

BASHABARI : A COMPLETE RENTAL SOLUTION FOR PROPERTY OWNERS & TENANTS

BY

**FAYSAL IBNEA HASAN JESAN
ID: 201-15-3681**

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Mr. Raja Tariqul Hasan Tusher
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Ms. Sharmin Akter
Lecturer (Senior Scale)
Department of Computer Science and Engineering
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

July 2024

APPROVAL

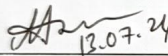
This Project titled “**BASHABARI : A COMPLETE RENTAL SOLUTION FOR PROPERTY OWNERS & TENANTS**”, submitted by Faysal Ibnea Hasan Jesan, ID: 201-15-3681 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 13-07-2024.

BOARD OF EXAMINERS



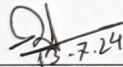
Dr. S.M Aminul Haque (SMAH)
Professor & Associate Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Board Chairman


13.07.24

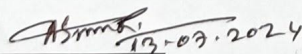
Ms. Nazmun Nessa Moon (NNM)
Associate Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner 1


13-7-24

Mr. Deawan Rakin Ahamed Remal (DRAR)
Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner 2


13-07-2024

Dr. Abu Sayed Md. Mostafizur Rahaman (ASMR)
Professor
Department of CSE
Jahangirnagar University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Raja Tariqul Hasan Tusher, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

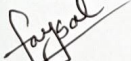


Mr. Raja Tariqul Hasan Tusher
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Ms. Sharmin Akter
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Submitted by:



Faysal Ibne Hasan Jesan
ID: 201-15-3681
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project successfully.

We are grateful and wish our profound indebtedness to **Mr. Raja Tariqul Hasan Tusher, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of web application department to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Sheak Rashed Haider Noori**, Head, Department of CSE, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

BashaBari is a comprehensive software solution designed to streamline the rental and property management process for both property owners and tenants. Managing properties, tenants, and their associated data has become increasingly challenging in today's fast-paced world. Simultaneously, finding a new place to live has turned into a time-consuming and often frustrating task. Despite the availability of numerous software options in the market, none offer a complete package that encompasses both maintenance and advertisement functionalities. BashaBari fills this gap by providing an all-in-one platform where property owners can effortlessly update and manage their properties. The software allows for efficient maintenance tracking, ensuring that all property-related issues are promptly addressed. Moreover, property owners can easily interact with their tenants, fostering a seamless communication channel that enhances tenant satisfaction and retention. On the other hand, tenants benefit from BashaBari by gaining access to a wide range of property listings that match their specific needs and budget. The software's intuitive search functionality enables tenants to find their desired properties quickly and conveniently. Additionally, tenants can communicate directly with property owners through the platform, making the process of renting a new home more straightforward and transparent. BashaBari stands out in the market by offering a holistic approach to property management and tenant services. It simplifies the complexities of property management while providing a user-friendly interface for tenants to find and secure their ideal living spaces. By bridging the gap between property owners and tenants, BashaBari ensures a more efficient and satisfying rental experience for all parties involved.

TABLE OF CONTENT

CONTENTS	PAGE
Board of Examiners	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
CHAPTER 1: INTRODUCTION	1-9
1.1 Overview	1-2
1.2 Background and Present State	2-3
1.3 Problem Statement	3
1.4 Objectives	3-4
1.5 Scope and Limitations	4-7
1.6 Report Organization	7-8
1.7 Summary	9
CHAPTER 2: LITERATURE REVIEW	10-15
2.1 Overview	10
2.2 Related Works	10-12
2.3 Comparison between existing works	12-13
2.4 Open Issues	14
2.5 Summary	15

CHAPTER 3: METHODOLOGY/ REQUIREMENT ANALYSIS AND DESIGN SPECIFICATION	16-35
3.1 Overview	16
3.2 Proposed Methodology/System Design	17-23
3.3 Hardware/ Software Requirement	23-32
3.4 Project Management and Financial Analysis	32-35
3.5 Summary	35
CHAPTER 4: IMPLEMENTATION	36-57
4.1 Overview	36
4.2 Implementation Requirements	37
4.3 Train Model/ Prototype Design	37-49
4.4 Implementation of Database	49-56
4.5 System Testing/ Model Evaluation	56-57
4.6 Summary	57
CHAPTER 5: RESULT AND ANALYSIS	58-62
5.1 Overview	58
5.2 Experimental/ Simulation Result	58-59
5.3 Performance/ Comparative Analysis	59-61
5.4 Summary	62
CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	63-66
6.1 Impact on Life	63

6.2 Impact on Society & Environment	63-64
6.3 Ethical Aspects	64-65
6.4 Sustainability Plan	65-66
6.5 Summary	66
CHAPTER 7: CONCLUSION AND FUTURE WORK	67-69
7.1 Conclusions	67
7.2 Further Suggested Works	67-68
7.3 Limitations/ Conflict of Interests	68-69
REFERENCES	70
APPENDIX A	71-75

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Waterfall Model	17
Figure 3.2: ER Diagram	18
Figure 3.3: Use Case Diagram	19
Figure 3.4: Laravel MVC Architecture	24
Figure 3.5: Laravel MVC Architecture	24
Figure 3.6: Laravel MVC Architecture	25
Figure 4.1: Home Page	37
Figure 4.2: Home Page	38
Figure 4.3: Home Page	38
Figure 4.4: Home Page	39
Figure 4.5: Contact Us	39
Figure 4.6: Create Search Agent	40
Figure 4.7: Add Listing	40
Figure 4.8: Owner Sign In	41
Figure 4.9: Tenant Sign In	41
Figure 4.10: Tenant Registration	42
Figure 4.11: Owner Home Page	42
Figure 4.12: Tenant List	43

Figure 4.13: Remove Tenant	43
Figure 4.14: Notice	44
Figure 4.15: Problem and Support	44
Figure 4.16: Profile	45
Figure 4.17: Create Property	45
Figure 4.18: Flat	46
Figure 4.19: Create Flat	46
Figure 4.20: Assign Tenant	47
Figure 4.21: Tenant Home Page	47
Figure 4.22: Search Property	48
Figure 4.23: Search Property	48
Figure 4.24: Problem and Support	49

LIST OF TABLES

FIGURES	PAGE NO
Table 1.1: Specific usage for property owners and tenants	3
Table 2.1: Comparative analysis with previous work	12
Table 2.2: Open issue	14
Table 3.1: Use case analysis	20
Table 3.2: System requirements of LARAVEL	27
Table 3.3: System requirements of XAMPP	28
Table 3.4: System requirements of POSTMAN	30
Table 3.5: System requirements of VSCODE	32
Table 3.6: Project costs	35
Table 4.1: Implementation Requirements	37
Table 4.2: Admin	49
Table 4.3: Owner	50
Table 4.4: Tenant	50
Table 4.5: Building	51
Table 4.6: Flat	51
Table 4.7: Rent	52
Table 4.8: Create Search Agent	53

Table 4.9: Notice	53
Table 4.10: Problem	54
Table 4.11: Supervisor	54
Table 4.12: Electrician	55
Table 4.13: Plumber	55
Table 4.14: Support	55
Table 5.1: Test cases and result analysis	58

CHAPTER 1

INTRODUCTION

1.1 Overview

In the dynamic realm of real estate, the key to success lies in seamless property management and rental systems. This paper delves into the transformative landscape of property management, unveiling a cutting-edge software solution poised to redefine industry standards. As traditional methods grapple with inefficiencies and challenges, our research explores how this software not only streamlines operations but also catalyzes a paradigm shift in the way we perceive and manage properties.

Embarking on an exploration of the historical backdrop, we unravel the intricacies that have traditionally plagued property management. From cumbersome paperwork to communication bottlenecks, the limitations of conventional approaches become apparent. Against this backdrop, our paper introduces an innovative software system, a technological marvel designed to transcend these barriers and elevate property management to unprecedented heights.

Through a meticulous analysis of the software's core features, we investigate how it empowers property managers, landlords, and tenants alike. Dynamic tools for tenant screening, lease management, and financial tracking promise to usher in an era of unprecedented efficiency. Furthermore, the integration of artificial intelligence and machine learning algorithms anticipates and mitigates potential issues, enhancing foresight and decision-making.

The impact of this software is not confined to operational enhancements alone. We delve into the broader implications on the user experience, examining how the system fosters transparent communication, fosters tenant satisfaction, and ultimately cultivates a thriving property ecosystem.

In a world constantly evolving, our paper aims to be a beacon illuminating the transformative potential of this software in the landscape of property management. By unearthing its capabilities, we seek to inspire industry stakeholders to embrace innovation, fostering a future where property management is not just a function but an art perfected by the seamless integration of technology and human ingenuity.

1.2 Background and Present State

In the real estate world, the management of properties and rentals remains entrenched in outdated practices. The industry is plagued by excessive paperwork, inefficient communication methods, and overall inefficiencies that hinder smooth operations. Property managers often find themselves juggling a myriad of manual tasks, from tenant screening to lease management, which frequently results in errors and oversights. This traditional approach not only consumes valuable time but also increases the likelihood of mistakes.

For property owners, the lack of real-time updates on the financial performance of their properties poses a significant challenge. Without timely and accurate data, making informed decisions becomes difficult, potentially impacting the profitability and efficiency of their investments. Furthermore, tenants face their own set of challenges within this antiquated system. Communication with property managers and owners is often cumbersome, and the tools available to them are not always user-friendly, complicating their rental experience.

BashaBari is dedicated to addressing these longstanding issues by introducing innovative software solutions designed to streamline and modernize the property management process. Our mission is to understand the shortcomings of the traditional methods thoroughly so that we can implement smarter, more efficient solutions. These advancements are not just about making operations smoother; they aim to transform the entire experience for property managers, landlords, and tenants.

By leveraging cutting-edge technology, BashaBari seeks to eliminate the inefficiencies that have long plagued the real estate industry. Our solutions provide property managers

with automated tools to handle tasks more effectively, reducing the risk of errors. Property owners gain access to real-time financial insights, empowering them to make better-informed decisions. For tenants, the platform offers a more user-friendly interface and streamlined communication channels, enhancing their overall rental experience.

In essence, BashaBari is poised to revolutionize the property management landscape, making it simpler, more efficient, and more enjoyable for everyone involved. Through our innovative approach, we aim to set a new standard in the industry, fostering a more connected and productive environment for property management.

1.3 Problem Statement

In the real estate world, the way we manage properties and rentals is kind of stuck in the past. There's too much paperwork, and outdated ways of talking to each other, and things aren't as smooth as they could be.

For property managers, it's like juggling a bunch of manual tasks – from checking out tenants to handling leases and it often leads to mistakes. Property owners don't get real-time updates on how their properties are doing financially, making it tough to make smart decisions. And for tenants, it's not always easy to communicate, plus the tools they use aren't always user-friendly.

BashaBari is all about fixing these issues. We want to understand the problems with the old ways, so we can introduce new, smarter software solutions. These solutions won't just make things run smoother; they'll also change how property managers, landlords, and tenants experience the whole property management scene, making it simpler and better for everyone involved.

1.4 Objectives

The main objectives of the project are:

Table 1.1: Specific usage for property owners and tenants

Owners	Tenants
--------	---------

1. Can create an account profile, update profile	1. Can create an account profile, update profile
2. Can create properties, update property details, delete properties	2. Can get property details, rent details, contact details
3. Can view the tenant's details	3. Can view the owner's details
4. Can add tenants to the property	4. Can view the assigned details
5. Can create notice, update notice, and delete notice	5. Can view notices
6. Can view the problems of the tenants	6. Can create problems and update their status
7. Can assign plumbers and electricians to the properties	7. Can contact the assigned plumbers and electricians
8. Can update the availability status of the properties	8. Can search and find the properties
9. Can add listing	9. Can create a search agent

1.5 Scope and Limitations

The project aims to develop an advanced Property Management and Rental System Software designed to address the inefficiencies and challenges prevalent in traditional real estate management. The software will be a comprehensive solution catering to property managers, landlords, and tenants, fostering a seamless and user-friendly experience.

The project scope of 'BashaBari' can be explored with the following aspects.

Objectives:

- Streamline property management processes, including managing tenants and their details.
- Enhance communication channels among property managers, landlords, and tenants.
- Provide real-time insights for landlords into property performance and financial metrics.

- Establish an efficient process for tenants to find their desired properties and packages
- Develop a user-centric interface for easy navigation and enhanced user experience.

Key Features:

- Tenant Management: Implement an efficient tenant management process to minimize the time complexity and hassle.
- Rent Management: Develop a robust system for managing rent, updating rent status, and ensuring quality performance.
- Communication Platform: Establish a user-friendly communication platform for property managers, landlords, and tenants to enhance interaction and address concerns promptly.
- Advertisement: Integrate an advertising system for tenants to find properties and observe the details in one place.

User Roles:

- Property Managers: Access to management system on behalf of property owners and control the system seamlessly.
- Property Owners: Real-time insights into property performance, details, availability status, manipulate properties, and seamless communication with tenants.
- Tenants: User-friendly interface for searching properties, streamlined communication, rent payments, and issue reporting.

Technology Stacks:

- Frontend: For the front-end development, we'll use HTML, CSS, and JavaScript. Additionally, we'll employ Vue.js and Tailwind CSS as frameworks and libraries to enhance our project.
- Design: In the context of user interface and user experience design, we have chosen to utilize the FIGMA platform.

- Backend: Secure and scalable backend architecture requires our selection for back-end development involves the use of the PHP Laravel framework.
- Database: For the database management system, we have chosen MySQL to meet our project's data storage and retrieval requirements.
- Integration: We have designated POSTMAN for the creation and testing of APIs, underscoring its pivotal role in ensuring seamless functionality.

Implementation Timeline:

- Phase 1 (3 months): Planning, research, and system design.
- Phase 2 (6 months): Software development with a focus on core features.
- Phase 3 (2 months): Testing, debugging, and user acceptance.
- Phase 4 (1 month): Deployment and initial user training.
- Phase 5 (Ongoing): Post-deployment support, updates, and feature enhancements.

Budget:

The budget for the project ranges from 17,825 to 24,150 BDT. This includes costs for Domain and Hosting, which are estimated between 5000 and 8000 BDT, and for meetings with stakeholders such as owners, tenants, and managers, which are projected to cost between 2500 and 3000 BDT. Testing and SQA are anticipated to require 8000 to 10000 BDT. Additionally, a contingency fund, set at 15% of the total costs, is calculated to be between 2325 and 3150 BDT to cover any unforeseen expenses.

Resources:

This aspect looks at what we need to build the system. For this system, we only need basic hardware and a compatible operating system, keeping things simple. Checking these requirements helps us decide if the project is doable. Considering everything about this system, we can confidently say it's practical to move forward with the project.

Evaluation:

- Regular progress assessments through milestones and deliverables.
- User feedback and testing throughout the development phases.
- Post-implementation evaluation to measure the system's effectiveness and user satisfaction.

Risks & Mitigation:

- Identify potential risks related to technology, user adoption, and external factors.
- Develop mitigation strategies to address and minimize risks during the project lifecycle.

The project scope outlines a comprehensive approach to developing an innovative Property Management and Rental System Software. By addressing key challenges in traditional systems, the software aims to revolutionize the real estate industry, fostering efficiency, transparency, and a superior user experience for property managers, landlords, and tenants.

