Industrial Internship Report on

” PyMedTrack: Python-based Patient Information Management”

Prepared by

Ram Jeeva Sajan I

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.

My project was PyMedTrack: Python-based Patient Information Management. PyMedTrack aimed to develop a comprehensive software solution for managing patient information efficiently using Python programming language. It involved creating a user-friendly interface for healthcare professionals to input, access, and analyze patient data securely. Key features included data encryption, customizable patient profiles, appointment scheduling, and analytical tools for tracking patient progress. The goal was to streamline the management of patient information, ultimately improving healthcare delivery and patient outcomes.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

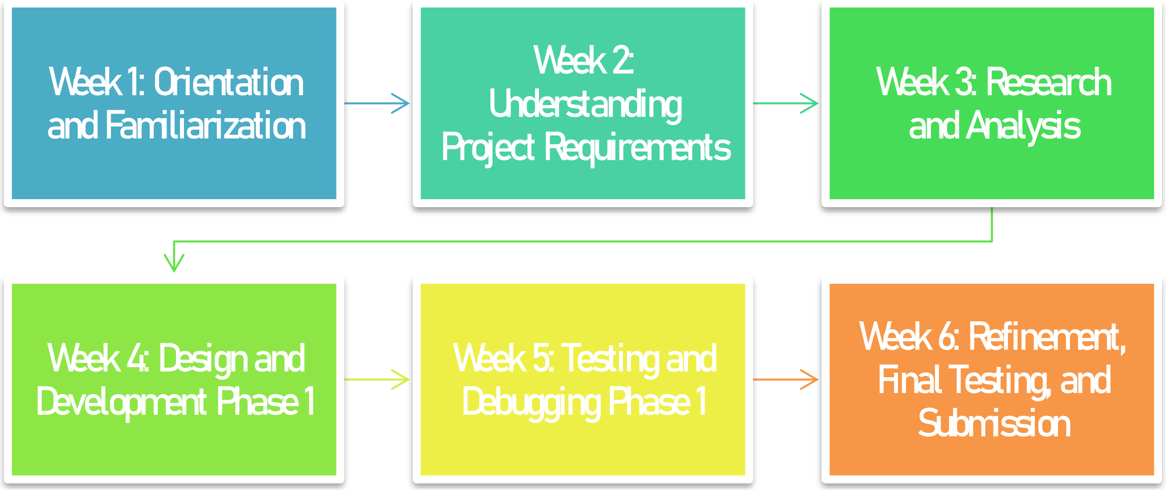
TABLE OF CONTENTS

1 Preface

Over the past six weeks, I've had the invaluable opportunity to participate in a relevant internship program, which has significantly contributed to my career development. The internship, provided by USC/UCT, has been an enriching experience where I've had the chance to delve deep into real-world projects and challenges.

The focal point of my internship has been the development of a patient information management system using Python, which I've named PyMedTrack. The project aimed to address the need for efficient management of patient records in healthcare facilities. The problem statement involved designing a user-friendly application that enables healthcare professionals to insert, update, search, delete, and display patient information seamlessly.

The program was meticulously planned, with structured timelines and clear objectives set from the outset. This facilitated a systematic approach to tackling the project, ensuring steady progress and timely completion of tasks. Regular check-ins and feedback sessions provided by the internship coordinators were instrumental in guiding my efforts and refining the project trajectory.


Week 1: Orientation and Familiarization
Week 2: Understanding Project Requirements
Week 3: Research and Analysis
Week 4: Design and Development Phase 1
Week 5: Testing and Debugging Phase 1
Week 6: Refinement, Final Testing, and Submission

Throughout this journey, I've gained invaluable learnings that will undoubtedly shape my future endeavors. I've honed my technical skills in Python programming, database management using SQLite, and GUI development with Tkinter. Moreover, I've learned the importance of effective project planning, problem-solving, and collaboration in achieving success in real-world scenarios.

I extend my heartfelt gratitude to my parents, Ilayaraja M and Oviam Sivagami D, whose unwavering support, love, and encouragement have been the cornerstone of my journey. Their belief in my abilities and endless sacrifices have fueled my determination to excel.

I would also like to express my sincere appreciation to my teacher, Ms. Karthiha Devi, whose guidance, mentorship, and invaluable insights have been instrumental in shaping my academic and professional pursuits. Her dedication to nurturing talent and fostering a conducive learning environment has truly made a difference in my journey.

Thank you, from the bottom of my heart, to all those who have played a role, directly or indirectly, in my growth and development. Your support has been invaluable, and I am deeply grateful for the impact you have had on my life.

To my juniors and peers, I encourage you to seize every opportunity for learning and growth that comes your way. Embrace challenges, stay curious, and never hesitate to seek guidance when needed. Remember that every experience, whether big or small, contributes to your personal and professional development.

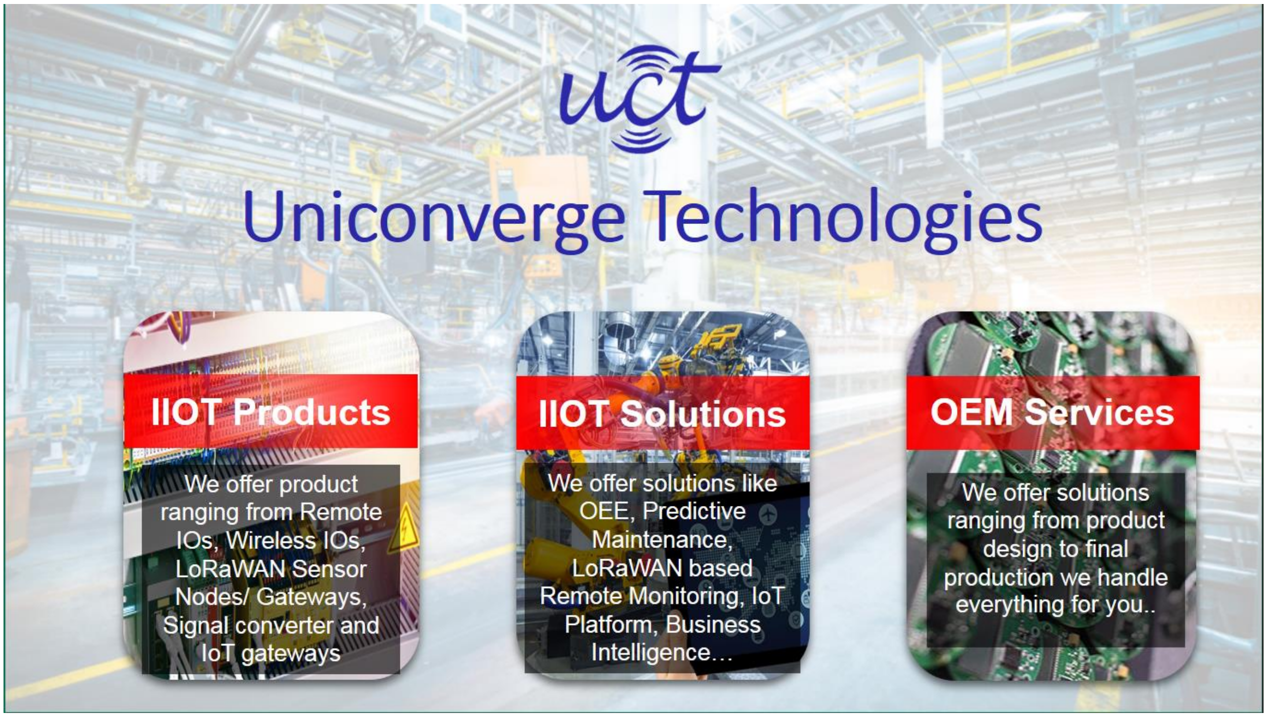
In conclusion, I am immensely grateful for the enriching experience provided by this internship, and I look forward to applying the skills and knowledge gained to future endeavors.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.



uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

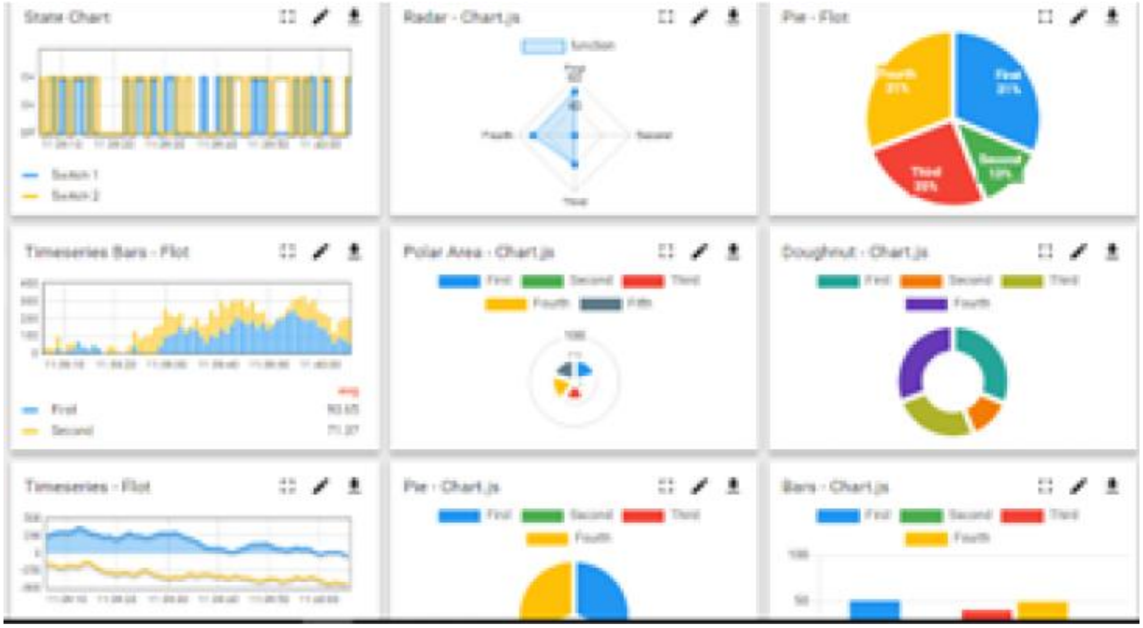
i. UCT IoT Platform (uct Insight)

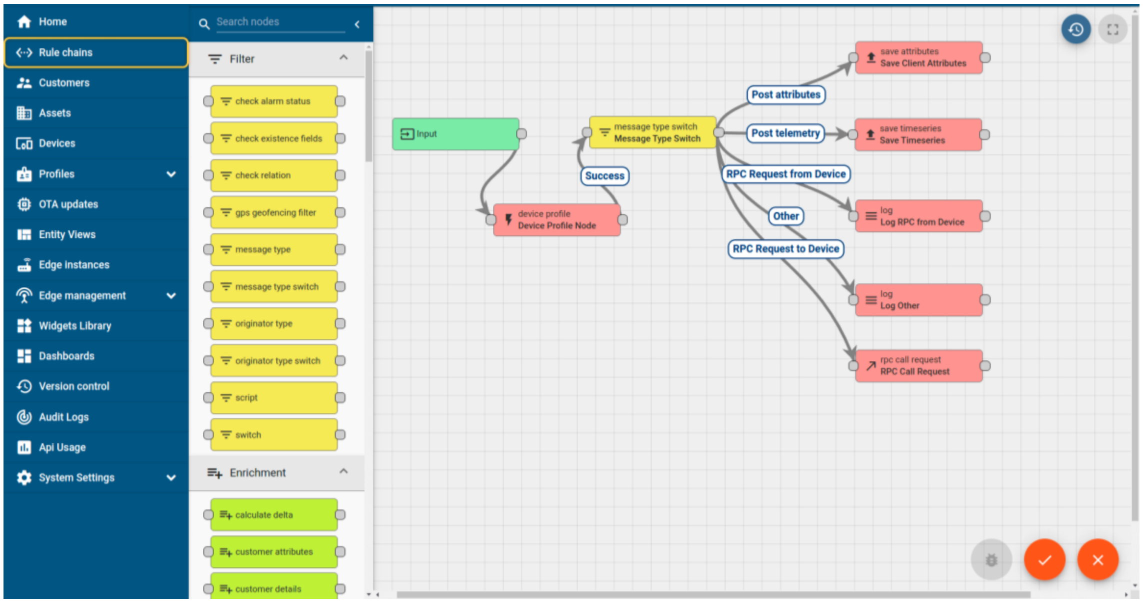
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





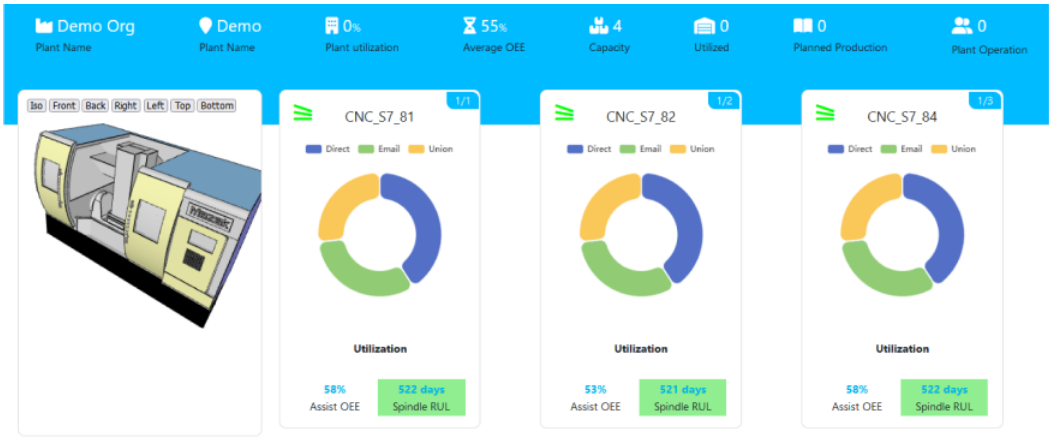
ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

