listing System maintains privacy, transparency, fairness, and accountability. This ensures that all participants buyers, agents, and administrators can confidently use the platform, knowing their data and interests are protected. Ethical compliance not only secures trust, however, it also enhances the long-term credibility and sustainability of the system.

Chapter 8

Lesson Learned

The Real Estate Property Listing System internship was a highly practical learning experience that gave me real exposure to the software development cycle, collaboration in a corporate setting, and the challenges of delivering a scalable web application. Over the course of my internship, I developed technical expertise, professional maturity and problem-solving skills that will remain valuable throughout my career. The lessons I learned can be categorized as follows:

Real-World Software Development Experience

One of the greatest learnings was understanding how a project evolves from an idea into a fully functional product in a professional environment. I experienced first-hand how to:

- Translate business requirements from stakeholders into technical specifications.
- Break down tasks into manageable modules for proper Progression.
- Develop adaptive and engaging user interfaces utilizing HTML, CSS, JavaScript, and Bootstrap.
- Implement authentication, database integration and back-end logic using Laravel and MySQL.

This hands-on work solidified my technical competence and boosted my confidence in working with production-level web systems.

Importance of Team Work and Communication

Working with my internship supervisors and colleagues helped me realize the value of teamwork and communication in ensuring successful project outcomes. I learned to:

- Ask for clarification when requirements were unclear, rather than proceeding with assumptions.

- Actively engage in feedback loops, where iterative development and testing refined the system.
- Collaborate on shared tasks such as integrating modules and troubleshooting issues collectively.

Team coordination was essential in meeting milestones and keeping all members aligned with the project's objectives and timelines.

Problem-Solving and Critical Thinking

Like any real-world project, developing the Real Estate Property Listing System involved technical and logical challenges. Some of the issues I faced included:

- Designing secure and intuitive user flows for buyers, agents, and administrators.
- Ensuring proper role-based access control and verification of property listings.
- Implementing efficient search and filtering for properties.

Solving these problems taught me to analyze issues systematically, break them into smaller tasks, test different solutions and choose the most optimal approach. This developed my ability to think critically under pressure.

Exposure to Real-World Tools and Frameworks

Working on this project exposed me to industry relevant tools and technologies, including:

- The Laravel framework for server-side development.
- MySQL is utilized for the design and management of databases.
- Bootstrap to creating mobile-responsive user interfaces.
- Git for version control and maintaining code consistency.

These tools gave me hands-on experience in modern software development practices, preparing me for professional work in web application development.

Balancing Non-Functional and Functional Requirements

Another critical lesson was that a good system is not just about functionality, it also requires performance, reliability, scalability and usability. I learned:

- To design user-friendly and responsive interfaces for better user experience.
- The importance of thoroughly testing each feature before release.
- How system performance (speed, uptime, error handling) affects long-term sustainability.

This balance is key to building systems that last beyond initial deployment.

Project Planning and Documentation

Throughout the internship, I realized how important proper documentation and planning are to project success. I participated in:

- Requirement analysis and feasibility studies.
- Drafting architecture and database designs.
- Writing technical notes and maintaining clarity for each phase.

This experience helped me appreciate the need for structured documentation, not only for future maintenance but also for smoother knowledge transfer.

Understanding Real-World Constraints

Finally, the internship gave me exposure to real-world limitations in terms of:

- Working within deadlines and resource constraints.
- Handling unexpected issues like integration errors or last-minute changes.
- Adapting to user behavior, which sometimes did not align with initial system assumptions.

These challenges made me flexible and adaptive, ensuring that I could continue progress despite constraints. Overall, this internship was an eye-opening experience that expanded both my technical knowledge and professional mindset. It gave me real insight into how software systems are conceived, developed, and improved in practical organizational settings. The lessons learned here will guide my academic journey and professional career, particularly in web system development, collaboration, and continuous improvement.