1.6 Report Organization

The report organization provides a concise summary of all the chapters included in the document.

- **Chapter 1:** The overview of BashaBari begins with its motivation and objectives, emphasizing the driving factors behind its development. The project scope defines the boundaries and focus areas, while the projected outcome outlines the expected deliverables and benefits. The report organization details how information is structured to provide clarity and coherence in presenting the project's findings and achievements.
- **Chapter 2:** The literature review of BashaBari begins with an exploration of relevant academic and industry literature. This is followed by a review of related works, providing insights into existing solutions. A comparative analysis of these works

highlights their strengths and limitations. Finally, a gap analysis identifies areas within the context of BashaBari that require further research or development.

- **Chapter 3:** The methodology of BashaBari starts with rigorous requirement analysis, followed by the development of the proposed methodology and system design. Data collection and input-output analysis ensure robust data handling. Project management oversees the implementation process, concluding with a comprehensive financial analysis.
- **Chapter 4:** The implementation of BashaBari begins with an assessment of implementation requirements, followed by the development of a prototype design. Subsequently, the database implementation phase ensures robust data management. Finally, comprehensive system testing concludes the implementation process, ensuring functionality and reliability.
- **Chapter 5:** The result and analysis of BashaBari commence with the presentation of experimental or simulation results, followed by a detailed performance and comparative analysis.
- **Chapter 6:** The impact of BashaBari on society, the environment, and sustainability begins with an assessment of its impact on life. This is followed by an analysis of its effects on society and the environment, addressing both positive and negative aspects. Ethical considerations are then explored to ensure responsible implementation. Finally, a sustainability plan outlines strategies for long-term viability and minimal environmental footprint.
- **Chapter 7:** The conclusion and future work of BashaBari begins with summarizing the key conclusions drawn from the project. This is followed by suggestions for further research and development to enhance and expand the project's scope. Finally, a discussion of limitations and potential conflicts of interest provides transparency and identifies areas for improvement.

1.7 Summary

Our project seeks to revolutionize property management and rentals in Bangladesh through the development of an advanced software solution. Motivated by the industry's need for innovation, we aim to bridge the gap between traditional practices and modern efficiency. The scope of our project includes creating a centralized platform for property owners to manage tenants, properties, and rent collection seamlessly. Simultaneously, tenants will benefit from a user-friendly interface, minimizing the hassle of finding TO-LET properties without physical interactions. The anticipated outcomes encompass streamlined operations, transparent communication, informed decision-making, enhanced user experiences, and the integration of advanced technologies. The project aspires to simplify rent collection, provide detailed rent summaries, and transform TO-LET advertising. In summary, our project is a pivotal step towards simplifying property management in Bangladesh, showcasing how technology can bring efficiency, transparency, and user satisfaction to the forefront of real estate practices. This initiative aims to set new standards for the industry, making property management not just a task but an intuitive and streamlined experience for property owners and tenants alike.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Few studies have delved into effective Property and Rent Management Systems in Bangladesh, leaving a notable gap in research for this specific context. Various software initiatives are currently engaged in the research and development of Property and Rent Management Systems tailored to the Bangladeshi context. VOBON and Bproperty are actively involved in exploring innovative solutions for property management, with a focus on the intricacies of the rental landscape in Bangladesh. While VOBON is positioned as a software solution, Bproperty operates as a comprehensive platform catering to both property sales and rentals. This research underscores the limited scope of existing studies within the Bangladeshi context and emphasizes the relevance of ongoing projects, such as VOBON and Bproperty, in contributing to the advancement of effective Property and Rent Management Systems in this region.

2.2 Related Works

In this section, we talked about the latest projects and other works happening in this area.

Bproperty:

Bproperty (www.bproperty.com) is a leading online real estate platform in Bangladesh, connecting property seekers with property owners, developers, and agents. Founded in 2008, it has grown to become the country's most popular real estate portal, with over 11,000 properties listed on the platform, offering a wide range of listings for properties across all categories, including[8]:

- Apartments: From studios to penthouses, Bproperty features apartments in various sizes, budgets, and locations across Bangladesh.

- Houses: Whether you're looking for a single-family home, a townhouse, or a villa, Bproperty has a diverse selection of houses to suit your needs.
- Plots: If you're planning to build your dream home, Bproperty offers a variety of plots in different sizes and locations.
- Commercial properties: Find offices, shops, warehouses, and other commercial properties for rent or sale on Bproperty.

VOBON:

VOBON (www.vobon.xyz), a rising star in Bangladesh's proptech scene, offers a comprehensive cloud-based platform for efficient building management. Founded in 2018, it empowers property owners, residents, and management teams with a suite of tools to simplify operations, improve communication, and optimize costs. These are some key features of VOBON[9]:

- Tenant Management: Rent collection, lease agreements, background checks, communication channels, and tenant portals.
- Financial Management: Billing, accounting, budgeting, payment gateways, and financial reports.
- Communication & Community: Resident/tenant announcements, surveys, and event management.
- Security & Access Control: CCTV monitoring, and visitor management.

Rentola:

Rentola (www.rentola.com) is a prominent online real estate platform operating in various countries, including Bangladesh, the Netherlands, France, Spain, and the United Kingdom. Established in 2008, it primarily focuses on facilitating rental transactions between tenants and landlords/property owners. Some key features and services of Rentola[10]:

- **Extensive Listings:** Rentola boasts a vast database of rental properties across different categories, including apartments, houses, rooms, and commercial spaces.
- **User-Friendly Interface:** The platform offers a user-friendly website and mobile app, allowing for easy navigation and property search through various filters like location, price range, size, and amenities.
- **Detailed Listings:** Each listing features comprehensive information like photos, floor plans, descriptions, and often virtual tours, providing potential tenants with a clear picture of the property.
- **Direct Communication:** Rentola facilitates direct communication between tenants and landlords/property owners, streamlining the inquiry and negotiation process.

2.3 Comparison between existing works

Comparison of existing works:

Table 2.1. Comparative analysis with previous work

SL No	Feature	Bproperty	VOBON	Rentola
1.	Primary Focus	Real estate listings (sale & rent).	Property management software	Rental listings
2.	Target Audience	Property seekers, owners, & agents	Building owners/managers & residents	Tenants & landlords/owners
3.	Key Services	Property search & listing, valuation,	Maintenance & work order management,	Finding rental properties, direct

		legal advice, relocation	tenant management, financial tools, communication & community features	communication with landlords/owners
4.	Strengths	Extensive listings, user-friendly interface, advanced search, diverse property types	Streamlines operations, reduces costs, improves resident/tenant experience	Large pool of rental properties, direct communication, user-friendly interface
5.	Weaknesses	Limited value-added services for property management, primarily focused on transactions	Requires adoption by building owners/managers, not directly relevant to individual tenants	Smaller market share in Bangladesh compared to some local competitors
6.	Market Position	Leading real estate portal in Bangladesh	Growing player in Bangladeshi property management software	Established an international platform with a smaller presence in Bangladesh
7.	Potential for Research	Impact on market transparency & trends, comparison with other listing platforms, analysis of user behaviour	Impact on Bangladeshi property management landscape, comparison with other software, analysis of user adoption	Comparison with other online rental platforms in Bangladesh, evaluation of challenges & opportunities in Bangladeshi market

2.4 Open Issue

Table 2.2. Open issue

Aspect	Established PRMS	BashaBari
Geographical Inclusivity	Primarily urban-focused.	Inclusive of both urban and rural areas.
User Interface (UI)	Often lacks intuitiveness, visually unappealing.	Prioritizes user-centric design with intuitive and engaging UI.
User Experience (UX)	Disjointed user experience, and compromised software performance.	Emphasis on seamless and accessible UX for non-technical users.
Target Audience	Predominantly technical users.	Designed for non-technical and regular users.
Functionalities	Valuable functionalities, but limited addressing rural property management needs.	Comprehensive features addressing both urban and rural property and rent management.
Adoption in Third-World Nations	Limited adoption, especially in countries like Bangladesh.	Aims to enhance adoption and effectiveness within the unique context of Bangladesh.
Overall Effectiveness	Varied, with room for substantial upgrades	Aims for elevated effectiveness and user satisfaction within the Bangladeshi context

2.5 Summary

This chapter provides a comprehensive overview of existing Property and Rent Management Systems (PRMS) through an analysis of various relevant scholarly works. This exploration drives into the functionalities, strengths, and limitations of these systems, laying the groundwork for further investigation and potential advancements in the field.

CHAPTER 3

METHODOLOGY/ REQUIREMENT ANALYSIS AND DESIGN SPECIFICATION

3.1 Overview

This project utilizes a carefully chosen technology stack to construct a comprehensive Property and Rent Management System. The front-end architecture leverages the established trio of HTML, CSS, and JavaScript for core structure and interactivity. This foundation is augmented by the Vue.js framework, which facilitates rapid development and component-based architecture[3]. Additionally, the Bootstrap library is employed to ensure responsive design and consistent presentation across diverse devices. The back-end utilizes the Laravel framework, a robust PHP platform offering a well-structured object-oriented foundation and MVC architecture. jQuery and AJAX libraries further enhance the back-end by enabling dynamic data retrieval and asynchronous communication for a more responsive user experience. API development and testing are rigorously conducted using Postman, guaranteeing reliable communication between the front-end and back-end components. Rounding out the project's infrastructure are Node.js and NPM for package management, PHP and Composer for server-side scripting and dependency management, and VS Code as the primary development environment.

3.2 Proposed Methodology/System Design

Development Model

As a development method, we will be using the Waterfall method. It is a proper software development life cycle (SDLC) and the efficiency of this model is better than Agile or other established methodologies.

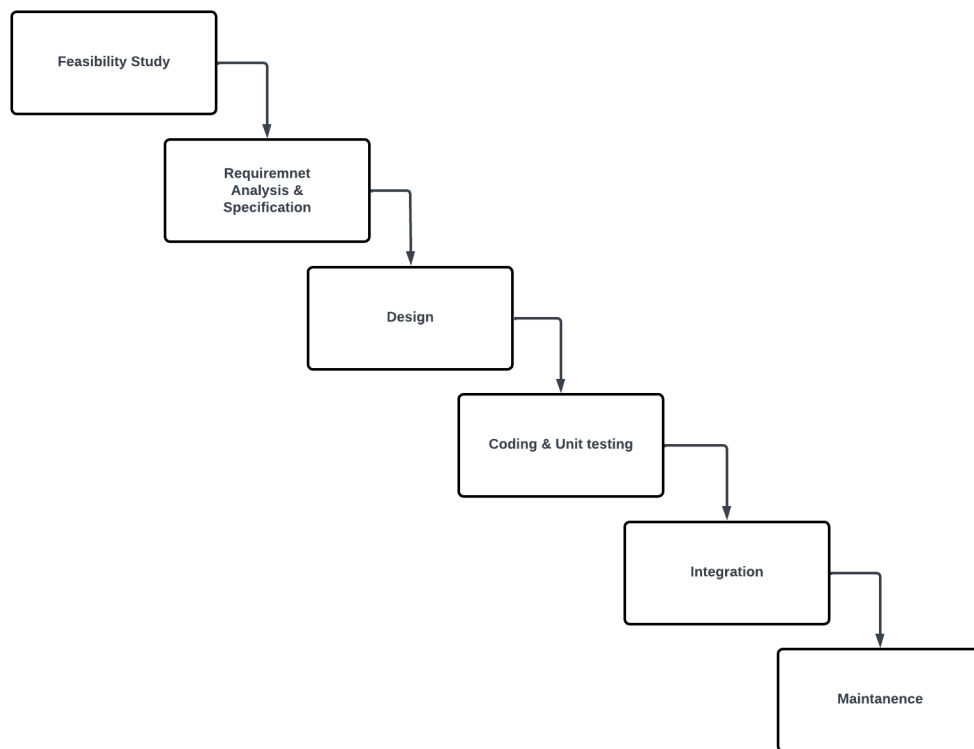


Figure 3.1: Waterfall Model

Use Case Diagram

Use Case diagram of BashaBari:

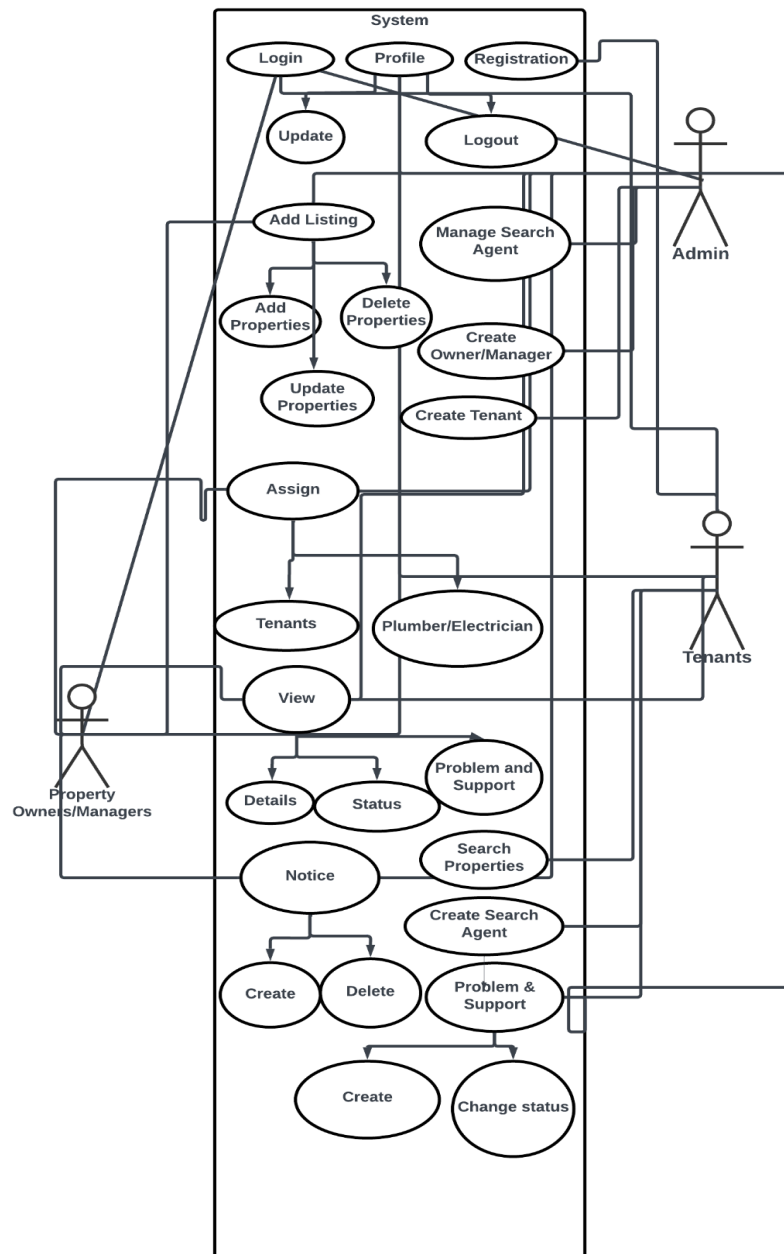


Figure 3.3: Use Case Diagram

Use Case Analysis

Table 3.1: Use case analysis

Use Case	Actors	Description	Steps
Tenant Registration	Tenant	Register a new tenant	1. Access the registration page. 2. Enter personal details and contact information. 3. Submit the registration form.
Log In	Property Owner/Manager, Tenant, Admin	Actors can log in to the system using valid credentials.	1. Access the login page. 2. Enter username and password. 3. Click the "Log In" button.
Create Property Owner/Manager	Admin	Admin can create new property owners/managers by entering their details, including personal information and login credentials.	1. Access the admin dashboard. 2. Navigate to the "Create User" section. 3. Enter owner/manager details and login credentials. 4. Submit the form to create a new property owner/manager.
Create Tenant	Admin	Admin can create new tenants by entering their details, including personal information and login credentials.	1. Access the admin dashboard. 2. Navigate to the "Create User" section. 3. Enter tenant details and login credentials. 4. Submit the form

			to create a new tenant.
Add Property Listing	Admin, Property Owner/Manager, Tenant, Admin	Admin and Property owners/managers can add new property listings, including details like property type, location, amenities, and rental information.	1. Navigate to the "Add Listing" section. 2. Enter property details, amenities, and rental information. 3. Submit the form to add a new property listing.
Update Property Listing	Admin, Property Owner/Manager, Tenant, Admin	Admin and Property owners/managers can delete property listings that are no longer relevant or available for rent.	1. Select the listing to be updated. 2. Modify the necessary details.
Delete Property Listing	Admin, Property Owner/Manager, Tenant, Admin	Admin and Property owners/managers can delete property listings that are no longer relevant or available for rent.	1. Select the listing to be deleted. 2. Confirm the deletion.
Update Profile	Property Owner/Manager, Tenant	Property owners/managers and tenants can update their profiles.	1. Log in to the system. 2. Navigate to the "Profile Settings" or "Edit Profile" section. 3. Modify the necessary personal and contact details. 4. Save the changes.
Logout	Property Owner/Manager, Tenant, Admin	Actors can log out of the system.	1. Locate the "Logout" or "Sign Out" option in the user interface. 2. Click on the "Logout" or "Sign Out" option.

