4/28/2025

Final Report

Elderly Care Management System(Care Net)

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

(Student C00270632) Qadeer Hussain
Supervisor: Paul Barry
Software Development Project Year 4

Table of Contents

Table of Figures

Introduction

The purpose of this document is to provide a detailed final report of the Elderly Care Management System(Care Net). This document will reflect on the application development process and the use of technologies, learning outcomes and an overview of the project as a whole.

This includes the project description, description of conformance to specification design, description of learning, review of the project, conclusion and acknowledgement thanking SETU who assisted during the development of the whole project.

Description of Project

The Elderly Care Management System(Care Net), is a secure web application which manages sensitive patient information. Webapp is to be used by nursing home and care homes. At its core, the Care Net application serves as the main interface for both Administrators and Carers, handling all patient data operations through a web interface. The Webapp retrieves the key from the Key server via API calls. The usage of tokens for authentication between the Webapp and the Key Server where all the data is then stored in the Care Net Database. The Key server has its own separate database for encryption keys and API tokens. This multi-tiered approach ensures that patient information remains secure while still being accessible to authorised users.

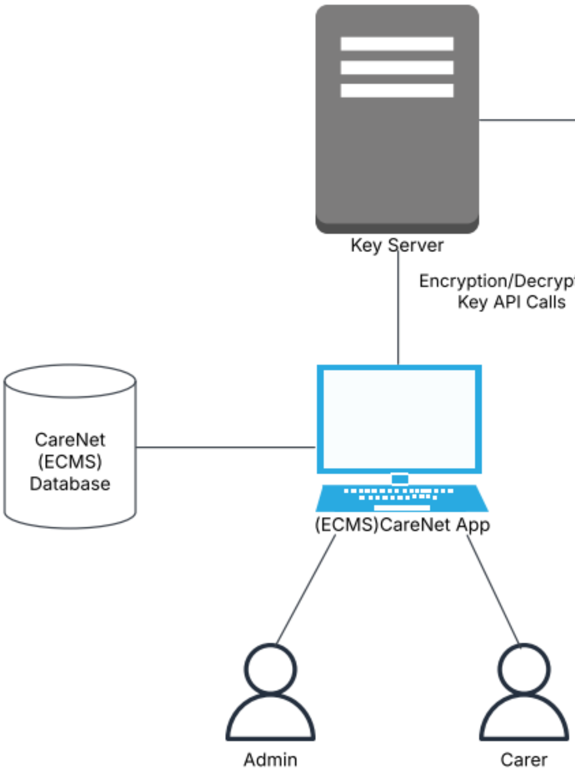


Figure 1: System Architecture

Main functionality

Below is the core functionality for this project

User Login: Only registered users can access the web application. Only Administrators have the authority to create login accounts for Carers.

Security: Data encryption and decryption are handled through strong cryptographic methods to ensure the confidentiality and integrity of patient information.

Patient Profile: Allows the creation of a secure record containing a patient's personal details.

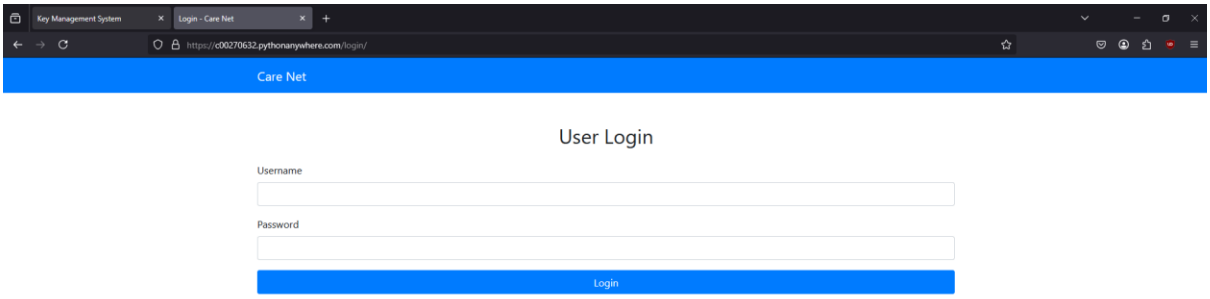
Medical Dashboard: Enables the secure recording and management of a patient's medical information.

Care Planner: Provides tools to create and manage individualized care plans for patients registered in the system.

Roster: Allows Administrators to assign Carers to patients and schedule their daily care activities.

Screenshots for Care Net

Here are some screenshots of the application in use.



User Login

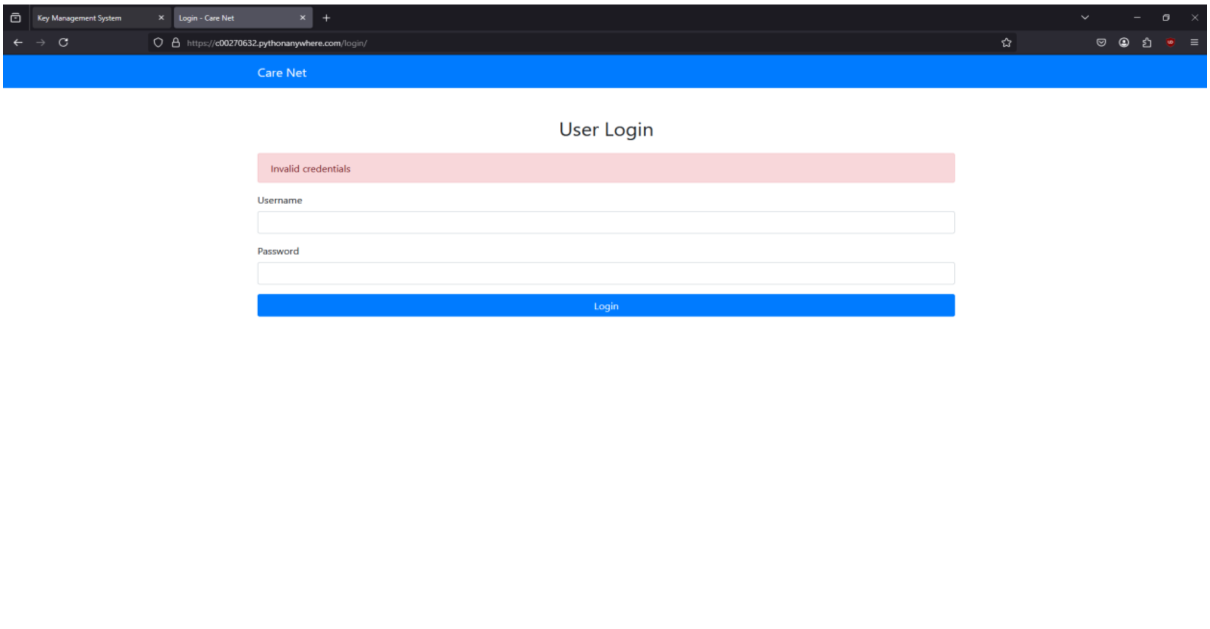
Username

Password

Login

Figure 2: Login Page

This is the Login page. It's the first page you land on when accessing the Care Net web application. Users must enter a valid username and password to gain access. Only registered Carers and Adminstartors whose accounts have been created by an Admin are permitted to log in. This ensures that only authorised personnel can view and manage patient sensitive data.



User Login

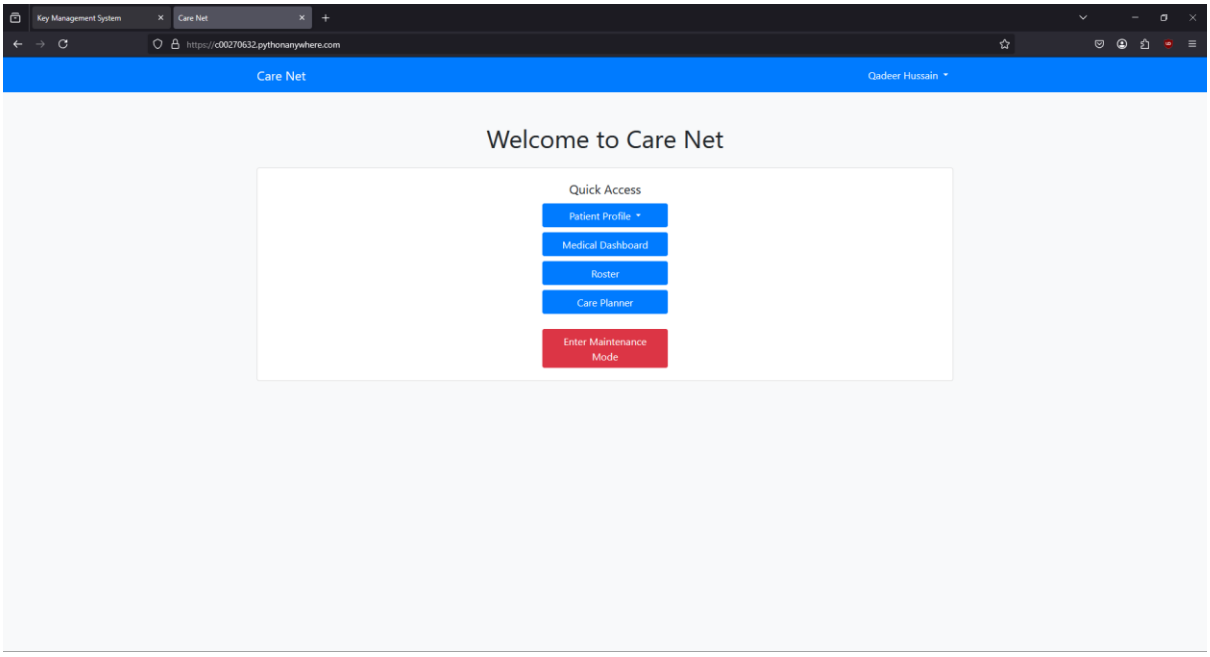
Invalid credentials

Username

Password

Figure 3: Error handling on Login Page

This is the login in screen demonstrating error handling. When users log in with incorrect credentials the error message is displayed.



