

An

# Internship Report

submitted by

**Ayush Kumar**  
**(202103103510253)**

in partial fulfillment of the requirements for the degree of

**Bachelors of Technology**  
in  
**Computer Science and Engineering**  
at  
**Uka Tarsadia University**

**Under the guidance of**

Prof. Urvisha Patel  
Assistant Professor



**Department of Computer Science and Engineering**  
**Asha M. Tarsadia Institute of Computer Science and Technology**  
**Uka Tarsadia University, Bardoli**  
**April 2025**



## CERTIFICATE

This is to certify that an **Internship Report** has been carried out by **Mr. Ayush Kumar** having enrollment number **202103103510253** for the partial fulfillment of **Bachelors of Technology in Computer Science and Engineering** at **Asha M. Tarsadia Institute of Computer Science and Technology** degree to be awarded by **Uka Tarsadia University**.

**Date:**

**Place:** AMTICS, Bardoli

**Prof. Urvisha Patel**

Assistant Professor,  
AMTICS

Uka Tarsadia University

**Dr. Vishvajit Bakrola**

Director,  
AMTICS  
Uka Tarsadia University

**Examiner's Signature**

# ABOUT COMPANY



## **Casepoint Private Limited**

Casepoint is the trusted data discovery platform for large corporations and government agencies. Our mission is to transform data into actionable intelligence and responsive insights. Using the power of AI and advanced analytics, our end-to-end eDiscovery platform empowers teams to seamlessly collect, preserve, and discover vast amounts of data from diverse sources.

**Website:** <https://www.casepoint.com/>

**Headquarters:** Tysons, Virginia

**Founded:** 2008

**Specialties:** eDiscovery, Investigations, Compliance, Software Development, Legal Technology, Litigation Support

# INTERNSHIP COMPLETION CERTIFICATE



Date: 7th Jan 2025

**To whomever it may concern**

This is to certify that Ayush Kumar has completed technical training at Casepoint Pvt. Ltd. from 2nd Sep 2024 to 5th Jan 2025.

Project Name: **Real Estate Hub ( Residize )**

Technology: **.NET CORE, C#.NET, PostgreSQL, KendoUI, Redis and RabbitMQ, Elastic Search, DTSearch, Azure DevOps and Git**

During this tenure, we found that he has worked with dedication. We found Ayush Active in whatever task we have provided. He is a professionally sound, hardworking, and devoted staff. He is motivated to take the initiative and we are gratified that he has been helpful in the advancement of our organization.

Thanks & Regards,

**Rakesh Sarvaiya**  
Manager – HR Operations  
Casepoint Private Limited



208, Shivalik Western, L.P. Savani Road, Adajan, Surat - 395009, Gujarat, India.  
[www.casepoint.com](http://www.casepoint.com) | [info@casepoint.in](mailto:info@casepoint.in) | +91-261-2745551 | CIN: U72900GJ2012PTC072588

# ACKNOWLEDGEMENT

I have made an effort throughout this internship. However, it would not have been possible without many individuals' kind support and help. I would like to extend my sincere thanks to all of them.

I would also like to express my gratitude to our respected Director, **Dr. Vishvajit Bakrola**, whose leadership and encouragement created a motivating academic environment that greatly supported this endeavor.

I am highly indebted to **Prof. Urvisha Patel** for her guidance and constant supervision and for providing necessary information regarding the internship work.

I would like to express my gratitude to my parents and other family members for their kind cooperation and encouragement, which helped me complete this project. My thanks and appreciation also go to the people who have willingly helped me out with their abilities.

Ayush Kumar

202103103510253

# ABSTRACT

*This report presents a detailed account of my internship experience as an Associate Software Engineer at Casepoint, with a focus on the technical training, project work, and professional development I underwent during the program. The internship commenced with an intensive technical training phase, covering a range of technologies including front-end development (HTML, CSS, JavaScript), back-end development (.NET Core), and database management (PostgreSQL). This foundational training was complemented by practical, hands-on experience through the development of a real estate platform, Residize, where I applied Agile methodologies and tools like Azure DevOps to work within a collaborative team environment. The report highlights key milestones achieved during the internship, including the completion of technical assessments, participation in live product development, and exposure to enterprise-level application design. The culmination of this internship resulted in an employment offer, allowing me to continue contributing to Casepoint's FOIA Team, where I am now applying my skills to develop legal tech solutions in compliance with the Freedom of Information Act (FOIA). This report also outlines the future scope of my career, including my goals to further specialize in legal technology, cloud computing, and data-driven solutions. Overall, this internship provided both a solid foundation for my software engineering career and valuable real-world experience, which I will continue to build upon in my current role at Casepoint.*

# TABLE OF CONTENTS

<b>CERTIFICATE</b>	<b>ii</b>
<b>ABOUT COMPANY</b>	<b>iii</b>
<b>INTERNSHIP COMPLETION CERTIFICATE</b>	<b>iv</b>
<b>ACKNOWLEDGEMENT</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Internship Commencement . . . . .	1
1.2 Technical Training . . . . .	1
1.3 Assessment and Evaluation . . . . .	4
1.4 Project: Residize – A Real Estate Platform . . . . .	5
1.5 Product Training and Transition . . . . .	8
1.6 Internship Timeline . . . . .	10
1.7 Conclusion . . . . .	11
<b>2 Residize</b>	<b>12</b>
2.1 Architecture Planning . . . . .	12
2.1.1 Project development approach . . . . .	12
2.1.2 System modules . . . . .	14
2.1.3 Functional requirements . . . . .	17
2.1.4 Non-Functional requirements . . . . .	17
2.1.5 Project timeline . . . . .	18

2.2	System Design . . . . .	19
2.2.1	Use case diagram . . . . .	19
2.2.2	Sequence diagram . . . . .	20
2.2.3	Activity diagram . . . . .	22
2.2.4	Entity relationship diagram . . . . .	24
2.2.5	Data flow diagram . . . . .	25
2.2.6	Database schema . . . . .	27
2.3	Implementation . . . . .	31
2.3.1	Hardware and software requirements . . . . .	31
2.4	Implementation Screenshots . . . . .	32
2.5	Test Cases . . . . .	42
2.6	Conclusion . . . . .	44
<b>3</b>	<b>Key Learning</b>	<b>45</b>
3.1	Technical Skills . . . . .	45
3.1.1	HTML and CSS . . . . .	45
3.1.2	JavaScript and jQuery . . . . .	47
3.1.3	PostgreSQL . . . . .	47
3.1.4	.NET Core . . . . .	48
3.1.5	Kendo UI . . . . .	49
3.1.6	Redis . . . . .	50
3.1.7	RabbitMQ . . . . .	51
3.2	Soft Skills . . . . .	52
3.2.1	Communication . . . . .	52
3.2.2	Teamwork and collaboration . . . . .	53
3.2.3	Problem solving . . . . .	54
3.2.4	Adaptability . . . . .	55
3.2.5	Conflict resolution . . . . .	55
3.2.6	Work ethics . . . . .	56
3.3	Conclusion . . . . .	57

<b>4 Conclusion and Future Scope</b>	<b>58</b>
4.1 Conclusion . . . . .	58
4.2 Future Scope . . . . .	59
<b>REFERENCES</b>	<b>60</b>

# LIST OF TABLES

1.1 Internship timeline . . . . .	10
2.1 Residize functional requirements . . . . .	17
2.2 Residize non-functional requirements . . . . .	18
2.3 Residize development timeline . . . . .	18
2.4 Amenities table . . . . .	27
2.5 Token table . . . . .	28
2.6 User table . . . . .	28
2.7 Reviews table . . . . .	28
2.8 Asset details table . . . . .	29
2.9 Projects table . . . . .	29
2.10 Asset amenities table . . . . .	29
2.11 Properties table . . . . .	30
2.12 Wishlist table . . . . .	30
2.13 Test case results of iteration 1 . . . . .	42
2.14 Test case results of iteration 2 . . . . .	43
2.15 Test case results of iteration 3 . . . . .	44

# LIST OF FIGURES

2.1	Use case diagram for Residize . . . . .	19
2.2	Sequence diagram for Admin . . . . .	20
2.3	Sequence diagram for User . . . . .	21
2.4	Activity diagram for Admin . . . . .	22
2.5	Activity diagram for User & Guest . . . . .	23
2.6	ER Diagram for Residize . . . . .	24
2.7	Level 0 data flow diagram for Admin . . . . .	25
2.8	Level 1 data flow diagram for Admin . . . . .	25
2.9	Level 0 data flow diagram for User & Guest . . . . .	26
2.10	Level 1 data flow diagram for User . . . . .	26
2.11	Level 1 data flow diagram for Guest . . . . .	27
2.12	Anonymous user's home page . . . . .	32
2.13	Register page . . . . .	32
2.14	Login page . . . . .	33
2.15	Admin home page . . . . .	33
2.16	User profile view . . . . .	34
2.17	User profile settings . . . . .	34
2.18	Notifications on the dashboard . . . . .	35
2.19	Website insights on the admin dashboard . . . . .	35
2.20	Second page of website insights on the admin dashboard . . . . .	36
2.21	Map view in the admin dashboard . . . . .	36
2.22	Number of active users list in the admin dashboard . . . . .	37
2.23	User dashboard . . . . .	37
2.24	Listed properties on the user dashboard . . . . .	38
2.25	Property detailed view . . . . .	38
2.26	User property management . . . . .	39
2.27	Wishlist section of user property management . . . . .	39
2.28	Add property form . . . . .	40
2.29	Second page of add property form . . . . .	40
2.30	Forget password page . . . . .	41
2.31	Reviews section in the user's home page . . . . .	41

# Chapter 1

## Introduction

This chapter provides a comprehensive overview of the internship program, covering key milestones, technical training, project involvement, examinations, and the transition to product training.

### 1.1 Internship Commencement

The internship officially commenced on 2nd September 2024. On this day, we were introduced to the company, its core products, operational structure, and the objectives of the internship program. The orientation session provided us with an understanding of our roles and expectations, as well as an overview of the technologies we would be working with throughout the training period.

### 1.2 Technical Training

Starting on 3rd September 2024, we embarked on a thorough and comprehensive technical training program aimed at equipping us with the foundational and advanced skills necessary for excelling in key technologies. This training phase was designed to not only provide an in-depth understanding of modern development practices but also to foster practical expertise in both front-end and back-end development, database management, and system integration. By the end of this program, we aimed to develop a well-rounded skill set essential for full-stack development and the efficient integration of complex systems. The training encompassed various areas of focus, with each module providing a deep dive into different aspects of software development:

- **Front-End development:** In this section, we delved into the core principles of front-end development, beginning with HTML, which served as the foundation for structuring the content of web pages. Our journey continued with CSS, where we learned how to enhance the aesthetic appeal of web pages through styling techniques. We explored various CSS frameworks, focusing on responsive design methodologies, to ensure our applications were accessible and well-optimized for all device types. We also gained hands-on experience with Bootstrap, a widely-used front-end framework, which provided us with pre-designed components and grids to help us streamline the development process. Additionally, we studied JavaScript[1], focusing on how to implement dynamic features that enhance user experience. This included creating interactive elements such as dropdowns, sliders, and form validation. In parallel, we learned the ins and outs of jQuery[2], a powerful JavaScript library that simplified the process of handling events, manipulating the Document Object Model (DOM), and working with animations. As we progressed, we also worked on integrating third-party APIs, helping us understand how to fetch and display data dynamically on the web.
- **Back-End development:** Our training in back-end development was centered around .NET Core[4], a robust and flexible framework for building scalable and high-performance applications. We began by learning to develop simple console applications, gaining a foundational understanding of how back-end systems process data and handle logic. The next step was diving into the creation of desktop applications using Windows Forms, a graphical user interface (GUI) toolkit for building rich, user-friendly desktop applications. The training then progressed to the Model-View-Controller (MVC) architecture, which is widely used to structure scalable, maintainable applications. By applying the MVC pattern, we were able to efficiently separate application logic, user interface, and data access, thereby improving both code organization and scalability. Furthermore, we received specialized instruction on building Web APIs to create RESTful services. These services enable communication between different parts of a system or with external systems, and we learned how to structure and implement them effectively

for secure, reliable, and efficient data exchange. This knowledge also laid the foundation for understanding microservices architecture and how to build loosely coupled services that can scale independently.

- **Database management:** A critical component of modern application development is effective database management. During this training phase, we worked extensively with PostgreSQL[3], an advanced, open-source relational database management system. We focused on PostgreSQL's powerful features such as ACID compliance, data integrity, and support for complex queries. Our learning journey included designing and optimizing database schemas, working with relational data, and mastering SQL queries. We were also introduced to indexing strategies and performance optimization techniques, allowing us to efficiently handle large datasets and complex queries. Furthermore, we studied the implementation of stored procedures and triggers to automate data-related tasks within the database, enhancing our ability to build robust data-driven applications. As the training progressed, we also gained hands-on experience with advanced features such as transactions and concurrency control, which allowed us to understand the challenges and best practices involved in managing multiple users and operations in a live production environment.
- **Advanced technologies:** In addition to mastering core technologies, we also explored a range of advanced tools and frameworks that are integral to building modern, high-performance applications. One such tool was Kendo UI[5], a comprehensive UI component library that enabled us to create rich, interactive web applications. By leveraging Kendo UI, we were able to integrate complex elements like grids, charts, and data visualization tools with ease, significantly enhancing the user experience. Another key area of focus was Redis[6], an in-memory data structure store used for caching and real-time analytics. We learned how to implement Redis to improve application performance by storing frequently accessed data in memory, reducing the load on backend systems, and enabling near-instantaneous data retrieval. We also gained expertise in RabbitMQ[7], a

powerful messaging broker that facilitates communication between different components of a system. By implementing RabbitMQ, we were able to create highly efficient, asynchronous messaging systems that support scalable, decoupled applications. Furthermore, we explored Elasticsearch[8], a distributed search and analytics engine used for handling large-scale data queries. With Elasticsearch, we learned how to implement full-text search and fast data retrieval, even when dealing with massive amounts of unstructured data. Finally, we gained valuable experience with DTSearch, a high-speed text searching engine that is used for indexing and querying large volumes of text. This tool is particularly useful for systems that require fast, real-time searching capabilities, and we learned how to integrate it into our applications to provide lightning-fast text search results.

Overall, the training program was designed to cover a broad spectrum of technologies and development practices, equipping us with the technical expertise needed for full-stack development and effective system integration. We not only acquired theoretical knowledge but also applied it through hands-on projects, which helped solidify our understanding and prepare us for real-world challenges. By the end of the training, we were fully prepared to tackle complex development tasks, from building user-friendly front-end interfaces to implementing powerful back-end systems and managing scalable databases. The skills we acquired during this training phase laid the foundation for our future work, ensuring that we were equipped to contribute to any aspect of software development and system integration projects.

### 1.3 Assessment and Evaluation

To ensure our understanding of the technologies and concepts, multiple evaluations were conducted throughout the internship. These included:

- **PostgreSQL exam** – A comprehensive 100-mark multiple-choice examination assessing understanding of relational database management systems. Topics included SQL querying, normalization, indexing strategies, transactions, and data integrity constraints.

- **C# OOP exam** – A multiple-choice assessment evaluating knowledge of object-oriented programming using C#. The exam covered core concepts such as classes and objects, inheritance, polymorphism, encapsulation, interfaces, and abstract classes.
- **Practical examinations** – Hands-on project-based assessments integrating .NET Core MVC and PostgreSQL. Candidates were required to design, develop, and deploy fully functional web applications demonstrating practical knowledge of front-end and back-end integration, data modeling, and MVC architecture. Sample projects included:
  - **Event Management System** – A CRUD-enabled application for managing event scheduling, participant registration, and venue allocation. Emphasized user authentication, form validation, and relational data handling.
  - **Doctor Appointment Booking System** – A patient-centric scheduling system for managing doctor availability, appointment booking, and medical records. Included features such as calendar integration, session-based user access, and responsive UI design.
- **Income Tax Buddy** – A user-friendly income tax calculator that allows users to manage their expenses and income. Based on the input data, the system calculates the applicable income tax. Key features include the ability to add and remove income and expense entries, real-time tax calculations, and an intuitive, responsive UI design for easy data management and tax computation.