Assign Tenants	Property Owner/Manager	Property owners/managers can assign tenants to their properties.	1. Log in to the system. 2. Navigate to the "Manage Properties" or "Assign Tenants" section. 3. Select the property. 4. Choose the tenant to be assigned to the property.
Assign Electricians/Plumbers	Admin, Property Owner/Manager	Admin and Property owners/managers can assign electricians/plumbers for property maintenance.	1. Log in to the system. 2. Select the property requiring maintenance. 3. Choose the electrician/plumber for the assignment.
Delete Notice	Property Owner/Manager, Admin	Property owners/managers and admin can delete notices.	1. Log in to the system. 2. Navigate to the "Notices" or "Manage Notices" section. 3. Select the notice to be deleted. 4. Confirm the deletion.
Create Search Agent	Tenant	Tenants can create search agents to receive notifications for matching properties.	1. Log in to the system. 2. Navigate to the "Search Agents" or "Create Search Agent" section. 3. Define search criteria, such as location, property type, and budget. 4. Save the search agent.

Search Properties	Tenant	Tenants can search for available properties based on their preferences.	1. Log in to the system. 2. Access the "Search Properties" section. 3. Enter search criteria, such as location, property type, and budget. 4. View the search results.
-------------------	--------	---	---

3.3 Hardware/Software Requirement

LARAVEL

Laravel is a high-level, open-source PHP framework designed to streamline web application development[1]. It offers a robust set of features and tools that promote rapid development, code maintainability, and scalability. Key aspects that make it a compelling choice for our project include:

1. Model-View-Controller (MVC) Architecture:

- Models: Represent data entities and interact with the database.

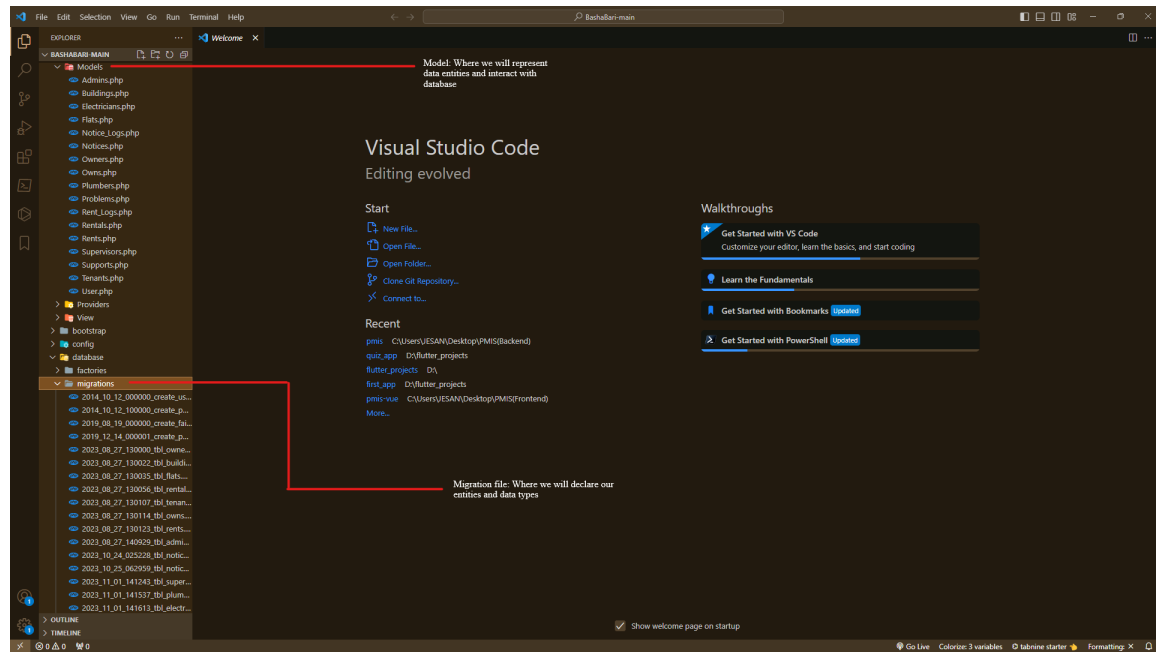


Figure 3.4: Laravel MVC Architecture

- Views: Handle user interface elements and presentation logic.

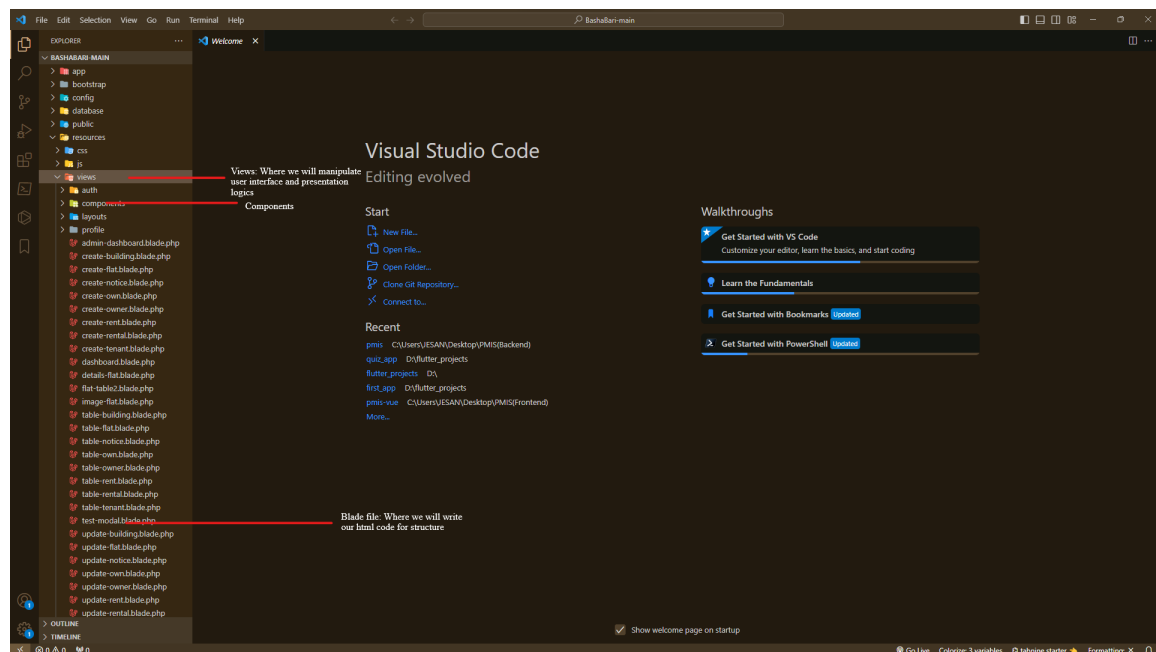


Figure 3.5: Laravel MVC Architecture

- Controllers: Mediate user requests, manage data flow, and invoke appropriate models and views.

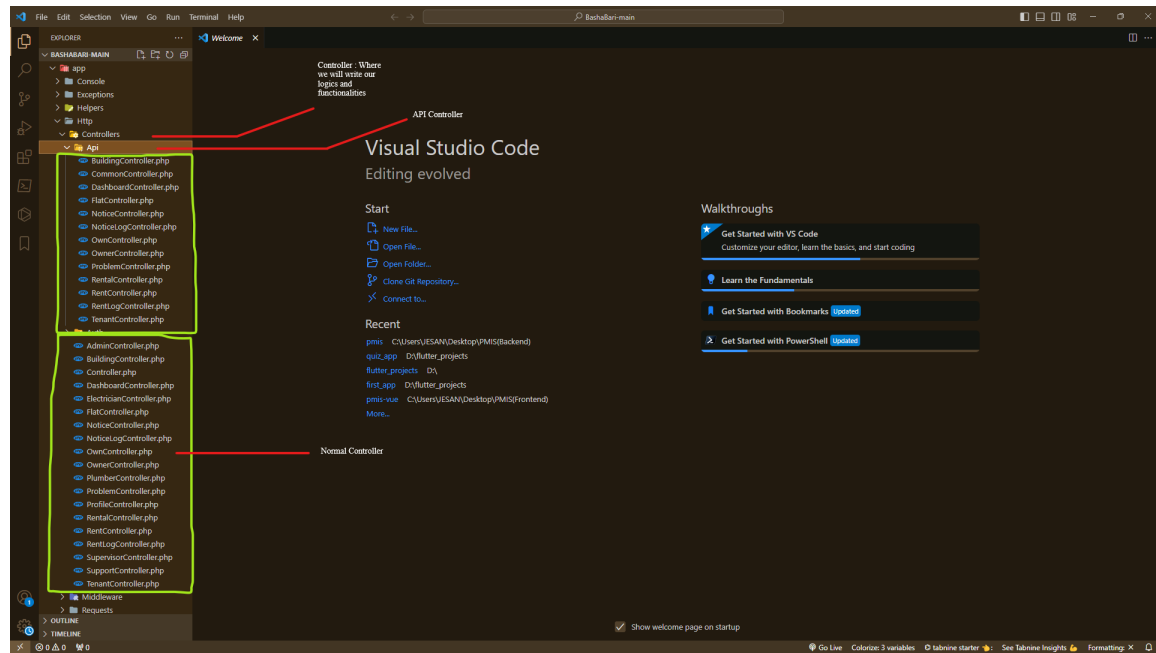


Figure 3.6: Laravel MVC Architecture

2. Object-Oriented Programming (OOP) Paradigm:

- Laravel fully embraces OOP principles, leading to modular, extensible, and well-structured code[7].
- It employs classes, objects, interfaces, and other OOP constructs to model real-world entities and interactions effectively[7].

3. Eloquent ORM:

- Laravel provides Eloquent, a powerful ActiveRecord implementation for interacting with databases[7].
- It simplifies database queries and relationships, enabling developers to work with database objects using intuitive object-oriented syntax[7].

4. Artisan Command-Line Interface (CLI):

- Laravel includes Artisan, a versatile CLI tool that automates common tasks and streamlines development workflows[7].

- It offers commands for generating code, managing database migrations, running tests, and performing various maintenance tasks[7].
5. Routing System:
- Laravel's expressive routing system allows for effortless definition of application routes that map user requests to specific controller actions[1].
 - It supports route parameters, middleware, and various HTTP methods for flexible and dynamic routing configuration[1].
6. Blade Templating Engine:
- Laravel utilizes Blade, a lightweight templating engine that facilitates the creation of clean and reusable HTML layouts[1].
 - Blade offers features like conditional statements, loops, and inheritance, enabling dynamic content generation within views[1].
7. Authentication and Authorization:
- Laravel provides built-in authentication and authorization features to secure your application[1].
 - It simplifies user registration, login, password resets, and role-based access control, ensuring data integrity and protecting sensitive information[1].
8. Security:
- Laravel prioritizes security, incorporating measures to prevent common vulnerabilities like cross-site scripting (XSS) and SQL injection.
 - It employs features like hashed passwords, prepared SQL statements, and input validation to safeguard applications.
9. Package Ecosystem:
- Laravel boasts a vibrant community and a vast ecosystem of packages and extensions[6].
 - These packages enrich functionality and expedite development in areas such as payment gateways, user management, file handling, and more[6].
10. Testing:
- Laravel encourages test-driven development, offering built-in support for unit and integration testing.

- This ensures code quality, reliability, and facilitates future maintenance and refactoring efforts.

System Requirements

System requirements of LARAVEL:

Table 3.2: System requirements of LARAVEL

Criterion	Minimum	Recommended
PHP Version	PHP 7.3.0 or higher	PHP 7.4.0 or higher
Web Server	Apache or Nginx	Apache or Nginx
Database	MySQL, PostgreSQL, SQLite, or SQL Server	MySQL 5.7.8 or PostgreSQL 9.6 or higher
Composer	Latest version of Composer	Latest version of Composer
RAM	2 GB RAM	4 GB RAM or more
Processor	1.8 GHz dual-core processor	2.5 GHz quad-core processor or higher
Storage	1 GB free hard disk space	10 GB free hard disk space or more
Operating System	Windows, Linux, macOS	Windows 10, Linux (Ubuntu, CentOS), macOS
Node.js and npm	Required for Laravel Mix (optional)	Required for Laravel Mix (optional)
Composer Dependencies	Extension modules required by Laravel	Extension modules recommended by Laravel

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It simplifies the process of setting up a local web development environment by bundling together several software components[4]. The name "XAMPP" is an acronym representing the included software components:

1. X (Cross-platform): XAMPP is designed to be cross-platform, meaning it can run on different operating systems, including Windows, Linux, and macOS.
2. A (Apache HTTP Server): Apache is a widely used open-source web server software that plays a central role in the XAMPP stack. It processes and serves web requests, handling tasks such as request parsing, file serving, and managing security features.
3. M (MySQL): MySQL is a popular open-source relational database management system (RDBMS). XAMPP includes MySQL to provide a database server for storing and retrieving data in web applications.
4. P (PHP): PHP is a server-side scripting language used for developing dynamic web pages. XAMPP includes PHP to enable the server to execute PHP scripts, which are embedded in HTML and run on the server rather than the client's browser.
5. P (Perl): XAMPP originally stood for "Personal Home Page" (PHP), but it was later expanded to include "Perl" as well. Perl is a high-level, general-purpose programming language often used for web development, text processing, and system administration.

In summary, XAMPP simplifies the setup of a local web development environment by bundling Apache, MySQL, PHP, Perl, and other tools. It is a convenient solution for developers to quickly create and test web applications without the need for complex manual installations.

