



An-Najah National University
Faculty of Engineering
Department of Computer Engineering

Software Graduation Project

Nablus Vet Care: A Unified Solution for Veterinary Operations, Client Services, and Business Management

Prepared by:

Baker Razi Wael Yaeesh

Yousef Faed Mahmoud Salman

Supervised by:

Dr.Hanal Abuzant

Presented in partial fulfillment of requirements of Bachelor's degree in Computer Engineering

January , 2026

Acknowledgment

We would like to express our deepest gratitude to Almighty Allah for His endless guidance, support, and blessings throughout the journey of completing this project, "*Nablus Vet Care*." His divine help has been our source of strength and perseverance, enabling us to overcome challenges and achieve success.

We extend our heartfelt appreciation to our supervisor, Dr. Hanal Abuzant, for his continuous support, encouragement, and valuable guidance. His insightful feedback, expertise, and dedication were instrumental in shaping our work and enhancing the quality of this project. Working under his supervision has been an enriching experience that greatly contributed to our academic and professional growth.

We are also sincerely thankful to our families for their unwavering love, patience, and encouragement. Their faith in us has been a constant source of motivation and inspiration. Finally, we would like to thank our friends for their continuous support, collaboration, and positivity, which made this journey both productive and enjoyable.

Disclaimer

This report was prepared by “Baker Razi Wael Yaeesh” and “Yousef Faed Mahmoud Salman” from the Department of Computer Engineering, Faculty of Engineering, An-Najah National University.

It has not been modified or corrected, except for minor editorial adjustments made during the assessment process, and it may contain language or content errors.

The opinions, findings, and recommendations presented in this report are solely those of the authors and do not necessarily reflect the views of An-Najah National University. The University bears no responsibility or liability for the use of this report for any purpose other than that for which it was originally prepared.

Important Notice Regarding System Data and Simulation

The authors further declare and emphasize that all data (including but not limited to: names of clients, veterinarians, and staff; pet and patient details; medical records; appointments; invoices; financial transactions; contact information; and any associated images) presented within this report—whether in system screenshots, diagrams, example outputs, or descriptive scenarios—are entirely synthetic, fictitious, and generated for simulation purposes only (100% Fake/Dummy Data).

This data was programmatically created using data generation tools and libraries (e.g., Faker) exclusively for the purposes of software development, functional testing, proof-of-concept demonstration, and academic illustration. None of this data represents, corresponds to, or is derived from any real individual, business, organization, or animal. All dates, numerical values, identifiers, and descriptive content are fabricated and do not reflect any real-world events, records, or statistics.

The sole intent behind using such data is to:

1. Demonstrate system functionality and user interfaces without compromising privacy.
2. Test database schemas and application logic in a controlled environment.
3. Fulfill the technical and academic requirements of this graduation project.

No real personal, medical, or financial information was used or disclosed in the creation of this project.

Contents

Chapter 1: Introduction	8
1.1 General Background	8
1.2 Problem Statement	8
1.3 Objectives	8
1.4 Significance and Importance of Our Work	9
1.5 Scope and Limitations	10
1.6 Organization of the Report	11
Chapter 2: Literature Review	12
2.1 The Digital Transformation Imperative in Veterinary Practice	12
2.2 Existing Veterinary Management Solutions: A Comparative Landscape	12
2.3 Foundational Technologies and Architectural Paradigms	13
2.4 Identifying the Research and Development Gap	13
2.5 Conclusion of the Literature Review	14
Chapter 3: Methodology	15
3.1 Standards and Specifications	15
3.2 Constraints	15
3.3 Academic Foundation	16
3.4 Tools, Methods, and Programming Languages	16
3.4.1 Client Side Architecture	16
3.4.2 Server Side Architecture	17
3.4.3 Database Architecture	18
3.4.3.1 Introduction to the System	18
3.4.3.2 The People: Roles and Responsibilities	19
3.4.3.3 The Patients: Pet Management	19
3.4.3.4 The Core Workflow: From Appointment to Payment	19
3.4.3.5 Financial Management and Salaries	20
3.4.3.6 Communication and Community Forum	20
3.4.3.7 Summary of Data Integrity	21

Chapter 4: System Features and Implementation	23
4.1 System Overview	23
4.2 Authentication and Security System	24
4.3 Admin Portal Features	27
4.4 Veterinarian Portal Features.....	31
4.5 Receptionist Portal Features	36
4.6 Pet Owner Portal Features	39
4.7 Accountant Portal Features	44
4.8 AI-Powered Features	48
4.9 Communication System.....	48
4.10 User Experience and Technical Implementation	49
Chapter 5: Results and Analysis	50
Chapter 6: Discussion	51
Chapter 7: Conclusion and Recommendations	53
References	55

Table Of Figures

Figure 1 Nablus Vet Care database schema. _____	22
Figure 2 Nablus Vet Care web landing page. _____	24
Figure 3 Nablus Vet Care web landing page in Hebrew. _____	25
Figure 4 Nablus Vet Care web landing page in dark mode _____	25
Figure 5 Nablus Vet Care web landing page in Arabic. _____	25
Figure 6 Nablus Vet Care web login page. _____	25
Figure 7 Nablus Vet Care web sign-up page. _____	25
Figure 8 Nablus Vet Care web “Forgot Password” page. _____	25
Figure 9 Nablus Vet Care web OTP verification via email. _____	25
Figure 10 Nablus Vet Care mobile landing page, login, and sign-up screens. _____	26
Figure 11 Nablus Vet Care web admin dashboard screen Cont. _____	28
Figure 12 Nablus Vet Care web admin dashboard screen. _____	28
Figure 13 Nablus Vet Care web admin appointments screen. _____	28
Figure 14 Nablus Vet Care web admin veterinarians screen. _____	28
Figure 15 Nablus Vet Care web admin users screen. _____	28
Figure 16 Nablus Vet Care web admin staff screen. _____	28
Figure 17 Nablus Vet Care web admin pets screen. _____	29
Figure 18 Nablus Vet Care web admin pet owners screen. _____	29
Figure 19 Nablus Vet Care web admin subscription payments screen. _____	29
Figure 20 Nablus Vet Care web admin finance screen. _____	29
Figure 21 Nablus Vet Care web admin performance screen. _____	29
Figure 22 Nablus Vet Care web admin finance screen Cont. _____	29
Figure 23 Nablus Vet Care web admin performance screen Cont. _____	29
Figure 24 Nablus Vet Care web admin expenses screen. _____	30
Figure 25 Sample Nablus Vet Care mobile admin portal screens. _____	30
Figure 26 Nablus Vet Care web salaries screen. _____	30
Figure 27 Nablus Vet Care web veterinarian dashboard. _____	32
Figure 28 Nablus Vet Care web veterinarian medical records details screen. _____	33
Figure 29 Nablus Vet Care web veterinarian forum screen. _____	33
Figure 30 Nablus Vet Care web veterinarian medical records Cont. _____	33
Figure 31 Nablus Vet Care web veterinarian medical records Cont. _____	33
Figure 32 Nablus Vet Care web veterinarian creating a post using AI classification or custom input. _____	33
Figure 33 Nablus Vet Care web veterinarian pet files management. _____	33
Figure 34 Nablus Vet Care veterinarian analytics screen. _____	34
Figure 35 Sample Nablus Vet Care mobile veterinarian portal screens. _____	35
Figure 36 Nablus Vet Care web receptionist dashboard screen. _____	37
Figure 37 Nablus Vet Care web receptionist “New Appointment” screen. _____	37
Figure 38 Nablus Vet Care messaging system (available across all portals). _____	37
Figure 39 Nablus Vet Care web receptionist appointments management screen. _____	37
Figure 40 Nablus Vet Care web receptionist veterinarian appointments screen. _____	37
Figure 41 Nablus Vet Care web receptionist “Add New Pet Owner” screen. _____	37
Figure 42 Nablus Vet Care web receptionist invoices management screen. _____	38
Figure 43 Nablus Vet Care web receptionist subscriptions and payments management screen. _____	38

<i>Figure 44 Sample Nablus Vet Care mobile receptionist portal screens.</i>	<u>38</u>
<i>Figure 45 Nablus Vet Care web pet owner portal dashboard screen.</i>	<u>40</u>
<i>Figure 46 Nablus Vet Care web pet owner notifications screen.</i>	<u>40</u>
<i>Figure 47 Nablus Vet Care web pet owner AI assistant chatbot screen.</i>	<u>40</u>
<i>Figure 48 Nablus Vet web Care pet owner chat screen.</i>	<u>40</u>
<i>Figure 49 Nablus Vet Care web pet owner upcoming appointments screen.</i>	<u>41</u>
<i>Figure 50 Nablus Vet Care web pet owner forum screen.</i>	<u>41</u>
<i>Figure 51 Nablus Vet Care web pet owner creating a forum post with AI classification or custom category.</i>	<u>41</u>
<i>Figure 52 Nablus Vet Care web pet owner creating a post using AI screen.</i>	<u>41</u>
<i>Figure 53 Nablus Vet Care web pet owner invoices screen.</i>	<u>41</u>
<i>Figure 54 Nablus Vet Care web pet owner portal – pet files screen.</i>	<u>41</u>
<i>Figure 55 Nablus Vet Care web pet owner payment successful screen.</i>	<u>41</u>
<i>Figure 56 Nablus Vet Care web pet owner paying an invoice using Stripe.</i>	<u>41</u>
<i>Figure 57 Nablus Vet Care web pet owner booking an appointment screen.</i>	<u>42</u>
<i>Figure 58 Nablus Vet Care web pet owner selecting an appointment from AI-suggested options.</i>	<u>42</u>
<i>Figure 59 Nablus Vet Care web incoming call from veterinarian (available across all portals).</i>	<u>42</u>
<i>Figure 60 Sample Nablus Vet Care mobile pet owner portal screens.</i>	<u>43</u>
<i>Figure 61 Nablus Vet Care web accountant portal dashboard.</i>	<u>45</u>
<i>Figure 62 Nablus Vet Care web accountant salaries management screen.</i>	<u>45</u>
<i>Figure 63 Nablus Vet Care web accountant expenses management screen.</i>	<u>45</u>
<i>Figure 64 Nablus Vet Care web subscription payment management screen.</i>	<u>45</u>
<i>Figure 65 Nablus Vet Care web accountant invoices management screen.</i>	<u>45</u>
<i>Figure 66 Nablus Vet Care web AI-generated reports screen.</i>	<u>46</u>
<i>Figure 67 Sample Nablus Vet Care mobile accountant screens.</i>	<u>47</u>

Abstract

The Nablus Vet Care System is a comprehensive, AI-powered veterinary clinic management platform designed to revolutionize veterinary operations through a unified web and mobile application ecosystem. This innovative system automates essential veterinary workflows including multi-role user management, intelligent appointment scheduling, medical record keeping, and financial operations, providing seamless experiences for pet owners, veterinarians, and administrative staff alike.

The platform features advanced capabilities such as AI-powered diagnostic assistance, real-time communication with voice and chat functionality, secure payment processing via Stripe integration, and comprehensive medical record management. With multi-language support for English, Arabic, and Hebrew including full RTL layout capabilities, the system serves diverse communities while maintaining cultural and linguistic appropriateness.

Built on a modern technology stack utilizing React 18 with TypeScript for web, React Native for mobile applications, and Node.js with Express for backend services, the system incorporates real-time features through Socket.io and Firebase push notifications. The architecture supports five distinct user roles—Admin, Veterinarian, Receptionist, Pet Owner, and Accountant—each with tailored interfaces and capabilities that streamline their specific workflows.

Key innovations include a 5-step appointment creation process for receptionists, AI veterinary diagnostic support for medical professionals, blocked appointment scheduling for pet owners with outstanding payments, and comprehensive financial management for accounting staff. The integration of community forums, voice calling capabilities, and intelligent notification systems fosters enhanced communication and collaboration across all stakeholders.

The implementation of the Nablus Vet Care System has demonstrated significant improvements in operational efficiency, reduced administrative overhead, enhanced patient care through AI assistance, and increased satisfaction among both veterinary staff and pet owners. By automating routine tasks, providing intelligent insights, and enabling seamless multi-platform access, the system represents a substantial advancement in veterinary practice management technology, particularly suited for the unique needs of the Middle Eastern region while maintaining global standards of veterinary care excellence.

Chapter 1: Introduction

1.1 General Background

Veterinary clinics are essential healthcare providers for companion animals, yet many continue to rely on manual, paper-based systems for appointments, medical records, and communication. These traditional methods often lead to operational inefficiencies, scheduling conflicts, and communication gaps between veterinary staff and pet owners.

While digital transformation has revolutionized human healthcare, the veterinary sector has been slower to adopt comprehensive management solutions. This technological gap is particularly evident in regions like Nablus, where clinics face additional challenges in accessing specialized expertise and efficient practice management tools.

The Nablus Vet Care System addresses these challenges through an integrated digital platform that modernizes veterinary practice management. This AI-powered solution combines web and mobile applications to streamline appointment scheduling, electronic medical records, billing, and client communication. By incorporating real-time features, multi-language support, and diagnostic assistance, the system enhances operational efficiency while improving the quality of veterinary care delivery.

This project represents a significant step toward digital transformation in veterinary medicine, providing clinics with the tools to manage growing patient loads effectively while offering pet owners better access to their animals' healthcare information and veterinary services.

1.2 Problem Statement

Veterinary clinics in Nablus and similar regions predominantly rely on manual, paper-based systems and fragmented digital tools for core operations such as appointment scheduling, medical record-keeping, client communication, and financial management. This approach leads to significant operational inefficiencies, including scheduling conflicts, data inconsistencies, communication delays between staff and pet owners, and a lack of integrated oversight for administrative and financial tasks. Furthermore, there is an absence of regionally tailored solutions that offer comprehensive multi-language support and culturally appropriate interfaces. This technological gap results in increased administrative burdens, reduced quality of patient care, diminished client satisfaction, and hindered business growth for veterinary practices.

1.3 Objectives

The primary objective of this project is to develop a comprehensive, user-friendly veterinary management system that modernizes and simplifies clinic operations. The specific aims are as follows:

- **Automate Administrative Tasks:** Streamline processes including appointment scheduling, patient registration, medical record keeping, and billing to reduce manual effort and minimize errors.
- **Enhance Multi-role Coordination:** Facilitate seamless collaboration between administrators, veterinarians, receptionists, and accounting staff through integrated workflows and real-time data sharing.
- **Improve Client Communication:** Provide pet owners with direct access to appointment booking, medical records, and secure messaging with veterinary staff, strengthening client-clinic relationships.
- **Support Clinical Decision-Making:** Integrate AI-powered diagnostic assistance and treatment recommendations to support veterinarians in providing evidence-based care.
- **Enable Digital Payment Processing:** Implement secure online payment capabilities through Stripe integration, simplifying billing and reducing administrative overhead.
- **Provide Accessible Pet Health Information:** Empower pet owners with comprehensive access to their pets' medical histories, vaccination records, and treatment plans through both web and mobile platforms.
- **Facilitate Community Knowledge Sharing:** Develop a forum system where pet owners can seek advice and veterinary professionals can share expertise and best practices.

The overarching purpose of this project is to enhance the overall efficiency, quality of care, and operational effectiveness of veterinary practices through digital transformation.

1.4 Significance and Importance of Our Work

The Nablus Vet Care System addresses critical needs in modern veterinary practice through digital transformation:

- **Operational Efficiency:** Automates manual processes like appointment scheduling and medical records, reducing administrative workload and minimizing errors.
- **Enhanced Patient Care:** AI-powered diagnostic support helps veterinarians make more accurate clinical decisions and improve treatment outcomes.
- **Improved Client Experience:** Provides pet owners with convenient access to medical records, easy appointment booking, and direct communication with veterinary staff.
- **Economic Benefits:** Streamlined operations and automated billing help clinics reduce costs and optimize resource utilization.
- **Competitive Advantage:** Positions veterinary practices as modern healthcare providers through innovative technology adoption.

Our system represents a significant advancement in veterinary management, benefiting clinics, professionals, and pet owners through improved efficiency and quality of care.

1.5 Scope and Limitations

Scope:

The Nablus Vet Care system is a comprehensive, unified platform designed to manage the complete operational lifecycle of a modern veterinary clinic. Its scope encompasses:

- **Core Management:** Multi-role portals for **Administrators, Veterinarians, Receptionists, Pet Owners, and Accountants.**
- **Core Features:** Automated **appointment scheduling, electronic medical records (EMR), client/pet management, and financial operations** (invoicing, expense tracking, payroll).
- **Advanced Capabilities:** **AI-powered diagnostic assistance, a real-time communication system** (chat, notifications, voice alerts), and a **community forum.**
- **Client Services:** **Online appointment booking, medical record access, and secure payment processing** via Stripe for pet owners.
- **Technical Implementation:** A full-stack application with a **responsive web interface** (React) and **cross-platform mobile apps** (React Native).
- **Regional Focus:** Full **multi-language support** (English, Arabic, Hebrew) with **RTL layout capabilities** and localized user experiences.

Limitations:

- **Medical Scope:** The system provides **AI-assisted decision support** but does not replace professional veterinary judgment, conduct physical diagnoses, or manage in-clinic medical equipment integration (e.g., X-ray or lab machine interfaces).
- **Geographic & Logistic Operations:** It does not handle **inventory management** for medical supplies, **pharmacy dispensary operations, or pet boarding/hotel services.**
- **Integration Boundaries:** While it includes payment processing, it does not offer native **accounting software integration** (e.g., with QuickBooks) or direct **insurance claim processing.**
- **Testing Scope:** Comprehensive testing was prioritized for **core functionality and the Android mobile platform;** extensive **cross-browser** and **iOS-specific** testing was limited by resource constraints.

- **Deployment & Scale:** The current implementation is designed as a **clinic-level solution** and does not natively support multi-branch franchise management or large-scale, multi-tenant SaaS architectures without significant modification.

1.6 Organization of the Report

This report is structured to methodically present the development and evaluation of the Nablus Vet Care System.

- **Chapter 1 (Introduction)** provides the project's background, problem statement, objectives, significance, and scope.
- **Chapter 2 (Literature Review)** analyzes existing veterinary solutions and foundational technologies, establishing the research gap our project addresses.
- **Chapter 3 (Methodology)** details the development process, including standards, tools, and the core architectural design (client-side, server-side, and database).
- **Chapter 4 (System Features and Implementation)** comprehensively describes the functionalities and implementation of each user portal (Admin, Veterinarian, Receptionist, Pet Owner, Accountant) and core system features like AI and communication.
- **Chapter 5 (Results and Analysis)** presents the outcomes, performance metrics, and an analysis of the implemented system's effectiveness.
- **Chapter 6 (Discussion)** interprets the results, evaluates the system against its objectives, and explores the implications of the findings.
- **Chapter 7 (Conclusion and Recommendations)** summarizes the project's achievements and proposes future enhancements and development directions.
- **References** lists all technical documentation, standards, and resources cited throughout the report.

Chapter 2: Literature Review

2.1 The Digital Transformation Imperative in Veterinary Practice

The global veterinary care market is undergoing a significant shift, mirroring the digital transformation seen in human healthcare. Historically, clinics have relied on fragmented systems—manual appointment books, paper medical records, and basic accounting software—leading to operational inefficiencies, data silos, and communication gaps with pet owners (a common challenge identified in your project's background). This manual approach is increasingly unsustainable, creating a clear market need for integrated practice management software (PMS). Modern solutions aim to consolidate scheduling, electronic medical records (EMR), billing, and client communication into a unified platform. Your project, *Nablus Vet Care*, directly addresses this need by proposing a comprehensive, AI-enhanced ecosystem, positioning itself within this wave of innovation aimed at improving clinical workflow, data-driven decision-making, and the overall client experience.

2.2 Existing Veterinary Management Solutions: A Comparative Landscape

A review of existing veterinary software reveals a spectrum of solutions, each with strengths and limitations that help define the niche for *Nablus Vet Care*.

- **Established Commercial Platforms (e.g., IDEXX Cornerstone, Covetrus):** These are full-featured, enterprise-grade systems offering deep functionality for large practices or chains. They are often characterized by high costs, complex implementations, and generic interfaces that may not address regional specificities. Their architecture can be monolithic, making them less agile for smaller clinics or for integrating novel features like advanced AI diagnostics or community forums, which are central to your project.
- **Cloud-Based & SaaS Solutions:** Newer entrants offer subscription-based, cloud-hosted platforms promising greater accessibility and regular updates. They typically improve upon legacy systems with better user interfaces and mobile access. However, many still function as digitized versions of traditional processes rather than re-imagining the workflow through a truly user-centric, multi-role lens. Furthermore, their support for multi-language and Right-to-Left (RTL) layouts for languages like Arabic and Hebrew—a core requirement you identified for the Nablus region—is often an afterthought or inadequately implemented.
- **Open-Source and Academic Projects:** Various academic and open-source initiatives have explored specific facets of veterinary informatics, such as standalone EMR systems or telemedicine portals. While innovative, these projects frequently suffer from limited scope, lack of long-term maintenance, or an absence of a cohesive, production-ready architecture that integrates all facets of clinic operation—administration, clinical work, finance, and client engagement—into a single, seamless experience.

2.3 Foundational Technologies and Architectural Paradigms

The *Nablus Vet Care* system is built upon a carefully selected modern technology stack, each component chosen to meet specific project requirements for performance, developer experience, and scalability.