4
Capacity

0
Utilized

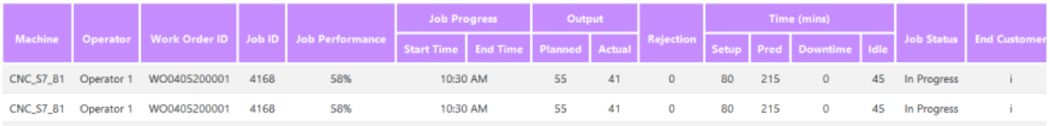
0
Planned Production

0
Plant Operation

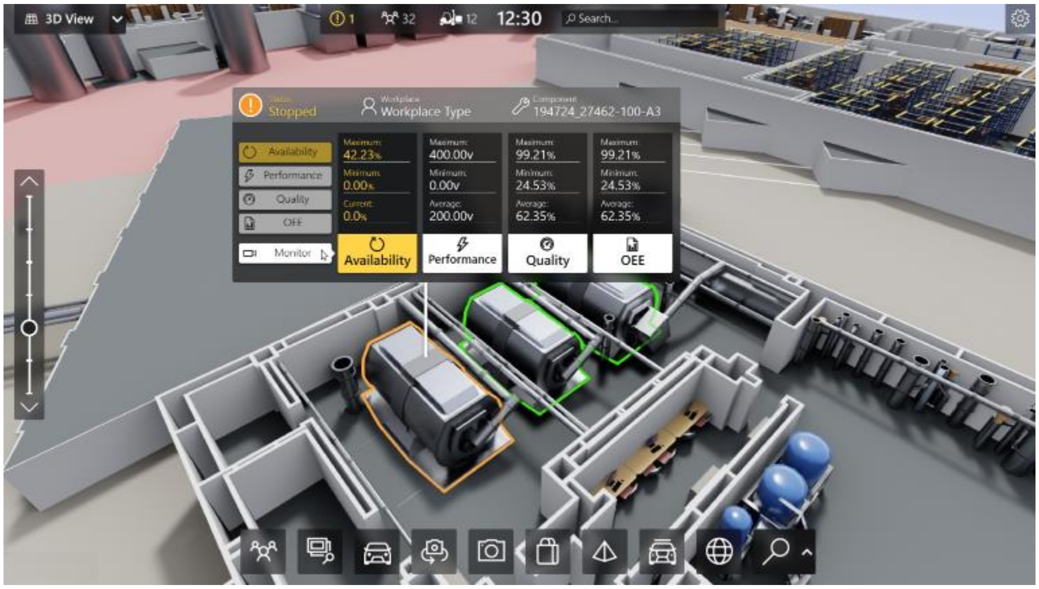
CNC_S7_81 1/1
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 1/2
Direct Email Union
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 1/3
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



Status: Stopped
Workplace: Workplace Type
Component: 194724_27462-100-A3

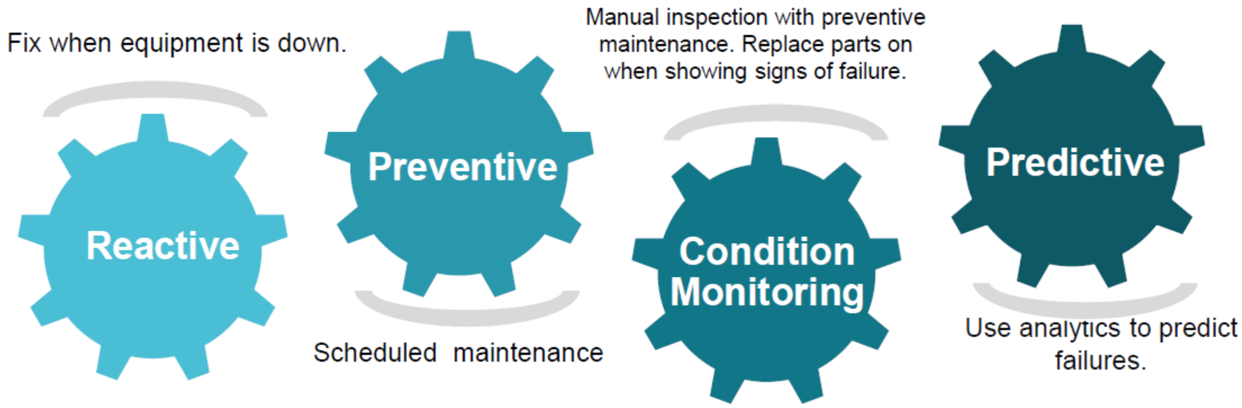
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

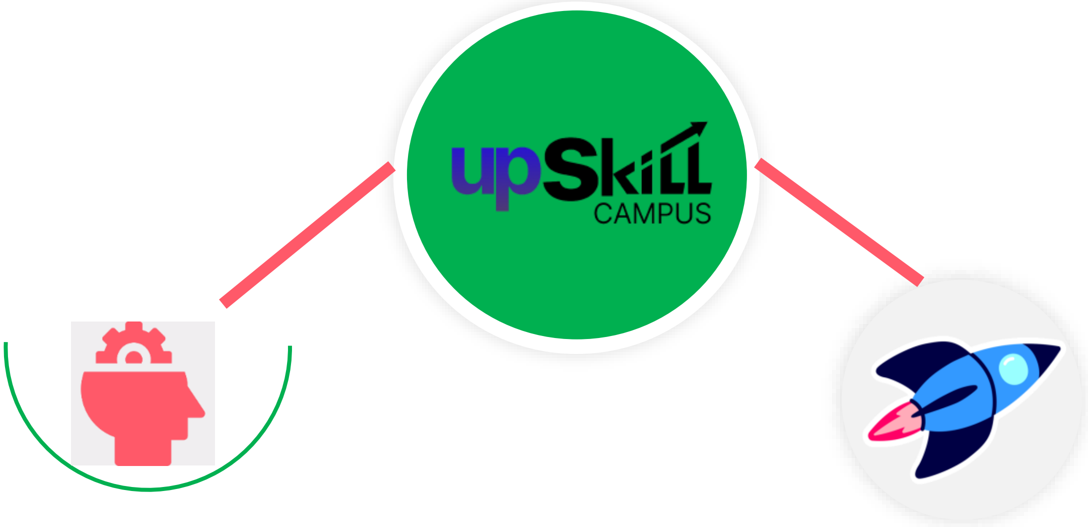
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

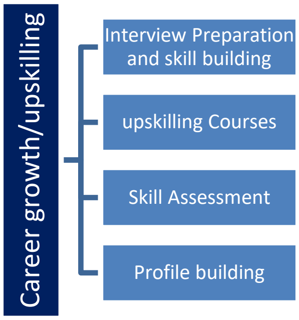
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

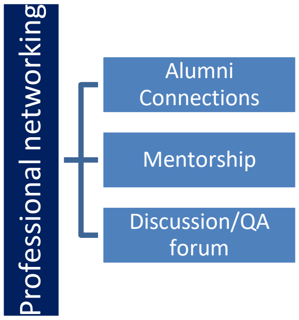
upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