## 1.4 Project: Residize – A Real Estate Platform

The core component of our internship was the development of **Residize**, a comprehensive real estate platform designed to facilitate property transactions across multiple categories, including residential, commercial, and land properties. Residize serves as a web-based application that enables users to list, browse, and request details about properties available for sale or rent. This platform aims to provide a seamless and

user-friendly experience for property buyers, sellers, renters, and agents, offering a wide range of functionalities tailored to each user type.

Residize was developed with the goal of addressing common challenges in the real estate industry, such as the fragmentation of property listings and lack of efficient communication between buyers and sellers. The platform integrates modern technologies, enabling the automation of several processes related to property management, user engagement, and data analysis. It also includes tools for real-time updates and notifications, ensuring that all users have access to the most up-to-date property information.

The development of Residize provided us with the opportunity to apply our technical training to a real-world scenario. It involved the integration of several technologies, including .NET Core for backend development, PostgreSQL for database management, and various modern front-end frameworks for a dynamic user interface. The system was designed to ensure performance, scalability, and security, accommodating the increasing demand of users as the platform grows. Furthermore, Residize follows the Agile development methodology, specifically utilizing the Scrum framework to manage development cycles, allowing us to adapt to changing requirements and deliver high-quality results incrementally. We utilized Azure DevOps to track sprints, manage tasks, and monitor the development progress, ensuring that our team worked collaboratively and efficiently to meet deadlines.

The project was structured around several critical functional requirements that were carefully planned and executed. These requirements were not only essential for ensuring the platform's core functionality but also played a pivotal role in defining the user experience and ensuring that the platform was scalable and adaptable to future enhancements. The key functional requirements of Residize are outlined below:

- **User authentication:** The user authentication system is a foundational feature of the platform, providing secure registration, login, and account management functionalities. We implemented a robust authentication mechanism that allows users to create personal accounts, log in securely using encrypted credentials, and manage their profile information. Users can recover their password if they

forget it. This ensures that only authorized users can access private data and manage their listings or perform specific actions such as shortlisting properties. The authentication system is built to comply with the latest security standards, ensuring that user data is handled with the utmost care.

- **Guest access:** For non-registered users, Residize offers limited access to the platform, allowing them to browse property listings without the need for account creation. This feature is designed to attract a larger audience and provide easy access to basic property information such as photos, prices, and brief descriptions. While guests can view listings, they are encouraged to sign up for an account to take full advantage of the platform's features, such as saving properties to a shortlist or contacting property owners directly. This feature helps balance the need for accessibility with the advantages of account-based engagement.
- **Property management:** Property management is a critical component of the platform, as it enables property owners, real estate agents, and administrators to list, edit, and manage their properties seamlessly. Residize provides an intuitive interface for property owners and agents to submit detailed property information, including descriptions, photos, pricing, and amenities. Listings are categorized by property type (e.g., residential, commercial, land) and can be easily updated or removed as needed. This feature also supports media uploads such as images and video tours, allowing property owners to provide comprehensive and engaging content. To further enhance the property listing process, we implemented validation features that ensure all required fields are completed and that listings comply with predefined standards.
- **Shortlisting:** The shortlisting feature is designed to help users save properties that they are interested in for future reference. This allows users to compare different properties, making it easier to track and evaluate their options. The feature is integrated with the user account system, meaning users can access their shortlisted properties from any device once logged in. This enhances the user experience by providing a personalized and organized way to save and revisit

properties.

- **Review system:** To build trust and transparency within the platform, we developed a review and rating system that allows users to provide feedback on their overall experience with the Residize website. The review system enables users to rate various aspects of the website, such as ease of use, design, functionality, responsiveness, and customer support. Users can also submit text-based feedback, offering detailed insights into their experience. To ensure that the content remains appropriate and relevant, we integrated a moderation system that allows administrators to review and approve user-generated content before it is published. This moderation process helps maintain a high-quality user experience by filtering out any inappropriate or irrelevant material. The review system not only empowers users to share their thoughts but also helps future visitors assess the website's performance, guiding them in making informed decisions about using the platform.
- **Admin dashboard:** The admin dashboard is a crucial feature that provides administrators with full control over the platform. Through this dashboard, admins can manage all aspects of the platform, including user accounts, property listings, reviews, and analytics. Administrators can also monitor user activity and generate reports on platform performance. The dashboard includes various tools for managing listings, popular property categories, and user engagement. The admin dashboard plays a key role in maintaining the integrity of the platform, ensuring that users have a positive and secure experience.

## 1.5 Product Training and Transition

Upon successful completion of the project phase, interns are evaluated and subsequently promoted to the Product Training stage. This phase is crucial for the interns' professional growth as it provides them with a deeper understanding of the company's actual products, the underlying architecture, and the technologies used to develop them. It marks the transition from learning in a controlled environment to engaging

in real-world software development projects.

During this phase, interns are integrated into ongoing development teams, where they contribute to live projects that have a direct impact on the company's product offering. This allows interns to gain hands-on experience with the entire software development lifecycle, from design and implementation to testing and deployment. The transition from project-based tasks to contributing to enterprise-level applications provides valuable exposure and a significant learning opportunity.

Key aspects of this phase include:

- **Exposure to enterprise-level application development:** Interns gain insight into how large-scale applications are designed, built, and maintained. They learn how various components of a product interact and the importance of scalability, performance, and security in real-world applications.
- **Hands-on experience with real-world problem-solving and debugging:** Interns are tasked with solving complex problems that arise during the development of live products. They develop critical problem-solving skills and gain a deeper understanding of the challenges involved in maintaining and improving software that is used by real customers. Additionally, they work on debugging and troubleshooting existing code, honing their technical skills in a live environment.
- **Engagement with experienced developers and participation in code reviews:** During this phase, interns work closely with senior developers, gaining invaluable mentorship and insight. They have the opportunity to participate in code reviews, where they can both receive feedback on their work and provide input on the work of others. This collaborative environment fosters professional growth and helps interns improve their coding practices, design skills, and understanding of best practices in software development.
- **Familiarization with Agile development processes:** Interns also become acquainted with Agile methodologies, commonly used in professional develop-

ment settings. They participate in sprint planning, daily stand-ups, and sprint retrospectives, where they learn the importance of communication, time management, and iterative development in the context of delivering high-quality software products.

- **Collaboration across departments:** Interns may also have the chance to interact with various departments such as marketing, product management, and quality assurance. This cross-departmental collaboration gives them a well-rounded view of how different functions contribute to the development, launch, and ongoing success of a product.
- **Development of soft skills:** In addition to technical expertise, interns are also encouraged to develop their interpersonal and communication skills. They present their work to different stakeholders, collaborate in teams, and handle constructive criticism, all of which contribute to their overall growth as professionals.

This stage of the internship ensures that interns are fully prepared for a seamless transition into full-time roles, equipped not only with technical skills but also with the soft skills necessary to thrive in a fast-paced, collaborative work environment.

## 1.6 Internship Timeline

Table 1.1: Internship timeline

Activity	Date(s)
Internship Commencement	2nd September 2024
Technical Training	3rd September – 15th November 2024
Project Phase	18th November 2024 – 3rd January 2025
Product Training	6th January – 28th February 2025
Employment Start Date	1st March 2025

## **1.7 Conclusion**

The internship provided valuable technical training and practical experience, particularly through the development of Residize. It enhanced my skills in software development, teamwork, and Agile practices, preparing me for a seamless transition into a full-time role.

# Chapter 2

## Residize

This chapter details the design and architecture of the Residize system, covering key components, design decisions, and system requirements. It focuses on the system's structure, user authentication, property management, and admin functionalities, all developed using an Agile methodology.

### 2.1 Architecture Planning

Effective system planning is fundamental to Residize, establishing a scalable and user-centric property management platform. This chapter outlines the project's development approach, system modules, functional and non-functional requirements, and project timeline, providing a structured execution framework.

#### 2.1.1 Project development approach

Residize adopts the Agile Software Development Model[9][10] , an iterative methodology that prioritizes flexibility, collaboration, and continuous feedback. The Agile approach ensures that the development process remains adaptable to changing needs while maintaining a strong focus on delivering high-quality products. It allows teams to break down the development process into manageable segments, known as sprints, which typically last between one to four weeks. This approach not only promotes transparency but also ensures that progress is continuously monitored and refined based on feedback and evolving requirements.

The Agile model's emphasis on small, incremental releases allows for continuous integration of features and quick adaptation to feedback from stakeholders. Teams can

adjust priorities and reallocate resources as necessary, responding promptly to changes in the business environment, user needs, or technological advancements. Through this iterative process, Residize ensures that its software products are developed efficiently, with high alignment to business objectives and user expectations.

### **Advantages of agile:**

- **Flexibility:** The iterative nature of Agile allows for easy accommodation of changing requirements, even late in the development process. As business needs evolve or new insights arise, the development process can be adjusted through regular sprint reviews and retrospectives, ensuring that the final product meets the most current demands and expectations.
- **Stakeholder engagement:** Agile fosters continuous communication between the development team and stakeholders. This constant collaboration results in a product that better reflects user needs and expectations, as stakeholders have frequent opportunities to provide feedback and influence the direction of development. Regular sprint reviews ensure that the product aligns with both user feedback and strategic goals.
- **Risk reduction:** The iterative nature of Agile allows for early identification of issues and challenges. By continuously testing, integrating, and reviewing small portions of the software, potential risks—such as technical difficulties, feature misalignments, or usability concerns—can be identified and addressed early, minimizing the likelihood of major roadblocks later in the project.
- **Efficiency:** Agile's focus on delivering small, functional increments of the product ensures that development teams remain productive and focused on the most valuable tasks. The short sprint cycles promote rapid development and frequent delivery of features, which helps optimize resources, reduce delays, and improve time-to-market. Moreover, the constant feedback loop ensures that teams are working on the most impactful tasks, maximizing overall efficiency.
- **Continuous improvement:** Agile encourages a culture of constant reflection

and improvement. Through regular sprint retrospectives, teams assess their performance, identify bottlenecks, and make adjustments to improve productivity and quality. This ongoing refinement leads to higher-quality products and more efficient development practices over time.

- **Transparency and accountability:** Agile promotes transparency within the team and with stakeholders. Daily stand-ups and sprint reviews keep everyone informed about progress, challenges, and upcoming tasks, creating a culture of accountability. This transparency ensures that issues are quickly identified and resolved and helps maintain alignment between team members and stakeholders.
- **Customer centric focus:** By involving stakeholders and end-users early and often, Agile ensures that the development process is focused on delivering value to the customer. Feedback from users helps shape the product, ensuring that it meets user needs and provides a positive experience. The frequent delivery of increments also allows for early validation of features, ensuring that the product remains aligned with user expectations throughout the development cycle.

The Agile Software Development Model empowers teams to be more adaptive, responsive, and focused on delivering high-quality software that meets both user and business needs. By breaking down development into manageable, iterative chunks and encouraging ongoing feedback, Residize can deliver products that are flexible, reliable, and highly valuable to customers.

### 2.1.2 System modules

Residize is a comprehensive web-based platform engineered to provide users with a seamless experience in property exploration, management, and administration. The platform integrates a variety of intuitive features designed for diverse user roles, including property seekers, owners, and platform administrators. The system modules cater to these needs, enhancing user engagement and streamlining the overall process.

- **Sign-up:** The registration process allows new users to create an account through

manual entry of personal details or via integration with social media platforms.

Upon successful registration, users receive email verification to ensure the accuracy of the provided information and enhance security.

- **Login:** Secure access is facilitated through email-password authentication, ensuring only authorized users can access their accounts. Additionally, users can log in via social media accounts, providing flexibility and convenience for those who prefer not to maintain separate credentials.
- **Password Recovery:** In case of forgotten credentials, users can securely reset their passwords by following an email verification process. This feature ensures that users can regain access to their accounts with minimal effort and maximum security.
- **Forget Password:** Users who forget their password can initiate a password reset process by providing their registered email address. A secure, time-sensitive link is sent to the user's email, allowing them to reset their password and regain access to their account.
- **Guest Mode:** For individuals who wish to explore available properties without committing to registration, Residize offers a guest mode. This mode grants limited browsing access, enabling users to search for properties and view key details without creating an account.
- **Home Page:** The central hub of the platform serves as the primary interface for users. It features a user-friendly search function, highlighting available properties based on various filters such as location, price, and property type. Users can easily navigate to discover properties that align with their preferences, making the home page a vital feature of the platform.
- **Property Management:** Property owners can efficiently manage their listings through a comprehensive set of tools. This includes the ability to add, edit, or remove property listings, as well as organize them into categories for better visibility. Owners can also shortlist properties of interest and establish direct

communication with potential buyers or renters.

- **Reviews:** Transparency is promoted through a review system, where users can submit ratings and feedback on properties they have visited or stayed in. This feature fosters a sense of trust within the community, helping potential tenants or buyers make informed decisions based on real user experiences.
- **Admin Dashboard:** Platform administrators have access to a robust dashboard that provides tools for managing user accounts, property listings, and platform analytics. Administrators can oversee all aspects of the platform's operation, from moderating user-generated content to tracking platform performance and user engagement metrics.
- **Profile Management:** Each registered user has access to a personalized profile where they can update personal details, manage saved properties, and view their communication history with property owners and administrators. The profile serves as a central location for users to manage their interactions and ensure that their preferences are always up-to-date.
- **Notifications:** Users are notified about relevant platform updates through real-time notifications. These may include property availability, price changes, new messages, and system alerts. Notifications help users stay informed and engaged with the platform's activities, ensuring they don't miss important updates.

These system modules are designed with both functionality and user experience in mind, offering a streamlined approach to property management and exploration. By addressing the needs of all user types—seekers, owners, and administrators—Residize aims to create a dynamic, user-centric platform that simplifies the property search and management process.

### 2.1.3 Functional requirements

Residize's core functionalities ensure an efficient property management experience:

Table 2.1: Residize functional requirements

ID	Requirement	Description
FR1	User Authentication	Secure registration, login, and account management.
FR2	Guest Access	Limited browsing for non-registered users.
FR3	Property Management	Listing, editing, and management of properties.
FR4	Shortlisting	Saving properties for future reference.
FR5	Review System	Property feedback and review moderation.
FR6	Admin Dashboard	User and listing management with analytics.
FR7	Analytics	Data tracking and reporting for insights.

### 2.1.4 Non-Functional requirements

Residize is designed with a strong focus on quality attributes that ensure long-term usability, performance, and adaptability. These non-functional requirements define how the system operates rather than what it does:

Table 2.2: Residize non-functional requirements

ID	Requirement	Description
NFR1	Performance	Optimized for fast response times and minimal latency, ensuring efficient performance under high load.
NFR2	Scalability	Supports growth through horizontal scaling and database optimization to handle increased demand.
NFR3	Usability	User-centric design with an intuitive interface, ensuring ease of use across devices.
NFR4	Cross-Platform Compatibility	Consistent functionality across major OS and browsers, ensuring broad accessibility.
NFR5	Reliability	High availability with fault tolerance, automated recovery, and redundancy mechanisms.
NFR6	Maintainability	Modular architecture and clear documentation ensure ease of maintenance and long-term sustainability.

### 2.1.5 Project timeline

Residize follows an iterative development approach with structured sprints:

Table 2.3: Residize development timeline

Phase	Timeline
Technical Training	2nd September – 15th November 2024
Iteration 1	18th – 29th November 2024
Iteration 2	9th – 20th December 2024
Iteration 3	23rd December 2024 – 3rd January 2025

Each sprint includes Planning, Development, and Review, ensuring incremental progress and continuous improvement.

## 2.2 System Design

The system design defines Residize's structural and functional blueprint. It includes use case diagrams, sequence diagrams, entity-relationship diagrams (ERD), data flow diagrams (DFD), and database schema, ensuring a clear development approach.