- **Frontend Architecture (React & React Native):** The project utilizes **React** for its component-based architecture, which is ideal for building complex, interactive user interfaces like the multi-role dashboards you described. The official React documentation emphasizes creating reusable components, managing state, and building dynamic applications—principles that are evident in your portal designs. For cross-platform mobile development, **React Native** is employed, enabling code reuse across iOS and Android while delivering a native experience, crucial for pet owners and staff on the go.
- **Styling and UI Components:** To ensure a consistent, accessible, and modern interface, the project leverages **Tailwind CSS** for utility-first styling, which promotes rapid UI development. It is complemented by **shadcn/ui**, which provides open-code, accessible components built on top of **Radix UI** primitives. As noted in the Radix UI documentation, this approach prioritizes accessibility, customization, and developer experience, allowing your team to build a high-quality design system without being locked into a pre-styled component library. **Lucide** provides a consistent set of open-source icons for the interface.
- **Backend and Real-Time Communication:** The backend is powered by **Node.js** and **Express.js**, a standard and performant combination for building RESTful APIs. A key innovation is the integration of [Socket.IO](#) for real-time, bidirectional communication. As highlighted in its documentation, [Socket.IO](#) provides a reliable channel for features like live chat, notifications, and voice call alerts by establishing WebSocket connections with fallback options, fulfilling the "real-time communication" requirement critical for a dynamic clinic environment.
- **Development Tooling:** The use of **Vite** as a build tool significantly enhances the frontend development experience. Vite offers extremely fast Hot Module Replacement (HMR) and optimized builds, which accelerates the development cycle. **TypeScript** is used throughout to add static type checking, reducing runtime errors and improving code maintainability for a large-scale project like this one.

2.4 Identifying the Research and Development Gap

The synthesis of the above analysis reveals the specific gap that *Nablus Vet Care* aims to fill. While solutions exist, there is a pronounced need for a **regionally-aware, integrally-designed platform** that:

1. Seamlessly unifies all clinic operations (administrative, clinical, financial, communicative) without the bloat or cost of enterprise software.
2. Genuinely prioritizes the user experience for *all* stakeholders (Admin, Vet, Receptionist, Pet Owner, Accountant) with role-optimized workflows.
3. Incorporates advanced, practical AI assistance not as a gimmick but as an integrated tool for diagnostics and client support.
4. Is built from the ground up with full multi-language (EN/AR/HE) and RTL support, ensuring cultural and linguistic appropriateness for its target market.
5. Employs a modern, modular tech stack (React, Node.js, [Socket.IO](#)) that is both powerful for developers and capable of delivering a fast, responsive experience for end-users on web and mobile.

2.5 Conclusion of the Literature Review

Therefore, *Nablus Vet Care* is positioned not merely as another veterinary management system, but as a specialized, next-generation platform. It draws upon the proven strengths of modern web technologies and architectural patterns to address the specific limitations observed in current market offerings. By combining a robust full-stack foundation with AI augmentation, comprehensive multi-role portals, and dedicated regional localization, the project contributes to the field by demonstrating how veterinary practice management can be re-engineered for efficiency, accessibility, and enhanced care in a digitally connected world. Your work stands to bridge the gap between generic commercial software and the nuanced needs of modern veterinary clinics, particularly in the Middle Eastern context.

Chapter 3: Methodology

The Nablus Vet Care System represents a transformative advancement in veterinary practice management by delivering comprehensive digital solutions for clinical operations, patient care, and administrative workflows. It incorporates essential features including AI-powered diagnostic assistance, real-time communication, and integrated payment processing, utilizing advanced technologies like WebRTC for voice consultations and Firebase Cloud Messaging for cross-platform notifications. This section details the tools, methodologies, and systematic approaches employed in developing the Nablus Vet Care System across web and mobile platforms.

3.1 Standards and Specifications

The Nablus Vet Care System adheres to numerous software engineering standards ensuring reliability, security, and compliance with industry best practices. The development followed IEEE software engineering standards for system design and maintained compliance with healthcare data protection protocols. The architecture implements RESTful API conventions, JWT authentication standards, and responsive web design principles. Alternative design approaches were evaluated against these established standards to ensure optimal system architecture.

3.2 Constraints

- **Complex Feature Integration:** We encountered significant challenges in identifying and integrating the comprehensive feature set required to meet the diverse needs of veterinary clinics, including medical professionals, administrative staff, and pet owners. Achieving an optimal balance between functionality, usability, and performance across multiple user roles presented substantial design constraints during system development.
- **Timeline and Learning Curve:** The project timeline was constrained by academic schedules and parallel commitments, requiring efficient time management across the development cycle. As this represented our first comprehensive implementation of a full-stack system integrating React, React Native, Node.js, and multiple third-party services, a considerable portion of the project duration was allocated to mastering these technologies and their integration patterns.
- **Testing Limitations:** Comprehensive testing was primarily conducted on web browsers and Android devices due to resource constraints, particularly regarding iOS device availability and extensive cross-browser compatibility testing. These limitations necessitated strategic prioritization of core functionality and user experience, with focus maintained on delivering a robust, stable application within available resources.

3.3 Academic Foundation

The Computer Engineering curriculum provided fundamental knowledge directly applicable to developing the Nablus Vet Care System. Core concepts from Software Engineering, Database Systems, and Web Technologies courses formed the theoretical foundation for system architecture, data modeling, and full-stack development principles. These academic components equipped us with essential understanding of frontend and backend development methodologies, project management frameworks, and scalable system design.

To complement university coursework, we engaged in extensive self-directed learning through online courses and documentation for React, React Native, Node.js, and cloud services integration. Database design principles acquired through academic study proved crucial in implementing secure, efficient data storage and retrieval systems. The integration of formal education with practical skill development created a robust technical foundation, enabling effective development of the veterinary management platform while addressing diverse stakeholder requirements.

3.4 Tools, Methods, and Programming Languages

The Nablus Vet Care System was developed using a comprehensive suite of tools, methodologies, and programming languages. For frontend development, we utilized React 18 with TypeScript for web interfaces and React Native for mobile applications, with styling implemented through Tailwind CSS for consistent design language. Backend services were built using Node.js with Express.js to establish a RESTful API architecture managing application data and business logic. PostgreSQL was employed for relational database design, maintaining consistent data schemas across web and mobile platforms. Firebase Cloud Messaging was integrated for unified push notifications. Throughout development, Git facilitated version control and collaborative workflow management among team members.

3.4.1 Client Side Architecture

- **Design Methodology:** The design process for the Nablus Vet Care System began with extensive research into veterinary clinic operations, identifying pain points in existing management solutions. Our investigation revealed significant gaps in integrated digital platforms capable of handling medical records, appointment scheduling, client communication, and billing within a unified system. This insight guided our development of a comprehensive platform that streamlines veterinary operations while enhancing patient care quality. We designed the system from foundational principles, creating specialized dashboards for each user role that optimize specific workflows. The design emphasizes modern aesthetics, intuitive navigation, and operational efficiency, incorporating research findings, user requirement analysis, and industry best practices. Careful attention was given to color schemes, typography, and interactive elements that enhance user experience while maintaining professional clinical appropriateness.

- **Framework Selection:** Our frontend development leverages React 18 for web applications and React Native for mobile platforms. React, maintained by Meta, provides a component-based architecture for building sophisticated user interfaces, while React Native enables cross-platform mobile development from a single codebase targeting both iOS and Android ecosystems. These frameworks offer extensive features including virtual DOM optimization, component reusability, extensive library ecosystems, and rapid development cycles with hot reload capabilities. Our selection of React and React Native was driven by their proven performance characteristics, vibrant developer communities, and robust cross-platform capabilities. This approach ensured development efficiency while maintaining consistent user experiences across device types, making these frameworks ideal choices for our veterinary management system.
- **Programming Languages:** TypeScript served as the primary programming language for client-side development with React and React Native. As a strongly-typed superset of JavaScript, TypeScript enhances code quality, developer productivity, and application reliability through static type checking. Our selection of TypeScript was influenced by its growing industry adoption, enhanced tooling support, and the maintainability benefits gained through type safety. TypeScript's flexibility and comprehensive ecosystem proved ideal for developing complex, interactive user interfaces. Additional libraries including TanStack Query for server state management and Axios for HTTP requests accelerated development and simplified implementation of sophisticated features. The language's continuous evolution and enterprise adoption made it a strategic choice for building robust, scalable veterinary applications.

3.4.2 Server Side Architecture

- **System Architecture:** We implemented a client-server architecture for the Nablus Vet Care System, enabling multiple client applications (web and mobile) to consume a unified API. The server infrastructure was designed to support RESTful API endpoints alongside WebSocket connections for real-time features, ensuring seamless communication between clients and servers while maintaining data consistency across platforms.
- **Framework Implementation:** We employed Node.js with Express.js as our primary server-side environment. This technology selection was guided by several factors including JavaScript ecosystem consistency across frontend and backend, extensive documentation, horizontal scalability, and exceptional I/O performance characteristics that align perfectly with our application requirements. Express.js, as the standard web application framework in the Node.js ecosystem, provided robust routing, middleware support, and HTTP utility methods essential for our backend development. The combination of Node.js and Express.js established a performant, maintainable server-side

architecture capable of supporting our veterinary management platform's complex requirements.

3.4.3 Database Architecture

Effective database design was fundamental to the Nablus Vet Care System development. This involved defining structured data organization, relationships, and integrity constraints within the database. Our database architecture was meticulously crafted to capture and manage all necessary information for comprehensive veterinary practice management.

We selected PostgreSQL as our database management system for its reliability, advanced features, and robust support for complex queries. As an open-source object-relational database system, PostgreSQL provides organized data storage and manipulation capabilities with strong ACID compliance. The primary design objective was to facilitate diverse database applications while delivering powerful features for data storage, access control, and management.

Our database schema incorporates comprehensive tables representing core system components including users, pets, medical records, appointments, prescriptions, and invoices. Each table contains specific fields describing attributes of corresponding entities, with appropriate data types and constraints. For example, the medical records table includes diagnosis documentation, treatment plans, and veterinary notes with proper relationship mapping.

Table relationships were carefully designed to express data dependencies across the system. For instance, appointment records reference both pet and veterinarian tables, while prescription records maintain relationships with medical records and medication information. This relational structure ensures data cohesion and supports efficient querying for the veterinary management system.

Beyond tables and relationships, database constraints play a crucial role in maintaining data integrity and consistency. These constraints enforce business rules and conditions that data must satisfy before insertion or modification. For example, appointment status fields are constrained to valid values like "scheduled," "completed," or "cancelled," while foreign key constraints maintain referential integrity across related tables.

The database design forms a critical component of the Nablus Vet Care System architecture, providing a structured framework for data organization, storage, and retrieval. This foundation enables efficient management of veterinary practice operations, with PostgreSQL delivering the robust database capabilities necessary for project success.

3.4.3.1 Introduction to the System

The Nablus Vet Care database is designed to manage a modern veterinary clinic, integrating medical, financial, administrative, and community functions. At its core is the users table, which acts as the single source of truth for all human actors in the system, storing essential authentication and profile information like email, password hash, name, and role.

3.4.3.2 The People: Roles and Responsibilities

The system distinguishes between five main roles, each extending from the base users table:

- **Administrators:** Have full permissions to manage and administer the system.
- **Pet Owners:** These are the clients. The `pet_owners` table links to a user and stores client-specific information like their subscription plan (e.g., 'free', 'premium'), subscription dates, and emergency contact details. They are the financial and emotional stakeholders for the pets.
- **Veterinarians:** These are the medical experts. The `veterinarians` table extends the users table with professional details like their specialization (e.g., 'Surgery', 'Dentistry'), license number (which must be unique), years of experience, and their consultation fee. Their availability is tracked via the `is_available` flag.
- **Staff:** This group represents the administrative backbone of the clinic. The `staff` table, also linked to users, defines their position (e.g., 'Receptionist', 'Accountant'), salary, and hire date. A receptionist and an accountant, while both are staff, have different permissions and responsibilities within the application logic.
- **Accountants:** A specialized role within the staff table, responsible for all financial operations, including recording expenses and processing salaries.

3.4.3.3 The Patients: Pet Management

The `pets` table is central to the medical operations. Each pet is owned by a single `pet_owner` (creating a one-to-many relationship) and records vital information such as species, breed, date of birth, weight, and a unique microchip number.

The life of a pet in the system is documented through several related records:

- `pet_files`: Stores documents like X-rays, lab reports, or insurance papers uploaded by staff or vets.
- `vaccinations`: Maintains a complete vaccination history, including the vaccine name, date administered, the next due date, and which veterinarian administered it.

3.4.3.4 The Core Workflow: From Appointment to Payment

This is the primary business process of the clinic.

1. **Scheduling an Appointment:** A `pet_owner` requests an appointment for their pet. A staff member with the receptionist role schedules this in the `appointments` table, assigning it to an available veterinarian. The appointment records the date, duration, type (e.g., 'check-up', 'surgery'), and reason.
2. **The Medical Consultation:** During the appointment, the veterinarian creates a `medical_record`. This crucial record links to the appointment, the pet, and

the veterinarian. It contains clinical notes: diagnosis, symptoms, vital signs (temperature, heart rate), and a plan for a follow-up date.

3. **Prescribing Medication:** Based on the `medical_record`, the vet can issue one or more prescriptions. This table details the medication, dosage, frequency, and instructions, forming a clear treatment plan.
4. **Billing and Invoicing:** After the appointment, the financial process begins. An invoice is generated, linked to the appointment and the `pet_owner`. A receptionist is often responsible for creating it. The invoice contains one or more `invoice_items` (e.g., 'Consultation Fee', 'Medication'), which automatically calculate the `total_price` based on quantity and unit price. The invoice tracks the payment status ('pending', 'paid'), method, and date.

3.4.3.5 Financial Management and Salaries

The clinic's internal finances are managed separately from client invoicing.

- **Expenses:** An accountant (a type of staff) records all clinic expenditures in the expenses table. This includes category, description, amount, and an optional receipt image. These records may require admin approval (`approved_by_admin`).
- **Salaries:** The salaries table is used to pay both staff (receptionists, accountants) and veterinarians. It records the payment amount, the period it covers, the payment date, and status. The accountant who processed the payment is also recorded for accountability.
- **Subscriptions:** For `pet_owners` on a subscription plan, their payments are logged in the `subscription_payments` table, tracking the amount, payment period, and connection to Stripe for online payments.

3.4.3.6 Communication and Community Forum

The system fosters communication through two main channels:

- **Real-time Chat:** The `chats` table allows direct messaging between users. Any user can be a sender or a receiver, enabling conversations between pet owners and vets, or between staff members. It supports text messages and file sharing.
- **Community Forum:** A structured forum allows users to seek advice and share knowledge.
 - `forum_categories` organize posts by topics and can be nested using a `parent_id`.
 - `forum_posts` are the initial questions or discussions, created by a user (the `author_id`).

- forum_replies are the answers or comments on those posts. They can also be nested to create threaded discussions using parent_reply_id. A reply can be marked as the is_solution.
- forum_likes allow users to show appreciation for both posts and replies, with a unique constraint preventing multiple likes from the same user on the same item.
- **Notifications:** The notifications system keeps users informed. It can send targeted messages to a user_id about various events (e.g., "Your appointment is confirmed," "A new reply to your post"), linking them directly to the relevant record (appointment, forum_post, etc.) via related_entity_type and related_entity_id.

3.4.3.7 Summary of Data Integrity

The database ensures reliability through several mechanisms:

- **Cascading Deletes:** For example, deleting a pet will automatically delete all its medical_records, vaccinations, and appointments, preventing orphaned records.
- **Unique Constraints:** Critical fields like user_email, veterinarian_license_number, and pet_microchip_number are enforced to be unique.
- **Generated Columns:** The total_price in invoice_items is automatically calculated as quantity * unit_price, ensuring accuracy.
- **Comprehensive Foreign Keys:** Every relationship is explicitly defined, ensuring that you cannot have an appointment for a non-existent pet or an invoice from a non-existent receptionist.

In conclusion, this schema creates a robust, interconnected ecosystem that seamlessly manages the journey of a pet and its owner through the clinic, while efficiently handling the clinic's internal operations, finances, and community engagement.

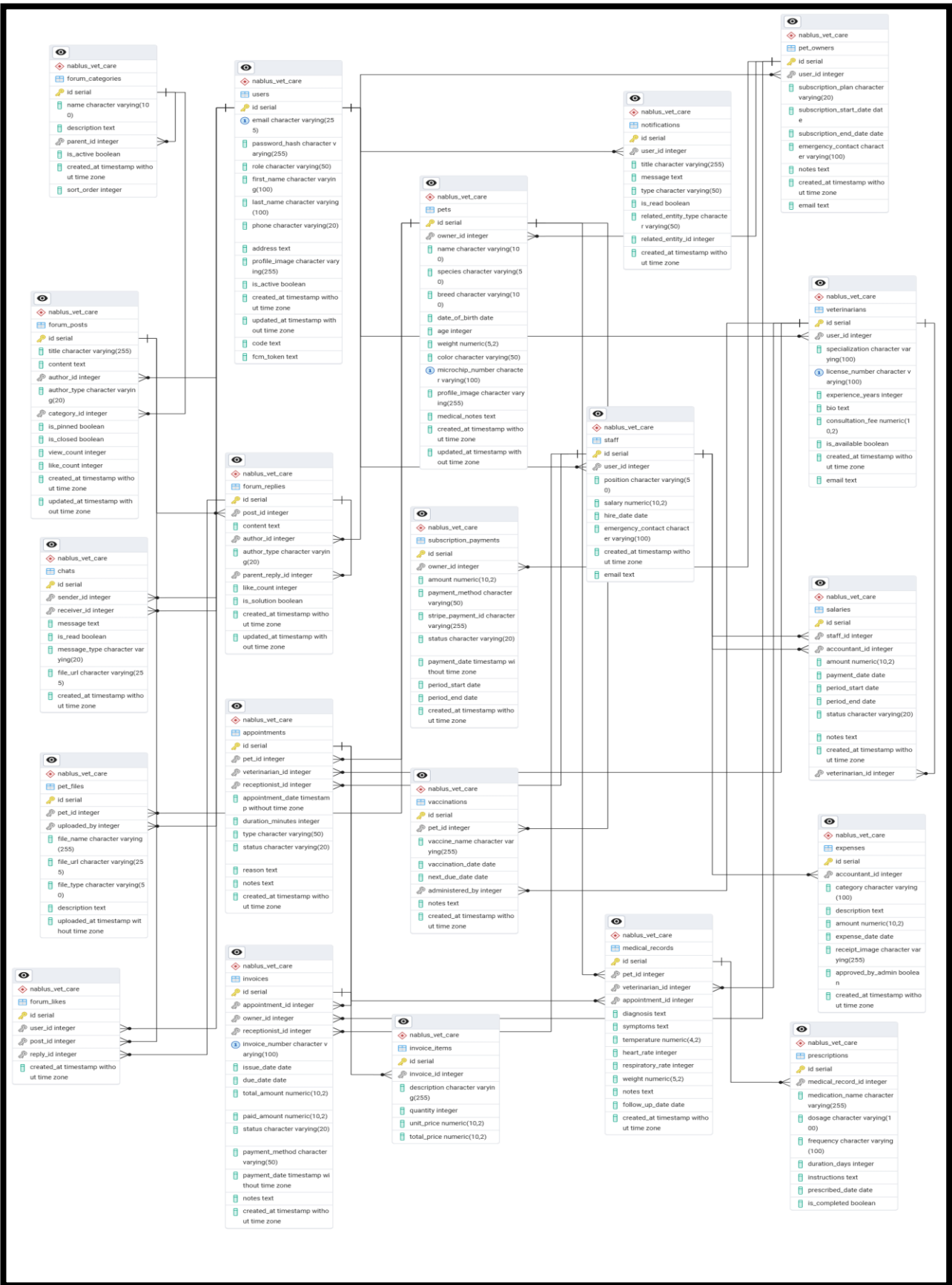


Figure 1 Nabus Vet Care database schema.

Half of this page has been intentionally left blank for layout purposes.

Chapter 4: System Features and Implementation

4.1 System Overview

The Nablus Vet Care System represents a comprehensive veterinary clinic management platform designed to streamline operations across all aspects of modern veterinary practice. The system implements a multi-tier architecture with role-based access control, supporting five distinct user roles: Administrators, Veterinarians, Receptionists, Pet Owners, and Accountants. Each role accesses a customized portal with specialized functionality while maintaining seamless data integration across the entire platform.

The system features robust multi-language support (English, Arabic, Hebrew) with complete RTL layout capabilities, AI-powered assistance tools, real-time communication systems, and

comprehensive financial management modules. Built on modern web technologies, the platform ensures scalability, security, and optimal user experience across all device types.

4.2 Authentication and Security System

Authentication Framework

The system implements a sophisticated two-step authentication process where users first authenticate with email and password credentials, then are automatically redirected to their role-specific portal endpoint. This approach ensures that users only access functionality appropriate to their assigned roles within the veterinary ecosystem.

Key Security Features:

- JSON Web Token (JWT) management with automatic token refresh mechanisms
- Role-based access control (RBAC) with granular permission management
- Multi-language authentication interfaces supporting English, Arabic, and Hebrew
- Secure credential storage with "Remember Me" functionality
- Google OAuth integration for alternative authentication methods
- Comprehensive logout procedures ensuring complete session cleanup

Security Implementation

The security architecture employs multiple layers of protection including route guards that prevent unauthorized access to role-specific features. Token validation occurs automatically with sophisticated error handling for expired or invalid credentials. The system implements multi-layer validation, particularly in financial transactions where invoice IDs are validated from multiple data sources to prevent unauthorized access.

