

بسم الله الرحمن الرحيم



Palestine Polytechnic University

College of Information Technology and Computer

Engineering

Department of Computer Science and Information

Technology

TechCare: medical platform with AI integration using Odoo

Project by:

Hasan Ahmad Ajweh

Aziz Zamel Wahesh

Under supervision of:

Suzzane Sultan

2024-2025

اهداء

الحمد لله الذي هدانا لهذا وما كنا لنهتدي لولا ان هدانا الله

إلى من لا توفيهم الكلمات والحروف حقهم في البر والإحسان، إلى من رضا الله في رضاهم
وما توفيقنا وسر نجاحنا إلا بهم، إلى من كلهم الله بالهبة والوقار، إلى من علمني العطاء
بدون انتظار، إلى من نحمل اسماءهم بكل افتخار، نرجو من

الله ان يمد بأعماركم لتروا ثمارا بعد طول انتظار، وستبقى كلماتكم نجوما نهتدي بها اليوم
وفى الغد والى الابد والدينا العزيزين

إلى من رافقونا طوال السنين وشاركونا الفرح والألم، إلى معنى الحب والحنان والتفاني، إلى
بسمة الحياة وسر الوجود الى من كان دعائهم سر نجاحنا وحنانهم سر تقدمنا، إلى أمهاتنا،
إلى من هم أقرب إلينا من روحنا، إلى من شاركونا حزن الألم وبهم نستمد عزتنا وإصرارنا
إخوتنا الأعزاء، إلى من أنسنا في دراستنا وشاركنا همومنا تذكراً وتقدير أصدقائنا، إلى من
ساندنا وشجعنا طوال مسيرتنا الدراسية جميعاً، اهدي ثمرة هذا الجهد المتواضع وفاء، إلى
حاملي لواء النور والسائرين في دربهم بإخلاص كل أساتذتي.

Abstract

Healthcare systems need to work together with different healthcare sectors such as hospitals and clinics. This will help both patients and developers. Patients won't need to worry about carrying medical files when moving from one sector to another. Developers can also use divide-and-conquer technique using modules to make the development process easier.

In this project, we are implementing an ERP (Enterprise Resource Planning) system using the Odoo platform. The implementation process is divided into two phases: programming a custom-built Clinic Management Module and customizing existing modules, such as the website module. We made healthcare management easier by using Odoo ERP for smooth appointment booking, AI-powered support, and better clinic operations. We successfully added an AI chatbot, customized Odoo modules without issues, and got great feedback from users, proving the system is easy to use, reliable, and scalable.

Table Of Contents

Contents

Chapter 1: The Introduction	8
1.1 Introduction	8
1.2 Project's Background: Enterprise Resource Planning (ERP)	8
1.3 Problem Space	9
1.4 Existing Systems.....	9
1.5 Alternatives.....	10
1.6 Project Id.....	11
1.7 Scope.....	12
1.8 Timeline.....	13
1.9 Context Diagram.....	14
Chapter 2: Requirements Specification	15
2.1 Introduction	15
2.2 Actors of the system	15
2.3 System Requirements	16
2.3.1 Functional Requirements	16
2.3.2 Non-Functional Requirements	17
2.4 Use Case Diagram	18
2.5 Functional Requirements Specifications	19

2.6 Sequence Diagrams	22
Chapter 3: System Architecture & Design	23
3.1 Introduction	23
3.2 Odoo's Multitier Architecture	23
3.3 Helpdesk Module Integration	25
3.4 AI Assistant Integration.....	26
3.5 TechCare interfaces	27
3.6 TechCare Module	37
3.7 Class Diagram.....	38
3.8 Mapping & Normalization	39
3.9 Description of database tables	39
Chapter 4: Implementation	43
4.1 Introduction	43
4.2 TechCare Implementation Process	43
4.2.1 Customization Phase	43
4.2.2 Programming Phase	44
4.3 Implementation Problems and Constraints.....	50
Chapter 5: Testing and Validation	51
5.1 Introduction	51
5.2 Website Testing Scenarios	51
5.3 TechCare Module Testing Scenarios	52
Chapter 6: Results and Future Work	54
6.1 Conclusion	54
6.2 Future Work.....	54
References	55

TABLE OF FIGURES

<i>Figure 1 Context diagram</i>	<i>14</i>
<i>Figure 2 Use-Case Diagram</i>	<i>18</i>
<i>Figure 3 Sequence Diagram for Interacting with AI Assistant</i>	<i>22</i>
<i>Figure 4 Registration Process Sequence Diagram.....</i>	<i>23</i>
<i>Figure 5 Multitier Architecture</i>	<i>24</i>
<i>Figure 6 Multitier Architecture Template (Figure 5 but simplified)</i>	<i>24</i>
<i>Figure 7 Helpdesk Main Page.....</i>	<i>25</i>
<i>Figure 8 Helpdesk ticket processing page.....</i>	<i>25</i>
<i>Figure 9 AI chatbot.....</i>	<i>26</i>
<i>Figure 10 Home screen.....</i>	<i>27</i>
<i>Figure 11 Trends interface</i>	<i>27</i>
<i>Figure 12 Trends example.....</i>	<i>28</i>
<i>Figure 13 Services.....</i>	<i>28</i>
<i>Figure 14 Appointment's interface</i>	<i>29</i>
<i>Figure 15 Booking appointment</i>	<i>29</i>
<i>Figure 16 Appointment scheduled.....</i>	<i>30</i>
<i>Figure 17 Blog</i>	<i>30</i>

<i>Figure 18 Blog example.....</i>	<i>31</i>
<i>Figure 19 Forum interface.....</i>	<i>31</i>
<i>Figure 20 Post creation.....</i>	<i>32</i>
<i>Figure 21 User's Posts.....</i>	<i>32</i>
<i>Figure 22 Contact us interface.....</i>	<i>33</i>
<i>Figure 23 Ticket creation.....</i>	<i>33</i>
<i>Figure 24 Email.....</i>	<i>34</i>
<i>Figure 25 Adding a clinic.....</i>	<i>34</i>
<i>Figure 26 Appointments making interface.....</i>	<i>35</i>
<i>Figure 27 Patients added to a clinic.....</i>	<i>35</i>
<i>Figure 28 Announcements.....</i>	<i>36</i>
<i>Figure 29 Dashboard.....</i>	<i>37</i>
<i>Figure 30 Class diagram.....</i>	<i>38</i>
<i>Figure 31 Mapping & normalization.....</i>	<i>39</i>
<i>Figure 32 Deploy an Ai model to use it.....</i>	<i>45</i>
<i>Figure 33 Publishing the .Net project to WebApp Service on Azure.....</i>	<i>46</i>
<i>Figure 34 ID parity check code.....</i>	<i>49</i>
<i>Figure 35 Validation error.....</i>	<i>53</i>
<i>Figure 36 validation succeeded.....</i>	<i>53</i>

List Of Tables

<i>Table 1 Gantt chart</i>	13
<i>Table 2 Login/Register</i>	19
<i>Table 3 Browsing the website</i>	20
<i>Table 4 AI assistant</i>	20
<i>Table 5 Appointment scheduling</i>	21
<i>Table 6 Helpdesk management</i>	21
<i>Table 7 Clinic's table</i>	39
<i>Table 8 Dashboard's table</i>	40
<i>Table 9 Medical specialist's table</i>	40
<i>Table 10 Patient's table</i>	41
<i>Table 11 Appointment's table</i>	41
<i>Table 12 Announcement's table</i>	42
<i>Table 13 Website Testing Scenario</i>	51
<i>Table 14 TechCare Module Testing Scenarios</i>	52

Chapter 1: The Introduction

1.1 Introduction

The introduction chapter is the opening section of a document or project. It gives readers a general idea of what the document is about and helps them understand the context

1.2 Project's Background: Enterprise Resource Planning (ERP)

Software solutions known as enterprise resource planning (ERP) systems are made to make daily tasks and administration easier for organizations. ERP platforms enable businesses to operate more effectively by integrating all of these features into a single system. Odoo, an open-source platform famous for its adaptability, scalability, and modular design, stands out among the several ERP systems on the market.

Odoo's unique feature is its modular design, which allows it to be highly customized to meet a wide range of company requirements. Consider each module as a distinct building block that is designed to perform particular functions, such as inventory management, accounting, or customer relationship management, while integrating seamlessly with the system as a whole. Workflows are streamlined and errors are reduced when data entered into one module is automatically synchronized and triggers actions in other modules. Because of this flexibility, companies can begin small, concentrating just on the capabilities they require, then grow with ease as their operations grow.

One of Odoo's biggest features is that it's open source, which makes it high-quality and reasonably priced. Companies also gain from a worldwide developer community that is always enhancing the platform. Odoo's adaptable design, robust Python backend, and user-friendly interface make it compatible with all devices. It is an excellent option for businesses of all sizes because of this feature set.

1.3 Problem Space

Healthcare systems are large and complex. They take a long time to develop and require a lot of effort to improve. These systems need to be integrated with hospitals, clinics, and other healthcare sectors. By using a divide-and-conquer approach with modules, the systems can be connected. This makes it easier and more flexible to move patients between sectors without needing to carry their records.

The existing systems are non-integrated, causing different systems to operate independently. This requires manual data transfer between systems, increasing the risk of human error and limiting scalability.

1.4 Existing Systems

1. OpenMRS (Open Medical Record System)

- **Overview:**
An open-source platform for electronic medical record (EMR) management, commonly used in clinics and hospitals.
- **Advantages:**
 - Strong focus on secure and scalable storage of medical records.
- **Limitations:**
 - The lack of tools like appointment scheduling or patient engagement features.
 - Requires significant technical knowledge.

2. OpenClinic GA

- **Overview:**
OpenClinic GA is a free, open-source system used for managing patient records, billing, and other hospital data. It is common in small healthcare facilities.
- **Advantages:**
 - It is free and can be customized.
 - Helps with storing patient records and managing bills.

- **Limitations:**
 - The design is old and not easy to use.
 - It does not have modern features like online appointment booking or tools to connect with patients.
 - Customizing the system requires technical skills.
-

3. MedKey

- **Overview:**

MedKey is a clinic management software for smaller clinics. It focuses on basic tasks like keeping patient records, managing prescriptions, and handling payments.
- **Advantages:**
 - It is affordable and easy to use for smaller clinics.
- **Limitations:**
 - It is not good for clinics that want to grow or need advanced tools.
 - It does not have features like online appointment reminders or patient communication.
 - It does not focus on improving patient engagement or providing extra services.

1.5 Alternatives

We have two alternatives:

1. Development from Scratch

Building the entire system from the ground up, including modules like appointment scheduling, helpdesk, and AI integration.

- **Advantages:**
 - Complete control over system design and functionality.
 - Fully independent.
- **Disadvantages:**

- Time-consuming.
 - Higher costs due to complexity and high maintenance.
 - Limited scalability.
-

2. Odoo stands out with its great features:

- **Modular Design:** Quick implementation with ready-to-use modules.
- **Efficiency:** Faster development compared to building from scratch.
- **Scalability:** Easily extendable with additional modules as needed.
- **Cost-Effective:** Open-source nature reduces costs significantly.

1.6 Project Id

Our platform has multiple modules that are divided into three categories:

- The module that we developed from scratch is called TechCare.
 - Built in Modules in Odoo: Calendar and Helpdesk.
 - Modules we Customized to meet platform needs: Website and Appointment.
-

Goals Achieved:

- **Helping Patients and the Community:**

Healthcare Blog: Provided users with useful medical tips, updates on health trends, and educational articles.

Interactive Forums: Created a space where patients could ask questions, connect with doctors, and share health concerns.

AI Chatbot: Gave quick answers to common health-related queries, improving response times for users.

- **Making Healthcare More Accessible for Providers:**

Appointment System: Made it easier for patients to book appointments and reduced the workload for clinics.

- **Clinic Management Module:**

A special module was developed to help clinics with:

- Organizing and managing patient appointments.
- Safely storing and updating patient and doctor information.
- Sharing important announcements easily.

1.7 Scope

Our project focuses on improving clinic management and healthcare accessibility through a modular platform. It aims to:

1. **Help Patients:**

- Provide tools like healthcare blogs, forums, and chatbot for better access to medical information and doctor interaction.

2. **Support Clinics:**

- Simplify appointment scheduling and clinic management with customized modules to organize records and announcements.

Our project deals with five modules as mentioned before and doesn't support Laboratory, Hospital Management and Accounting Modules.