### 2.2.1 Use case diagram

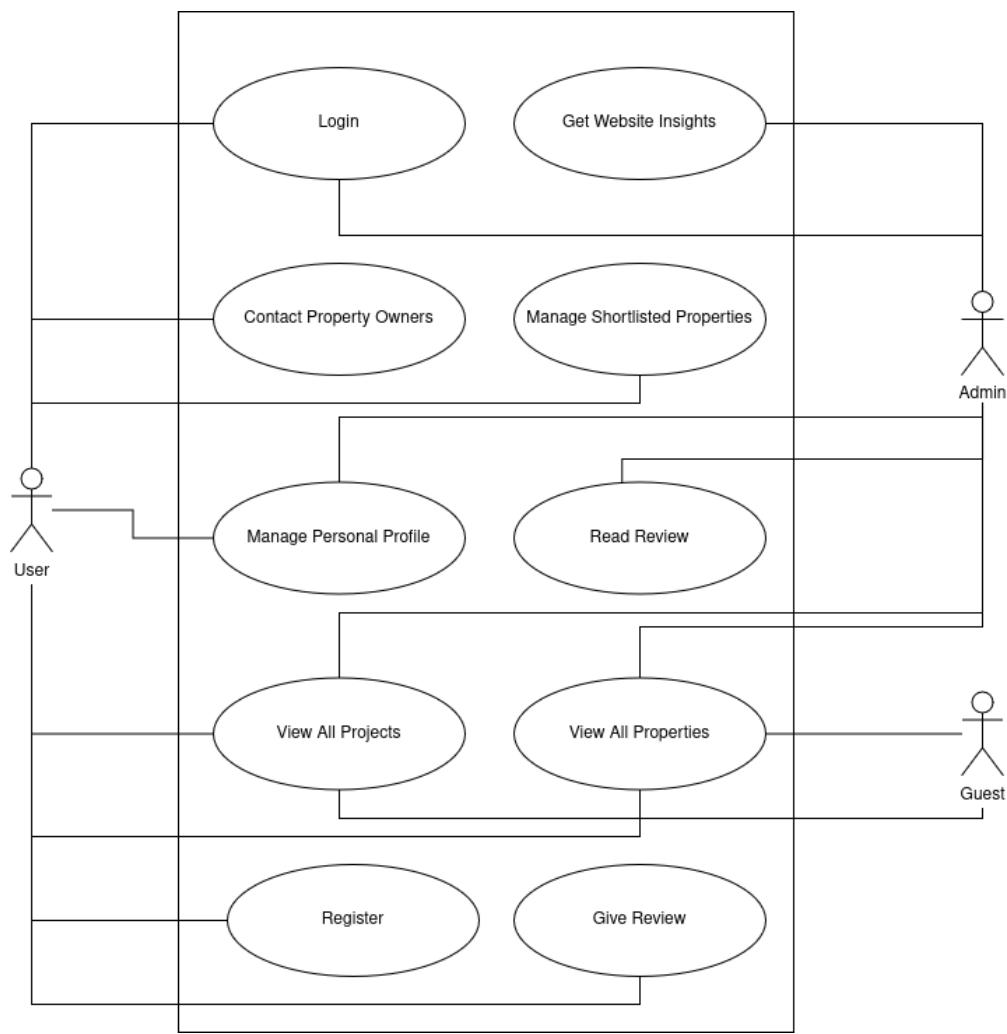


Figure 2.1: Use case diagram for Residize

The Use Case Diagram for Residize visually represents the interactions between user roles (administrators, property owners, tenants, buyers) and the system, highlighting

key functionalities like property management, user authentication, and transaction handling.

### 2.2.2 Sequence diagram

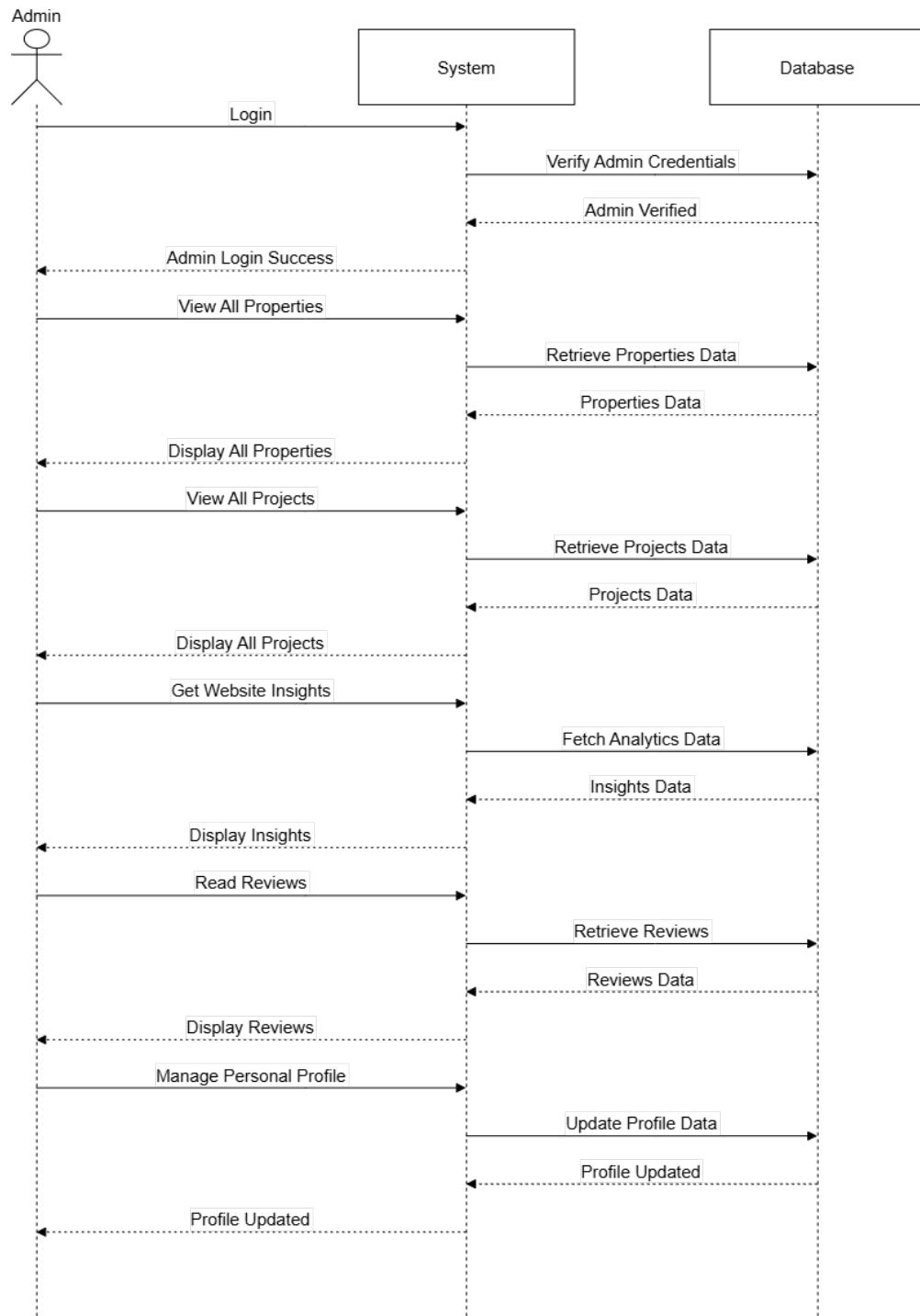


Figure 2.2: Sequence diagram for Admin

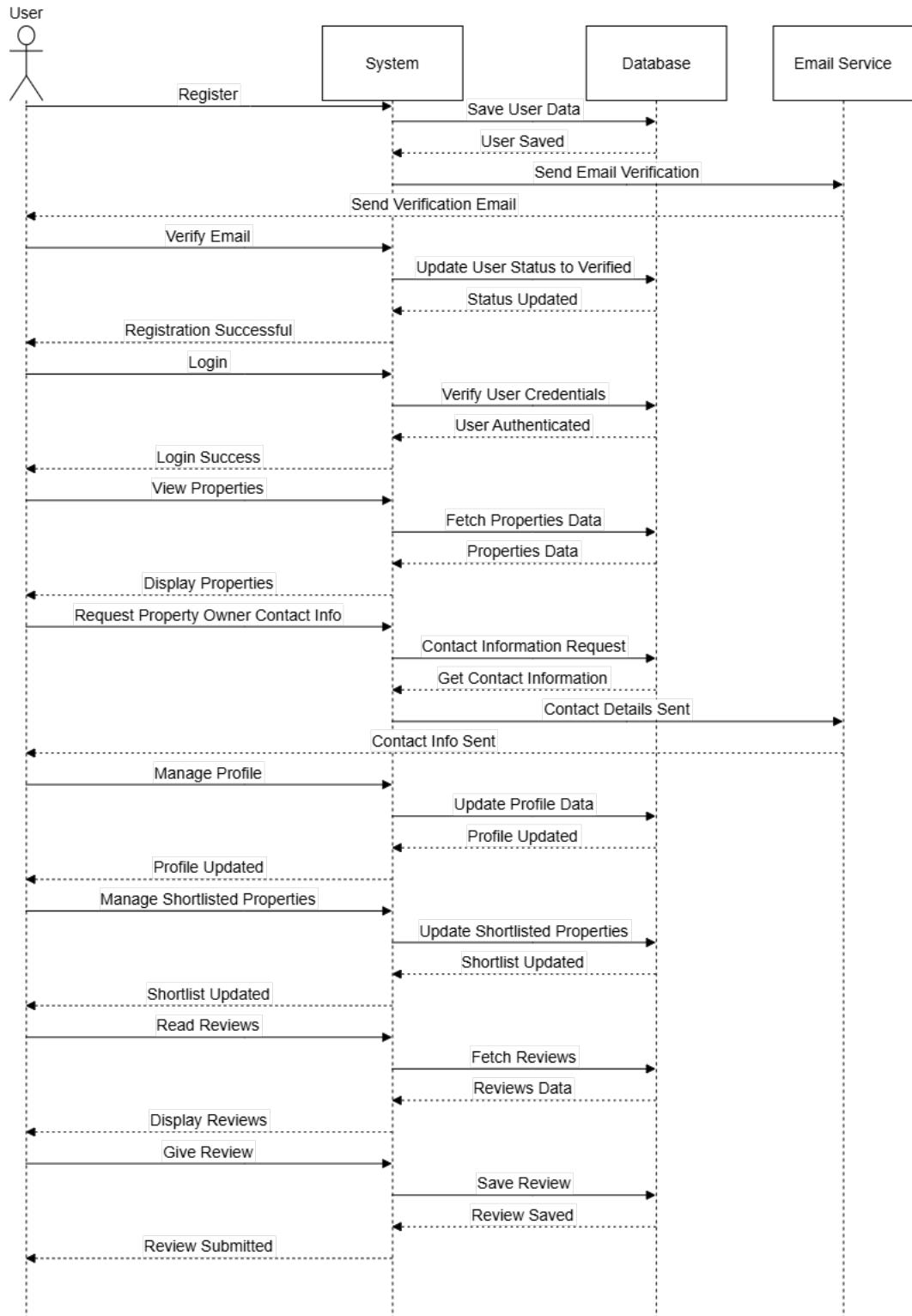


Figure 2.3: Sequence diagram for User

The Sequence Diagram for Residize illustrates the chronological flow of interactions between system components and users, capturing how various objects communicate to

accomplish specific tasks such as user registration, property listing, and booking. It provides a clear view of the dynamic behavior of the system over time.

### 2.2.3 Activity diagram

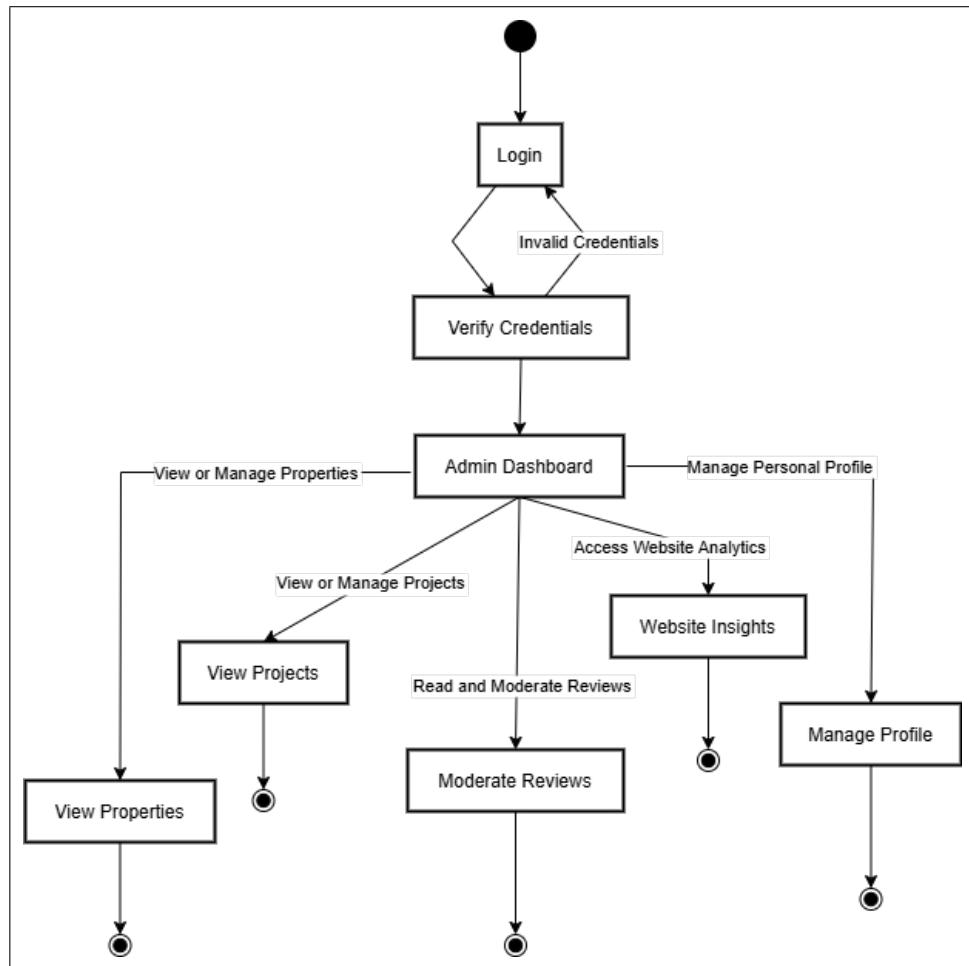


Figure 2.4: Activity diagram for Admin

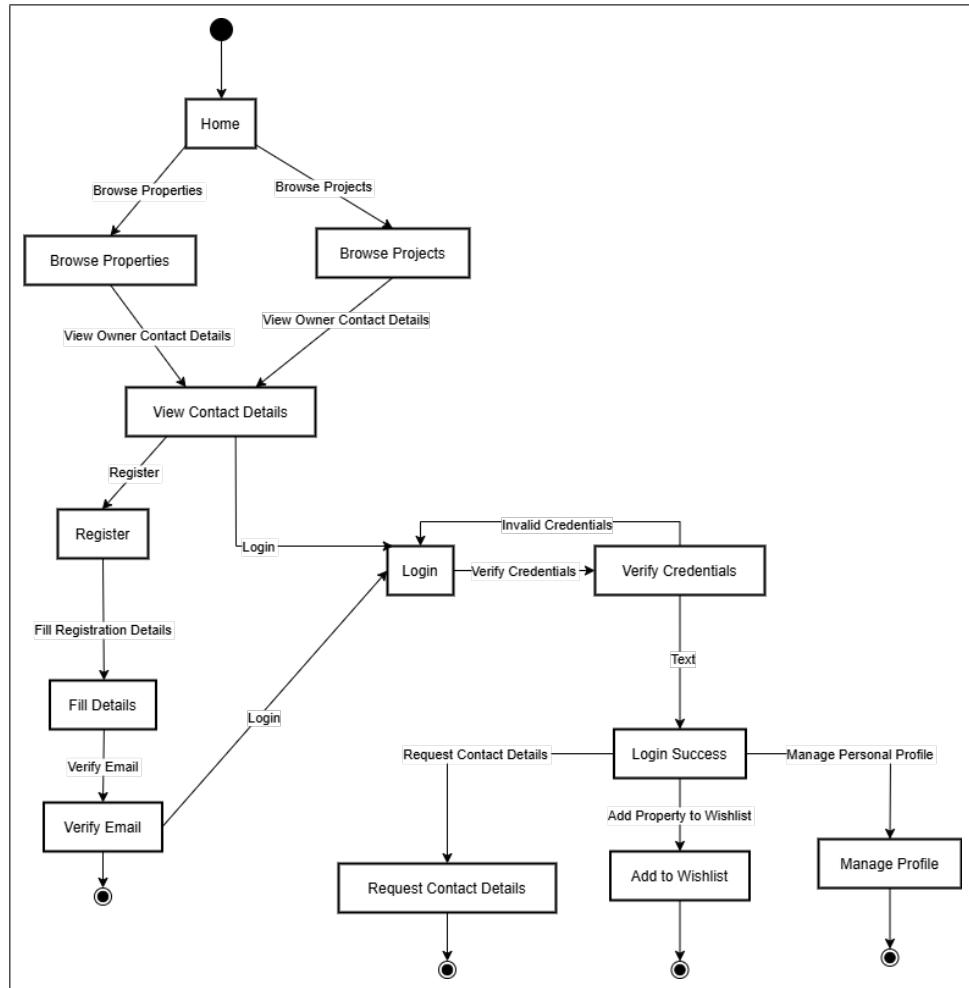


Figure 2.5: Activity diagram for User &amp; Guest

The Activity Diagram of Residize presents a flowchart-style visualization of the sequential and parallel workflows within the system. It captures the progression of activities—such as user registration, property browsing, and booking—while highlighting decision points and conditional logic that drive user interactions and system behavior.