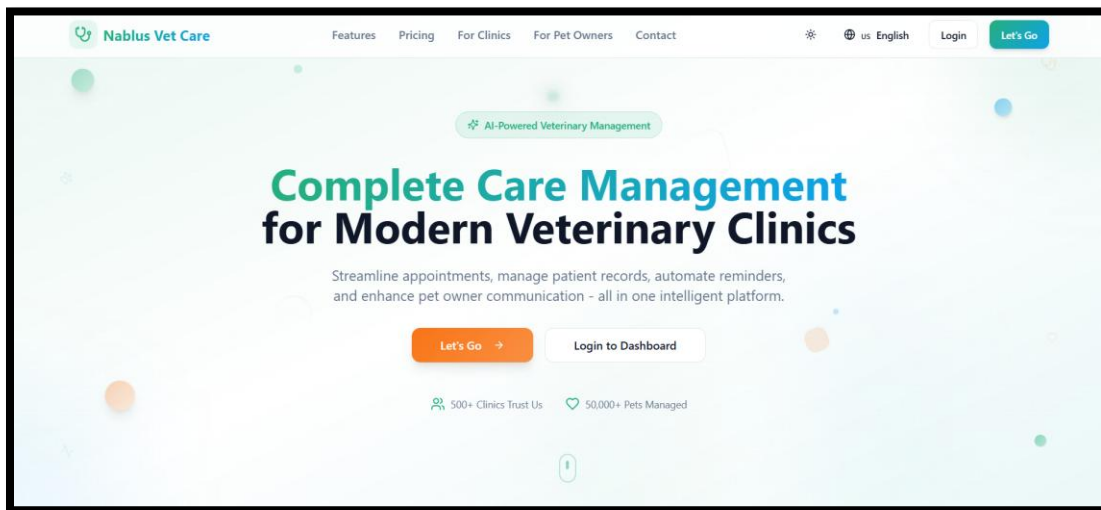


Figure 2 Nablus Vet Care web landing page.



Figure 4 Nablus Vet Care web landing page in dark mode

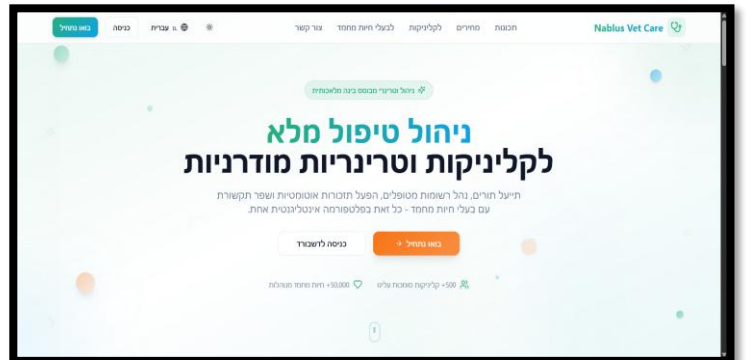


Figure 3 Nablus Vet Care web landing page in Hebrew.

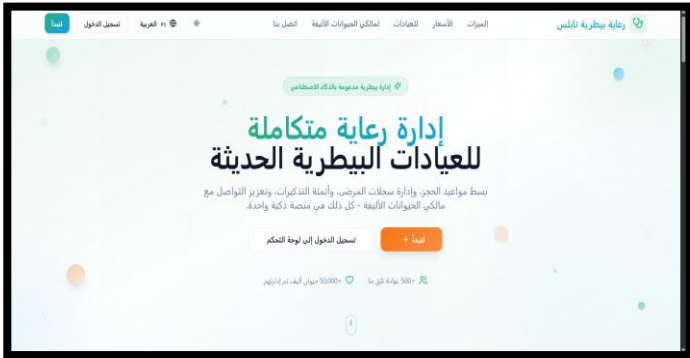


Figure 5 Nablus Vet Care web landing page in Arabic.

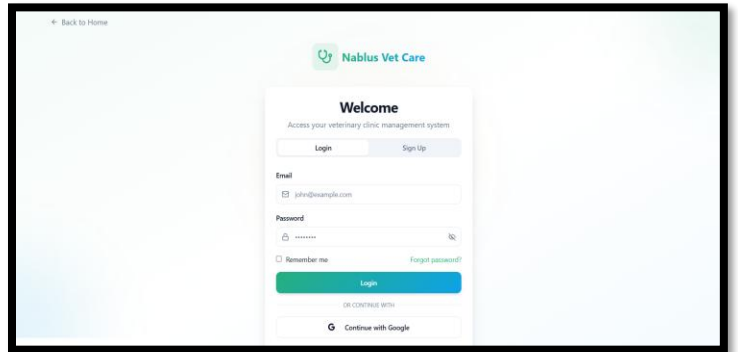


Figure 6 Nablus Vet Care web login page.

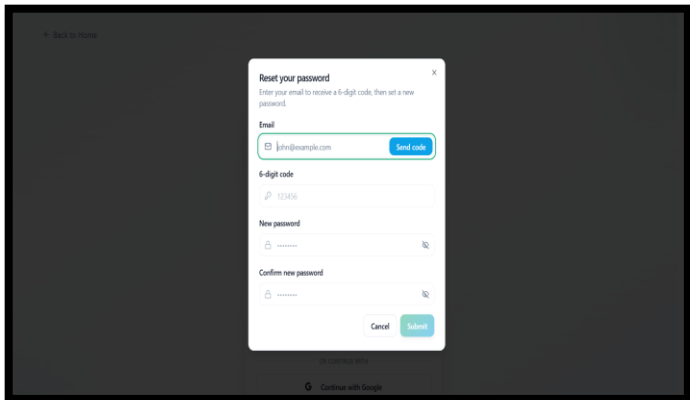


Figure 8 Nablus Vet Care web "Forgot Password" page.

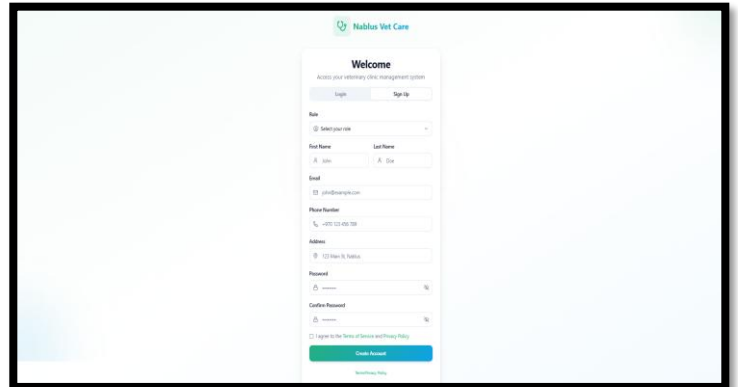


Figure 7 Nablus Vet Care web sign-up page.

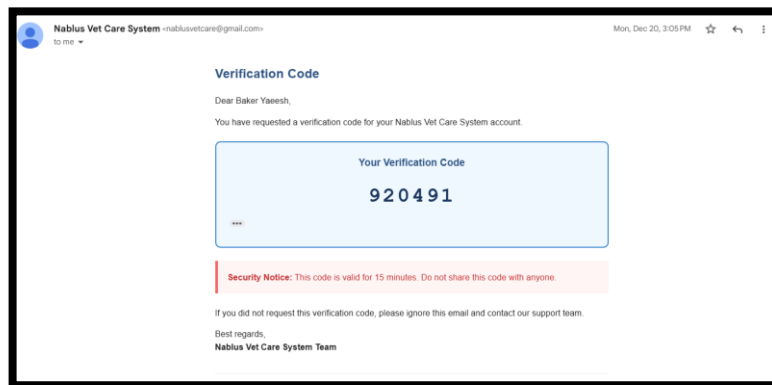


Figure 9 Nablus Vet Care web OTP verification via email.

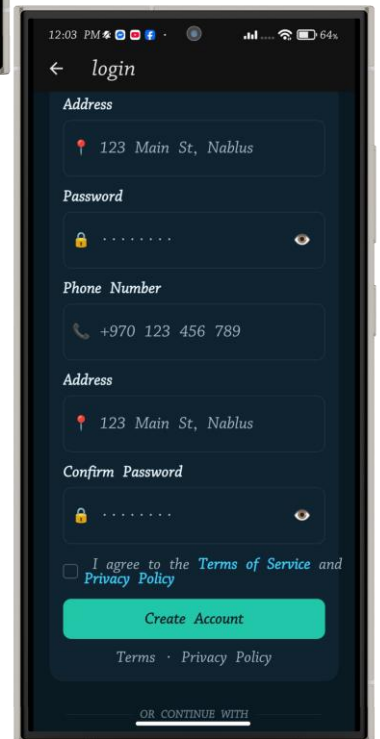
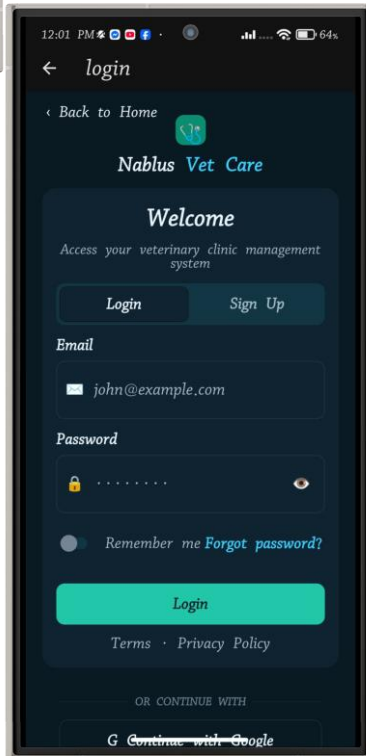
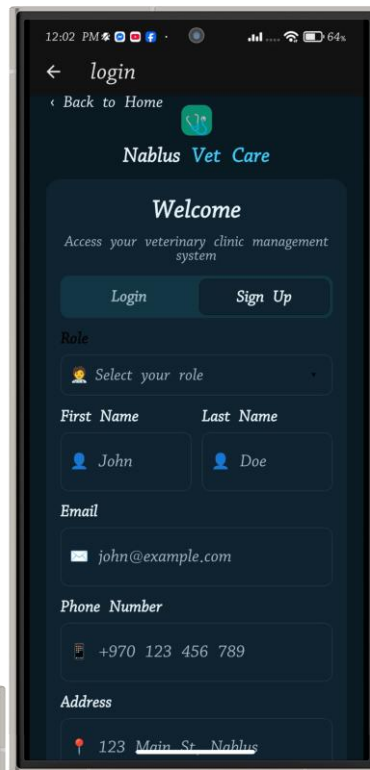


Figure 10 Nablus Vet Care mobile landing page, login, and sign-up screens.

4.3 Admin Portal Features

Dashboard and Analytics

Administrators access a comprehensive dashboard providing real-time visibility into clinic operations. The dashboard displays key performance indicators including total registered pets, active appointments, veterinarian availability, staff counts, and system users. Interactive visualization components present:

- User and pet registration trends with 12-month historical view
- Pet species distribution analysis through intuitive bar charts
- Monthly registration analytics for strategic planning
- Live appointment tracking with status indicators
- Recent financial transactions and invoice summaries

User Management System

The admin portal includes complete user lifecycle management capabilities supporting all five system roles. Administrators can create, edit, and manage user accounts with precise role assignment and account status control. The user directory provides comprehensive overviews with role-based filtering and search functionality, enabling efficient user administration across the entire organization.

Appointment Oversight

Administrators maintain system-wide visibility into appointment scheduling with capabilities to view, modify, and reassign appointments as needed. The system tracks appointment status through complete lifecycle management from scheduled through confirmed, in-progress, completed, and cancelled states. Advanced analytics provide insights into appointment patterns and veterinarian workload distribution.

Comprehensive Management Modules

The admin portal integrates several specialized management modules:

- **Veterinarian Management:** Complete vet profiles with specialization tracking across 16 medical disciplines, performance metrics, and license management
- **Pet and Owner Management:** Full registry of pets with medical history access and owner subscription status monitoring
- **Financial Oversight:** Revenue analytics, expense tracking, invoice management, and subscription revenue monitoring

- **Staff Administration:** Role assignment, performance tracking, and compensation management for support staff
- **System Configuration:** Performance analytics, data export capabilities, backup management, and audit logging

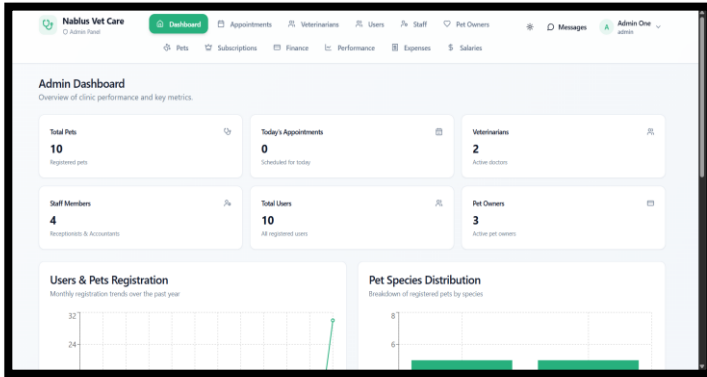


Figure 12 Nablus Vet Care web admin dashboard screen.

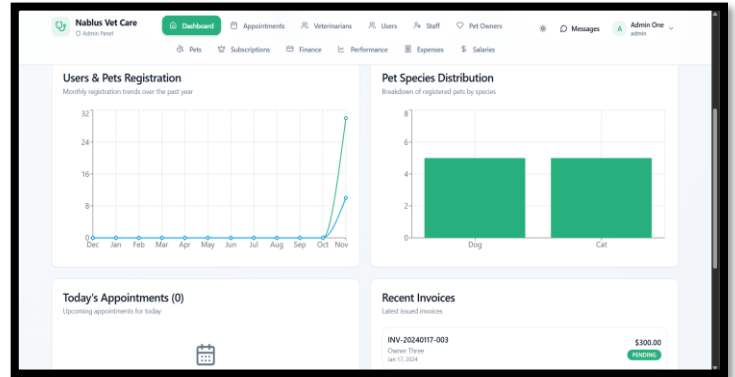


Figure 11 Nablus Vet Care web admin dashboard screen Cont.

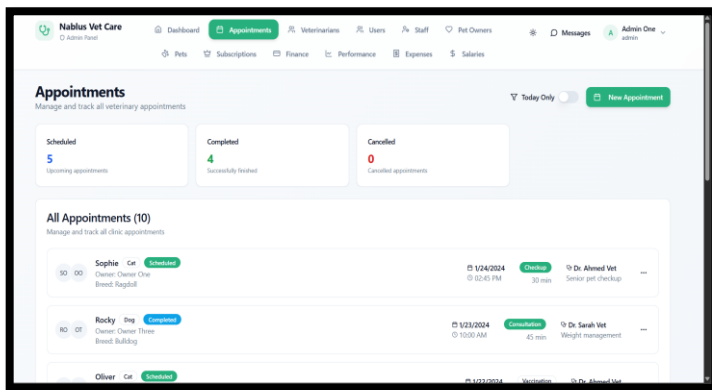


Figure 13 Nablus Vet Care web admin appointments screen.

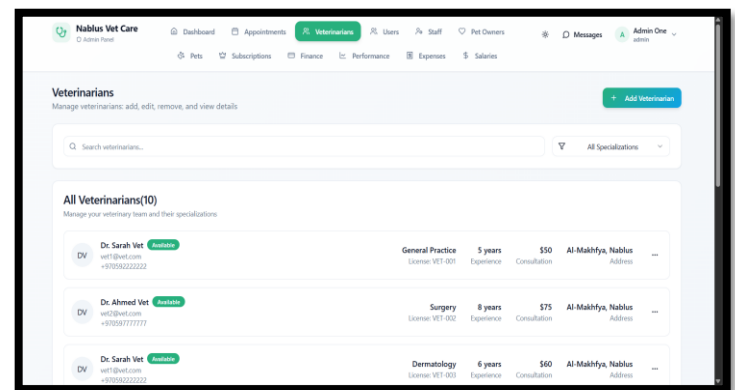


Figure 14 Nablus Vet Care web admin veterinarians screen.

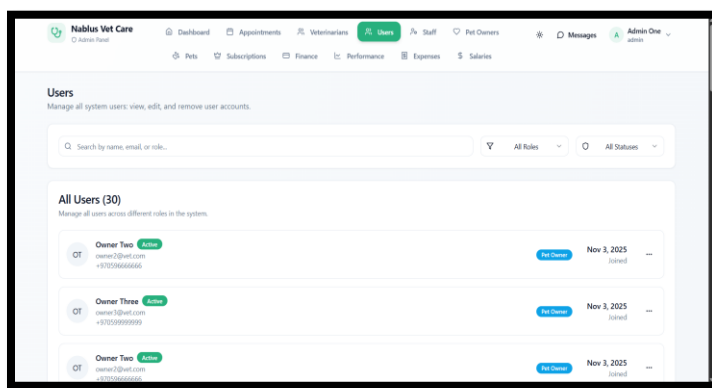


Figure 15 Nablus Vet Care web admin users screen.

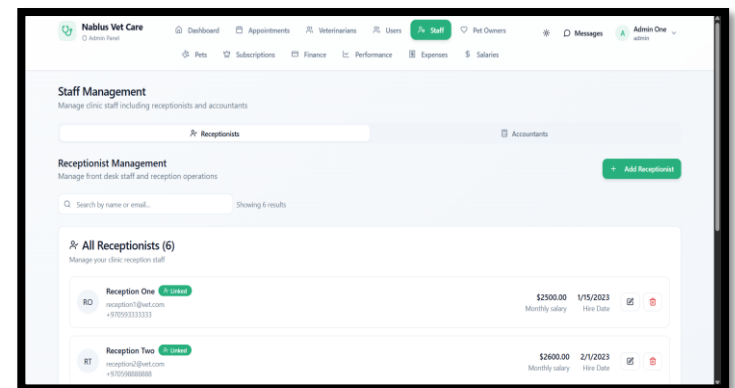


Figure 16 Nablus Vet Care web admin staff screen.

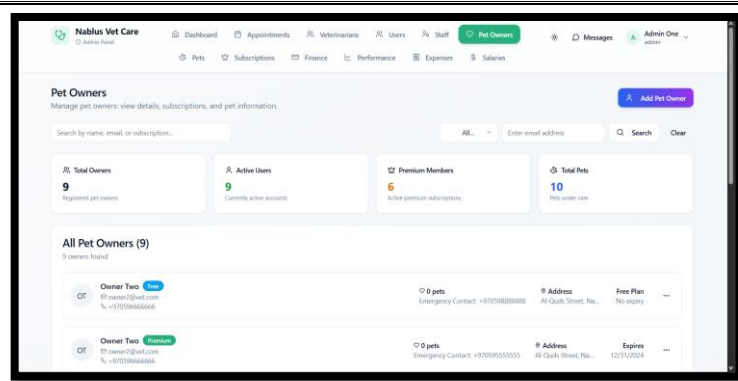


Figure 17 Nablus Vet Care web admin pets screen.

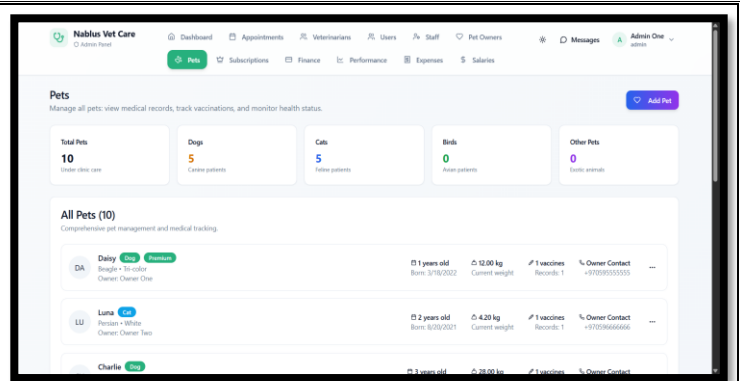


Figure 18 Nablus Vet Care web admin pet owners screen.

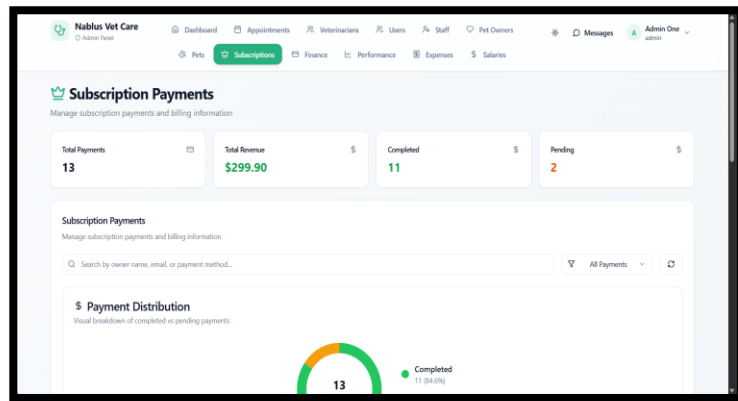


Figure 19 Nablus Vet Care web admin subscription payments screen.

