A Final Project Report On

LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN

Submitted in the Partial Fulfillment of the
Requirements for the Degree of Bachelor of Computer Engineering
Awarded by Pokhara University

Submitted By:

Ashish Pandey (21075560)
Saru Tiwari (21075581)
Sushil Giri (21075591)
Tirtha Singh Khadka (21075593)

Supervisor:

Er. Sushant Paudel
Assistant Professor

School of Engineering
Faculty of Science and Technology
POKHARA UNIVERSITY

February, 2026

A Final Project Report On

LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN

Submitted in the Partial Fulfillment of the
Requirements for the Degree of Bachelor of Computer Engineering
Awarded by Pokhara University

Submitted By:

Ashish Pandey (21075560)
Saru Tiwari (21075581)
Sushil Giri (21075591)
Tirtha Singh Khadka (21075593)

Supervisor:

Er. Sushant Paudel
Assistant Professor

School of Engineering
Faculty of Science and Technology
POKHARA UNIVERSITY

February, 2026

COPYRIGHT

The author has granted authorization to the Pokhara University Central Library and Library of School of Engineering to grant unrestricted access to this report for inspection. Furthermore, the author has agreed to the potential for extensive reproduction of this report for educational purposes by either the supervisors or, in their absence, by the program coordinator wherein the project was done. It is understood that proper acknowledgment will be accorded to both the author of the report and the School of Engineering, Pokhara University in any utilization of the report's contents. Unauthorized copying, publication, or any other form of exploitation of this report for financial gain without explicit consent from the School of Engineering, Pokhara University and the author is strictly prohibited. Requests seeking permission to replicate or employ any segment of the report's content should be directed to:

Program Coordinator
School of Engineering
Pokhara University

STUDENT'S DECLARATION

We hereby declare that this project work entitled **LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN** is based on our original work. All concepts, data, code, and any other work from external sources have been properly cited and referenced in accordance with the guidelines provided by School of Engineering, Pokhara University.

We owe all the liabilities relating to the authenticity and originality of this project work and project report.

1. Ashish Pandey (21075560) ……………………………..
2. Saru Tiwari (21075581) ……………………………..
3. Sushil Giri (21075591) ……………………………..
4. Tirtha Singh Khadka (21075593) ……………………………..

Date: ……………………….

SUPERVISOR'S RECOMMENDATION

This is to certify that this project report entitled **LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN** prepared and submitted by below listed team of students in partial fulfilment of the requirements of the degree of Bachelor of Computer Engineering awarded by Pokhara University, has been prepared and completed under my supervision.

I hereby recommend the same for acceptance by School of Engineering, Pokhara University.

1. Ashish Pandey (21075560)
2. Saru Tiwari (21075581)
3. Sushil Giri (21075591)
4. Tirtha Singh Khadka (21075593)

…………………………..

Er. Sushant Paudel

Assistant Professor

School of Engineering

Pokhara University

Date: ……………………….

School of Engineering

POKHARA UNIVERSITY

EXTERNAL EXAMINER'S RECOMMENDATION

The undersigned certified that they have evaluated this project report entitled **LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN** submitted by **Ashish Pandey, Saru Tiwari, Sushil Giri, Tirtha Singh Khadka** and their oral presentation for partial fulfillment of the degree of Bachelor of Computer Engineering and recommended to the School of Engineering, Pokhara University for acceptance of this project work/report.

…………………………….

Er. Sudip Dahal

Assistant Professor

Paschimanchal Campus

Lamachaur, Pokhara-16

Date: ……………………….

School of Engineering

POKHARA UNIVERSITY

LETTER OF APPROVAL

This project report entitled **LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN** submitted by **Ashish Pandey, Saru Tiwari, Sushil Giri, Tirtha Singh Khadka** for partial fulfillment of the degree of Bachelor of Computer Engineering has been accepted by the School of Engineering, Pokhara University upon the recommendations of Supervisor and with the approval by the following examiner.

Er. Sudip Dahal

Assistant Professor

Paschimanchal Campus

Lamachaur, Pokhara-16

Er. Bhesh Bahadur Thapa

Assistant Professor

(Coordinator)

Bachelor of Computer Engineering

Bachelor of Software Engineering

School of Engineering

Pokhara University

Date: ……………………….

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who supported us in completing our project entitled **LUNGS DISEASE DETECTION IN CHEST X-RAY IMAGES USING CNN**.

Firstly, we would like to extend our heartfelt thanks to our respected supervisor **Asst. Prof. Er. Sushant Paudel** for his continuous guidance, encouragement, valuable suggestions, and constructive feedback throughout the project period.

We are also grateful to the School of Engineering, Pokhara University for providing us with the opportunity and necessary resources to carry out this project as a part of our academic curriculum.

We would also like to thank all faculty teachers and our friends who directly or indirectly helped and suggested us during the project development and report preparation. Finally, we express our deepest gratitude to our families for this constant motivation, patience and support throughout our academic journey.

Ashish Pandey (21075560)

Saru Tiwari (21075581)

Sushil Giri (21075591)

Tirtha Singh Khadka (21075593)

ABSTRACT

A lungs disease detection in chest X-ray images using CNN introduces a mobile based system designed to detect lungs diseases specifically pneumonia and tuberculosis through the analysis of X-ray images. By leveraging deep learning techniques and CNN, the system can automatically identify pathological patterns and localize the affected regions in lungs radiographs. The application integrates OCR for efficient patient data handling and features a user-friendly interface optimized for mobile devices, making it especially valuable in remote or underserved areas where access to radiological expertise is limited. The backend architecture supports real-time inference and scalable deployment, enabling rapid diagnostics and clinical decision support. This report outlines the system's architecture, model training process, dataset preparation, and performance evaluation, demonstrating the potential of AI-powered mobile health tools to enhance early detection, reduce diagnostic delays, and promote equitable access to healthcare.

Key Words: *OCR, Deep Learning, pneumonia, TB, image detection, CNN*

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

2FA	Two-Factor Authentication
AI	Artificial Intelligence
API	Application Programming Interface
CNN	Convolutional Neural Network
IoU	Intersection over Union
LLM	Large Language Model
ML	Machine Learning
MySQL	My Structured Query Language
OCR	Optical Character Recognition
OTP	One-Time Password
ReLU	Rectified Linear Unit
ResNet	Residual Network
RSNA	Radiological Society of North America
SDLC	Software Development Life Cycle
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TB	Tuberculosis
XAI	Explainable Artificial Intelligence

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Respiratory diseases such as Tuberculosis (TB) and Pneumonia are among the most serious health problems affecting millions of people around the world. Both of these diseases mainly affect the lungs and can cause symptoms such as persistent cough, chest pain, fever, weakness, and difficulty in breathing. Pneumonia is a lung infection that causes inflammation in the air sacs of the lungs, which may fill with fluid or pus, making breathing difficult for the patient. Tuberculosis, on the other hand, is a bacterial disease caused by Mycobacterium tuberculosis that usually affects the lungs and spreads through the air from one infected person to another through coughing or sneezing.

According to the World Health Organization (WHO), Tuberculosis is one of the leading infectious disease killers worldwide, and Pneumonia is also a major cause of death, especially among children under the age of five and elderly people. Early detection of these diseases is very important because delayed diagnosis can lead to serious health complications and even death. If these lung infections are not diagnosed at an early stage, they can spread inside the lungs and may become life-threatening, making proper and timely diagnosis essential for effective treatment and recovery.

Chest X-ray imaging is one of the most commonly used medical techniques for detecting lung infections such as TB and Pneumonia. In traditional diagnosis, doctors or radiologists manually examine chest X-ray images to identify abnormal patterns in the lungs that indicate infection. However, manual examination has several limitations. It is a time-consuming process, depends heavily on the availability of expert radiologists, and may sometimes result in incorrect diagnosis due to human error or heavy workload. In addition, in rural or low-resource areas, access to experienced medical professionals may be limited, which makes early diagnosis more difficult.

With the rapid advancement of Artificial Intelligence (AI) and Deep Learning techniques, it has become possible to develop automated systems that can assist doctors in diagnosing lung diseases more accurately and efficiently. Convolutional Neural Networks (CNN), which are widely used in medical image analysis, can automatically learn important features from chest X-ray images and identify disease patterns. In this project, a deep learning-based system is developed to automatically detect and localize Tuberculosis and

Pneumonia from chest X-ray images. The proposed system aims to detect the presence of disease and highlight the infected regions in the lungs, thereby helping healthcare professionals in making faster and more reliable diagnostic decisions.

1.2 Problem Statement

In the context of Nepal, TB and pneumonia remain persistent public health burdens in Nepal despite continuous national efforts and global support. Nepal reports approximately 30,000 new TB cases annually, with an estimated 117 cases per 100,000 population, reflecting a slow decline in incidence compared to regional trends [1]. Likewise, pneumonia remains one of the leading causes of illness and death among children under five. Although pneumonia diagnosis and treatment protocols exist, many rural and remote health facilities lack adequate resources and diagnostic capacity, resulting in delayed or inappropriate management [2]. Specially in the rural areas, people could not get access to radiography experts about their X-ray reports analysis which creates the need for the development of such system which addresses these issues.