### 2.2.4 Entity relationship diagram

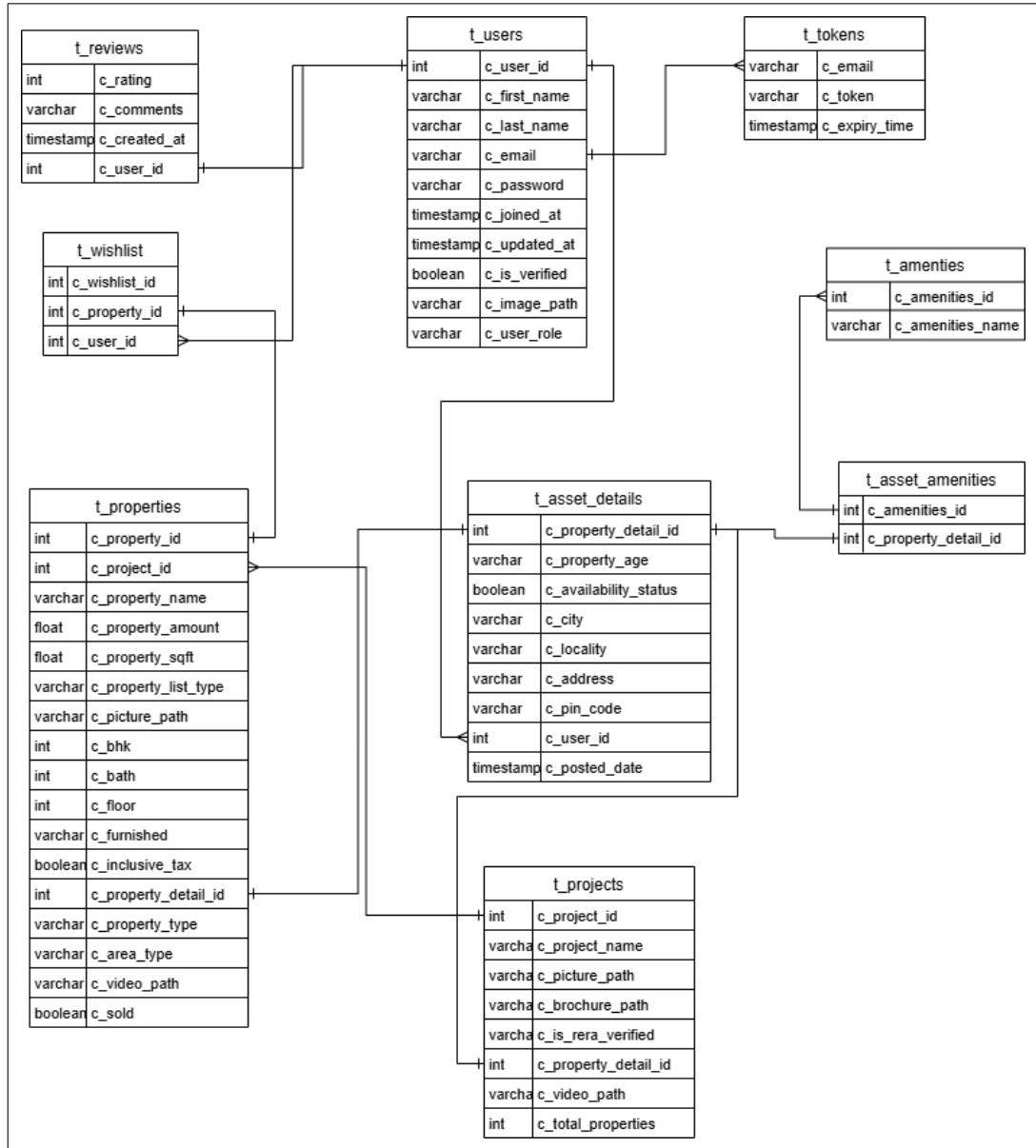


Figure 2.6: ER Diagram for Residize

The Entity Relationship (ER) Diagram of Residize illustrates the core data entities within the system, their respective attributes, and the interrelations among them. It provides a structural overview of the underlying database schema, supporting efficient data management and integrity across modules such as user accounts, property listings, bookings, and transactions.