8.1 Problems Faced During this Period

Throughout my internship, I encountered several challenges that tested both my technical knowledge and adaptability. The first challenge was adjusting to a structured work environment, where tasks were assigned with strict timelines and accountability. Initially, it was overwhelming to manage development in Laravel, especially implementing role-based authentication and search filters with accuracy. I struggled with database migrations in Laravel, where schema mismatches and migration conflicts sometimes caused unexpected errors. For example, changing column types or renaming fields without rolling back migrations led to crashes during deployment. Debugging SQL errors and ensuring referential integrity of foreign keys was more complex than I anticipated. Another difficulty was handling authentication. Configuring Laravel's built-in authentication guards for multiple user roles (buyer, admin, agent) required careful design. At times, users were redirected incorrectly due to routing conflicts, which took time to resolve. On the front-end side, creating fully responsive dashboards was a challenge.

Some UI elements looked fine on desktop but broke on smaller devices. Debugging CSS conflicts between Bootstrap and custom styles also consumed significant time. Performance optimization was another issue. When the property database grew, search queries became slower due to inefficient SQL joins. I had to explore indexing and query optimization to improve performance. Finally, working with a team posed its own challenges. There were delays in receiving feedback, and sometimes requirement changes arrived late, forcing me to refactor code under pressure. Integrating modules developed by different team members also resulted in merge conflicts and unexpected bugs.

8.2 Resolution of those Issues

I successfully navigated these obstacles by means of a combination of self-learning, proactive communication and structured time management. For Laravel migration conflicts, I studied the official documentation and practiced rollback and re-run commands until I fully understood schema versioning. I also learned to use seeders and factories for testing data, which improved consistency in the database. To fix authentication issues, I carefully restructured middleware and guards, ensuring each user role had clear access permissions. I tested different user journeys thoroughly to catch misrouting bugs. For front-end problems, I used browser developer tools to identify CSS conflicts and applied Bootstrap's responsive grid system more effectively. I also created small test components before implementing them in the main dashboard to avoid breaking the layout. When performance issues arose, I optimized SQL queries by adding indexes and limiting the number of joins. I also used Laravel's Eloquent ORM more efficiently by applying eager loading, which reduced the number of queries executed. To manage requirement changes and team delays, I broke large tasks into smaller milestones and maintained a personal task tracker. This allowed me to deliver incremental updates on time. I also made a habit of pushing code regularly to Git, which reduced merge conflicts and made collaboration smoother.

In summary, I overcame the problems through continuous learning, effective debugging and better communication with teammates. These experiences not only helped me finish my tasks but also strengthened my ability to work on real-world projects where constraints and unexpected issues are inevitable.

Chapter 9

Future Work & Conclusion

The Real Estate Property Listing System effectively offers a centralized, secure and user-friendly platform for overseeing property transactions. Developed with Laravel, MySQL and Bootstrap, it guarantees scalability, data integrity and efficient role-based access. Future improvements, such as AI recommendations, payment integration and blockchain-based records, have the potential to enhance its sophistication and intelligence. This project not only connects academic knowledge with practical application but also aids in the digital transformation of the real estate sector. In summary, it bolsters my technical, analytical and professional capabilities for upcoming software development challenges.

9.1 Future Works

The Real Estate Property Listing System has established a robust foundation for a centralized, secure, and automated platform to connect buyers, admin and agents. While the current version meets the core functional requirements, there are opportunities for improvement and expansion to enhance usability, scalability and long-term value. The following are the proposed future works for this project:

Advanced Role-Based Access Control (RBAC): At present, the system supports basic role-based authentication for administrators, agents and buyers. A more advanced RBAC model could provide finer-grained permissions such as property approval rights, financial transaction oversight and limited agent access. This will improve security and maintain better segregation of responsibilities.

Property Media and Document Attachments: Future versions may allow attaching supporting files such as legal documents, ownership certificates, property images and videos. This will build trust, improve documentation and eliminate the need for third-party communication.

Interactive Dashboard and Data Analytics: The system can be enhanced with data visualization features to present real estate trends, pricing analytics, property availability and user engagement insights. These dashboards will enable stakeholders to make informed, data-

driven decisions.