1.3 Objectives

The main objective of this project are as follows:

- To identify the pulmonary infections and diseases
- To localize the affected area and display to user

1.4 Scope and Applications

The scope of this project includes the automatic detection and localization of lung diseases such as Tuberculosis and Pneumonia using chest X-ray images. The system performs binary classification tasks, such as distinguishing between Normal and TB-infected lungs, as well as Normal and Pneumonia-infected lungs. The project also involves image preprocessing and data augmentation techniques to enhance the dataset quality and improve model learning capability. Furthermore, the trained CNN model is evaluated using different performance metrics to ensure that the system performs reliably in detecting lung diseases from medical images.

This project has a wide range of applications in the healthcare sector. The developed system can be used in hospitals and diagnostic centers to support doctors in analyzing chest X-ray images quickly and efficiently. It can also be useful in rural healthcare centers where experienced radiologists may not be available, helping to improve early disease detection in remote areas. In addition, this system can be integrated into telemedicine platforms to provide remote diagnostic support to patients. Researchers can also use this system for medical image analysis and disease detection studies. It is important to note that this system is designed to act as a decision support tool for healthcare professionals and is not intended to replace doctors, but rather to assist them in making accurate and faster diagnostic decisions.

1.5 Project Features

The features of this project are follows:

- Allows the user to scan X-ray images of the lungs and detect pulmonary diseases accurately.
- Provides a user-friendly interface, making the application easy to use for people of all age groups.
- Enables users to update their personal information, such as name and password.
- Displays the user’s disease history, showing previous X-ray results along with diagnosis and confidence scores.
- Ensures secure login using OTP via email (two-factor authentication) to protect user accounts.

1.6 Report Organization

This report is organized into different chapters to clearly explain the overall development and implementation of the proposed system. Chapter 1 introduces the project, including the background information, problem statement, objectives, scope, and applications of the system. Chapter 2 presents the background study and literature review related to Tuberculosis, Pneumonia, deep learning techniques, and previously developed disease detection systems. Chapter 3 describes the system analysis and design, including development methodology, requirement analysis, feasibility analysis, and system architecture. Chapter 4 explains the implementation and testing process, including dataset usage, model training, evaluation, and result analysis. Finally, Chapter 6 presents the conclusion of the project along with its limitations and future recommendations for further improvement of the system.

CHAPTER 2
BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Tuberculosis (TB) is a serious bacterial infection caused by Mycobacterium tuberculosis that mainly affects the lungs and spreads through the air when an infected person coughs or sneezes. Common symptoms of TB include persistent cough, weight loss, night sweats, fever, and chest pain. In chest X-ray images, TB often appears in the form of white patches, cavities, or lung opacity, which indicate infection in the lung tissues.

Pneumonia is another lung infection that causes inflammation in the air sacs of one or both lungs. These air sacs may fill with fluid or pus, making breathing difficult for the patient. Symptoms of Pneumonia include fever, chest pain, breathing difficulty, and cough with mucus. In chest X-ray images, Pneumonia appears as cloudy areas, fluid-filled regions, or lung consolidation, which help doctors identify the presence of infection in the lungs.

Deep Learning is a subfield of Artificial Intelligence that uses neural networks with multiple layers to analyze complex patterns in data. It is widely used in medical image processing for tasks such as image classification, object detection, and pattern recognition. One of the most popular deep learning models used for image analysis is the Convolutional Neural Network (CNN). CNN is specially designed for processing image data and consists of several layers such as convolution layers, pooling layers, flatten layers, and fully connected dense layers.

The convolution layer extracts important features such as edges, shapes, and textures from the image. The pooling layer reduces the image size while preserving important information. The flatten layer converts the two-dimensional feature maps into a one-dimensional vector, which is then passed to the dense layer for classification. Finally, the sigmoid activation function is used in the output layer to produce a binary classification result, which helps in identifying whether the X-ray image is normal or infected with TB or Pneumonia.

2.2 Literature review

The detection of lung diseases such as Tuberculosis (TB) and Pneumonia using chest X-ray images has improved significantly over the years with the advancement of computer-aided diagnostic technologies. In earlier studies, the diagnosis of lung diseases was mainly performed manually by radiologists through visual examination of chest X-ray images. Later, researchers introduced computer-aided image processing techniques such as edge detection, contrast enhancement, and texture analysis to assist medical experts in identifying abnormal lung regions. These traditional image processing approaches were able to extract important visual features from X-ray images and support early abnormality detection. However, these techniques required manual feature extraction and were limited in terms of accuracy and performance when applied to complex lung disease detection tasks [3].

Around 2014, the availability of publicly accessible medical imaging datasets such as the Montgomery County Chest X-ray dataset and Shenzhen Chest X-ray dataset significantly contributed to the advancement of automated TB detection research. These datasets provided labeled chest X-ray images for normal and TB-infected cases, which enabled researchers to train machine learning and deep learning models more effectively. In many research studies, preprocessing techniques such as image resizing, normalization, noise removal, and data augmentation were applied before model training. Augmentation methods such as rotation, flipping, and zooming were particularly useful in increasing dataset diversity and reducing overfitting during model training, which ultimately improved the performance of deep learning-based diagnostic systems [4].

With the rapid growth of Artificial Intelligence and deep learning technologies after 2015, Convolutional Neural Networks (CNNs) became widely used for medical image classification tasks. CNN models have the capability to automatically extract meaningful features from raw chest X-ray images without requiring manual feature engineering. One of the most significant developments in pneumonia detection was the introduction of the CheXNet model, a deep CNN architecture trained on a large chest X-ray dataset. This model achieved performance comparable to expert radiologists in detecting pneumonia and also introduced heatmap-based localization techniques to highlight the infected lung regions. This demonstrated the ability of deep learning models to not only classify diseases but also provide visual explanations of their predictions [5].

Similarly, CNN-based models have also shown promising results in the detection of Tuberculosis from chest X-ray images. Several research studies have reported high accuracy and sensitivity in distinguishing TB-infected lungs from normal lungs using deep learning techniques. In addition, pretrained deep learning architectures such as ResNet, VGG16, and U-Net have been widely used by researchers for classification and localization of lung diseases. These models are capable of learning complex radiographic patterns associated with lung infections and have demonstrated improved performance compared to traditional machine learning approaches [6].

Recent studies have focused on developing lightweight and efficient deep learning models that can be deployed in real-world healthcare environments, especially in rural and low-resource areas where access to expert radiologists may be limited. These models aim to provide fast and accurate disease detection using portable X-ray devices or cloud-based healthcare platforms. Despite achieving high accuracy levels, many existing systems still face challenges such as overfitting due to limited dataset size, lack of proper infection localization, and higher False Negative rates. Compared to previous research works, this project focuses on detecting both Tuberculosis and Pneumonia simultaneously while also localizing the infected regions in chest X-ray images using improved preprocessing techniques and strong evaluation metrics to enhance diagnostic reliability [7].

CHAPTER 3
ANALYSIS AND DESIGN

3.1 Development Methodology

The development of this project involves two parallel but complementary approaches: traditional software development and AI model development. The software component provides a structured framework for user interaction, data management, and system functionality, while the AI component performs intelligent analysis and prediction using chest X-ray images. By developing them in parallel, the project ensures clarity of concept, well-defined goals, and efficient handling of datasets, while also allowing smooth integration of the AI model into the software system. This approach enables systematic planning, precise implementation, and accurate evaluation, resulting in a robust, reliable, and intelligent system that meets both functional and analytical objectives.

3.1.1 Software Development Life Cycle

The software part of this project follows the Software Development Life Cycle (SDLC), which provides a structured approach for developing reliable, maintainable, and efficient software. Among the various SDLC models, the Waterfall model has been adopted in this project due to its sequential and well-organized nature, which is particularly suitable for projects that have well-defined requirements and clear objectives from the beginning.

The Waterfall model was chosen for the following reasons:

- **Clear Goals and Requirements:**
  The project has clearly defined objectives, including the integration of AI-based chest disease detection and user-friendly software features. Each phase of the Waterfall model ensures that these goals are thoroughly analyzed, designed, and implemented before moving forward.
- **Structured Planning and Concept Clarity:**
  The sequential flow helps maintain clarity of the overall system design and AI integration. Developer can focus on one phase at a time (e.g., requirement gathering, system design) without confusion or overlap.

- **Predictable Milestones and Risk Minimization:**
  Each phase is completed and verified before the next begins, which reduces the chance of introducing errors during AI integration or software development. This approach ensures that both the software functionality and AI model performance are systematically evaluated at each stage.
- **Documentation and Maintenance:**
  Waterfall emphasizes proper documentation in each phase, which is crucial for understanding the AI model, dataset details, and software logic. It makes future updates, model retraining, or feature enhancements easier and more organized.