1.8 Timeline

Gantt Chart

Table 1 Gantt chart

Tasks	Time per week																
	2nd semester									1st Semester							
	2	4	6	8	10	12	14	16		2	4	6	8	10	12	14	16
Planning	█	█															
Requirements gathering			█	█	█												
System architecture and design					█	█	█	█	█	█	█						
Implementation									█	█	█	█	█				
Testing Phase											█	█	█	█	█	█	
Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
Estimated	█		█						█								

1.9 Context Diagram

The **Context Diagram** is a simple, big-picture overview of how the TechCare platform works and interacts with other parts. It shows what's inside the system, like its main functions, and how it connects with people (like users and administrators) and other systems (like third-party tools). Think of it as a map that explains who uses the platform, what they do with it, and how information flows between everyone involved.

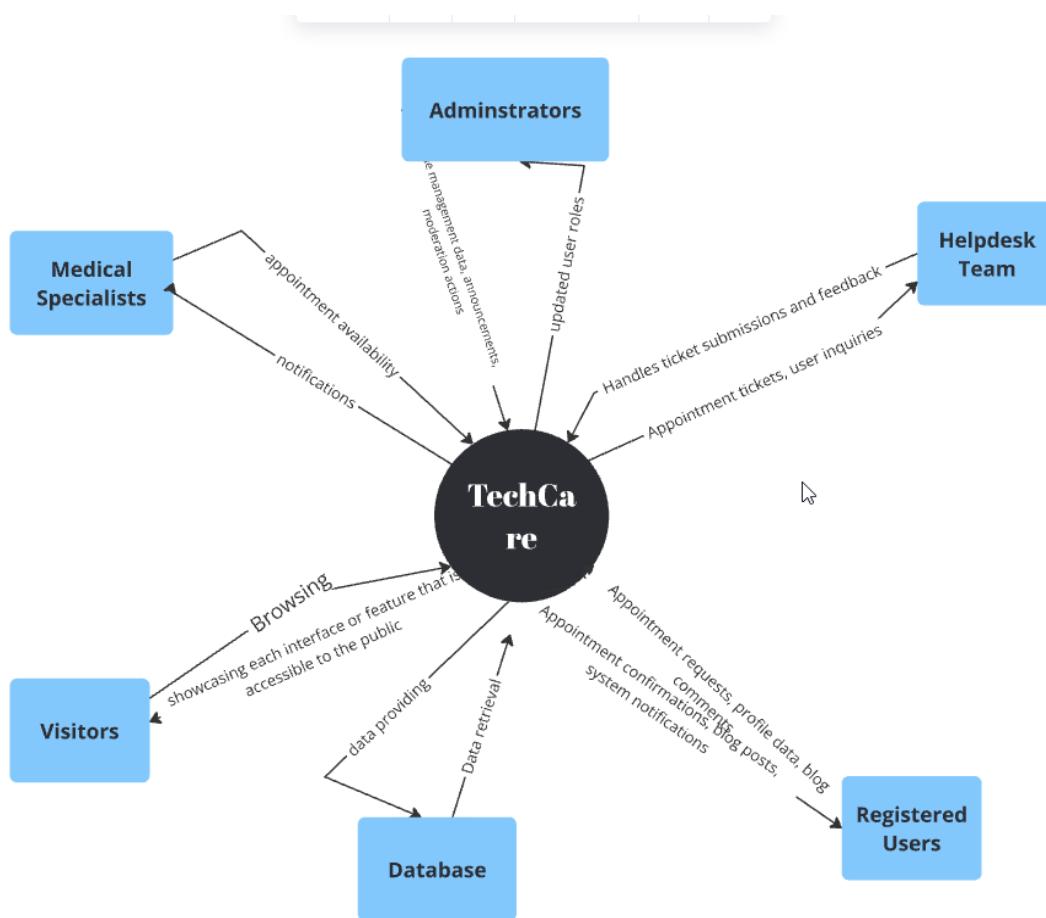


Figure 1 Context diagram

Chapter 2: Requirements Specification

2.1 Introduction

This chapter explains what the system needs to work properly and how it's designed. It covers both the main functions the system must have (functional requirements) and extra features that improve how it works (non-functional requirements). It also includes diagrams, like use case and sequence diagrams, to show how the system is supposed to work.

2.2 Actors of the system

Registered User

- **Role:** A person who registers on the platform to access additional features and engage with its content.
- **Interactions:**
 - Browse the platform's content.
 - Use the AI assistant.

System Admin

- **Role:** The administrator responsible for managing the platform to ensure smooth operations.
- **Interactions:**
 - Manage user roles
 - Manage Blog and Forum content.
 - Manage appointments and send notifications.
 - Manage helpdesk module.

Medical Specialist

- **Role:** A healthcare professional who uses the platform to share knowledge and manage their own appointments.
- **Interactions:**
 - Engage in forums by sharing knowledge or answering questions.
 - Review appointment requests.
 - Schedule and manage their own appointments.

Visitor

- **Role:** A non-registered user who accesses the platform to view public content or use the AI assistant.
- **Interactions:**
 - Browse the platform’s public content.
 - Use the AI assistant.

Helpdesk Team/Admin

- **Role:** Support staff responsible for resolving user issues and managing platform operations.
- **Interactions:**
 - Manage tickets for user support and appointment requests.
 - Notify users about ticket updates.
 - Manage reminders for appointments.

AI Assistant

- **Role:** A virtual assistant integrated into the platform to help users by answering their questions.
- **Interactions:**
 - Process queries and provide answers.

2.3 System Requirements

One of the most important steps in building a project is understanding what the system needs to do and breaking those needs into clear, manageable tasks. These tasks are organized to make sure the system works as expected. The system requirements are divided into two main types:

1. **Functional Requirements** – What the system must do.
2. **Nonfunctional Requirements** – How the system should perform.

2.3.1 Functional Requirements

User Registration and Management:

- **New User Sign-Up:** Users can register by providing the required details.
- **Profile Customization:** Users can edit their personal details.

- **Role-Based Access Control:** Different roles (admin, medical specialist, or regular user) have specific permissions.

AI Assistant Integration:

- **Query Processing:** The AI assistant can receive and interpret user queries.
- **Quick Responses:** AI provides real-time answers.

Appointment Scheduling:

- **Appointment Requests:** Users can request appointments with healthcare professionals.
- **Calendar Management:** Medical Specialists can manage schedules.
- **Appointment Confirmation:** Users receive confirmation of their appointments.
- **Reminder System:** reminders are sent to users and doctors for upcoming appointments.

2.3.2 Non-Functional Requirements

- **Scalability**
Odoos module integration makes it easier to manage and expand without any problems.
- **Usability**
The system focuses on being easy to use, with a clear and simple design. The user interface is made to help users quickly find what they need and complete tasks without confusion.

2.4 Use Case Diagram

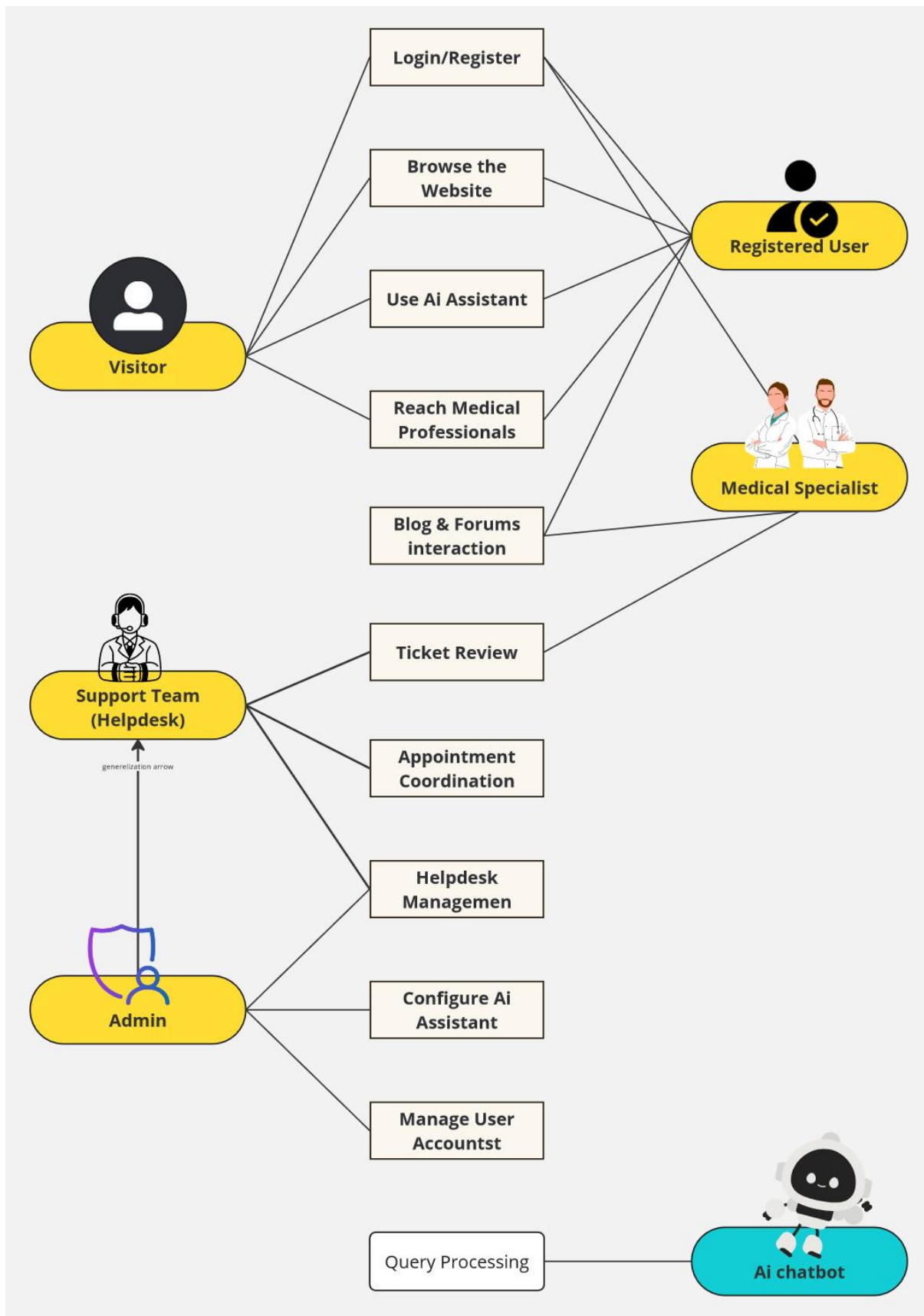


Figure 2 Use-Case Diagram

2.5 Functional Requirements Specifications

Use Case Table 1: Login/Register

Table 2 Login/Register

Requirement Name	User Login and Registration
Main User	Visitor, Registered User
Target	Allow users to register and access their accounts.
Conditions	<ul style="list-style-type: none"> - Internet connection is available. - User provides valid credentials or registration details.
Procedures	<ul style="list-style-type: none"> - Navigate to the registration or login page. - For new users: <ul style="list-style-type: none"> - Fill out the registration form. - Submit the form. - For existing users: <ul style="list-style-type: none"> - Enter username and password. - Click "Login" to access the system.
Exceptions	<ul style="list-style-type: none"> - Username or password is incorrect.
Exceptions Solution	<ul style="list-style-type: none"> - Show error message for invalid login or missing information.
	<ul style="list-style-type: none"> - Provide "Forgot Password" or "Resend Email Verification" options.

Use Case Table 2: Browsing the Website

Table 3 Browsing the website

Requirement Name	Browsing Platform Content
Main User	Visitor, Registered User
Target	Enable users to view platform content.
Conditions	<ul style="list-style-type: none"> - Access to the internet and website. - User can navigate through pages.
Procedures	<ul style="list-style-type: none"> - Visit the homepage. - For public content: - Browse categories or search for topics. - For registered content: - Log in to access .
Exceptions	-
Exceptions Solution	

Use Case Table 3: AI Assistant Query Processing

Table 4 AI assistant

Requirement Name	AI Assistant Interaction
Main User	Visitor, Registered User
Target	Allow users to interact with the AI assistant for queries.
Conditions	- User start a chat with the AI assistant.
	- Query is entered in text format.
Procedures	<ul style="list-style-type: none"> - User clicks on the AI assistant widget. - Enter a question in the text input box. - Submit the query. - AI processes the query and provides a response.
Exceptions	- Delayed response because of the Ai model version

Requirement Name	AI Assistant Interaction
Exceptions Solution	- ues advanced models

Use Case Table 4: Appointment Scheduling

Table 5 Appointment scheduling

Requirement Name	Schedule Appointments
Main User	Registered User, Medical Specialist
Target	Facilitate appointment scheduling between users and medical specialists.
Conditions	- User selects an appointment slot. - Specialist availability is confirmed.
Procedures	- User submits an appointment request. - Admin notifies the specialist. - Specialist confirms the request. - User receives confirmation details. - Reminder is sent to both before the appointment.
Exceptions	- Time slot is unavailable.
Exceptions Solution	- Notify user of unavailability

Use Case Table 5: Helpdesk Management

Table 6 Helpdesk management

Requirement Name	Helpdesk Ticket Management
Main User	Registered User, Helpdesk Team, Admin
Target	Allow users to submit support tickets and receive help.
Conditions	- User describes the issue in a ticket. - Helpdesk team reviews and assigns tickets.
Procedures	- User fills out the support ticket form.