Real-Time Notification and Messaging System: Currently, notifications are limited to basic alerts. Real-time in-app notifications, instant messaging between buyers and agents and integration with platforms such as WhatsApp or email would create smoother and faster communication.

Dedicated Mobile Application: While the system is responsive on web browsers, a dedicated Android/iOS application would allow buyers, agents to access listings on the go, improving accessibility and engagement.

Multi-language Support: To cater to a diverse user base, especially in multilingual regions, the platform can integrate translation features and support for multiple languages, increasing inclusivity.

Integration with Payment Gateways: Adding secure payment integration will allow buyers to pay booking fees, deposits or service charges directly through the platform, making the process more seamless.

Geo-Mapping and Virtual Tours: Future enhancements may include map-based property search, location filtering and even 3D virtual property tours. These features will enrich user experience and help buyers make informed decisions without physical visits.

AI-Based Recommendation Engine: Integrating machine learning models may offer tailored property recommendations that take into account user search history, budget preferences, and location interests.

Blockchain Integration for Property Records (Long-Term Scope): As a longer-term goal, blockchain technology can be leveraged to maintain tamper-proof property records, ownership details and transaction histories. This will ensure maximum transparency and trust in real estate transactions.

These proposed improvements are aimed at making the Real Estate Property Listing System not only a transactional platform but also a smart, scalable and secure ecosystem that enhances the overall real estate experience.

9.2 Conclusion

The Real Estate Property Listing System is a web-based platform designed to simplify property transactions by providing centralized access to buyers and agents. It streamlines property listing, searching, and management, ensuring secure role-based access and user-friendly interfaces. Developed using Laravel, MySQL, and Bootstrap, the system ensures responsiveness, data integrity, and efficient management of property records. The project has successfully implemented both functional and non-functional requirements, including usability, maintainability and scalability. Security features such as authentication and property approval mechanisms were

integrated to safeguard sensitive data and maintain trust among stakeholders. The successful development of this system marks a significant step toward the digitalization of real estate management, reducing dependency on fragmented offline methods. Furthermore, this project demonstrated the practical application of academic knowledge in a real-world context, covering requirement analysis, system design, coding, testing, and deployment. Beyond technical achievements, this internship experience improved my teamwork, communication and problem-solving skills. By addressing real challenges such as database conflicts, UI responsiveness and performance optimization, I gained a deeper understanding of professional software development practices.

Overall, this internship has represented a significant milestone in my academic journey, equipping me with the confidence and practical skills necessary to make effective contributions in the realm of web application development and beyond. The Real Estate Property Listing System addresses existing challenges within the property market while simultaneously establishing a foundation for ongoing growth and innovation in the future.

Bibliography

[1] "Php: Hypertext preprocessor." https://www.php.net, 2025. Accessed: 2025-10-03.

[2] "Html: Hypertext markup language." https://developer.mozilla.org/en-US/docs/Web/HTML, 2025. Accessed: 2025-10-03.

[3] "Css: Cascading style sheets." https://developer.mozilla.org/en-US/docs/Web/CSS, 2025. Accessed: 2025-10-03.

[4] "Bootstrap: The most popular html, css, and js library." https://getbootstrap.com, 2025. Accessed: 2025-10-03.

[5] "Javascript: The programming language of the web." https://developer.mozilla.org/en-US/docs/Web/JavaScript, 2025. Accessed: 2025-10-03.

[6] "Laravel: The php framework for web artisans." https://laravel.com, 2025. Accessed: 2025-10-03.

[7] "Lamudi real estate platform." https://www.lamudi.com.bd, 2023. Accessed: 2025-09-09.

[8] "Zillow real estate and rental listings." https://www.zillow.com, 2023. Accessed: 2025-09-09.

[9] B. Alqaralleh, A. Almomani, and M. Alauthman, "Smart real estate system based on cloud and ai," *International Journal of Computer Applications*, vol. 183, no. 24, pp. 15–21, 2021.