The structured nature of the Waterfall model ensured clarity, minimized development risks, and supported smooth integration of the AI-based disease detection module into the overall software system.

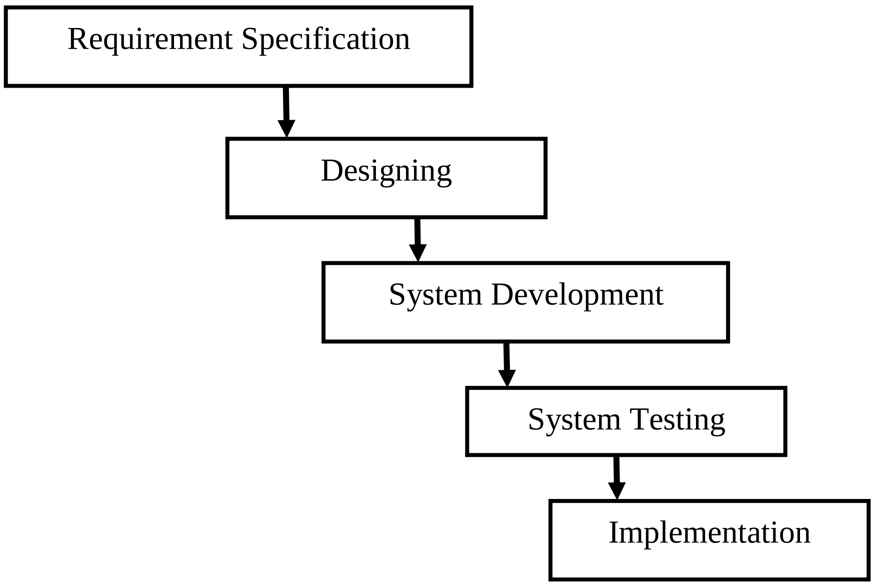


Figure 3.1: Waterfall Model

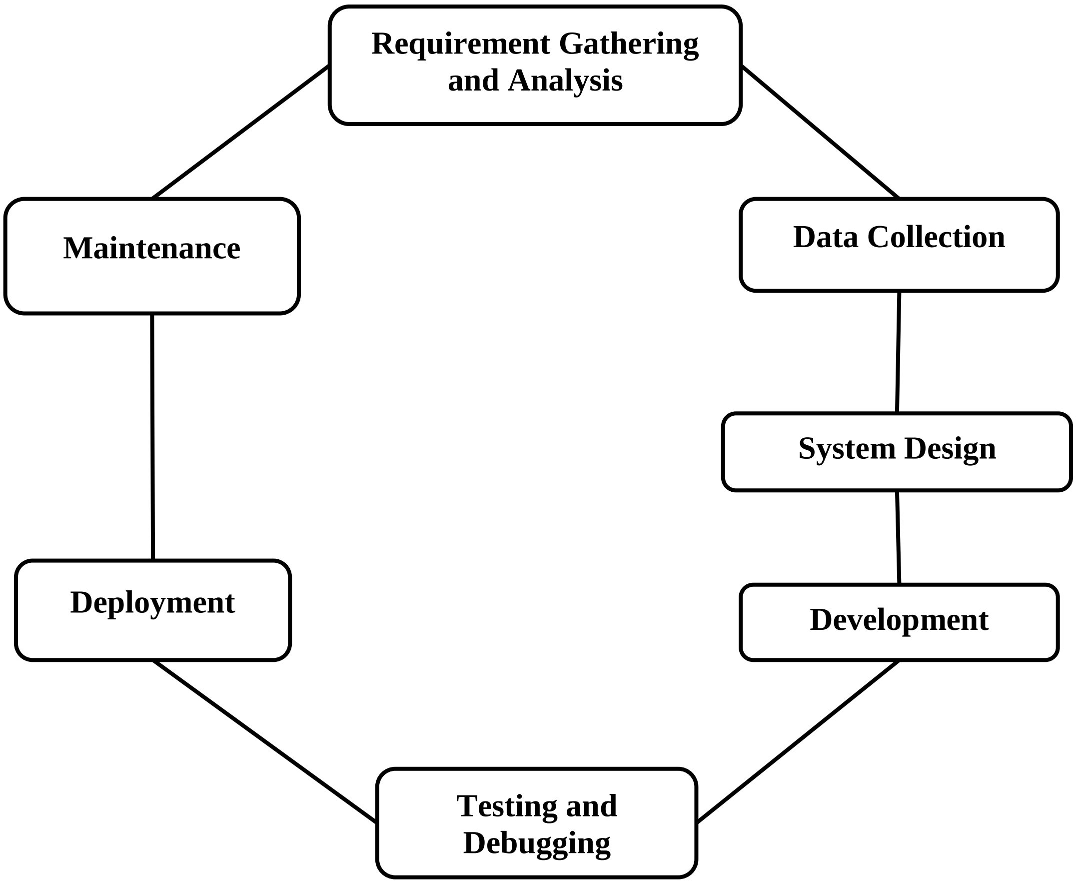


Figure 3.2: SDLC Life Cycle

3.1.2 AI Model Development Lifecycle

The development of the AI-based disease detection module differs from traditional software development processes, as it requires continuous experimentation and performance improvement. Therefore, an iterative development approach has been adopted for training and optimizing the convolutional neural network (CNN) used for detecting Tuberculosis and Pneumonia from chest X-ray images.

The iterative AI model development approach was chosen for the following reasons:

- **Performance Optimization Through Experimentation:** Deep learning models such as CNN require multiple training cycles in order to achieve satisfactory prediction accuracy. The iterative approach allows continuous adjustment of model parameters such as learning rate, batch size, and number of training epochs based on performance evaluation results.

- **Improvement of Model Generalization:** Medical image datasets are often limited in size, which may lead to overfitting if the model is trained only once. The iterative approach allows repeated training using different preprocessing and augmentation techniques to improve the model's ability to perform accurately on unseen data.
- **Flexibility in Model Architecture Design:** Unlike software modules, AI models require modifications in network architecture during development. The iterative method enables experimentation with convolutional layers, pooling layers, and dense layers in order to identify the most suitable architecture for disease classification.
- **Reduction of False Negative Predictions:** In medical diagnosis, missing a disease condition (False Negative) can lead to serious consequences. The iterative approach allows continuous monitoring of evaluation metrics such as recall and F1-score to ensure reliable disease detection.
- **Reliable Model Integration:** After achieving satisfactory performance through iterative training and validation, the finalized AI model can be integrated into the backend system for real-time prediction within the mobile application.

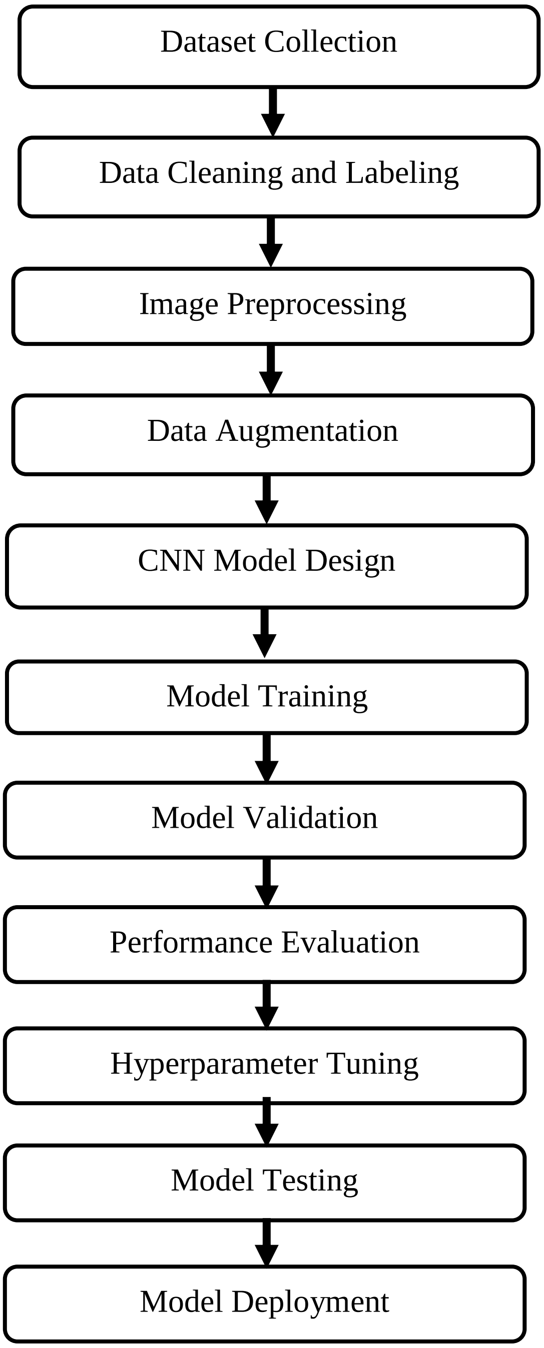


Figure 3.3: AI Model Development Flowchart

3.2 Requirement Gathering and Analysis

Requirement analysis is an essential step in system development that helps in identifying the functionalities and performance criteria required for the successful operation of the proposed system. The requirements of the system are classified into functional and non-functional requirements.