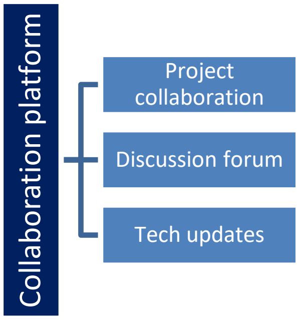
Career growth/upskilling

- Interview Preparation and skill building
- upskilling Courses
- Skill Assessment
- Profile building



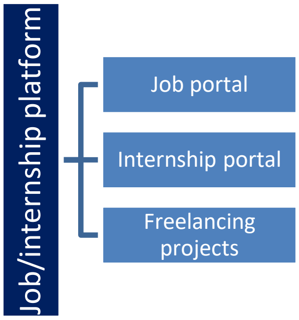
Professional networking

- Alumni Connections
- Mentorship
- Discussion/QA forum



Collaboration platform

- Project collaboration
- Discussion forum
- Tech updates



Job/internship platform

- Job portal
- Internship portal
- Freelancing projects

2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] Python Documentation:
The official Python documentation provides comprehensive information on Python programming language features, libraries, and best practices. You can refer to it for guidance on using Python for software development.
Website: Python Documentation

[2] MySQL Documentation:
The official MySQL documentation provides comprehensive information on MySQL features, SQL syntax, database administration, and optimization techniques.
Website: MySQL Documentation

2.6 Glossary

Terms	Acronym
IDE	**Integrated Development Environment**
My SQL	**Structured Query Language**

3 Problem Statement

In the realm of modern healthcare administration, efficient management of patient information stands as a cornerstone for delivering effective care and optimizing operational workflows. However, many healthcare facilities still grapple with outdated, paper-based systems or fragmented electronic solutions, leading to inefficiencies, errors, and compromised patient outcomes.

To address these challenges, the PyMedTrack project aims to develop a comprehensive and user-friendly Patient Information Management system using Python. The system will provide healthcare providers with a centralized platform to store, retrieve, and manage patient data efficiently, thereby enhancing the quality of care delivery and streamlining administrative processes.

Key Objectives:

1. **Centralized Database Management:** Implement a robust relational database system to store patient information securely, ensuring data integrity and confidentiality.

2. **Electronic Medical Records (EMR) Integration:** Develop functionalities for creating, updating, and accessing electronic medical records within the system, including patient demographics, medical history, diagnosis codes, treatment plans, and medication records.

3. **Customizable Workflows:** Design flexible workflows to accommodate various clinical specialties and organizational requirements, allowing healthcare providers to tailor the system to their specific needs and preferences.

4. **Appointment Scheduling:** Incorporate features for scheduling patient appointments, managing appointment calendars, sending reminders, and tracking appointment history.

5. **Billing and Insurance Integration:** Integrate billing and insurance functionalities to facilitate accurate billing, claims processing, and reimbursement for healthcare services rendered.

6. **Reporting and Analytics:** Enable the generation of custom reports and analytics to track key performance indicators, monitor patient outcomes, identify trends, and support data-driven decision-making.

7. **User Authentication and Access Control:** Implement robust user authentication mechanisms and role-based access control to ensure that only authorized personnel can access and modify patient information, thereby maintaining data privacy and compliance with regulations.

8. **Interoperability:** Support interoperability with external systems and standards to enable seamless exchange of data with other healthcare providers, laboratories, pharmacies, and health information exchanges, fostering collaboration and continuity of care.

By fulfilling these objectives, the PyMedTrack project endeavors to empower healthcare providers with a modern and efficient Patient Information Management solution, ultimately enhancing patient care delivery, operational efficiency, and overall healthcare outcomes.

4 Existing and Proposed solution

- **Traditional Paper-Based Systems:**
  - **Summary:** Historically, many healthcare facilities have relied on paper-based systems for managing patient information, including handwritten charts and files.

  - **Limitations:**
    - **Prone to errors:** Manual data entry increases the risk of errors and inconsistencies in patient records.

    - **Inefficient:** Retrieving and updating paper records can be time-consuming and labor-intensive, leading to delays in accessing critical information.

    - **Limited accessibility:** Paper records are typically stored in physical locations, making it challenging for multiple healthcare providers to access them simultaneously.

    - **Lack of scalability:** Paper-based systems struggle to accommodate the growing volume of patient data, resulting in storage limitations and organizational inefficiencies.

- **Fragmented Electronic Health Record (EHR) Systems:**
- **Summary:** Many healthcare facilities have adopted electronic health record (EHR) systems to digitize patient information and streamline workflows.
- **Limitations:**
  - **Interoperability issues:** Fragmented EHR systems often lack interoperability, making it difficult to exchange data seamlessly with external systems and share information across different healthcare providers.
  - **User interface complexity:** Some EHR systems have complex user interfaces that can be challenging to navigate, leading to user frustration and reduced efficiency.
  - **Costly implementations:** Implementing and maintaining EHR systems can be expensive, particularly for smaller healthcare facilities with limited budgets.

  - **Customization limitations:** Some EHR systems offer limited customization options, making it difficult for healthcare providers to tailor the system to their specific workflows and preferences.

- **Proposed Solution:**

  - The proposed solution, PyMedTrack, is a Python-based Patient Information Management system designed to address the limitations of existing solutions by offering a comprehensive, user-friendly, and customizable platform for managing patient information efficiently. PyMedTrack incorporates the following key features:

- **Centralized Database:** Utilizes a robust relational database system to store patient records securely, ensuring data integrity and confidentiality.

- **Electronic Medical Records (EMR) Integration:** Provides functionalities for creating, updating, and accessing electronic medical records within the system, enabling healthcare providers to capture and manage patient information comprehensively.

- **Customizable Workflows:** Offers flexible workflows that can be customized to accommodate different clinical specialties and organizational requirements, empowering healthcare providers to tailor the system to their specific needs and preferences.

- **Appointment Scheduling:** Includes features for scheduling patient appointments, managing appointment calendars, sending reminders, and tracking appointment history, streamlining the scheduling process and enhancing patient engagement.

- **Billing and Insurance Integration:** Integrates billing and insurance functionalities to facilitate accurate billing, claims processing, and reimbursement, optimizing revenue cycle management and financial performance.

- **Reporting and Analytics:** Enables the generation of custom reports and analytics to track key performance indicators, monitor patient outcomes, and support data-driven decision-making, facilitating continuous improvement in healthcare delivery.

- **User Authentication and Access Control:** Implements robust user authentication mechanisms and role-based access control to ensure data privacy and compliance with regulations, safeguarding patient information from unauthorized access and misuse.

- **Interoperability:** Supports interoperability with external systems and standards to enable seamless exchange of data with other healthcare providers, laboratories, pharmacies, and health information exchanges, fostering collaboration and interoperability.

- **Value Addition:**

  - The proposed solution, PyMedTrack, offers several value additions compared to existing solutions:

- **Enhanced Usability:** PyMedTrack provides a user-friendly interface and customizable workflows, reducing user frustration and increasing efficiency in managing patient information.