Requirement Name	Helpdesk Ticket Management
	- Support staff work on resolving the issue. - Updates are sent to the user.
Exceptions	- Ticket lack to information.
Exceptions Solution	- Notify the user to give more info

2.6 Sequence Diagrams

Sequence Diagram for Interacting with AI Assistant: This diagram illustrates the workflow involved when a user interacts with the AI Assistant. It shows the step-by-step process from the user entering a query to receiving a response, highlighting how the system processes and retrieves the necessary information to help.

Sequence Diagram for Interacting with AI Assistant

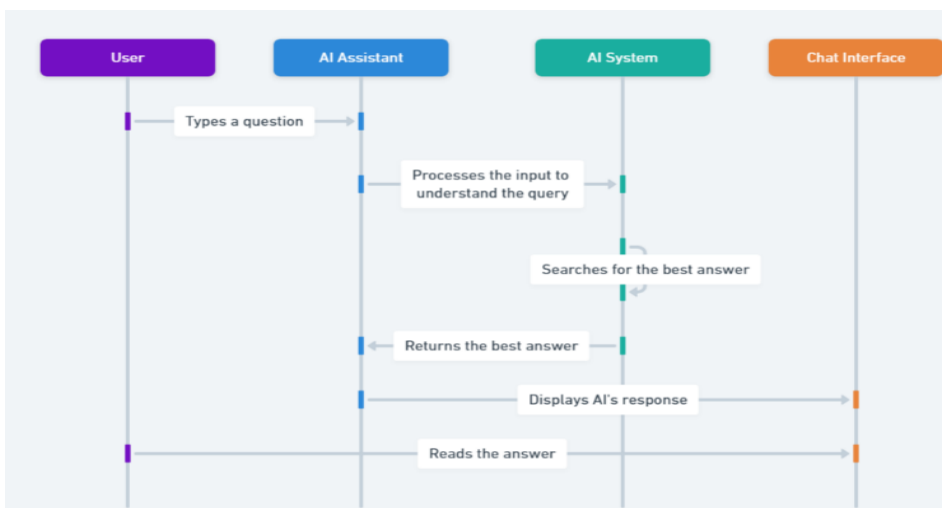


Figure 3 Sequence Diagram for Interacting with AI Assistant

Registration Process Sequence Diagram: This sequence diagram demonstrates the chronological order of operations that occur during the user registration process. It shows how the system and email service components interact following a user's attempt to register, including the handling of potential errors and the verification

Registration Process Sequence Diagram

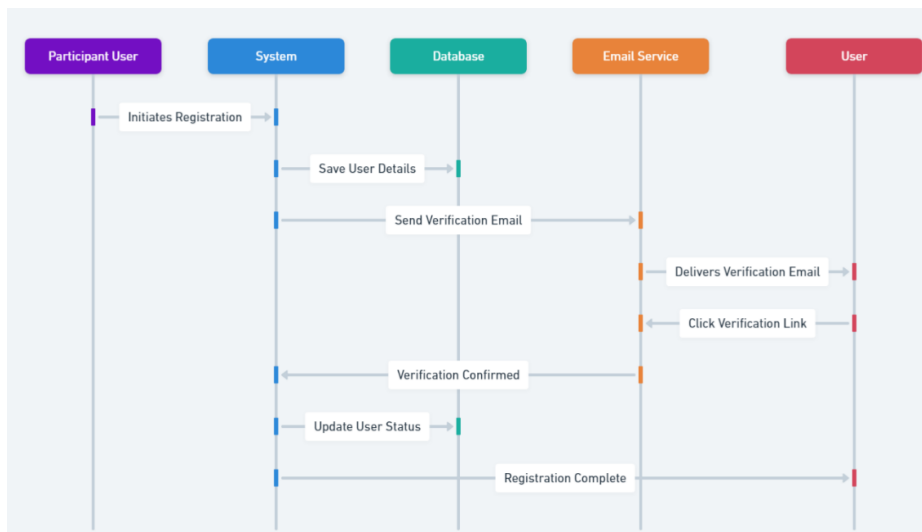


Figure 4 Registration Process Sequence Diagram

Chapter 3: System Architecture & Design

3.1 Introduction

This chapter Explores the structural framework of the system, presenting its architecture and design principles. It discusses the integration of components and highlights the strategies used to ensure scalability and flexibility.

3.2 Odoo's Multitier Architecture

Odoo use multitier architecture that separates the application into three main layers: presentation, logic, and data. This design enhances scalability, maintainability, and security.

- **Presentation Tier:** Handles user interactions through dynamic web interfaces.
- **Logic Tier:** Implements the business logic using Python, processing user requests and managing operations.
- **Data Tier:** Uses PostgreSQL as the database to securely store and manage data.

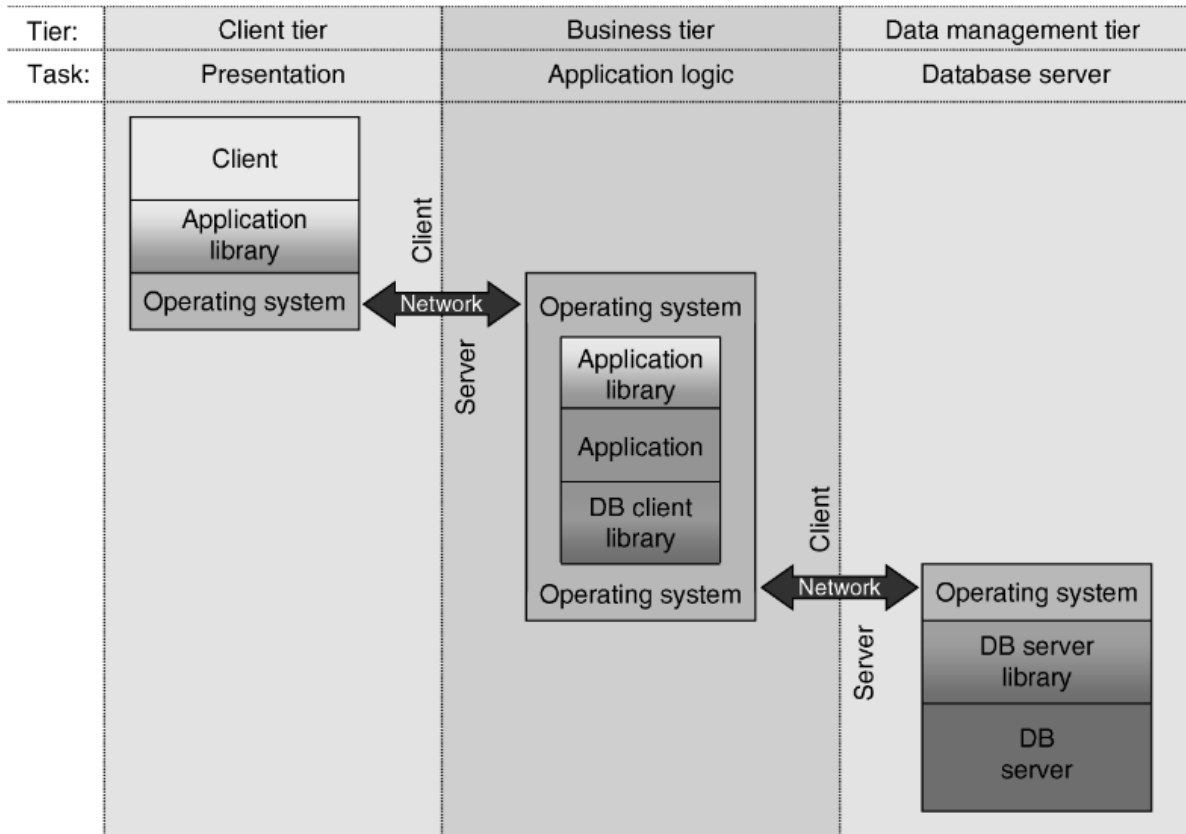


Figure 5 Multitier Architecture

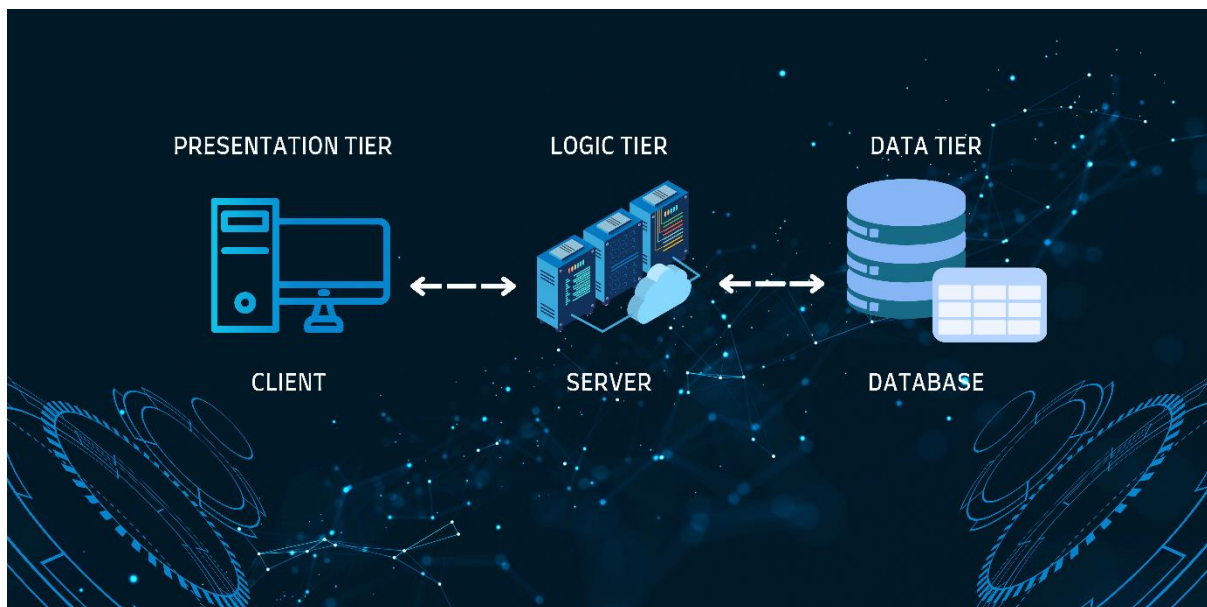


Figure 6 Multitier Architecture Template (Figure 5 but simplified)

3.3 Helpdesk Module Integration

The Helpdesk Module in Odoo is a key tool for managing customer service tasks effectively. Its simple interface makes it easy to track, prioritize, and resolve customer tickets. One of its standout features is multi-channel support, allowing inquiries to be managed smoothly no matter where they come from. The helpdesk team also plays an important role in connecting doctors and users by providing clear communication and support. They assist users in scheduling appointments, making the booking process quick and easy.

