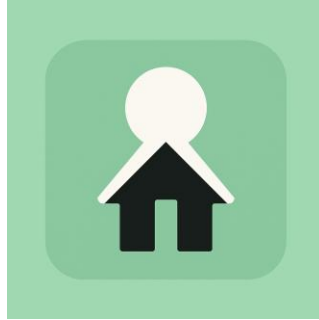




An Najah National University
Faculty of Engineering & Information
Technology
Department of Computer Engineering
Software Graduation Project 1

Student Nest



Prepared by:

Mohamad Kittani

Osama Salah

Supervised by:

Dr. Shareef Yaseen

June 2025

Acknowledgment

We extend our sincere gratitude to everyone who supported the successful creation of *Student Nest*, a property housing website and mobile app. Special thanks to our academic supervisor, Dr. Shareef Yaseen, for his valuable guidance and encouragement throughout the project. As the developers, we take pride in building this platform to support and uplift the housing system in our country. We are especially grateful to our families and friends for their unwavering motivation and support. Finally, we appreciate everyone who offered feedback and encouragement during our development journey.

Table of contents

Abstract

Chapter 1: Introduction

1.1 Statement of the Problem

1.2 Objectives of the Work

1.3 Scope of the Work

1.4 Significance of the Work

1.5 Organization of the Report

Chapter 2: Constraints, Standards/Codes, and Earlier Coursework

2.1 Constraints and Limitations

2.2 Standards and Codes

2.2.1 Backend – Node.js & Express.js

2.2.2 Database – MongoDB

2.2.3 Frontend – Next.js (Web) & React Native (Mobile)

2.2.4 Lahza Authorization & Payment Integration

2.2.5 Software Engineering Methodology – Agile

2.3 Earlier Coursework

Chapter 3: Literature Review

Chapter 4: Methodology

4.1 Tools, Methods, and Programming Languages

4.1.1 Tools

4.1.2 Programming Languages & Frameworks

4.1.3 Database

Chapter 5: Results and Discussion

5.1 Mobile Application

5.1.1 Common Features Among Users

Chapter 6: Conclusions and Recommendations

Chapter 7: Future Works

References

Abstract

Student Nest is a comprehensive housing platform developed to make it easier for individuals in occupied Palestine to post their properties online and for students to access these properties and contact owner to book them. The platform supports two primary user roles: **students** and **owners**, each with personalized experiences, services, and interfaces. Students can explore categorized properties, view owner profiles, chat directly with service providers, and securely place orders using Lahza payments. Owners can create and manage properties, interact with potential clients, and track their bookings. The system includes real-time messaging, a user-friendly navigation bar, and role-based dashboards for efficient interaction. Student Nest is built using **NextJS** for the frontend side, and MongoDB, **NodeJS/ExpressJS**, **REST Framework** for the backend, and **React Native** for the mobile app. By combining modern technology with a mission to facilitate local businesses, the platform promotes online housing booking and digital independence within the Palestinian community.

Chapter 1: Introduction

1.1 Statement of the Problem

In occupied Palestine, students face significant challenges in finding suitable housing due to a lack of localized, reliable, and feature-rich digital platforms. Property owners also struggle to effectively reach students, manage listings, and handle bookings, and process payments in a secure and digital manner. Existing international platforms often overlook the specific socio-economic and infrastructural needs of the Palestinian context, leaving a gap in access to trusted housing solutions. This situation underscores the need for a tailored platform that connects students with housing opportunities in a safe, efficient, and user-friendly way.

1.2 Objectives of the Work

Student Nest aims to develop a full-stack student housing platform accessible via web and mobile, connecting students with verified property owners in Palestine. The system provides role-based access for students and owners, enabling property listings, apartment bookings, secure messaging, notification delivery, and online payment management. The backend is developed with Node.js, Express, and MongoDB to ensure scalability and flexibility, while the frontend is built using Next (web) and optionally React Native (mobile) for an interactive and responsive user experience. By leveraging modern web technologies and best practices, Student Nest seeks to streamline the housing search process and promote digital empowerment in the local student community.

1.3 Scope of the Work

The project involves designing and developing a robust student housing management system, including modules for user registration and authentication, profile management, property listing and search, availability tracking, booking and payment, chat-based communication, notification handling, reporting, and administrative control. The backend is powered by Express.js and MongoDB, with RESTful APIs supporting secure and scalable data transactions. The frontend provides a clean and intuitive interface for both students and property owners. Role-based dashboards, JWT-based authentication, availability calendars, and integrated payments contribute to a seamless and reliable experience. The system is engineered for future scalability and maintainability.

1.4 Significance of the Work

Student Nest addresses a critical gap in the Palestinian job market by providing a comprehensive freelance platform tailored to local users. It empowers skilled individuals—particularly youth—to earn income and showcase their talents through a trusted digital medium. The platform's role-based approach

enhances usability, while real-time messaging and integrated payments simplify collaboration between clients and service providers. By combining technical innovation with a mission of community upliftment, *Student Nest* contributes to economic resilience and digital inclusion in occupied Palestine.

1.5 Organization of the Report

Student Nest addresses a pressing need within the Palestinian academic ecosystem by providing a localized and trustworthy digital housing platform. It empowers students to easily find accommodation near universities, while enabling owners to manage properties and communicate with tenants through an all-in-one system. Real-time messaging, availability updates, and integrated booking/payment flows reduce friction and improve transparency. With a focus on accessibility, safety, and community impact, Student Nest contributes to both educational support and digital inclusion, helping students and owners alike navigate housing challenges in a modernized way.

Chapter 2: Constraints, Standards/Codes, and Earlier Coursework

2.1 Constraints and Limitations

- **Internet & Infrastructure Limitations:** In occupied Palestine, challenges like unstable internet connectivity and restricted access to global financial tools can hinder seamless development, deployment, and real-time features such as messaging and notifications.
- **Payment Gateway Restrictions:** Since Stripe is not supported in Palestine, we integrated a secure third-party payment authorization

provider (Lahza) to simulate and handle transactions in a localized, regulation-compliant way.

- **Time Constraints:** Building a full-stack platform—including mobile, web, backend, and database layers—within the academic schedule required prioritizing core functionality and balancing trade-offs in design and testing.
- **Real-Time Features:** Implementing secure, scalable messaging and notification systems demanded research into WebSocket protocols, JWT handling, and persistent connection management on both client and server sides.
- **Mobile Optimization:** Ensuring responsive design and smooth performance on Android devices involved testing screen adaptability, native gesture handling, and consistent state management across sessions.

2.2 Standards and Codes

2.2.1 Backend – Node.js & Express.js

The backend was developed using Node.js and Express.js, allowing scalable routing and middleware customization. RESTful APIs were implemented to handle authentication, role management, property listings, bookings, messaging, and reporting. JWT tokens were used for stateless authentication, and request validation was enforced using middleware such as Zod to maintain API integrity. Code followed REST API best practices including versioning, controller-based routing, and modular file organization.

2.2.2 Database – MongoDB

MongoDB, a NoSQL database, was selected for its flexible document-based structure, scalability, and suitability for handling nested

relationships such as user messages, property reviews, bookings, and contracts. The schema design utilized Mongoose for model definition, data validation, and reference population across related collections like users, properties, messages, and notifications. The database also supports efficient indexing and query performance for search and filtering.

2.2.3 Frontend – Next.js (Web) & React Native (Mobile)

- The web application is built using Next.js, a React framework that provides server-side rendering and API route support. It ensures faster load times, SEO-friendliness, and smooth client-server transitions. Styling is managed using Tailwind CSS and component libraries like shadcn-ui, while authentication state and protected routes are managed using context and hooks.
- The mobile application is developed using React Native, delivering a native-like experience on Android with shared components and logic from the web app. Navigation is handled via React Navigation with role-based UI customization. State is shared using React Context and AsyncStorage to maintain persistent login sessions.

2.2.4 Lahza Authorization & Payment Integration

To address Stripe’s regional limitations, the platform integrates Lahza Authorization—a secure, third-party API that manages OAuth2-based payment and identity authorization. It acts as a middleware between the client and payment services, issuing access and refresh tokens for session management and verifying user identity before allowing sensitive transactions. This ensures safe, scalable payment workflows while complying with local and international regulations. All interactions with Lahza are securely handled through the backend, using HTTPS, JWT, and role-based authorization.

2.2.5 Software Engineering Methodology – Agile

Development was managed using Agile methodology with weekly sprints focused on implementing specific modules—authentication, messaging, contracts, payments, and reporting. Each sprint included task breakdowns, daily standups, testing, and retrospectives. Version control was handled via GitHub and Git, and tasks were tracked using Jira.

2.3 Earlier Coursework

Our academic background in Computer Engineering directly supported the technical foundation of this project:

- **Web Programming:** Taught us modern frontend frameworks and tools such as React and Tailwind CSS, enabling development of modular, responsive user interfaces for both mobile and web platforms.
- **Database Systems:** Covered schema design, indexing, and data modeling—skills applied when building and optimizing MongoDB collections and relationships.
- **Software Engineering:** Introduced us to design documentation, Agile workflows, and UML diagrams that we used for planning and feature scoping.
- **Computer Networks:** Provided understanding of HTTP protocols, WebSockets, and API communication—knowledge we applied in real-time messaging and RESTful API development.
- **Cybersecurity:** Covered JWT, OAuth2, API security, and role-based access—all of which were essential for building a secure, multi-role user platform.

In addition, we engaged in self-learning and hands-on experimentation with Node.js, Express, MongoDB, React Native, and third-party APIs like Lahza to bridge academic theory with real-world software development practices.

Chapter 3: Literature Review

The development of Student Nest stems from a pressing need to provide accessible, localized housing solutions for university students and property owners in Palestine. In many regions, particularly under political and economic constraints like those in occupied Palestine, students face challenges in finding verified, affordable accommodations. Simultaneously, property owners lack centralized, secure platforms to manage listings and communicate with potential tenants. Global real estate platforms are often inaccessible or ill-suited for the local context due to limitations such as payment restrictions, internet instability, and the absence of region-specific features.

Academic literature supports the need for localized digital platforms that address both infrastructural and social barriers. Studies by the World Bank and regional housing reports emphasize the role of digital services in reducing friction between supply and demand in the rental market—especially for transient student populations. Student Nest aims to bridge this gap by offering a full-stack housing platform that facilitates secure property listings, transparent booking workflows, real-time messaging, and role-based access for both students and property owners.

From a technical perspective, the platform architecture is built with Express.js and Node.js for a modular, maintainable backend capable of handling user authentication, property listings, and bookings, contracts, messaging, and reporting. MongoDB is used as the primary database due to its flexibility in handling nested data structures—such as messages within chats or contracts tied to bookings—while enabling fast, document-oriented queries. The use of Mongoose allows for well-defined schemas and data validation across collections like users, properties, bookings, and reviews.

On the frontend, the web interface is developed using Next.js to provide fast server-side rendering, dynamic routing, and SEO-friendly pages. The mobile application, built with React Native, ensures a native-like experience for Android users, making the platform easily accessible to students on the go. Both interfaces share a unified design system using Tailwind CSS and component libraries like shadcn-ui to maintain consistent UI/UX across devices.