### 2.2.5 Data flow diagram

#### Admin scope

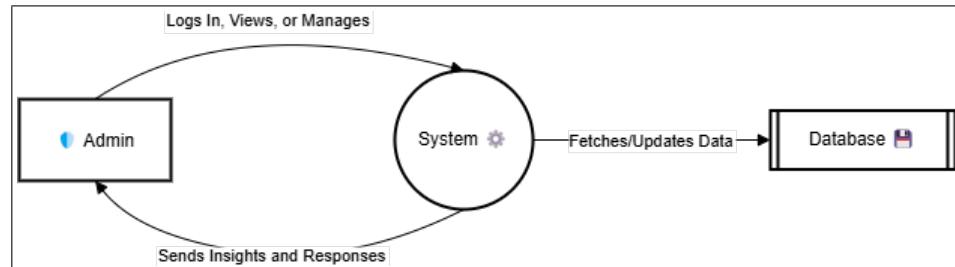


Figure 2.7: Level 0 data flow diagram for Admin

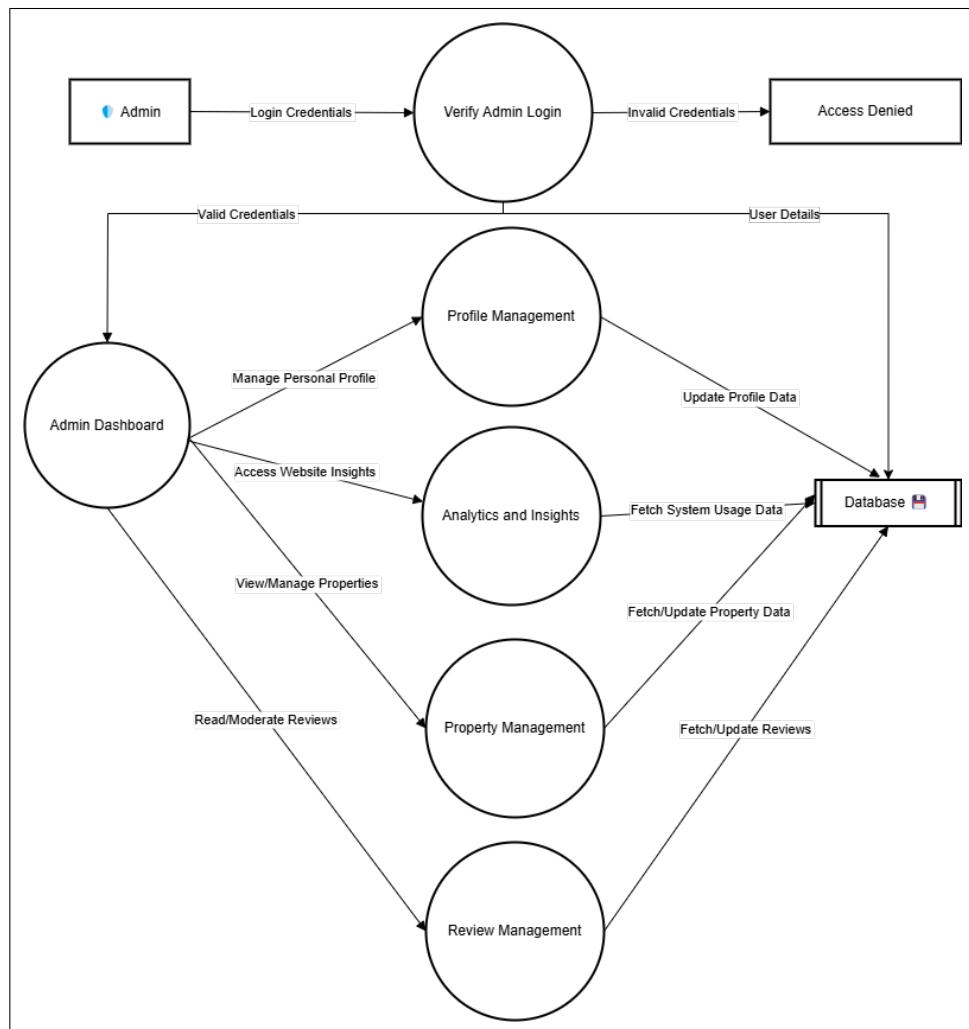


Figure 2.8: Level 1 data flow diagram for Admin

### Guest & User scope

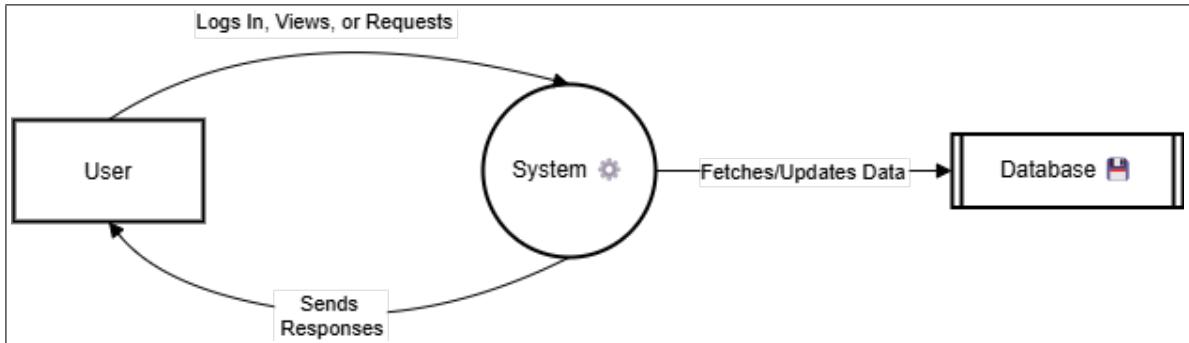


Figure 2.9: Level 0 data flow diagram for User & Guest

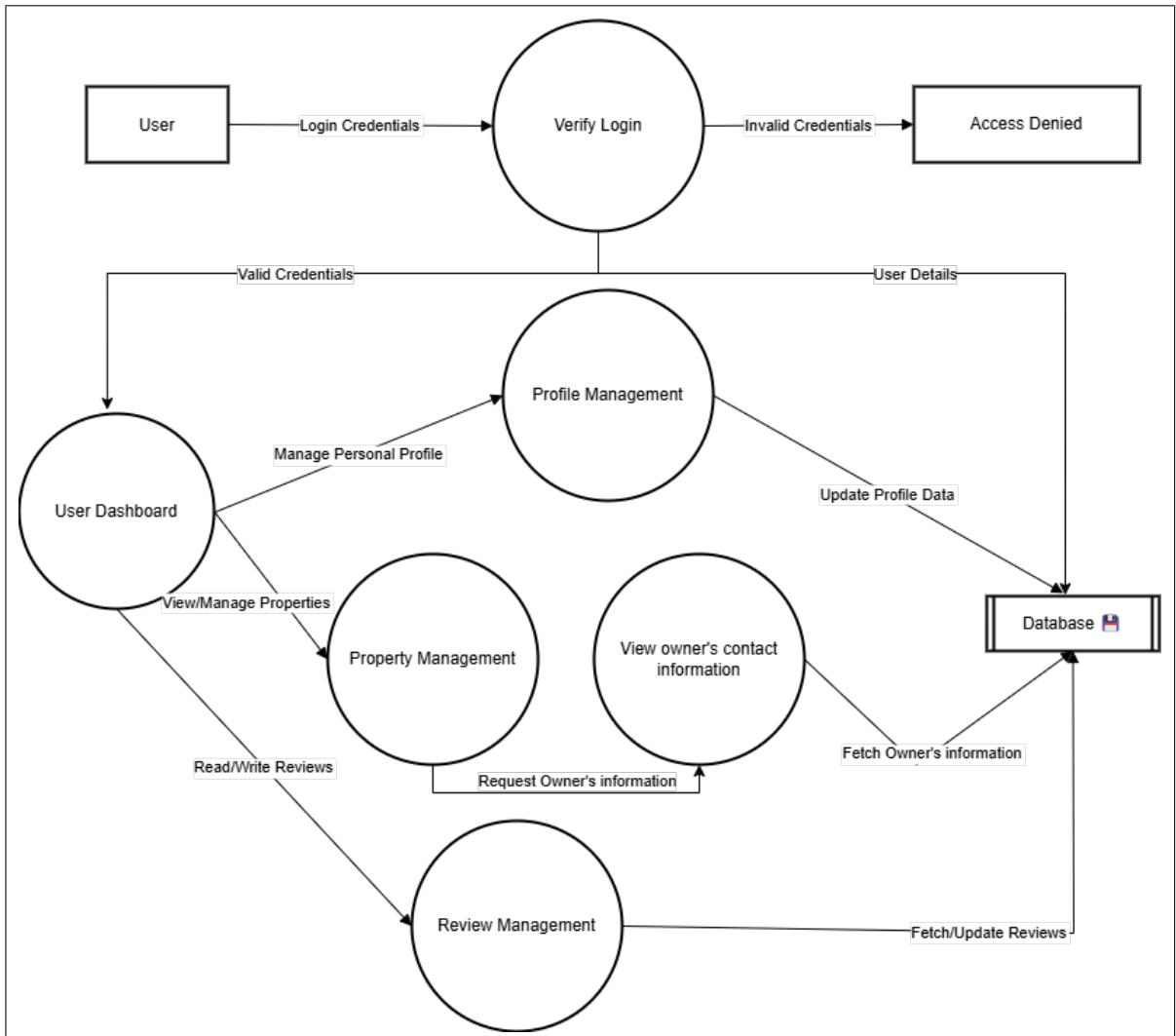


Figure 2.10: Level 1 data flow diagram for User

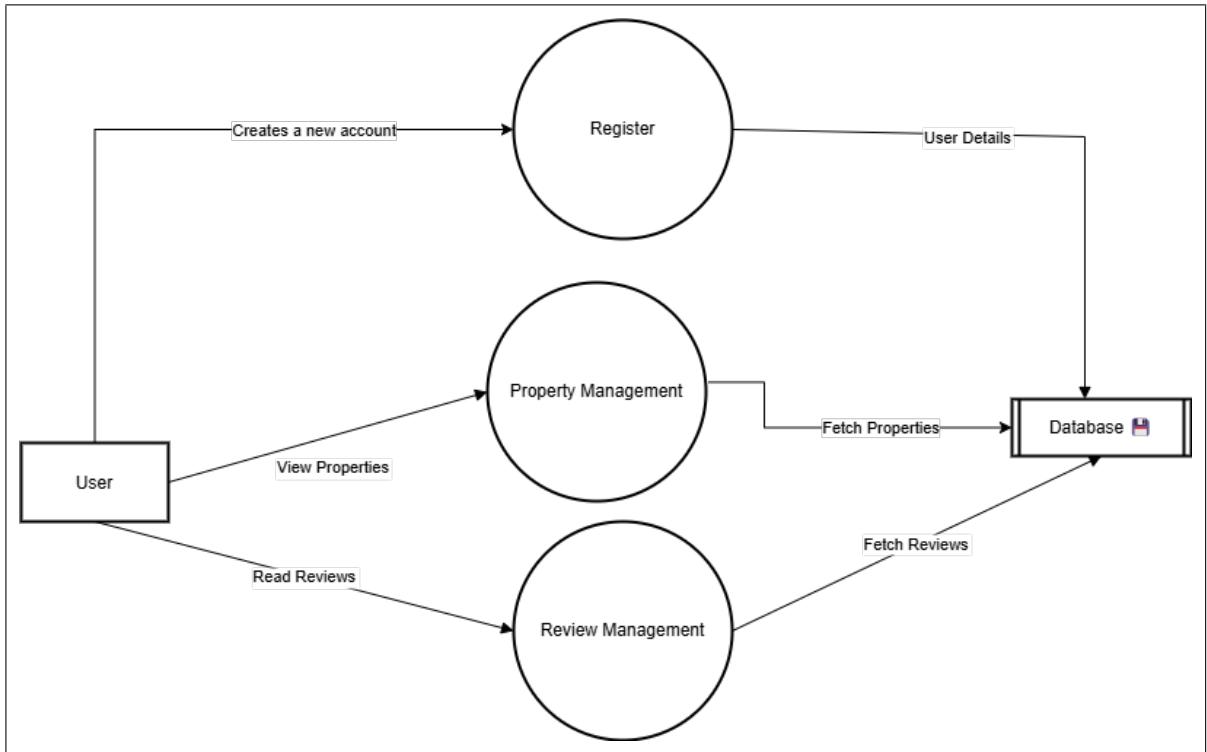


Figure 2.11: Level 1 data flow diagram for Guest

The Data Flow Diagram (DFD) of Residize visualizes how data moves through the system, highlighting the interaction between external entities, internal processes, and data stores. It provides a structured overview of how information is captured, processed, and routed within the platform, ensuring clarity in system functionality and data handling.

### 2.2.6 Database schema

Database schema is a structured framework that defines how data is organized and stored in a database.

Table 2.4: Amenities table

Attributes	Datatype	Description
c_amenities_id	Integer	PRIMARY KEY
c_amenities_name	Character Varying	-

Table 2.5: Token table

Attributes	Datatype	Description
c_email	Character Varying	FOREIGN KEY User(c_email)
c_token	Character Varying	-
c_expiry_time	Timestamp	-

Table 2.6: User table

Attributes	Datatype	Description
c_user_id	Integer	PRIMARY KEY
c_first_name	Character Varying	-
c_last_name	Character Varying	-
c_email	Character Varying	-
c_password	Character Varying	-
c_joined_at	Timestamp	DEFAULT(CURRENT_TIMESTAMP)
c_updated_at	Timestamp	DEFAULT(CURRENT_TIMESTAMP)
c_is_verified	Boolean	DEFAULT(false)
c_image_path	Character Varying	-
c_user_role	Character Varying	DEFAULT('User')

Table 2.7: Reviews table

Attributes	Datatype	Description
c_rating	Integer	-
c_comments	Character Varying	-
c_created_at	Timestamp	DEFAULT(CURRENT_TIMESTAMP)
c_user_id	Integer	FOREIGN KEY User(c_user_id)

Table 2.8: Asset details table

Attributes	Datatype	Description
c_property_detail_id	Integer	PRIMARY KEY
c_property_age	Character Varying	-
c_availability_status	Boolean	-
c_city	Character Varying	-
c_locality	Character Varying	-
c_address	Character Varying	-
c_pin_code	Character Varying	-
c_user_id	Integer	FOREIGN KEY User(c_user_id)
c_posted_date	Timestamp	-

Table 2.9: Projects table

Attributes	Datatype	Description
c_project_id	Integer	PRIMARY KEY
c_project_name	Character Varying	-
c_picture_path	Character Varying	-
c_brochure_path	Character Varying	-
c_is_rera_verified	Character Varying	-
c_property_detail_id	Integer	FOREIGN KEY AssetDetails(c_property_detail_id)
c_video_path	Character Varying	-
c_total_properties	Integer	-

Table 2.10: Asset amenities table

Attributes	Datatype	Description
c_amenities_id	Integer	FOREIGN KEY Amenities(c_amenities_id)
c_property_detail_id	Integer	FOREIGN KEY AssetDetails(c_property_detail_id)

Table 2.11: Properties table

<b>Attributes</b>	<b>Datatype</b>	<b>Description</b>
c_property_id	Integer	PRIMARY KEY
c_project_id	Integer	FOREIGN KEY Project(c_project_id)
c_property_name	Character Varying	-
c_property_amount	Integer	-
c_property_sqft	Numeric	-
c_property_list_type	Character Varying	-
c_picture_path	Character Varying	-
c_bhk	Integer	-
c_bath	Integer	-
c_floor	Integer	-
c_furnished	Character Varying	-
c_inclusive_tax	Boolean	-
c_property_detail_id	Integer	FOREIGN KEY AssetDetails(c_property_detail_id)
c_property_type	Character Varying	-
c_area_type	Character Varying	-
c_video_path	Character Varying	-
c_sold	Boolean	-

Table 2.12: Wishlist table

<b>Attributes</b>	<b>Datatype</b>	<b>Description</b>
c_wishlist_id	Integer	PRIMARY KEY
c_property_id	Integer	FOREIGN KEY Property(c_property_id)
c_user_id	Integer	FOREIGN KEY User(c_user_id)

## 2.3 Implementation

### 2.3.1 Hardware and software requirements

This section outlines the necessary infrastructure and technologies for the development and deployment of the Residize project. It covers the hardware specifications and the software platforms used to ensure efficient system performance.

#### Hardware specifications

- **Processor:** Intel Core i5 - 7200U or Above
- **RAM:** 16 GB (DDR4 - 2133 Mhz)
- **Storage:** 256 GB Solid State Drive

#### Software specifications

- **Operating System:** Fedora GNU/Linux 41 (x86\_64) or Windows 11 (x64)
- **Technology:** .NET Core Stack (Version 8.0.10)
- **Front end:** .NET Core MVC
- **Back end:** .NET Core WebAPI
- **Database:** PostgreSQL (Version 17)
- **IDE:** Microsoft Visual Studio Code (Version 1.95)

## 2.4 Implementation Screenshots

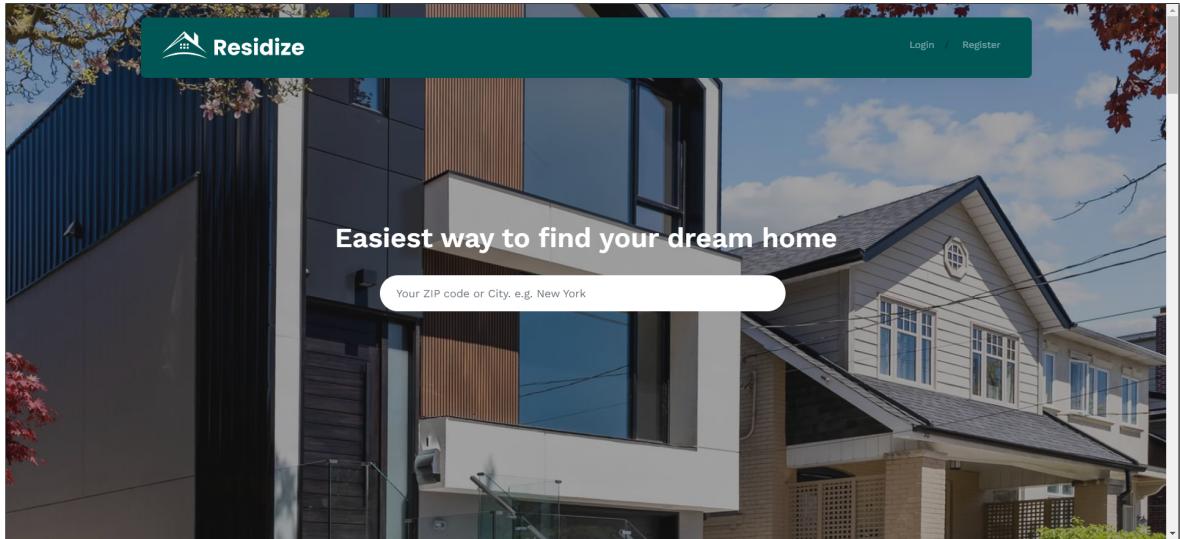


Figure 2.12: Anonymous user's home page

Homepage of the Residize platform for guest users with a search bar to find properties by their name. Guest users can login or register to their accounts.

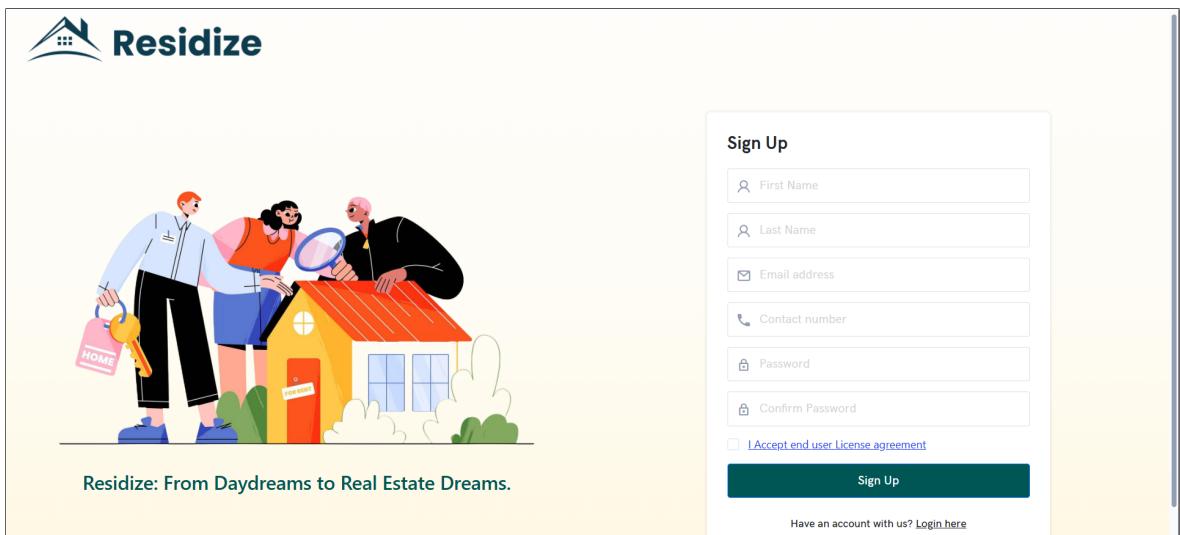


Figure 2.13: Register page

The Register Page of Residize facilitates account creation for guest users, serving as the initial step toward accessing the platform's comprehensive property management services.

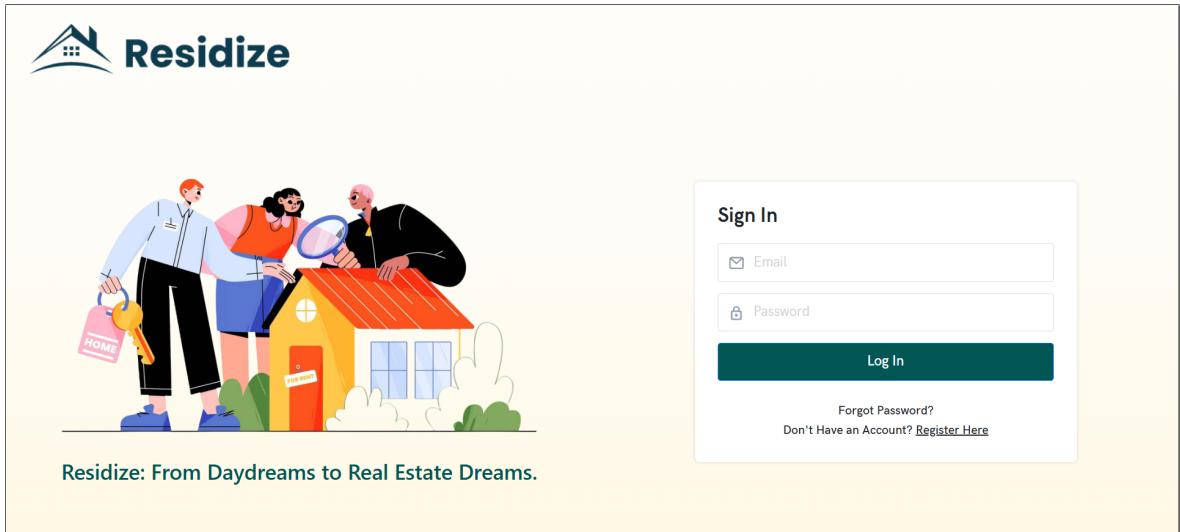


Figure 2.14: Login page

The Login Page of Residize provides returning users with secure access to their accounts, enabling a seamless transition into the platform's property management environment through email and password authentication.

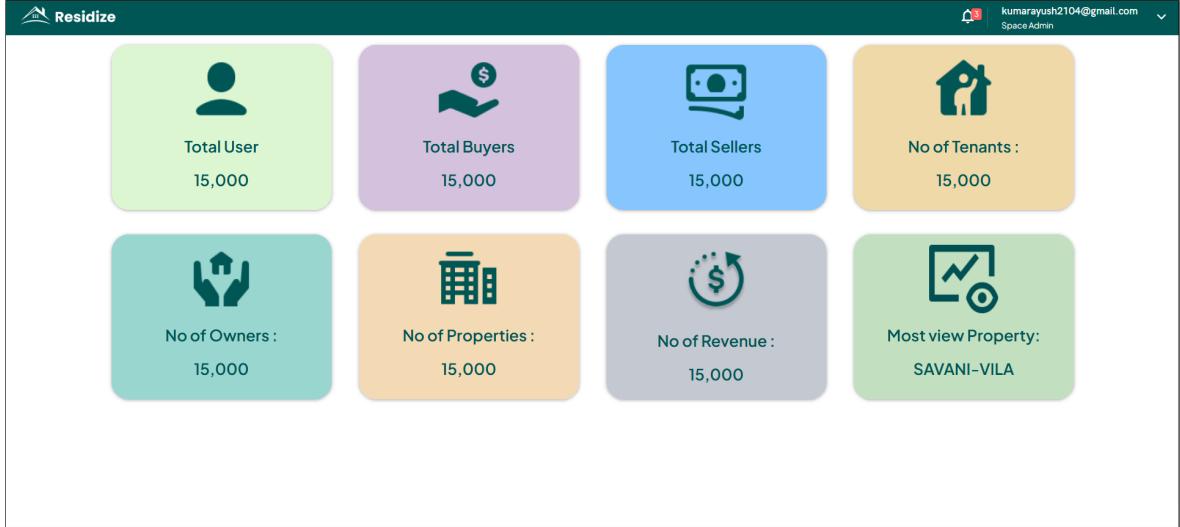


Figure 2.15: Admin home page

The Admin Dashboard in Residize presents a concise overview of platform-wide metrics, offering real-time insights into user distribution, property data, revenue, and the most viewed listing, enabling efficient administrative monitoring and decision-making.



Figure 2.16: User profile view

The user profile panel in the Residize Admin Dashboard provides quick access to personal account details, including full name, contact information, and options for settings and logout. This feature ensures that administrators can manage their identity and session seamlessly while overseeing platform operations.



Figure 2.17: User profile settings

The User Profile Settings in the Residize Admin Dashboard enable administrators to update personal details, manage profile photos, and change passwords—all in a streamlined, user-friendly interface.