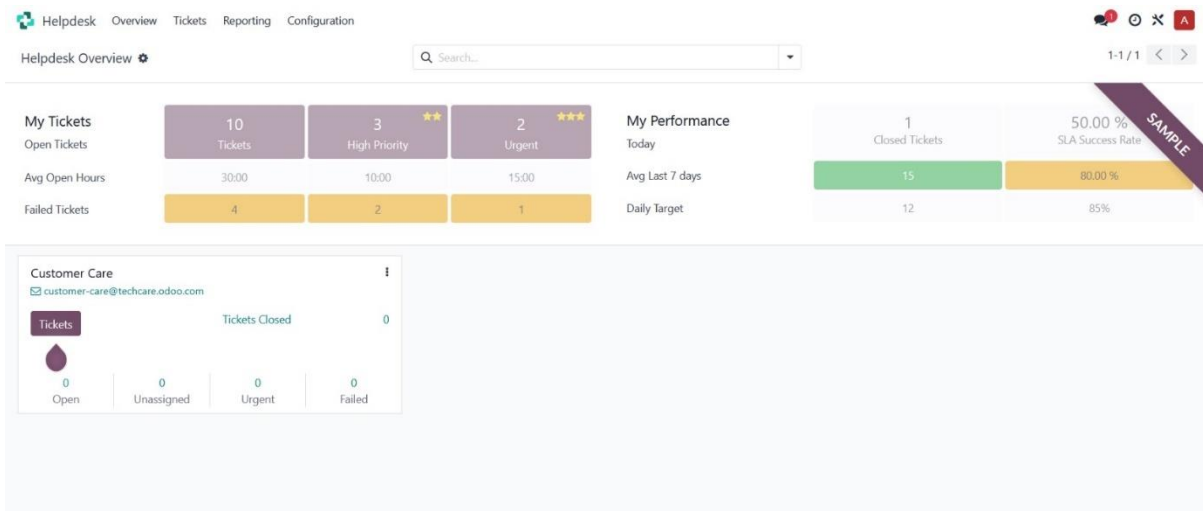


Figure 7 Helpdesk Main Page

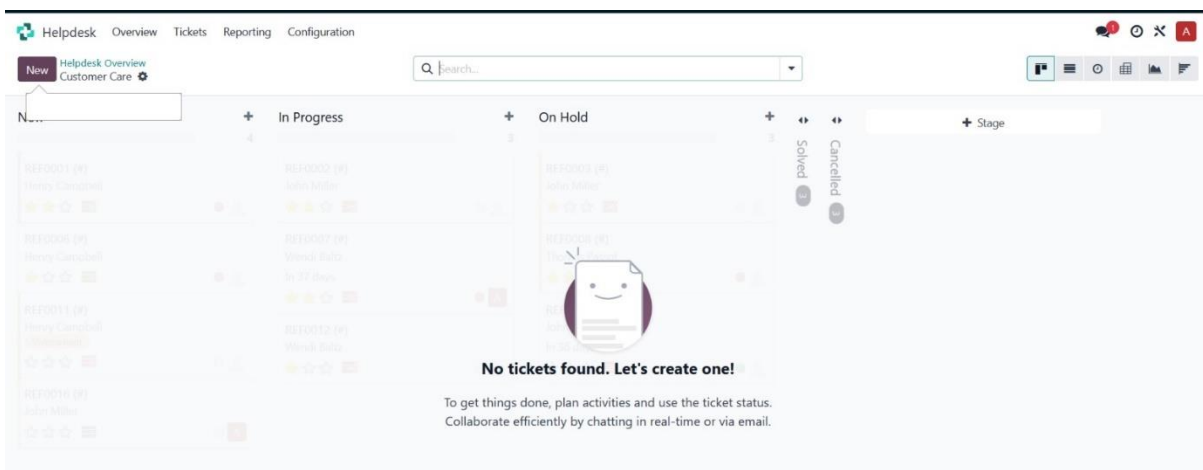


Figure 8 Helpdesk ticket processing page

3.4 AI Assistant Integration

1. Chatbot Development

- The chatbot was built using **.NET MVC**, providing a structured and scalable architecture for handling user interactions.

2. AI Model Training & Deployment

- The AI model was trained and deployed using **Azure AI Studio**, which automatically fine-tuned the model based on predefined datasets and real-time interactions.
- Azure AI Studio generated an **API key, endpoint URL, and deployment ID**, allowing seamless integration with the chatbot's backend.

3. Hosting & Deployment

- The chatbot was deployed on **Azure WebApp Service**, ensuring high availability, scalability, and efficient performance.

4. Integration with TechCare Platform

- To embed the chatbot into the TechCare website, we implemented it as a **JavaScript widget**, allowing users to interact with it directly within the platform.
-