For secure payment handling, Student Nest integrates Lahza Authorization—a regional OAuth 2.0-based gateway that allows users to authenticate and authorize financial transactions safely. Unlike Stripe or other globally dominant platforms, Lahza is designed to comply with the regulatory and infrastructural conditions of the Palestinian context. This aligns with Digital Financial Services (DFS) research that recommends region-specific APIs and secure token flows to enable digital transactions in politically restricted environments.

Real-time communication is another key pillar of the platform. Student-owner interactions are facilitated through in-app messaging using WebSocket technology, enabling low-latency chat capabilities. This decision is grounded in research by the Mozilla Developer Network and industry standards on WebSocket-based systems, which emphasize the importance of seamless real-time communication in user-centric applications.

By synthesizing academic research and adapting proven industry tools to the local context, Student Nest establishes itself as a culturally aware, technically sound housing platform. It not only addresses infrastructural and usability gaps in Palestine's housing ecosystem but also lays the groundwork for future scalability and social impact through ethical software design and user-centered engineering.

Chapter 4: Methodology

4.1 Tools, Methods, and Programming Languages

4.1.1 Tools

- **Visual Studio Code:** Used as the primary IDE for writing and debugging both backend and frontend code.
- **Android Studio:** Employed for testing and running the React Native mobile app on emulators and physical devices.
- **Atlas mongoDB Tools:** Used for managing and querying the database.
- **GitHub:** Used for version control, collaboration, and managing code through branches and pull requests.
- **Notion:** Helped with documentation, task management, and sprint tracking.
- **Figma:** Used for designing wireframes, user interfaces, and flow diagrams before implementation.

4.1.2 Programming Languages & Frameworks

Frontend Development:

- **Web App:** Built using **NextJS**, a JavaScript framework for building responsive, component-based UIs. Styled using **Tailwind CSS** and UI components.
- **Mobile App:** Developed using **React Native**, which allows us to write cross-platform mobile apps using JavaScript and native modules for Android support.

Backend Development:

- **ExpressJS:** The backend framework used to create RESTful APIs, handle business logic, and manage user roles and authentication. Express REST Framework helped structure and serialize the APIs.
- **JWT Authentication:** Used for secure user login and session management, integrated with MongoDB.

Third-Party Integration:

- **Lahza Authorization:** Integrated for secure user authentication and regional payment access via OAuth 2.0 flow.

4.1.3 Database

- **MongoDB:** A robust and scalable relational database used to manage structured data including users, services, categories, gigs, orders, payments, messages, and reviews.
- The database schema is normalized and includes foreign key constraints to preserve data integrity.
- **Mongo Atlas:** Used to visually manage and monitor the database during development and testing.

Chapter 5: Results and Discussion

This chapter presents the results of the Student Nest project by highlighting its core features, accompanied by relevant screenshots and detailed explanations. The system was developed with distinct interfaces for web and mobile platforms, each tailored to optimize user experience and performance. It incorporates role-based access control, offering differentiated functionalities for students and property owners.

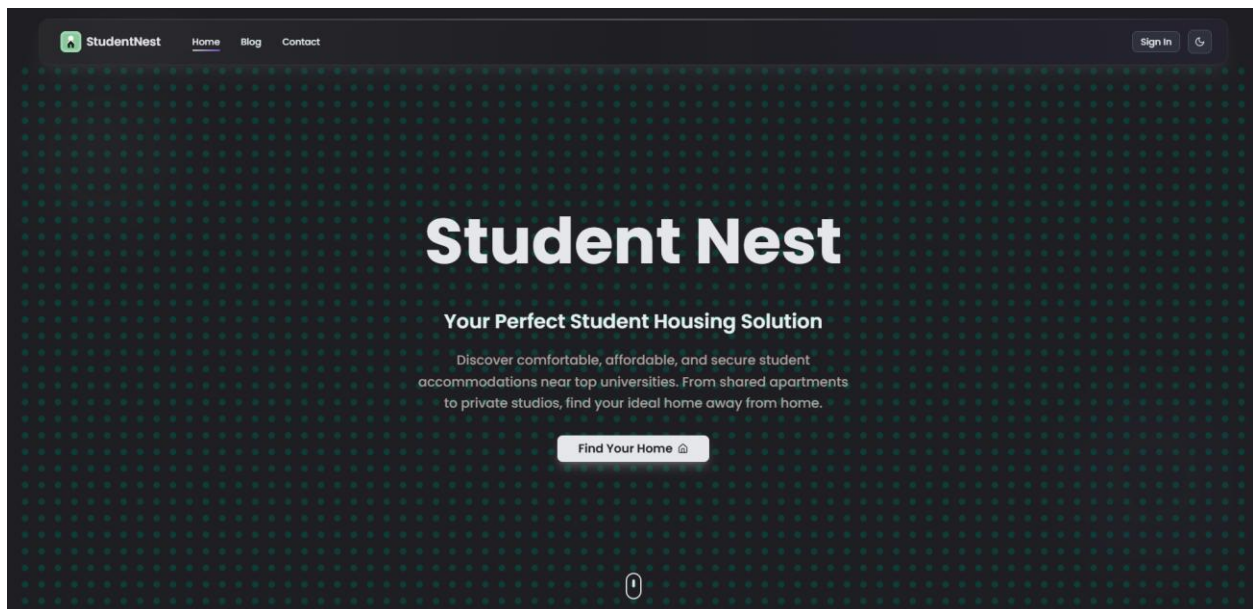
Throughout the development process, certain features required divergent design and technical approaches across platforms. In such cases, the implementation that offered the best usability, performance, or maintainability was selected. This chapter details both the shared features across roles and

platforms, as well as role-specific functionalities, providing a comprehensive overview of the platform's structure, user flow, and technical execution.

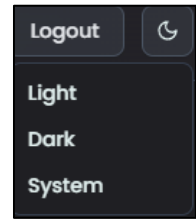
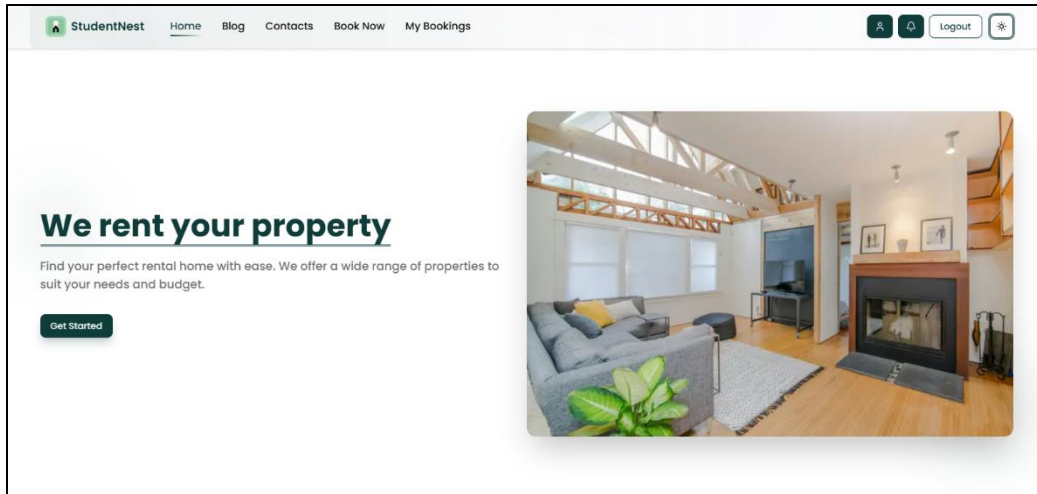
5.1 Mobile Application

5.1.1 Common Features Among Users

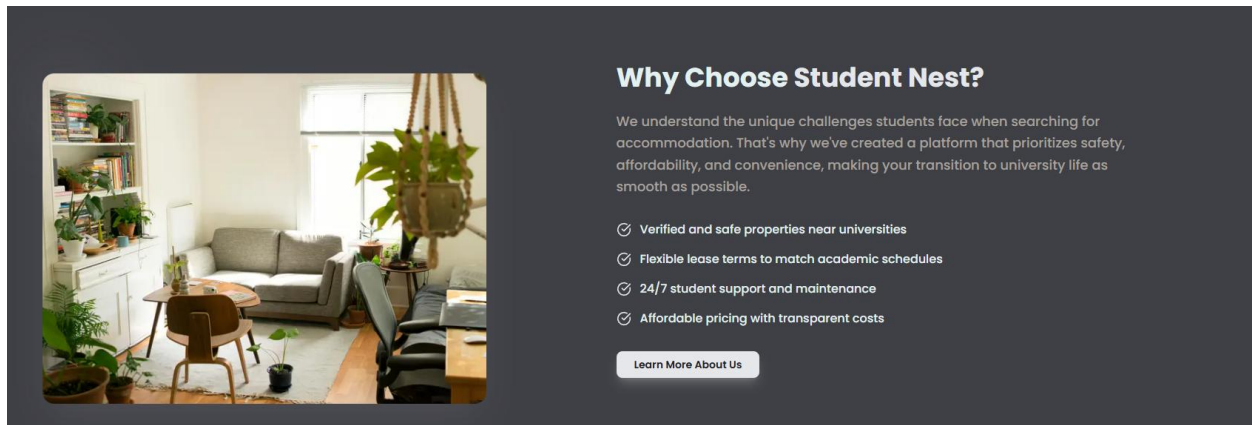
These features are available to all users regardless of their role (student or owner), and they help create a unified and intuitive experience within the mobile app.



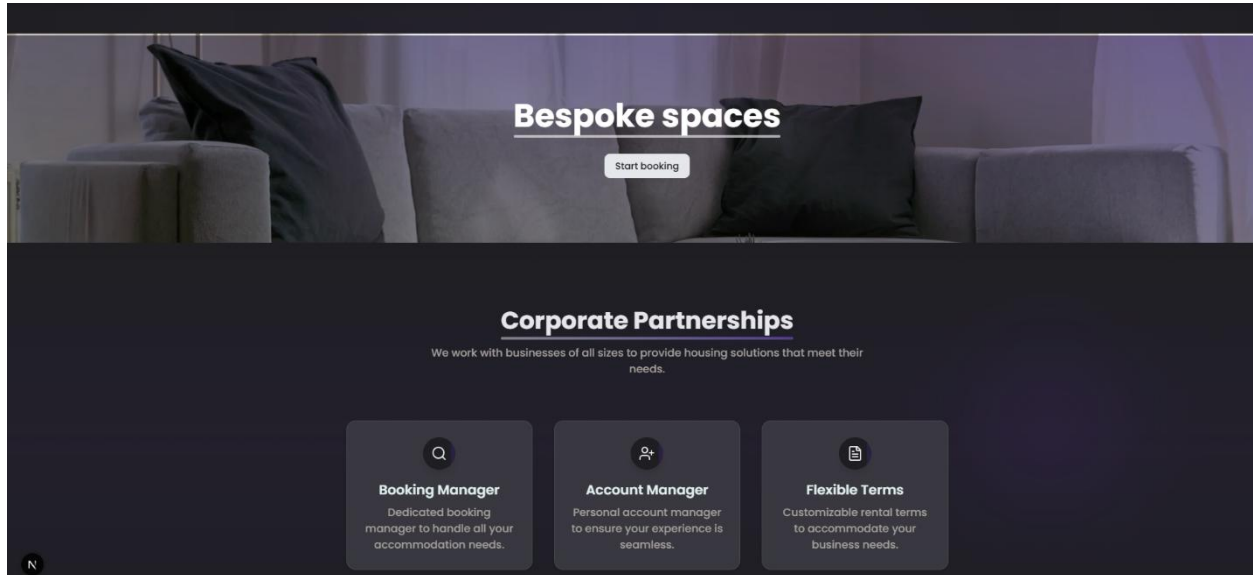
The landing page of the StudentNest platform, featuring a modern and user-friendly interface. The page includes a navigation bar with links to Home, Blog, Contacts, Booking, and My Bookings, along with login/logout controls and the theme selection button. The main section highlights a welcoming message and a call-to-action button that guides the user to sign in or sign up alongside a featured rental property image, emphasizing ease of use and a broad property selection for students.