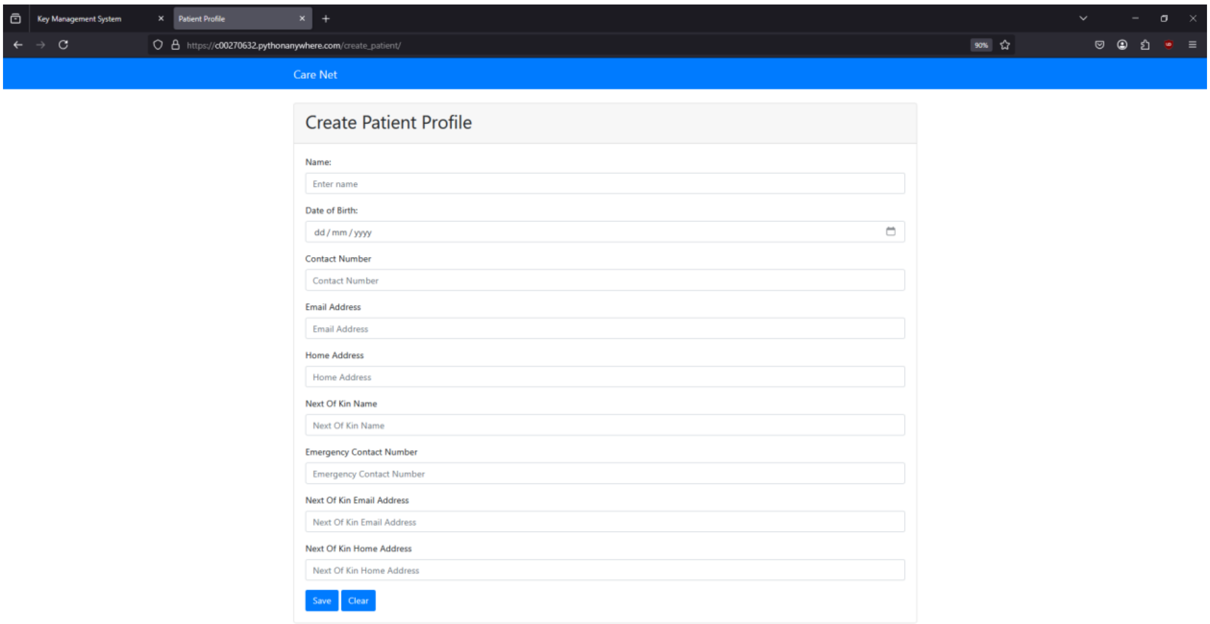
Welcome to Care Net

Quick Access

Figure 4: Home page

After logging in you get to the home page. The home pages serves as the main navigation of the Care Net web application. It presents a quick access menu with several options. Additionally, the Enter maintenance mode button stops the application to perform a re-encryption process. Once re-encryption is completed, the application automatically resumes normal operation.

- Patient Profile: Manage Patient personal information.
- Medical Dashboard: Manage patient medical information.
- Roster: Assign Carer to Patients for the day.
- Care Planner: Manage care plans for patients.



Create Patient Profile

Name:
Enter name

Date of Birth:
dd / mm / yyyy

Contact Number
Contact Number

Email Address
Email Address

Home Address
Home Address

Next Of Kin Name
Next Of Kin Name

Emergency Contact Number
Emergency Contact Number

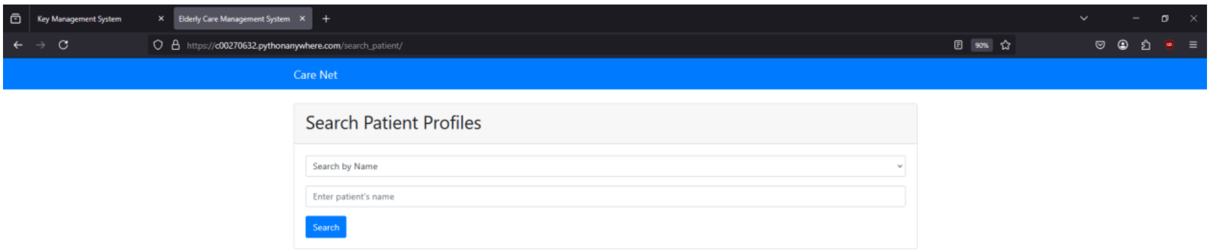
Next Of Kin Email Address
Next Of Kin Email Address

Next Of Kin Home Address
Next Of Kin Home Address

Save Clear

Figure 5: Create Patient Profile

Create patient profile page allows users to register patients into the Care Net system. Once the form is completed and submitted the data is encrypted and securely saved in the database.


Key Management System
Elderly Care Management System
https://c00270632.pythonanywhere.com/search_patient/
90%
Care Net

Search Patient Profiles

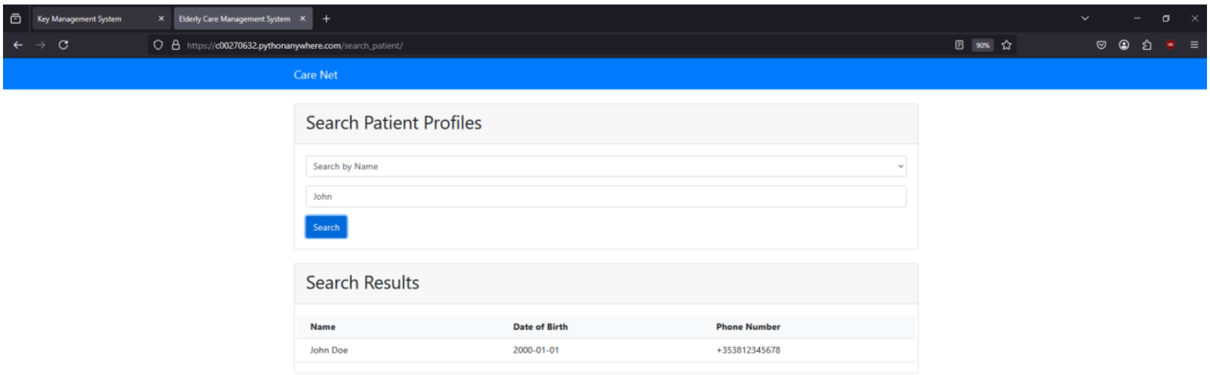
Search by Name

Enter patient's name

Search

Figure 6: Search Patient Profile

The Search Patient Profile allows users to quickly search for patient within the Care Net system. Users can search by Name or Date of Birth. After submitting the system retrieves and displays matching patient profiles.


Key Management System
Elderly Care Management System
https://c00270632.pythonanywhere.com/search_patient/
90%
Care Net

Search Patient Profiles

Search by Name

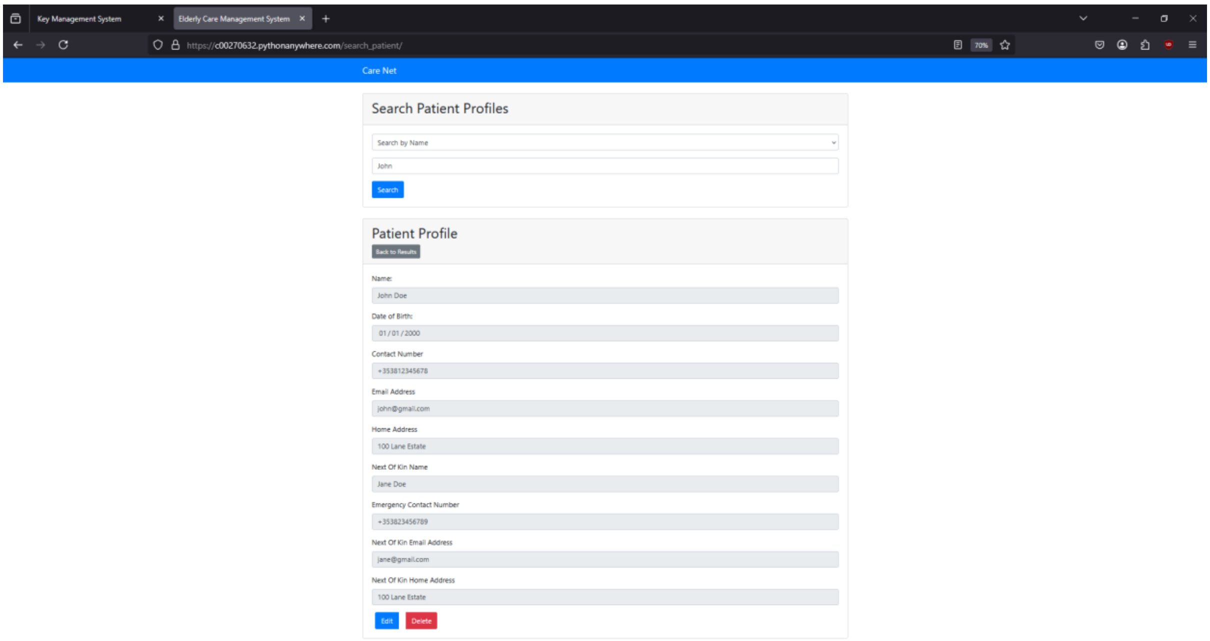
John

Search Results

Name	Date of Birth	Phone Number
John Doe	2000-01-01	+353812345678

Figure 7: Patient Results

This screenshot shows an example of search results for a patient named John Doe. The results are displayed in the results card.