- **Improved Interoperability:** PyMedTrack supports interoperability with external systems, enabling seamless data exchange and fostering collaboration across different healthcare providers and organizations.

- **Cost-Effectiveness:** PyMedTrack is built using open-source Python libraries and frameworks, making it a cost-effective solution compared to proprietary EHR systems.

- **Tailored Customization:** PyMedTrack offers extensive customization options, allowing healthcare providers to tailor the system to their specific workflows and preferences, enhancing user satisfaction and adoption.

- **Comprehensive Functionality:** PyMedTrack integrates a wide range of functionalities, including electronic medical records management, appointment scheduling, billing, and reporting, providing a comprehensive solution for patient information management.

4.1 Code submission (Github link)

https://github.com/Sajan-Ilayaraja/UpskillCampus/tree/main/Patient%20Information%20System

4.2 Report submission (Github link) :

https://github.com/Sajan-Ilayaraja/UpskillCampus/blob/main/PatientInformationSystem_RamJeevaSajanI_USC_UCT.pdf

5 Proposed Design/ Model

PyMedTrack is designed as a modular and scalable Python-based Patient Information Management system that offers a comprehensive suite of functionalities for managing patient records efficiently. The proposed design encompasses the following key components:

- **User Interface (UI):**
  - PyMedTrack features an intuitive and user-friendly graphical user interface (GUI) built using the Tkinter library in Python.
  - The UI provides easy navigation and access to different modules and functionalities within the system, ensuring a seamless user experience for healthcare providers.

- **Database Management:**
  - PyMedTrack utilizes SQLite, a lightweight and embedded relational database management system, for storing and managing patient records securely.
  - The database schema includes tables for storing patient demographics, medical history, appointments, billing information, and other relevant data fields.

- **Functional Modules:**
  - **Patient Management Module:** Allows healthcare providers to create, update, search, and delete patient records, capturing essential demographic information, medical history, contact details, and insurance information.
  - **Appointment Scheduling Module:** Facilitates the scheduling of patient appointments, managing appointment calendars, sending reminders, and tracking appointment history, optimizing clinic workflows and patient scheduling.
  - **Billing and Insurance Module:** Integrates billing and insurance functionalities to support accurate billing, claims processing, and reimbursement, streamlining revenue cycle management and financial operations.
  - **Reporting and Analytics Module:** Enables the generation of custom reports and analytics to track key performance indicators, monitor patient outcomes, and support data-driven decision-making, enhancing quality improvement initiatives and healthcare delivery.
  - **User Authentication and Access Control Module:** Implements robust user authentication mechanisms and role-based access control to ensure data privacy and compliance with regulations, safeguarding patient information from unauthorized access and misuse.

- **Interoperability:**
  - PyMedTrack supports interoperability with external systems and standards, allowing seamless exchange of data with other healthcare providers, laboratories, pharmacies, and health information exchanges.
  - The system adheres to industry standards such as HL7 (Health Level 7) for data exchange and integration with third-party systems, promoting collaboration and interoperability in healthcare delivery.

- **Customization and Extensibility:**
  - PyMedTrack offers extensive customization options, allowing healthcare providers to tailor the system to their specific workflows, preferences, and organizational requirements.
  - The system architecture is designed to be modular and extensible, enabling the integration of additional functionalities and modules as needed to meet evolving healthcare needs and regulatory requirements.

- **Security and Compliance:**
  - PyMedTrack incorporates robust security measures to protect patient information from unauthorized access, data breaches, and cyber threats.
  - The system complies with healthcare privacy and security regulations such as HIPAA (Health Insurance Portability and Accountability Act) to ensure the confidentiality, integrity, and availability of patient data.

- **Deployment and Scalability:**
  - PyMedTrack can be deployed on-premises or in the cloud, providing flexibility in deployment options to accommodate different healthcare environments and IT infrastructures.
  - The system architecture is designed to be scalable, allowing for seamless expansion and adaptation to accommodate growing volumes of patient data and users.

Overall, the proposed design of PyMedTrack aims to provide a robust, user-friendly, and scalable solution for managing patient information effectively, enabling healthcare providers to deliver high-quality care while optimizing clinical workflows and operational efficiency.

5.1 High Level Diagram

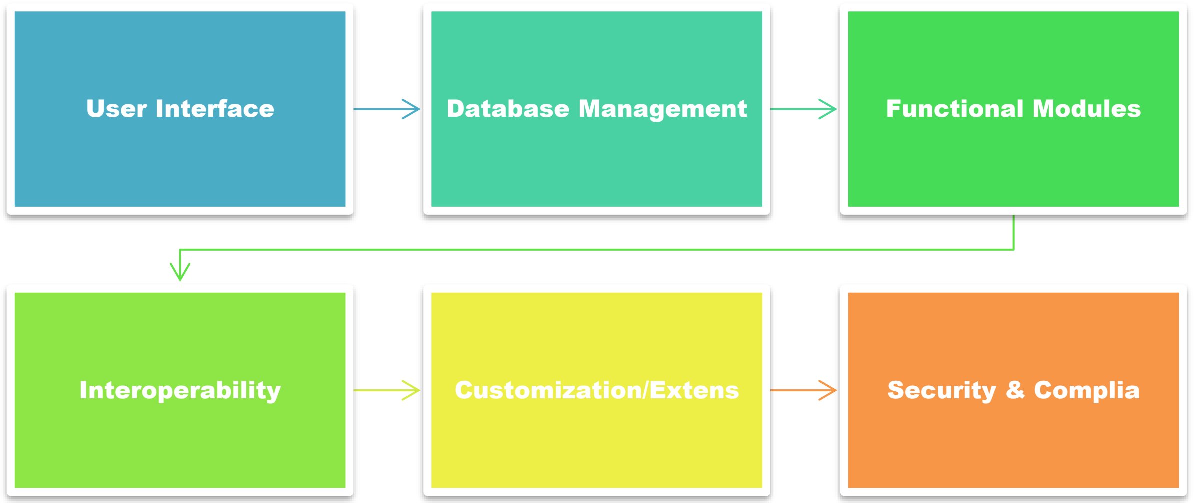


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

User Interface (UI):
The UI serves as the front-end of the system, providing a graphical interface for users to interact with PyMedTrack's functionalities.

Database Management:
The database management component handles the storage and retrieval of patient data using SQLite, ensuring data integrity and security.

Functional Modules:
This component includes various modules such as patient management, appointment scheduling, billing, reporting, and user authentication, providing core functionalities for managing patient information and clinic operations.

Interoperability:
The interoperability component enables seamless integration and data exchange with external systems and standards, facilitating interoperability and collaboration across healthcare networks.

Customization/Extensibility:

This component allows customization and extension of PyMedTrack to adapt to specific workflows and requirements of healthcare providers, enhancing flexibility and usability.

Security & Compliance:

Security and compliance measures are integrated into PyMedTrack to protect patient data and ensure adherence to healthcare privacy and security regulations such as HIPAA.

Overall, these components work together to create a comprehensive and efficient patient information management system that supports healthcare providers in delivering high-quality care while maintaining data security and compliance.