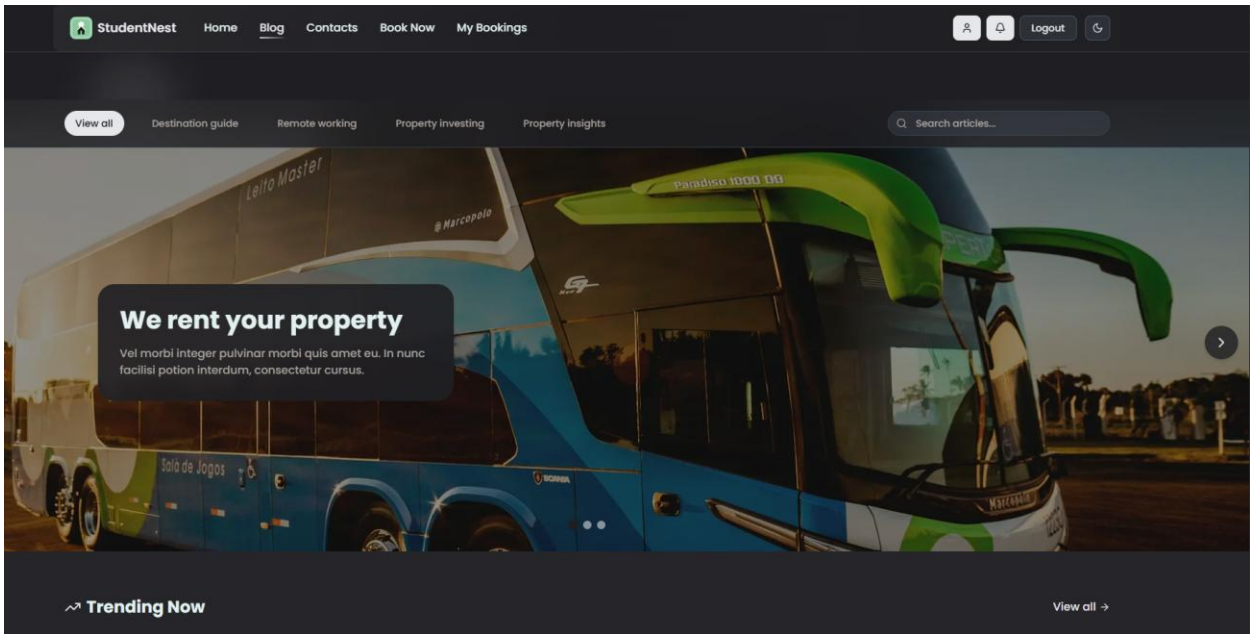
Same homepage and same features but with the light theme applied. In which there is a “system” option that allows the application to adapt based on the user’s main theme.



This section of the StudentNest platform emphasizes adaptability with the message “Why choose Student Nest.” Featuring a cozy, well-furnished living space, it highlights the platform’s variety of property options. Users can click the Learn More button to explore available listings and book accommodations that match their lifestyle and duration preferences.



This section of the platform highlights customizable housing options under the theme “Bespoke spaces.” Users are invited to start booking directly, while the Corporate Partnerships section below outlines services for business clients, including a dedicated Booking Manager, a Personal Account Manager, and Flexible Terms.



The Blog page provides students valuable content on property rentals, remote living, investment tips, and destination guides. With a clean, user-friendly layout, visitors can explore trending articles, learn about property ownership opportunities, and gain insights to make informed booking decisions.

Everything You Need for Student Life

Our comprehensive platform offers all the tools and services you need to find, secure, and enjoy your perfect student accommodation.

- Smart Search**
Advanced filters to find properties that match your exact needs and budget.
- Verified Properties**
All listings are verified for safety, quality, and legitimacy.
- Flexible Booking**
Book for semesters, academic years, or custom periods.
- 24/7 Support**
Round-the-clock assistance for all your housing needs.

StudentNest offers a seamless rental experience with features tailored to modern living. Users benefit from **flexible rental periods**, allowing stays from a few days to several months. All properties are **move-in ready**, fully furnished for immediate comfort. **High-speed Wi-Fi** ensures uninterrupted connectivity, and **24/7 support** guarantees peace of mind with around-the-clock assistance from a dedicated team.

Latest Articles

Relocation

Turpis elit in dictum eget eget

Egestas viverra nunc vestibulum et placerat condimentum tincidunt commodo sem.

🕒 1 min read

Remote working

Feugiat gravida sed sit lacus sagittis

Ultrices lectus, purus praesent volutpat vitae et consequat tellus commodo magna.

🕒 3 min read

Guest experience

Aliquam id ullamcorper eu

Egestas viverra nunc vestibulum et placerat tincidunt commodo sem. Risus, tincidunt ultrices in dictum vitae.

🕒 2 min read

Guest experience

Placerat purus habitasse malesuada

Egestas viverra nunc vestibulum et placerat vitae tincidunt commodo sem.

Apartment lifestyle

Proin tristique ipsum nulla cursus

Ultrices lectus, purus praesent volutpat vitae et consequat tellus.

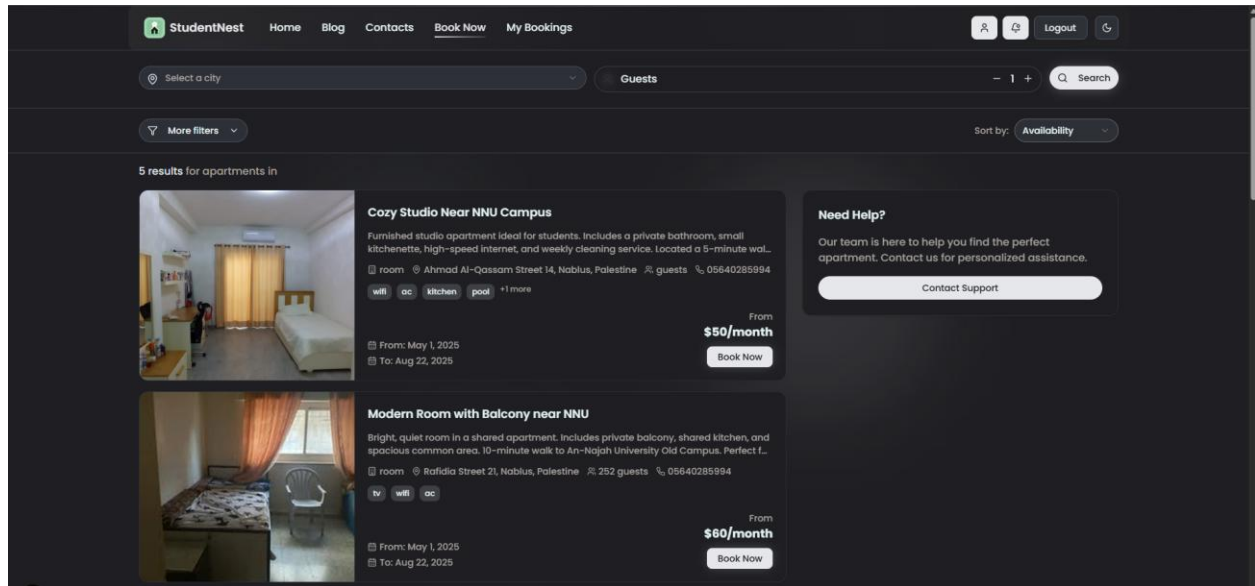
Guest experience

Vulputate id sem felis, at at eget nulla

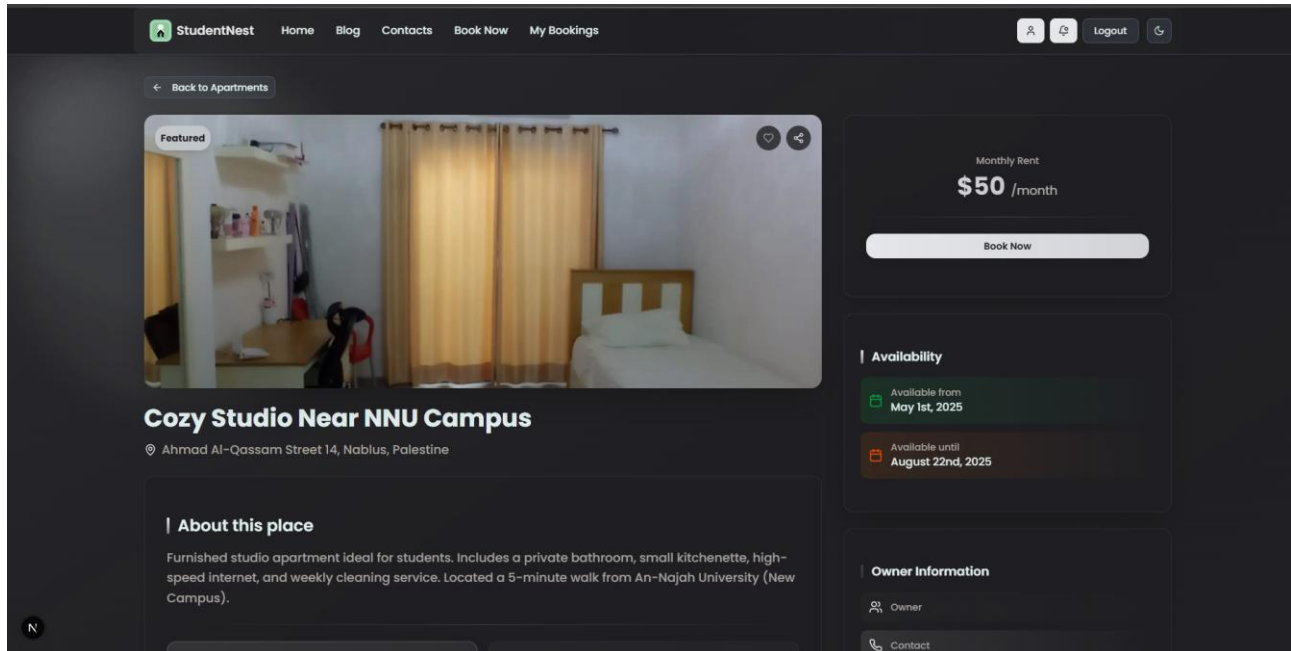
Vitae enim morbi vitae neque pellentesque faucibus.