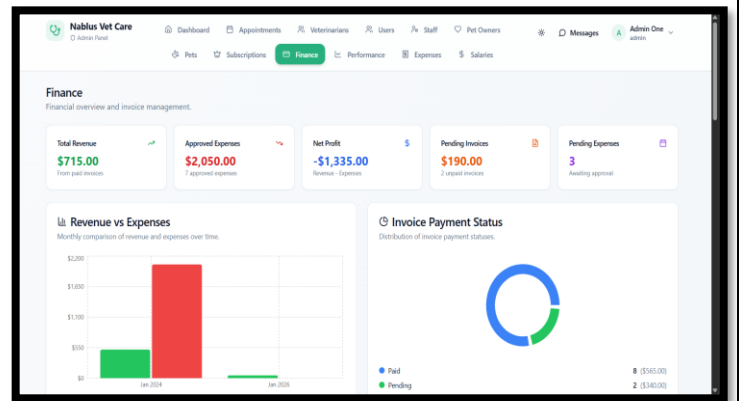


Figure 20 Nablus Vet Care web admin finance screen.

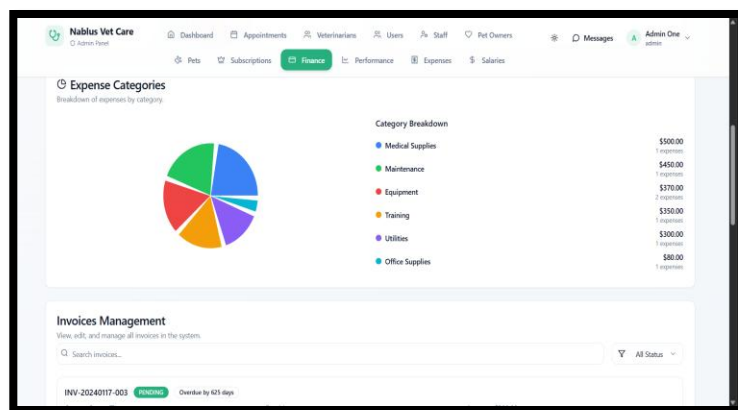


Figure 22 Nablus Vet Care web admin finance screen Cont.

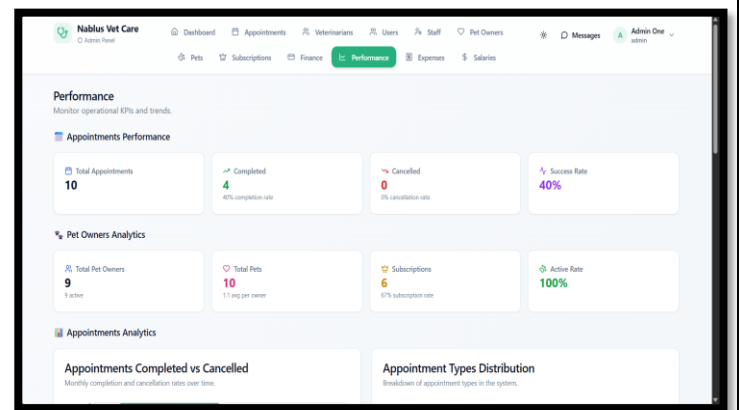


Figure 21 Nablus Vet Care web admin performance screen.

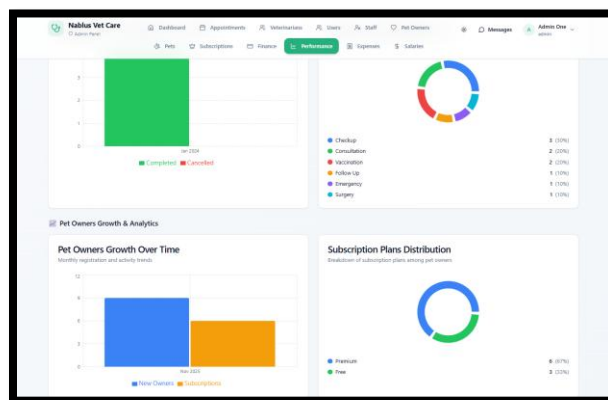


Figure 23 Nablus Vet Care web admin performance screen Cont.

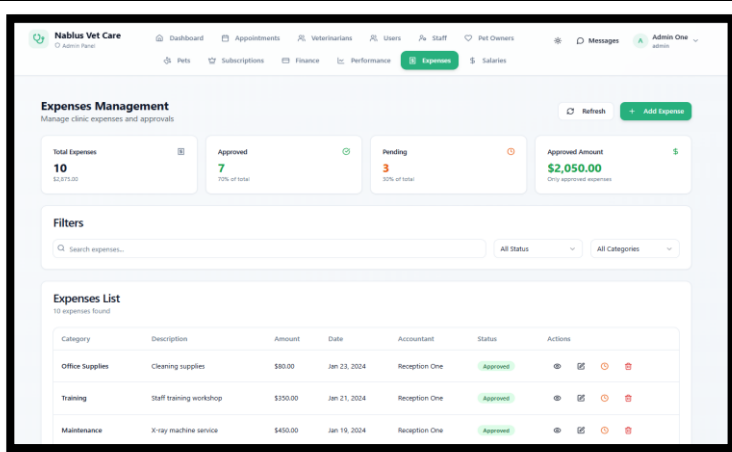


Figure 24 Nablus Vet Care web admin expenses screen.

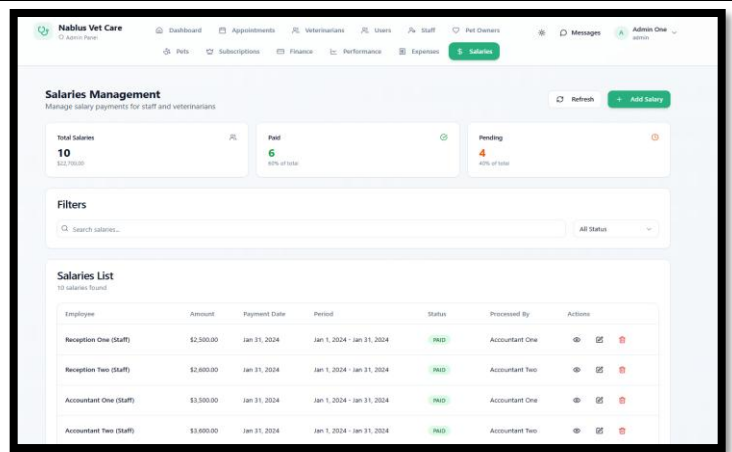


Figure 26 Nablus Vet Care web salaries screen.

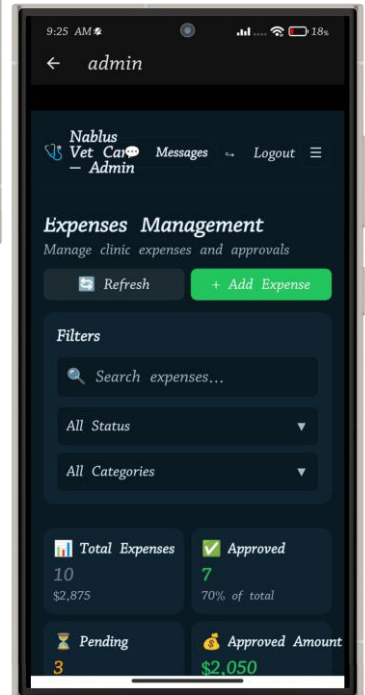
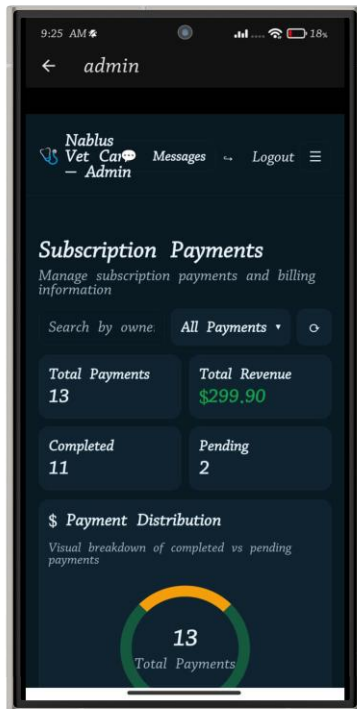
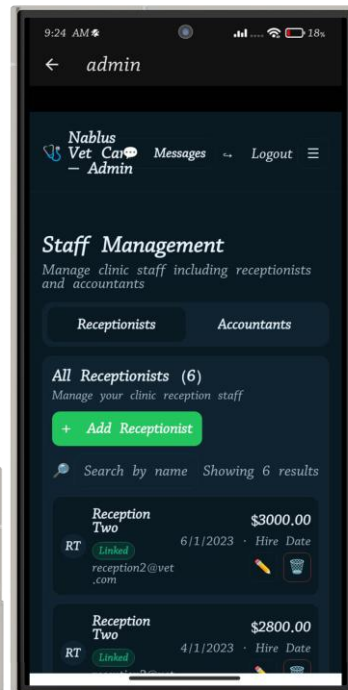
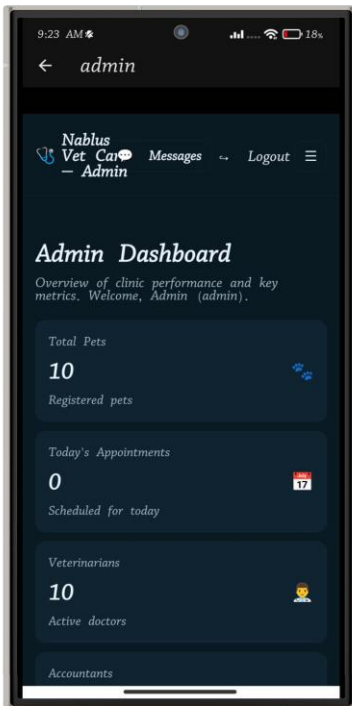


Figure 25 Sample Nablus Vet Care mobile admin portal screens.

4.4 Veterinarian Portal Features

Clinical Dashboard

Veterinarians access a specialized clinical dashboard optimized for medical workflow efficiency. The interface presents today's appointment schedule with real-time updates, patient alerts for vaccinations and medications, and quick-access statistics including daily appointment counts, pending medical records, and priority alerts. Emergency notifications are prominently displayed for urgent cases requiring immediate attention.

Appointment Management

The veterinarian portal provides sophisticated appointment management tools supporting real-time appointment loading and status control. Veterinarians can accept appointments (transitioning from scheduled to confirmed status), cancel with detailed notes, or reschedule using an intuitive date and time picker interface. Multi-status filtering enables efficient viewing of appointments by current state, while detailed appointment views provide comprehensive patient and owner information.

Medical Records System

A comprehensive medical records management system enables veterinarians to create detailed clinical documentation including:

- Structured diagnosis tracking with symptom analysis
- Complete treatment planning with protocol documentation
- Medication management with prescription tracking
- Vaccination records with immunization scheduling
- Clinical observations and progress notes

The system maintains complete medical history access for each patient, supporting informed clinical decision-making and continuity of care across multiple visits.

AI-Powered Clinical Support

Integration of veterinary AI assistance provides clinical decision support through:

- Medical consultation AI offering professional guidance
- Diagnostic assistance with evidence-based suggestions
- Treatment recommendations aligned with current standards
- Symptom analysis and evaluation tools
- Instant medical information access with professional disclaimers

Additional Clinical Tools

The portal includes prescription management with digital prescription creation, dosage management, and PDF export capabilities. Pet files management supports medical document storage, image management for diagnostic visuals, and organized file classification. Communication tools enable real-time messaging with pet owners and staff, while analytics provide individual performance metrics and patient statistics.

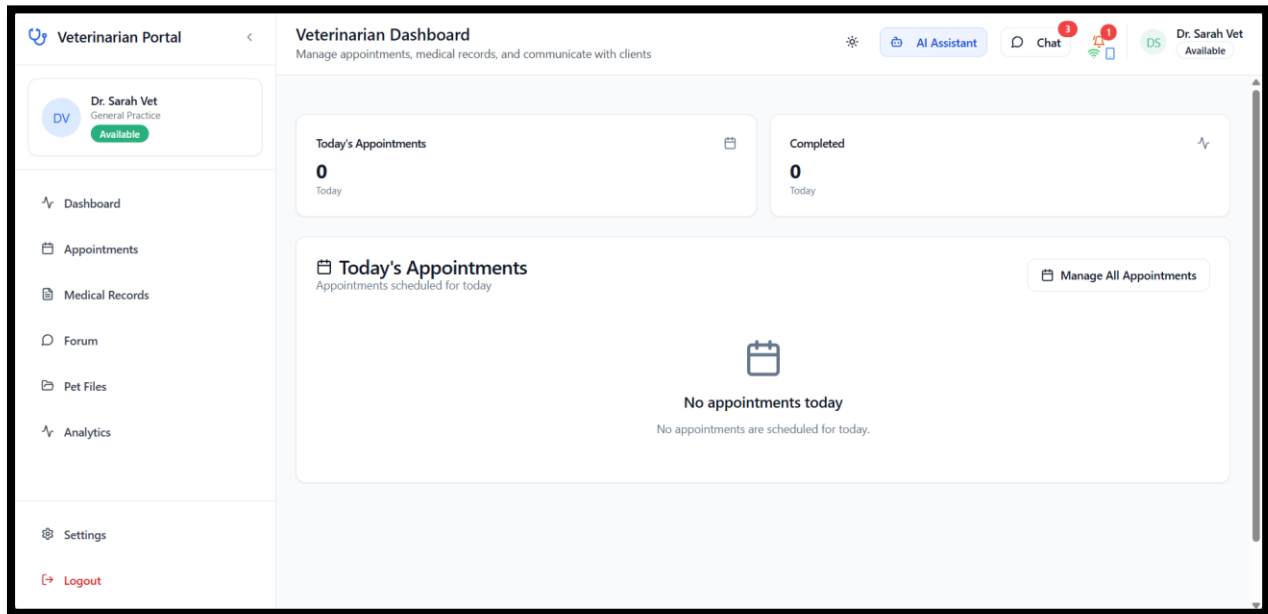


Figure 27 Nablus Vet Care web veterinarian dashboard.

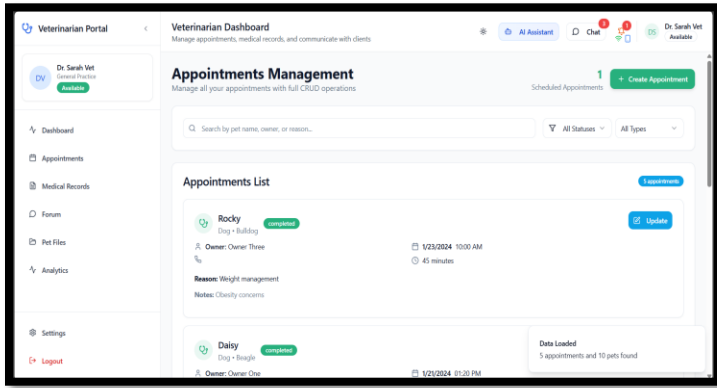


Figure 28 Nablus Vet Care web veterinarian appointments management screen.

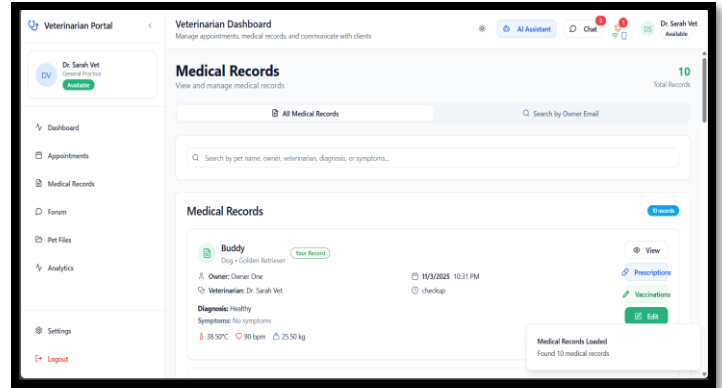


Figure 29 Nablus Vet Care web veterinarian medical records screen.

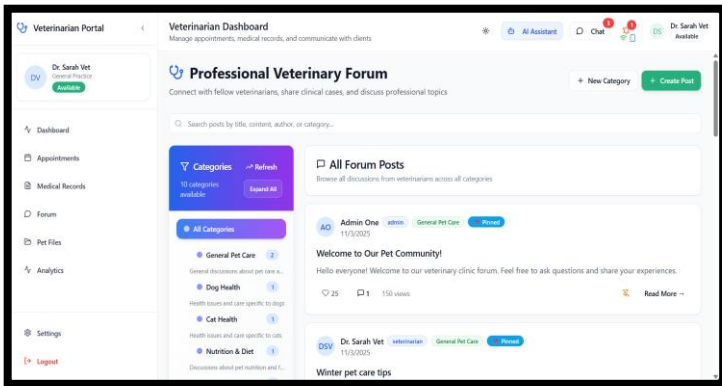


Figure 29 Nablus Vet Care web veterinarian forum screen.

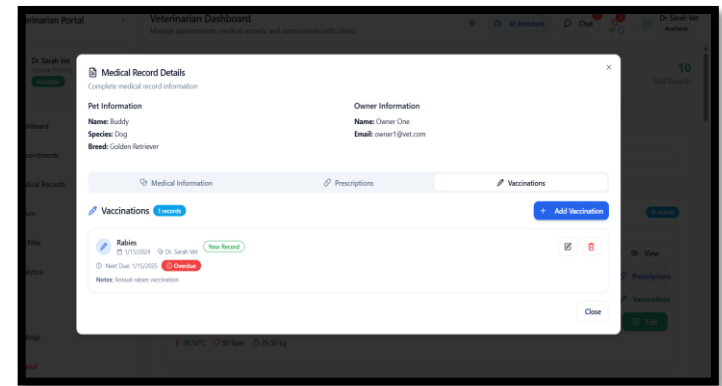


Figure 28 Nablus Vet Care web veterinarian medical records details screen.

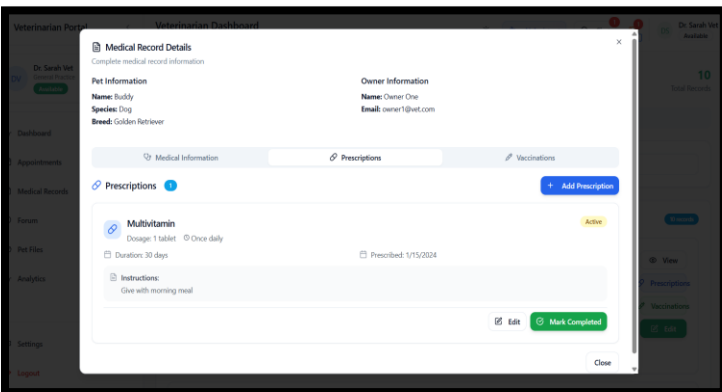


Figure 30 Nablus Vet Care web veterinarian medical records Cont.

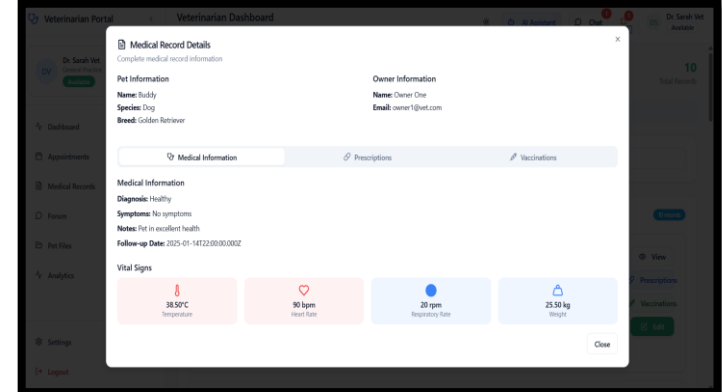


Figure 31 Nablus Vet Care web veterinarian medical records Cont.

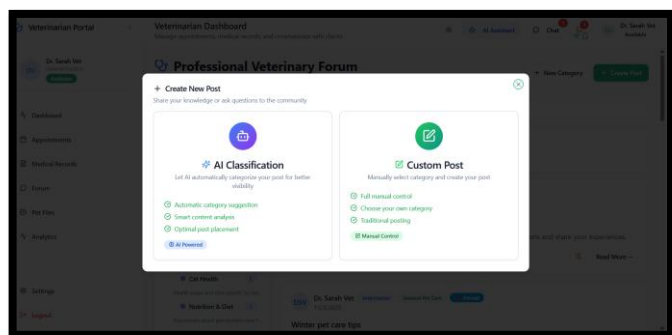


Figure 32 Nablus Vet Care web veterinarian creating a post using AI classification or custom input.

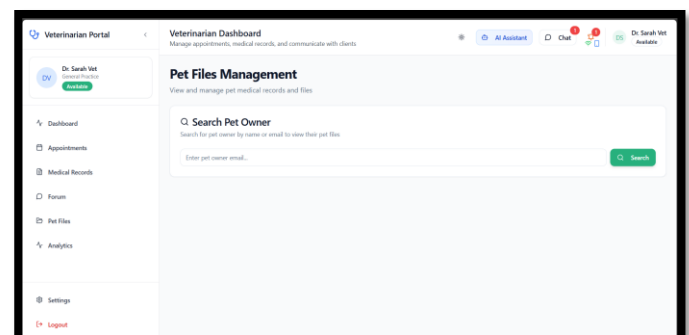


Figure 33 Nablus Vet Care web veterinarian pet files management.