Key Management System
Elderly Care Management System
https://c00270632.pythonanywhere.com/search_patient/
70%
Care Net

Search Patient Profiles

Search by Name

John

Patient Profile

Name: John Doe

Date of Birth: 01 / 01 / 2000

Contact Number: +353812345678

Email Address: john@gmail.com

Home Address: 100 Lane Estate

Next Of Kin Name: Jane Doe

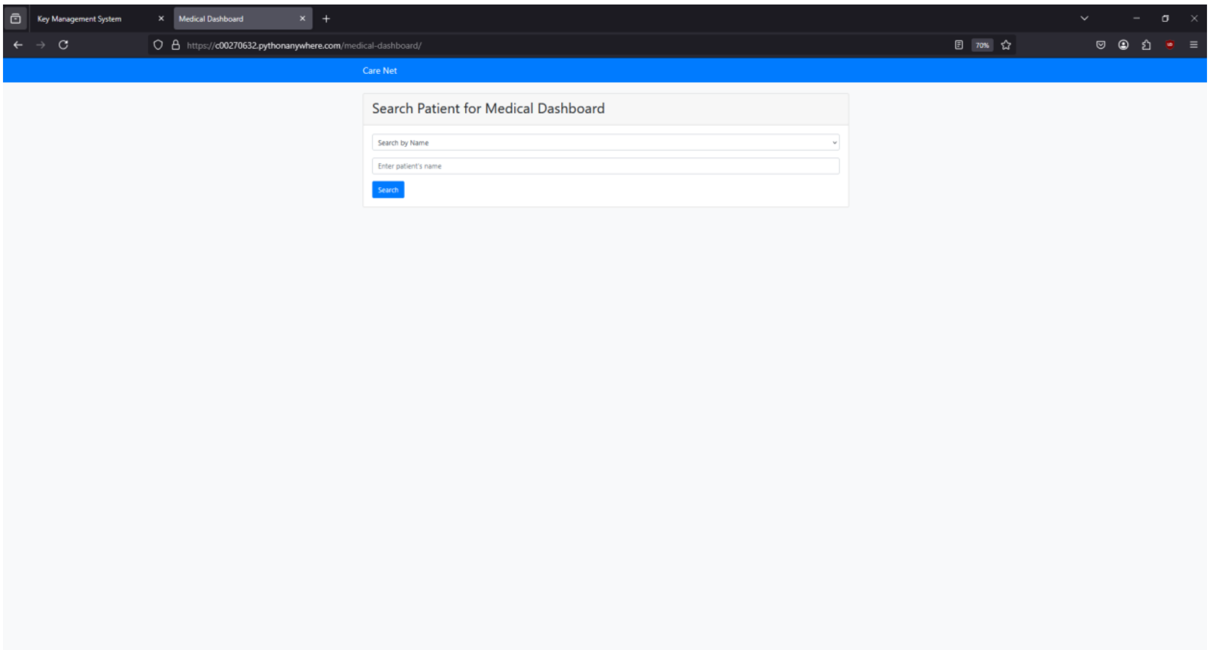
Emergency Contact Number: +353823456789

Next Of Kin Email Address: jane@gmail.com

Next Of Kin Home Address: 100 Lane Estate

Figure 8: View results

After clicking on the patient from the search results shown in Figure 7, users are able to view the full details of the selected patient. Here, the user can edit or delete the patient record. The functionality provides a convenient way to keep patient information up-to-date or to remove records.



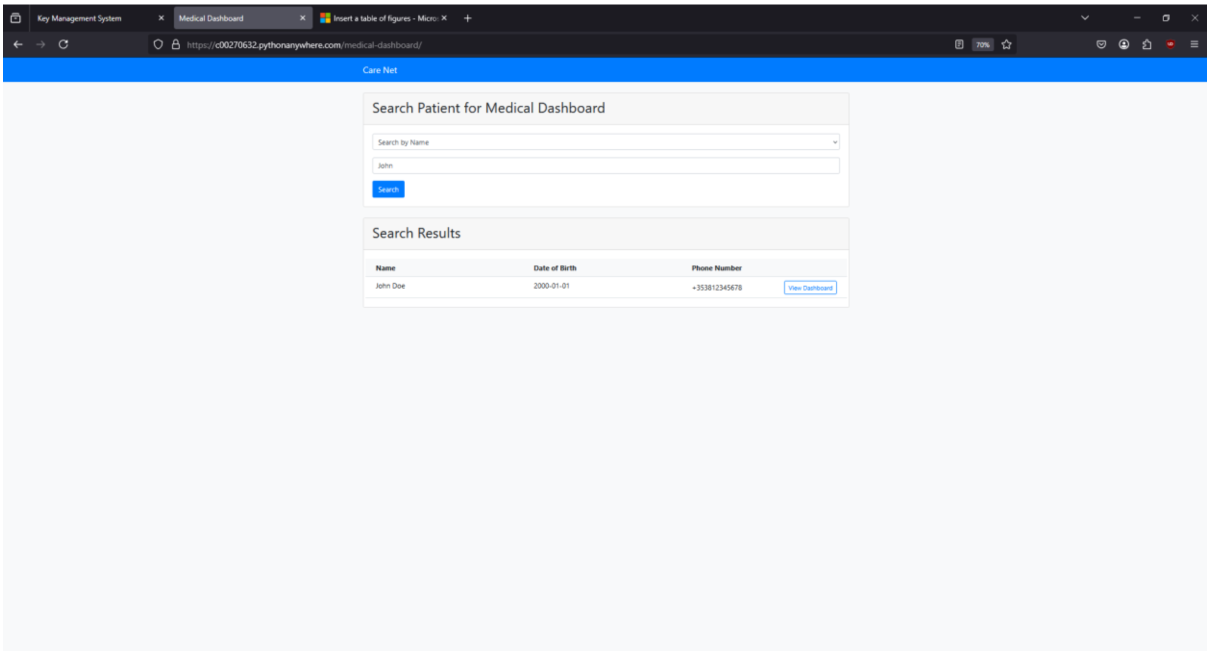
Search Patient for Medical Dashboard

Search by Name

Enter patient's name

Figure 9: Search Patient Medical Dashboard

The Search Patient Medical Dashboard allows users to quickly search for patient medical dashboard within the Care Net system. Users can search by Name or Date of Birth. After submitting the system retrieves and displays matching patient profiles.


Key Management System
Medical Dashboard
Insert a table of figures - Micro:
https://c00270632.pythonanywhere.com/medical-dashboard/
70%
Care Net
Search
View Dashboard

Search Patient for Medical Dashboard

Search by Name

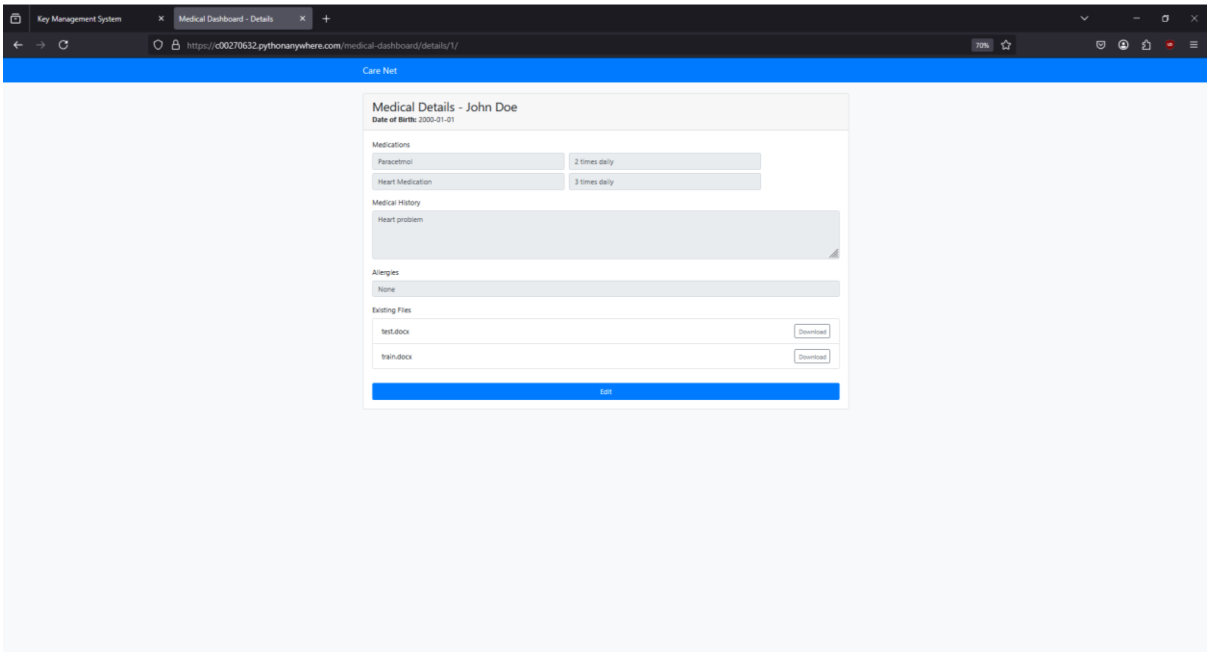
John

Search Results

Name	Date of Birth	Phone Number	
John Doe	2000-01-01	+353812345678	View Dashboard

Figure 10: Patient Medical Dashboard Results

This screenshot shows an example of search results for a patient's medical dashboard. The results are displayed in the results card.