🕒 5 min read

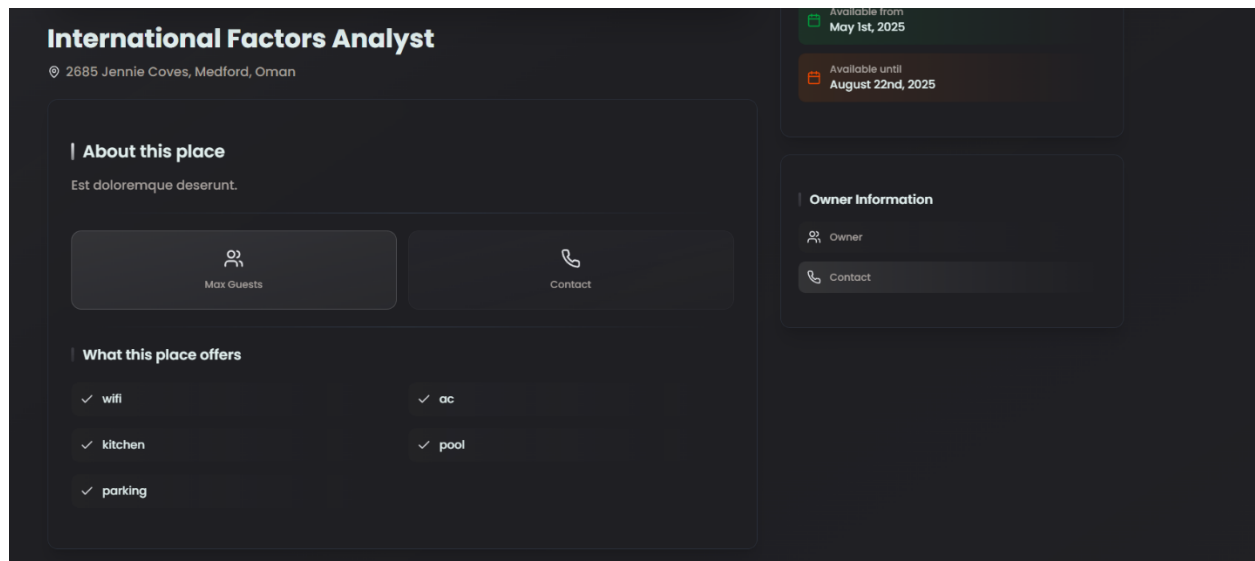
The Latest Articles section offers curated content to guide users through every step of the rental journey. From relocation tips and apartment lifestyle advice to negotiating rental prices and enhancing guest experiences, these insightful articles help users make confident, well-informed decisions when choosing their ideal property. Each post is categorized and time-stamped for quick and easy access.



The Booking Page empowers users to effortlessly search, filter, and compare available student apartments. With intuitive controls like city selection, guest count, and advanced filters for amenities and availability, users can tailor their search to match personal preferences and needs. Each listing is clearly presented with key details—price, location, amenities, and availability—alongside a direct "Book Now" option for streamlined decision-making. A built-in support module ensures users have access to real-time help, making the booking process smooth, responsive.



The Apartment Details page offers a complete snapshot of each property, helping users make informed booking decisions with confidence. Users can view images, key details such as location, rent, and availability dates, and a full description of the apartment. This page also displays essential amenities, owner contact information, and a secure “Book Now” button. Through seamless integration with the Lahza Secure API, users can safely complete payments using their credit cards.



This section provides a clear breakdown of the apartment’s features and direct communication options for prospective tenants. Users can easily review the full list of amenities—such as Wi-Fi, air conditioning, kitchen, pool, and parking—ensuring the property aligns with their lifestyle needs. Key details like maximum guest capacity and owner contact information are displayed, allowing users to inquire further or clarify details before booking.

Apartments for rent in Palestine

Finding the perfect apartment near Najah National University in Nablus has never been easier. Our platform offers a wide selection of quality accommodations tailored to students and professionals alike. Each neighborhood in Nablus has its own unique charm and character, providing diverse living experiences throughout the city.

Whether you're looking for a cozy studio apartment or a spacious multi-bedroom unit, our listings cover a range of options to suit different needs and budgets. Many of our properties are within walking distance to Najah National University, making them ideal for students who want to minimize commute time.

All our apartments come fully furnished with modern amenities, reliable WiFi, and essential appliances. We prioritize comfort, convenience, and security to ensure you have a pleasant living experience during your stay in Nablus. Browse our listings today and find your perfect home away from home.

Frequently Asked Questions

What amenities are included in the apartments? ▾

How close are the apartments to Najah National University? ▾

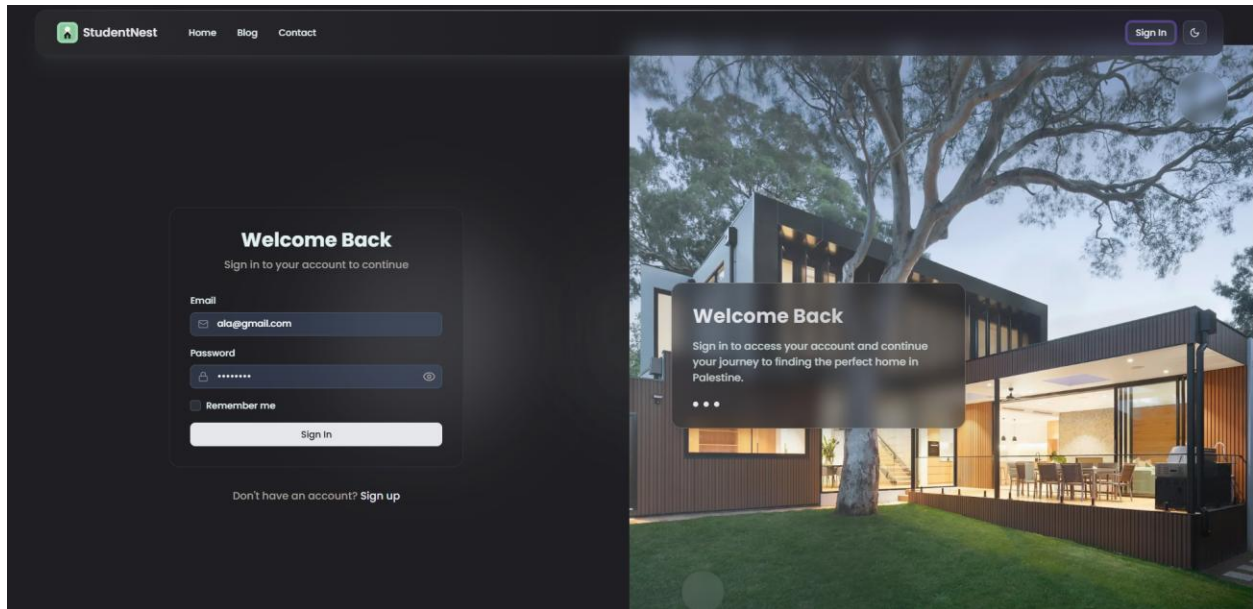
What is the booking process? ▾

Is there a security deposit required? ▾

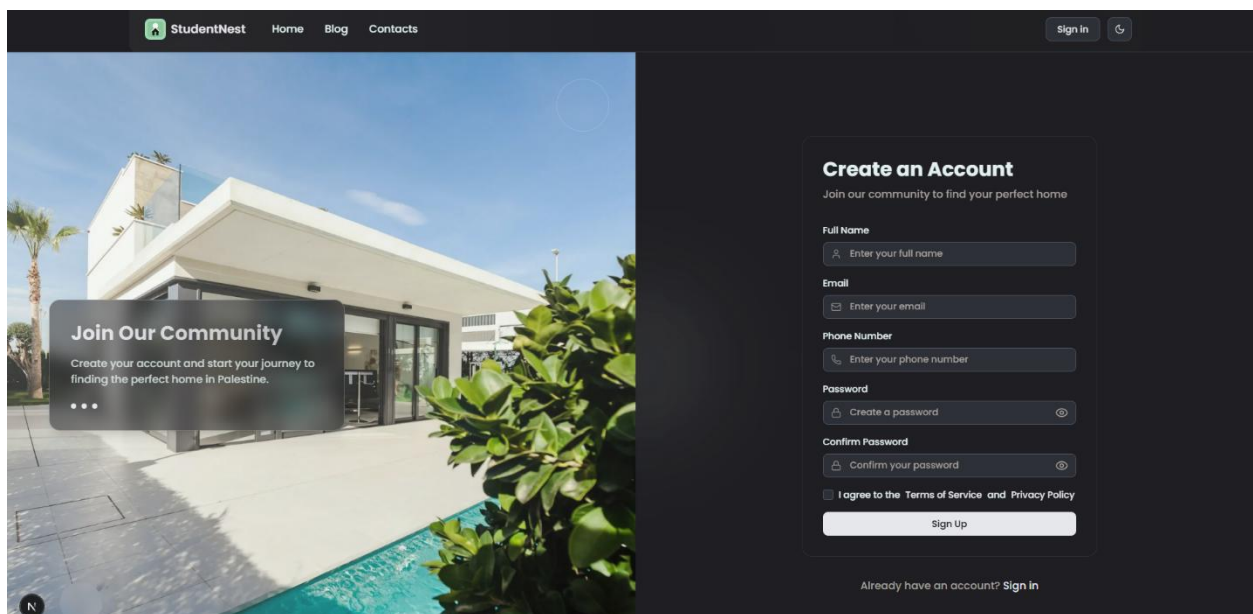
Can I book for a semester or academic year? ▾

Are utilities included in the rent? ▾

This section outlines the convenience of nearby locations, modern amenities, and flexible rental terms. The accompanying FAQ section addresses common concerns such as booking procedures, included utilities, security deposits, and rental durations—ensuring users can make informed decisions with clarity and confidence.



This login page offers a secure and user-friendly gateway for students and owners to access their StudentNest accounts. It provides features like a clean sign-in form with options for email/password login.



This registration page invites new users to join the StudentNest community and begin their search for the perfect home in Palestine. It features a straightforward sign-up form that collects essential information such as full name, email, phone number, and password, along with agreement to the platform's Terms of Service and Privacy Policy.

Add New Property

Fill in the details below to add a new property listing.