System Requirements

System requirements of XAMPP:

Table 3.3: System requirements of XAMPP

Criterion	Minimum	Recommended
Operating System	Windows, Linux, macOS	Windows 10, Linux (Ubuntu, CentOS), macOS
Processor	1.5 GHz dual-core processor	2.5 GHz quad-core processor or higher

RAM	1 GB RAM	4 GB RAM or more
Storage	500 MB free hard disk space	10 GB free hard disk space or more
Web Browser	Any modern web browser	The latest version of Chrome, Firefox, or Edge

POST-MAN

Postman is a popular collaboration platform for API development, testing, and management. It provides a user-friendly interface for creating, sharing, and testing APIs. Postman simplifies the development process by allowing users to design, test, and document APIs, making it an essential tool for developers, QA engineers, and other stakeholders involved in the API lifecycle[5]. What are the purposes of POSTMAN?:

1. **API Design:** Postman offers a visual API design environment where users can create API requests, define endpoints, and structure API data. This helps in designing APIs before implementation.
2. **API Testing:** Postman allows users to create and execute automated tests for APIs. Test scripts can be written using JavaScript, and the platform supports a variety of testing scenarios, including unit testing, integration testing, and performance testing.
3. **Collaboration:** Teams can use Postman to collaborate on API development. Collections, which are groups of API requests, can be shared, versioned, and synchronized across team members.
4. **Monitoring:** Postman provides monitoring capabilities to track the performance of APIs in real-time. This includes response times, error rates, and other key metrics.
5. **Documentation:** Postman automatically generates API documentation based on the requests and responses defined in the platform. This documentation can be shared with stakeholders for better understanding and integration.
6. **Mock Servers:** Users can create mock servers in Postman to simulate the behavior of an API even before it is implemented. This is useful for front-end developers who need to interact with an API that is still in development.

7. Automation: Postman supports automation through the Postman API, allowing integration with CI/CD pipelines and other automation tools. This ensures consistent testing and monitoring throughout the development lifecycle.
8. Environments: Postman allows users to define different environments (e.g., development, staging, production) with variables, making it easy to switch between different configurations for testing.
9. Security: Postman provides features to secure API requests, including support for OAuth, API key management, and encryption.

Postman is a comprehensive platform for API development and testing, offering a user-friendly interface along with powerful features for collaboration, automation, and monitoring throughout the API lifecycle.

System Requirements

System requirements of POSTMAN:

Table 3.4: System requirements of POSTMAN

Criterion	Minimum	Recommended
Operating System	Windows, Linux, macOS	Windows 10, Linux (Ubuntu, CentOS), macOS
Processor	1.5 GHz dual-core processor	2.5 GHz quad-core processor or higher
RAM	2 GB RAM	4 GB RAM or more
Storage	500 MB free hard disk space	1 GB free hard disk space or more
Network	Internet connection for API testing	Internet connection for API testing

VSCODE

Visual Studio Code (VSCode) is an advanced and feature-rich source code editor developed by Microsoft, celebrated for its versatility and widespread adoption within the

developer community. With its foundation built on a lightweight and fast architecture, VSCode ensures quick startup times and smooth performance across different operating systems, including Windows, macOS, and Linux. The editor is designed with an emphasis on intelligent code editing, incorporating features such as syntax highlighting, code completion, and robust navigation tools to enhance developers' productivity and code comprehension[2].

One of VSCode's notable strengths lies in its extensibility. The editor supports a vast ecosystem of extensions available through the Visual Studio Code Marketplace. This extensibility allows developers to customize their coding environment with language support, debugging tools, themes, and various integrations, tailoring the editor to meet specific project requirements.

Integrated Git support is another hallmark feature of VSCode. The editor seamlessly integrates with Git, providing version control features directly within the user interface. This includes Git status indicators in the file explorer, commit histories, and easy access to Git commands, facilitating a streamlined version control workflow.

Beyond basic code editing and version control, VSCode excels in its debugging capabilities. Developers can set breakpoints, inspect variables, and step through code with an intuitive and efficient debugging interface. The inclusion of an integrated terminal further enhances the development experience, allowing users to run command-line tools and scripts without leaving the editor[2].

The editor's user interface is highly customizable, offering flexibility in adjusting themes, icons, and layouts to suit individual preferences. With Live Share, VSCode facilitates real-time collaborative coding sessions, allowing multiple developers to work together on the same codebase, irrespective of physical location.

In conclusion, Visual Studio Code stands out as a premier code editor, seamlessly blending speed, intelligent features, and a comprehensive toolset. Its unwavering dedication to adaptability, collaboration, and cross-platform functionality positions it as an essential tool for developers involved in a wide spectrum of software development

projects. With its versatile ecosystem of extensions, integrated Git support, and innovative features like Live Share and Remote Development, Visual Studio Code continues to be the preferred choice for developers seeking a powerful, customizable, and efficient coding environment.

System Requirements

System requirements of VSCODE:

Table 3.5: System requirements of VSCODE

Criterion	Minimum	Recommended
Operating System	Windows 7 (64-bit), macOS 10.10+, Linux	Windows 10 (64-bit), macOS 10.14+, Linux
Processor	1.6 GHz or faster	2.0 GHz dual-core processor or higher
RAM	1 GB	4 GB or more
Storage	200 MB free hard disk space	500 MB free hard disk space or more
Graphics	Integrated graphics with OpenGL 3.3 or higher	Integrated or discrete GPU with OpenGL 3.3 or higher

3.4 Project Management and Financial Analysis

3.4.1 Project Management

The project aims to develop an advanced Property Management and Rental System Software designed to address the inefficiencies and challenges prevalent in traditional real estate management. The software will be a comprehensive solution catering to property managers, landlords, and tenants, fostering a seamless and user-friendly experience.

Objectives:

- Streamline property management processes, including managing tenants and their details.
- Enhance communication channels among property managers, landlords, and tenants.
- Provide real-time insights for landlords into property performance and financial metrics.

Implementation Timeline:

- Phase 1 (3 months): Planning, research, and system design.
- Phase 2 (6 months): Software development with a focus on core features.
- Phase 3 (2 months): Testing, debugging, and user acceptance.
- Phase 4 (1 month): Deployment and initial user training.
- Phase 5 (Ongoing): Post-deployment support, updates, and feature enhancements.

Resources:**A. Equipments and Tools:**

- For the code environment, we will use VS CODE.
- For version control, we will use Git.
- For testing purposes, we will use POST-MAN.

B. Software and Technologies:

- For front-end development, we will use HTML, CSS, Java-Script, Vue.Js, Bootstrap, Tailwind, etc.
- For back-end development, we will use Laravel and PHP.
- For Database Management System we will use MySQL.
- For UI/UX design we will use Figma.
- For API integration we will use POST-MAN.

C. Data and Content:

- Property Data: Provided by property owners and managers.

- Tenant Data: Through user registration.
- Other Data: Gathered based on property owner's and manager's information.

Risks & Mitigation:

- Identify potential risks related to technology, user adoption, and external factors.
- Develop mitigation strategies to address and minimize risks during the project lifecycle.
- Dealing with data security of the stakeholders and concern about privacy.
- Evaluate the market situation and deal with the competitors.

Communication Plan:

A. Meeting with Stakeholders:

- Purpose: Discussing the new aspects with the property owners and managers and taking feedback from the tenants.
- Participants: Team members, property owners/managers, tenants.
- Frequency: Based on the need for information but at least 1-2 times in 3 months.

B. Change Control Meetings:

- Purpose: Addressing the new changes and taking feedback from stakeholders.
- Participants: Team members, and stakeholders.
- Frequency: When new changes arise and need to inform the stakeholders.

By addressing key challenges in traditional systems, the software aims to revolutionize the real estate industry, fostering efficiency, transparency, and a superior user experience for property managers, landlords, and tenants.

3.4.2 Financial Analysis

In developing our software, we aim to keep costs low, relying mainly on hard work and time. The major expense we foresee is when we launch the software on a live production server. These costs are specifically for keeping the software up and running online. Apart

from that, we just need a decent computer with good specs. Our focus is on putting in effort and time rather than worrying about a financial crunch during development[11].

Table 3.6: Project costs

SN	Fundamental	Cost (BDT)
1.	Domain and Hosting	5000-8000
2.	Meeting with stakeholders(Owners, Tenants, Managers)	2500-3000
3.	Testing and SQA	8000-10000
4.	Contingency(15%)	2325-3150
	Total	17,825-24,150

3.5 Summary

The chapter on BashaBari's methodology takes you through key steps in a straightforward manner. It kicks off by carefully figuring out what's needed in the Requirement Analysis section. Then, it moves on to the Proposed Methodology/System Design, where the plan and features of the system are laid out. Next up is the Data Collection/Input Output Analysis, where we gather data and take a close look at how inputs lead to outputs. This step is crucial for shaping the rest of the project. Jumping into Project Management, the chapter explores how we organize and coordinate to make the project happen smoothly. It emphasizes the importance of effective management practices. Lastly, Financial Analysis looks at the money side of things. It's a strategic review of the costs and benefits associated with BashaBari, ensuring we understand the project's financial aspects well. In a nutshell, this chapter gives a clear overview of BashaBari's methodology, covering requirements, design, data details, project management, and financial considerations. Each part plays a role in making sure the project is well-planned and viable.

CHAPTER 4

IMPLEMENTATION

4.1 Overview

In this chapter, we have delved into the essential components and processes required to bring the BashaBari software solution to life. Our focus has been on four critical areas: Implementation Requirements, Prototype Design, System Testing, and Database Implementation. Each of these elements plays a vital role in ensuring that BashaBari effectively addresses the challenges faced by property managers, owners, and tenants. We began by outlining the implementation requirements necessary for the development and deployment of BashaBari. This section detailed the technical specifications, hardware and software prerequisites, and resource allocations essential for the successful execution of the project. By clearly defining these requirements, we established a solid foundation for the subsequent stages of development. Next, we discussed the prototype design, which serves as a preliminary model of the final software product. This section highlighted the importance of creating an initial prototype to visualize the user interface, functionality, and overall user experience. The prototype design phase allowed us to gather valuable feedback from stakeholders, make necessary adjustments, and ensure that the final product meets the needs and expectations of its users. System testing was another crucial area covered in this chapter. We emphasized the importance of rigorous testing procedures to identify and rectify any bugs or issues within the software. This section detailed the various testing methodologies employed, including unit testing, integration testing, and user acceptance testing. Thorough system testing ensures the reliability, stability, and performance of BashaBari, providing a robust and dependable solution for its users. Finally, we explored the database implementation aspect of the project. This section focused on the design and integration of a secure, scalable, and efficient database system to manage the vast amounts of data associated with property management. We discussed the choice of database technology, data modeling, and the implementation of data security measures to protect sensitive information.

4.2 Implementation Requirements

Table 4.1: Implementation Requirements

Front-End	Back-End	Tools	Database
Html5	PHP	Post-Man	MySQL
Css3	Laravel	Xampp	
JavaScript	Composer	Figma	
Vue.Js	Node.Js	VS CODE	
Tailwind CSS		Git-Hub	
		ChatGpt	