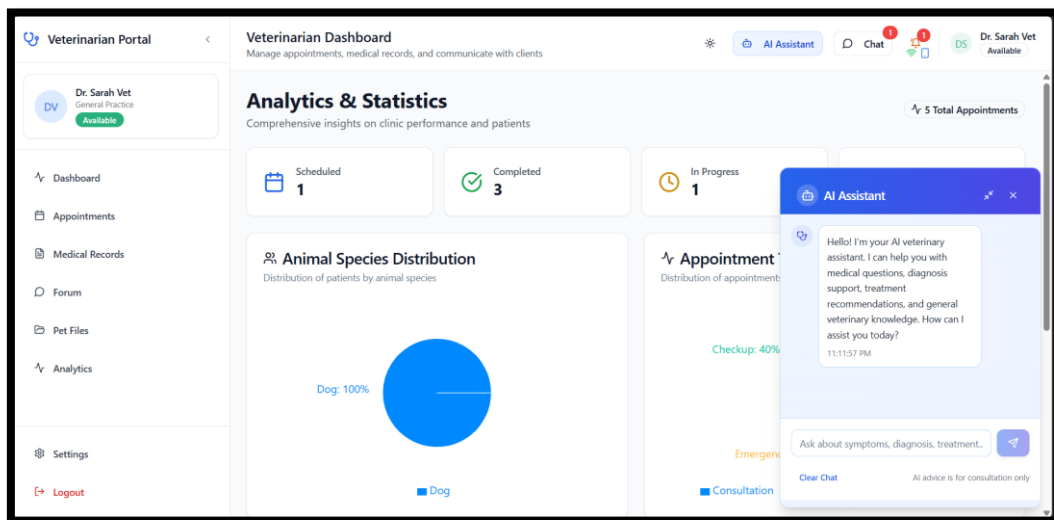
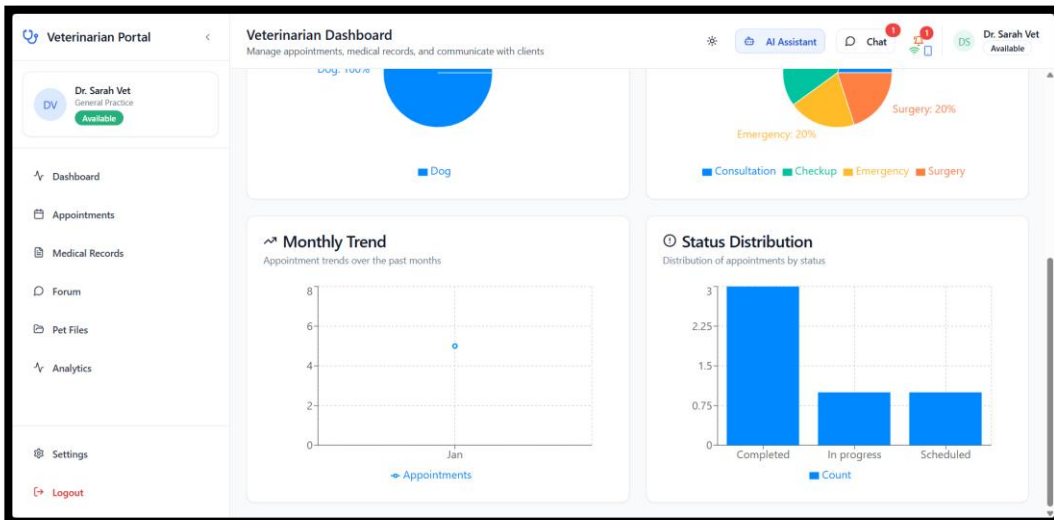
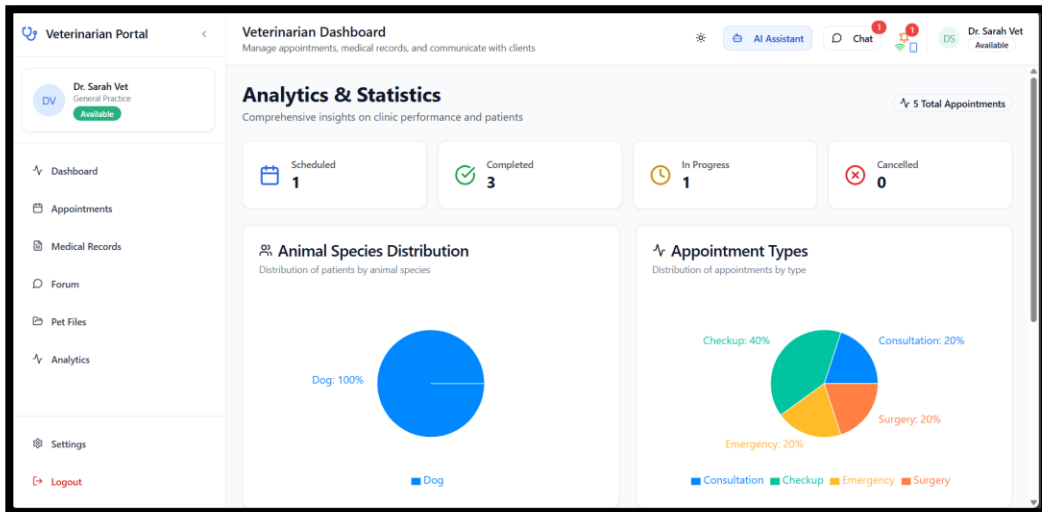


Figure 34 Nablus Vet Care veterinarian analytics screen.

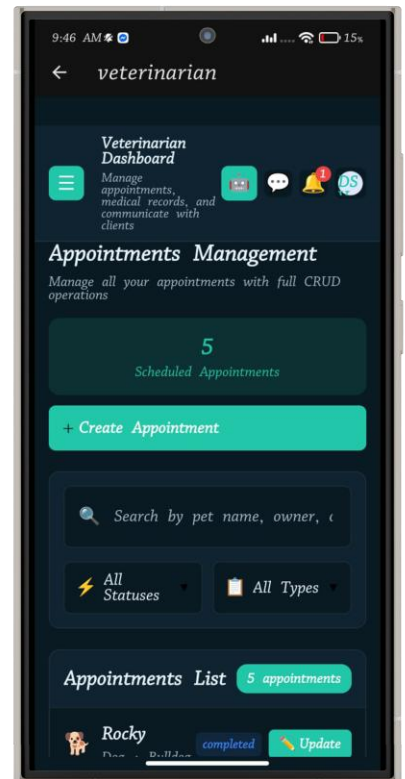
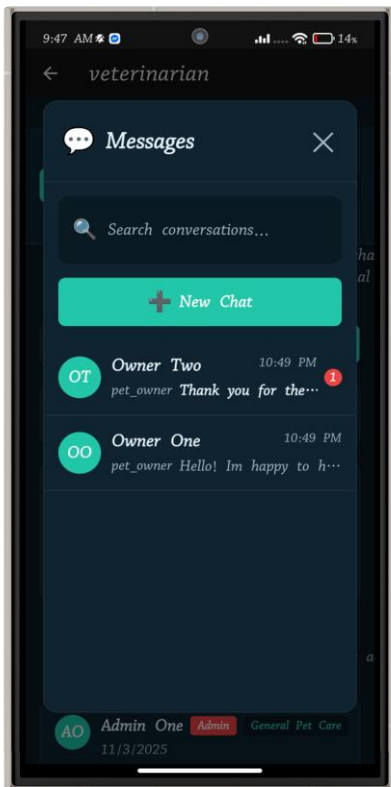
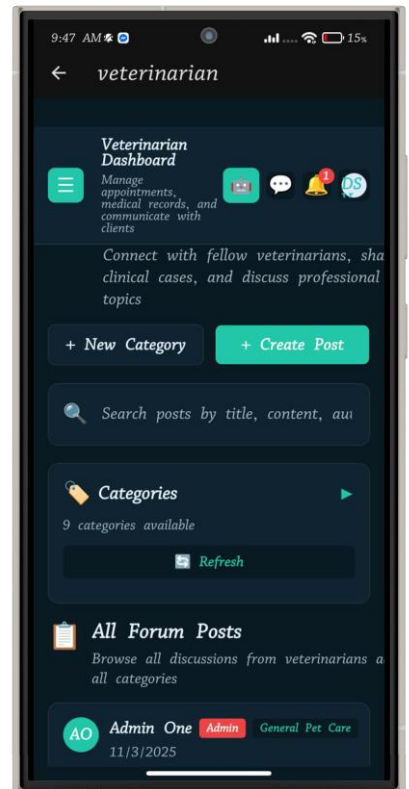
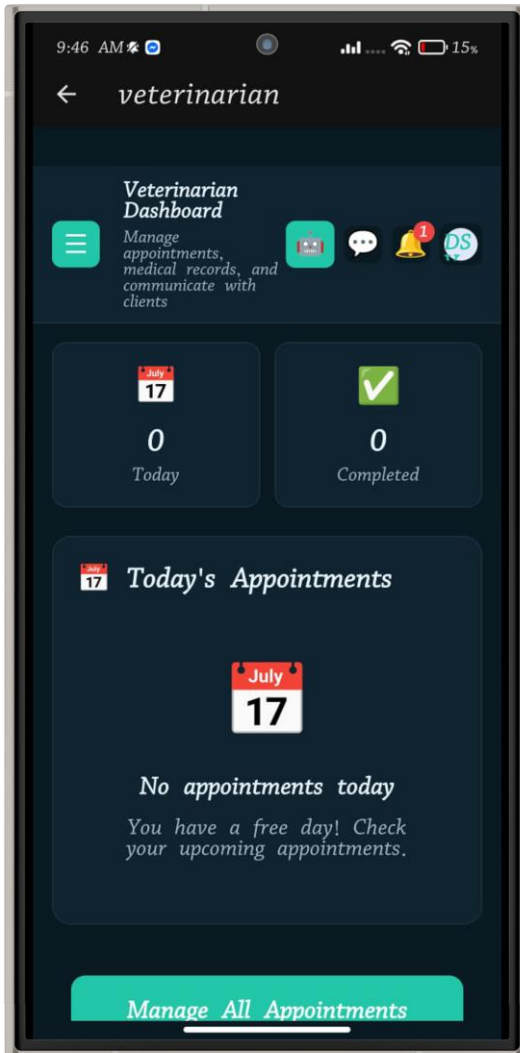


Figure 35 Sample Nablus Vet Care mobile veterinarian portal screens.

4.5 Receptionist Portal Features

Operational Dashboard

Receptionists access an operational dashboard designed for front-desk efficiency, featuring daily appointment overviews, quick-action buttons for common tasks, and visual invoice status indicators. Real-time data updates ensure receptionists have current information for client interactions and scheduling coordination.

Appointment Scheduling System

The receptionist portal includes a comprehensive appointment booking system with advanced features:

- Intelligent appointment creation with pet and veterinarian selection
- Date picker interface with future date validation
- Time slot management from 8:00 AM to 6:30 PM
- Duration selection supporting 15-120 minute appointments
- Multi-type appointment categorization (checkup, vaccination, surgery, emergency)
- Reason for visit documentation and note-taking capabilities
- Conflict detection and resolution for scheduling optimization

Patient Registration and Management

Receptionists manage complete client onboarding processes including new pet owner registration, pet profile creation, and emergency contact setup. The system supports subscription plan assignment and maintains comprehensive owner information for efficient client service delivery.

Financial and Communication Coordination

The portal integrates payment processing capabilities with invoice generation, receipt management, and transaction history tracking. A communication hub enables real-time messaging with veterinarians and pet owners, appointment reminder systems, and phone integration with call logging. Receptionists also coordinate veterinarian schedules, manage appointment assignments, and resolve scheduling conflicts to optimize clinic operations.

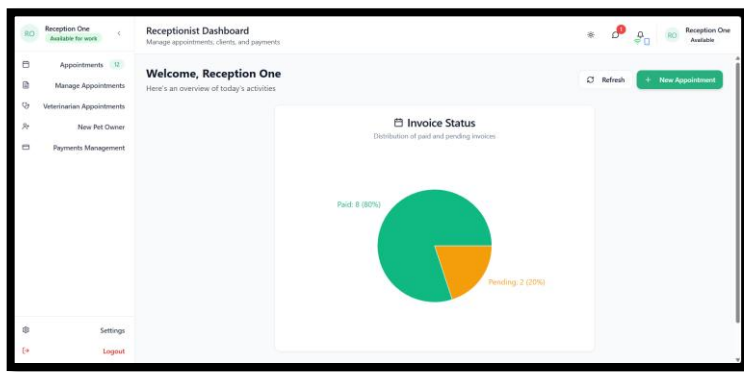


Figure 36 Nablus Vet Care web receptionist dashboard screen.

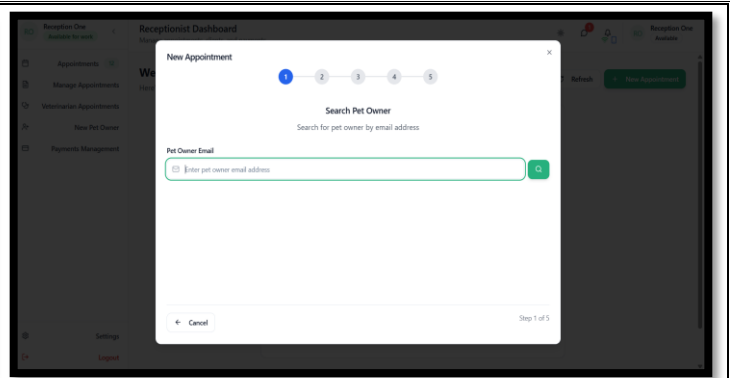


Figure 37 Nablus Vet Care web receptionist "New Appointment" screen.

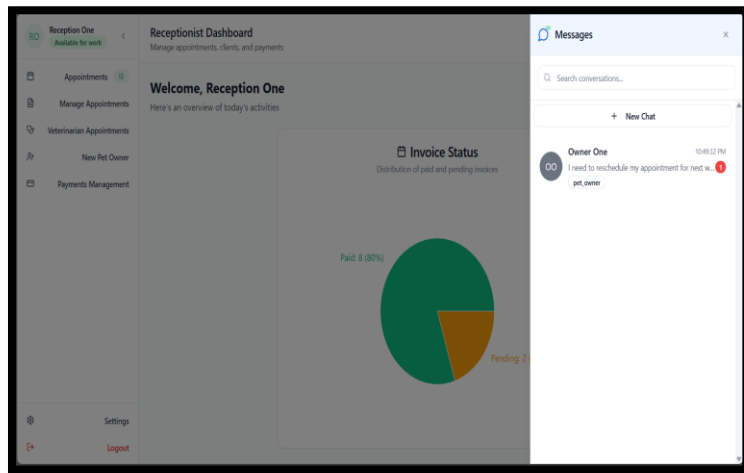


Figure 38 Nablus Vet Care messaging system (available across all portals).

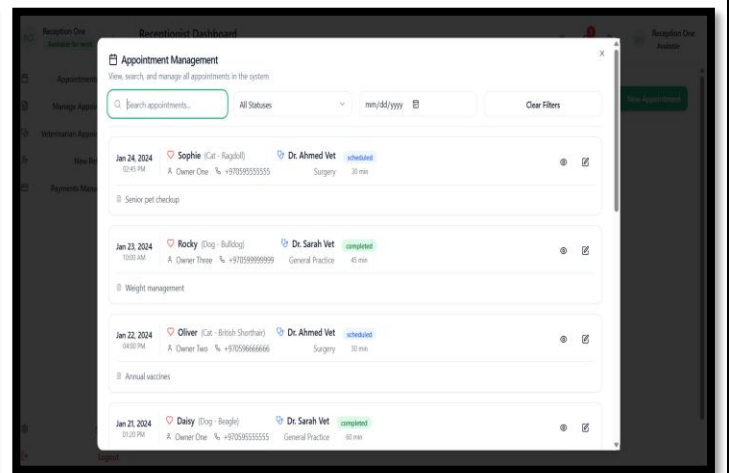


Figure 39 Nablus Vet Care web receptionist appointments management screen.

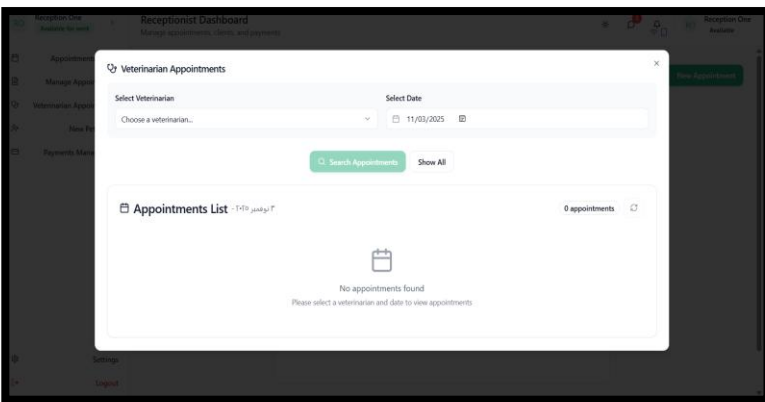


Figure 40 Nablus Vet Care web receptionist veterinarian appointments screen.

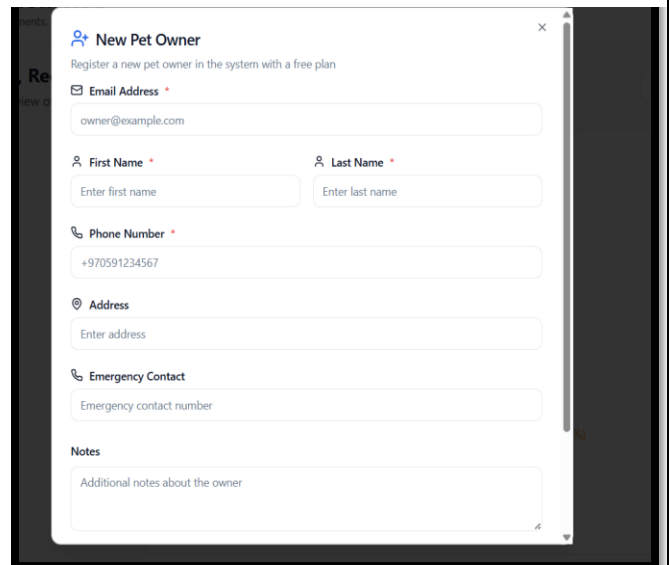


Figure 41 Nablus Vet Care web receptionist "Add New Pet Owner" screen.

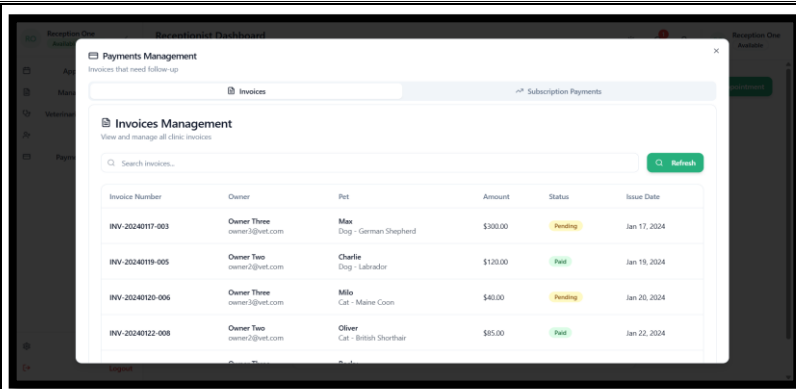


Figure 42 Nablus Vet Care web receptionist invoices management screen.

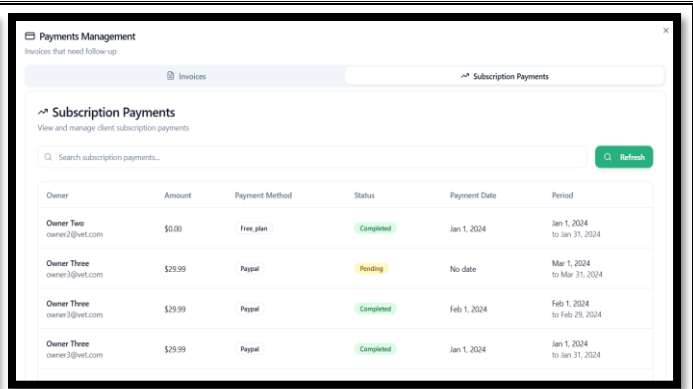


Figure 43 Nablus Vet Care web receptionist subscriptions and payments management screen.