Property Title Cozy Apartment near University	Description A cozy and modern apartment, ideal for students...	
Property Type Select property type	Owner Name Ahmad Khalid	Owner Phone +970 59 123 4567
Monthly Rent (USD) 0	Available From Start date	Maximum Guests
Address 123 University St, Ramallah	Available To End date	
City	Country	
Amenities		
<input type="checkbox"/> WiFi	<input type="checkbox"/> Parking	
<input type="checkbox"/> Air Conditioning	<input type="checkbox"/> TV	
<input type="checkbox"/> Kitchen	<input type="checkbox"/> Gym	
<input type="checkbox"/> Swimming Pool	<input type="checkbox"/> Security System	
<input type="checkbox"/> Balcony		
Property Images		

This interface allows property owners to easily add a new apartment listing to the platform. It includes all essential fields such as title, description, rent, and address to ensure the listing is complete and informative. Owners can specify the type of property, available dates, and maximum number of guests, ensuring accurate availability for potential renters. Additionally, the form supports contact details and a comprehensive list of amenities like Wi-Fi, air conditioning, kitchen, and parking. Owners can also upload property images to enhance the listing's appeal.

Booking Requests
Manage booking requests from students for your properties

Search by student or property... All Statuses All Properties

Property	Student	Check-in / Check-out	Status	Actions
Lead Mobility Director	t test123	Mar 20, 2024 to Mar 25, 2024	Confirmed	[Action]
Dynamic Solutions Planner	t test123	Mar 20, 2024 to Mar 25, 2024	Confirmed	[Action]
Lead Mobility Director	E Elena30	May 25, 2025 to Jul 26, 2025	Confirmed	[Action]
Dynamic Solutions Planner	E Elena30	May 13, 2025 to Jul 17, 2025	Rejected	[Action]
Dynamic Solutions Planner	t test123	Jun 11, 2025 to Jun 11, 2026	Pending	[Action] [Action] [Action]

This Booking Requests interface enables property owners to efficiently manage all reservation requests from students. Owners can view key booking details, such as property name, student information, check-in/check-out dates, and current booking status (Confirmed, Pending, or Rejected), in a structured table. The interface offers filters by booking status and property, as well as a search function to quickly locate specific requests. Action buttons allow owners to approve or decline requests directly from the panel, streamlining the decision-making process.

Edit Property

Update your property details below.

Property Title Dynamic Solutions Planner	Description Perspiciatis quas nostrum corporis dolorem amet ullam necessitatibus consequuntur.	
Property Type Condominium	Owner Name Ahmad Khalid	Owner Phone +970 59 123 4567
Monthly Rent (USD) 628	Available From May 13, 2025	Maximum Guests 79
Address 20246 Schuster Mill	Available To Jul 17, 2025	
City Visalia	Country Nepal	
Amenities		
<input checked="" type="checkbox"/> WiFi	<input checked="" type="checkbox"/> Parking	
<input checked="" type="checkbox"/> Air Conditioning	<input checked="" type="checkbox"/> TV	
<input type="checkbox"/> Kitchen	<input type="checkbox"/> Gym	
<input type="checkbox"/> Swimming Pool	<input type="checkbox"/> Security System	
<input type="checkbox"/> Balcony		

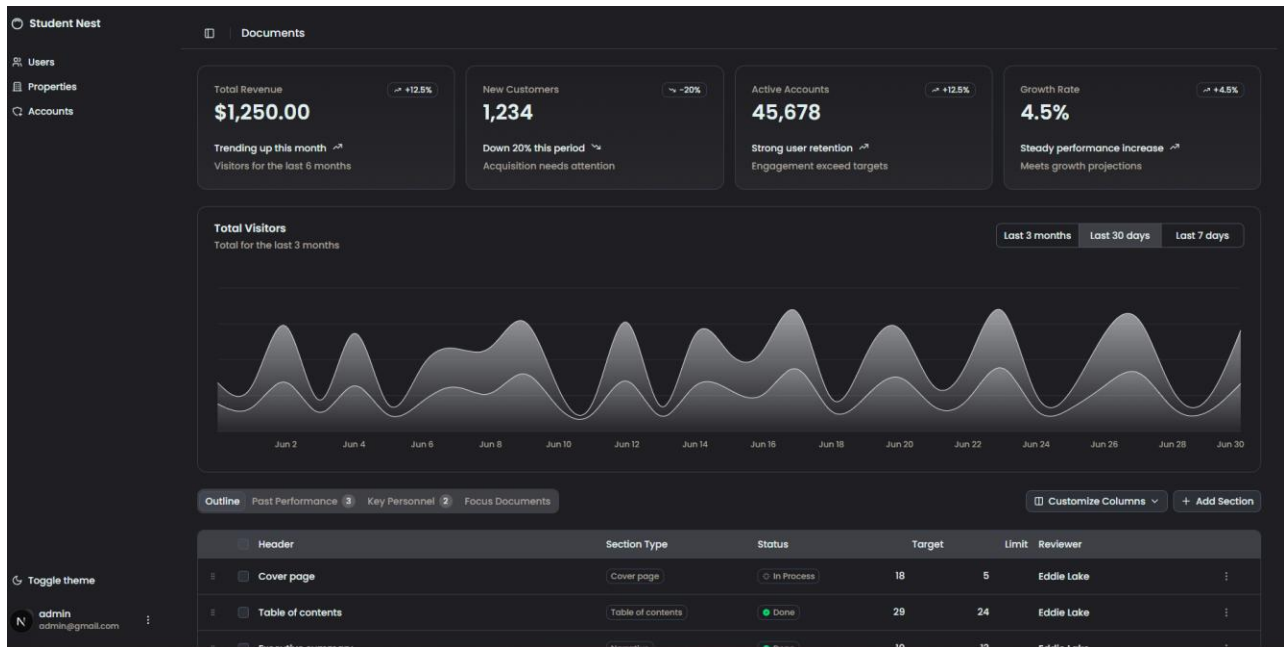
Property Images

Are you sure?

This action cannot be undone. This will permanently delete the property and remove it from our servers.

This Edit Property interface allows property owners to update existing property listings. Owners can modify all key property details, such as title, type, rent, address, availability dates, capacity, contact info, and amenities. Property images can also be updated or removed.

Additionally, if an owner decides to remove the listing entirely, a confirmation dialog ensures deliberate action before deletion. This dialog clearly warns that the deletion is permanent and irreversible, helping prevent accidental loss of data. This interface gives full control over managing property information with both flexibility and security.



This image presents the admin dashboard of the StudentNest platform, offering a comprehensive analytics view. Key metrics include total revenue, new customers, active accounts, and growth rate, with performance indicators and trends. A visitor graph visualizes user activity over time, while detailed document tracking at the bottom allows admins to monitor review status, section types, and progress targets.

Users Management

View and manage all users in your application

Users
Showing 8 of 8 users

Search users... Student

ID	Username	Email	Phone Number	Actions
681a048ec95d02acc4aed08	Elena30	Hadilrshaid8722@gmail.com	3444903319	
682a10e41fbab8f42f842b2e	mohamad	mohamad.kittani0502@gmail.com	0594052864	
682a1b7efbab8f42f842b6e	owner	owner1@gmail.com	0594052864	
6830fc184c9a6ac7c67de599	Nettie.Daugherty	your.email+fakedata9731@gmail.com	9749392030	
68334fbd7bf60ed1b307e67	mohamad	mohamad.kittani05021@gmail.com	0594052864	
68379c4b395ffad8e646aa48	helloworld	ala@gmail.com	05957261	
683c475620bbe0a302e8583d	osama	ossalah2002@gmail.com	0595726195	
683c54419fb8bde7698cea53	yay	mree@gmail.com	0595726195	

This interface allows administrators to view and manage all registered users within the StudentNest application. The panel displays essential user details including unique ID, username, email, and phone number. It includes search and filter capabilities (e.g., by user role like "Student") and provides deletion actions for user management.

Properties Management
View and manage all property listings in your application

Properties
Showing 4 of 4 properties

Search properties...

Property	Type	Price	Address	Owner	Actions
International Factors Analyst Est doloremque deserunt.	Room	\$133/month	2685 Jennie Coves	testuser 1234567890	
Legacy Communications Supervisor Deserunt laborum sed vel voluptatem...	Room	\$482/month	981 Kessler Port	testuser 1234567890	
Dynamic Solutions Planner Perspiciatis quos nostrum corporis ...	Condo	\$628/month	20246 Schuster Mill	Owner 0566520089	
Lead Mobility Director Possimus minima iusto autem enim ac...	Room	\$74/month	8394 Oberbrunner Avenue	Owner 0566520089	

This interface provides administrators with full visibility and control over all property listings in the StudentNest platform. It displays key property details including name, type (e.g., Room or Condo), monthly price, physical address, and associated owner information. The table includes a search bar to quickly locate specific listings, and each entry features an action icon for deleting properties as needed.

Account Management
Create and manage owner and admin accounts

Create New Account

Add a new owner or admin account to the system

Admin Account Owner Account

Username johndoe **Email Address** john.doe@example.com

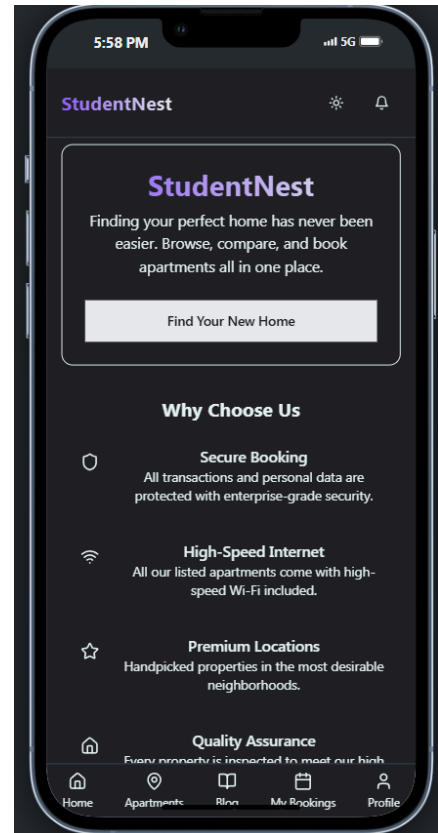
Phone Number +1 (555) 123-4567 **Password**

Confirm Password

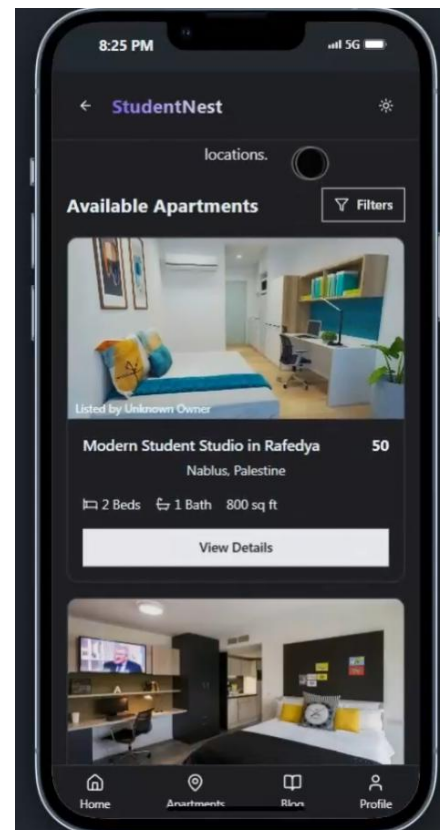
Create Account

This interface enables platform administrators to create new admin or owner accounts within the StudentNest system. Users can toggle between creating an Admin Account or an Owner Account. The form collects essential information including username, email address, phone number, password, and password confirmation. This feature supports secure onboarding of new management personnel and ensures proper role assignment for system access and control.