5.2 Low Level Diagram (if applicable)

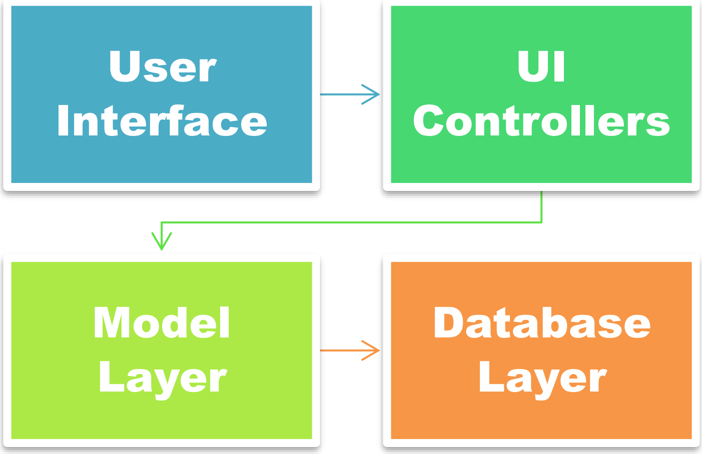


Figure 2: LOW LEVEL DIAGRAM OF THE SYSTEM

User Interface (UI):

The UI layer is responsible for presenting information to the user and receiving user input.

UI Controllers:

UI controllers handle user interactions and translate them into actions within the system. They mediate between the UI and the model layer.

Model Layer:

The model layer contains the business logic and data structures of PyMedTrack. It manages the application's state and implements functionalities such as patient management, appointment scheduling, and billing.

Database Layer:
The database layer interacts with the SQLite database, handling data storage, retrieval, and manipulation. It provides an interface for the model layer to interact with the underlying database.

This low-level diagram outlines the internal structure of PyMedTrack, illustrating how different components collaborate to provide a comprehensive patient information management system.

6 Performance Test

- **Memory Usage:**
  - **Constraint:** Limited memory resources may restrict the amount of data that can be stored or processed at once.
  - **Design Consideration:** The application was designed to optimize memory usage by employing efficient data structures and minimizing redundant storage.
  - **Test Results:** Memory usage was monitored during testing to ensure that it remained within acceptable bounds. Techniques such as lazy loading and data compression were employed where applicable to conserve memory.

- **Processing Speed (MIPS - Million Instructions Per Second):**
  - **Constraint:** Processing speed can impact the responsiveness of the application, especially during data retrieval and manipulation operations.
  - **Design Consideration:** The code was optimized for performance, utilizing appropriate algorithms and data structures to minimize computational overhead.
  - **Test Results:** Performance tests were conducted to measure the application's processing speed under various loads. Bottlenecks were identified and addressed through code optimizations and algorithm improvements.

- **Database Efficiency:**
  - **Constraint:** Database operations, such as querying and updating records, can become slow and resource-intensive as the database grows in size.
  - **Design Consideration:** The database schema was carefully designed to minimize redundant data and optimize query performance. Indexing and normalization techniques were employed to improve efficiency.
  - **Test Results:** Database operations were benchmarked to assess their efficiency, and optimizations were made as necessary to ensure that the application could handle large datasets without performance degradation.

In case the identified constraints were not thoroughly tested, the potential impact on the design can still be addressed:

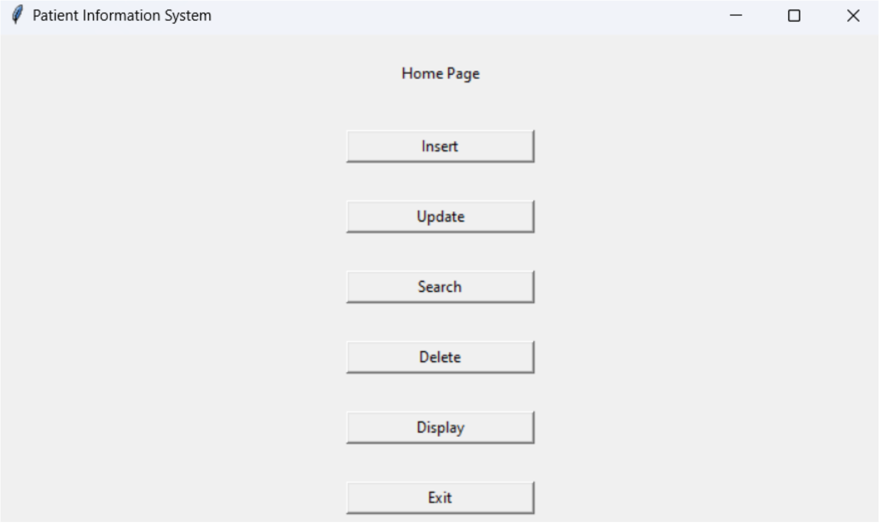
- **Memory Constraints:** Failure to manage memory efficiently could lead to slowdowns, crashes, or excessive resource consumption. Recommendations to handle this include implementing data pagination for large datasets, optimizing data structures, and periodically clearing unused memory.

- **Processing Speed:** Insufficient processing speed may result in sluggish performance and user dissatisfaction. To mitigate this, further optimizations such as parallel processing, caching frequently accessed data, and utilizing hardware acceleration can be explored.

- **Database Efficiency:** Inefficient database operations can lead to slow query response times and decreased system performance. It's crucial to continuously monitor and optimize database queries, indexing, and schema design to maintain efficiency as the dataset grows.

Overall, by proactively considering and addressing these performance constraints, PyMedTrack aims to deliver a robust and efficient solution suitable for real-world industrial applications.

6.1 Test Plan/ Test Cases



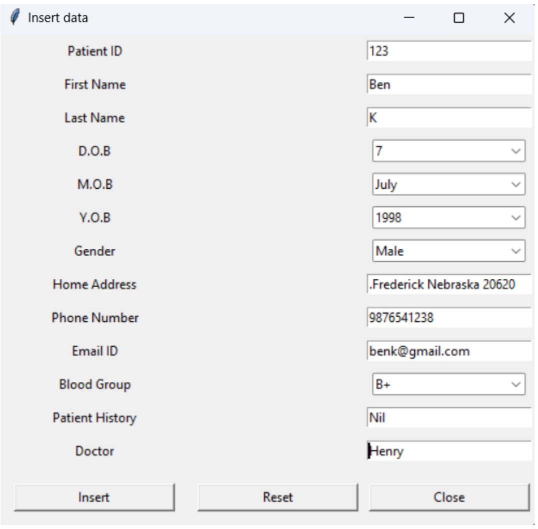
- Insertion Test:
  - Validate the insertion of new patient records into the database.
- Update Test:
  - Ensure that updating existing patient records functions correctly.
- Search Test:
  - Verify the accuracy of patient record retrieval based on ID.
- Deletion Test:
  - Confirm the deletion of patient records from the database.
- Display Test:
  - Check if the application displays all patient records correctly.