[10] Y. Zhou, L. Zhang, and K. Chen, "Personalized recommendation in real estate property search," *Journal of Web Engineering*, vol. 19, no. 4, pp. 317–335, 2020.

[11] M. Rahman and T. Hossain, "Challenges of online real estate platforms in bangladesh," *Bangladesh Journal of Information Systems*, vol. 5, no. 2, pp. 45–52, 2019.

[12] S. Hasan and M. Jahan, "Laravel framework in web application development: A case study on property management," in *Proceedings of the International Conference on Software Engineering*, pp. 122–130, 2022.

[13] R. Ahmed and N. Chowdhury, "Adoption of e-property platforms in bangladesh: Opportunities and barriers," *Asian Journal of Information Technology*, vol. 19, no. 8, pp. 765–773, 2020.

[14] A. Singh and P. Kumar, "Comparative study of real estate portals in emerging markets," *International Journal of Computer Science Trends and Technology*, vol. 9, no. 6, pp. 50–58, 2021.

[15] "Xampp: Apache + mariadb + php + perl." `https://www.apachefriends.org`, 2025. Accessed: 2025-10-03.

[16] "Mysql: The world's most popular open source database." `https://www.mysql.com`, 2025. Accessed: 2025-10-03.

[17] "Agile manifesto: Principles behind the agile software development." `https://agilemanifesto.org/principles.html`, 2025. Accessed: 2025-10-03.