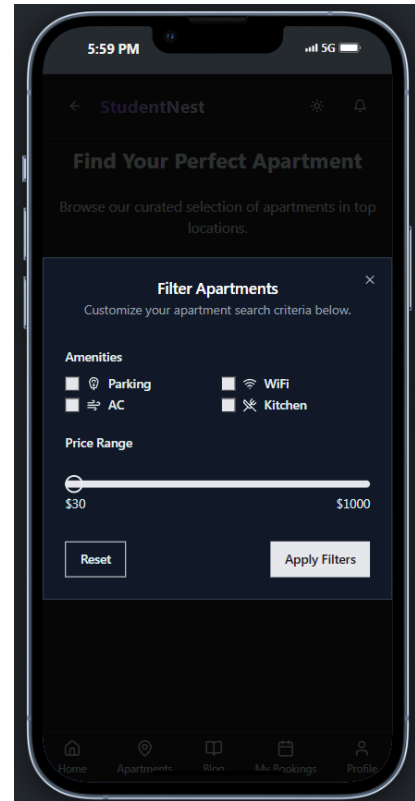
This mobile interface of StudentNest offers users a streamlined experience to discover and explore property listings. The top section features a clear call-to-action encouraging users to begin their home search, while the “Recommended For You” area displays curated listings with key details like location, price, size, and amenities. A prominent “View Details” button allows users to dive deeper into each listing. With intuitive navigation tabs for Home, Apartments, Blog, and Profile at the bottom.



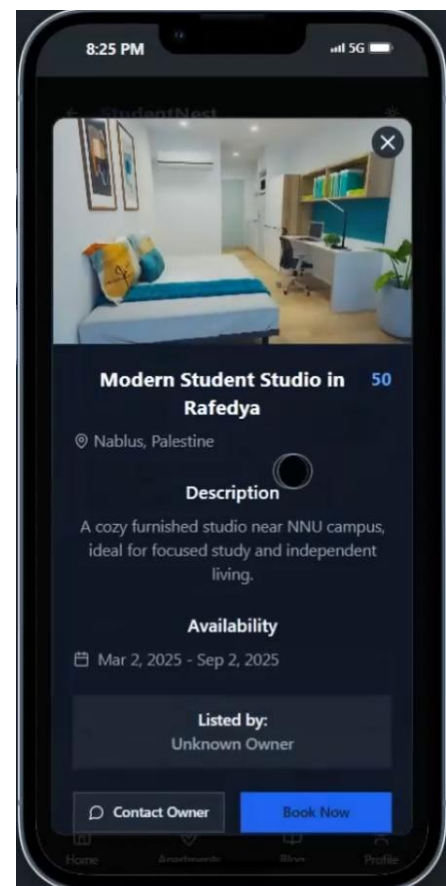
This screen provides a centralized mobile interface for users to browse all available properties on the StudentNest platform. It features a clean layout with a filter button to refine search results based on user preferences. While the current message indicates no listings are available, it shows only the apartments for the current logged in owner.



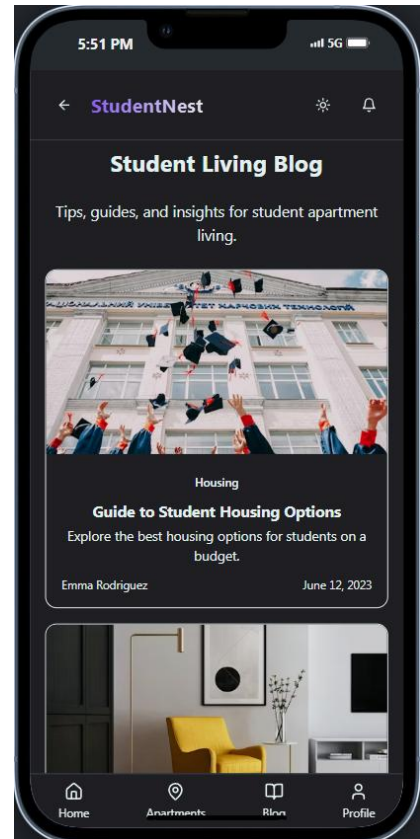
This filter interface facilitates users to tailor their apartment search based on personal preferences. Users can adjust parameters such as amenities like WiFi or parking availability and prices. With clearly segmented sections and intuitive button controls. Options to reset or apply filters allow for flexible and efficient navigation.



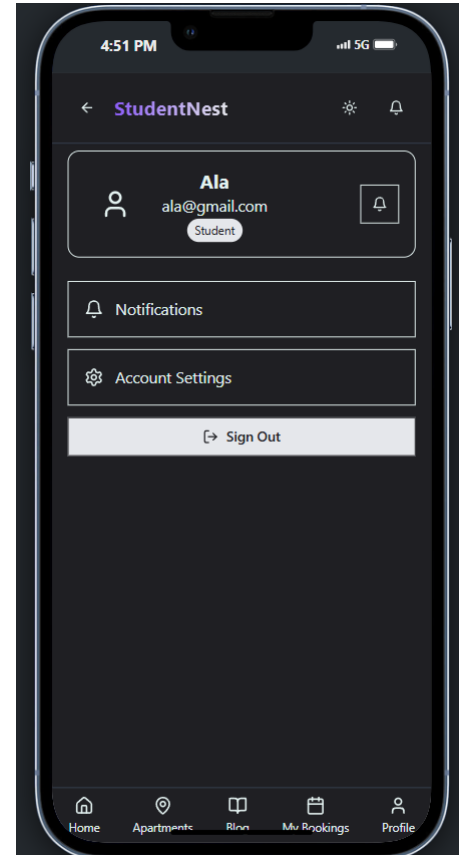
This interface allows the user view all the details about a property they wish to rent in which it shows all the amenities it provides like wifi, parking, ac, and more. It also provides an option for the user to book the property in which it creates a pending booking for the owner to approve and sends a notification for the owner.



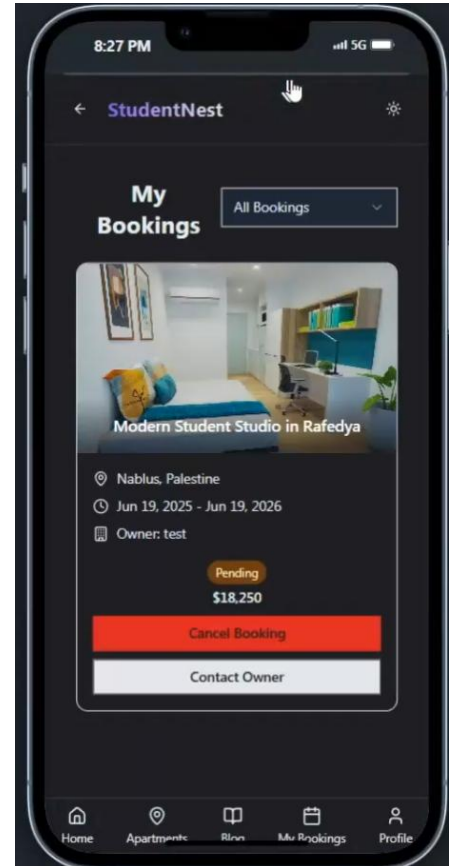
The blog interface offers users valuable content focused on student apartment living. From housing guides to decoration tips, the section provides curated articles designed to inform and inspire students navigating their rental journey. Each blog post includes a featured image, title, author name, and publication date, making it easy for users to explore helpful insights and lifestyle advice.



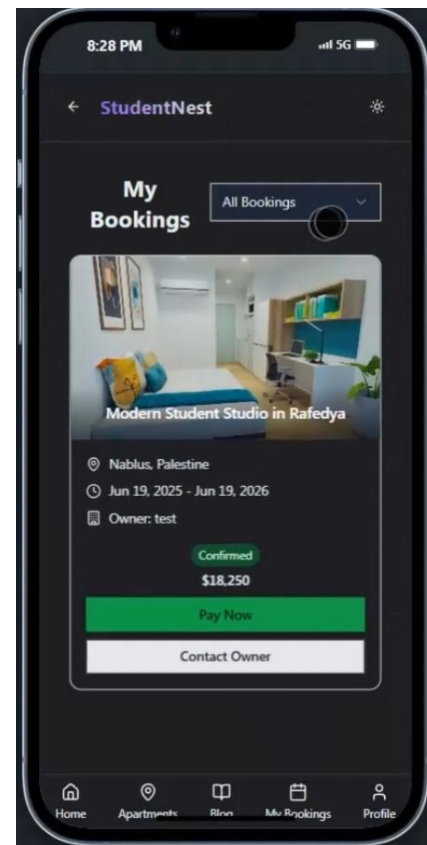
This profile section offers a quick summary of the owner's identity, email address, and account role. It features streamlined navigation buttons to key areas such as the Owner Dashboard and Property Listings. Additional options include accessing notifications, adjusting account settings, and securely signing out.



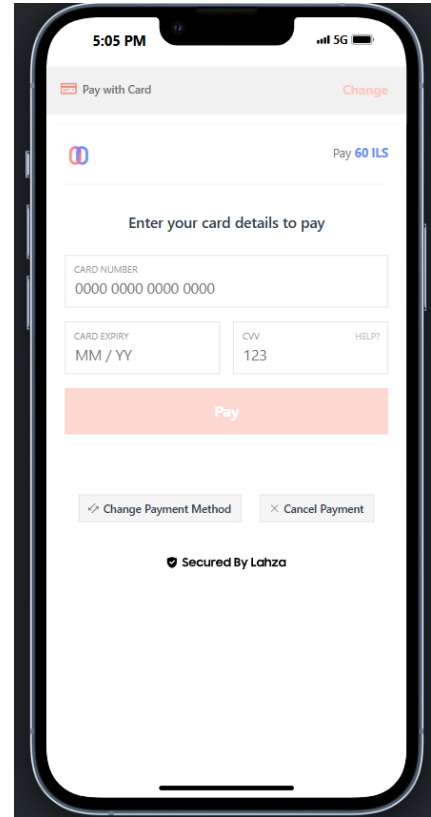
This interface allows the student to cancel his pending booking he just created so that only if the booking is still pending then he cancel it and it gets deleted totally or the student could press on the “Contact Owner” button to chat with the owner and possibly get more details about the property.