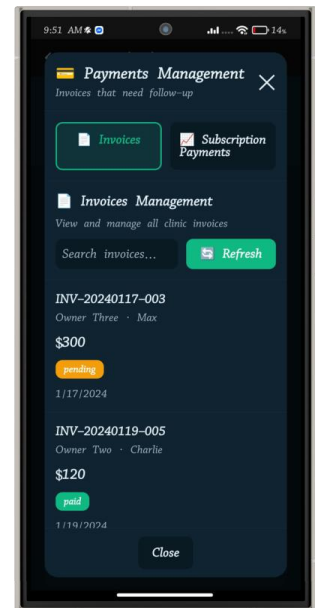
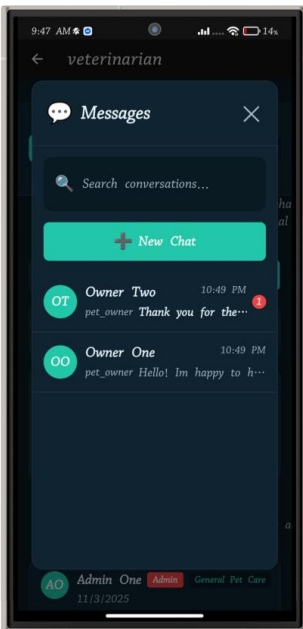
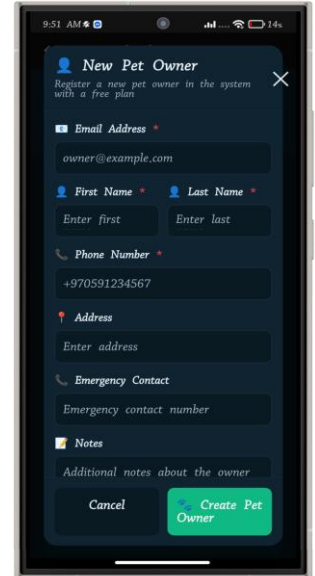
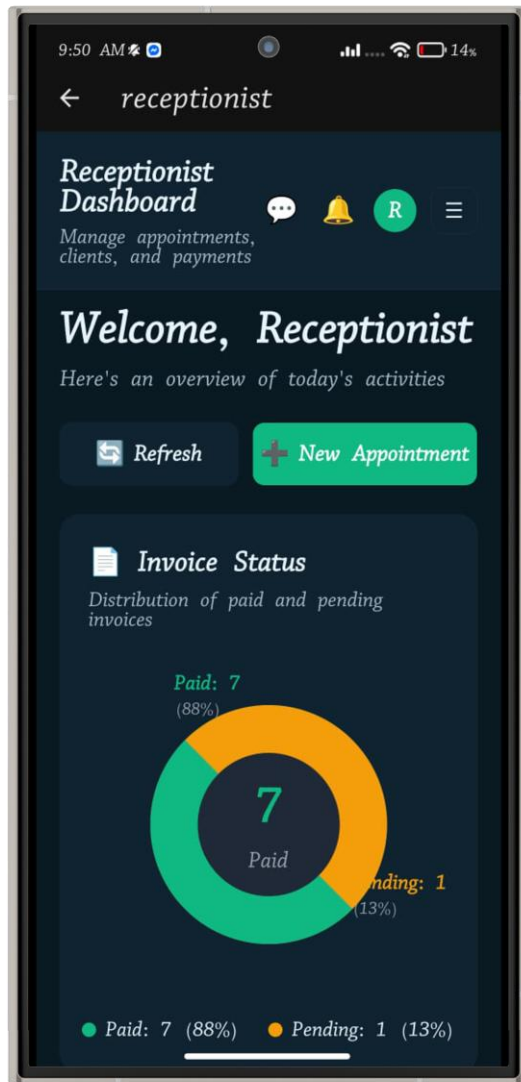


Figure 44 Sample Nablus Vet Care mobile receptionist portal screens.

4.6 Pet Owner Portal Features

Pet Management Interface

Pet owners access a user-friendly portal centered around their pets' health and care. The dashboard provides overviews of all registered pets, upcoming appointments, health alerts for vaccinations and medications, and quick-access functions for common tasks. Subscription status displays keep owners informed of their current service plan and upgrade options.

Self-Service Functionality

The portal empowers pet owners with comprehensive self-service capabilities:

- **Online Appointment Booking:** Self-service scheduling with veterinarian selection, service type choice, and available time slot booking
- **Medical Records Access:** Complete digital medical history including prescriptions, diagnostic results, and treatment plans
- **Health Tracking:** Vaccination reminders, medication tracking, and health metric monitoring
- **Document Management:** Pet photo galleries, medical document storage, and file sharing with veterinary staff

Financial Management

Pet owners can view and manage invoices, process secure online payments via Stripe integration, and monitor payment history with digital receipt access. The system supports multiple payment methods and provides financial summaries showing paid, pending, and overdue amounts.

Communication and Support

Integrated communication tools include real-time chat with veterinary staff, AI pet assistant access for care guidance, and notification systems for health alerts and appointment reminders. Voice calling capabilities enable direct communication with the clinic when needed.

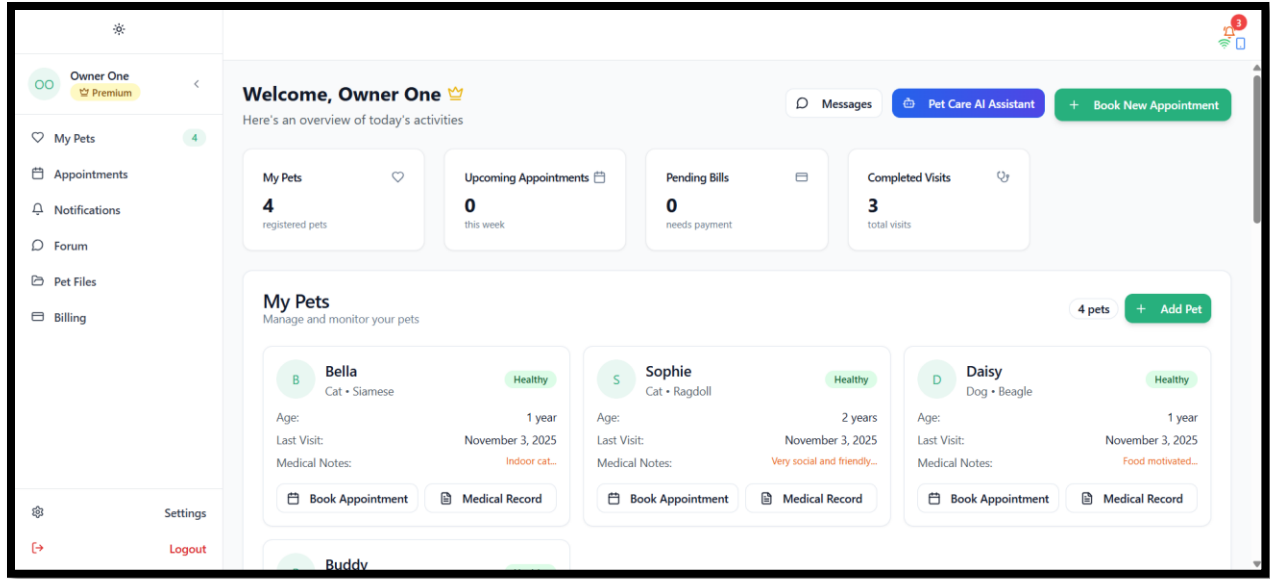


Figure 45 Nablus Vet Care web pet owner portal dashboard screen.

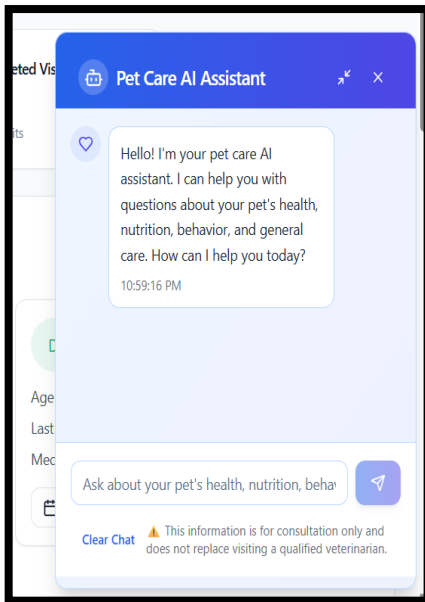


Figure 47 Nablus Vet Care web pet owner AI assistant chatbot screen.

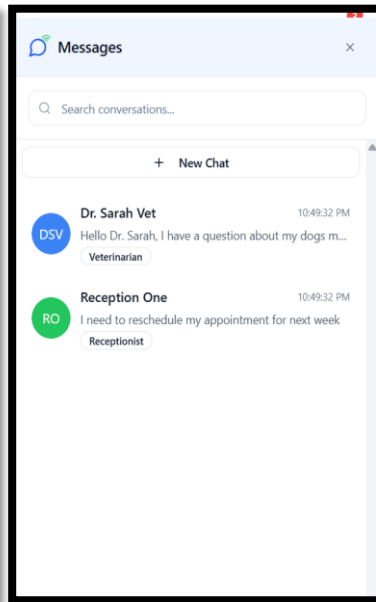


Figure 48 Nablus Vet web Care pet owner chat screen.

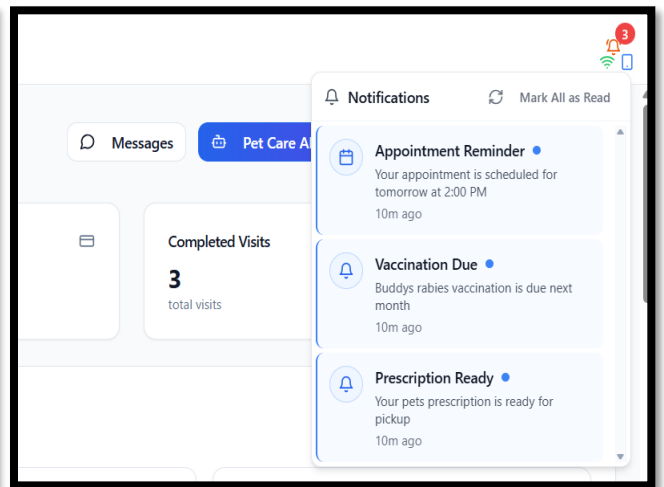


Figure 46 Nablus Vet Care web pet owner notifications screen.

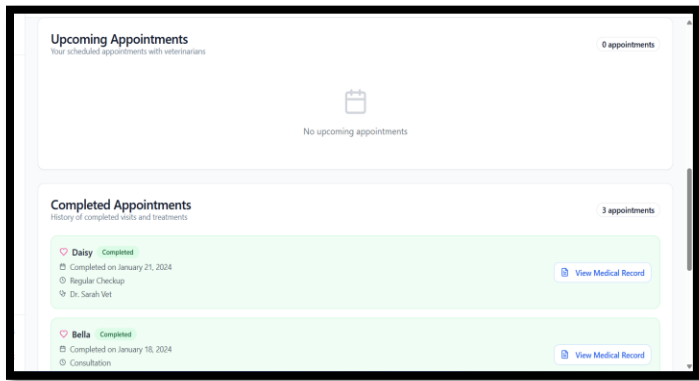


Figure 49 Nablus Vet Care web pet owner upcoming appointments screen.

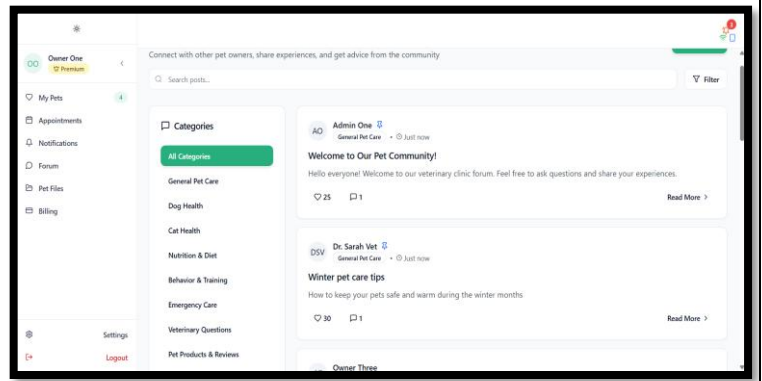


Figure 50 Nablus Vet Care web pet owner forum screen.

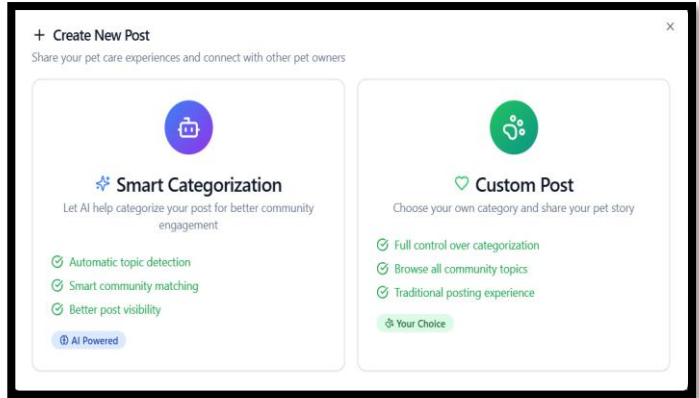


Figure 51 Nablus Vet Care web pet owner creating a forum post with AI classification or custom category.

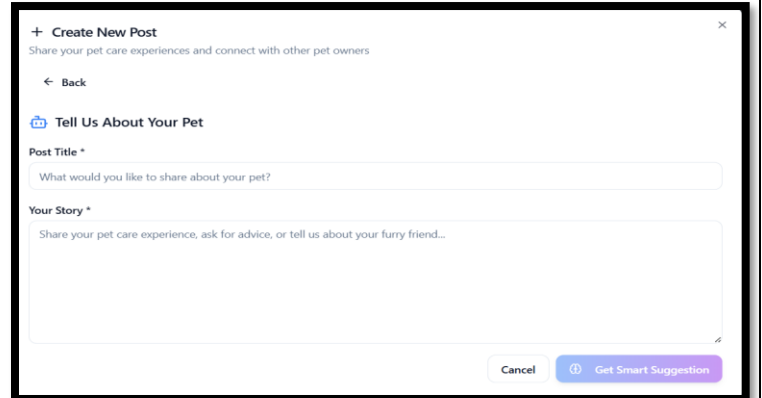


Figure 52 Nablus Vet Care web pet owner creating a post using AI screen.

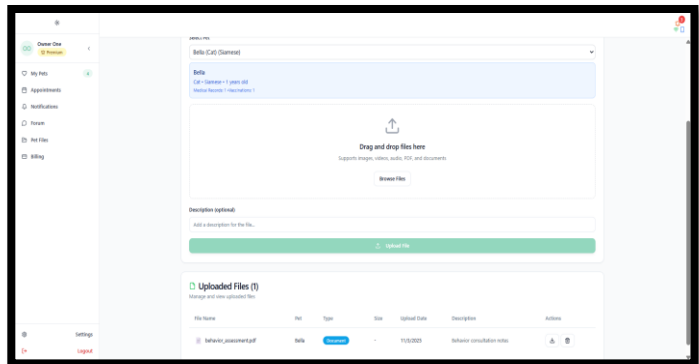


Figure 54 Nablus Vet Care web pet owner portal – pet files screen.

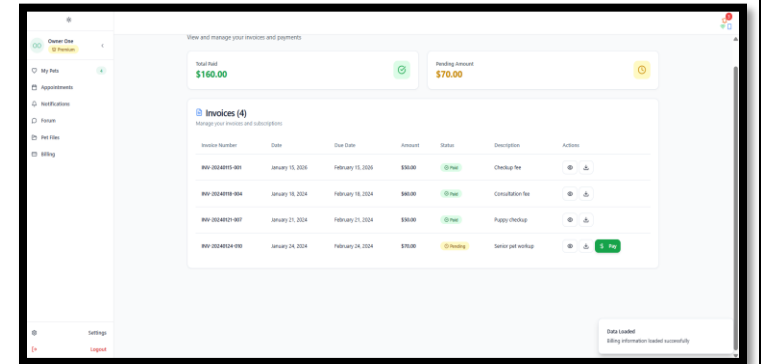


Figure 53 Nablus Vet Care web pet owner invoices screen.

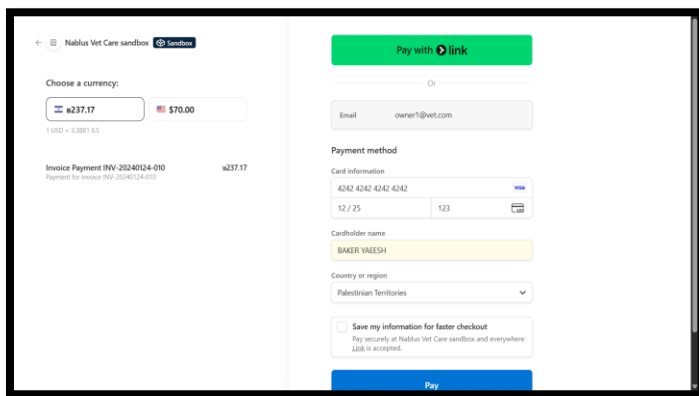


Figure 56 Nablus Vet Care web pet owner paying an invoice using Stripe.

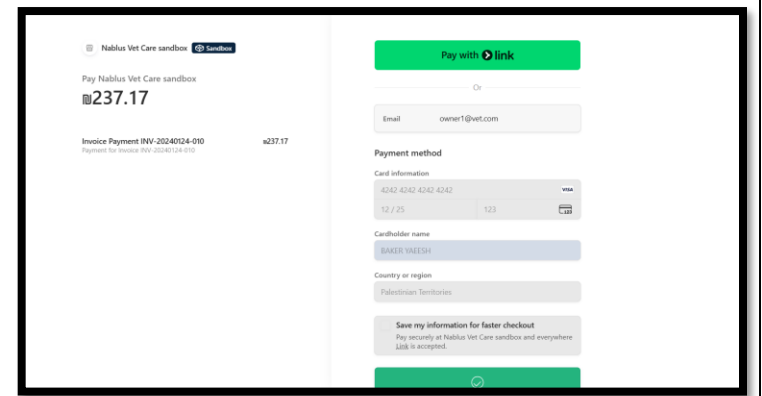


Figure 55 Nablus Vet Care web pet owner payment successful screen.

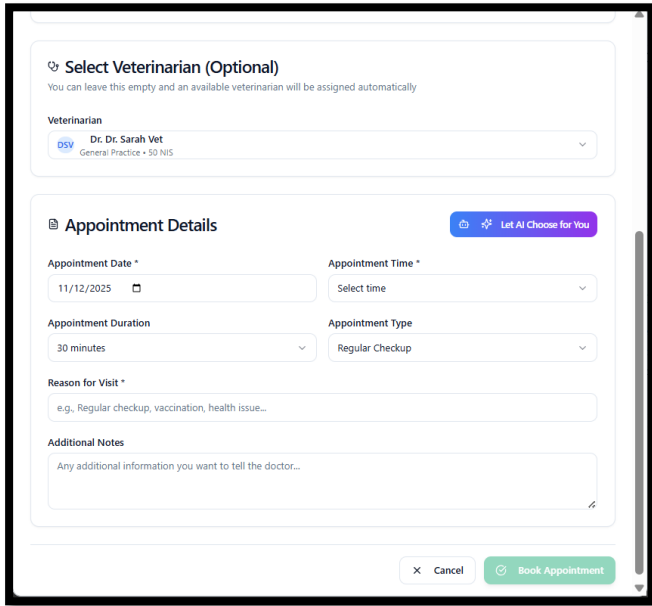


Figure 57 Nablus Vet Care web pet owner booking an appointment screen.

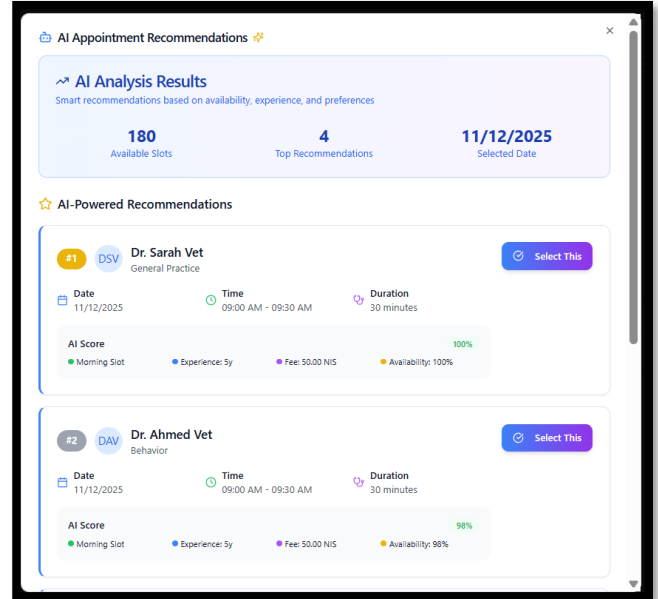


Figure 58 Nablus Vet Care web pet owner selecting an appointment from AI-suggested options.