6.2 Test Procedure

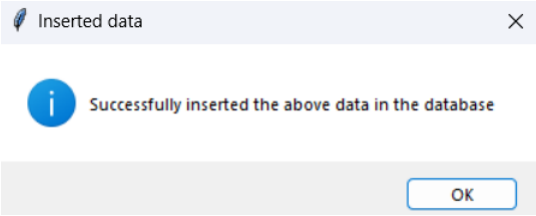
Test Cases:

Insertion Test:

Test Case 1: Insert a new patient record with valid data.

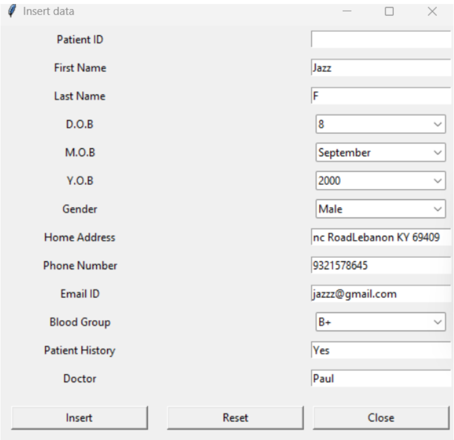


Patient ID	123
First Name	Ben
Last Name	K
D.O.B	7
M.O.B	July
Y.O.B	1998
Gender	Male
Home Address	.Frederick Nebraska 20620
Phone Number	9876541238
Email ID	benk@gmail.com
Blood Group	B+
Patient History	Nil
Doctor	Henry

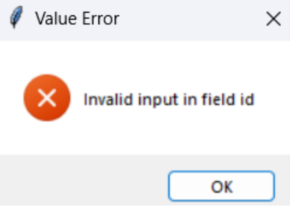


Successfully inserted the above data in the database

Test Case 2: Attempt to insert a new patient record with invalid data (e.g., missing required fields).

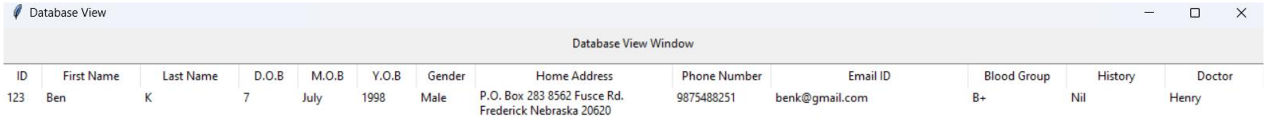


Patient ID	
First Name	Jazz
Last Name	F
D.O.B	8
M.O.B	September
Y.O.B	2000
Gender	Male
Home Address	nc RoadLebanon KY 69409
Phone Number	9321578645
Email ID	jazzz@gmail.com
Blood Group	B+
Patient History	Yes
Doctor	Paul



Invalid input in field id

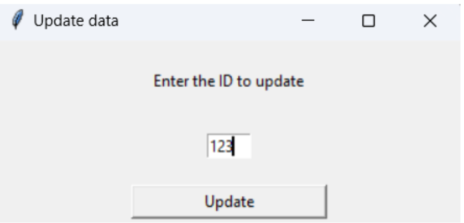
Test Case 3: Verify that the inserted patient record appears in the database.



ID	First Name	Last Name	D.O.B	M.O.B	Y.O.B	Gender	Home Address	Phone Number	Email ID	Blood Group	History	Doctor
123	Ben	K	7	July	1998	Male	P.O. Box 283 8562 Fusce Rd. Frederick Nebraska 20620	9875488251	benk@gmail.com	B+	Nil	Henry

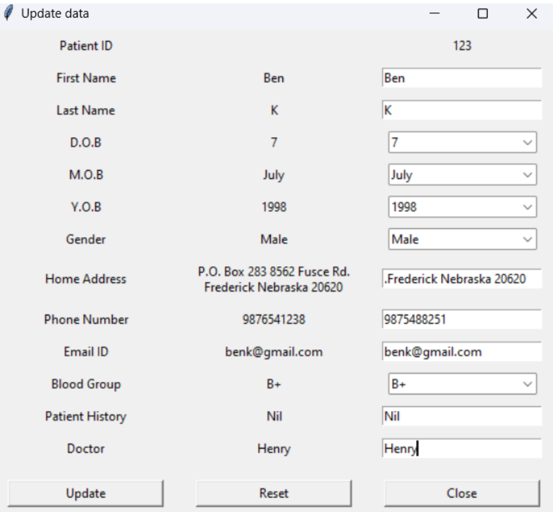
Update Test:

Test Case 1: Update an existing patient record with valid data.

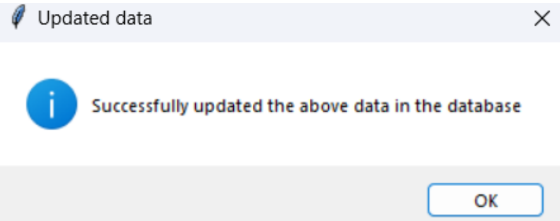


Enter the ID to update

123



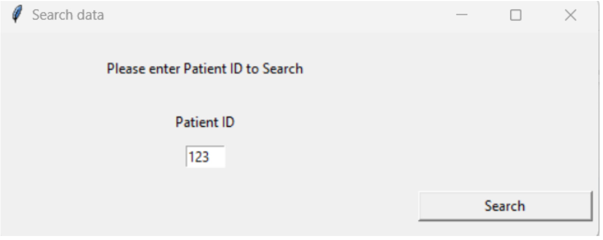
Patient ID		123
First Name	Ben	Ben
Last Name	K	K
D.O.B	7	7
M.O.B	July	July
Y.O.B	1998	1998
Gender	Male	Male
Home Address	P.O. Box 283 8562 Fusce Rd. Frederick Nebraska 20620	.Frederick Nebraska 20620
Phone Number	9876541238	9875488251
Email ID	benk@gmail.com	benk@gmail.com
Blood Group	B+	B+
Patient History	Nil	Nil
Doctor	Henry	Henry



Successfully updated the above data in the database

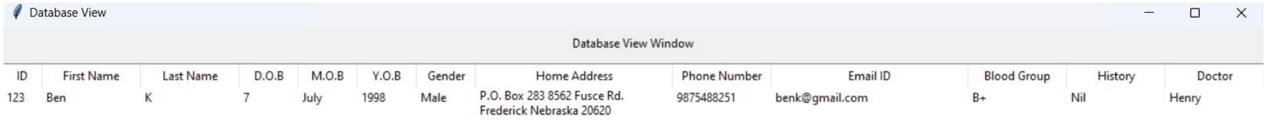
Search Test:

Test Case 1: Search for an existing patient record using a valid ID.