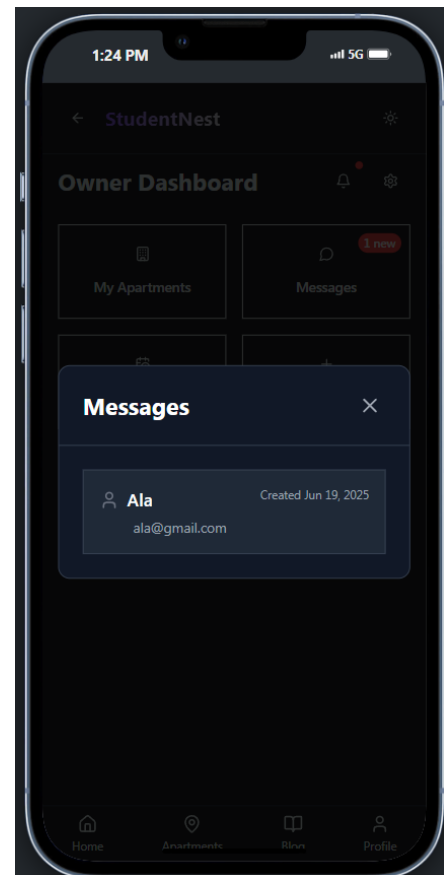
In case the owner approved the booking request then a notification is sent to the student and then he can visit his “My Bookings” page and browse his confirmed bookings using the filter and then he can press the “Pay Now” button to get rerouted to Lahza payment api to complete his payment.



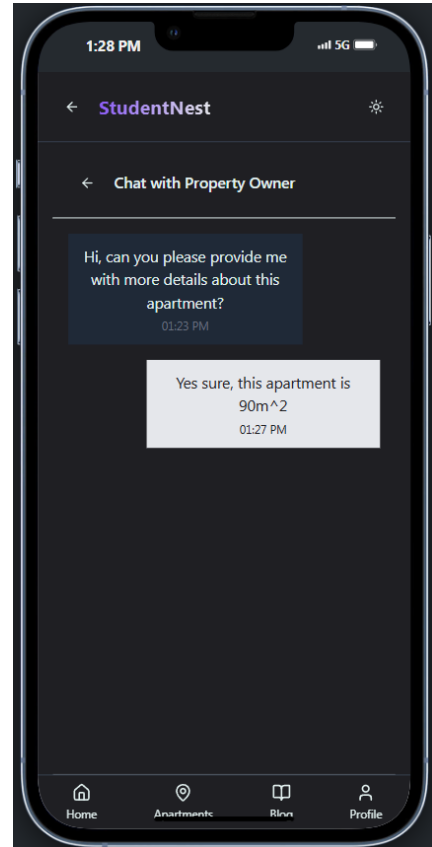
Lahza payment api where the user could enter his credit card information securely and all his payment process is handled securely and each party gets its cut from the payment automatically.



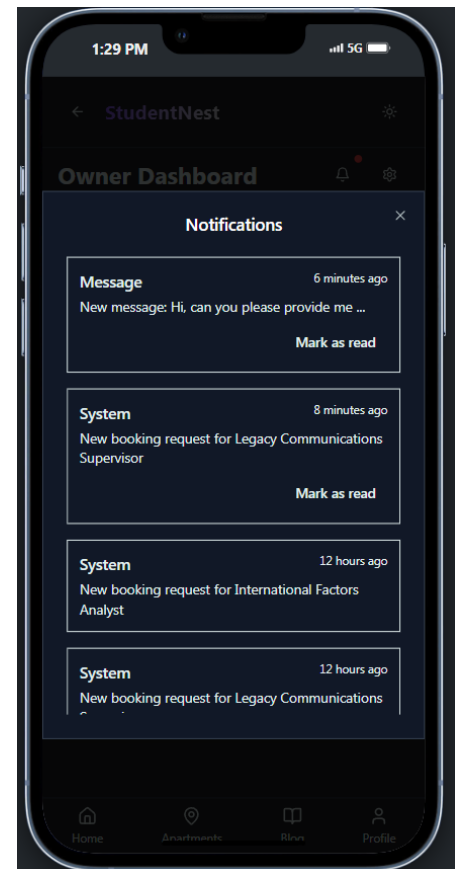
After a student contacts an owner, the owner then can browse their messages and see the new and unread ones and he can click on any message to continue chatting or mark it as seen.



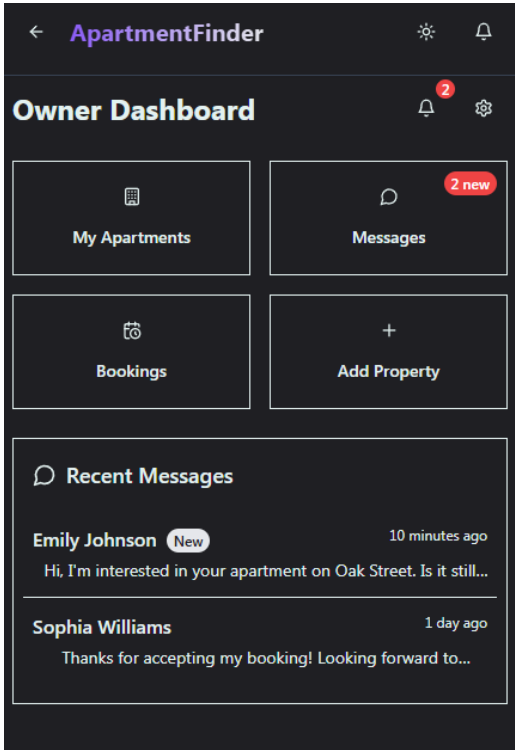
The chatting process between the student and owner about any topic like asking for more details about any property and timestamps are added with each message.



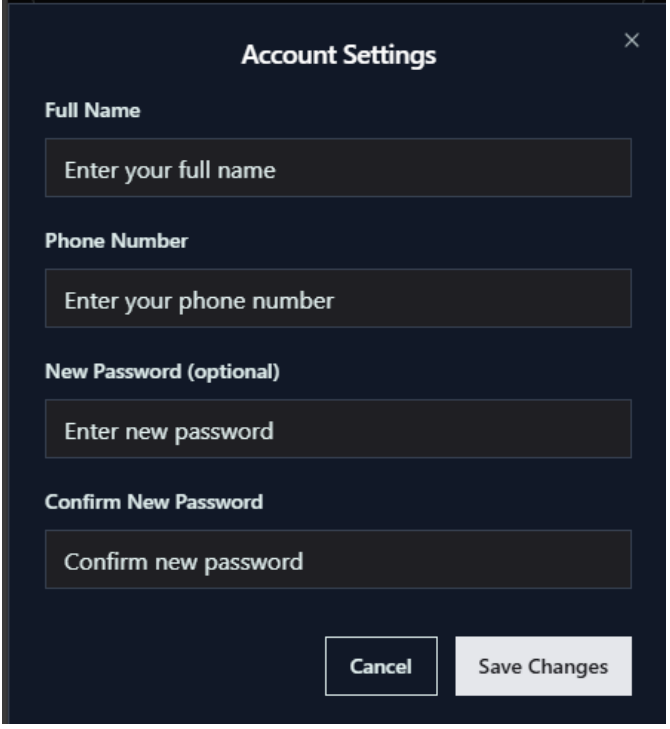
This pop up provides all notifications for a user like a new booking has been created or a new message was received or even a booking was confirmed and you can pay for it now.



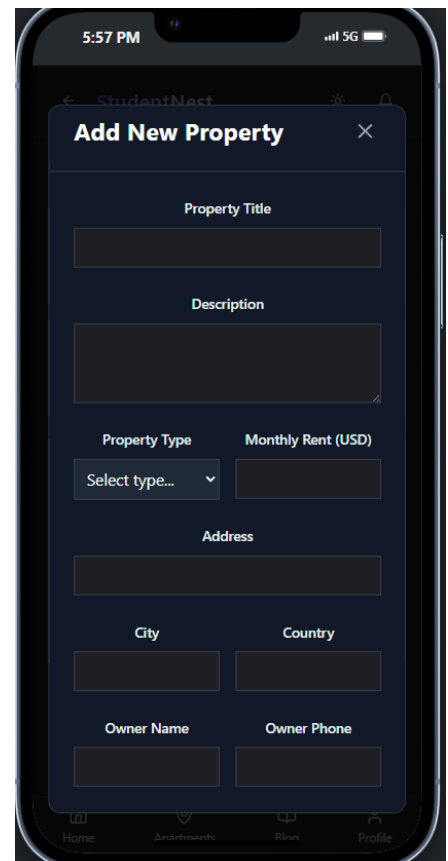
The Owner Dashboard acts as the central hub for managing apartment listings and tenant interactions. Property owners can add new properties, review current bookings, and respond to incoming messages—all from a single screen. A live notification badge highlights unread messages, while a message preview section gives real-time insight into tenant inquiries, promoting fast and informed responses.



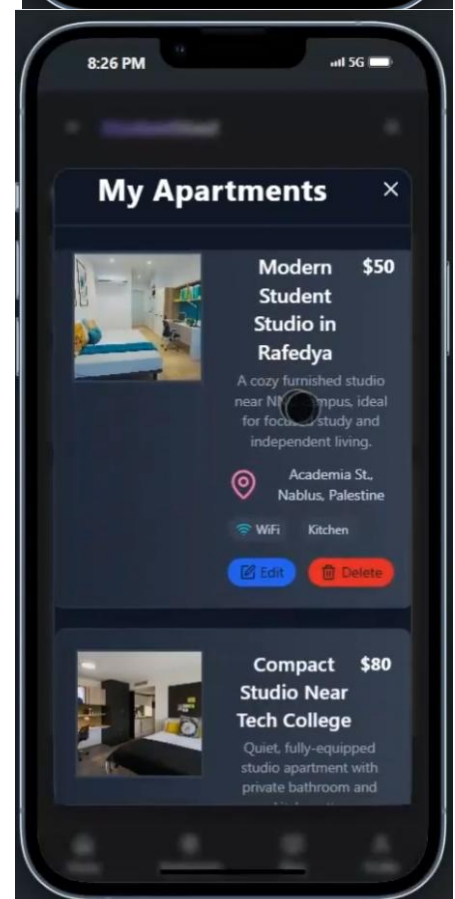
This interface enables property users to update their personal information securely. Fields are provided for full name, phone number, and password changes. A clear two-step password confirmation ensures account security. With prominent Cancel and Save Changes buttons.



This mobile interface allows to add new property listings directly to the StudentNest platform. The form includes fields for essential details such as property title, description, type, monthly rent, full address, and the owner's name and phone number. Designed for mobile use, this feature ensures property data can be submitted quickly and efficiently.