Key Management System
Medical Dashboard - Details
https://c00270632.pythonanywhere.com/medical-dashboard/details/1/
70%
Care Net
Download
Download
Edit

Medical Details - John Doe

Date of Birth: 2000-01-01

Medications

Paracetmol	2 times daily
Heart Medication	3 times daily

Medical History

Heart problem

Allergies

None

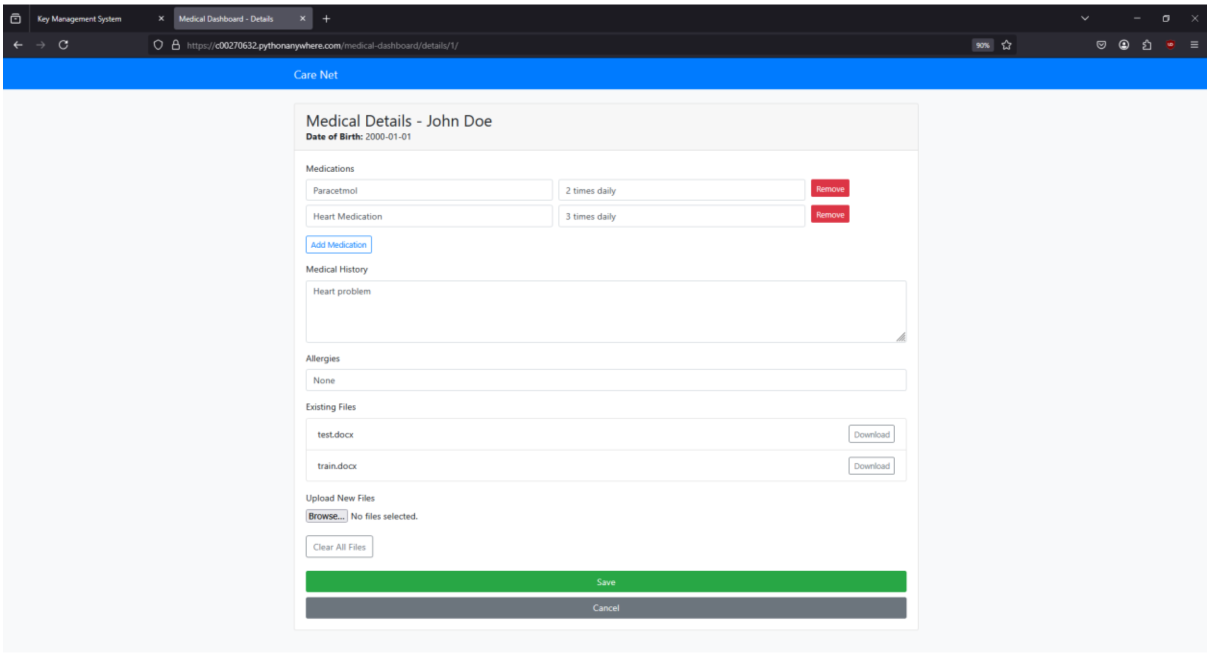
Existing Files

test.docx

train.docx

Figure 11: View Medical Dashboard Results

After clicking on the patient from the search results shown in Figure 10, users are able to view the full details of the patient's Medical Dashboard, provided that medical information already exists. If no medical records are available, the system will display a prompt asking the user if they would like to add new medical details.



Medical Details - John Doe

Date of Birth: 2000-01-01

Medications

Paracetmol	2 times daily
Heart Medication	3 times daily

Medical History

Heart problem

Allergies

None

Existing Files

test.docx

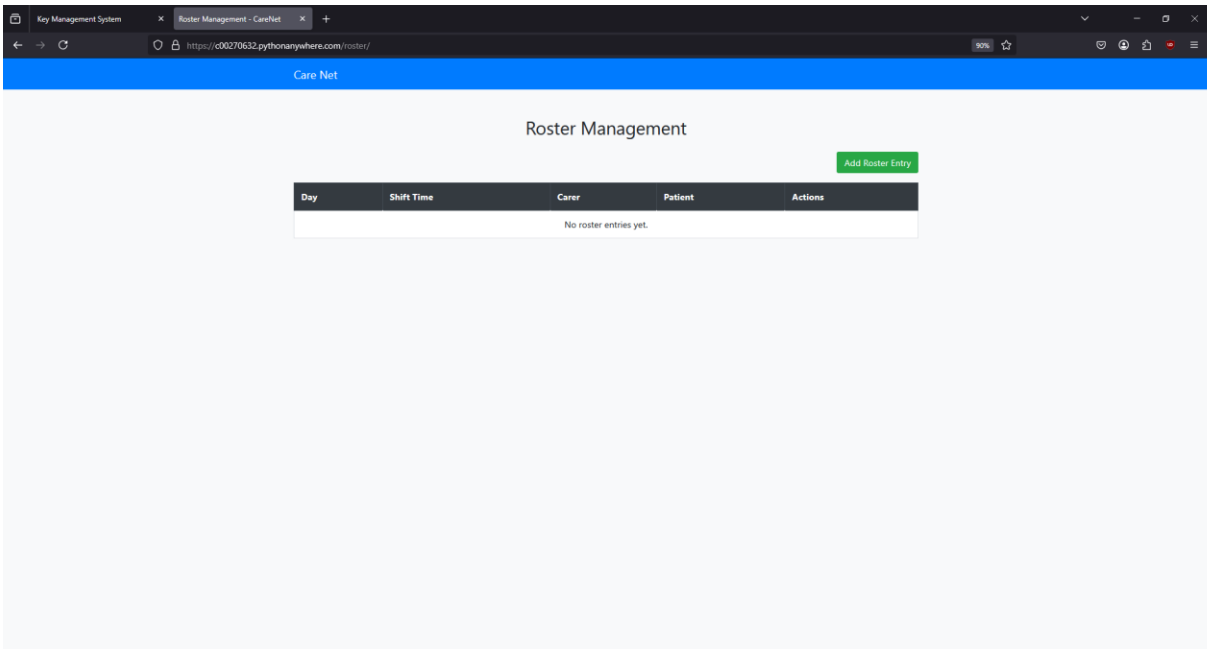
train.docx

Upload New Files

No files selected.

Figure 12: Edit Medical Dashboard Results

Within this dashboard, users can edit, and manage important information such as Medications, Medical History, Allergies and Medical Files. This functionality ensures that patient medical records are always accurate, up-to-date.

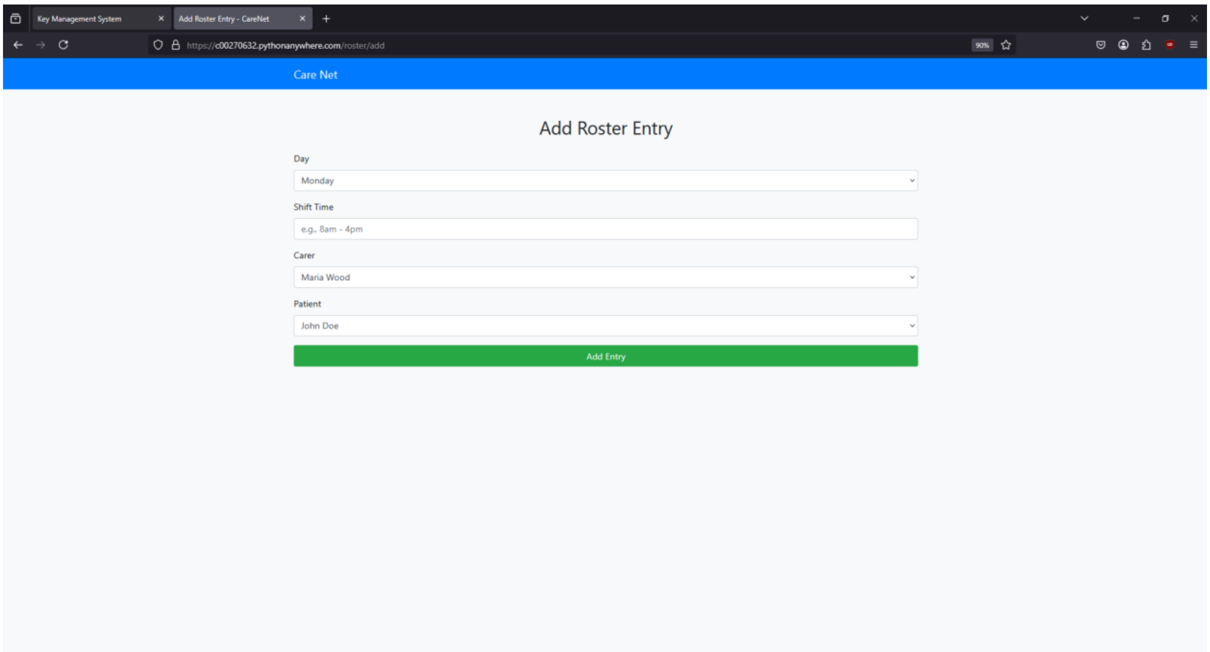

Key Management System
Roster Management - CareNet
https://c00270632.pythonanywhere.com/roster/
90%
Care Net
Add Roster Entry

Roster Management

Day	Shift Time	Carer	Patient	Actions
No roster entries yet.				

Figure 13: Roster

The Roster Management page allows users to view and manage the daily roster within the Care Net system. This page displays a table listing scheduled shifts, including the Day, Shift Time, Assigned Carer, Patient and action.