3.2.1 Functional Requirement

Functional requirements describe the operations that the system must perform to achieve its intended objective. The proposed system allows users to upload chest X-ray images for disease analysis. Once the image is uploaded, the system preprocesses the image by resizing and normalizing it before passing it to the trained CNN model for prediction. The system detects the presence of lungs and analyzes the condition of the lungs to identify diseases such as pneumonia and tuberculosis. It highlights the infected region within the chest X-ray image using visualization techniques such as heatmaps. The system also provides a probability score indicating the confidence level of prediction. Furthermore, the application maintains user records for future reference and provides authentication features for secure login and registration.

3.2.2 Non-Functional Requirements

Non-functional requirements describe the quality attributes and performance characteristics of the system. The proposed system should provide accurate disease detection results with minimal prediction time. The user interface of the application must be simple and user-friendly so that individuals with varying technical knowledge can operate it easily. The system should also be reliable and scalable to handle large volumes of medical image data in the future. It must ensure stable performance even when tested with new and unseen X-ray images. Additionally, the system should support easy maintenance and updates for future improvements and model retraining.

3.2.3 System Requirements

The system required a combination of hardware and software components to function effectively. As, it is a mobile application, there is not any big loads to operate. So, the minimal requirement of operation of this system is light and simple.

3.2.3.1 Software Requirement

As the project is a mobile-based application, the Flutter framework is utilized for frontend development using Dart as the primary programming language to ensure a responsive and user-friendly interface across multiple platforms. The backend is developed using Spring Boot, providing a robust and scalable environment for managing application logic and APIs, while MySQL is employed for data storage and management due to its reliability and strong data handling capabilities. Furthermore, to enable accurate disease detection, the system utilizes a large collection of real-world medical image datasets that has been used to train and validate the machine learning model, allowing it to identify patterns and classify diseases with higher precision and reliability.

3.2.3.2 Hardware Requirement

To develop and program this application, a computer or laptop with a minimum of 8GB RAM was used to ensure smooth development and testing processes. If there is dedicated GPU in computer, that would be more fruitful if using hardware acceleration for ML works. Since the project outcome is a mobile application, the minimum requirement for end users to operate it is a smartphone with adequate processing capability and storage.

3.3 Feasibility Analysis

Feasibility analysis evaluates the practicality of implementing the proposed system under available resources and constraints.

3.3.1 Technical Feasibility

The system is technically feasible as it utilizes widely available technologies such as the Flutter framework for frontend development using Dart programming language, Spring Boot for backend services, and MySQL database for data storage and management. The AI model is developed using deep learning techniques based on CNN architecture for analyzing chest X-ray images. These tools provide sufficient support for building an efficient medical image classification system.

3.3.2 Operational Feasibility

The system is designed with a user-friendly interface that allows healthcare professionals to upload chest X-ray images and receive disease prediction results easily. The application does not require advanced technical skills for operation, making it suitable for use in hospitals, clinics, and diagnostic centers.

3.3.3 Economic Feasibility

The proposed system is economically feasible as it utilizes open-source tools and publicly available medical image datasets for development and training of the AI model. No expensive hardware or licensed software is required, making the system cost-effective for healthcare organizations.

3.3.4 Schedule Feasibility

The project has been completed within the planned timeframe by dividing the development process into different phases such as requirement analysis, system design, implementation, training, testing, and documentation.

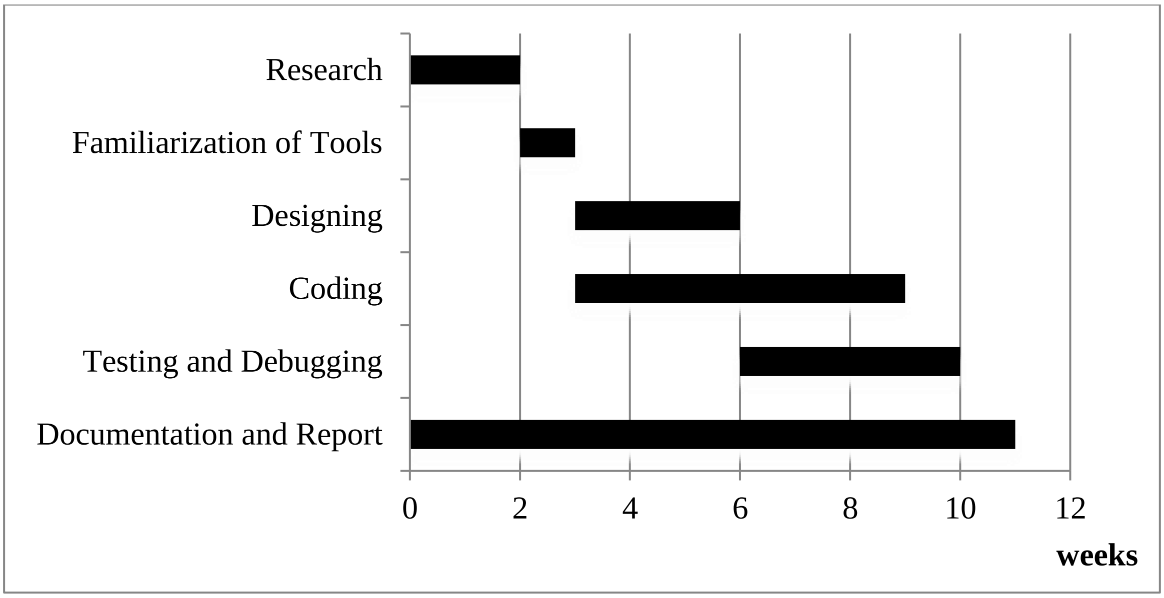


Figure 3.4: Gantt Chart

- **Research:** This initial phase, spanning weeks 1 to 2, involved an in-depth review of existing research and related technologies, it established a strong conceptual foundation and guides the direction of the project.
- **Familiarization with Tools:** Conducted during Weeks 2 and 3, this stage ensured the project team becomes proficient with essential tools and platforms. It was crucial for technical readiness and smooth implementation in subsequent phases.
- **Designing:** Scheduled from Weeks 3 to 5, the designing phase focused on creating a robust system architecture and planning the overall workflow. It translated theoretical understanding into a structured project blueprint.
- **Coding:** Taking place between Weeks 3 and 9, this was the core development stage where the actual system was built through programming. It converted the design framework into a tangible and functional application
- **Testing and Debugging:** Running from Weeks 6 to 10, this critical quality control phase involved identifying, isolating, and fixing errors in the system. It ensured the reliability, performance, and readiness for deployment.

- **Documentation and Report:** This stage, spanning up to Week 11, focused on compiling technical documentation and the final report. It captured the full project journey, highlighting methodologies, outcomes, and key insights.

3.4 System Design

For the design phase of this project, the focus is on creating a robust, scalable, and user-friendly architecture. The system is designed to efficiently handle chest X-ray images for the detection of diseases such as pneumonia and tuberculosis while providing accurate localization of affected areas. The design will cover the following aspects:

3.4.1 Theme Selection

To identify the most appropriate user interface theme for the application, a survey was conducted among potential users using Google Forms. A total of 103 responses were collected and analyzed to evaluate user preferences. The analysis revealed that Theme 5 received the highest number of votes, accounting for 25.2% of the total responses. Theme 4 and Theme 3 followed with 22.3% and 21.4% respectively. In comparison, Theme 1 and Theme 2 each received 15.5% of the votes, representing the lowest level of user preference.

Based on the statistical evaluation of the survey results, Theme 5 was selected for implementation in the application, as it reflects the majority preference of users. Selecting the most preferred theme enhances user satisfaction, improves usability, and contributes to a more engaging and intuitive overall user experience. The detailed designs of all proposed themes used in the survey are provided in the appendices section for reference.