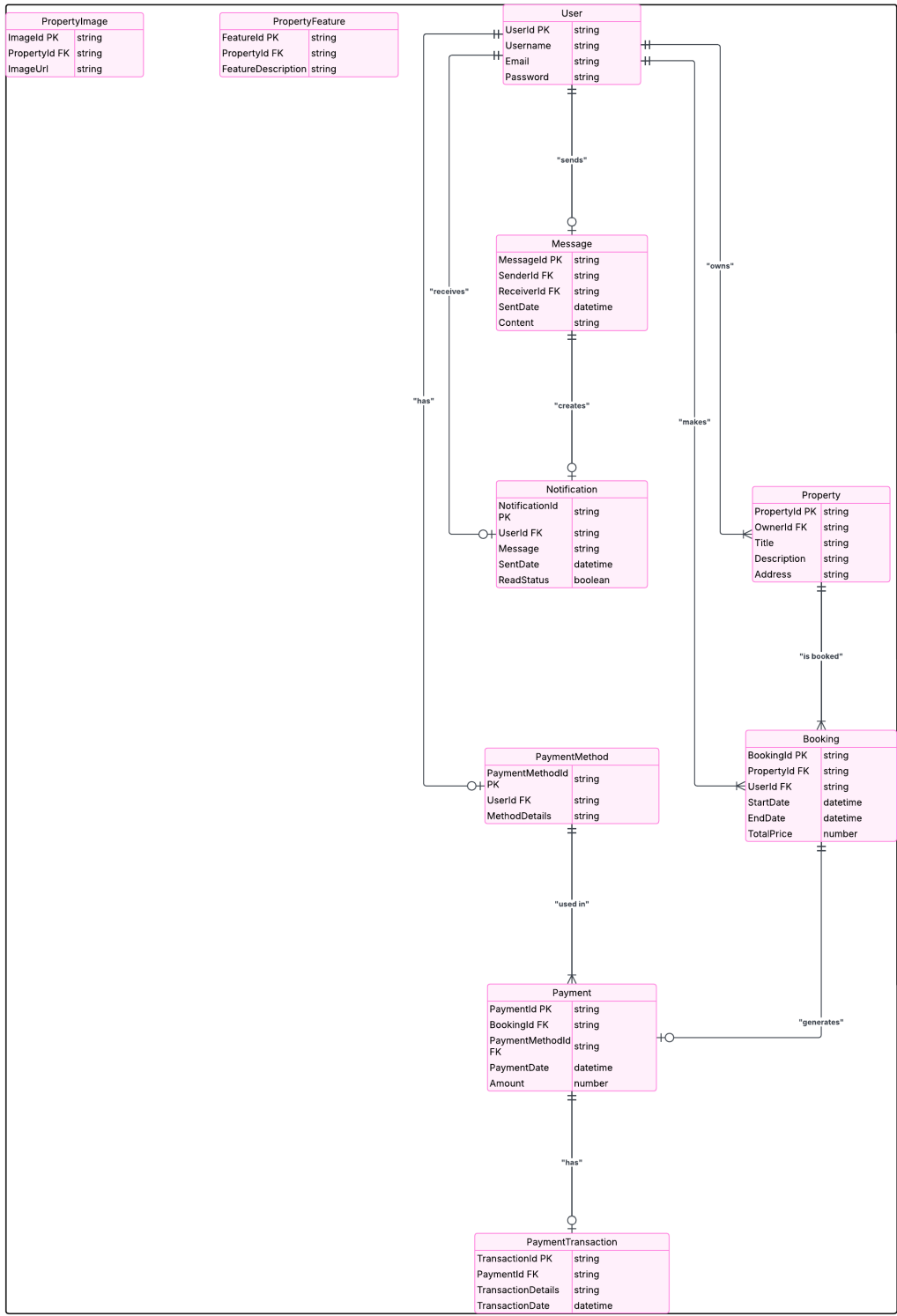
This interface provides property owners with a comprehensive overview of all their listed rentals on StudentNest. Owners can quickly scan key details for each property, including names, pricing, locations, and amenities, while maintaining full control through intuitive Edit and Delete options. The clean, mobile-friendly design enables efficient property management, allowing owners to update listings or remove outdated ones with just a few taps.



This page allows property owners to monitor and manage all their rental bookings in one place. Owners can view bookings in three statuses, pending, confirmed or cancelled. Each booking displays essential details including tenant contact info, stay dates, payment amount, and booking date. For pending requests, owners can instantly accept or decline reservations with one click.

The screenshot displays a vertical list of three booking cards on a dark background. Each card contains the following information:

- Lead Mobility Director** (Confirmed): Amount \$4,588, booked May 31, 2025. Contact: Hadilrshaid8722@gmail.com, 3444903319. Stay dates: May 25, 2025 - Jul 26, 2025. Action: Contact Student.
- Dynamic Solutions Planner** (Cancelled): Amount \$40,820, booked May 31, 2025. Contact: Hadilrshaid8722@gmail.com, 3444903319. Stay dates: May 13, 2025 - Jul 17, 2025. Action: Contact Student.
- Dynamic Solutions Planner** (Pending): Amount \$229,220, booked Jun 11, 2025. Contact: ala@gmail.com, 123. Stay dates: Jun 11, 2025 - Jun 11, 2026. Actions: Confirm, Decline, Contact Student.



The StudentNest platform uses MongoDB as the primary database, chosen for its flexibility, scalability, and ability to store complex, nested data structures efficiently. The database focuses on all core functionalities of the application, including user management, property listings, bookings, messaging, and payments. Each of these entities is modeled as a separate collection, linked through references or embedded documents as appropriate. The “Add New Property” form shown in the mobile interface directly maps to the properties collection. Each document in this collection contains fields like title, description, Type, Rent, address, owner’s Name. Data validation and schema enforcement are handled using Mongoose, ensuring consistency and integrity across all data entries. Furthermore, the platform supports advanced querying, filtering, and indexing, particularly for enabling location-based search and efficient retrieval of available listings. The properties are also relationally linked to users (owners), and bookings, with role-based access determining who can create, modify, or delete records. MongoDB Atlas serves as the remote database host, providing monitoring, security, and performance tools throughout development and deployment. Altogether, the database architecture not only supports the mobile and web interfaces seamlessly but is also engineered for future scaling, supporting high-throughput queries and transactions such as chat messages, bookings, and payment workflows.

Discussion

The development of the StudentNest system represents a thoughtful intersection of user experience, technical scalability, and regional relevance. In creating a housing platform for students in occupied Palestine, the team had to navigate limitations ranging from internet connectivity to payment infrastructure. As a result, the use of MongoDB as a non-relational database was an effective choice. Its schema-less flexibility enabled the storage of nested structures such as chat threads, booking histories, and user roles without the overhead of relational mapping. Paired with Mongoose, MongoDB allowed for robust data modeling while preserving speed and scalability—crucial for mobile-first systems where users expect real-time interaction.

The frontend, split between Next.js for web and React Native for mobile, connects smoothly to a RESTful API backend built with Node.js and Express. Together, these technologies foster rapid development and consistent UI logic across platforms. Key features like real-time messaging, booking systems, and secure payments required not only technical integration but also social awareness—accommodating users unfamiliar with digital housing tools or modern web interactions.

An important discussion point is the adaptability of the platform. By designing modular APIs, centralized error handling, and a layered architecture (with separate services, routes, and controllers), the system can evolve to support future user needs such as expanded payment gateways or analytics dashboards. The success of the mobile "Add Property" feature, for instance, shows how backend flexibility paired with frontend simplicity can enable non-technical users (owners) to participate in the digital economy with ease.

Ultimately, the discussion reveals that a blend of strong technical decisions and contextual awareness has enabled StudentNest to be both practical and

impactful—meeting the current housing needs while paving the way for future expansion.

Conclusions and Recommendations

The development of the StudentNest platform marks a significant step toward modernizing the student housing experience in Palestine. By addressing critical gaps in accessibility, property management, and secure digital transactions, the system provides a comprehensive solution tailored to the socio-economic conditions of the region. Its modular architecture—built using Node.js, Express, MongoDB, Next.js, and React Native—ensures flexibility, scalability, and maintainability across both mobile and web platforms. Key features such as real-time messaging, booking workflows, and Lahza-based payment integration position StudentNest as a regionally optimized alternative to international platforms that often lack contextual relevance.

One of the standout features is the intuitive property management interface, particularly the mobile-based "Add New Property" form, which empowers non-technical owners to easily participate in the rental economy. Secure user authentication via JWT, localized payments, and role-based access further strengthen platform reliability and user trust.

Moving forward, several strategic recommendations can enhance the platform's reach and effectiveness. First, continuing to design with local constraints in mind—such as unstable internet and regulatory payment barriers—is essential. Implementing offline caching capabilities will increase usability for broader audiences. Second, investing in admin analytics tools and owner dashboards can drive smarter decision-making and platform moderation. Lastly, embedding an automated recommendation engine and smart contract features will further streamline the rental process and enhance user engagement.

By staying aligned with user needs while leveraging evolving technologies, StudentNest not only delivers on its promise as a reliable student housing

platform but also lays the groundwork for regional innovation in the digital housing sector. It serves as a blueprint for future systems aiming to bridge local challenges with global-grade solutions.

Future Works

While StudentNest currently delivers a full-featured, localized housing platform, several directions for future development can extend its impact and technical capabilities:

1. **Automated Recommendation Engine:** Integrate a machine learning model to analyze user behavior and preferences, offering smart property suggestions based on location, budget, and prior interactions.
2. **iOS Compatibility and Multi-Language Support:** Expand the mobile app to iOS using Expo or Swift, and introduce multi-language interfaces (e.g., Arabic and English) to make the platform more inclusive.
3. **In-App Digital Contracts:** Build a secure contract generation and e-signature feature where both parties can sign agreements directly within the platform. This would reduce legal ambiguity and streamline tenant-owner relations.
4. **Progressive Web App (PWA):** Convert the web interface into a PWA to allow offline access and better mobile performance without needing full mobile installation.
5. **Data Visualization for Owners:** Develop dashboard tools to help property owners visualize income, occupancy trends, and booking patterns, empowering them to make better business decisions.
6. **Integration with Mapping APIs:** Enhance property discovery through GPS-enabled maps and real-time location-based filtering using APIs like Google Maps or Mapbox.
7. **Automated Moderation System:** Introduce machine learning or rule-based tools to scan for spam listings, fraudulent activities, or inappropriate content, increasing platform trust.

These features will help scale StudentNest beyond its current form, ensuring it evolves with user expectations, technical trends, and market demands while remaining rooted in its mission to support student housing in Palestine.

References

1. MongoDB Inc. (2023). MongoDB Documentation. Retrieved from <https://www.mongodb.com/docs>
2. Axios. (2023). Axios: Promise-based HTTP Client for the Browser and Node.js. Retrieved from <https://axios-http.com/docs/intro>
3. Node.js Foundation. (2023). Node.js Documentation. Retrieved from <https://nodejs.org/en/docs/>
4. Express.js. (2023). Express - Node.js Web Application Framework. Retrieved from <https://expressjs.com/>
5. React Native. (2023). React Native Docs. Meta. Retrieved from <https://reactnative.dev/docs/getting-started>
6. Next.js by Vercel. (2023). Next.js Documentation. Retrieved from <https://nextjs.org/docs>
7. LahzaPay. (2024). Lahza Secure Payment API Documentation. Retrieved from <https://docs.lahza.io/>
8. Zod – TypeScript-first schema validation with static type inference. (2023). Retrieved from <https://zod.dev/>
9. GitHub. (2023). Git Version Control Documentation. Retrieved from <https://git-scm.com/doc>
10. Figma. (2023). Collaborative Interface Design Tool. Retrieved from <https://www.figma.com/>