Key Management System
Add Roster Entry - CareNet
https://c00270632.pythonanywhere.com/roster/add
90%
Care Net
Add Entry

Add Roster Entry

Day: Monday

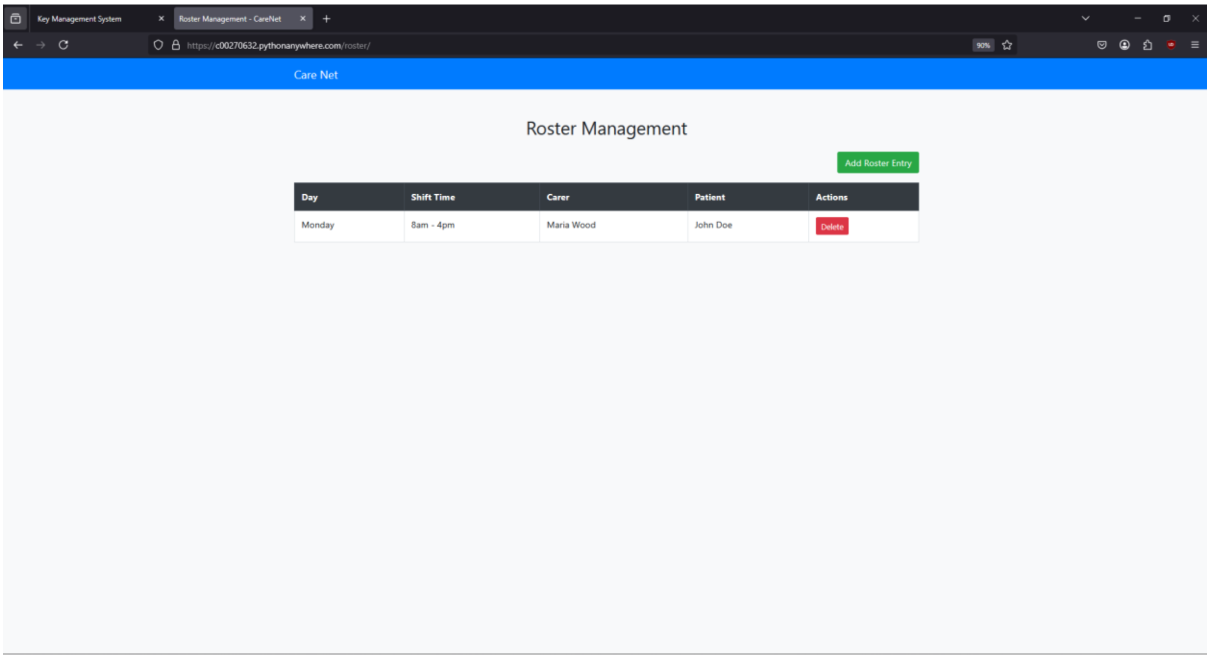
Shift Time: e.g., 8am - 4pm

Carer: Maria Wood

Patient: John Doe

Figure 14: Add Roster Entry

After clicking add roster entry, the user is directed to this page to create a new roster entry. Admin can select the Day, define the Shift Time, and choose both a Carer and a Patient from dropdown menus populated with existing records.

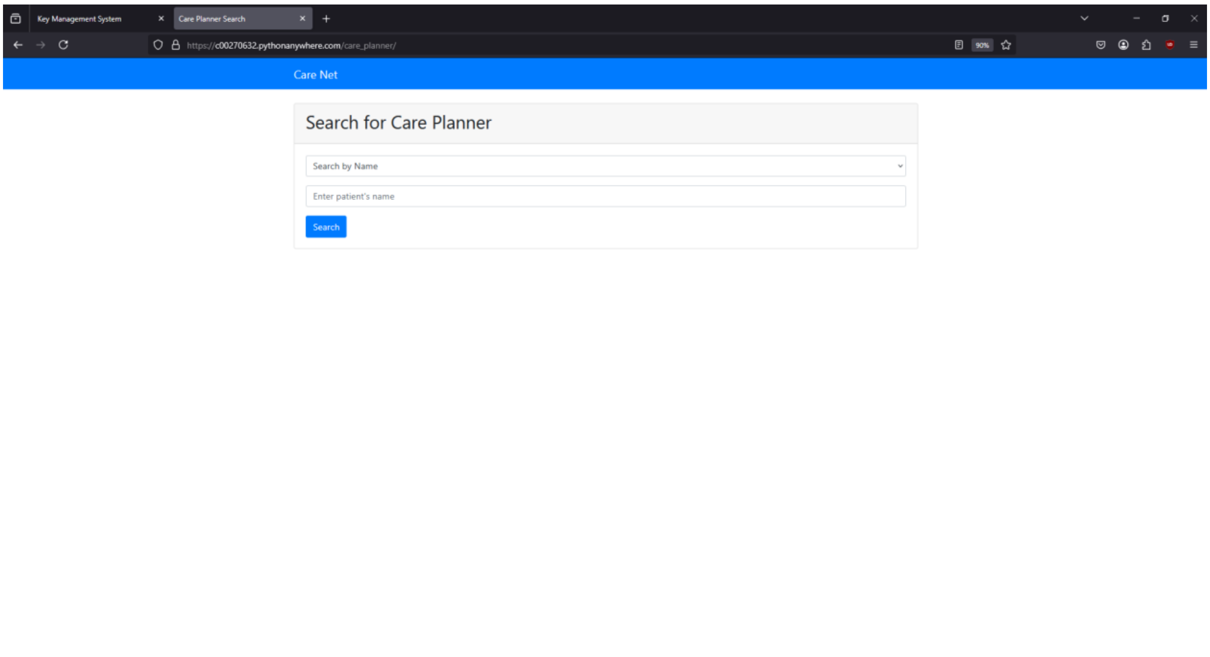

Key Management System
Roster Management - CareNet
https://c00270632.pythonanywhere.com/roster/
90%
Care Net
Add Roster Entry

Roster Management

Day	Shift Time	Carer	Patient	Actions
Monday	8am - 4pm	Maria Wood	John Doe	Delete

Figure 15: Roster Management View

This screenshot shows an example of the roster in the Care Net system. In this example, a shift has been scheduled for Monday from 8 a.m. to 4 p.m. with Carer Maria Wood for Patient John Doe. User is able to delete the roster entry after completion.



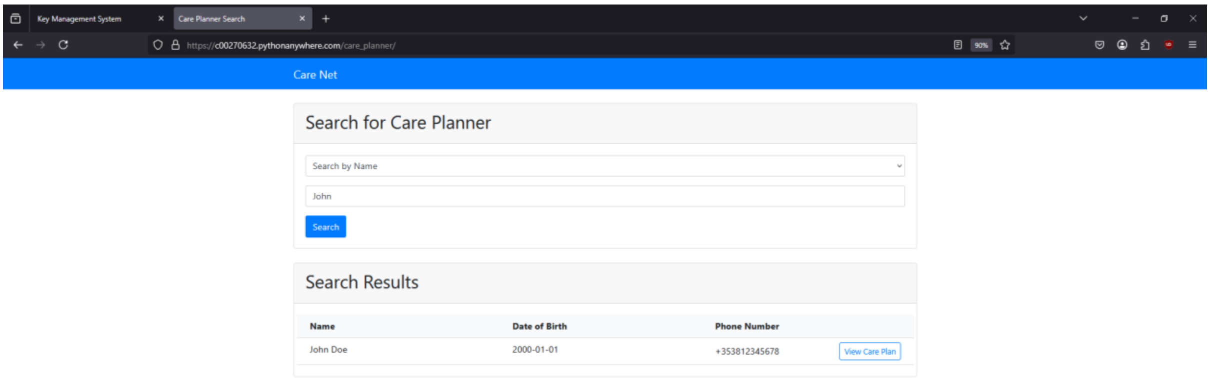
Search for Care Planner

Search by Name

Enter patient's name

Figure 16: Search Care Plan

The Search Patient Care Plan allows users to quickly search for patient care plan within the Care Net system. Users can search by Name or Date of Birth. After submitting the system retrieves and displays matching patient profiles.



Search for Care Planner

Search by Name

John

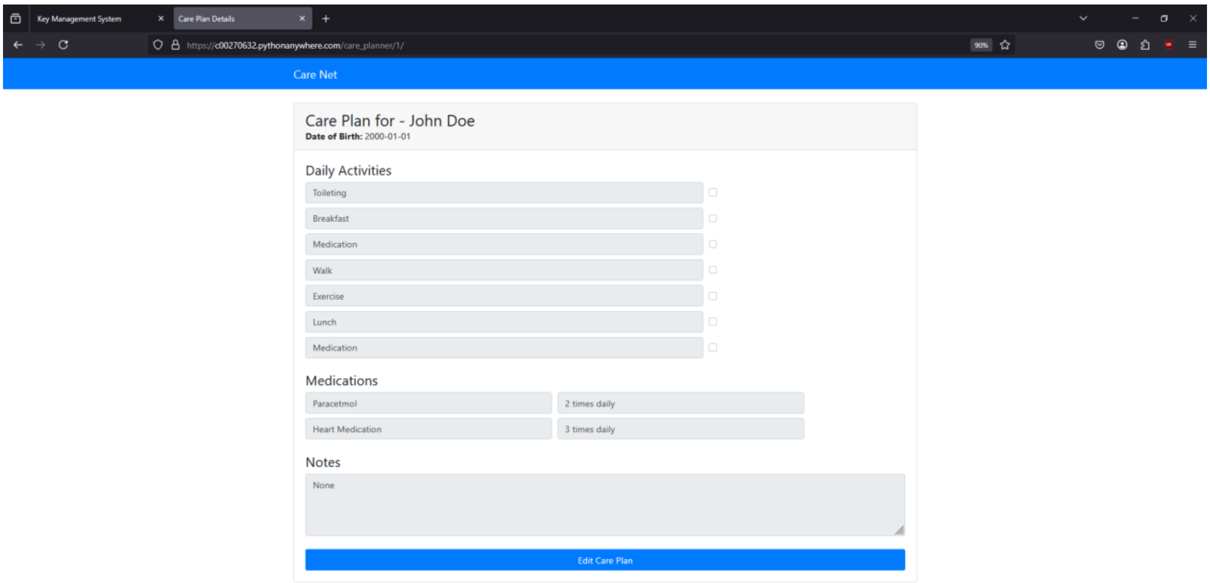
Search

Search Results

Name	Date of Birth	Phone Number	
John Doe	2000-01-01	+353812345678	View Care Plan

Figure 17: Patient Care Plan Result

This screenshot shows an example of search results for a patient's care plan. The results are displayed in the results card.