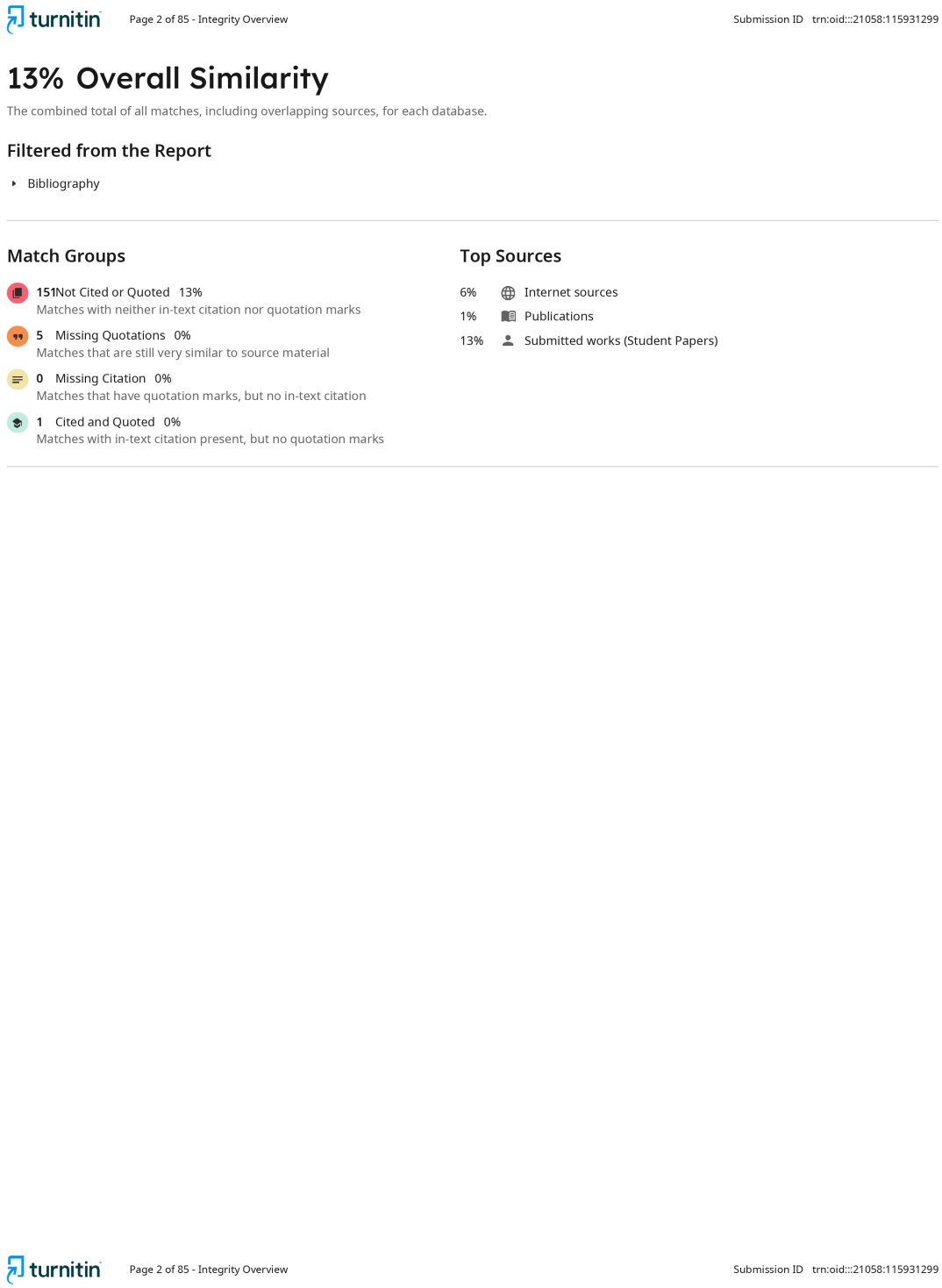
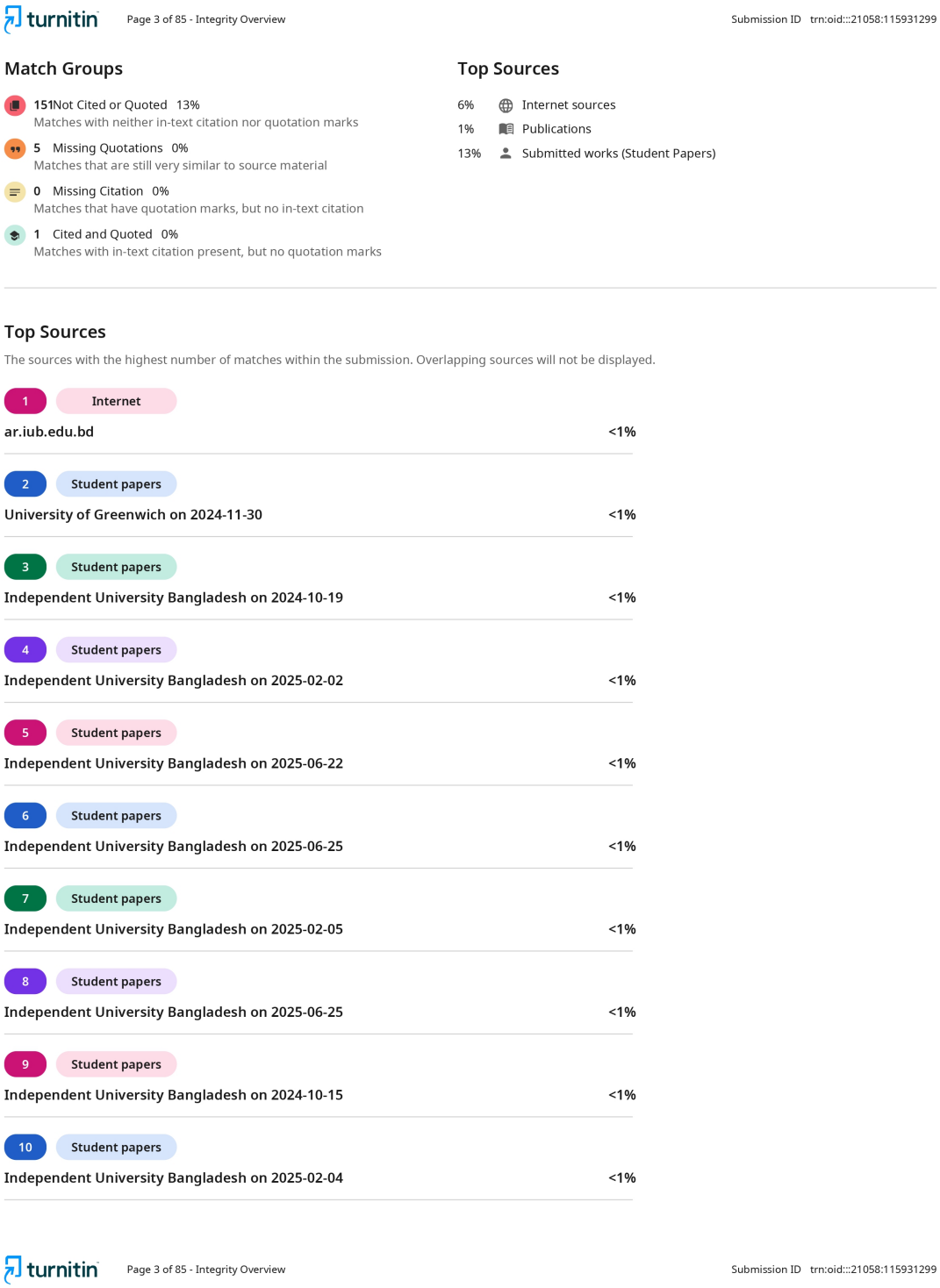