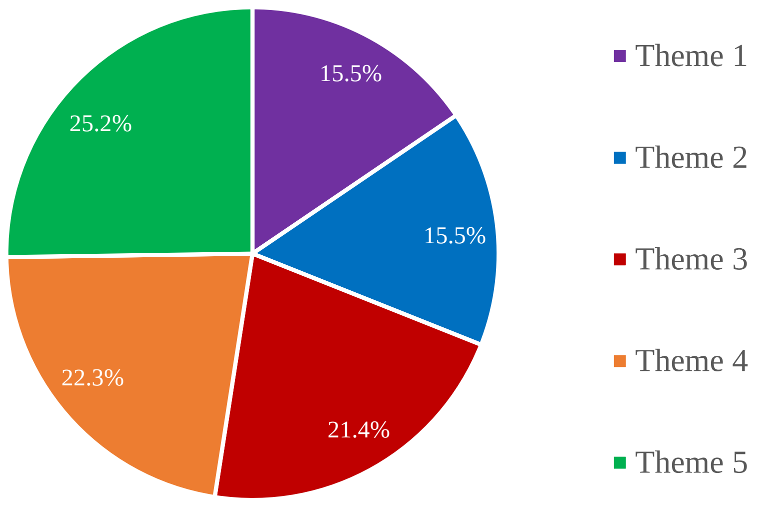


Figure 3.5: Pie Chart Representation of Survey

3.4.2 Three-Tier Architecture

A three-tier architecture comprises the following layers:

- **Presentation Layer (Frontend):** User Interface (UI) for uploading X-ray images, displaying results, and providing guidance.
- **Application Layer (Backend):** Processes images, runs AI Models and generates results.
- **Database Layer:** Stores user data, image metadata, and model predictions.

3.4.3 Structured Schema for Datasets and Models

- Organized storage of X-ray images and trained models to ensure efficient retrieval and processing.
- Enables the AI model to access high-quality input data for accurate disease detection.
- The system identifies affected lungs areas quickly, reducing waiting time for the user.
- Supports visualization of disease-affected regions on the X-ray image.

3.4.4 Input Image Acquisition

The proposed lung disease detection system requires chest X-ray images as an input to perform disease classification and prediction. Since users cannot directly generate digital X-ray images within the application, the system allows users to capture an image of an existing chest X-ray using a mobile phone camera or upload it from the device storage. The uploaded image is then processed by the system for further analysis using the trained convolutional neural network (CNN) model.

For accurate disease detection, the following basic guidelines are considered during image acquisition:

- The user selects the chest X-ray image from the device gallery.
- The uploaded image should be clear and properly focused.
- The chest X-ray image should be captured in a straight orientation.
- The lung region must be fully visible in the uploaded image.
- Images that are blurred, distorted, or unclear may affect the prediction accuracy of the model.

Once the image is uploaded, it is automatically resized and normalized by the system before being fed into the trained CNN model for disease prediction.

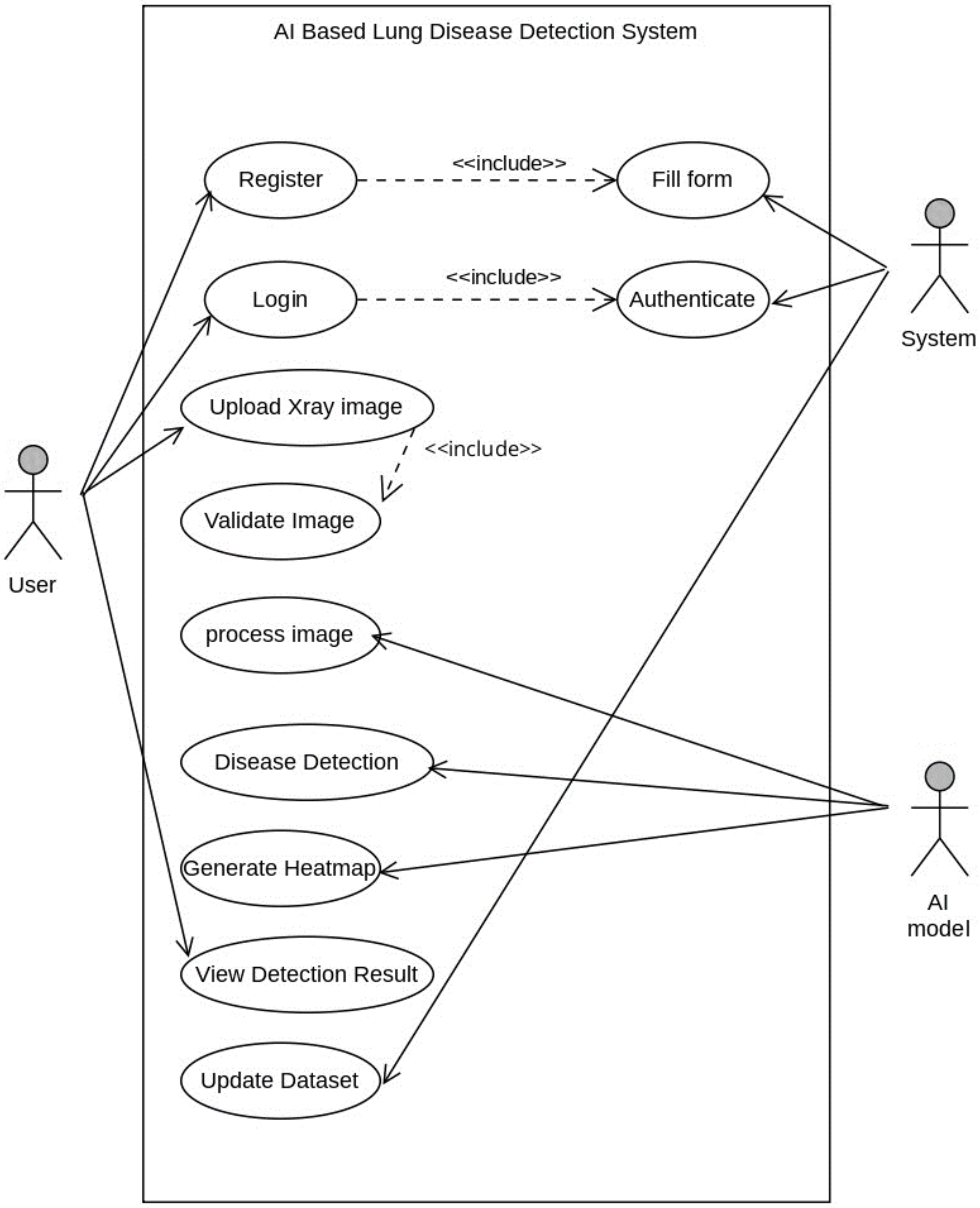


Figure 3.6: Use Case Diagram

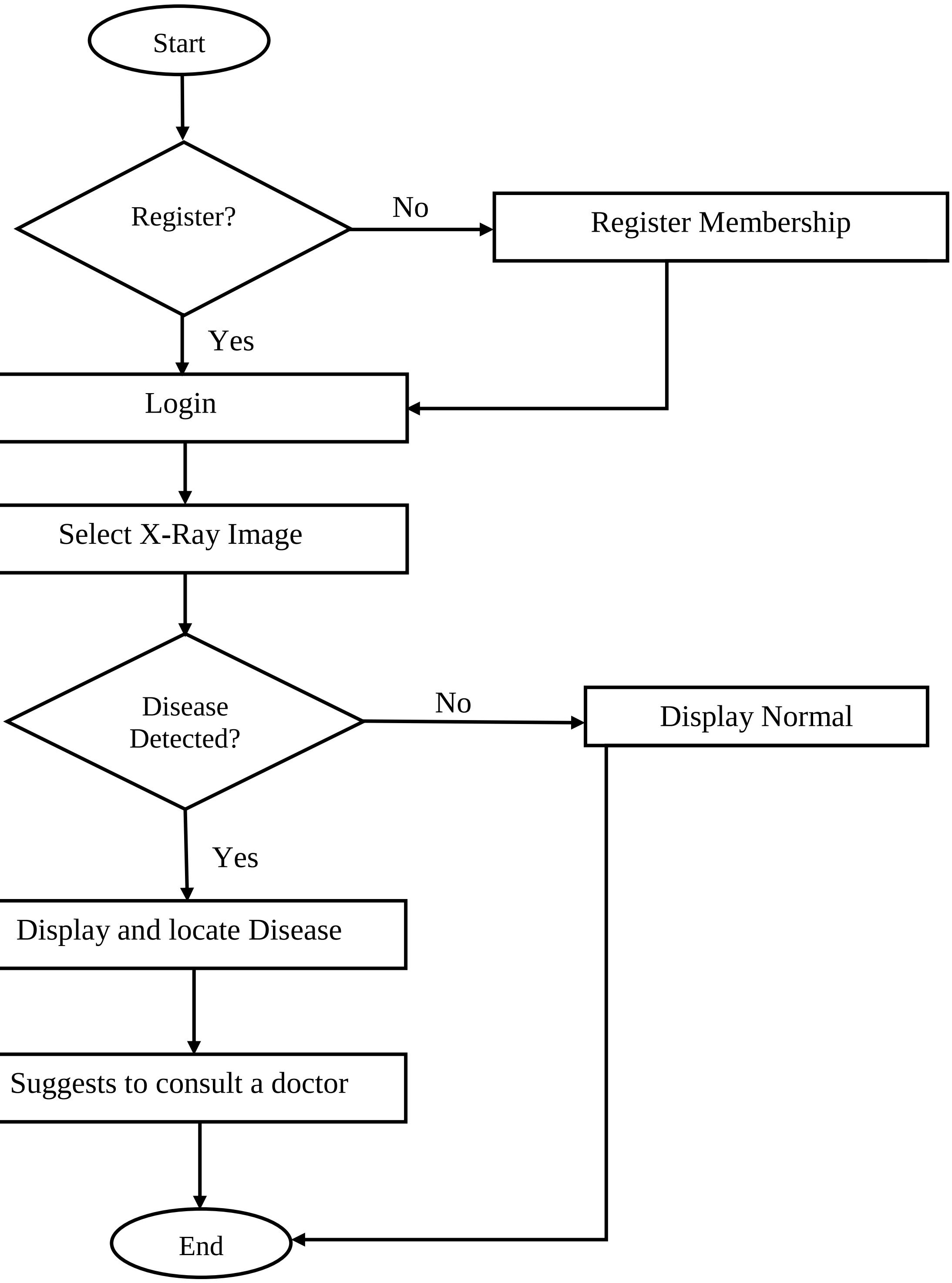


Figure 3.7: Flow Chart

3.5 System Development

The system development phase for this system involves implementing the proper planned designs into a functional system. This phase will include:

3.5.1 Frontend Development

- Create user interface for login, signup, profile update (name, password), uploading X-ray images, and viewing results.
- Display user's disease history and previous X-ray predictions.
- Provide input validation (name, email, password format) for better user experience.
- Ensure UI is user-friendly and responsive across devices.
- Connect with backend APIs to fetch and display user-specific data (profile, history, predictions).

3.5.2 Backend Development

- Handle user authentication (login, signup) and profile updates (name, password).
- Implement OTP/2FA via email using SMTP for secure login or password reset.
- Process requests from frontend and return user-specific data or AI predictions.
- Integrate the AI model for disease detection and send predictions to the frontend.

3.5.3 Database Development

- Store all user data, including profile (name, email, password), AI prediction results, and disease history.
- Ensure data integrity and security for sensitive information (passwords, email, OTP).
- Support efficient retrieval of user-specific information for frontend display.

3.6 Chatbot Integration

To enhance user interaction and provide additional health guidance, a chatbot feature has been integrated into the application using the DeepSeek API. The chatbot acts as an intelligent assistant that provides informative responses specifically related to lung diseases, particularly Pneumonia and Tuberculosis (TB). The primary objective of integrating the chatbot is to assist users in understanding disease related information after receiving AI-based prediction results. While the CNN models perform image-based diagnosis, the chatbot provides textual explanations, symptom descriptions, precautionary measures, and general awareness regarding these diseases.

The chatbot is powered by a Large Language Model (LLM), which is a deep learning-based natural language processing model trained on vast amounts of textual data. LLMs are typically built using transformer architectures that enable them to understand context, semantics, and relationships between words in a sentence. By leveraging the DeepSeek API, the application sends user queries to the LLM, which processes the input and generates context-aware and human-like responses.

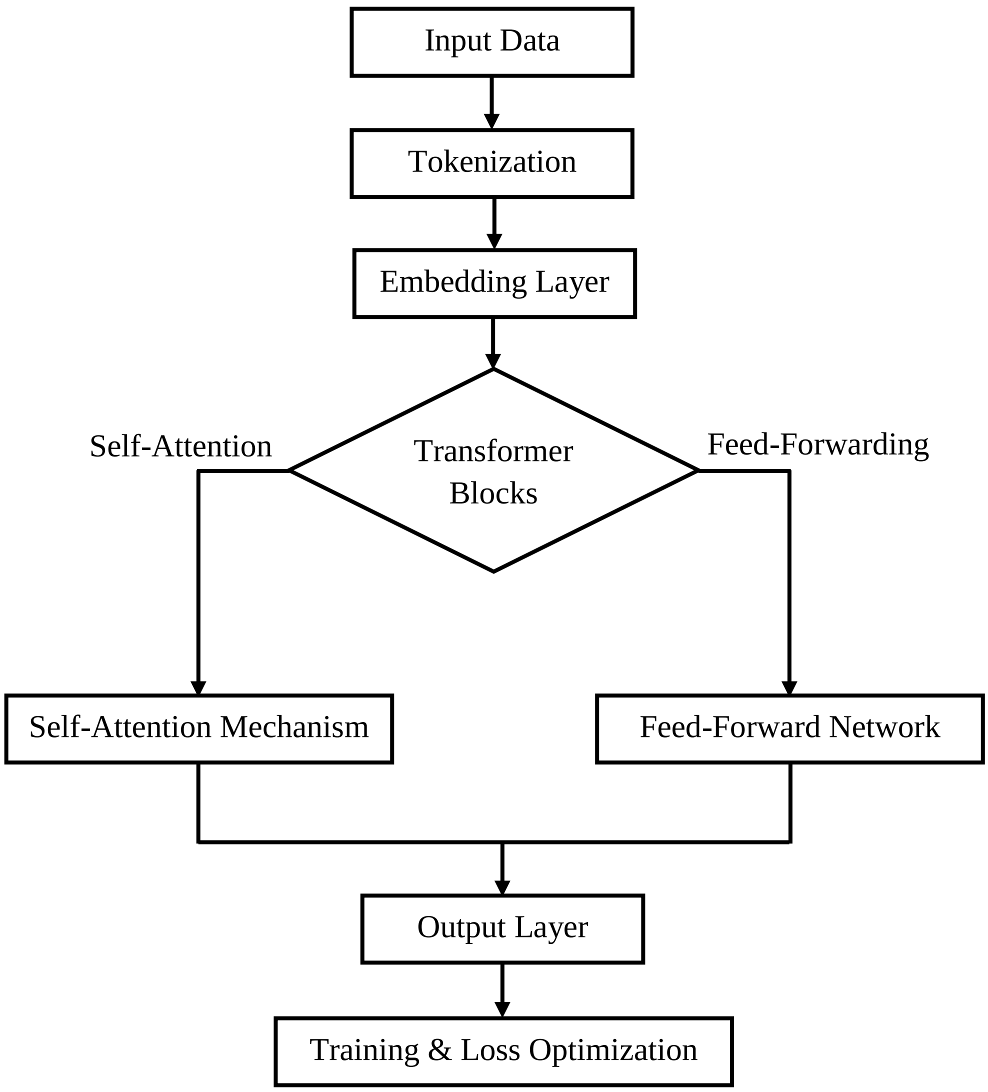


Figure 3.8: LLM Architecture

The chatbot integrated in the proposed system is powered by a transformer-based Large Language Model (LLM) architecture. This architecture enables the chatbot to understand user queries, process contextual meaning, and generate meaningful responses related to lung diseases. The general architecture of an LLM consists of several layers including input processing, embedding layer, transformer blocks, and output decoding layer which together help in natural language understanding and generation.

CHAPTER 4
IMPLEMENTATION AND TESTING

4.1 Data Collection and Dataset Used

While working on this project, a lot of resources and datasets were studied. Among the crowd of different dataset, those with highly rated and trusted dataset had to be chosen and well-furnished in order to get better results while using them in the different models.

4.1.1 Dataset Collection

The implementation phase of the proposed lung disease detection system began with the collection of high-quality chest X-ray image datasets required for training and evaluating the deep learning model. Publicly available medical image datasets containing labeled images of Normal lungs, Pneumonia-infected lungs, and Tuberculosis (TB)-affected lungs were utilized in this study. The availability of labeled data is essential for supervised learning because the convolutional neural network (CNN) learns to identify disease patterns by mapping input X-ray images to their corresponding class labels.

For Pneumonia detection, the RSNA Pneumonia Detection Challenge dataset was used. This dataset consists of 30,227 chest X-ray images annotated by expert radiologists with bounding boxes indicating pneumonia-affected lung regions. The dataset was originally divided into three categories: Normal, No Lung Opacity or Not Normal, and Lung Opacity (Pneumonia). Since this project focuses on detecting pneumonia infection based on lung opacity patterns, the "No Lung Opacity or Not Normal" class was not relevant to the binary classification requirement of this system. Therefore, this category was excluded from further use.

Initially, the dataset contained 8,851 Normal images, 11,821 No Lung Opacity or Not Normal images, and 9,555 Pneumonia images, making a total of 30,227 images. The removal of the "Not Normal" class ensured that the dataset only contained meaningful samples required for detecting pneumonia infection. In addition, selected healthy (Normal) chest X-ray images were included to maintain class balance and reduce model bias during training. After performing dataset selection and balancing, the final Pneumonia dataset used for model training consisted of 10,500 Normal images and 9,555 Pneumonia images, resulting in a total of 20,055 chest X-ray images.

Table 4.1: Dataset of Pneumonia

S. No.	Datasets Name for Pneumonia	Normal	No Lung Opacity / Not Normal	Lung Opacity	Total
1	RSNA Pneumonia Detection Challenge	8,851	11,821	9,555	30,227

Similarly, for Tuberculosis detection, a single dataset was not sufficient for effective model training due to limited image availability and class imbalance. Using a small dataset may lead to overfitting and poor generalization performance of the deep learning model. Therefore, multiple publicly available TB chest X-ray datasets were combined to form a single comprehensive dataset. The merged dataset included images from TBX11K, Dataset of Tuberculosis Chest X-Rays, Montgomery Dataset, TB Chest Radiography Database, and Shenzhen Dataset. After merging all datasets, a total of 8,220 Normal images and 4,388 TB-infected images were obtained, resulting in 12,608 chest X-ray images used for TB model development.