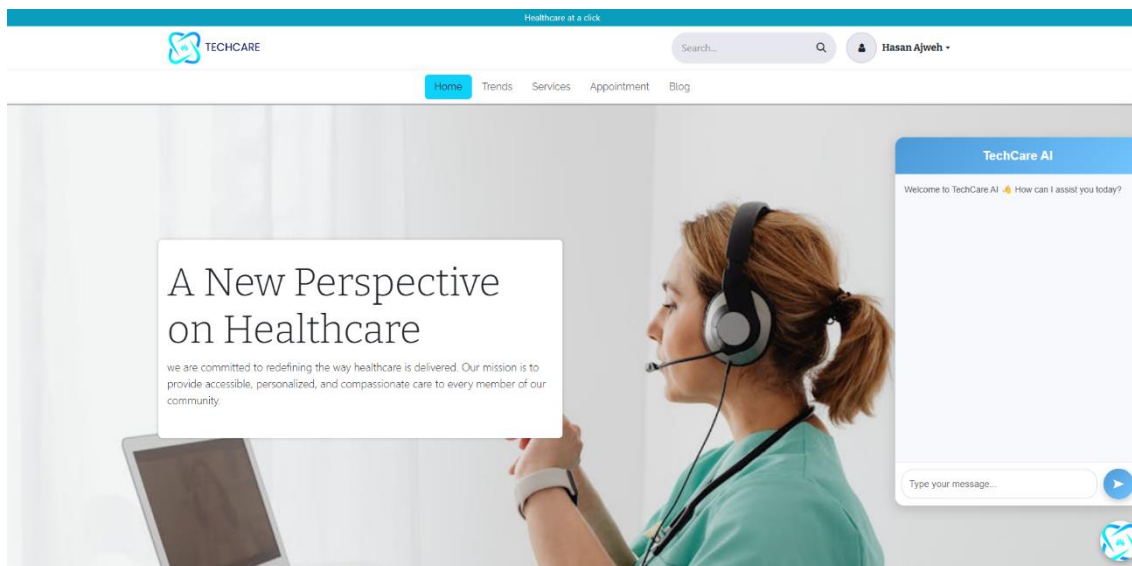


Figure 9 AI chatbot

3.5 TechCare interfaces

Home screen

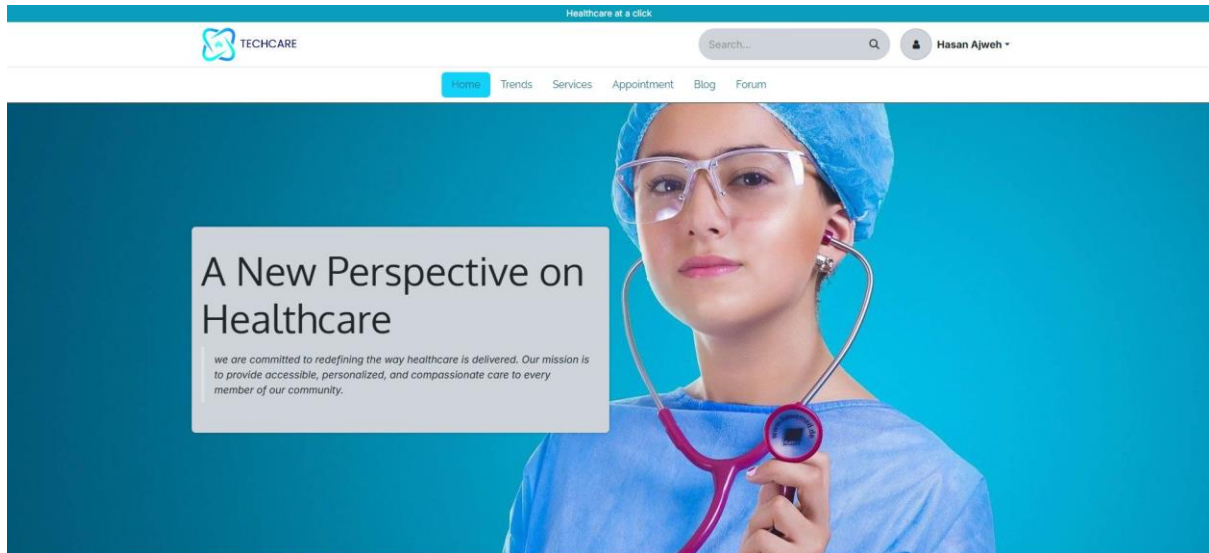


Figure 10 Home screen

Trends interface

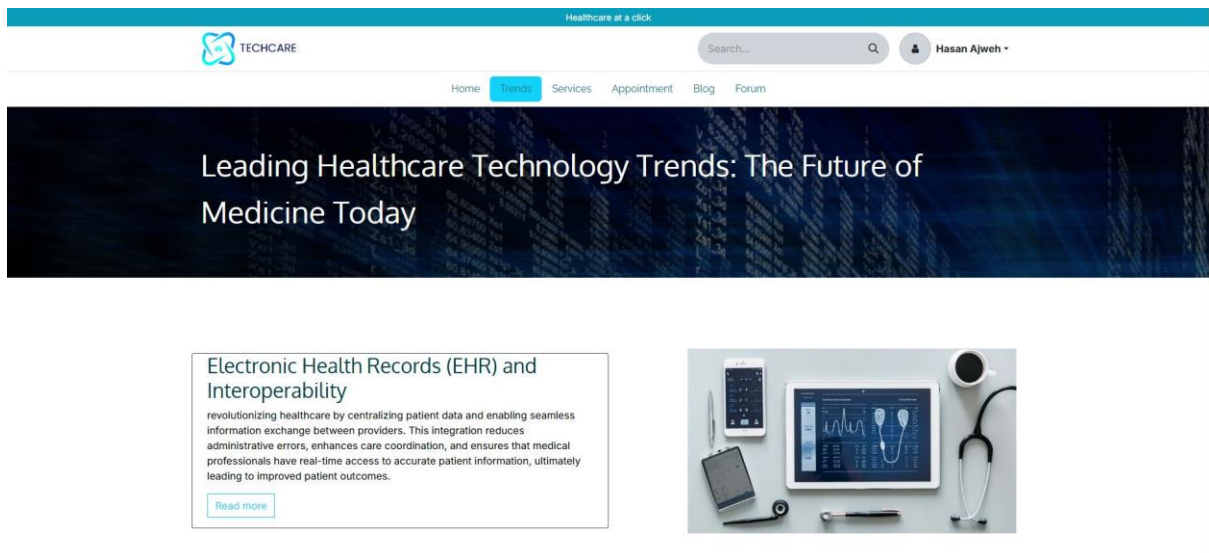


Figure 11 Trends interface

Trends example

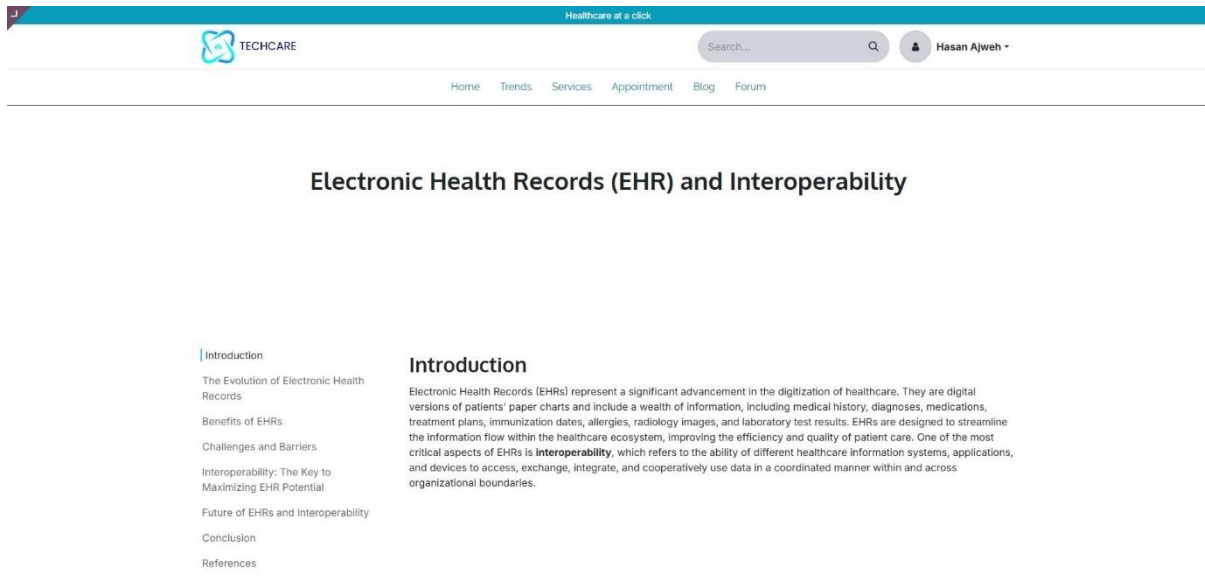


Figure 12 Trends example

Services interface

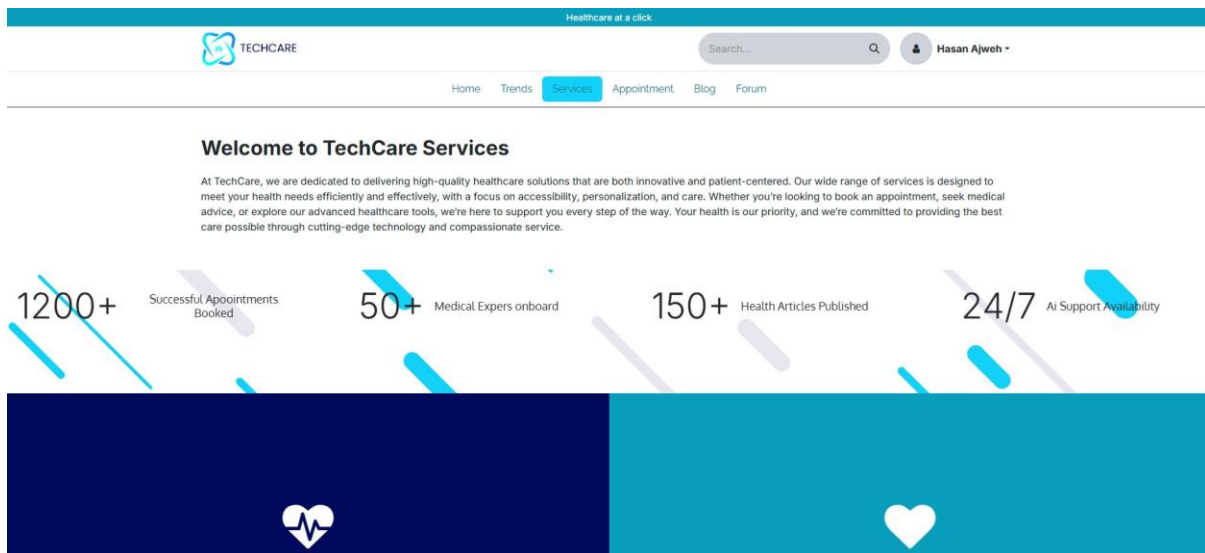


Figure 13 Services

Appointment's interface

The screenshot shows the 'Appointment's interface' on the TECHCARE website. The header includes the TECHCARE logo, a search bar, and the user name 'Hasan Ajweh'. The main navigation menu contains 'Home', 'Trends', 'Services', 'Appointment', 'Blog', and 'Forum'. The breadcrumb trail is 'Date & time > Details > Booked'. The primary action is 'Select a date & time'. A calendar for January 2025 is displayed, with the 28th selected. To the right, a 'Select a time' dropdown menu shows options for 2:00 PM, 3:00 PM, and 4:00 PM. The right sidebar, titled 'Dr. Alaa Appointment Schedule', provides 'MEETING DETAILS' (Hebron, 1 hour) and 'OPERATOR' information (Hasan Ajweh, 217843@ppu.edu.ps, +970 594303163). The 'Timezone' is set to 'Asia/Hebron'.

Figure 14 Appointment's interface

Booking an appointment

The screenshot shows the 'Booking an appointment' interface. The header and navigation are identical to Figure 14. The breadcrumb trail is 'Date & time > Details > Booked'. The primary action is 'Add more details about you'. A form contains the following fields: 'Full name*' (Hasan Ajweh), 'Email*' (217843@ppu.edu.ps), and 'Phone number*' (+970 594303163). Below the form is a text area with the placeholder 'give a small detail about what u r facing' and a button 'Add Custom Questions'. At the bottom right is a 'CONFIRM APPOINTMENT' button. The right sidebar, titled 'Dr. Alaa Appointment Schedule', shows 'MEETING DETAILS' (Hebron, 1 hour) and 'OPERATOR' information (Hasan Ajweh, 217843@ppu.edu.ps, +970 594303163). The 'DATE & TIME' section shows 'Tue Jan 28, 2025 3:00 PM Asia/Hebron'.

Figure 15 Booking appointment

Appointment Scheduled

The screenshot displays the TechCare website's appointment confirmation page. At the top, there is a navigation bar with the TechCare logo, a search bar, and the user's name 'Hasan Ajweh'. Below the navigation bar, the page title is 'Appointment Scheduled!'. The main content area shows the appointment details for 'Hasan Ajweh - Dr. Alaa Appointment Schedule Booking'. The appointment is scheduled for 'Tue Jan 28, 2025, 3:00:00 PM (Asia/Hebron)' in 'Hebron' for a duration of '1 hour'. The attendees are listed as 'Hasan Ajweh'. Contact details for Hasan Ajweh are provided, including an email address '217843@ppu.edu.ps' and a phone number '+970 594303163'. There are also buttons to 'Add to iCal/Outlook' and 'Add to Google Agenda'. A link to 'Cancel your appointment' is visible at the bottom of the details section.

Figure 16 Appointment scheduled

Blog interface

The screenshot shows the TechCare Blog interface. The header features the TechCare logo, a search bar, and the user's name 'Hasan Ajweh'. Below the header, there is a large banner for 'TechCare Blog' with the tagline 'Our blog is your go-to resource for the latest insights, tips, and news in the world of healthcare'. The main content area displays two featured blog posts. The first post is titled 'The Role of Big Data in Predicting Health Trends' by Hasan Ajweh, dated Aug 21, 2024, with 135 views. The second post is titled 'The Rise of Mental Health Awareness: Breaking the Stigma' by Hasan Ajweh, dated Aug 21, 2024, with 128 views. To the right of the blog posts, there is an 'ABOUT US' section with a search bar and a paragraph of text: 'Welcome to the TechCare blog, your go-to source for the latest advancements in healthcare technology. At TechCare, we are dedicated to bridging the gap between innovation and patient care. Our mission is to empower individuals with reliable health information and connect them with cutting-edge solutions that improve their well-being. Through insightful articles, expert advice, and the latest trends, we aim to keep you informed and inspired on your healthcare journey. Join us as we explore the future of medicine, one innovation at a time.'

Figure 17 Blog

Blog example

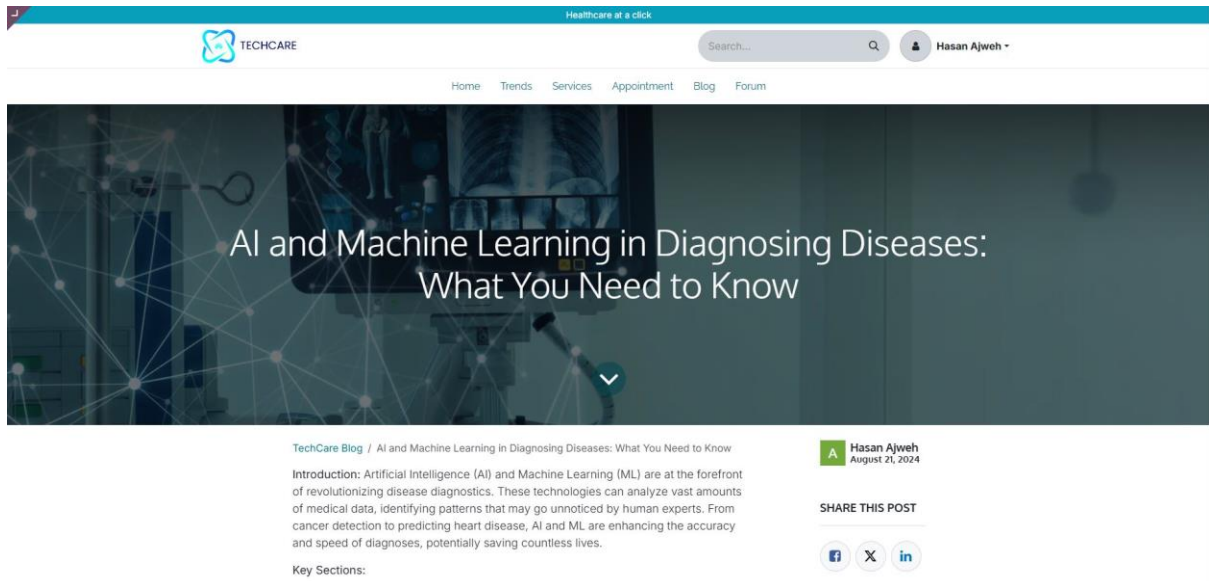


Figure 18 Blog example

Forum interface

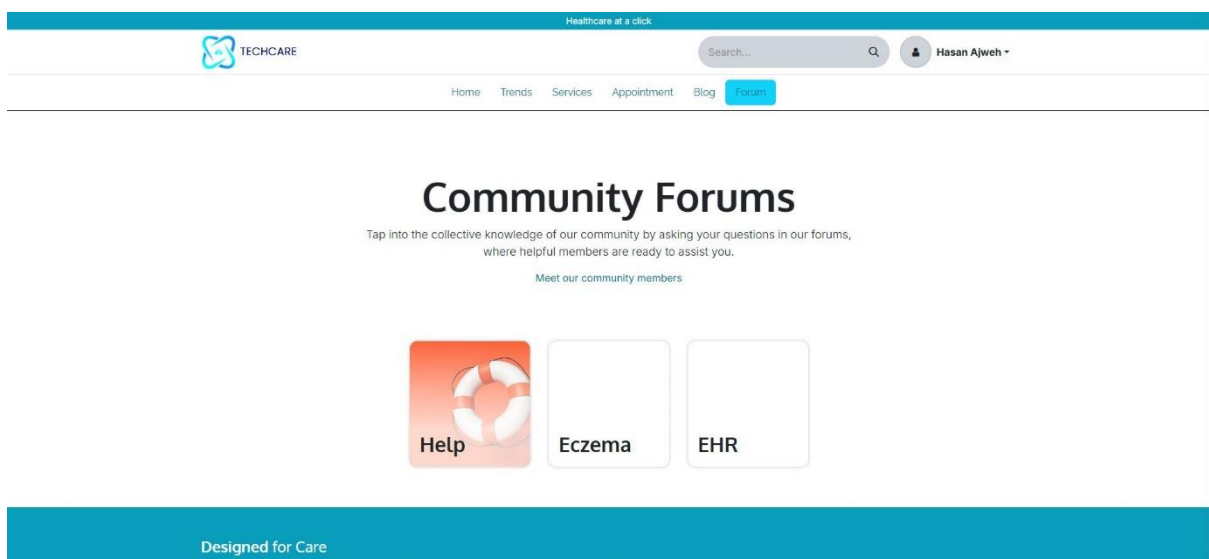


Figure 19 Forum interface

Post creation

The screenshot shows the 'EHR / Ask your question' page. On the left is a user profile for Hasan Ajweh (2504 XP) with navigation links for All Posts, My Posts, Favourites, People, and Badges. Below these are 'Moderation tools' (To Validate, Flagged, Closed) and 'My forums' (Eczema, Help). The main content area has a 'Title' field with the text 'what would be the resonos for constant headache and feeling sleepy all the time'. Below it is a 'Description' text area containing 'my constant headache feels like pressure on the sides of my head, this pain leads to me being lazy and sleepy all the time'. A 'Tags' dropdown menu is set to 'Tags'. At the bottom are 'POST YOUR QUESTION' and 'Discard' buttons.

Figure 20 Post creation

User's posts

The screenshot shows the 'EHR' page with a list of posts. The user profile for Hasan Ajweh (2506 XP) is visible on the left. The post list includes a search bar, a 'NEW POST' button, and columns for Replies, Views, and Activity. One post is shown with the title 'what would be the resonos for constant headache and feeling sleepy all the time', 0 replies, 1 view, and posted 3 hours ago.

	Replies	Views	Activity
what would be the resonos for constant headache and feeling sleepy all the time	0	1	3 hours ago

Figure 21 User's Posts

Contact us interface

Healthcare at a click

TECHCARE

Search... Hasan Ajweh

Home Trends Services Appointment Blog Forum

Contact us

Contact us about anything related to our company or services. We'll do our best to get back to you as soon as possible.

Name * Mohammed Saeed

Phone Number +970 598303142

Email * 201121@pu.edu.ps

Company

Subject * help me find my posts

Question * how to find my posts in forums?