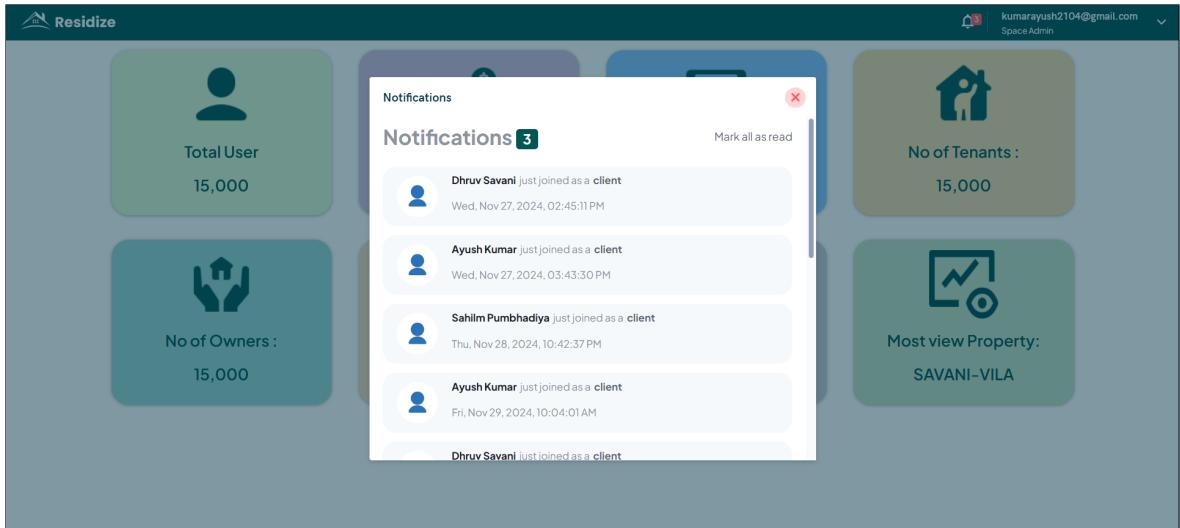


Figure 2.18: Notifications on the dashboard

The Notifications panel in the Residize Admin Dashboard provides real-time updates on user activities, such as new client registrations. It allows administrators to stay informed and manage engagement efficiently with features like timestamped entries and bulk marking as read.



Figure 2.19: Website insights on the admin dashboard

The Admin Dashboard charts in Residize offer clear visual insights into property status and verification. The “Sold vs Unsold Properties” and “Verified vs Unverified Projects” bar graphs help administrators monitor inventory and compliance effectively, enabling data-driven decision-making.

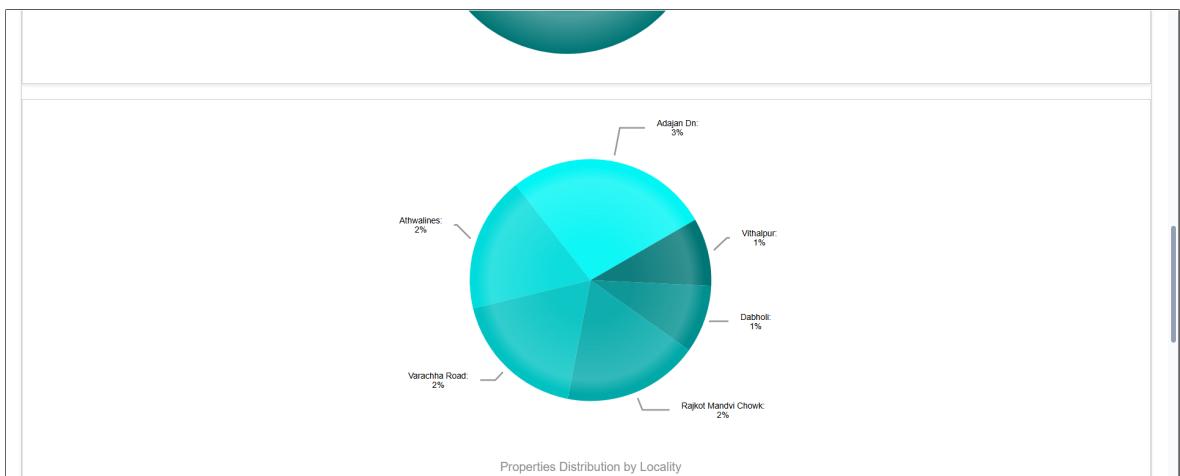


Figure 2.20: Second page of website insights on the admin dashboard

The “Properties Distribution by Locality” chart in the Admin Dashboard visually breaks down property availability across key areas. This enables admins to identify location trends, monitor demand hotspots, and strategize property promotions accordingly.

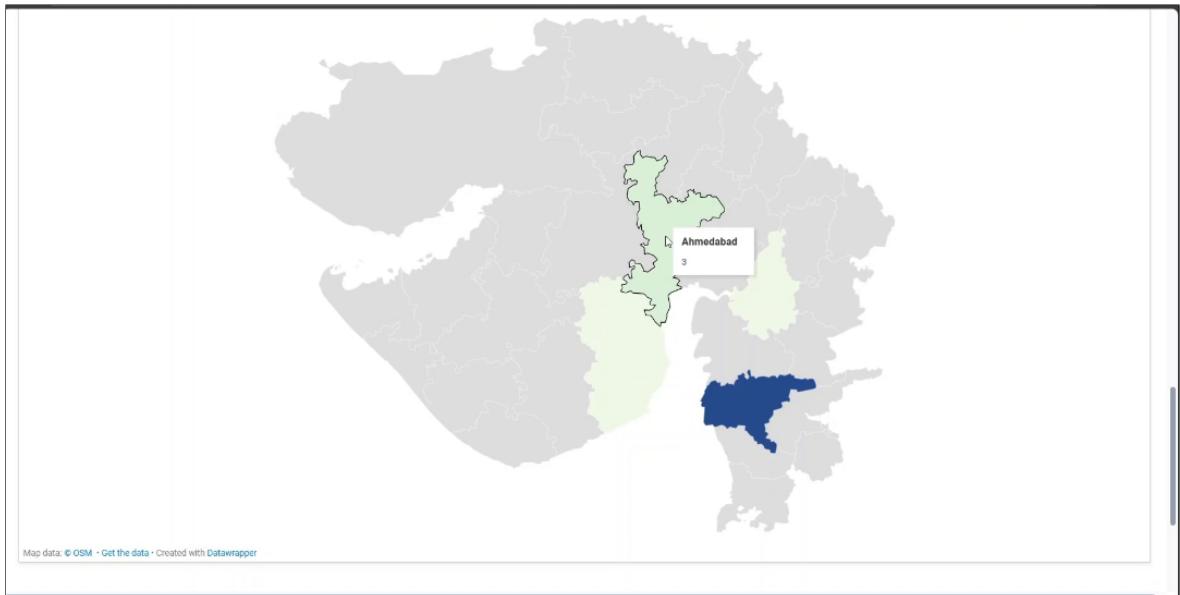


Figure 2.21: Map view in the admin dashboard

The map view in the admin dashboard allows the admin to gain insights about the number of properties and projects posted in each city of Gujarat.

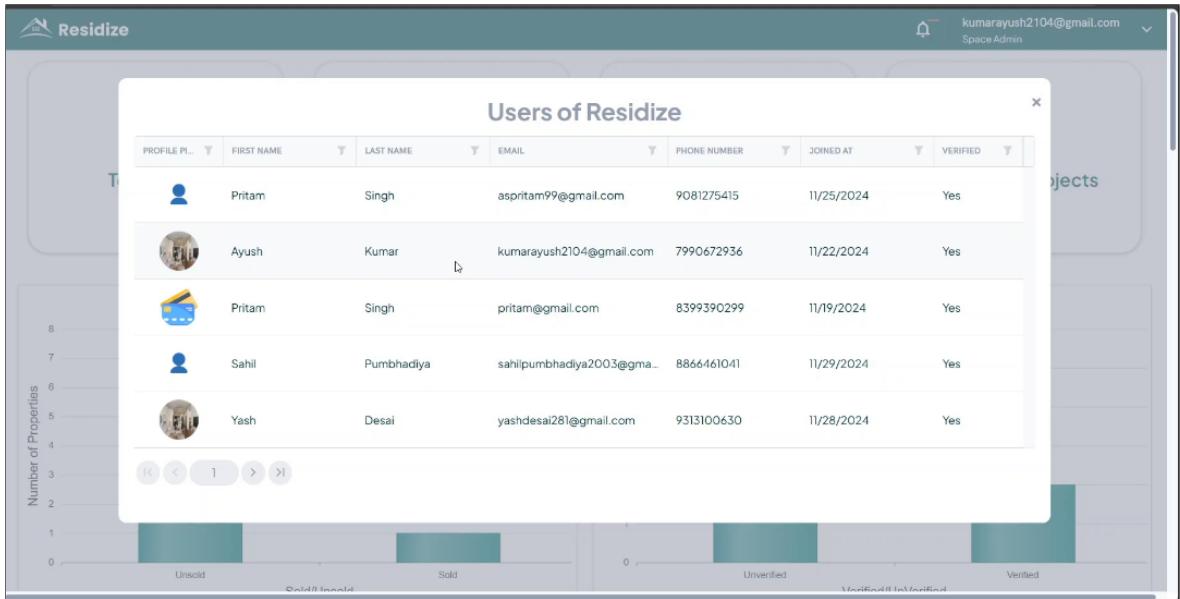


Figure 2.22: Number of active users list in the admin dashboard

The users view in the admin dashboard allows the admin to gain insights about the number of active users in the website. It also includes the user's contact information.

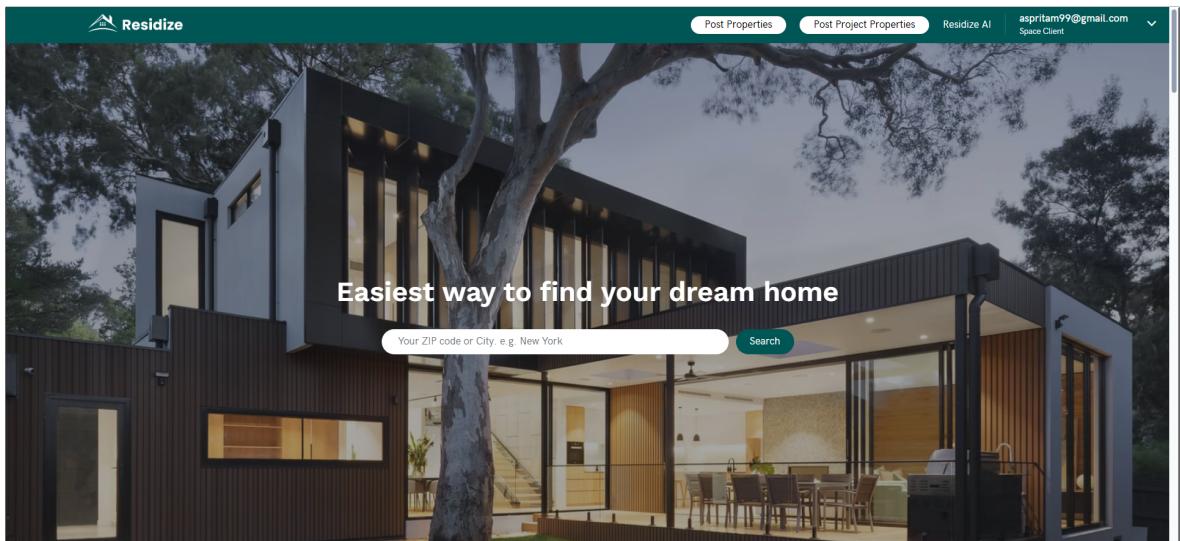


Figure 2.23: User dashboard

The User Home Page offers a clean, modern interface focused on simplicity and ease of use. With a prominent search bar for ZIP code or city input, users can quickly begin exploring listings. Key features include intuitive navigation to post properties or access smart recommendations via Residize AI, along with a visible user email to confirm login status.

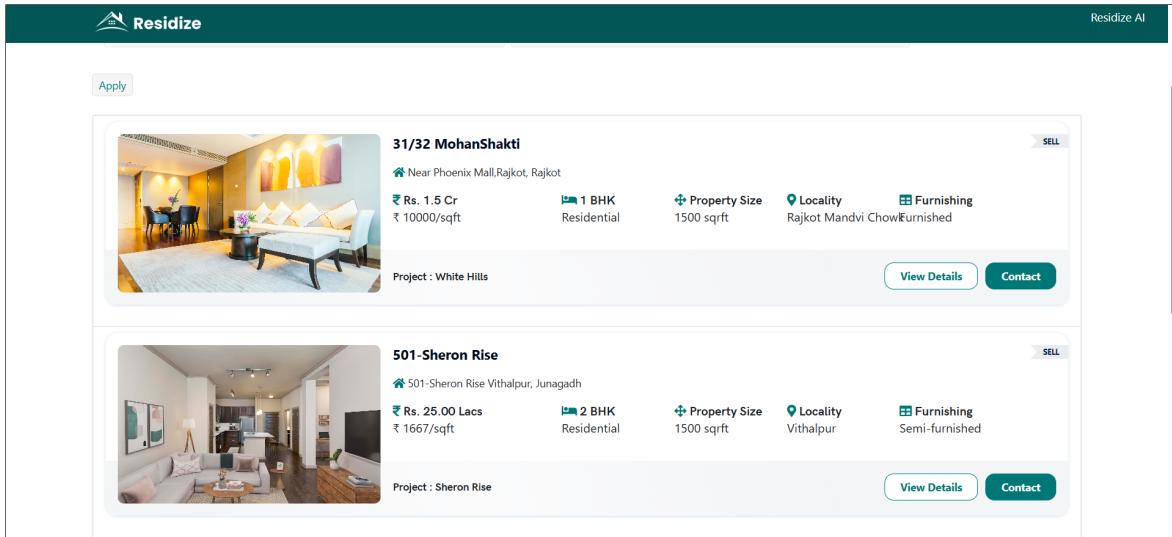


Figure 2.24: Listed properties on the user dashboard

The “View All Properties” page lists detailed property cards, including price, size, location, and furnishing status. Users can easily compare listings and access options like ”View Details” and ”Contact” for streamlined property exploration and inquiries.

The screenshot shows the Property Details page for "A-703 Shiv Shakti". It features a large photo of a modern living room, a summary card with key details, and sections for amenities. The summary card includes: Area 3000 sq ft, Configuration 2 BHK, Price ₹60.00 Lacs, Property Age 14 Year, Total Floors 6 Floors, Locality Varachha Road, Address Near Umiyadham, Varachha Road, Varachha Road, Surat, and Posted Date 1/3/2025. Below the summary card are buttons for "Property Listing Type: Sell", "Property Type: Residential", "Furnishing: Semi-furnished", and "Area Type: Carpet Area". The "Amenities" section lists: CCTV, Power Backup, Wi-Fi, Fire Safety, Library, and Sports Area.

Figure 2.25: Property detailed view

The Property Details page showcases a comprehensive listing with photos, price, area, configuration, age, location, and amenities like Wi-Fi, library, and power backup. Users can view the full details and request contact info easily.