Table 4.2: Datasets for Tuberculosis

S. No.	Datasets Name	Normal	TB	Total
1	TBX 11K	3,800	800	4,600
2	Dataset of Tuberculosis Chest X-Rays Image	514	2,494	3,008
3	Montgomery-Metadata	80	58	138
4	TB_Chest_Radiography_Database	3,500	700	4,200
5	Shenzhen_Metadata	326	336	662
Total		**8,220**	**4,388**	**12,608**

The use of a sufficiently large and balanced dataset improves the learning capability of the CNN model and enhances its ability to accurately distinguish between healthy and infected lungs.

4.1.2 Data Preprocessing

Before feeding the collected chest X-ray images into the convolutional neural network (CNN), several preprocessing steps were performed to ensure data consistency and improve model performance. The collected datasets contained images with different resolutions, brightness levels, and orientations, which could negatively affect the training process if used directly.

Data preprocessing was performed to improve model performance and ensure data consistency. First, corrupted and low-quality chest X-ray images were removed through data cleaning. All remaining images were resized to 224 × 224 pixels to maintain uniformity and match the CNN input requirements, which also reduced computational complexity. Next, pixel normalization was applied by scaling intensity values from 0–255 to 0–1, improving numerical stability and enabling faster model convergence.

To handle class imbalance and reduce overfitting, data augmentation techniques such as rotation, horizontal flipping, zooming, and shifting were applied. This increased dataset diversity and helped the CNN model generalize better on unseen data.

Finally, the preprocessed dataset was divided into training, validation, and testing sets in the ratio of 70%, 20%, and 10%, respectively. The training set was used to train the CNN model, the validation set was used for hyperparameter tuning and performance monitoring, and the testing set was used for final evaluation of the model's prediction capability.

These preprocessing steps ensured that the input data provided to the CNN model was clean, consistent, and suitable for effective disease classification.

Table 4.3: Pneumonia Dataset after Preprocessing

S. No.	Datasets Name	Normal	Lung Opacity	Total
1	RSNA Pneumonia Detection Challenge	10,624	9,555	20,179

As TB database were fetched from different source, those need to be combined in such way that the combined data must be good enough to train, test and validate the model. So, the datasets were combined and a new bulk dataset including 8,220 normal images, 4,388 TB infected images leading the total 12,608 images for the model.

4.2 Implementation

The lungs disease detection application is developed to address the increasing needs for the accessible and to evaluate the probability of lungs disease occurrence by using the chest X-ray image into major two disease categories using CNNs.

4.2.1 Model Architecture

For this project, a Convolutional Neural Network (CNN) has been selected as the AI model due to its proven effectiveness in medical image classification, particularly for chest X-rays. CNNs are capable of automatically learning important features, such as edges, textures, and lesions, which are critical for detecting Pneumonia, Tuberculosis (TB), and Normal lungs.

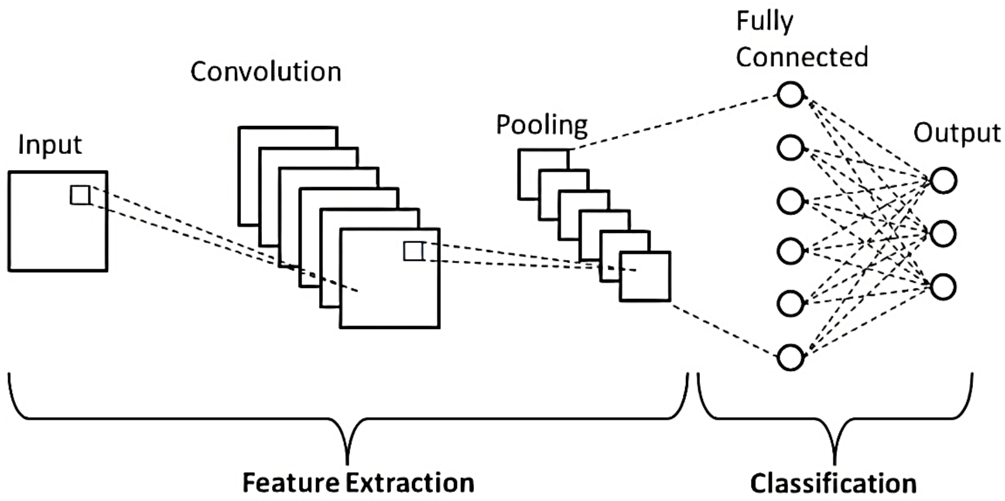


Figure 4.1: CNN Architecture

Key components of the CNN architecture:

- **Convolutional Layers:** Convolutional layers are the primary building blocks of a CNN and are responsible for automatically extracting meaningful features from chest X-ray images. When filters slide across the image, they capture low-level patterns such as edges and corners, mid-level features like textures and shapes, and high-level disease-related structures including opacities, consolidations, or abnormal lung patterns seen in pneumonia and tuberculosis. This multi-level feature extraction enables the model to understand the visual characteristics of diseased and healthy lungs without requiring manual feature engineering. Convolutional layers are essential because they help the system learn exactly which pixel regions indicate disease, allowing accurate and robust classification.

- **Pooling Layers (Max/Average Pooling):** Pooling layers are used to progressively reduce the spatial dimensions of feature maps while preserving the most important information. Max pooling selects the highest activation value in a region, while average pooling computes the average value. By reducing the size of feature maps, pooling decreases computational cost, prevents the network from focusing on noisy or irrelevant details, and improves generalization by avoiding overfitting. This simplification helps the model retain only the essential disease-related features such as highlighted regions in the lungs, making the network faster, more efficient, and more reliable.
- **ReLU Activation Functions:** The ReLU (Rectified Linear Unit) activation function introduces non-linearity into the network, enabling the CNN to learn complex patterns that cannot be captured by linear transformations alone. ReLU works by converting all negative pixel values to zero while keeping positive values unchanged, which speeds up the training process and prevents the vanishing gradient problem. This non-linear behavior helps the model understand intricate lung abnormalities visible in X-rays, such as irregular opacities, blurred boundaries, or unusual texture variations associated with pneumonia and tuberculosis. Without activation functions like ReLU, the network would fail to capture such complex medical patterns.
- **Fully Connected Layers:** Fully connected layers act as the final decision-making components of the model. After the convolutional and pooling layers extract and compress image features, the fully connected layers combine all learned information to classify the image into one of the output categories: Pneumonia, TB, or Normal. These layers analyze high-level features such as the overall lung structure, presence of abnormalities, and intensity patterns.
- **Optimizer and Loss Function:** The training of the model relies on two key components: the optimizer and the loss function. The Adam optimizer is used because it adaptively adjusts learning rates during training, ensuring faster convergence and stable performance even on complex medical images. It combines the advantages of momentum and RMSprop, making it highly efficient for deep learning tasks. The cross-entropy loss function measures how different the model's

predictions are from the true labels and is ideal for multi-class classification problems like Pneumonia, TB, and Normal. During training, the loss function evaluates errors, and the optimizer updates the network's weights to minimize those errors. Together, Adam and cross-entropy ensure that the model learns accurately and improves consistently over time.

4.2.1.1 ResNet-UNet Model Architecture

The ResNet-UNet model combines the strengths of ResNet and UNet architectures. The encoder part of the network uses a ResNet backbone for deep feature extraction, allowing the model to capture complex lung patterns. The decoder part follows the UNet structure, which includes up-sampling layers and skip connections to reconstruct high-resolution segmentation maps.