My Company
P720, Hebron, Palestine
+970 594303163
info@techcare.com

Figure 22 Contact us interface

Customer Care

Healthcare at a click

TECHCARE

Search... Hasan Ajweh

Home Trends Services Appointment Blog Forum

Submit a Ticket

Your Name * Mohammed Saeed

Your Email * 201121@ppu.edu.ps

Subject * Appointment change

Description I want to change my appointment but the one I want is Taken

Attachment Choose file No file chosen

SUBMIT TICKET

Designed for Care

We are a dedicated team committed to improving healthcare access for our community in Palestine. Our mission is to connect people with essential medical services through innovative, user-friendly

TechCare
Palestine, Hebron

info@techcare.com

Figure 23 Ticket creation

Sent email for the ticket

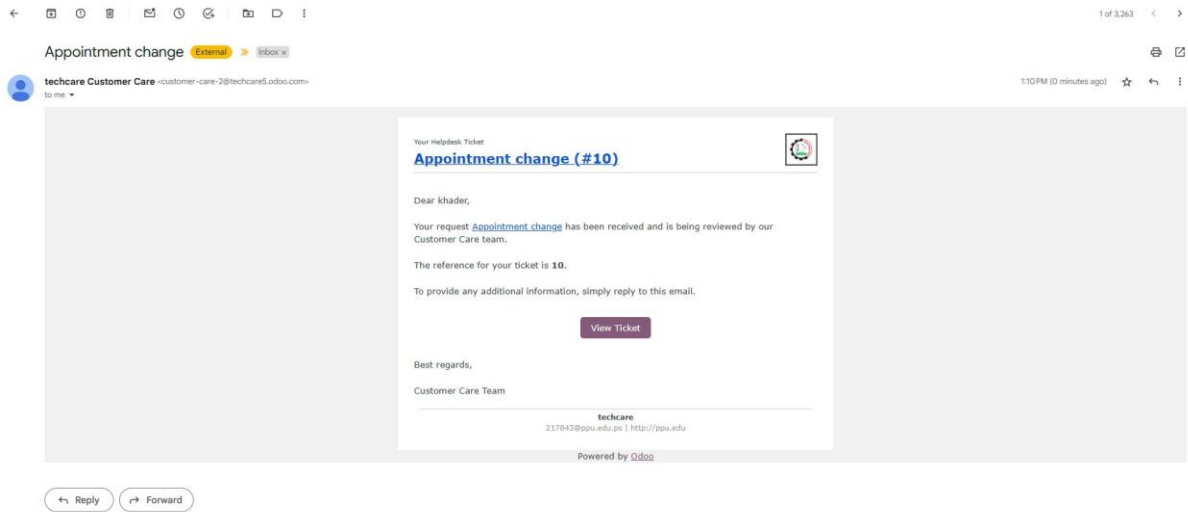


Figure 24 Email

Adding clinics

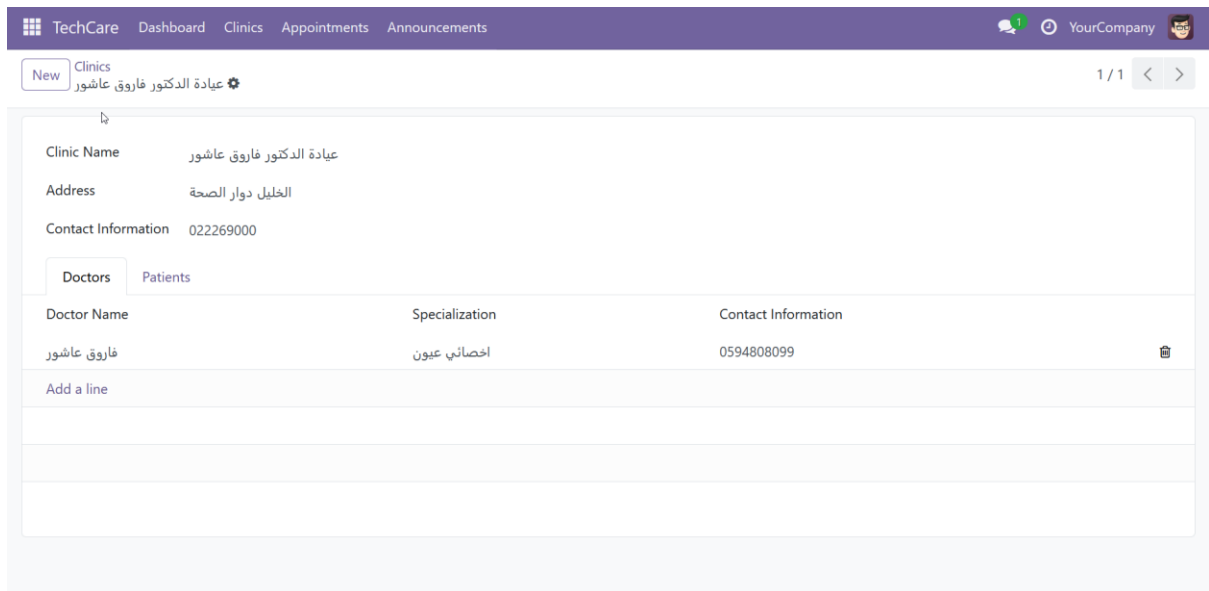


Figure 25 Adding a clinic

Making an Appointment

TechCare Dashboard Clinics Appointments Announcements

New Appointments
حسن عجوه

1 / 1 < >

Clinic عيادة الدكتور فاروق عاشور

Medical Specialist فاروق عاشور

Appointment Date 02/02/2025 10:00:00

Patient حسن عجوه

Status

Draft Confirmed Done Canceled

Figure 26 Appointments making interface

Adding a patient

TechCare Dashboard Clinics Appointments Announcements

New Clinics
عيادة الدكتور فاروق عاشور

1 / 1 < >

Clinic Name عيادة الدكتور فاروق عاشور

Address الخليل دوار الصحة

Contact Information 022269000

Doctors Patients

Patient Name	ID Number	Birth Date
حسن عجوه	408801512	02/01/2003
Add a line		

Figure 27 Patients added to a clinic

* patient id will be validated

Announcement interface

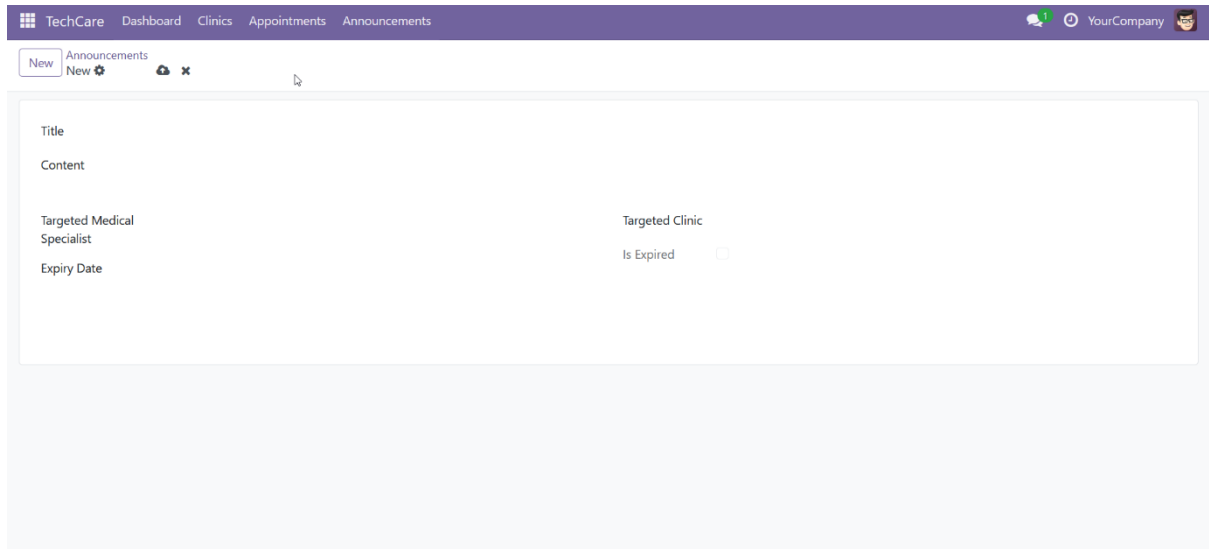


Figure 28 Announcements

Dashboard's interface

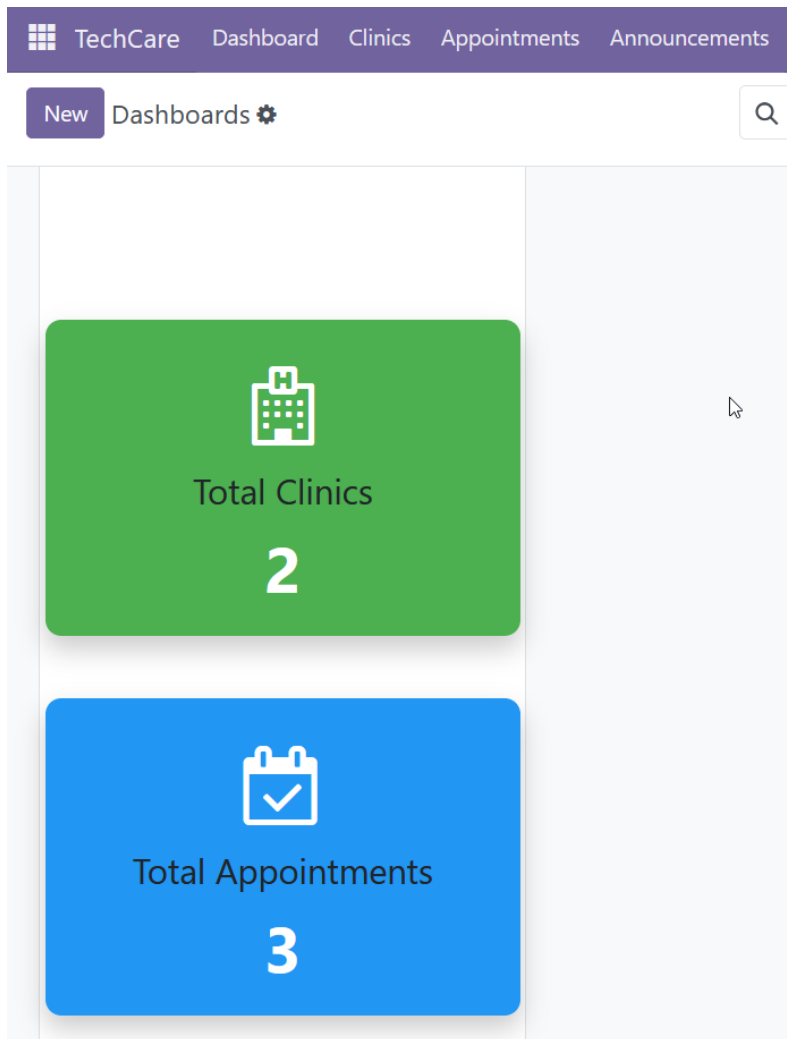


Figure 29 Dashboard

3.6 TechCare Module

The TechCare module is a custom add-on to the Odoo ERP platform. It is made to help clinics manage their work and appointments easily. The module is designed to work smoothly with Odoo's system and focuses on being simple to use and easy to grow as needed.

This module has the following features:

1. **Dashboard:** A main screen where you can see and manage everything in one place.
2. **Clinics Menu:** A section where you can add clinics, medical specialist, and patients to keep things organized.
3. **Appointments Menu:** Helps with scheduling and managing appointments quickly and easily.
4. **Announcements Menu:** A place to share important updates or news with the team

The menus are set up like all other Odoo modules, so they are flexible and easy to use. This makes managing clinics less stressful and more efficient.