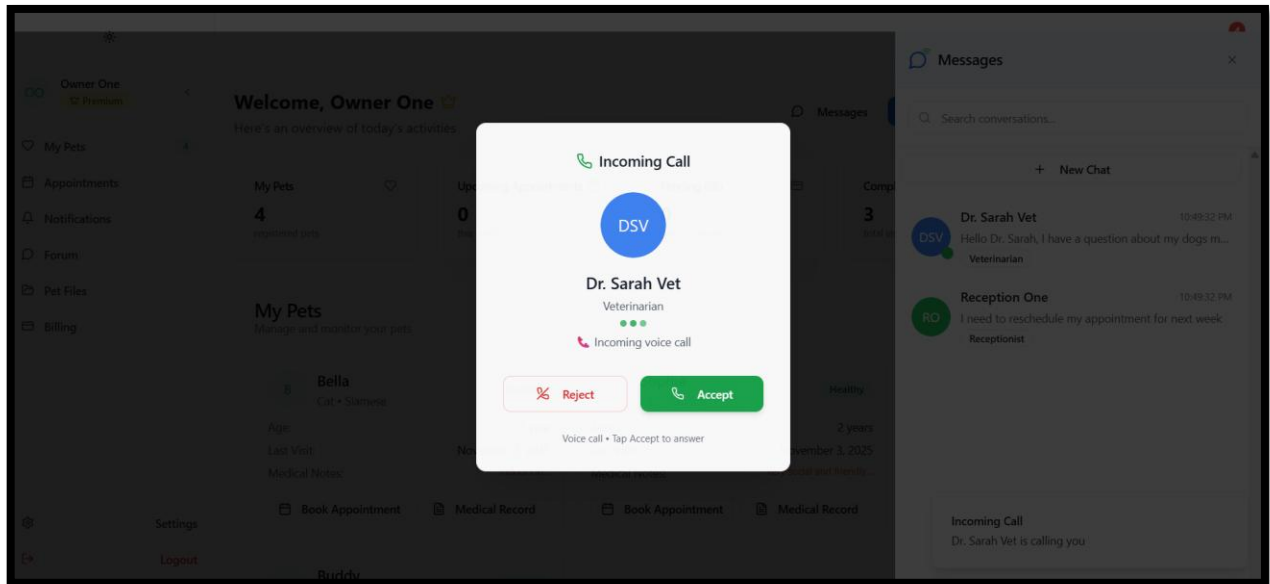


Figure 59 Nablus Vet Care web incoming call from veterinarian (available across all portals).

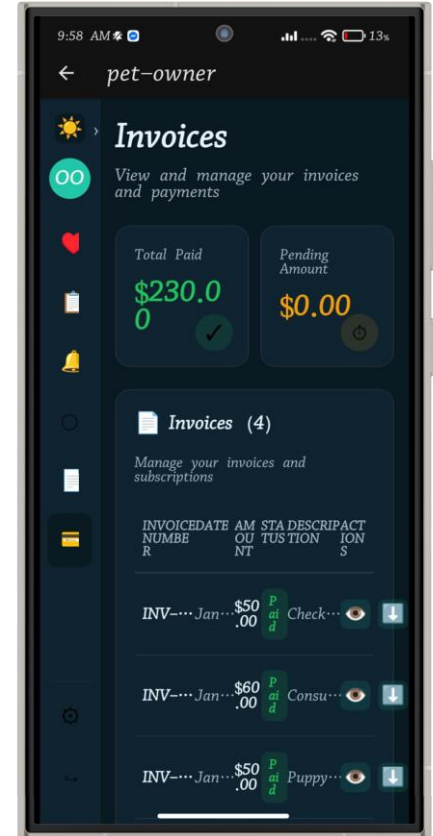
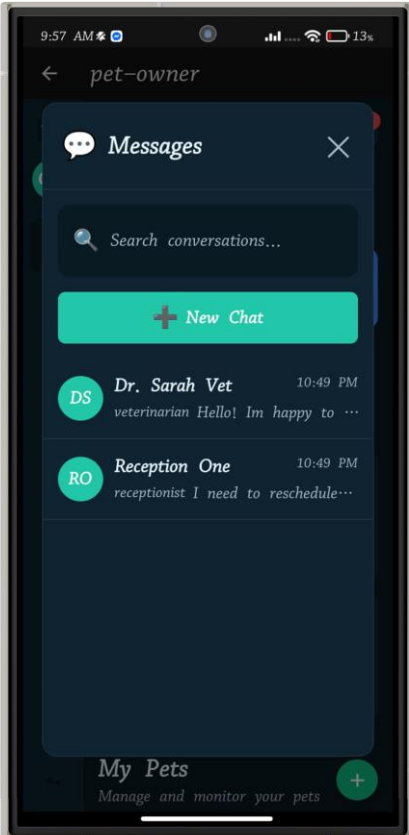
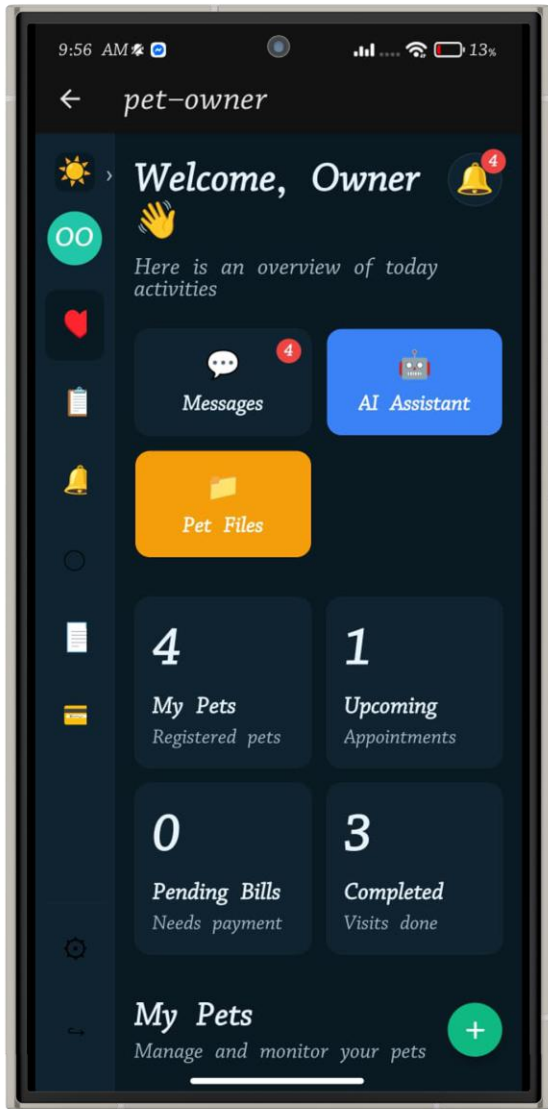
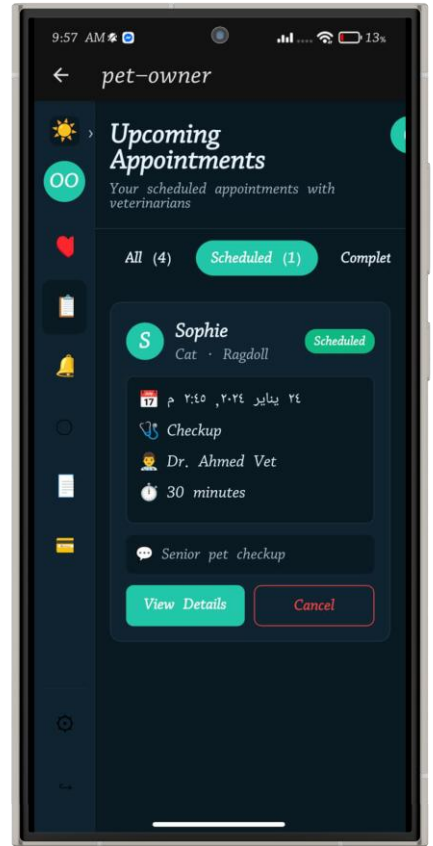


Figure 60 Sample Nablus Vet Care mobile pet owner portal screens.

4.7 Accountant Portal Features

Financial Dashboard

Accountants access a specialized financial dashboard with real-time revenue analytics, key financial metrics, and interactive visualization tools. The interface presents income, expense, and net profit displays with mobile-optimized charting capabilities. Color-coded summary cards highlight critical financial indicators, while monthly trend analysis supports strategic financial planning.

Comprehensive Financial Management

The accountant portal integrates multiple financial management modules:

- **Invoice Management:** Complete invoice lifecycle management with status tracking, client history, and payment processing
- **Subscription Revenue Tracking:** Monthly subscription income analysis with payment method breakdowns and revenue forecasting
- **Expense Management:** Operational cost tracking with category management, approval workflows, and receipt management
- **Salary Administration:** Staff and veterinarian payroll processing with payment scheduling, tax management, and individual payment records

Reporting and Analytics

Advanced financial reporting capabilities include profit and loss statements, cash flow analysis, executive financial dashboards, tax preparation reports, and comprehensive audit trails. The system provides detailed financial analytics supporting business intelligence and strategic decision-making.

Mobile-Optimized Interface

The accountant portal features a mobile-first design with touch-optimized interfaces, responsive financial visualizations, card-based mobile layouts, and intuitive navigation patterns ensuring accessibility across all device types.

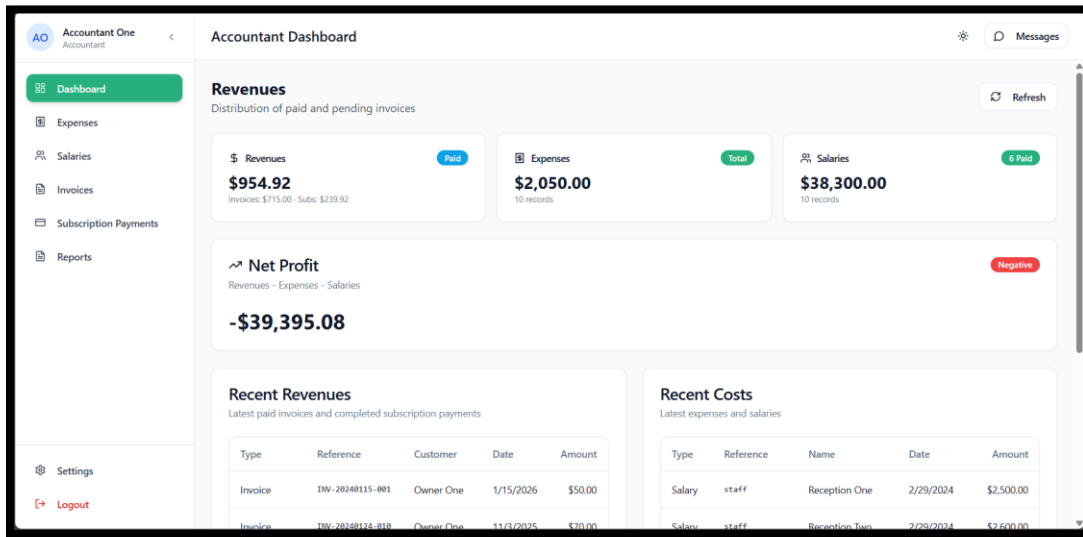


Figure 61 Nablus Vet Care web accountant portal dashboard.

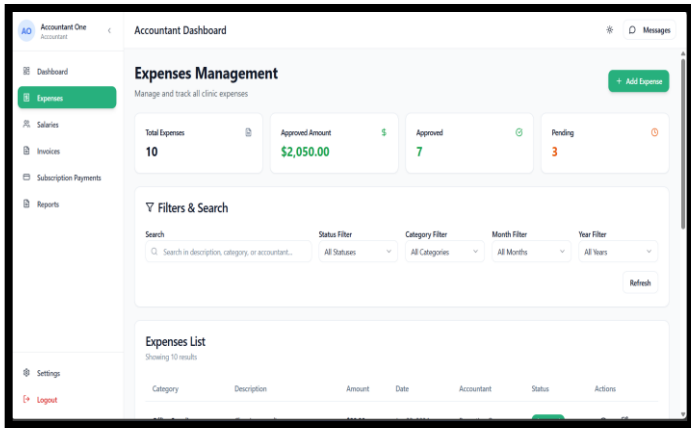


Figure 63 Nablus Vet Care web accountant expenses management screen.

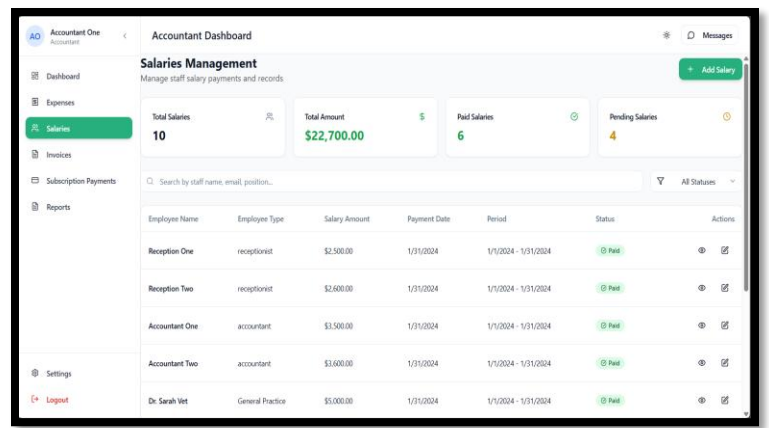


Figure 62 Nablus Vet Care web accountant salaries management screen.

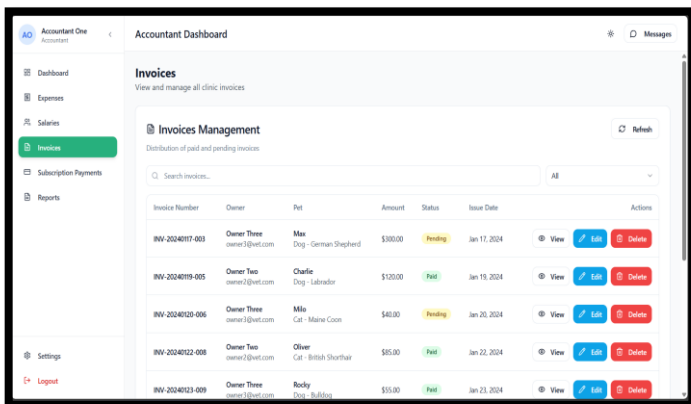


Figure 65 Nablus Vet Care web accountant invoices management screen.

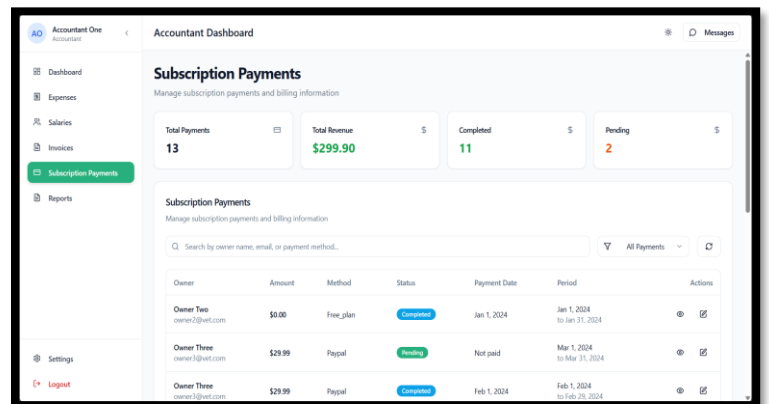


Figure 64 Nablus Vet Care web subscription payment management screen.

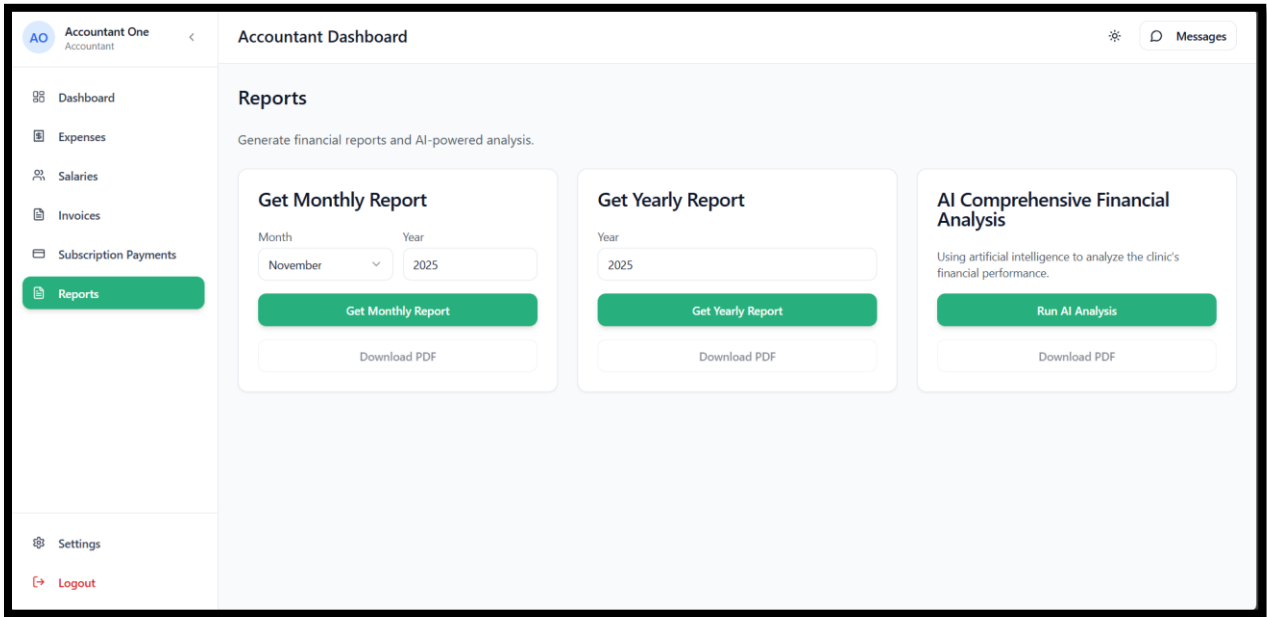


Figure 66 Nablus Vet Care web AI-generated reports screen.

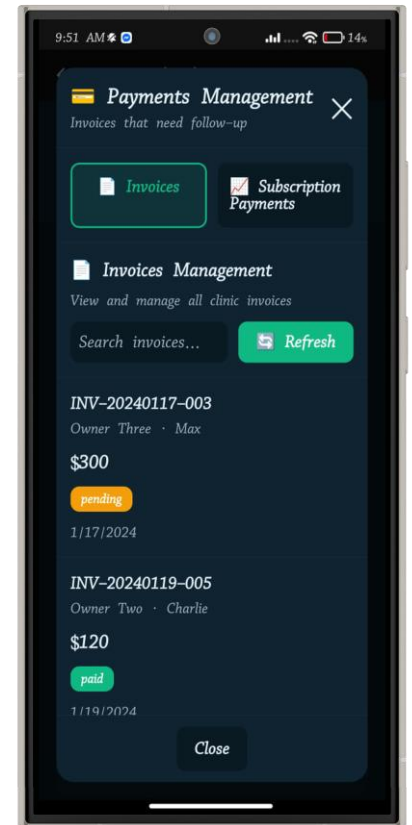
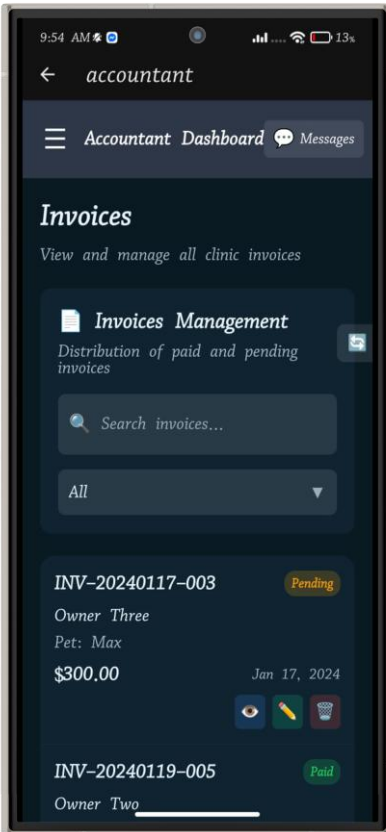
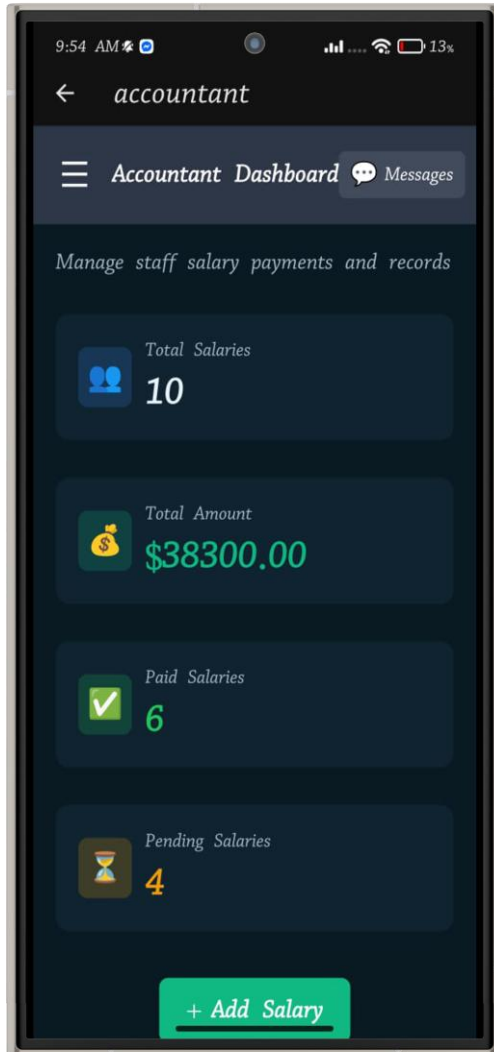
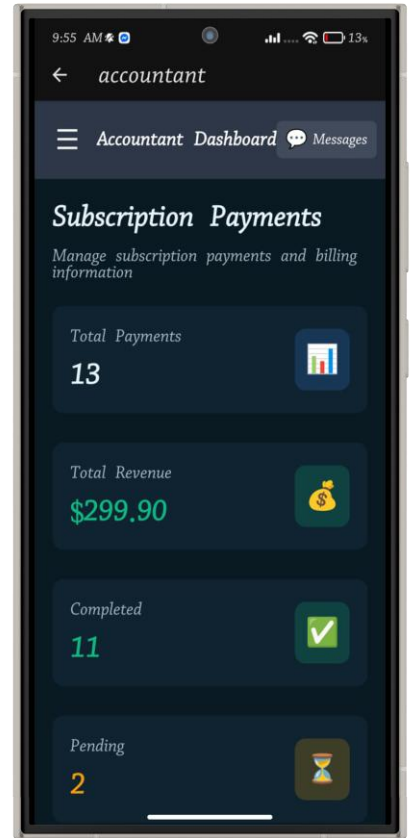
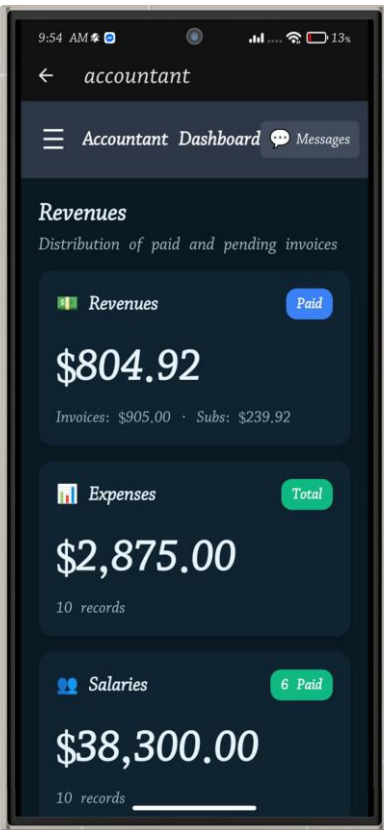


Figure 67 Sample Nablus Vet Care mobile accountant screens.