4.3 Train Model/ Prototype Design

4.3.1 Home Page

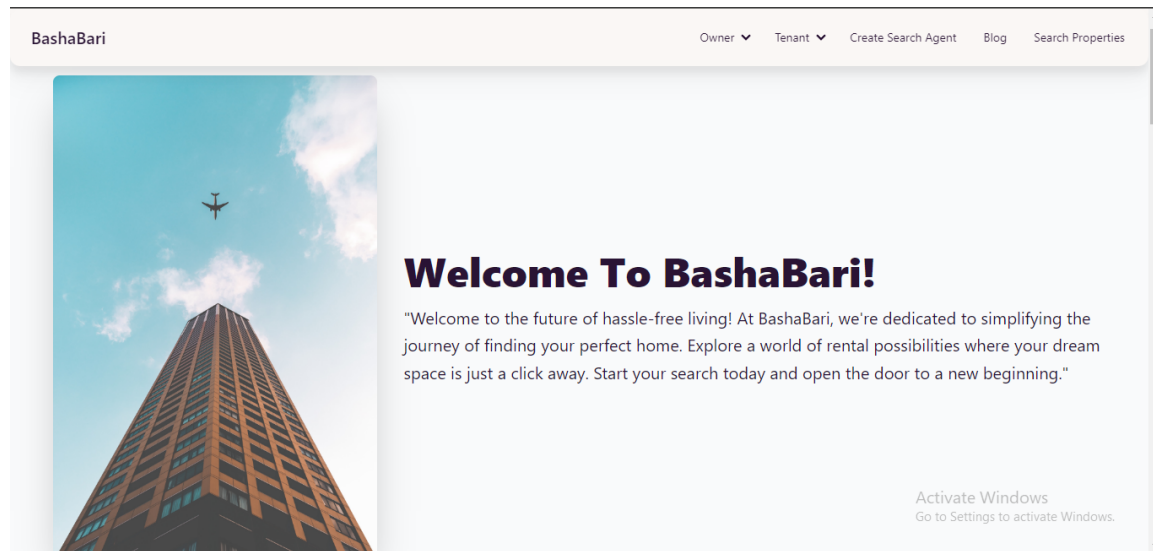


Figure 4.1: Home Page

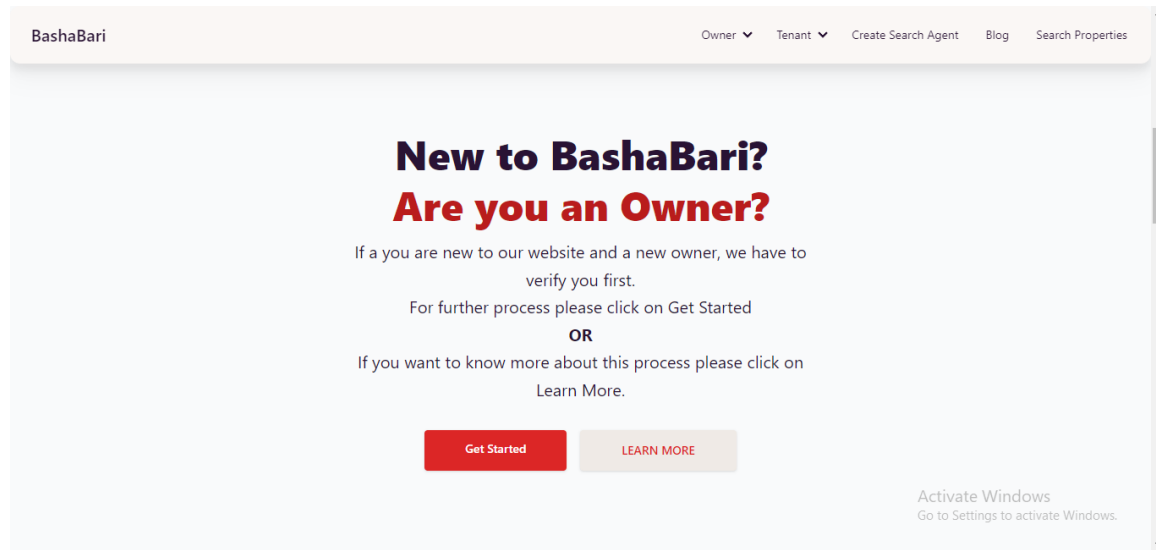


Figure 4.2: Home Page

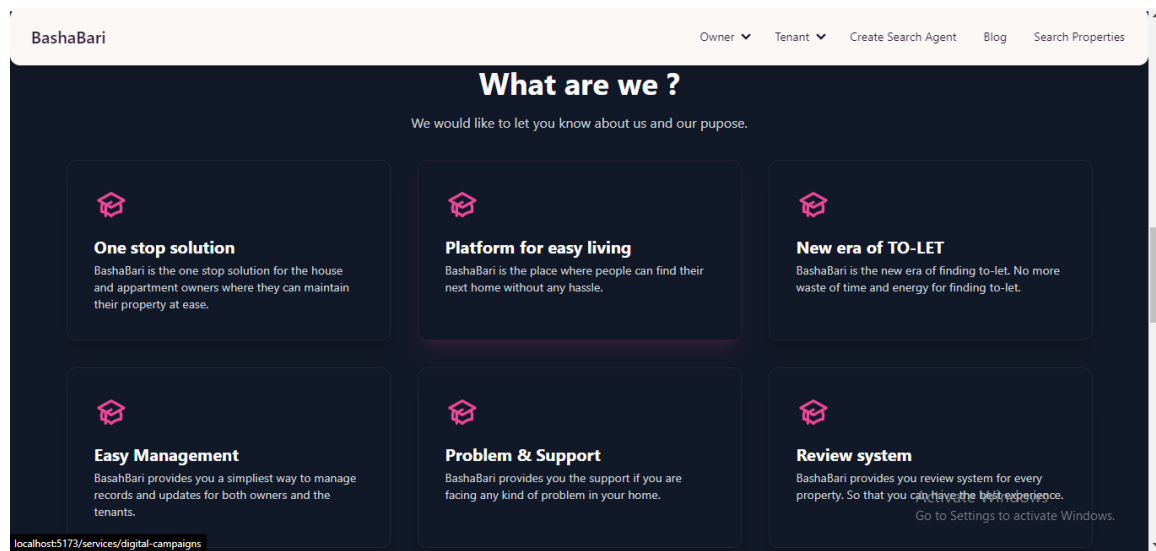


Figure 4.3: Home Page

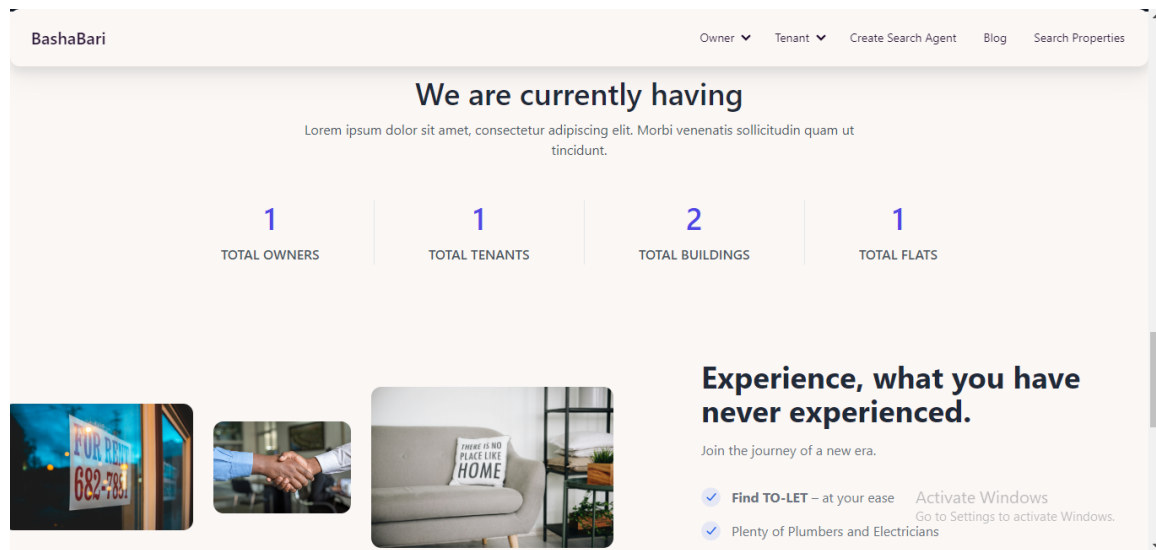


Figure 4.4: Home Page

4.3.2 Contact Us

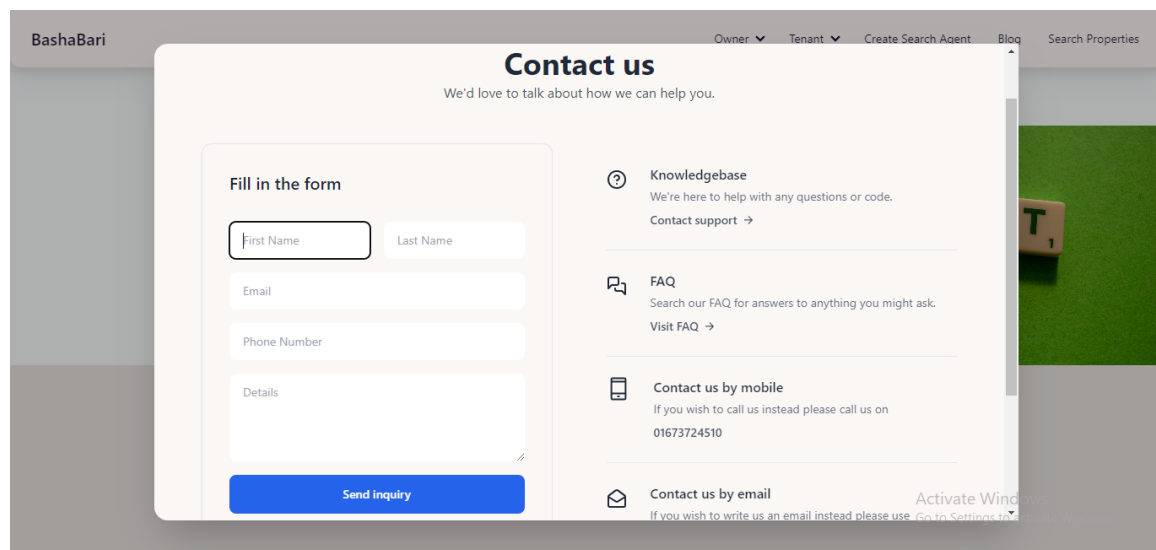


Figure 4.5: Contact Us

4.3.3 Create Search Agent

The screenshot shows the 'Create Search Agent' form in the BashaBari application. The header includes the BashaBari logo and navigation links: Owner, Tenant, Create Search Agent, Blog, and Search Properties. The form is divided into two main sections. The left section, titled 'Create Search Agent', contains a description: 'Send a request to our agents and they will find you your required home according to you desire.' and a note: 'You will be notified via email once we have found your desired home!'. The right section contains the form fields: 'Full Name*', 'Email address*', 'Phone Number', 'Where would you like to live?*', 'Enter your desired rent value', 'Number of rooms you want (min)*', 'Select your rent type' (dropdown), 'Select your package' (dropdown), 'Which month you want get the rent?' (dropdown), 'Additional Requirements', and 'Activate Windows' (button with text 'Go to Settings to activate Windows.').

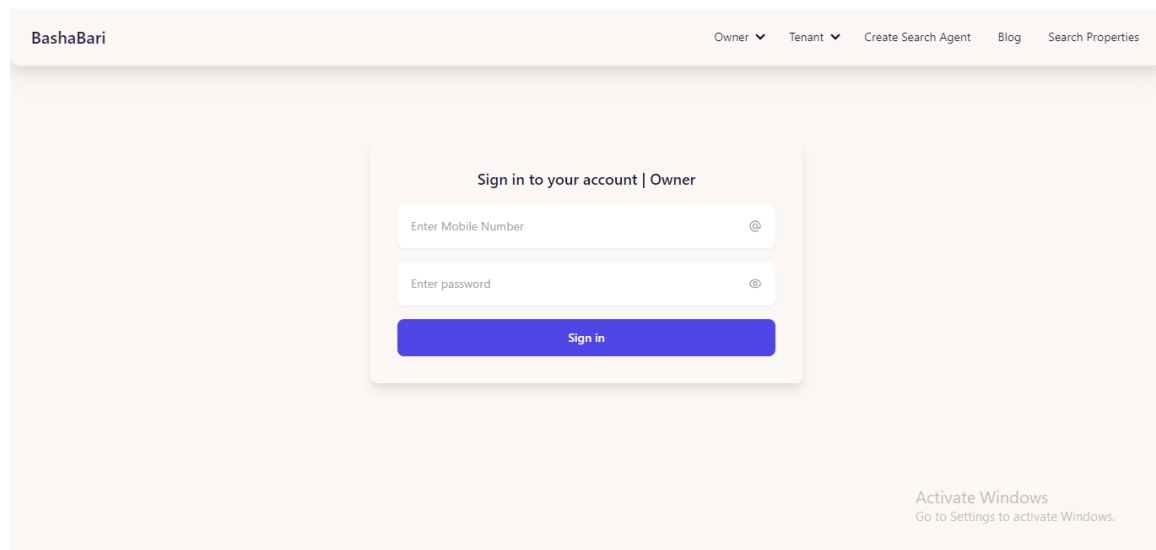
Figure 4.6: Create Search Agent

4.3.4 Add Listing

The screenshot shows the 'Add Listing' form in the BashaBari application. The header includes the BashaBari logo and navigation links: Owner, Tenant, Create Search Agent, Blog, and Search Properties. The form is divided into two main sections. The left section, titled 'Add Listing', contains a description: 'Create your property listing with proper and detailed information.' and a note: 'You can manage your property once you have created.' Below the note are four buttons: 'Building', 'Flat', 'Shop', and 'Office'. The right section contains the form fields: 'Building Name*', 'Building Address*', 'Developer Name', 'mm/dd/yyyy' (date picker), 'What is the name of the area?*', 'What is the name of the city?*', 'What is the name of the district', 'What is the postal code of the area', 'What is the avg price range of the flats of the building?*', 'Gas (Line)', 'Electricity (DESCO)', 'Lift (Available)', 'Is parking available?', 'Activate Windows' (button with text 'Go to Settings to activate Windows.'), and 'Choose File' (button with text 'No file chosen').

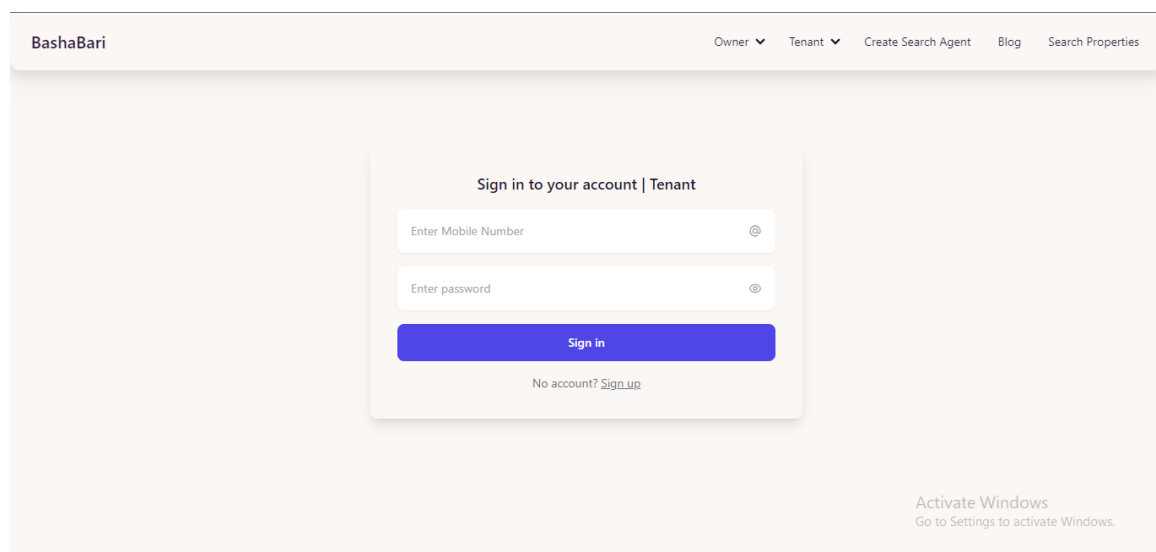
Figure 4.7: Add Listing

4.3.5 Sign In



The image shows the 'Owner Sign In' form on the BashaBari website. The header includes the 'BashaBari' logo and navigation links for 'Owner', 'Tenant', 'Create Search Agent', 'Blog', and 'Search Properties'. The main form is titled 'Sign in to your account | Owner' and contains two input fields: 'Enter Mobile Number' with a phone icon and 'Enter password' with an eye icon. A blue 'Sign in' button is positioned below the fields. In the bottom right corner, there is a message: 'Activate Windows Go to Settings to activate Windows.'

Figure 4.8: Owner Sign In



The image shows the 'Tenant Sign In' form on the BashaBari website. The header is identical to the previous form. The main form is titled 'Sign in to your account | Tenant' and contains the same two input fields: 'Enter Mobile Number' and 'Enter password'. A blue 'Sign in' button is located below the fields. Below the button, there is a link that says 'No account? [Sign up](#)'. In the bottom right corner, the same 'Activate Windows' message is present.

Figure 4.9: Tenant Sign In

4.3.6 Registration

BashaBari

Owner ▼ Tenant ▼ Create Search Agent Blog Search Properties

Welcome to your home

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Eligendi nam dolorum aliquam, quibusdam aperiam voluptatum.

Full Name*

Mobile Number*

Email (optional)

Address (optional)

National Identity Number (optional)

Password*

Password Confirmation*

By creating an account, you agree to our [terms and conditions](#) and [privacy policy](#).

Create an account

Already have an account? [Log in](#).

Activate Windows
Go to Settings to activate Windows.