3.7 Class Diagram

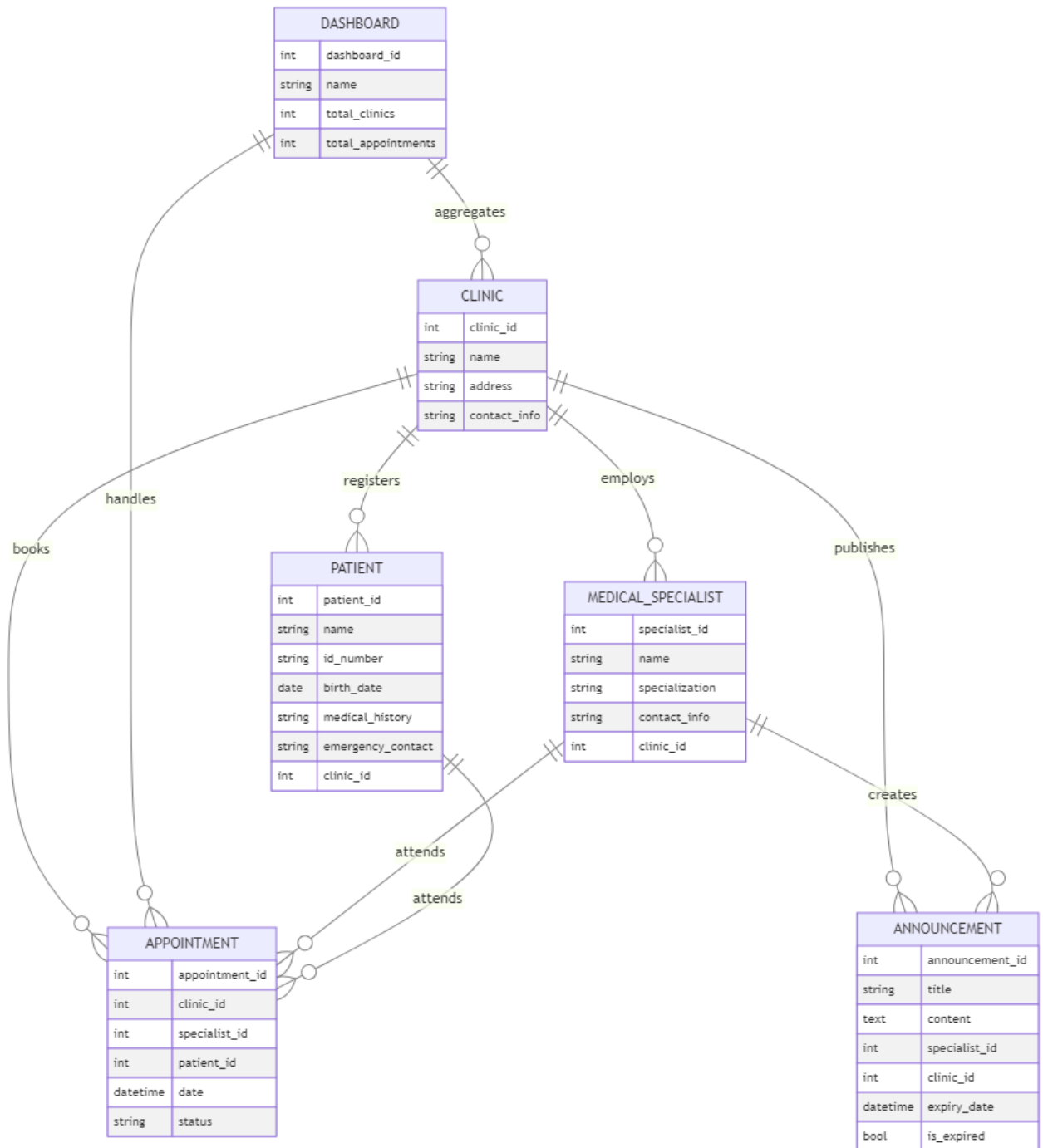


Figure 30 Class diagram

3.8 Mapping & Normalization

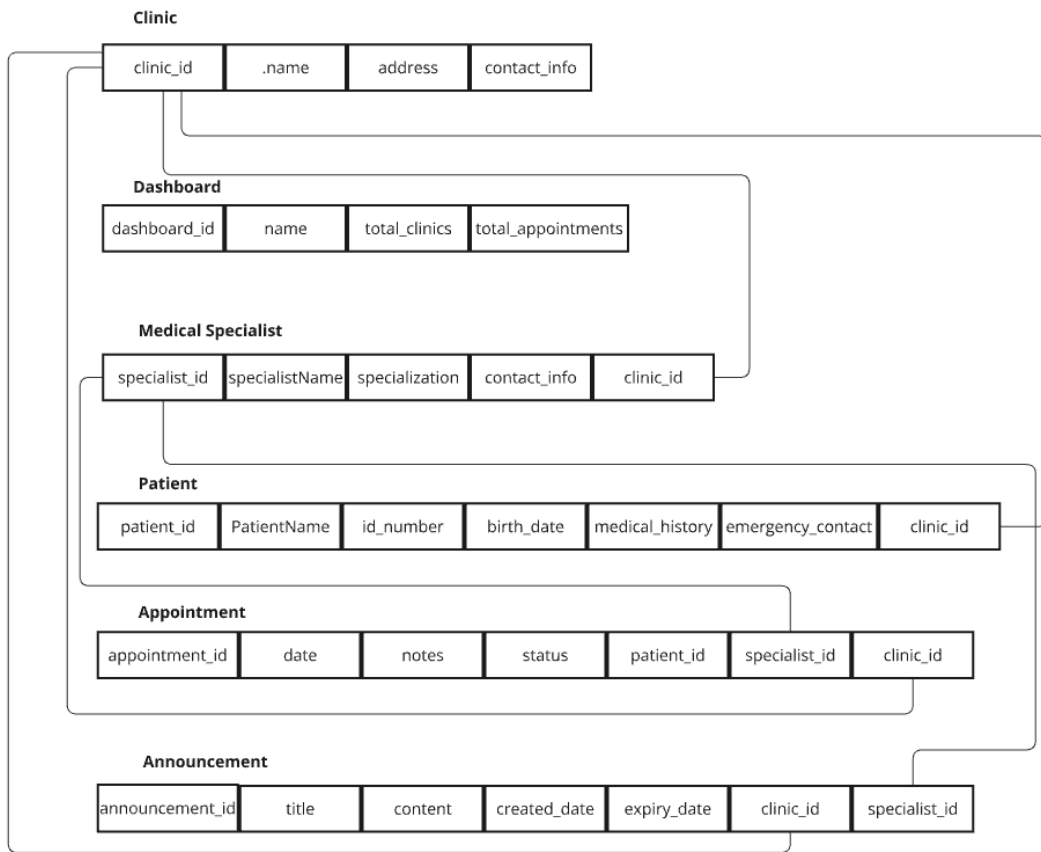


Figure 31 Mapping & normalization

3.9 Description of database tables

Table: Clinic

Table 7 Clinic's table

Attribute	Data Type	Unique	Length	Null	Description
clinic_id	Integer	Yes	N/A	No	Primary Key, unique identifier for the clinic.
name	String	No	255	No	Name of the clinic.
address	String	No	255	No	Physical address of the clinic.

Attribute	Data Type	Unique	Length	Null	Description
contact_info	String	No	100	No	Contact information for the clinic.

Table: Dashboard

Table 8 Dashboard's table

Attribute	Data Type	Unique	Length	Null	Description
dashboard_id	Integer	Yes	N/A	No	Primary Key, unique identifier for the dashboard.
name	String	No	255	No	Name of the dashboard.
total_clinics	Integer	No	N/A	No	Total number of clinics associated.
total_appointments	Integer	No	N/A	No	Total number of appointments handled.

Table: Medical Specialist

Table 9 Medical specialist's table

Attribute	Data Type	Unique	Length	Null	Description
specialist_id	Integer	Yes	N/A	No	Primary Key, unique identifier for the specialist.
name	String	No	255	No	Name of the medical specialist.
specialization	String	No	100	Yes	Specialist's area of expertise.
contact_info	String	No	100	Yes	Specialist's contact information.
clinic_id	Integer	No	N/A	No	Foreign Key referencing Clinic.

Table: Patient*Table 10 Patient's table*

Attribute	Data Type	Unique	Length	Null	Description
patient_id	Integer	Yes	N/A	No	Primary Key, unique identifier for the patient.
name	String	No	255	No	Name of the patient.
id_number	String	Yes	100	No	Unique ID number for the patient.
birth_date	Date	No	N/A	No	Patient's date of birth.
medical_history	String	No	1000	Yes	Medical history of the patient.
emergency_contact	String	No	100	Yes	Emergency contact information.
clinic_id	Integer	No	N/A	No	Foreign Key referencing Clinic.

Table: Appointment*Table 11 Appointment's table*

Attribute	Data Type	Unique	Length	Null	Description
appointment_id	Integer	Yes	N/A	No	Primary Key, unique identifier for the appointment.
clinic_id	Integer	No	N/A	No	Foreign Key referencing Clinic.
doctor_id	Integer	No	N/A	No	Foreign Key referencing Medical Specialist.
patient_id	Integer	No	N/A	No	Foreign Key referencing Patient.
date	DateTime	No	N/A	No	Appointment date and time.
status	enum	No	50	No	Status of the appointment (Draft, Confirmed, etc.).

Table: Announcement*Table 12 Announcement's table*

Attribute	Data Type	Unique	Length	Null	Description
announcement_id	Integer	Yes	N/A	No	Primary Key, unique identifier for the announcement.
title	String	No	255	No	Title of the announcement.
content	Text	No	N/A	No	Content of the announcement.
doctor_id	Integer	No	N/A	No	Foreign Key referencing Medical Specialist.
clinic_id	Integer	No	N/A	No	Foreign Key referencing Clinic.
expiry_date	DateTime	No	N/A	No	Expiration date of the announcement.
is_expired	Boolean	No	N/A	Yes	Computed field indicating if the announcement is expired.

Chapter 4: Implementation

4.1 Introduction

In this chapter, we will discuss the implementation phase of the system development process. Here, we will focus on the practical aspects of transforming design and requirements into an effective system. This chapter will cover multiple topics related to implementation, including programming languages, frameworks, tools, and methodologies used during the development process.

4.2 TechCare Implementation Process

The implementation of TechCare was carried out in two main stages:

- Customization.
- Programming.

4.2.1 Customization Phase

The Customization Phase focused on adjusting Odoo's existing modules and features to match the specific requirements of TechCare. Here's what we did to achieve this:

Website Module

With Odoo's drag-and-drop website builder, we created a simple and responsive interface for the platform. This tool made it easy to arrange elements visually, making sure that the website looked professional and was easy to use. Odoo's IDE editor allowed us to edit the frontend code.

AI Chatbot Integration

We integrated a JavaScript-based AI chatbot into the platform, configured through Odoo's IDE. (<script> tag)

4.2.2 Programming Phase

the Programming stage involved creating a custom module to enhance Odoo's functionality, allowing us to develop a more efficient digital healthcare platform.

We divided the programming Phase into distinct streams for better focus and efficiency:

TechCare Ai

Programming

- .NET Framework and MVC
- C#



- HTML, CSS, and JavaScript



From Ai Studio We can get the endpoint, deploymentId and API Key to ensure fetching and processing from the model deployed in the Studio and put them into a Class, the code can be found besides the training/testing interface related to the model.

```
2 using System.Text;
3 using System.Text.Json;
4
5 4 references
6 public class OpenAIService
7 {
8     private readonly HttpClient _httpClient;
9     private readonly string _endpoint;
10    private readonly string _apiKey;
11    private readonly string _deploymentId;
12
13    0 references
14    public OpenAIService(IConfiguration configuration)
15    {
16        _httpClient = new HttpClient();
17        _endpoint = "https://21784-m482dwuo-eastus2.openai.azure.com/";
18        _apiKey = "GDj16h3USwFAMkdfEg5mIussTFFvjLMKi09DBY1qiHDKI1iCnSSZJQQJ99ALACHYHv6XJ3w3AAAAACOGmk55";
19        _deploymentId = "gpt-4";
20    }
21 }
```

Deployment

Azure AI Studio:

- A low-code tool that put us on the first step of using azure ai services.

Azure WebApp Service:

- Deployed the chatbot as a web application.

Script Embedding for Integration:

- Generated the chatbot's JavaScript widget code using the Azure WebApp service.
- Embedded the code as a script tag into the website frontend code.

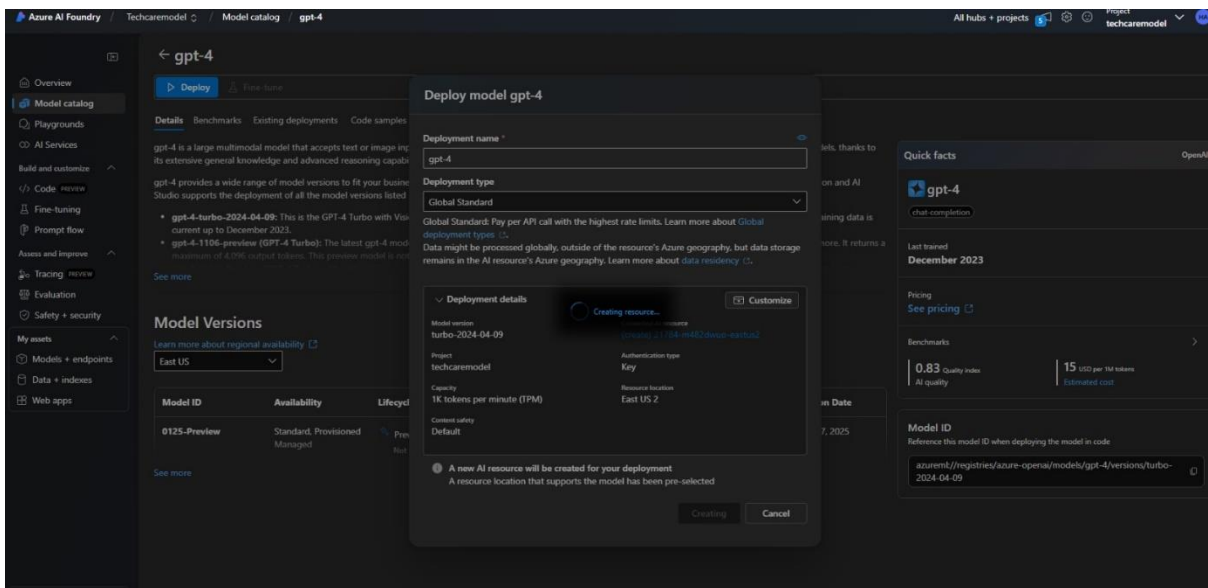


Figure 32 Deploy an Ai model to use it

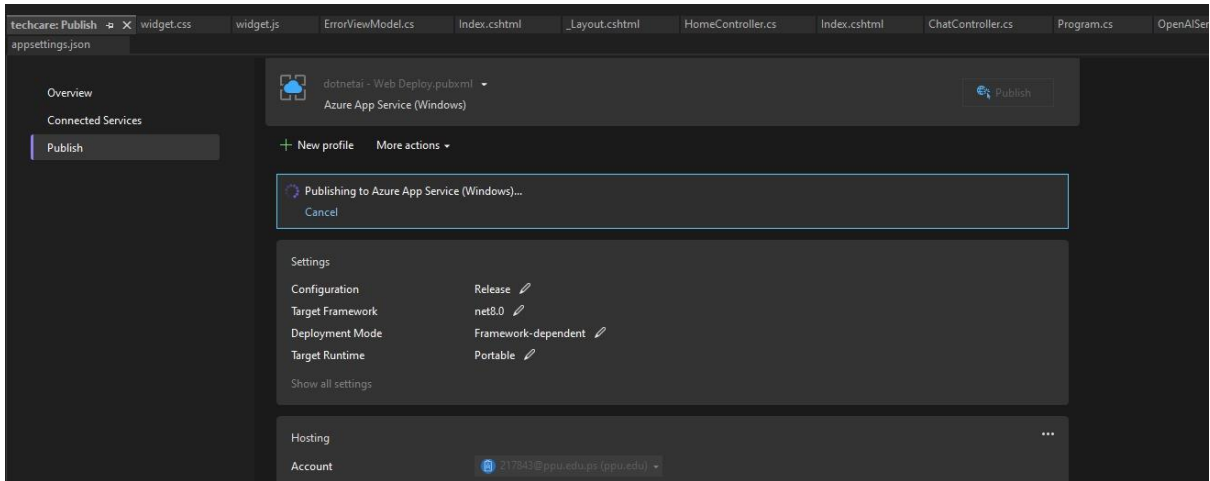


Figure 33 Publishing the .Net project to WebApp Service on Azure

Developing our module couldn't be met through customization alone, it required backend and frontend development

Technologies for Development

- **Pycharm:** Python Based IDE.