Figure 9.1: Plagiarism Report



turnitin Page 3 of 85 - Integrity Overview Submission ID trn:oid:::21058:115931299

Match Groups

151 Not Cited or Quoted 13%
Matches with neither in-text citation nor quotation marks

5 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

1 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
1% Publications
13% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
ar.iub.edu.bd		<1%
2	Student papers	
University of Greenwich on 2024-11-30		<1%
3	Student papers	
Independent University Bangladesh on 2024-10-19		<1%
4	Student papers	
Independent University Bangladesh on 2025-02-02		<1%
5	Student papers	
Independent University Bangladesh on 2025-06-22		<1%
6	Student papers	
Independent University Bangladesh on 2025-06-25		<1%
7	Student papers	
Independent University Bangladesh on 2025-02-05		<1%
8	Student papers	
Independent University Bangladesh on 2025-06-25		<1%
9	Student papers	
Independent University Bangladesh on 2024-10-15		<1%
10	Student papers	
Independent University Bangladesh on 2025-02-04		<1%

turnitin Page 3 of 85 - Integrity Overview Submission ID trn:oid:::21058:115931299

Figure 9.2: Plagiarism Report

INTERNSHIP CERTIFICATE

Date: October 8, 2025

This is to certify that Shahriar Hosen has successfully completed his internship at IT WAY BD as a Software Developer from 07 May 2025 to 24 September 2025. During this period, he was actively involved in the development and maintenance of websites. His responsibilities included coding, debugging, testing and optimizing various website development.

Shahriar Hosen consistently demonstrated a strong work ethic, excellent problem solving skills and a keen ability to quickly adapt to new technologies and challenges. He worked diligently with the development team and contributed significantly to various development projects.

We are confident that his technical skills and proactive approach will make him an asset in his future career endeavors and we wish him continued success in his professional journey.

Regrads,

08.10.25
Kalam Hossain
HR & Admin
IT WAY BD

IT WAY BD

Phone
01854125454

Website
www.itwaybd.com

Address
House #9, Road #4, Sector #12
Uttra, Dhaka-1230.

Figure 9.3: Internship Completion Certificate

An Undergraduate Internship Project on Real Estate Property Listing System

By

Shahriar Hosen

Student ID: **1910032**

Summer, 2025

The student modified the internship final report as per the recommendation made by his or her academic supervisor and/or panel members during final viva, and the department can use this version for achieving.

Signature of the Supervisor

Asif Mahmud
Lecturer
Department of Computer Science & Engineering
School of Engineering, Technology & Sciences
Independent University, Bangladesh