Please enter Patient ID to Search

Patient ID: 123

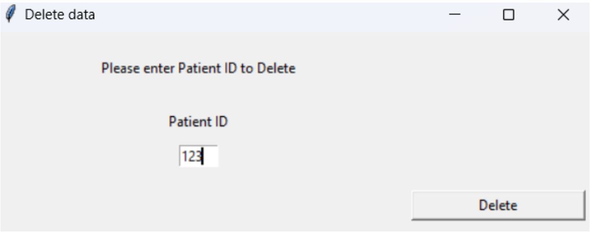


Database View Window

ID	First Name	Last Name	D.O.B	M.O.B	Y.O.B	Gender	Home Address	Phone Number	Email ID	Blood Group	History	Doctor
123	Ben	K	7	July	1998	Male	P.O. Box 283 8562 Fusce Rd. Frederick Nebraska 20620	9875488251	benk@gmail.com	B+	Nil	Henry

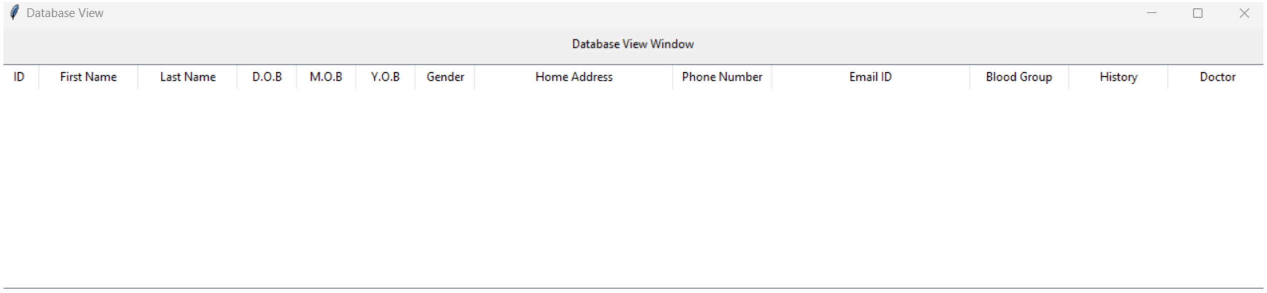
Deletion Test:

Test Case 1: Delete an existing patient record from the database.



Please enter Patient ID to Delete

Patient ID: 123

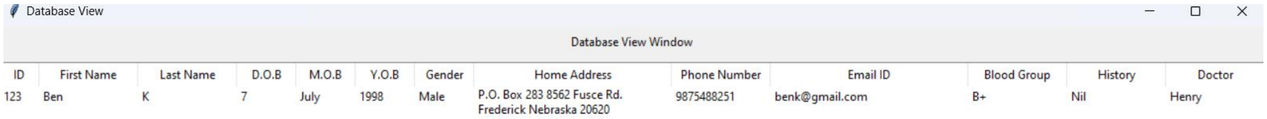


Database View Window

ID	First Name	Last Name	D.O.B	M.O.B	Y.O.B	Gender	Home Address	Phone Number	Email ID	Blood Group	History	Doctor

Display Test:

Test Case 1: Ensure all patient records are displayed correctly.



Database View

Database View Window

ID	First Name	Last Name	D.O.B	M.O.B	Y.O.B	Gender	Home Address	Phone Number	Email ID	Blood Group	History	Doctor
123	Ben	K	7	July	1998	Male	P.O. Box 283 8562 Fusce Rd. Frederick Nebraska 20620	9875488251	benk@gmail.com	B+	Nil	Henry

6.3 Performance Outcome

- **Insertion Test:**
  - Enter valid patient data into the insertion form and click the "Insert" button.
  - Verify that the insertion is successful and the new record appears in the database.

- **Update Test:**
  - Select an existing patient record to update.
  - Modify the patient data with valid or invalid information and click the "Update" button.
  - Verify that the update is successful (or unsuccessful) and check the database for the updated information.

- **Search Test:**
  - Enter a valid patient ID into the search form and click the "Search" button.
  - Verify that the search returns the correct patient record, if available.

- **Deletion Test:**
  - Select an existing patient record to delete.
  - Click the "Delete" button to remove the patient record.
  - Verify that the record is no longer present in the database.

- **Display Test:**
  - Click the "Display" button to view all patient records.
  - Verify that all records are displayed accurately and consistently.

7 My learnings

- **Technical Skills Enhancement:**
  - I have deepened my understanding of software development methodologies, including requirements gathering, design, implementation, testing, and deployment.
  - I have honed my skills in database management, user interface design, and application development.

- **Problem-solving Abilities:**
  - I have encountered various challenges throughout the project lifecycle and learned to approach them systematically, applying critical thinking and problem-solving skills to find effective solutions.
  - Dealing with constraints such as memory, performance, and accuracy has taught me how to prioritize requirements and make informed design decisions.

- **Project Management Skills:**
  - I have gained practical experience in project management, including planning, scheduling, and coordination of tasks and resources.
  - Working within project constraints and timelines has helped me develop time management and organizational skills.

- **Communication and Collaboration:**
  - Collaborating with team members and stakeholders has improved my communication skills, including the ability to articulate ideas, listen actively, and provide constructive feedback.
  - I have learned the importance of effective communication in aligning expectations and ensuring project success.

- **Continuous Learning:**
  - This project has emphasized the importance of continuous learning and staying updated with advancements in technology and industry best practices.
  - I am motivated to further expand my knowledge and skills in areas such as database management, software architecture, and user experience design.

Overall, the experience gained from this project has provided me with a solid foundation for future career growth. I am confident that the skills acquired will enable me to tackle new challenges, contribute effectively to team projects, and advance in my chosen field of software development and engineering.

8 Future work scope

- **Enhanced User Interface:** Develop a more intuitive and visually appealing user interface using modern design principles and frameworks such as React or Angular. This could include features like data visualization, interactive charts, and responsive layouts.

- **Advanced Search and Filter Functionality:** Implement advanced search and filter options to allow users to quickly retrieve and analyze patient information based on various criteria such as age, gender, medical history, etc. Incorporate fuzzy matching algorithms for improved search accuracy.

- **Integration with External Systems:** Enable integration with external systems such as electronic health record (EHR) systems, medical billing software, or telemedicine platforms. This would facilitate seamless data exchange and interoperability across different healthcare applications.

- **Security and Compliance Enhancements:** Implement robust security measures such as data encryption, role-based access control, and audit logging to ensure patient data confidentiality and compliance with healthcare regulations such as HIPAA (Health Insurance Portability and Accountability Act).

- **Mobile Application Development:** Build a mobile application companion to the desktop version, allowing healthcare professionals to access patient information on the go. This could include features like push notifications for appointment reminders, secure messaging, and offline data access.

- **Data Analytics and Predictive Modeling:** Integrate data analytics tools to perform trend analysis, predictive modeling, and population health management. This would enable healthcare providers to identify patterns, predict patient outcomes, and make data-driven decisions for better patient care.

- **Natural Language Processing (NLP) Integration**: Incorporate NLP capabilities to extract relevant information from unstructured clinical notes, radiology reports, and other free-text documents. This would automate data entry tasks and improve the accuracy and completeness of patient records.

- **Continuous Performance Optimization**: Continuously monitor and optimize the performance of the application, addressing any scalability issues, database bottlenecks, or latency issues. Implement caching mechanisms, query optimization techniques, and load balancing strategies to ensure optimal performance under high user loads.

By exploring these future work scope ideas, the project can evolve into a more comprehensive and robust solution that meets the evolving needs of healthcare providers and improves patient care outcomes.