- **Python**



- **XML**

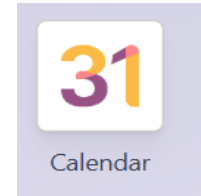


- **PostgreSQL**



- Odoo Calendar Module

Took the advantage of modular system and used prebuilt module in Odoo called Calendar so I can use it in my appointment menu.



We developed a parity check operation that the patient that added to the Clinic must be holding a real id number a Palestinian ID Number specifically We put the required function in the patient.py which is in our module models to meet what expected.

ID parity check code

```

@api.depends('id_number')
def _compute_is_id_valid(self):
    """
    Compute method to validate the ID number whenever it is modified.
    Updates the 'is_id_valid' field based on the result of the validation.
    """
    for record in self:
        record.is_id_valid = self._check_id_parity(record.id_number)

def _check_id_parity(self, id_number):
    """
    Validates Palestinian ID numbers using a parity check algorithm.

    Args:
        id_number (str): The ID number to validate.

    Returns:
        bool: True if the ID is valid, False otherwise.
    """
    if not (id_number and id_number.isdigit() and len(id_number) == 9):
        return False

    digits = List(map(int, id_number))
    total = sum(
        digit if idx % 2 == 0 else sum(divmod(digit * 2, 10))
        for idx, digit in enumerate(digits[:-1])
    )
    check_digit = (10 - total % 10) % 10
    return check_digit == digits[-1]

@api.constrains('id_number')
def _check_valid_id(self):
    """
    Constraint to ensure the validity of the ID number.

    This method is triggered whenever the 'id_number' field is modified.
    It validates the ID number using the parity check algorithm.
    If the ID number is invalid, a validation error is raised to prevent saving.
    """
    for record in self:
        if not self._check_id_parity(record.id_number):
            raise ValidationError(
                "The ID Number is invalid. Please enter a valid Palestinian ID."
            )

```

AJWEH

Figure 34 ID parity check code

4.3 Implementation Problems and Constraints

- **Debugging:** the error tracking in Python and XML was time-consuming due to the unclear error logs.
for example any addition in the manifest file will cause a 500 Internal Server Error, instead of giving clear message that manifest file got extra content.
- We were unable to connect and integrate our module with the website because the required Odoo.sh subscription is too expensive (Odoo.sh: is a cloud platform provided by Odoo for deploying, hosting, and managing Odoo applications and custom modules).

Chapter 5: Testing and Validation

5.1 Introduction

In this chapter, we will discuss the importance of testing and validation for our system.

5.2 Website Testing Scenarios

Website Testing Scenario

Table 13 Website Testing Scenario

Scenario	Steps Performed	Expected Outcome	Results
Schedule Appointment	Navigated to the appointment page. Selected a specialist and time. Confirmed.	Appointment was successfully scheduled	Pass
Use the Blog	Navigated to the blog section. Opened a blog post and read its content. Added a comment on the post.	The post loaded and the comment was successfully displayed under the blog post.	Pass
Chat-with the Ai Chatbot	Opened the chatbot widget. Asked a question.	Chatbot responded and provided quick response.	Pass
Access Forums	Logged into a user account. Opened the forums section and selected a discussion. Posted a reply.	The forum loaded and the reply was successfully saved and displayed.	Pass

5.3 TechCare Module Testing Scenarios

TechCare Module Testing Scenarios

Table 14 TechCare Module Testing Scenarios

Scenario	Steps Performed	Expected Outcome	Results
Adding a Clinic medical specialist and a patient	Navigated to the Clinics Menu. Added a new clinic with its staff details. Saved the changes.	The clinic was successfully added and appeared in the clinics list.	Pass
Making sure that parity check code works correctly	Navigate to the clinics menu. Added the necessary details Moved to the patient and entered a valid and real Palestinian ID Number	Patient added with no validationError message.	Pass
Making sure that parity check code works correctly	Navigate to the clinics menu. Added the necessary details Moved to the patient and entered a wrong ID Number	ValidationError Message Appears "Enter A valid Palestinian ID Number".	Pass
Assigning Appointment functionality	Navigate to the Appointments fill and choose the required details.	Appointment successfully assigned.	Pass
Calendar module functionality	Navigate to the Appointments Menu and choose date in the past	ValidationError Message Appears .	Pass

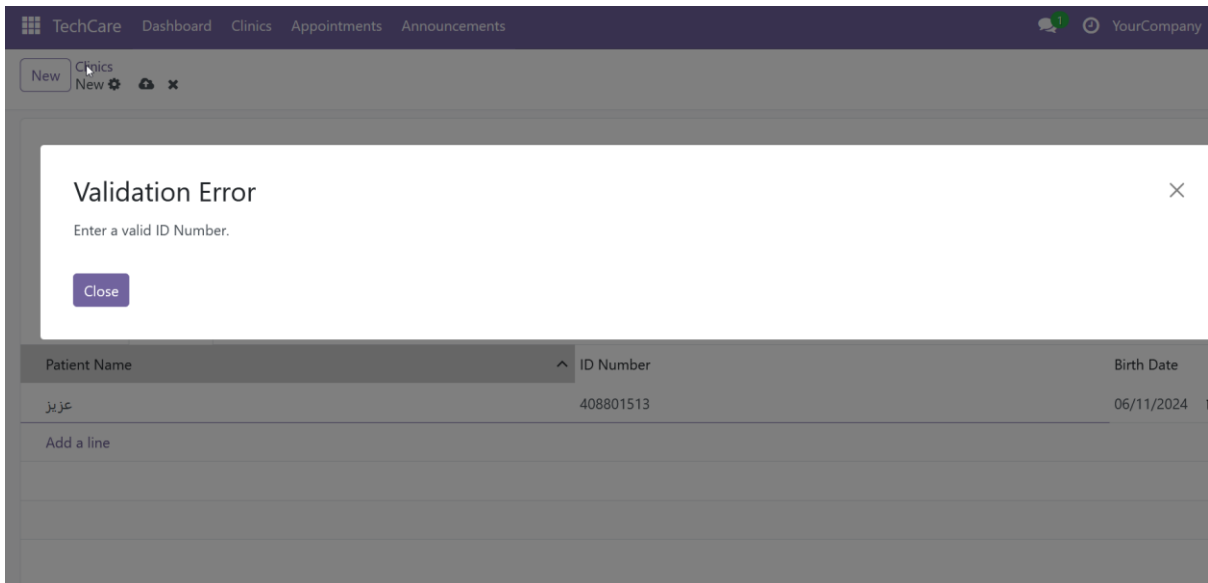


Figure 35 Validation error

When the id number isn't Palestinian or wrong the following validationError message will appear.

If the id number is correct the clinic or the patient will be added successfully:

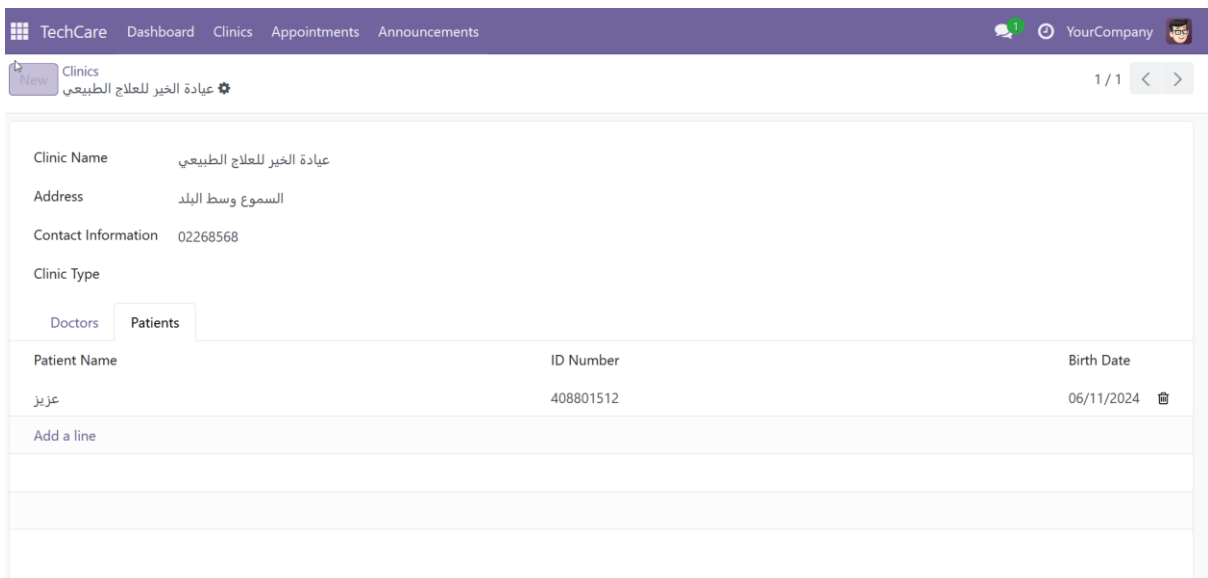


Figure 36 validation succeeded

Chapter 6: Results and Future Work

6.1 Conclusion

TechCare platform made it easier for medical specialists and people in the community to connect by using Odoo ERP to make a digital healthcare system that works well. It combines cool features like AI tools, booking appointments, and a clinic management module, giving a modern way to handle healthcare.

Using Odoo for this project was a smart move because it made the system flexible, easy to upgrade, scalable, and cheaper to make. It also saved time in development. Some big wins were adding the AI chatbot, customizing Odoo modules without problems, and getting good feedback during testing, which showed the system is reliable and easy to use.

This project shows how systems like Odoo can really change healthcare. It makes things easier to access, improves how everything works, and helps patients and doctors communicate better. TechCare proves how technology can help both patients and medical workers, making healthcare more focused on the patient and more efficient.

6.2 Future Work

While TechCare platform is doing its job well now, there is still a lot of room to make it better:

1. Deployment on Odoo.sh for Wider Accessibility

- We will deploy the TechCare module on Odoo.sh making it easily accessible through the Odoo App Store. This will allow organizations worldwide to integrate and use our module.

2. Access Rights Expansion

Following the deployment of our module we will give access rights to each group of users so the module will be more scalable and more efficient so the expansion of access rights means that the module will meet higher hopes and higher levels of users perspectives and features.

3. AI Chatbot Expansion

- We will expand the capabilities of the chatbot.
- Including Image importing, Voice chatting and the latest Ai models available.

4. Integrating laboratory, hospital management and accounting modules.

References

- 1) Sommerville, I. 2015. *Software Engineering, 10th Edition*. Pearson Education.
- 2) OdoO Documentation. 2023. OdoO Documentation:
<https://www.odoo.com/documentation/17.0/>.
- 3) University of Basel. 2009. Multi-Tier Architecture in Database Systems.
University of Basel Database and Information Systems:
<https://dbis.dmi.unibas.ch/publications/2009/db-encyclopedia-multi-tier-architecture/db-encyclopedia-multi-tier-architecture>.