Care Plan for - John Doe

Date of Birth: 2000-01-01

Daily Activities

Toileting □
Breakfast □
Medication □
Walk □
Exercise □
Lunch □
Medication □

Medications

Paracetmol	2 times daily
Heart Medication	3 times daily

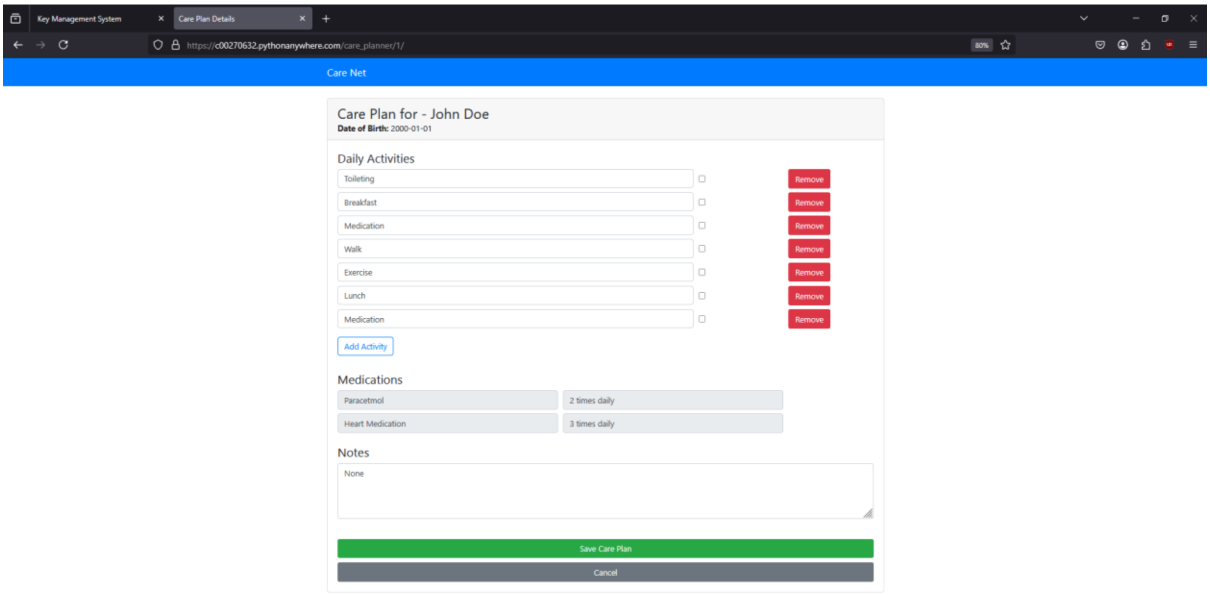
Notes

None

Edit Care Plan

Figure 18: View Patient Care Plan

After clicking on the patient from the search results shown in Figure 17, users are able to view the full details of the patient's Care Plan, provided that the care plan already exists. If no care plans are available, the system will display a prompt asking the user if they would like to add a new care plan.



Care Plan for - John Doe
Date of Birth: 2000-01-01

Daily Activities

Toileting □ Remove
Breakfast □ Remove
Medication □ Remove
Walk □ Remove
Exercise □ Remove
Lunch □ Remove
Medication □ Remove

Medications

Paracetmol	2 times daily
Heart Medication	3 times daily

Notes

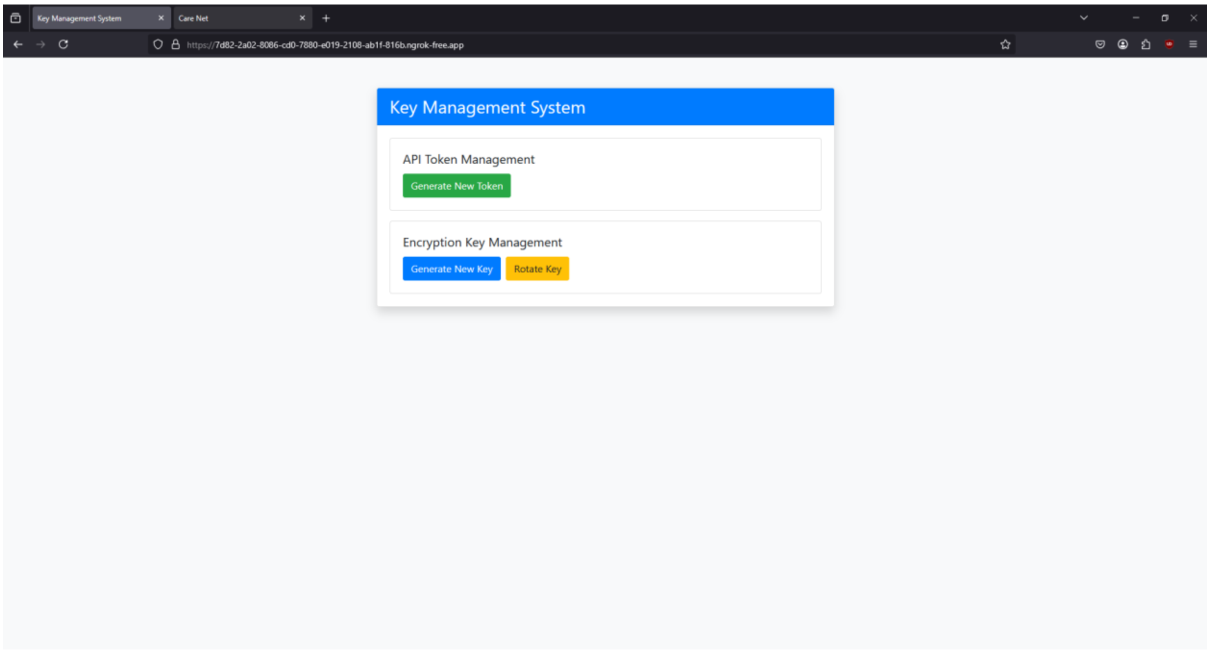
None

Figure 19: Edit Care Plan

Within this Care Plan, users can edit, information such as Daily activities and any notes taking during the day. Activities can be added, modified, or removed as needed, and any completed tasks can be ticked off.

Screenshot for Key Server

Here are some screenshot of the key server in use.


Key Management System
Care Net
https://7d82-2a02-8086-cd0-7880-e019-2108-ab1f-816b.ngrok-free.app

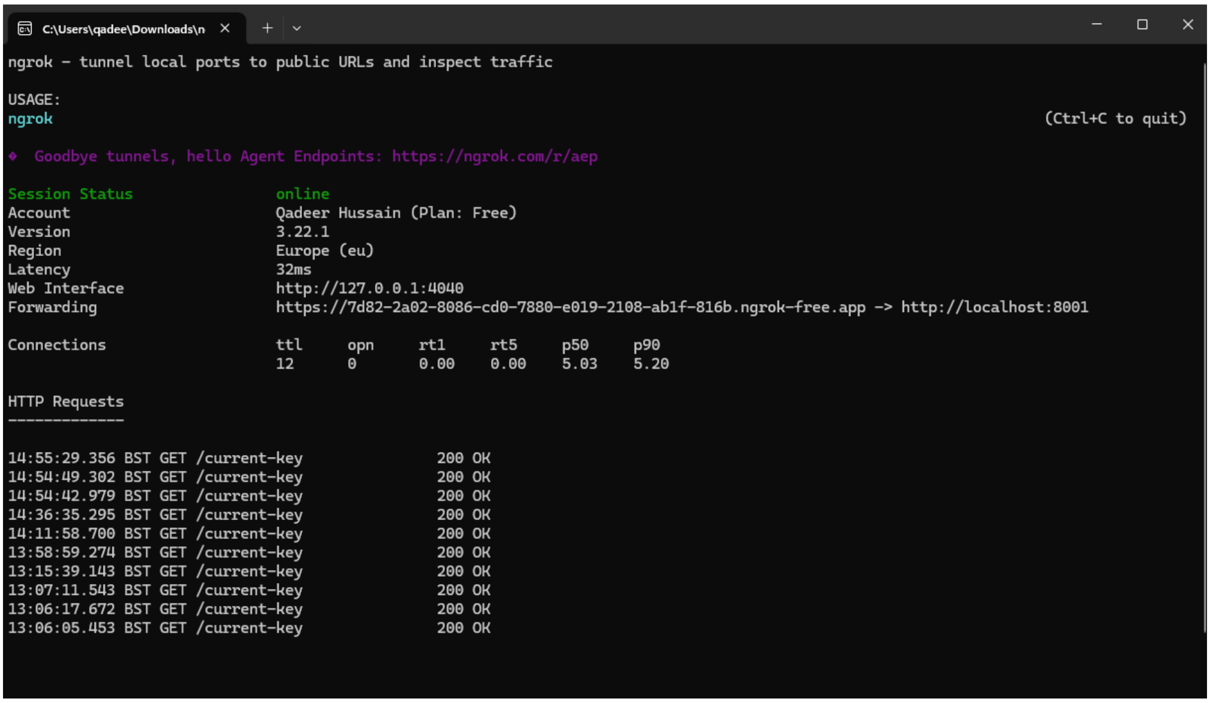
Key Management System

API Token Management

Encryption Key Management

Figure 20: Key Server

The Key Management System provides a web interface for managing API tokens and encryption keys. Users can generate new API tokens for authentication and create or rotate key when needed.