4.8 AI-Powered Features

All Project Development Powered by Qwen AI

This project was entirely developed using **Qwen AI**, leveraging its advanced capabilities to create a comprehensive AI-driven veterinary platform. The solution utilizes Qwen AI's sophisticated language and reasoning models to power the following intelligent features:

Clinical AI Assistance

The system integrates cutting-edge AI to enhance veterinary care delivery:

- **Veterinary AI Assistant:** Provides professional medical consultation including diagnosis support, treatment recommendations, and detailed symptom analysis for veterinary professionals
- **Pet Owner AI Assistant:** Delivers accessible care guidance, health monitoring, emergency assistance, and behavioral support directly to pet owners
- **Smart Diagnostics:** Enables AI-assisted treatment recommendations and predictive analytics for forecasting health outcomes
- **Intelligent Scheduling:** Features AI-optimized appointment management and automated reminder systems to streamline clinic operations

Business Intelligence

Utilizes Qwen AI for comprehensive financial analysis and strategic insights:

- Generates actionable business insights and predictive financial forecasts
- Provides automated reporting capabilities for performance tracking
- Supports data-driven clinic management and strategic planning decisions

Technical Foundation: All AI capabilities—from natural language processing in the assistants to predictive analytics in diagnostics and business intelligence—are developed and powered exclusively by the **Qwen AI** platform, ensuring consistent, reliable, and advanced intelligence across all system components.

4.9 Communication System

Real-Time Communication Infrastructure

The system implements a comprehensive communication framework using Socket.io for real-time message delivery. Multi-user chat supports conversations between all role types with

complete message history, file sharing capabilities, typing indicators, and read receipt functionality.

Notification Management

A sophisticated notification system delivers priority-alert messages categorized by medical, appointment, payment, and system types. Users maintain complete notification histories with configurable alert settings to manage communication preferences.

Voice Integration

Integrated voice calling enables direct communication between users with call management interfaces, communication logging, multi-party call capabilities, and call quality controls.

4.10 User Experience and Technical Implementation

Multi-Language and Accessibility

The system delivers comprehensive multi-language support for English, Arabic, and Hebrew with complete RTL layout capabilities. Professional translations ensure accurate medical and business terminology across all languages. Accessibility features include screen reader support, keyboard navigation, high contrast options, and font size scaling.

Technical Architecture

Built on modern web technologies including React 18 with TypeScript, Tailwind CSS, and shadcn/ui component library, the system ensures optimal performance and maintainability. RESTful API integration, code splitting, lazy loading, and strategic caching deliver responsive user experiences across all portals.

Security and Compliance

The implementation emphasizes data protection through encryption, HIPAA compliance for healthcare data, GDPR compliance for European users, granular access controls, and comprehensive audit logging. Automated backup systems, disaster recovery planning, and data redundancy ensure business continuity.

Chapter 5: Results and Analysis

This section highlights the key performance indicators and outcomes arising from the operations of the Nablus Vet Care System. The data ranges from user adoption metrics across different roles, through appointment management efficiency, financial transaction processing, to the effectiveness of AI-assisted diagnostics and multi-language platform utilization. This analysis provides a clear picture of the system's effectiveness and efficiency in transforming veterinary care delivery for further improvements.

All data collected was subjected to statistical processing to identify meaningful trends and relationships. The results encompass user engagement patterns across different stakeholder roles, appointment scheduling and completion effectiveness, payment processing efficiency, and the impact of AI integration on diagnostic workflows. It should be noted that while the system was fully implemented across both web and mobile platforms with extensive mobile interfaces for all user roles, this report includes representative web screenshots and selected mobile examples due to the large volume of mobile screens. These findings provide valuable insights into how different users interact with the platform and where enhancements could optimize veterinary service delivery.

Various visualization methods including graphs, bar charts, and line plots were employed in the data analysis. These visual tools provide a clearer understanding of performance trends across different user categories, service types, and time periods. For instance, bar graphs comparing appointment completion rates across different veterinary specializations help identify patterns in service delivery efficiency. Similarly, pie charts illustrating the distribution of user roles and language preferences reveal important insights about system adoption across different user demographics.

The data processing maintained careful consideration for potential measurement inaccuracies that might have occurred during automated tracking of user interactions or AI consultation outcomes. Error margins and variations in the data collection methodology were accounted for to enhance the reliability and validity of the findings. This included normalization of data recording practices and validation of automated tracking mechanisms.

In summary, the Results and Analysis section provides a comprehensive overview of the Nablus Vet Care System's performance across key operational metrics, user engagement trends, and statistical patterns. It serves to highlight areas of successful implementation while identifying opportunities for refinement, ensuring the system continues to effectively serve its primary purpose of enhancing veterinary care management and pet health outcomes through technological innovation.

Chapter 6: Discussion

The Nablus Vet Care System was designed to bring efficiency and technological advancement to veterinary clinic management, addressing the needs of veterinarians, pet owners, receptionists, accountants, and administrators. We sought to create an integrated platform that would streamline appointment scheduling, enhance medical record management, improve communication, and optimize financial operations. In this project, we have delivered a comprehensive solution that serves the diverse requirements of all stakeholders in the veterinary ecosystem.

We have addressed core challenges by developing an intuitive system that incorporates essential features such as AI-assisted diagnostics, multi-language support, real-time communication tools, and automated payment processing. The system facilitates seamless interaction between pet owners and veterinary staff, ensures accurate tracking of medical histories, and provides clinic administrators with comprehensive oversight of operations. These capabilities significantly enhance the efficiency, organization, and accessibility of veterinary services for all involved parties.

Our primary contribution lies in creating a unified platform that connects the various roles within a veterinary practice while incorporating advanced AI capabilities to support clinical decision-making. The system is designed to optimize daily workflows and reduce administrative burdens, allowing veterinary professionals to focus more on patient care rather than paperwork. By incorporating features such as the AI Diagnostic Assistant, multi-role dashboards, and integrated payment processing, we have created a versatile tool adaptable to various veterinary practice sizes and specializations.

Our findings logically suggest that by automating and centralizing key veterinary processes, the Nablus Vet Care System enhances both operational efficiency and the quality of pet healthcare services. Veterinary clinics can now allocate more resources to their primary mission of animal care, as administrative functions have been streamlined through digital transformation.

While the system has demonstrated significant success in its current implementation, certain limitations present opportunities for future enhancement. User feedback has indicated that expanded analytics capabilities for tracking treatment outcomes and predictive health monitoring would provide additional value for veterinary professionals. Some clinics expressed interest in extending the AI capabilities to include more specialized diagnostic support for uncommon animal species and complex medical conditions.

Future development of the Nablus Vet Care System could explore integration of more advanced machine learning algorithms for predictive health analytics and personalized treatment recommendations based on breed-specific and individual pet medical histories. Enhanced mobile capabilities, including offline functionality and expanded telemedicine features, would further improve accessibility for pet owners and veterinary professionals in various operational

environments. Additionally, exploring integration with broader veterinary networks and specialist referral systems could create more comprehensive care ecosystems for complex medical cases.

Chapter 7: Conclusion and Recommendations

The Nablus Vet Care System has successfully delivered a comprehensive technological solution that transforms veterinary clinic management through its integrated, web-based platform. The system provides each stakeholder group—veterinarians, pet owners, receptionists, accountants, and administrators—with specialized tools to streamline appointment scheduling, enhance medical record management, improve communication, and optimize financial operations. It enables pet owners to easily manage their pets' healthcare journey, while veterinary professionals can focus on delivering quality care through reduced administrative burdens and AI-assisted clinical support.

The integration of AI diagnostic assistance, real-time communication tools, and multi-language support has created a seamless, organized, and effective veterinary care ecosystem. Furthermore, the comprehensive administrative functions, including performance analytics, financial reporting, and user management, empower clinics to optimize their operations and maintain high standards of pet healthcare delivery.

Based on the system's current performance and user feedback, several recommendations for future development and enhancement emerge. First, expanding the predictive analytics capabilities would enable more advanced health monitoring and early intervention systems. By analyzing patterns in medical records and treatment outcomes, the system could proactively identify potential health risks and recommend preventive care measures, ultimately improving long-term pet health outcomes.

Second, enhancing the telemedicine features would better support remote consultations and follow-up care. Incorporating video consultation capabilities, remote vital monitoring integration, and digital prescription services would create a more comprehensive virtual care platform, particularly valuable for post-operative monitoring, chronic condition management, and rural pet owners with limited access to veterinary facilities.

Third, developing more advanced interoperability with external veterinary systems and diagnostic equipment would streamline data exchange and reduce manual entry requirements. Integration with laboratory systems, imaging equipment, and specialized diagnostic tools would create a more connected healthcare ecosystem, improving both efficiency and data accuracy across the veterinary care continuum.

Lastly, expanding the mobile application's functionality to include offline capabilities and enhanced push notification systems would improve accessibility for users in areas with limited connectivity. This would ensure continuous access to critical pet health information and appointment reminders, particularly important for emergency situations and medication schedules.

These enhancements, when implemented, would further revolutionize veterinary practice management, increase user satisfaction across all stakeholder groups, and contribute to more

effective and accessible pet healthcare delivery. While the Nablus Vet Care System has made significant strides in modernizing veterinary clinic operations, continuous improvement will ensure the platform evolves alongside advancing veterinary medicine and changing user expectations.

Future development should prioritize artificial intelligence and machine learning advancements to create more personalized treatment plans and predictive health analytics. By analyzing breed-specific data, individual medical histories, and treatment outcomes, the system could provide increasingly accurate diagnostic support and tailored care recommendations. This data-driven approach would empower veterinarians with deeper insights while enhancing preventive care strategies.

Mobile health integration represents another significant opportunity for expansion. While the current system provides robust web and mobile applications, deeper integration with wearable pet health monitors and IoT devices could enable real-time health tracking and early warning systems for conditions such as epilepsy, diabetes, or mobility issues. This would transform reactive veterinary care into proactive health management.

The incorporation of educational resources and client engagement tools would further enhance the system's value. Interactive pet care guides, medication reminder systems, and nutritional planning tools would empower pet owners to take more active roles in their pets' healthcare between veterinary visits. Such features would also strengthen the client-clinic relationship through continuous engagement.

Additionally, exploring blockchain technology for secure medical record sharing between different veterinary practices could create a unified pet health history system. This would be particularly valuable for pets that receive care from multiple providers or during emergency situations when complete medical history access is critical.

Finally, expanding the multi-language support to include additional languages and regional dialects would make the system more accessible to diverse communities. As veterinary care continues to globalize, supporting broader linguistic diversity would ensure the platform remains inclusive and effective across different cultural contexts.

These future developments would solidify the Nablus Vet Care System's position as a comprehensive digital health platform that not only manages veterinary practice operations but also actively contributes to improved pet health outcomes through technological innovation and data-driven insights.

References

Frontend Technologies

- Axios. (2025). *Axios documentation*. <https://axios-http.com/docs/intro>
- Bootstrap. (2025). *Bootstrap documentation*. <https://getbootstrap.com/docs/>
- Class Variance Authority. (2025). *CVA documentation*. <https://cva.style/docs>
- cmdk. (2025). *cmdk documentation*. GitHub. <https://github.com/pacocoursey/cmdk#readme>
- date-fns. (2025). *date-fns documentation*. <https://date-fns.org/docs/Getting-Started>
- ECMA International. (2025). *ECMAScript specification*. <https://tc39.es/ecma262/>
- Embla Carousel. (2025). *Embla Carousel documentation*. <https://www.embla-carousel.com/get-started/>
- ESLint. (2025). *ESLint documentation*. OpenJS Foundation. <https://eslint.org/docs/latest/>
- Firebase. (2025). *Firebase documentation*. Google. <https://firebase.google.com/docs>
- Font Awesome. (2025). *Font Awesome documentation*. <https://fontawesome.com/docs>
- Google. (2025). *Google OAuth 2.0 documentation*. <https://developers.google.com/identity/protocols/oauth2>
- Hookform Resolvers. (2025). *Hookform Resolvers documentation*. GitHub. <https://github.com/react-hook-form/resolvers#readme>
- html2canvas. (2025). *html2canvas documentation*. <https://html2canvas.hertzen.com/documentation>
- Input OTP. (2025). *Input OTP documentation*. <https://input-otp.rodz.dev/docs>
- jsPDF. (2025). *jsPDF documentation*. GitHub. <https://artskydj.github.io/jsPDF/docs/>
- jsPDF AutoTable. (2025). *jsPDF AutoTable documentation*. GitHub. <https://github.com/simonbengtsson/jsPDF-AutoTable#readme>
- Lucide. (2025). *Lucide icons documentation*. <https://lucide.dev/guide/>
- MDN Web Docs. (2025a). *Cross-Origin Resource Sharing (CORS)*. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- MDN Web Docs. (2025b). *CSS3 documentation*. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/CSS>

MDN Web Docs. (2025c). *HTML5 documentation*. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/HTML>

MDN Web Docs. (2025d). *JavaScript documentation*. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

next-themes. (2025). *next-themes documentation*. GitHub. <https://github.com/pacocoursey/next-themes#readme>

OpenAI. (2025). *OpenAI API documentation*. <https://platform.openai.com/docs/>

Radix UI. (2025). *Radix UI primitives documentation*. WorkOS. <https://www.radix-ui.com/primitives/docs/overview/introduction>

React. (2025). *React documentation*. <https://react.dev/learn>

React Day Picker. (2025). *React Day Picker documentation*. <https://react-day-picker.js.org/docs>

React Hook Form. (2025). *React Hook Form documentation*. <https://react-hook-form.com/docs>

React Native. (2025). *React Native documentation*. <https://reactnative.dev/docs/getting-started>

React Resizable Panels. (2025). *React Resizable Panels documentation*. GitHub. <https://github.com/bvaughn/react-resizable-panels#readme>

React Router. (2025). *React Router documentation*. React Training. <https://reactrouter.com/en/main>

Recharts. (2025). *Recharts documentation*. <https://recharts.org/en-US/>

shadcn/ui. (2025). *shadcn/ui documentation*. <https://ui.shadcn.com/docs>

Socket.IO. (2025). *Socket.IO documentation*. <https://socket.io/docs/v4/>

Sonner. (2025). *Sonner documentation*. <https://sonner.emilkowal.ski/>

Stripe. (2025). *Stripe.js documentation*. <https://stripe.com/docs/js>

tailwind-merge. (2025). *tailwind-merge documentation*.

GitHub. <https://github.com/dcastil/tailwind-merge#readme>

tailwindcss-animate. (2025). *tailwindcss-animate documentation*.

GitHub. <https://github.com/jamiebuilds/tailwindcss-animate#readme>

Tailwind CSS. (2025). *Tailwind CSS documentation*. Tailwind Labs. <https://tailwindcss.com/docs>

TanStack Query. (2025). *TanStack Query documentation*.

TanStack. <https://tanstack.com/query/latest/docs/framework/react/overview>

TypeScript. (2025). *TypeScript documentation*. TypeScript Foundation. <https://www.typescriptlang.org/docs/>

Vaul. (2025). *Vaul documentation*. <https://vaul.emilkowal.ski/>

Vite. (2025). *Vite documentation*. <https://vitejs.dev/guide/>

Vite React SWC Plugin. (2025). *Vite React SWC Plugin documentation*. GitHub. <https://github.com/vitejs/vite-plugin-react-swc#readme>

W3C. (2025a). *CSS specifications*. World Wide Web Consortium. <https://www.w3.org/Style/CSS/>

W3C. (2025b). *HTML Living Standard*. Web Hypertext Application Technology Working Group. <https://html.spec.whatwg.org/>

W3Schools. (2025). *SQL tutorial*. <https://www.w3schools.com/sql/>

Zod. (2025). *Zod documentation*. <https://zod.dev/>

Backend Technologies

12-Factor App. (2025). *The twelve-factor app*. <https://12factor.net/config>

Autoprefixer. (2025). *Autoprefixer documentation*. GitHub. <https://github.com/postcss/autoprefixer#readme>

bcrypt. (2025). *bcrypt documentation*. npm. <https://www.npmjs.com/package/bcrypt>

bcryptjs. (2025). *bcryptjs documentation*. npm. <https://www.npmjs.com/package/bcryptjs>

CORS. (2025). *CORS documentation*. npm. <https://www.npmjs.com/package/cors>

dotenv. (2025). *dotenv documentation*. npm. <https://www.npmjs.com/package/dotenv>

Express.js. (2025a). *Express.js documentation*. OpenJS Foundation. <https://expressjs.com/>

Express.js. (2025b). *Express middleware*. OpenJS Foundation. <https://expressjs.com/en/guide/using-middlewares.html>

Express.js. (2025c). *Security best practices*. OpenJS Foundation. <https://expressjs.com/en/advanced/best-practice-security.html>

Firebase Admin. (2025). *Firebase Admin documentation*. Google. <https://firebase.google.com/docs/admin/setup>

Git. (2025). *Git documentation*. <https://git-scm.com/doc>

GitHub. (2025). *GitHub guides*. <https://guides.github.com/>

Google Auth Library. (2025). *Google Auth Library documentation*. Google. <https://googleapis.dev/nodejs/google-auth-library/latest/>

Helmet.js. (2025). *Helmet.js documentation*. <https://helmetjs.github.io/>

IETF. (2015). *JSON Web Token (JWT) RFC 7519*. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc7519>

IETF. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3 RFC 8446*. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc8446>

JWT. (2025). *JWT introduction*. <https://jwt.io/introduction>

MDN Web Docs. (2025e). *Async functions*. Mozilla. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function

MDN Web Docs. (2025f). *CSS direction property*. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/CSS/direction>

MDN Web Docs. (2025g). *HTTP methods*. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

MDN Web Docs. (2025h). *MVC architecture*. Mozilla. <https://developer.mozilla.org/en-US/docs/Glossary/MVC>

MDN Web Docs. (2025i). *WebRTC API*. Mozilla. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API

Multer. (2025). *Multer documentation*. npm. <https://www.npmjs.com/package/multer>

Node.js. (2025). *Node.js documentation*. OpenJS Foundation. <https://nodejs.org/docs/>

Node.js Best Practices. (2025). *GitHub repository*. <https://github.com/goldbergonyi/nodebestpractices>

node-postgres. (2025a). *node-postgres documentation*. <https://node-postgres.com/>

node-postgres. (2025b). *Connection pooling*. <https://node-postgres.com/features/pooling>

Nodemailer. (2025). *Nodemailer documentation*. <https://nodemailer.com/about/>

Nodemon. (2025). *Nodemon documentation*. <https://nodemon.io/>

PostCSS. (2025). *PostCSS documentation*. <https://postcss.org/docs/>

PostgreSQL. (2025a). *PostgreSQL documentation*. PostgreSQL Global Development Group. <https://www.postgresql.org/docs/>

PostgreSQL. (2025b). *PostgreSQL SQL documentation*. PostgreSQL Global Development Group. <https://www.postgresql.org/docs/current/sql.html>

PostgreSQL. (2025c). *Performance optimization*. PostgreSQL Global Development Group. https://wiki.postgresql.org/wiki/Performance_Optimization

RESTful API. (2025). *REST API tutorial*. <https://restfulapi.net/>

Socket.IO. (2025). *Performance tuning*. <https://socket.io/docs/v4/performance-tuning/>

Stripe. (2025). *Stripe documentation*. <https://stripe.com/docs>

Unicode Consortium. (2025). *Arabic Unicode chart*. <https://unicode.org/charts/PDF/U0600.pdf>

W3C. (2025c). *WebRTC specification*. World Wide Web Consortium. <https://www.w3.org/TR/webrtc/>

WebRTC. (2025). *WebRTC getting started*. <https://webrtc.org/getting-started/>

Wikipedia. (2025). *Bcrypt*. <https://en.wikipedia.org/wiki/Bcrypt>

AI Technology

Qwen ai (2025) <https://qwen.readthedocs.io/en/latest/>