Figure 4.10: Tenant Registration

4.3.7 Owner Home Page

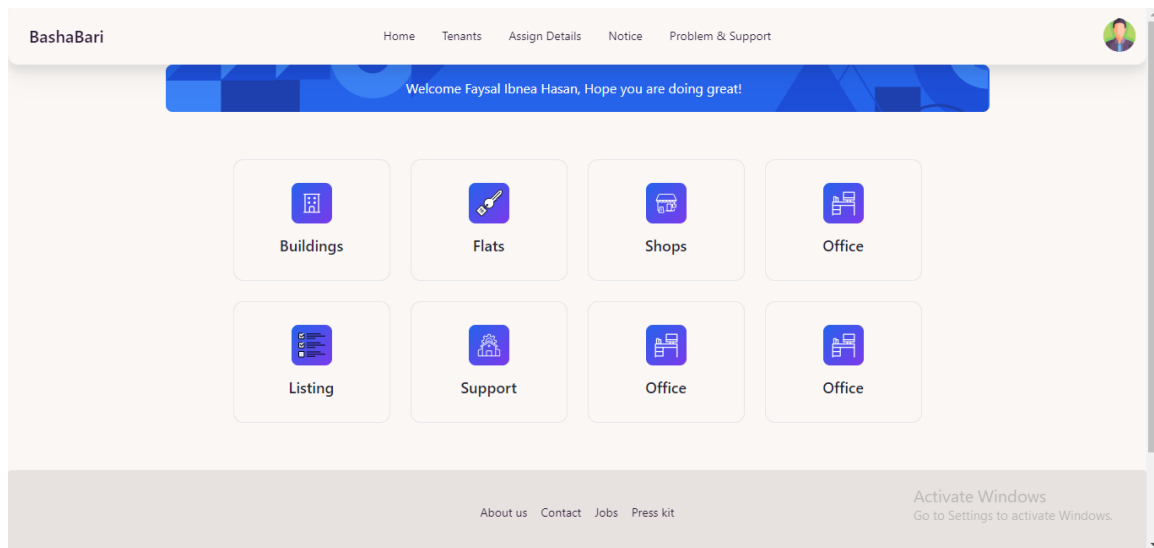


Figure 4.11: Owner Home Page

4.3.8 Tenant List

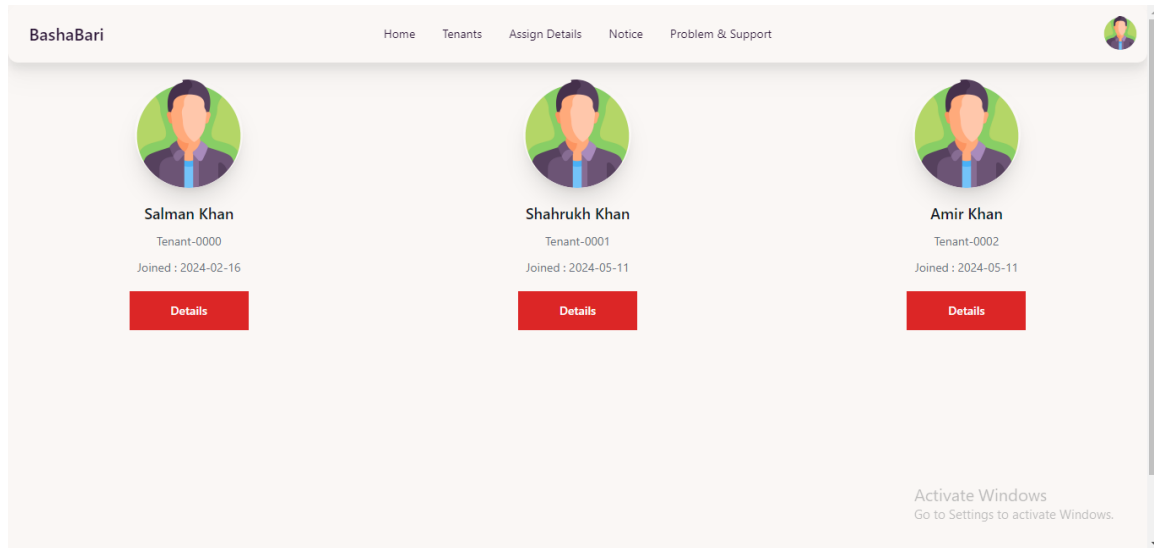


Figure 4.12: Tenant List

4.3.9 Remove Tenant

BashaBari					
Home Tenants Assign Details Notice Problem & Support					
SN	Building ID	Flat ID	Tenant	Joined Date	
1	Building-0000	Flat-0000	Salman Khan (Tenant-0000)	2024-02-16	REMOVE
2	Building-0000	Flat-0001	Shahrukh Khan (Tenant-0001)	2024-05-11	REMOVE
3	Building-0000	Flat-0002	Amir Khan (Tenant-0002)	2024-05-11	REMOVE

Activate Windows
Go to Settings to activate Windows.

Figure 4.13: Remove Tenant

4.3.10 Notice

The screenshot shows a web application interface for 'BashaBari'. At the top, there is a navigation bar with links: Home, Tenants, Assign Details, Notice, and Problem & Support. A user profile icon is visible in the top right corner. Below the navigation bar, there is a blue header bar. The main content area displays a table with columns 'Building ID' and 'Title'. One row is visible with 'Building-0000' and 'Water Issue'. To the right of the table, there are 'Update' and 'Delete' buttons. A modal form is open in the center, titled 'Notice'. It contains the following fields: 'Building ID' (text input), 'Title' (text input), 'Description' (text area), and a date picker set to 'mm/dd/yyyy'. A 'CREATE' button is at the bottom of the modal. In the bottom right corner, there is a watermark that says 'Activate Windows Go to Settings to activate Windows'.

Figure 4.14: Notice

4.3.11 Problem and Support(Owner)

The screenshot shows the 'Problem & Support' section of the BashaBari application. It features a table with four columns: FLAT, PROBLEM, DATE, and STATUS. The table contains three rows of data. Each row has a flat icon, the flat number 'Flat-0000', a problem description, a date, and a status 'Unsolved' with a red dot icon. The bottom right corner of the page has a watermark that says 'Activate Windows Go to Settings to activate Windows'.

FLAT	PROBLEM	DATE	STATUS
Flat-0000	Electricity line damaged Electricity line is damaged.Please fix this asap	2023-11-03	Unsolved
Flat-0000	aejhfbw s.dkfw	2023-11-07	Unsolved
Flat-0000	sdf sdfsdf	2023-11-07	Unsolved

Figure 4.15: Problem and Support

4.3.12 Profile

BashaBari Home Tenants Assign Details Notice Problem & Support

Profile

Manage your name, password and account settings.

Profile photo  [Choose File](#) No file chosen [Upload photo](#)

Full name

Password

Phone

Email

Address

National Identity Number

Activate Windows
Go to Settings to activate Windows.

Figure 4.16: Profile

4.3.13 Create Property

BashaBari Home Tenants Assign Details Notice Problem & Support

At the same time, the fact that we are wholly owned and totally independent from manufacturer and other group control gives you confidence that we will only recommend what is right for you.

0151 475 4450
282 Kevin Brook, Imogeneborough, CA 58517

Building Name

Building Address

Developer

Price Range

mm/dd/yyyy

☐ Gas (Line) ☐ Electricity (DESCO)

☐ Lift (Available)

PARKING AREA CITY

DISTRICT POSTAL CODE

[Create](#)

Activate Windows
Go to Settings to activate Windows.

Figure 4.17: Create Property

4.3.14 Flat

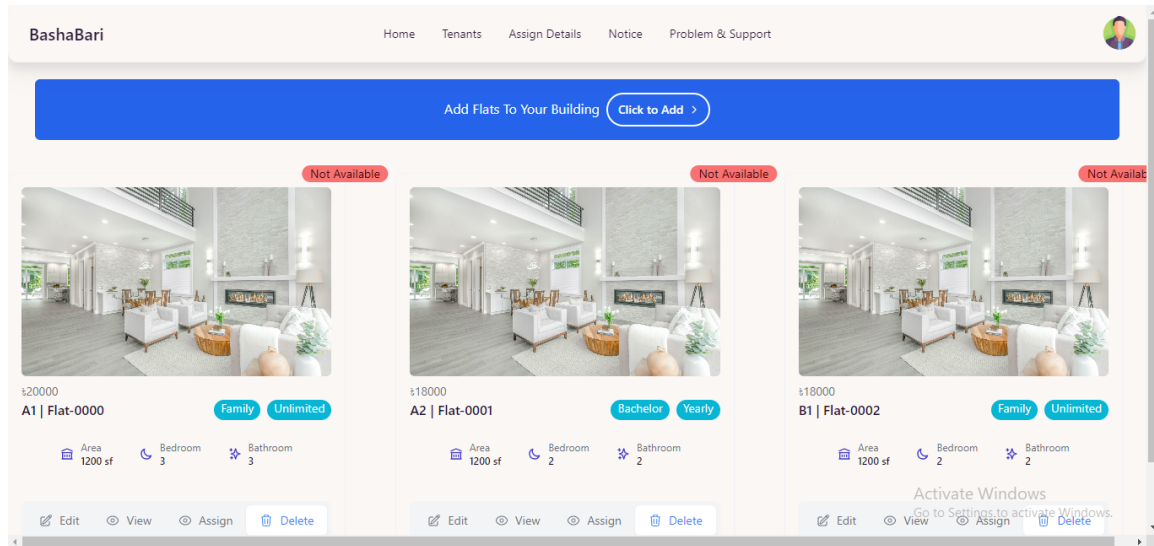


Figure 4.18: Flat

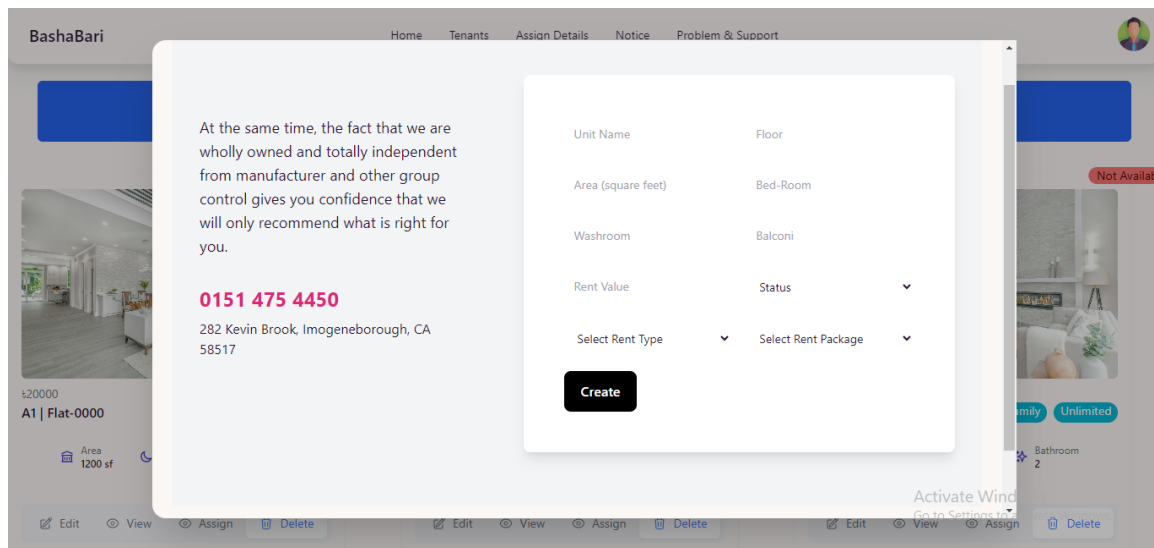


Figure 4.19: Create Flat

4.3.15 Assign Tenant

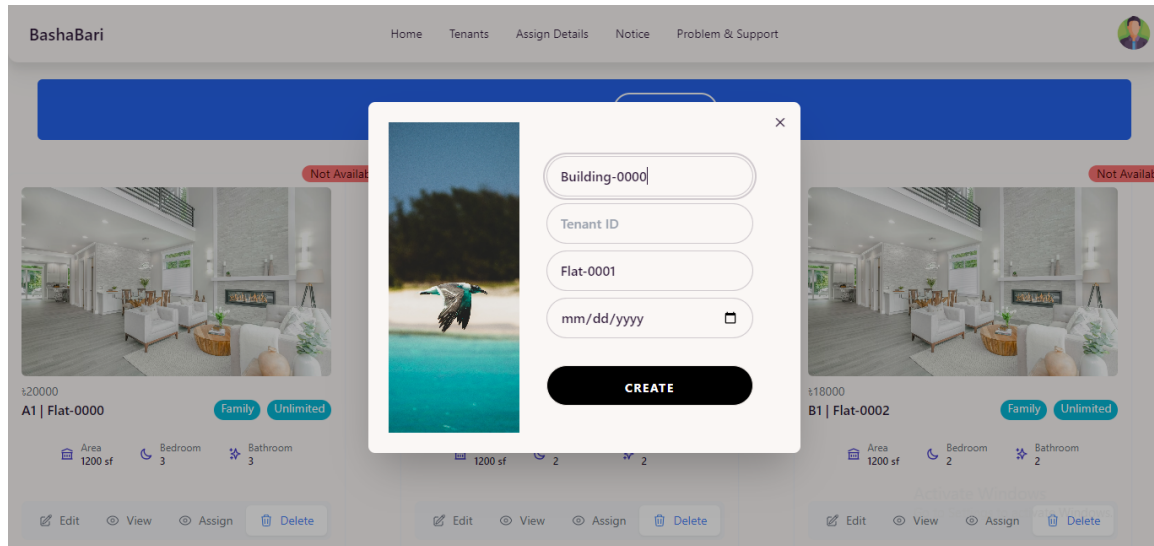


Figure 4.20: Assign Tenant

4.3.16 Tenant Home Page

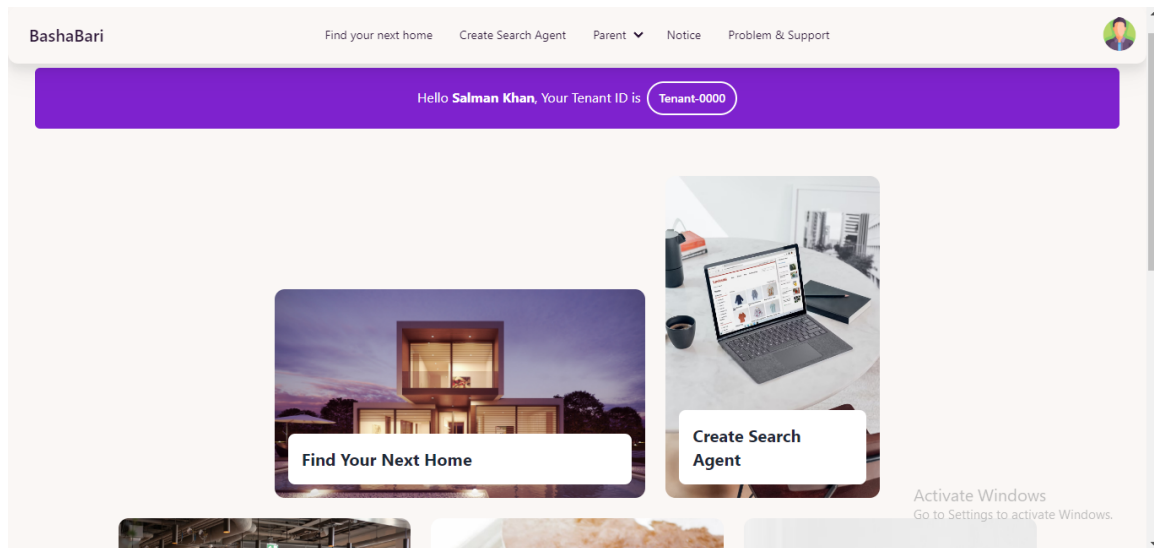


Figure 4.21: Tenant Home Page

4.3.17 Search Property

BashaBari

Find your next homeCreate Search AgentNoticeProblem & Support

Building Name

Daffodil Apartment

Total Flats

3

Address

60 feet,Mirpur-2

Developer

Daffodil Ltd

Developed Date

2024-02-13

Owner

Faysal Ibnea Hasan

Owner's Contact Number

01673724510

Parking Facility

Available

All

Activate Windows
Go to Settings to activate Windows.

Figure 4.22: Search Property

Parking FacilityAvailable

All

Not Available

₹20000

A1

Family

Unlimited

Area1200 sf

Bedroom3

Bathroom3

Not Available

₹18000

A2

Bachelor

Yearly

Area1200 sf

Bedroom2

Bathroom2

Not Available

₹18000

B1

Family

Unlimited

Area1200 sf

Bedroom2

Bathroom2

Activate Windows
Go to Settings to activate Windows.