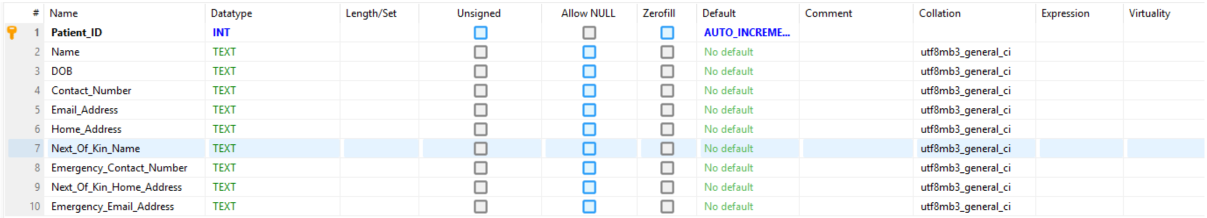


```
ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
ngrok                                                                                          (Ctrl+C to quit)

◆  Goodbye tunnels, hello Agent Endpoints: https://ngrok.com/r/aep

Session Status                online
Account                       Qadeer Hussain (Plan: Free)
Version                       3.22.1
Region                        Europe (eu)
Latency                       32ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://7d82-2a02-8086-cd0-7880-e019-2108-ab1f-816b.ngrok-free.app -> http://localhost:8001

Connections                   ttl     opn     rt1     rt5     p50     p90
                              12      0       0.00    0.00    5.03    5.20

HTTP Requests
-------------

14:55:29.356 BST GET /current-key              200 OK
14:54:49.302 BST GET /current-key              200 OK
14:54:42.979 BST GET /current-key              200 OK
14:36:35.295 BST GET /current-key              200 OK
14:11:58.700 BST GET /current-key              200 OK
13:58:59.274 BST GET /current-key              200 OK
13:15:39.143 BST GET /current-key              200 OK
13:07:11.543 BST GET /current-key              200 OK
13:06:17.672 BST GET /current-key              200 OK
13:06:05.453 BST GET /current-key              200 OK
```

Figure 21: Ngrok Tunnel for Key Server

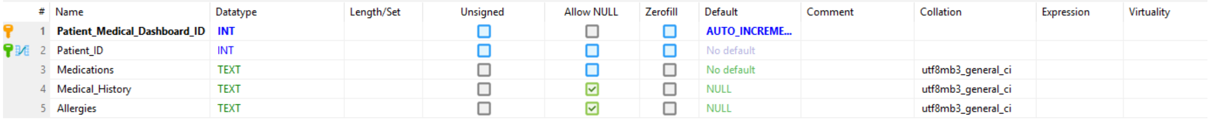
The Key Server runs locally on a laptop, while the Care Net web application is deployed on PythonAnywhere. To enable secure communication between the remote Care Net app and the locally hosted Key Server, Ngrok is used. Ngrok creates a secure tunnel to the local Key Server, exposing it to the internet with a URL.

Screenshots of Database for Care Net



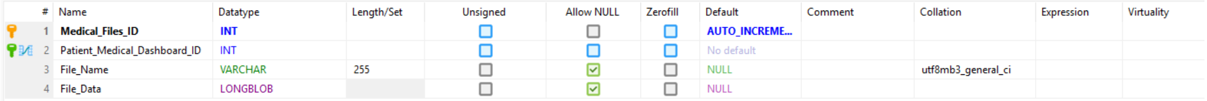
#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Patient_ID**	INT		□	□	□	AUTO_INCREME...				
2	Name	TEXT		□	□	□	No default		utf8mb3_general_ci		
3	DOB	TEXT		□	□	□	No default		utf8mb3_general_ci		
4	Contact_Number	TEXT		□	□	□	No default		utf8mb3_general_ci		
5	Email_Address	TEXT		□	□	□	No default		utf8mb3_general_ci		
6	Home_Address	TEXT		□	□	□	No default		utf8mb3_general_ci		
7	Next_Of_Kin_Name	TEXT		□	□	□	No default		utf8mb3_general_ci		
8	Emergency_Contact_Number	TEXT		□	□	□	No default		utf8mb3_general_ci		
9	Next_Of_Kin_Home_Address	TEXT		□	□	□	No default		utf8mb3_general_ci		
10	Emergency_Email_Address	TEXT		□	□	□	No default		utf8mb3_general_ci		

Figure 22: Patient Profile



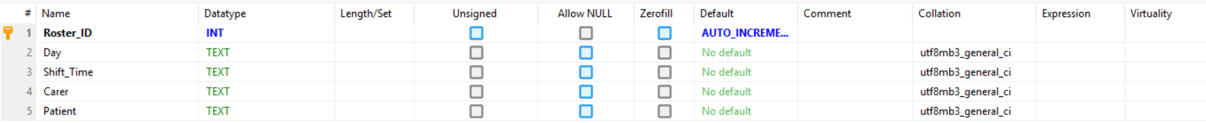
#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Patient_Medical_Dashboard_ID**	INT		□	□	□	AUTO_INCREME...				
2	Patient_ID	INT		□	□	□	No default				
3	Medications	TEXT		□	□	□	No default		utf8mb3_general_ci		
4	Medical_History	TEXT		□	☑	□	NULL		utf8mb3_general_ci		
5	Allergies	TEXT		□	☑	□	NULL		utf8mb3_general_ci		

Figure 23: Medical Dashboard



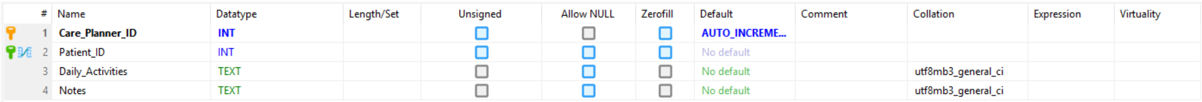
#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Medical_Files_ID**	INT		□	□	□	AUTO_INCREME...				
2	Patient_Medical_Dashboard_ID	INT		□	□	□	No default				
3	File_Name	VARCHAR	255	□	☑	□	NULL		utf8mb3_general_ci		
4	File_Data	LONGBLOB		□	☑	□	NULL				

Figure 24: Medical Files



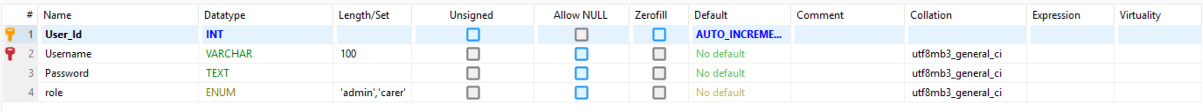
#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Roster_ID**	INT		□	□	□	AUTO_INCREME...				
2	Day	TEXT		□	□	□	No default		utf8mb3_general_ci		
3	Shift_Time	TEXT		□	□	□	No default		utf8mb3_general_ci		
4	Carer	TEXT		□	□	□	No default		utf8mb3_general_ci		
5	Patient	TEXT		□	□	□	No default		utf8mb3_general_ci		

Figure 25: Roster



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Care_Planner_ID**	INT		□	□	□	AUTO_INCREME...				
2	Patient_ID	INT		□	□	□	No default				
3	Daily_Activities	TEXT		□	□	□	No default		utf8mb3_general_ci		
4	Notes	TEXT		□	□	□	No default		utf8mb3_general_ci		

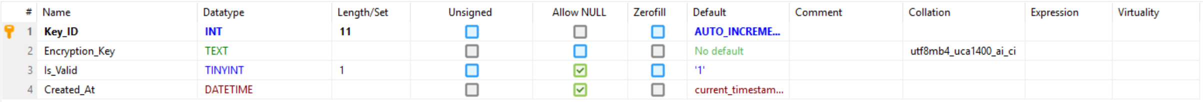
Figure 26: Care Planner



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**User_Id**	INT		□	□	□	AUTO_INCREME...				
2	Username	VARCHAR	100	□	□	□	No default		utf8mb3_general_ci		
3	Password	TEXT		□	□	□	No default		utf8mb3_general_ci		
4	role	ENUM	'admin','carer'	□	□	□	No default		utf8mb3_general_ci		

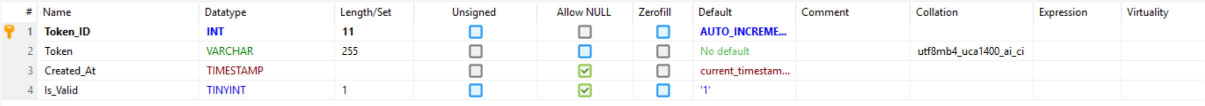
Figure 27: Users

Screenshots of Database for Key Management Server



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Key_ID**	INT	11	□	□	□	AUTO_INCREME...				
2	Encryption_Key	TEXT		□	□	□	No default		utf8mb4_uca1400_ai_ci		
3	Is_Valid	TINYINT	1	□	☑	□	'1'				
4	Created_At	DATETIME		□	☑	□	current_timestam...				

Figure 28: Key Management



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**Token_ID**	INT	11	□	□	□	AUTO_INCREME...				
2	Token	VARCHAR	255	□	□	□	No default		utf8mb4_uca1400_ai_ci		
3	Created_At	TIMESTAMP		□	☑	□	current_timestam...				
4	Is_Valid	TINYINT	1	□	☑	□	'1'				

Figure 29: Api Token

Technologies/Tools Used

Technologies used for Care Net

- Framework: Django
- Front End: Bootstrap, HTML, JS
- Backend: MariaDB and MYSQL, Python,
- Libraries: Cryptography, Secrets

Tools used for Care Net

- GitHub
- Jet Brains IDE (PyCharm)
- MariaDB
- PythonAnywhere

Technologies used for Key Server

- Framework: FastAPI
- Front End: Bootstrap, HTML, JS
- Backend: MariaDB, Python,
- Libraries: Secrets

Tools used for Key Server

- GitHub
- Jet Brains IDE (PyCharm)
- MariaDB
- Postman
- Runs Locally using NGROK (for tunnelling localhost to public URL)

Description of Conformance to Specification Design

This section outlines how the project conforms to the original specification and design. It includes a summary of the functionality achieved, not achieved and potential improvements that could be made to enhance the system further.