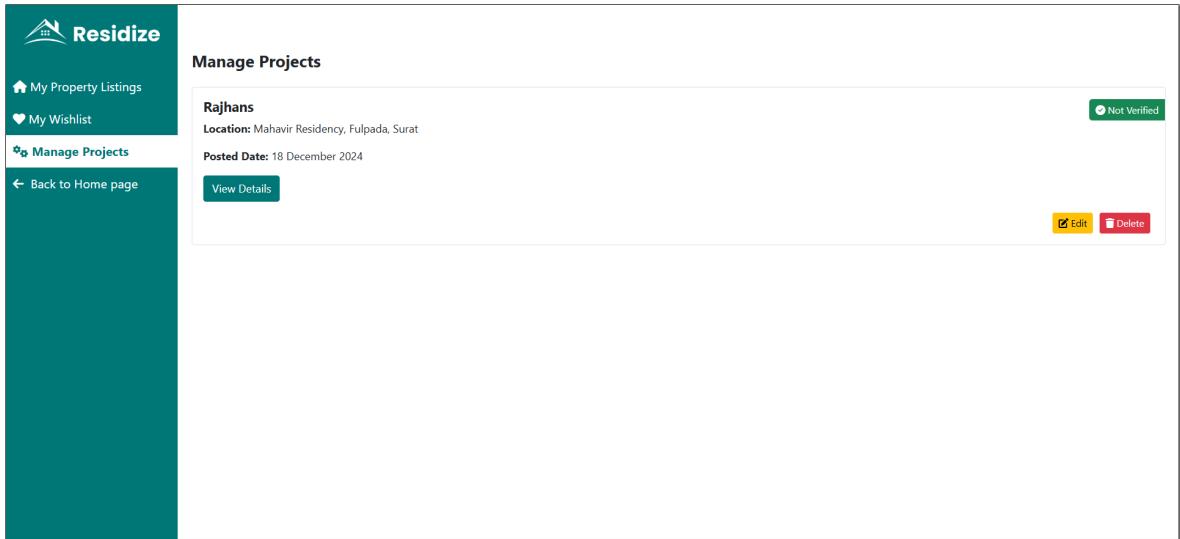


Figure 2.26: User property management

The Manage Projects page displays a list of created projects with details like location, post date, and verification status. Users can view, edit, or delete each project entry directly from the interface.

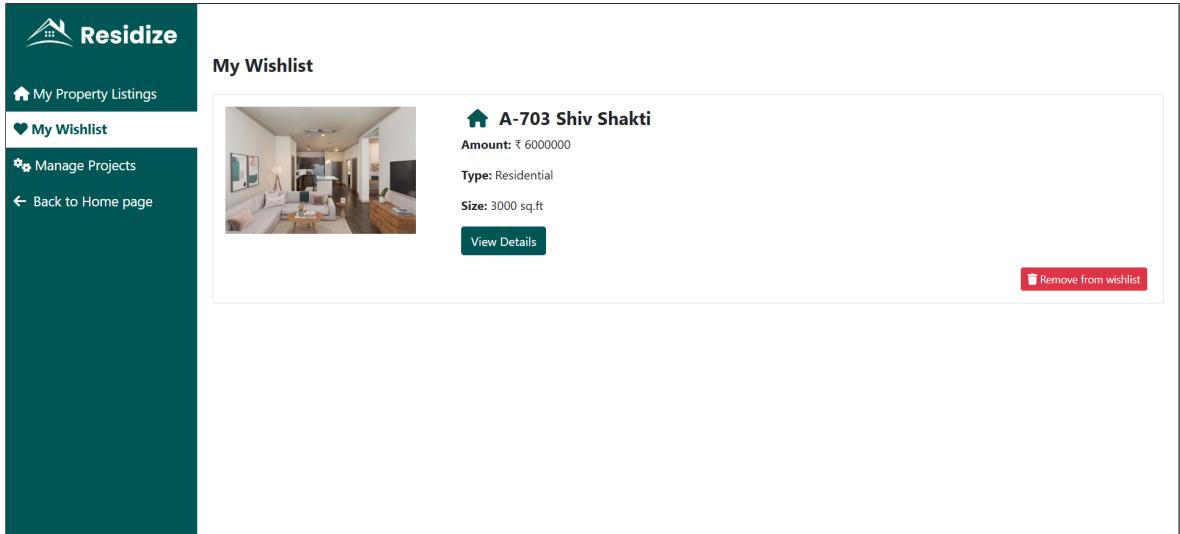


Figure 2.27: Wishlist section of user property management

The My Wishlist page allows users to keep track of their favorite properties. Each entry displays a thumbnail image, property title, price, type, and size. Users can view more details or remove the property from their wishlist with the "Remove from wishlist" button.

The screenshot shows the 'Location Details' section of the Residize Add Property form. On the left, a vertical sidebar lists four steps: 1. Location Details (highlighted in blue), 2. Property Profile, 3. Photos, Videos & Voice-over, and 4. Amenities Section. The main area is titled 'Location Details' with the sub-instruction 'Provide the location details of the property.' It contains several input fields: 'Property Name' (A-501 Nakshatra Heights), 'Pincode' (395009) with a 'Get City' button, 'City' (Surat), 'Locality' (Adajan Dn), and an 'Address' field (A-501 Nakshatra Heights, Adajan Road). A small map icon is visible at the bottom right of the address field.

Figure 2.28: Add property form

The "Location Details" section of the Residize Add Property form allows users to enter essential property information, including name, pincode, city, locality, and full address. It is the first step in a streamlined, multi-step property submission workflow.

The screenshot shows the 'Photos, Videos & Voice-over' section of the Residize Add Property form. The sidebar indicates this is step 3. The main area starts with the heading 'Add photos of your property' and a sub-instruction 'A picture is worth a thousand words. 87% of buyers look at photos before buying'. Below this is a 'Upload from desktop' button. Four thumbnail images of the property are displayed in a 2x2 grid: a construction site view, an aerial view of the building complex, a view of the building's exterior, and another view of the building's exterior. The background of the main area features a large, faint image of a modern building complex.

Figure 2.29: Second page of add property form

The "Photos, Videos & Voice-over" section in the Residize Add Property form enables users to upload visual media of the property. This feature helps attract potential buyers by showcasing high-quality images, as visual content significantly influences purchasing decisions.

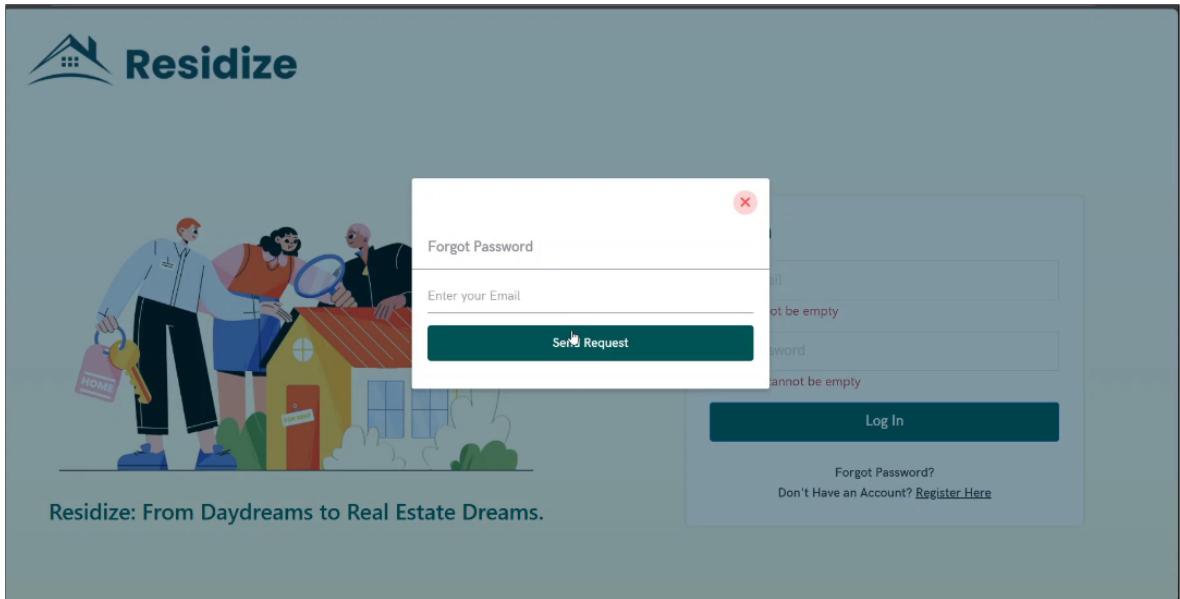


Figure 2.30: Forget password page

Forget password section in the Residize enables users to reset their password securely.

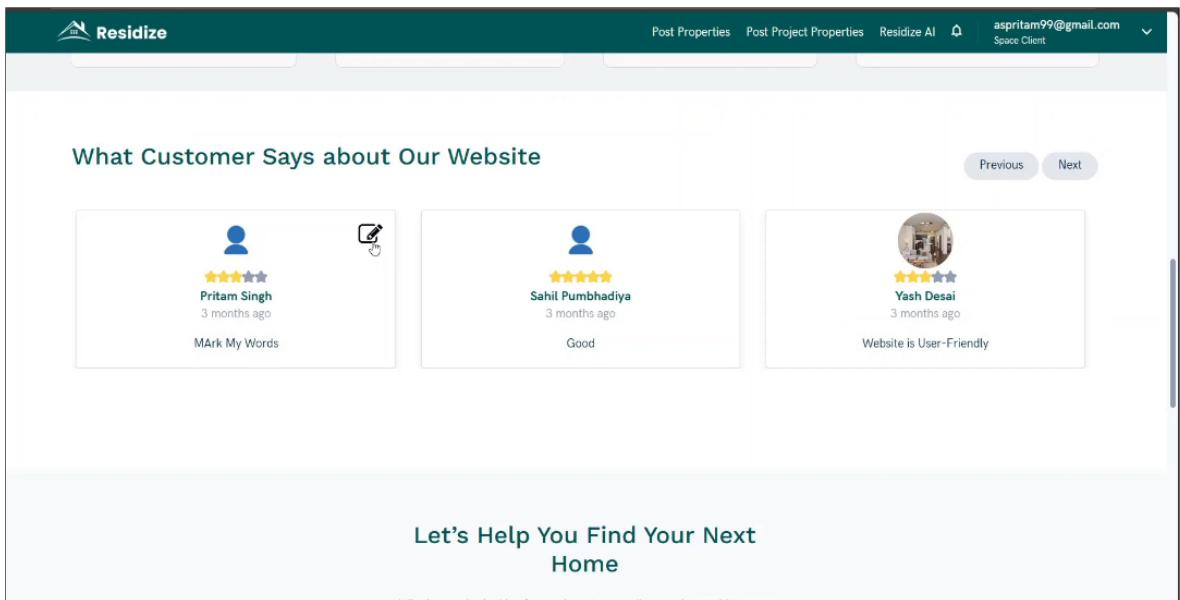


Figure 2.31: Reviews section in the user's home page

Reviews section enables the user to read the reviews given by the other users for the website's experience, which helps the developers to improve the website experience according to the users.

## 2.5 Test Cases

Table 2.13: Test case results of iteration 1

Iteration 1 (18th November 2024 to 29th November 2024)				
Test ID	Modules	Expected Result	Actual Result	Pass/Fail
T01	Registration	Verify all the details are valid or not and register the user	As Expected	Pass
T02	Login	Verify credentials and redirect to their dashboard	As Expected	Pass
T03	User Profile	User should be able to view and change their personal details	As Expected	Pass
T04	Forget Password	If user forget their password, they should be able to reset the password	As Expected	Pass
T05	Notification System	Admin should get notifications about the user registrations	As Expected	Pass

Table 2.14: Test case results of iteration 2

<b>Iteration 2 (9th December 2024 to 20th December 2024)</b>				
<b>Test ID</b>	<b>Modules</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
T01	Manage Projects	Users can post their project for sell or rent.	As Expected	Pass
T02	Manage Properties	Users can post their properties for sell or rent.	As Expected	Pass
T03	View listed properties	Users can view the properties and perform filter operations to get their desired properties.	As Expected	Pass
T04	Exchange contact information	Users can request the contact information of the owners.	As Expected	Pass
T05	Add reviews about the website	User can post their review about the website.	As Expected	Pass
T06	View reviews about the website	Users can view the reviews about the website which is written by other users.	As Expected	Pass

Table 2.15: Test case results of iteration 3

<b>Iteration 3 (23rd December 2024 to 3rd January 2025)</b>				
<b>Test ID</b>	<b>Modules</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
T01	Admin Dashboard	Admin can get the insights of the website.	As Expected	Pass
T02	User Wishlist	Users can add and view the properties from the wishlist.	As Expected	Pass
T03	RERA verified projects	User can view the RERA-verified projects.	As Expected	Pass
T04	Search Properties based on user criteria	User can search the properties and projects based on their requirements.	As Expected	Pass

## 2.6 Conclusion

In conclusion, the Residize system is built for scalability, efficiency, and security, with an Agile approach ensuring continuous improvement. The design supports seamless user interaction, and the identified hardware and software requirements ensure optimal performance.

# Chapter 3

## Key Learning

This chapter provides an overview of the training activities undertaken during the project. The training focused on building a strong foundation in key technologies essential for the development of *Residize*, equipping interns with essential full-stack development skills.

### 3.1 Technical Skills

#### 3.1.1 HTML and CSS

HTML (HyperText Markup Language) forms the structural foundation of web content, providing a semantic framework for defining the organization, hierarchy, and structure of web pages. CSS (Cascading Style Sheets) is responsible for controlling the visual presentation, layout, and responsiveness of web elements. Together, these two technologies are indispensable in front-end web development, forming the backbone of accessible, responsive, and visually compelling user interfaces.

Throughout the training, emphasis was placed on creating clean, maintainable, and standards-compliant code, which not only improves usability but also enhances performance and scalability. The focus was on developing a deep understanding of best practices, modern design principles, and accessibility guidelines. Key competencies developed include:

- **Separation of content and presentation:** Emphasized the importance of separating the content structure (HTML) from the visual presentation (CSS) using external and modular stylesheets. This approach promotes better maintainability and scalability.

ity, scalability, and reusability, ensuring that future updates to the design do not interfere with the underlying content and logic. It also supports a more organized and consistent approach to large-scale web development.

- **Responsive design:** Mastered mobile-first design techniques using media queries to adapt the layout of web pages to different screen sizes and devices. Utilized modern CSS layout systems such as Flexbox and CSS Grid to create fluid and flexible layouts that adjust automatically to different screen dimensions, ensuring optimal viewing experiences across desktops, tablets, and mobile devices. This approach was foundational in delivering seamless user experiences across multiple platforms.
- **Customizability and theming:** Leveraged CSS variables, utility classes, and preprocessor languages such as Sass/SCSS to create dynamic, maintainable, and themeable stylesheets. This allowed for easier customization, particularly in adapting styles for different brands or user interfaces. The use of animations, transitions, and pseudo-elements was integrated to create engaging and interactive user interfaces, aligned with brand guidelines and visual design requirements.
- **Accessibility and semantics:** Focused on ensuring that HTML content adhered to semantic principles, making it both machine-readable and user-friendly, particularly for users with disabilities. Implemented ARIA (Accessible Rich Internet Applications) attributes to enhance accessibility for screen readers and other assistive technologies. Adhered to WCAG (Web Content Accessibility Guidelines) to ensure that all web pages were navigable, readable, and usable by all users, regardless of their abilities.
- **Cross-Browser compatibility:** Applied vendor prefixes and fallback techniques to ensure consistent styling and behavior of web elements across various web browsers, including Chrome, Firefox, Safari, and Internet Explorer. Tested and resolved layout issues specific to different browser engines, ensuring that the user experience was uniform across all major platforms.

- **Performance optimization:** Focused on optimizing CSS for performance by minimizing the use of expensive properties and reducing unnecessary reflows and repaints. Techniques such as CSS minification, image optimization, and using critical CSS for above-the-fold content were employed to ensure fast load times and smooth page rendering.
- **Modern CSS frameworks:** Gained experience with popular CSS frameworks such as Bootstrap and Tailwind CSS, which provided pre-built responsive grid systems, components, and utilities for rapid development. This was balanced with custom CSS to meet specific design and functionality requirements.
- **CSS architecture and methodologies:** Emphasized the use of methodologies such as BEM (Block Element Modifier) and SMACSS (Scalable and Modular Architecture for CSS) to maintain consistent, scalable, and modular CSS code, particularly in large web applications. This approach improves collaboration and ensures that the codebase remains organized as it grows in complexity.

### 3.1.2 JavaScript and jQuery

JavaScript is a core scripting language for adding interactivity and dynamic behavior to web applications. Training focused on modern ES6+ syntax, event handling, asynchronous programming, and modular design patterns. jQuery, a widely-used JavaScript library, was employed to simplify DOM traversal, manipulation, and AJAX requests, while ensuring cross-browser compatibility. Key areas of focus included:

- Developing interactive UI components (e.g., modals, sliders, tabs).
- Implementing form validation and client-side error handling.
- Enhancing performance through event delegation and minimized DOM reflows.