Figure 4.23: Search Property

©Daffodil International University, Bangladesh

48

4.3.18 Problem and Support(Tenant)

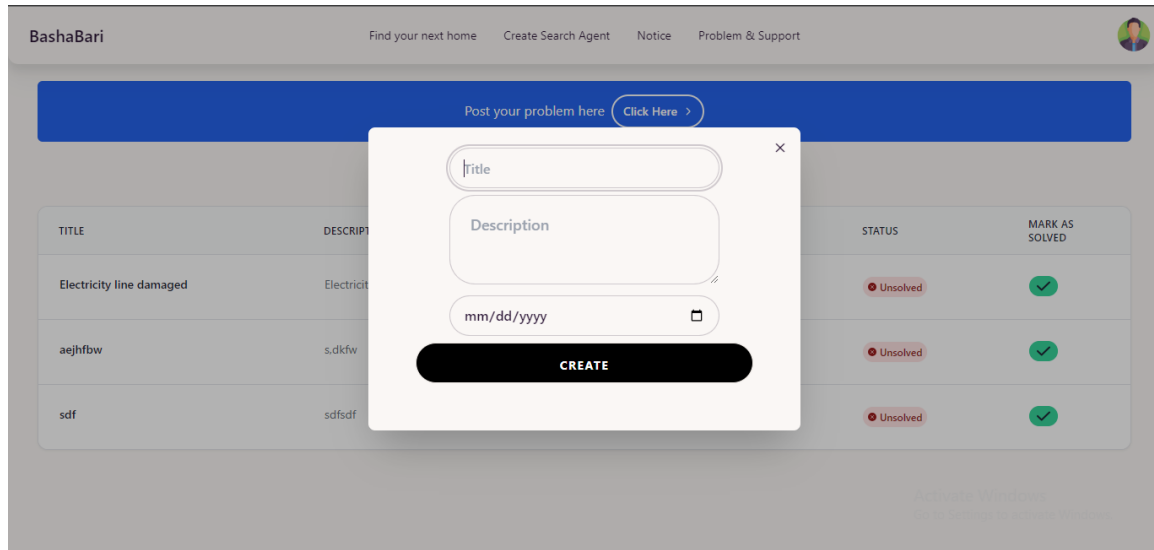


Figure 4.24: Problem and Support

4.4 Implementation of Database

For storing and managing data we have used MySQL database. Here we can create, update, read, and delete data. As MySQL is a relational database management system, so we will experience better performance and efficiency.

4.4.1 Data Tables

Table 4.2: Admin

Field	Data Type
id	bigInt
admin_Id	String
name	String
password	String
status	String
created_at	timestamps

updated_at	timestamps
------------	------------

Table 4.3: Owner

Field	Data Type
id	bigInt
name	String
mobile	String
password	String
address	String
email	String
image	String
nid	String
created_at	timestamps
updated_at	timestamps

Table 4.4: Tenant

Field	Data Type
id	bigInt
tenant_Id	String
name	String
mobile	String
password	String
address	String
image	String
nid	String
email	String

assign_status	String
created_at	timestamps
updated_at	timestamps

Table 4.5: Building

Field	Data Type
id	bigInt
owner_Id	String
building_Id	String
name	String
developer	String
date	date
parking	String
area	String
city	String
district	String
postal_code	String
address	String
facilities	String
price_range	String
image	String
created_at	timestamps
updated_at	timestamps

Table 4.6: Flat

Field	Data Type
-------	-----------

id	bigInt
building_Id	String
flat_Id	String
owner_Id	String
unit_name	String
floor	String
area	String
room	String
washroom	String
balconi	String
rent_value	String
image	String
rent_type	String
rent_package	String
status	String
created_at	timestamps
updated_at	timestamps

Table 4.7: Rent

Field	Data Type
id	bigInt
tenant_Id	String
flat_Id	String
owner_Id	String
building_Id	String
joined_at	date

left_at	date
created_at	timestamps
updated_at	timestamps

Table 4.8: Create Search Agent

Field	Data Type
id	bigInt
name	String
mobile	String
email	String
city	String
room	String
rent_value	String
rent_type	String
rent_package	String
rent_month	String
additional_requiremnets	String
status	String
created_at	timestamps
updated_at	timestamps

Table 4.9: Notice

Field	Data Type
id	bigInt
title	String
description	String

building_Id	String
owner_Id	String
date	date
created_at	timestamps
updated_at	timestamps

Table 4.10: Problem

Field	Data Type
id	bigInt
tenant_Id	String
flat_Id	String
building_Id	String
title	String
description	String
date	date
status	String
created_at	timestamps
updated_at	timestamps

Table 4.11: Supervisor

Field	Data Type
id	bigInt
name	String
supervisor_Id	String
mobile	String
image	String

created_at	timestamps
updated_at	timestamps

Table 4.12: Electrician

Field	Data Type
id	bigInt
name	String
electrician_Id	String
mobile	String
image	String
created_at	timestamps
updated_at	timestamps

Table 4.13: Plumber

Field	Data Type
id	bigInt
name	String
plumber_Id	String
mobile	String
image	String
created_at	timestamps
updated_at	timestamps

Table 4.14: Support

Field	Data Type
id	bigInt

owner_Id	String
building_Id	String
supervisor_Id	String
plumber_Id	String
electrician_Id	String
created_at	timestamps
updated_at	timestamps

4.5 System Testing/ Model Evaluation

4.5.1 Testing Types

- Unit Testing: Validate the functionality of individual parts, features, and modules to ensure they operate as intended.
- Integrity Testing: Evaluate the interaction between different modules to ensure seamless functionality and integration.
- System Testing: Verify that the system's behavior and overall functionality meet specified criteria.
- User Acceptance Testing (UAT): Engage stakeholders in scenario-based testing, using real-world scenarios to ensure the platform meets user demands.
- Security Testing: Conduct assessments to identify weaknesses and ensure robust defense against attacks.
- Performance Testing: Measure the responsiveness and stability of the system under various loads to ensure optimal performance.

4.5.2 Testing Procedure

- Test Case Development: Develop thorough test cases covering all functionalities, edge cases, and user scenarios comprehensively.
- Execution and Reporting: Systematically execute test cases, record results, and generate detailed reports outlining passed and failed scenarios.
- Bug Tracking: Utilize bug tracking tools to log and monitor identified issues, prioritize

them, and track their resolutions effectively.

- Regression Testing: Conduct regression testing post bug fixes or new feature implementations to confirm existing functionalities remain unaffected.
- User Feedback Incorporation: Integrate user feedback obtained from UAT into testing cycles to address user experience issues and enhance overall usability.

4.5.3 Testing Environments

- Development Environment: Developers validate individual components and functionalities in their local development environments.
- Staging Environment: Integration testing and overall system functionality assessment occur in an environment that mirrors the production environment.
- User Testing Environment: Actual users or stakeholders engage in UAT within a controlled testing environment.

4.6 Summary

In summary, this chapter has provided a comprehensive overview of the key components involved in the development of BashaBari. By meticulously addressing the implementation requirements, designing a functional prototype, conducting thorough system testing, and implementing a robust database system, we have laid the groundwork for a successful software solution. Each of these elements is integral to the overall effectiveness of BashaBari, ensuring that it meets the needs of property managers, owners, and tenants, and ultimately revolutionizes the property management experience.

CHAPTER 5

RESULT AND ANALYSIS

5.1 Overview

This chapter presents the experimental test results and comparative analysis of the software and its various aspects. Through meticulously designed test cases, we have obtained empirical data that provides a comprehensive understanding of the software's performance under varied conditions. These results not only validate our design choices but also highlight areas for improvement and optimization. They serve as a pivotal foundation for enhancing the software's functionality, ensuring its reliability, scalability, and user satisfaction in future iterations of development.

5.2 Experimental/ Simulation Result

The testing phase is a crucial step in project development, involving the evaluation of predefined results to enhance efficiency. By meticulously conducting various tests and recording the results in a structured table, we ensure thorough assessment and optimization of the project.

Table 5.1: Test cases and result analysis

Test Case	Input Test Data	Expected Outcome	Actual Outcome	Pass/Fail
Owner Login	Mobile and password	Login successfully	Login successfully	Pass
Tenant Login	Mobile and password	Login successfully	Login successfully	Pass
Owner Register	Required details	Created successfully	Created successfully	Pass
Tenant Register	Required details	Created successfully	Created successfully	Pass
Add Listing	Required details	Added successfully	Added successfully	Pass

Create Search Agent	Required details	Created successfully	Created successfully	Pass
Search Property	Select filter	Filtered successfully	Filtered successfully	Pass
Create Property	Required details	Created successfully	Created successfully	Pass
Assign Tenant	Required details	Assigned successfully	Assigned successfully	Pass
Create Notice	Required details	Created successfully	Created successfully	Pass
Create Problem Details	Required details	Created successfully	Created successfully	Pass
Register With a Pre-used Number	Required details	Registration failed	Registration failed	Pass
Update Profile	Required details	Updated successfully	Updated successfully	Pass
Login With Invalid Details	Mobile and password	Login failed	Login failed	Pass

5.3 Performance/ Comparative Analysis

The comparative analysis examines key aspects of property rental management systems (PRMS), contrasting the features and benefits of established PRMS with the innovative approach of BashaBari. The comparison highlights the strengths and areas for improvement in existing systems while showcasing the unique advantages BashaBari brings to the table.

Geographical Inclusivity

- **Established PRMS:** Traditionally, established PRMS are primarily focused on urban areas. This limited geographical inclusivity means that rural property management needs are often overlooked, resulting in a gap in service provision for a significant portion of the population.

- **BashaBari:** In contrast, BashaBari is designed to be inclusive of both urban and rural areas. This comprehensive approach ensures that property owners and tenants in rural regions are not neglected, providing a more balanced and extensive service coverage.

User Interface (UI)

- **Established PRMS:** The user interfaces of many established PRMS often lack intuitiveness and are visually unappealing. This can result in a steeper learning curve and reduced user satisfaction.
- **BashaBari:** Prioritizing a user-centric design, BashaBari features an intuitive and engaging UI. This focus on user-friendly design makes the software accessible and appealing, encouraging widespread adoption among users of varying technical proficiencies.

User Experience (UX)

- **Established PRMS:** Users of established PRMS frequently encounter a disjointed user experience, with compromised software performance leading to frustration and inefficiency.
- **BashaBari:** BashaBari places a strong emphasis on providing a seamless and accessible UX, especially for non-technical users. By ensuring smooth interactions and consistent performance, BashaBari enhances overall user satisfaction and productivity.

Target Audience

- **Established PRMS:** These systems are predominantly designed for technical users, which can alienate non-technical property owners and tenants who might struggle with complex functionalities.
- **BashaBari:** Designed with non-technical and regular users in mind, BashaBari simplifies property management processes. This inclusive approach makes the software accessible to a broader audience, including those with limited technical expertise.

Functionalities

- **Established PRMS:** While established PRMS offer valuable functionalities, they often fall short in addressing the specific needs of rural property management, limiting their overall effectiveness.
- **BashaBari:** BashaBari offers comprehensive features that cater to both urban and rural property and rent management needs. This dual focus ensures that all users, regardless of their location, have access to essential tools and services.

Adoption in Third-World Nations

- **Established PRMS:** The adoption of established PRMS is limited, especially in countries like Bangladesh. The lack of localized solutions and adaptability to specific contexts hampers widespread usage.
- **BashaBari:** BashaBari aims to enhance adoption and effectiveness within the unique context of Bangladesh. By addressing local needs and challenges, BashaBari is poised to achieve higher adoption rates and provide more relevant services to its users.

Overall Effectiveness

- **Established PRMS:** The overall effectiveness of established PRMS varies, with significant room for substantial upgrades. Their limitations in inclusivity, user-friendliness, and adaptability often hinder their full potential.
- **BashaBari:** BashaBari aims for elevated effectiveness and user satisfaction, particularly within the Bangladeshi context. By addressing the shortcomings of existing systems and introducing innovative features, BashaBari strives to set a new standard in property management software.

5.4 Summary

In summary, BashaBari represents a forward-thinking alternative to established Property Management and Rental Systems (PRMS) by not only addressing their inherent limitations but also introducing innovative features tailored to meet the diverse needs of today's users. With a strong emphasis on inclusivity, BashaBari ensures accessibility across different demographics and property types, enhancing overall user experience. Its localized adoption strategy further distinguishes it as a versatile solution capable of adapting to various market dynamics and regulatory environments. By prioritizing user feedback and leveraging cutting-edge technology, BashaBari stands out as a comprehensive and future-ready platform for overcoming the multifaceted challenges in modern property management.

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY

6.1 Impact on Life

The impact of "BASHABARI: A COMPLETE RENTAL SOLUTION FOR PROPERTY OWNERS & TENANTS" on people's lives can be significant in several ways:

1. **Convenience:** It makes the process of finding rental properties or tenants more convenient and efficient by centralizing listings and providing streamlined services like online applications and lease management.
2. **Time Savings:** Both property owners and tenants save time that would otherwise be spent searching for properties or tenants, screening applicants, negotiating leases, and managing rental agreements.
3. **Improved Transparency:** Can enhance transparency in the rental process by providing clear terms, legal information, and guidelines, reducing misunderstandings between landlords and tenants.
4. **Financial Management:** For property owners, services that include rent collection and financial management can streamline income tracking and ensure consistent cash flow.
5. **Legal Compliance:** By offering guidance on legal matters related to renting, these platforms help ensure that landlords and tenants adhere to local regulations, promoting a fair and lawful rental environment.

6.2 Impact on Society & Environment

6.2.1 Impact on Society

"BashaBari: A complete rental solution property owners and tenants" is like an asset for the society. The motivation behind this project is to simplify the hazards of property and rent management, especially in Bangladesh. Here is a glimpse of the impacts on society:

- Encourage people to use technology in our traditional works like property and rent management.

- By reducing the paperwork of rent management both the property manager and tenant will enjoy the taste of less time complexity and efficiency.
- Replacing our traditional old-fashioned TO-LET system with the new age technology, so that we can keep up with other fast-growing countries.
- Plumbers and Electricians will find a better and more prominent platform for their earnings.
- Having a great impact on the economic growth of the country and other co-related sectors like technology, youth encouragement, job placement, and so on.
- Motivating upcoming tech ideas and entrepreneurship as well as organizations and companies.

6.2.2 Impact on Environment

"BashaBari: A complete rental solution property owners and tenants" has no direct impact on the environment but it has a impacts on the environment. Here is a glimpse of the impacts on the environment:

- By digitizing property and rent management processes, the platform eliminates the requirement for manual agreements, itineraries, and other paperwork, contributing to paper conservation and waste reduction.
- Reduce the travel hazards and minimize the time complexity as all the pre-processing is done before the execution of the work.
- By enlisting data and using data analytics it can be determined whether a city is ideal or not based on how many properties are there and how many should be there.

6.3 Ethical Aspects

"BashaBari: A complete rental solution property owners and tenants" is strongly based on ethical aspects. Without ethically transparent, anything can collapse in the blink of an eye. Here is a glimpse of the ethical aspects:

- Implementing robust security and privacy protocols is essential to protect user data, including tenant information, building details, and personal data. Adhering to data privacy laws like GDPR, the platform ensures user consent before collecting and

utilizing their data.