What was achieved

The following is a list of key features and functionalities that were successfully developed and implemented as part of the project for both the core application and the key management server:

- Cryptography: Secure encryption and decryption mechanisms were implemented to protect all patient data.
- Key server: A dedicated server was developed using FastAPI to manage cryptographic keys and token.
- API Token Authentication: Implemented secure token-based access control for encryption key server endpoints.
- Key rotation: Key rotation is manually triggered through the Key Server, followed by an admin-activated Maintenance Mode in Care Net to safely re-encrypt all data. An automated re-encryption also runs weekly, but it does not fully cover edge cases where new data could be added after a rotation. Manual maintenance remains essential to prevent data corruption.
- Session Management: Sessions expire after 30 minutes of inactivity or on browser close, with secure cookies and automatic login redirection via custom middleware.
- User Login, and registration: User names and password were hashed and salted.
- Creating Patient Profile
- Search Patient Profile
- Viewing Patient Profile
- Edit Patient Profile
- Delete Patient Profile
- Create Medical Dashboard
- Search Medical Dashboard
- View Medical Dashboard
- Edit Medical Dashboard
- Create Roster Entry
- Delete Roster Entry

- Create Care Plan
- Search Care Plan
- View Care Plan
- Edit Care Plan

What was not achieved

Overall, the core functionality of the system was successfully achieved. However, the following two non-core features were not implemented, as they were not essential to the main system goals:

- Incident reporting: System for Carers to report incidents during patient visits.
- Alerts: Notifications to admins and carers for events such as new incident reports or task reminders.

These features were not developed due to time constraints. Additionally, while key rotation and re-encryption were implemented, the automated re-encryption process does not fully cover edge cases. Currently manual activation of Maintenance Mode is required immediately after a key rotation. If the application is used before Maintenance Mode is activated, it will retrieve and cache the new key while the old key has already been invalidated, which can lead to data corruption when the scheduled automated re-encryption eventually runs.

If additional time had been available, improvements would have been made to enhance the robustness of the features that were achieved. The automated re-encryption process could be made more reliable and to implement stricter input validation.

Description of Learning

This project provided me with the opportunity to increase my technical and personal knowledge.

Technical Learning

Cryptography

Before this project, I had no prior experience in implementing cryptography. Implementing encryption and decryption using python cryptography library was a significant learning achievement. I learned how to securely encrypt sensitive information, such as patient data, and manage encryption keys.

Django

I had never worked with the Django framework. Throughout the development, I learned how to use Django's framework to build web application, including setting up views, middleware, session management, and secure handling of user authentication and authorisation.

Fast API

During the early stages of the project, my supervisor Paul Barry recommended using FastAPI for developing the key management server. This introduced me to building lightweight API. I learned how to implement token-based authentication, manage secure endpoints.

Bootstrap

For the front end I used bootstrap to style and structure the application. Using bootstrap allowed me to create clean and consistent interface without having to manually write extensive custom CSS.

Personal Learning

Agile Development

This project was developed using agile practices. The development process was divided into three iterations, with feedback received after each one. Weekly meetings with my supervisor allowed me to discuss project progress, address any challenges, and plan upcoming tasks. I gained valuable experience working with the Agile framework. I learned to be flexible through out the project and manage my time and I took the feedback from my supervisor positively and improved on it.

Time Management

Throughout this project, I learned to prioritise my time effectively over my other modules. I worked hard to balance the workload for each module and to allocate my time appropriately to meet deadlines. Completing this project alongside regular coursework helped me improve my skills in organising and prioritising my workload, and thought me the importance of careful planning and time management.

Public speaking

I also had the opportunity to work on my communication skills during weekly meetings with my supervisor and through presentations conducted at the end of each iteration.

Organisation

My organisation skills improved significantly through regular weekly meeting. I learned to manage my time and I became more flexible in achieving my goals and deadlines for this project and other workload.

Review of Project

What went well

- Successfully created a Key Management Server using FastAPI to manage the keys and token and requests.
- Completed the encryption and decryption logic for securing patients data.
- Implemented the core features of the application (patient profile, dashboard, roster, care plan) to a usable level.
- Got feedback from supervisor regularly.

What went wrong/difficulties

- Cryptography was difficult initially due to lack of prior experience.
- Learning and setting up the Django framework was challenging at first.
- Setting up FastAPI.
- Automation of the re-encryption process faced issues and didn’t handle all edge cases.

What is outstanding/missing

- The non core features Incident reporting and alert were not implemented however there absence does not effect the functionality of the project.

- Further improvements needed on existing features, such as stricter input validation and better separation of functionality between Admin and Carer roles.

Starting again

- Spend more time early on learning and understanding Django framework better.
- Plan re-encryption automation more carefully to avoid data corruption.

Advice for future developers

- Be open to feedback from supervisor.
- Research chosen technologies thoroughly before starting development.
- Make important technology decisions early, as it is easier to change frameworks in the early stages.
- Spend more time on the automated process of re-encryption.

Improvements If I had more time

- Finish non-core features.
- Improve the strength of automated and manual re-encryption to handle edge cases better.
- Improve existing features, including implementing stricter input validation.
- Enhance role-specific functionality.
- Make the system more secure.

Technology Review

- I feel the technologies used (Django, FastAPI, Bootstrap, cryptography library, secrets library) were good choices overall.
- I would spend more time using Django to its full capabilities.
- Spending additional time experimenting with Django's advanced features.

Conclusion

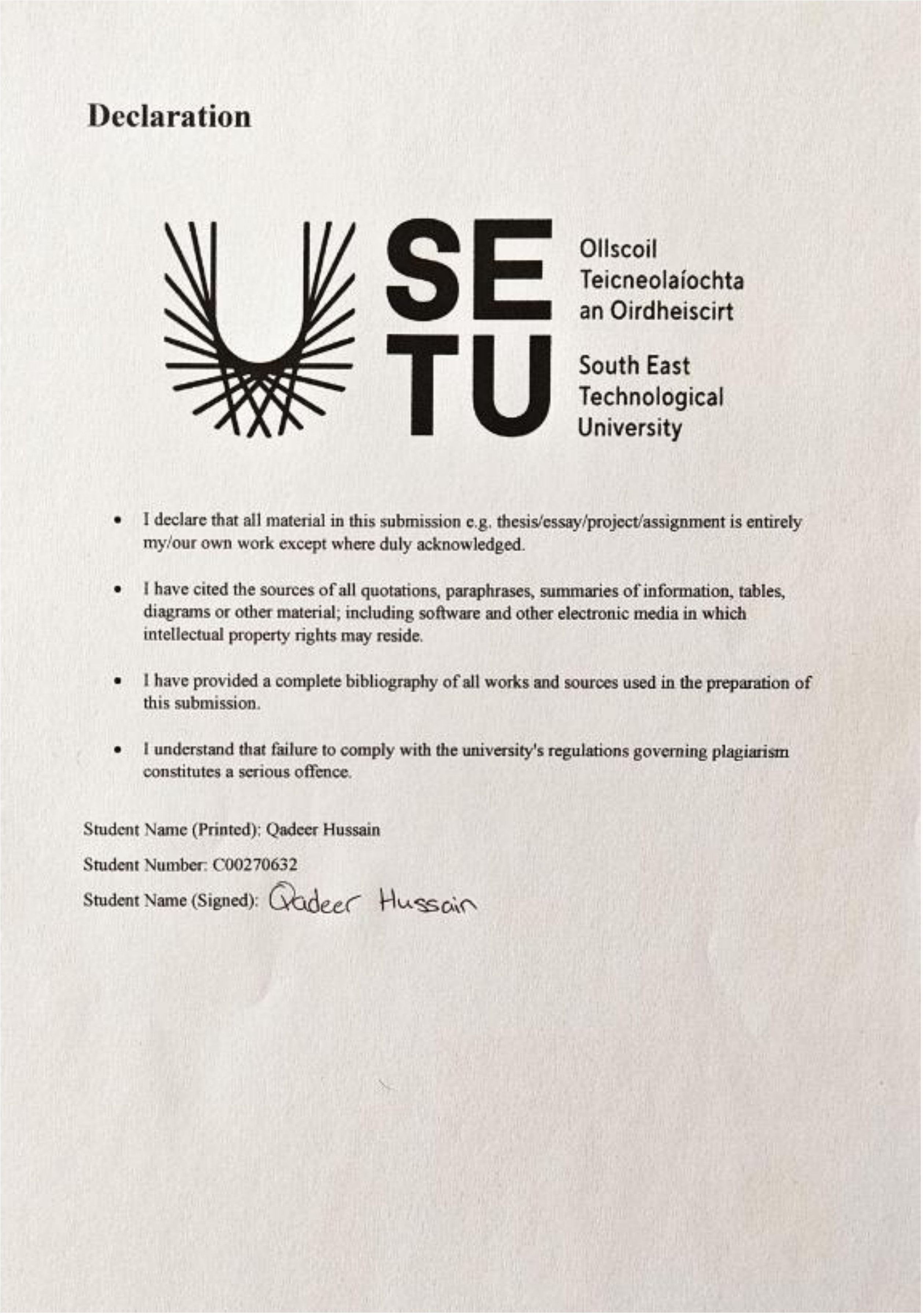
The goal of this project was to develop a secure web application for nursing homes to safely store and manage patient data. Core functionality such as cryptography, patient profile, medical dashboards, roster management and care planning were successfully developed.

In conclusion, I believe this project was a success. It not only demonstrated the technical possibility of building a secure patient management system but also helped me significantly grow my personal skills such as time management, problem solving and organisational skills which will help me in the future.

Acknowledgements

I would like to thank my project supervisor Paul Barry for his continued support throughout the duration of the project. Our weekly meetings were extremely helpful, providing continuous feedback that kept me on track and helped me adhere to the project schedule. Paul was also available outside of our scheduled meetings via email to assist with any questions or issues that arose.

Finally I would like to extend my thanks to all the lectures in SETU who have thought and supported me during my four years at SETU.



Declaration

SE
TU

Ollscoil
Teicneolaíochta
an Oirdheiscirt

South East
Technological
University

- I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

- I have provided a complete bibliography of all works and sources used in the preparation of this submission.

- I understand that failure to comply with the university's regulations governing plagiarism constitutes a serious offence.

Student Name (Printed): Qadeer Hussain

Student Number: C00270632

Student Name (Signed): Qadeer Hussain