### 3.1.3 PostgreSQL

PostgreSQL is a powerful, open-source object-relational database management system (ORDBMS) known for its reliability, feature richness, and adherence to SQL standards.

Within the Residize platform, PostgreSQL was selected as the primary data store due to its advanced capabilities, performance, and extensibility. The training program covered a broad spectrum of topics, ranging from foundational concepts to advanced database techniques:

- **Database design:** Emphasized the importance of sound relational design principles, including data normalization (up to 3NF), use of foreign keys to maintain referential integrity, and application of indexing strategies (B-tree, GIN, and partial indexes) to optimize data access patterns and reduce query latency.
- **Query optimization:** Techniques such as join optimization, subquery rewriting, and use of materialized views were applied to improve execution times and resource utilization.
- **Advanced data types:** Leveraged PostgreSQL's support for non-relational data through native JSON and JSONB types, allowing hybrid document-relational schemas. Arrays, composite types, and user-defined types were also utilized to model complex data structures without sacrificing performance.
- **Stored Procedures and Triggers:** Used PL/pgSQL to write stored procedures and triggers for encapsulating business logic at the database layer, ensuring data consistency and enabling reactive workflows.

### 3.1.4 .NET Core

.NET Core is a high-performance, open-source, cross-platform development framework created by Microsoft. It is designed for building modern, cloud-based, and internet-connected applications. Within the Residize project, .NET Core was utilized to implement both the back-end logic and the API layer, offering scalability, modularity, and seamless integration with various services. The training encompassed core development concepts and advanced features, including:

- **.NET Core MVC:** Developed modular and maintainable web applications using the Model-View-Controller (MVC) pattern. Emphasized the use of Razor views,

strongly-typed models, and view components to deliver dynamic and responsive user interfaces. Dependency injection was used throughout the application to support testability and reduce tight coupling between components.

- **.NET Core WebAPI:** Built RESTful APIs that serve as the backbone for communication between the front-end and back-end systems. Implemented authentication and authorization protocols using JWT (JSON Web Tokens) and OAuth 2.0, enabling secure access control. APIs followed best practices, including proper status codes, versioning, and input validation to ensure robust service contracts.
- **Middleware and configuration:** Configured and extended the HTTP request pipeline using custom middleware for logging, exception handling, and request processing. Employed environment-based configuration (e.g., appsettings.json, environment variables) to manage application behavior across development, staging, and production environments.
- **Entity Framework Core (EF Core):** Used EF Core for object-relational mapping (ORM), enabling seamless interaction with the PostgreSQL database. Implemented code-first migrations, LINQ queries, and repository patterns for clean data access and maintainability.
- **Asynchronous programming and performance optimization:** Leveraged asynchronous programming with `async/await` for non-blocking operations and optimized API response times under concurrent load.

### 3.1.5 Kendo UI

Kendo UI by Telerik is a powerful front-end component library that offers a wide range of ready-to-use UI widgets and tools designed for building modern, interactive, and responsive web applications. In the context of the Residize project, Kendo UI significantly accelerated front-end development and enhanced user experience. Training focused on effectively leveraging the library's capabilities in real-world scenarios, with

emphasis on the following areas:

- **Integration with .NET Core and JavaScript frameworks:** Kendo UI was seamlessly integrated with ASP.NET Core MVC and Razor views, as well as vanilla JavaScript modules. This allowed for dynamic rendering of complex data-driven components such as data grids, charts, and dropdowns without compromising performance.
- **Component customization:** The training covered extensive configuration and customization of Kendo UI components—including data grids, schedulers, charts, modals, and form controls—to align with business-specific requirements. This included templating, conditional formatting, and custom validation logic to enhance user workflows.
- **Usability and accessibility:** Kendo UI's built-in support for themes, localization, and ARIA accessibility standards was utilized to improve the overall user interface design. Custom themes were applied to align the visual identity of the Residize platform with modern UX best practices.
- **Data binding and state management:** Explored one-way and two-way data binding techniques with Kendo components, enabling real-time updates and interactive interfaces. Integration with back-end APIs ensured seamless CRUD operations through AJAX and observable data models.
- **Performance optimization:** Implemented virtual scrolling, lazy loading, and data paging in Kendo Grid components to maintain responsive performance, even when handling large datasets.

### 3.1.6 Redis

Redis is a high-performance, in-memory key-value data store widely used for building fast, scalable, and real-time applications. As part of the Residize system, Redis was utilized to improve system responsiveness and reliability in various contexts. Key areas of implementation include:

- **Caching:** Redis was employed to cache frequently accessed data such as user sessions, property listings, and dashboard statistics. This significantly reduced the load on the primary database and improved overall response times, particularly during peak usage periods.
- **Session management:** By storing session information in Redis, Residize supports scalable and stateless application instances, which is critical for distributed deployments and load balancing. This approach ensures consistent session handling across multiple servers.
- **Real-Time analytics:** Redis' publish/subscribe (pub/sub) model and support for time-series patterns enabled real-time data tracking and live updates. This was particularly useful for monitoring user activity, system metrics, and notification delivery without polling the database.
- **Data expiration and persistence:** Configurable key expiration was used to manage temporary data, such as verification codes and cache entries. Redis' persistence options (RDB and AOF) ensured critical in-memory data could be restored after service restarts.

### 3.1.7 RabbitMQ

RabbitMQ is a robust, open-source message broker that enables asynchronous communication and decoupled processing in distributed systems. In the context of Residize, RabbitMQ was introduced to improve system scalability, fault tolerance, and modularity by enabling reliable message-based workflows between services. Key aspects covered during training include:

- **Message queuing and routing:** Leveraged RabbitMQ's support for different exchange types (direct, topic, fanout, headers) to efficiently route messages to appropriate queues. This allowed for decoupling producers and consumers, reducing service dependencies and enhancing system responsiveness.
- **Scalability and fault tolerance:** Employed features such as durable queues,

persistent messages, and message acknowledgments to ensure reliability. Retry logic and dead-letter queues (DLQ) were implemented to gracefully handle failed message processing and prevent data loss. Clustering and high-availability configurations were also explored to support horizontal scaling.

- **Integration patterns:** Applied enterprise messaging patterns including:
  - **Publish/Subscribe:** Enabled real-time notifications and event broadcasting across microservices.
  - **Work queues:** Distributed workload across multiple consumers to improve throughput and system responsiveness.
  - **Remote Procedure Calls (RPC):** Facilitated asynchronous service-to-service communication, enabling decoupled business logic execution while maintaining the ability to receive responses.
- **Monitoring and management:** Utilized the RabbitMQ management console to track message flow, queue status, and system health. Integration with logging and alerting tools ensured observability of the message pipeline.

## 3.2 Soft Skills

Soft skills are essential for professional success and can significantly impact workplace dynamics. While technical skills are crucial for performing specific tasks, soft skills play a pivotal role in building effective working relationships, driving innovation, and fostering a positive work environment. Below are several key soft skills that are highly valued in any professional setting:

### 3.2.1 Communication

Effective communication is the cornerstone of successful collaboration in any workplace. It involves the ability to share and receive information clearly and respectfully, both within teams and with external stakeholders. Mastering communication involves

several key components:

- **Verbal communication:** The ability to articulate ideas, thoughts, and information clearly and effectively is a crucial skill in the workplace. Clear verbal communication minimizes misunderstandings and ensures that messages are delivered and received accurately. Professionals who communicate effectively can convey complex concepts in a way that is easy for their audience to understand. They are also adept at adjusting their tone, language, and style to fit the situation, whether in a meeting, presentation, or one-on-one conversation.
- **Non-Verbal communication:** Non-verbal communication is an often overlooked but vital aspect of interactions. It includes facial expressions, body language, gestures, posture, and eye contact. Professionals who are attuned to non-verbal cues can gauge the emotions and intentions of others, even when they are not verbally expressed. Likewise, being aware of one's own non-verbal signals can help maintain professionalism and reduce the risk of miscommunication.
- **Written communication:** In the digital age, written communication is a fundamental skill. Professionals are often required to write emails, reports, proposals, and other forms of documentation. Being able to write clearly and concisely ensures that the message is understood without ambiguity. Good written communication also demonstrates professionalism and attention to detail, contributing to the overall effectiveness of an organization's operations.

### 3.2.2 Teamwork and collaboration

In today's interconnected workplace, the ability to work effectively within a team is more important than ever. Collaboration is essential not only for completing tasks but also for generating new ideas, solving problems, and achieving common goals. The following aspects are critical for fostering successful teamwork:

- Collaborating effectively with others means being open to diverse viewpoints, sharing ideas, and offering constructive feedback. Professionals who contribute to

a team environment encourage open dialogue and create a space where everyone feels heard and valued. They understand that the sum of the team's knowledge and experience often leads to better outcomes than individual effort alone.

- Effective collaboration also involves managing conflicts constructively. Disagreements are inevitable in any team, but how they are handled makes all the difference. Professionals who excel in teamwork can mediate differences, seek compromise, and ensure that conflict doesn't disrupt productivity or morale. They view disagreements as opportunities for growth and better understanding, rather than as threats to team cohesion.
- Being adaptable to team dynamics is also important. Team structures can change over time, and successful professionals embrace these changes and continue to collaborate effectively. Whether a team is large or small, cross-functional, or remote, maintaining a cooperative and positive attitude is essential for long-term success.

### 3.2.3 Problem solving

Problem-solving is a skill that allows individuals to identify and address challenges proactively. In any professional environment, issues are bound to arise, and having the ability to respond quickly and effectively is crucial. The key components of problem-solving include:

- **Critical thinking:** This involves the ability to analyze situations logically, identify potential issues, and assess possible solutions. Professionals who are skilled problem-solvers don't just rely on gut feelings—they use data, research, and experience to make informed decisions. They break complex problems into smaller, more manageable components and explore alternative solutions to find the best course of action.
- **Creativity and innovation:** Problem-solving isn't just about finding the most straightforward solution—it's also about thinking creatively. Innovative solu-

tions can often solve problems in new and more efficient ways. Professionals who demonstrate creativity are open to unconventional ideas and approaches, contributing to a culture of continuous improvement and innovation.

- **Decision-Making under pressure:** Many problems require quick, decisive action, especially in high-pressure situations. Strong problem-solvers remain calm under stress and are able to make thoughtful decisions quickly. They prioritize tasks, gather necessary information swiftly, and make the best decision available in the time they have.

### 3.2.4 Adaptability

The modern workplace is constantly evolving, and professionals must be able to adapt to new challenges, technologies, and ways of working. Flexibility is key to staying relevant and maintaining productivity in the face of change. Essential aspects of adaptability include:

- **Embracing change:** Professionals who embrace change are not afraid of new challenges. They are open to learning new skills, adopting new technologies, and adapting to new organizational structures. Rather than resisting change, they see it as an opportunity for personal and professional growth.
- **Learning agility:** Being adaptable means having the ability to quickly acquire new skills and knowledge. Professionals with strong learning agility can pivot to new tasks or roles with ease and thrive in environments that require constant learning.

### 3.2.5 Conflict resolution

In any workplace, conflicts are inevitable. How they are managed, however, can determine the success of the team and the organization. Effective conflict resolution requires both emotional intelligence and strong communication skills. Key elements of conflict resolution include:

- **Addressing conflicts respectfully:** It's essential to address conflicts with professionalism and respect for all parties involved. Professionals who excel in conflict resolution understand that personal issues or miscommunications often underlie workplace disagreements. They approach these situations with empathy, taking the time to listen to everyone's perspective before proposing solutions.
- **Mediating disputes:** Mediating conflicts is not about choosing sides—it's about finding a mutually beneficial resolution. Professionals skilled in conflict resolution can facilitate conversations between differing parties and help them find common ground. By focusing on solutions and shared goals, they encourage collaboration and preserve relationships within the team.

### 3.2.6 Work ethics

A strong work ethic is the foundation of professional success. It involves a commitment to quality, responsibility, and continuous improvement. The key elements of work ethic include:

- **Reliability and accountability:** Professionals with a strong work ethic consistently meet deadlines, follow through on commitments, and take responsibility for their work. They understand the importance of being dependable and recognize the impact their performance has on the overall success of the team and organization.
- **Self-Motivation and initiative:** Self-motivated individuals don't wait for instructions to take action. They demonstrate initiative by seeking out opportunities for improvement, anticipating challenges, and taking proactive steps to solve problems. This level of drive not only contributes to personal success but also encourages a culture of excellence within the workplace.
- **Commitment to quality:** Professionals with a strong work ethic strive to consistently produce high-quality work. They take pride in their performance, paying attention to detail and ensuring that their contributions meet or exceed

expectations.

- **Time management:** Effective time management is essential for maintaining a strong work ethic. Professionals who manage their time well are able to balance multiple tasks, prioritize their responsibilities, and avoid procrastination. They plan ahead, stay organized, and ensure that they are consistently meeting deadlines and delivering results.

### 3.3 Conclusion

In conclusion, the training equipped interns with essential technical and soft skills for a full-stack development career. The hands-on experience from the Residize project allowed them to apply these skills in real-world scenarios, preparing them for success in the tech industry.

# Chapter 4

## Conclusion and Future Scope

### 4.1 Conclusion

The internship program has provided a robust framework for personal and professional growth. It effectively combined theoretical learning with practical application, enabling me to acquire essential technical skills in both front-end and back-end development, as well as database management. By engaging in the development of the Reside real estate platform, I was able to translate my classroom knowledge into real-world coding, problem-solving, and application-building, using technologies like .NET Core, PostgreSQL, and modern front-end frameworks. Moreover, the transition from structured technical training to a real-world project environment under Agile development practices equipped me with crucial collaborative skills. Tools such as Azure DevOps allowed me to experience firsthand the dynamics of team collaboration, sprint planning, and code version control. The internship also provided exposure to enterprise-level application development and introduced me to the complexities of working on live product teams. These experiences were invaluable in fostering a deeper understanding of both the software development lifecycle and industry best practices. In summary, the internship not only sharpened my technical expertise but also nurtured my ability to collaborate effectively within cross-functional teams, paving the way for a successful career in software development.

## **4.2 Future Scope**

I am pleased to share that the internship has resulted in an employment offer for the role of Associate Software Engineer with Casepoint, where I am currently working as part of the FOIA Team. This role marks the next step in my career journey, offering me the opportunity to continue developing my skills in a real-world environment. At Casepoint, my focus is on helping the organization streamline processes related to the Freedom of Information Act (FOIA), contributing to the development of robust, scalable, and efficient software solutions. The work I am doing with the FOIA team is providing me with a unique perspective on legal technology and compliance, areas that are increasingly critical in the tech industry. Looking ahead, my future goals include expanding my expertise in legal tech solutions and pursuing deeper knowledge in cloud computing, machine learning, and data analytics. These areas of growth align with the rapidly evolving technology landscape and will allow me to contribute to innovative projects that have real-world impact. As I continue my journey at Casepoint, I am excited to work on challenging projects, learn from experienced colleagues, and further refine my skills to meet the growing demands of the software development field. My experience from the internship has laid a strong foundation for this next phase of my career, and I am eager to make meaningful contributions to the team and organization.

# References

- [1] Introduction to JavaScript — MDN: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- [2] jQuery API Documentation: <https://api.jquery.com/>
- [3] PostgreSQL Documentation: <https://www.postgresql.org/docs/>
- [4] .NET Documentation — Microsoft: <https://learn.microsoft.com/en-us/dotnet/>
- [5] Kendo UI for jQuery: <https://www.telerik.com/kendo-jquery-ui>
- [6] Docs — Redis: <https://redis.io/docs/latest/>
- [7] RabbitMQ Documentation: <https://www.rabbitmq.com/docs>
- [8] Documentation — Elastic Search: <https://www.elastic.co/docs>
- [9] Agile Methodology — buzzybrains: <https://www.buzzybrains.com/blog/agile-methodology-for-software-development/>
- [10] What is Scrum Methodology — Institute of Project Management: <https://institute-projectmanagement.com/blog/understanding-scrum/>