- Provide users with clear and comprehensive information regarding data usage, privacy regulations, and service terms. Obtain written consent from users before gathering and utilizing their data.
- To ensure accessibility for all users, it's important to design the platform to accommodate individuals with impairments and adhere to accessibility principles, thus guaranteeing inclusiveness for everyone.
- Maintain honesty and integrity in company practices by being transparent in pricing, services offered, and customer support, thereby fostering accountability and trustworthiness.
- The practice of ethical advertising and marketing involves ensuring that platform strategies adhere to moral guidelines and avoid using deceptive or misleading tactics.

6.4 Sustainability Plan

Creating a sustainability strategy for "BashaBari: A complete rental solution property owners and tenants" involves implementing approaches to ensure long-term sustainability, reduce environmental impact, and promote ethical and responsible conduct. Below is an outline:

6.4.1 Data Processing and Management

- Ensure regular updates to privacy rules to comply with international data protection laws while safeguarding user privacy.
- Continuously enhance data encryption and security protocols to protect user data from breaches.
- Adhere to data minimization principles by collecting and retaining only the necessary amount of user data required for platform functionality.

6.4.2 Improvement and Review

- Consistently gather user input to enhance features and ensure alignment with user needs and preferences.
- Stay abreast of technological advancements and integrate new features to enhance user

interfaces and overall productivity.

- Continuously monitor and report on privacy compliance, user satisfaction, and environmental impact data to stakeholders.

6.4.3 Business Tactics and Transparency

- Maintain transparency regarding costs, services offered, and company operations to foster consumer loyalty and trust.
- Uphold ethical corporate standards by establishing fair and principled partnerships with suppliers and other stakeholders.

6.5 Summary

BashaBari is a multifaceted platform poised to deliver substantial benefits across various sectors. Beyond its direct advantages to property owners and tenants, it promises positive societal and environmental impacts. By incorporating ethical considerations and a robust sustainability plan, BashaBari aims to set a precedent in Property and Rental Management Software (PRMS). This platform enhances convenience and efficiency in property transactions, facilitating seamless interactions between landlords and tenants. It promotes transparency through clear terms and legal guidance, fostering trust and minimizing disputes. Moreover, BashaBari contributes to environmental sustainability by implementing practices that reduce ecological footprint, thereby exemplifying responsible corporate citizenship. Through these efforts, BashaBari not only elevates operational standards within PRMS but also establishes itself as a role model for ethical business practices and sustainable development in the industry.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions

Along with the vision of revolutionizing the traditional work tactics of property and rent management, "BashaBari: A complete rental solution property owners and tenants" is a step toward the future. By offering a seamless, user-friendly interface, personalized recommendations, and robust security measures, the platform addresses diverse user needs while prioritizing data privacy and ethical conduct.

The platform exemplifies its dedication to ethical business practices by employing environmental sustainability methods, prioritizing ongoing development, and addressing ethical considerations. Making a reliable platform for easy property management and a proper rental solution, was a great journey. With the vision of digitalizing both urban and rural area rental systems, our platform brings all the major and minor aspects to one root.

7.2 Further Suggested Works

"BashaBari: A complete rental solution property owners and tenants" behold the potential of expanding in vast aspects. Here is a glimpse of the future development and scope:

- Implement Machine Learning (ML) and Artificial Intelligence (AI) in the project.
- Work with automated eco-system and IOT-based technology, for making smart property management.
- Implement blockchain technology to enhance trust and security in payments and data management by facilitating transparent and secure transactions.
- Expand the platform's global reach by incorporating additional languages, currencies, and region-specific property and rental information to cater to a broader audience of international users.
- Add more property options like single seats, hostels, warehouses, factories, and so on.
- Employ state-of-the-art security protocols to safeguard user data and stay ahead of evolving cybersecurity threats through continuous security measures.
- Build a blog section where users can explore the real state area more precisely and it

will be updated regularly.

- Connect stakeholders on a social community platform as they can discuss many aspects regarding their thoughts and opinions.

By embracing these opportunities for expansion, "BashaBari: A complete rental solution property owners and tenants" has the potential to evolve into a highly innovative and comprehensive Property and Rent Management System (PRMS) platform. Through enhanced functionality, increased accessibility, and a focus on delivering a superior user experience, the platform aims to cater to the diverse needs of property owners and tenants worldwide.

7.3 Limitations/ Conflict of Interests

1. **Data Privacy and Security Concerns:** BashaBari may face challenges related to ensuring robust data privacy and security measures, especially concerning sensitive tenant and landlord information stored on the platform.
2. **Legal and Regulatory Compliance:** Compliance with local real estate laws and regulations in various regions where BashaBari operates could pose a challenge. Ensuring that the platform adheres to all legal requirements might require continuous updates and adjustments.
3. **User Adoption and Engagement:** Encouraging landlords and tenants to adopt and actively use BashaBari could be a hurdle, particularly among those accustomed to traditional methods of property management.
4. **Maintenance of Property Data Accuracy:** Maintaining accurate and up-to-date property listings and information could be challenging, especially as properties are added, removed, or modified over time.
5. **Conflicts of Interest Between Landlords and Tenants:** BashaBari's role as an intermediary between landlords and tenants may occasionally lead to conflicts of interest, particularly in cases involving disputes over lease terms, maintenance responsibilities, or rental payments.
6. **Technological Dependencies and Risks:** Reliance on technology for platform operation introduces risks such as system downtime, data breaches, or technical glitches that could impact user experience and trust in the service.

7. **Market Competition and Differentiation:** Differentiating BashaBari from other property management solutions in a competitive market may require continuous innovation, unique features, or specialized services that meet specific user needs.
8. **Financial Sustainability:** Maintaining a sustainable business model that balances revenue generation with providing value to landlords and tenants may present ongoing challenges, especially in fluctuating economic conditions.
9. **Ethical Considerations:** Addressing ethical considerations, such as fairness in property listing prioritization or handling tenant information with integrity, is crucial to maintaining trust and reputation among users.
10. **Cultural and Social Acceptance:** Adapting BashaBari's services to fit cultural norms and social expectations in different regions or communities may require careful consideration and customization to foster acceptance and usage.

Navigating these limitations and conflicts of interest effectively is essential for BashaBari to establish itself as a reliable and trusted platform for property management solutions.

References:

- [1] Learn about Laravel, available at << <https://laravel.com/docs> >>, last accessed on 02-26-2024 at 05:13pm.
- [2] Learn about Visual Studio Code, available at << <https://code.visualstudio.com/> >>, last accessed on 02-26-2024 at 05:13pm.
- [3] Learn about Vue.js, available at << <https://vuejs.org/> >>, last accessed on 02-26-2024 at 05:13pm.
- [4] Learn about XAMPP, available at << <https://www.apachefriends.org/index.html> >>, last accessed on 02-26-2024 at 05:13pm.
- [5] Learn about POST-MAN, available at << <https://www.postman.com/> >>, last accessed on 02-26-2024 at 05:13pm.
- [6] Learn about Stack Overflow, available at << <https://stackoverflow.com/> >>, last accessed on 02-26-2024 at 05:13pm.
- [7] Learn about Laracasts, available at << <https://laracasts.com/> >>, last accessed on 02-26-2024 at 05:13pm.
- [8] Learn about Bproperty, available at << <https://www.bproperty.com/> >>, last accessed on 02-26-2024 at 05:13pm.
- [9] Learn about VOBON, available at << <https://vobon.xyz/> >>, last accessed on 02-26-2024 at 05:13pm.
- [10] Learn about Rentola, available at << <https://rentola.com/> >>, last accessed on 02-26-2024 at 05:13pm.
- [11] Henry Peter Gommans, George Mwenda Njiru, Arphaxad Nguka Owange, “Rental House Management System”, International Journal of Scientific and Research Publications, Volume 4, Issue 11, pp.3-9, November 2014.
- [12] B. Henderson, C. Isolda, B. Traditi, C. Matheus, “RENTL: A CLOUD BASED SYSTEM PROVIDING A SIMPLE SOLUTION TO THE COMPLEXITY OF MANAGING RENTAL PROPERTIES”, 13th International Technology, Education and Development Conference, pp. 3038-3043, 2019.

Appendix A

Complex Engineering Problems (EP) and Complex Engineering Activities (EA) Analysis

Title: BASHABARI : A COMPLETE RENTAL SOLUTION FOR PROPERTY OWNERS & TENANTS

Students ID: 201-15-3681

CO Description for FYDP-Phase-I

CO	CO Descriptions	PO
CO1	Integrate recently gained and previously acquired knowledge to identify a real-life complex engineering problem for the Final Year Design Project	PO1
CO2	Analyze different aspects of the goals in designing a solution for the Final Year Design Project	PO2
CO3	Explore diverse problem domains through a literature review, delineate the issues, and establish the goals for the Final Year Design Project	PO4
CO4	Perform economic evaluation and cost estimation and employ suitable project management procedures throughout the development life cycle of the Final Year Design Project	PO11

Addressing of COs, Knowledge Profile (K), and Attainment of Complex Engineering Problems (EP):

SN	EP Definition	Attainment	CO	Justification (with Knowledge Profile)	References
1.	EP1: Depth of Knowledge required	Yes (must be Yes)	CO1, CO2, and CO3	K3 (Engineering Fundamentals) The successful execution of the project necessitates a comprehensive understanding of K3. In the methodology section we have discussed that we will require the knowledge of coding and programming languages. Extending the knowledge aspect we will also need to learn about some technology stacks like Laravel, Vue.js.	Page no: K3(16-35), K4(36-57) Section: K3(3.1, 3.3, 3.4), K4(3.4)

				For database we need the wide knowledge of Database Management System(DBMS) and for that we need to learn MySQL. For API and testing we have to know how POST-MAN works. K4 (Specialist Knowledge) The successful implementation of the project demands a proficient command of K4, ensuring a thorough grasp of contemporary web development technologies and methodologies. This expertise is pivotal in proficiently constructing a Property and Rent Management System, as explicitly articulated in the comprehensive requirement analysis. In requirement analysis we have discussed the technology and tools we will use for the development of the software such as Laravel, Xampp, Post-man etc.	
				K5 (Engineering Practice [Design]) We have stated the use case diagram associated with the use case analysis in the proposed methodology section. This emphasizes the critical role these components play in the project's overall framework and analysis. In this section we can analyze the process step by step and understand the roles of the stakeholders. In this section we have introduced the model we will be using through-out the development phase. As a SDLC method we will be using Waterfall model for better project management and optimization. K6 (Engineering Practice [Technology]) The project necessitates expertise in a technology stack comprising HTML, CSS, JavaScript, Vue.js, Bootstrap,	Page no: K5(37-56), K6(37) Section: K5(4.3, 4.4), K6(4.1)

				Laravel, PHP, jQuery, AJAX, Postman, Node.js, NPM, Composer, and Visual Studio Code, for constructing a comprehensive Property and Rent Management System, which is stated in the requirement analysis. In this section we have discussed the technology we will need. Such as we will use Laravel as the backand of the software and it also requires PHP. On the other side we will use Vue JS for the frontend development which also requires JavaScript, jQuery, AJAX and Bootstrap-Tailwind for designing purpose. In addition we have to use some third-party packages and for that we will need Node.js and NPM. VS CODE will be used as our development environment.	
				K8(Research Literature) The project requires thorough research into industry best practices, user requirements, and market trends to inform the design and implementation of features, ensuring alignment with user needs, scalability, and competitiveness within the property and rent management sector, which is stated in related works of chapter 2. We have studied some related papers for the knowledge of where we can improve rather than the other competitor in the market. This section helps us to build a unique and effective software despite of other Property and Rent Management System(PRMS).	Page no: K8(10-15) Section: K8(2.1, 2.2, 2.3, 2.4)

2.	EP2: Range of Conflicting Requirements	Yes	CO2	The project involves navigating a diverse range of conflicting requirements, including balancing the need for a user-friendly interface with comprehensive functionality, addressing scalability concerns while maintaining performance efficiency, which is stated in requirement analysis. The development of efficient code and optimizing performance could be a challenge we may specifically face.	Page no: 67-68 Section: 7.1, 7.2
3.	EP3: Depth of analysis required	Yes	CO2	The depth of analysis required for this project encompasses extensive examination of user behaviors, market demands, and technological capabilities to effectively address complex challenges and optimize decision-making processes, which is stated in open issues. As we have to compete with the other PRMS softwares, we need to understand and analyze deeply the weakness and strengths of the other competitors.	Page no: 14 Section: 2.4
4.	EP4: Familiarity of Issues	No	CO3		Page no:
5.	EP5: Extends of application codes	No	CO3		Page no:

6.	EP6: Extends of stakeholders involved and conflicting requirements	No	CO3		Page no:
7.	EP7: Interdependence	Yes	CO3	Interdependence within this project manifests in various ways, including the reliance of front-end development on back-end functionality, the integration of external APIs for data exchange, the coordination between different development aspects working on front-end and back-end components, and the interconnectedness of various system features and modules, which is stated in project management.	Page no: 36-57 Section: 4.1, 4.2, 4.3, 4.4
8.	-	Yes	CO4	In the financial analysis section of our project, we have meticulously addressed CO4, which focuses on the ability to evaluate and estimate costs. Our project's budgeting process has been thoroughly outlined, demonstrating a comprehensive understanding of the financial aspects involved for FYDP. In the section we have analyzed the estimated costs such as Development costs, Testing costs, Server and Domain-Hosting costs and the other aspects.	Page no: 35 Section: 3.4

[Document Viewer](#)

Turnitin Originality Report

Processed on: 25-May-2024 10:53 +06

ID: 2387708336

Word Count: 11613

Submitted: 1

FAYSALIBNEAHASANJESAN ID: 201-15-3681
By Faysalibneahasanjesan
Id: 201-15-3681

Faysalibneahasanjesan Id: 201-15-3681

Similarity Index

19%

Similarity by Source

Internet Sources: 16%
Publications: 2%
Student Papers: 13%

exclude quoted

exclude bibliography

exclude small matches

mode:

quickview (classic) report



print

refresh

download

3% match (Internet from 25-Apr-2024)

<http://dspace.daffodilvarsity.edu.bd:8080>

2% match (Internet from 10-Nov-2023)

<http://dspace.daffodilvarsity.edu.bd:8080>

2% match (student papers from 03-Mar-2024)

[Submitted to Daffodil International University on 2024-03-03](#)

1% match (Internet from 10-Apr-2022)

<http://dspace.daffodilvarsity.edu.bd:8080>

1% match (Internet from 06-May-2023)

<http://dspace.daffodilvarsity.edu.bd:8080>

1% match (student papers from 24-May-2024)

[Submitted to Daffodil International University on 2024-05-24](#)

1% match (student papers from 30-Mar-2022)

[Submitted to Daffodil International University on 2022-03-30](#)

1% match (student papers from 14-Aug-2019)

[Submitted to Universiti Teknikal Malaysia Melaka on 2019-08-14](#)

<1% match (Internet from 21-Jul-2023)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 23-Jul-2023)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 26-Oct-2022)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 21-Jul-2023)

<http://dspace.daffodilvarsity.edu.bd:8080>

<1% match (Internet from 29-Jun-2023)