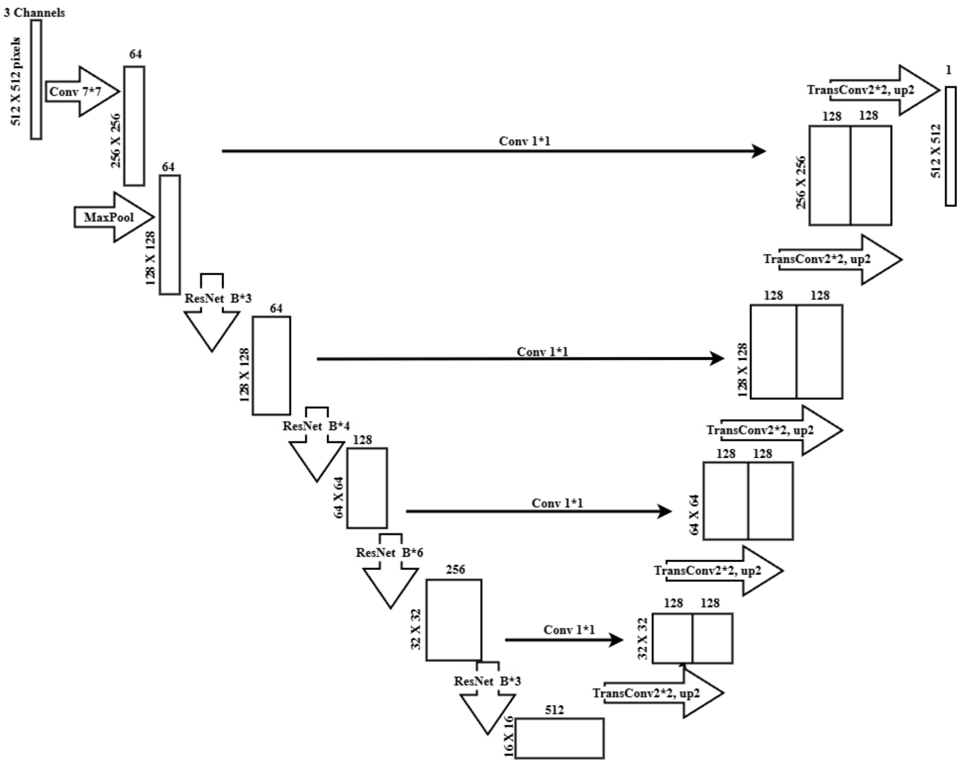


Figure 4.2: ResNet-UNet Architecture

The key components of ResNet-UNet architecture are as follows:

- **Encoder (ResNet Backbone):** The encoder utilizes a ResNet (Residual Network) backbone, which is responsible for deep feature extraction. By using residual blocks, the model can go deeper without the risk of vanishing gradients, allowing it to capture intricate and complex lung patterns from chest X-rays.

- **Decoder (U-Net Structure):** The decoder follows the traditional U-Net structure, featuring up-sampling layers that expand the feature maps back to the original image resolution.
- **Skip Connections:** A critical feature of this architecture is the use of skip connections, which pass high-resolution information directly from the encoder to the decoder. This ensures that spatial details lost during the down-sampling process are preserved, which is essential for accurately localizing and reconstructing the infected regions as high-resolution segmentation maps.

4.2.1.2 ResNet50 Model Architecture

The ResNet50 is a deep Convolutional Neural Network consisting of 50 layers and is part of the Residual Network (ResNet) family. It was introduced to address the degradation problem in very deep neural networks, where increasing depth leads to reduced performance due to vanishing gradients. ResNet50 overcomes this issue using residual learning with shortcut (skip) connections. ResNet50 is structured into multiple convolutional blocks organized into stages. Each stage contains residual blocks that allow direct information flow from earlier layers to later layers.

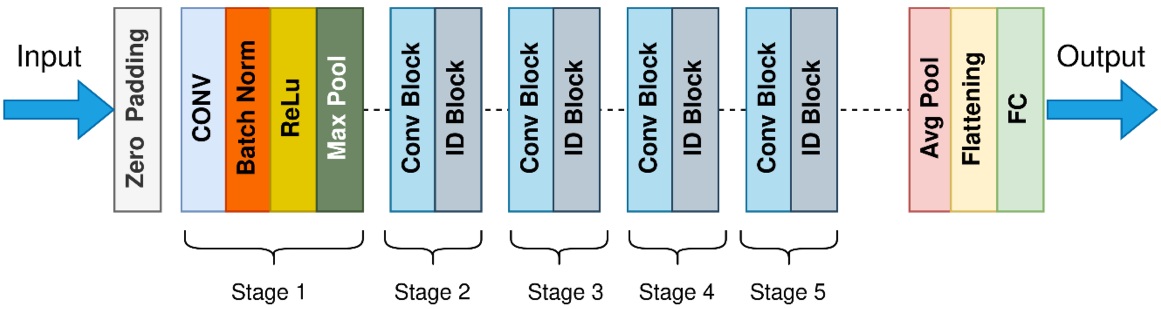


Figure 4.3: ResNet50 Model Architecture

The architecture of ResNet50 contains following key components:

- **Residual (Skip) Connections:** Residual connections are the most important feature of ResNet50. They allow the input of a block to be added directly to its output. This helps the network learn residual mappings instead of full transformations, prevents vanishing gradient problems, and enables successful training of very deep networks.
- **Bottleneck Architecture:** ResNet50 uses a bottleneck design inside each residual block. It consists of a 1×1 convolution for reducing feature dimensions, a 3×3

convolution for main feature extraction, and another 1×1 convolution for restoring dimensions. This design improves computational efficiency while maintaining strong feature learning capability.

- **Deep Layer Structure (50 Layers):** The architecture contains 50 layers, allowing it to learn hierarchical features. Early layers capture simple patterns like edges, while deeper layers learn complex structures and abstract image representations.
- **Batch Normalization:** Batch Normalization is applied after convolution layers to normalize feature distributions. It stabilizes training, accelerates convergence, and reduces internal covariate shift.
- **ReLU Activation Function:** The Rectified Linear Unit (ReLU) introduces non-linearity into the network. It helps the model learn complex patterns and improves computational efficiency by avoiding saturation problems found in other activation functions.
- **Global Average Pooling:** Instead of using large fully connected layers, ResNet50 uses a global average pooling layer before the final classification layer. This reduces the number of parameters, minimizes overfitting, and maintains strong generalization performance.

4.2.2 Training Process

The training process was conducted separately for Pneumonia detection and Tuberculosis (TB) detection models. For each disease category, multiple deep learning architectures were trained and evaluated to determine the most suitable model for final deployment.

For Pneumonia detection, mainly two architectures were trained and compared ResNet-UNet and MobileNet-UNet. Both models were trained under similar experimental settings to ensure a fair comparison. The objective was to evaluate which architecture provided better classification performance and segmentation capability.

For TB detection, the following classification models were trained and evaluated ResNet50 and DenseNet121. These models were selected due to their strong performance in medical image classification tasks.

Training was performed for multiple epochs while monitoring model performance to observe learning behavior and stability.

4.2.3 Testing

Comprehensive testing was performed to ensure the system's accuracy, reliability, and smooth user experience.

4.2.3.1 Testing Methodology

After the completion of training, all models were evaluated using the test dataset to measure their performance. Testing was conducted independently for Pneumonia and TB detection systems.

For Pneumonia detection, both ResNet-UNet and MobileNet-UNet were tested. The results showed that ResNet-UNet achieved higher accuracy and better balance between precision and recall compared to MobileNet-UNet. In addition, ResNet-UNet demonstrated improved segmentation performance.

For TB detection, ResNet50 and DenseNet121 were tested and compared. Although both models produced satisfactory results, ResNet50 achieved superior performance in terms of accuracy, recall, and F1-score. Since recall is particularly important in medical diagnosis to minimize false negative predictions, ResNet50 was considered more reliable for TB detection.

Based on the overall evaluation results, ResNet-UNet was selected as the final model for Pneumonia detection, and ResNet50 was selected as the final model for TB detection.

4.2.3.2 Evaluation Metrices

The AI model’s performance is evaluated using standard metrics for medical image classification, ensuring clinical relevance:

- **Confusion Matrix:** Provides True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) counts.
- **Accuracy:** Measures the overall correctness of predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Precision:** Measures the proportion of correctly predicted positive cases.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall (Sensitivity):** Assesses how effectively the model detects actual diseased cases.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **Specificity**: Evaluates how well healthy cases are correctly identified.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- **F1-Score**: Provides a balance between precision and recall, particularly useful for datasets with class imbalance.

$$\text{F1} - \text{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.2.4 Explainable AI

Explainable AI (XAI) refers to methods that make the decisions of an AI model understandable to users. Deep learning models like CNNs are highly accurate, but they normally operate like a "black box," meaning they provide results without showing how they reached those results. XAI helps solve this problem by providing clear, visual explanations of what the model is focusing on during prediction.

In this project, XAI is used to show the important regions of the lungs that the AI model uses to identify Pneumonia or Tuberculosis. Techniques like heatmaps or activation maps highlight the infected or suspicious areas on the X-ray. This helps users visually understand why the AI classified the image as "Pneumonia", "Tuberculosis" or "Normal" making the system more transparent and reliable.

By including XAI, the system becomes more trustworthy for both medical professionals and general users. It ensures that the model's predictions are not only accurate but also easy to interpret. This is especially important in medical applications, where understanding the reasoning behind a diagnosis is just as important as the diagnosis itself. XAI makes the overall system safer, clearer, and more user-friendly.

4.2.5 Result Evaluation and Analysis

After all testing and evaluating result, the models were finalized. In the result analysis, the Confusion matrix of both system models were analyzed.

4.2.5.1 Pneumonia Detection Model:

After all the train, test and evaluation were done, the confusion matrix was visualized and displayed the information about the model in different matrices. The test was done in different models, but for the Pneumonia detection model, the better outcome was seen in the ResNet-UNet Model based on the CNN architecture.

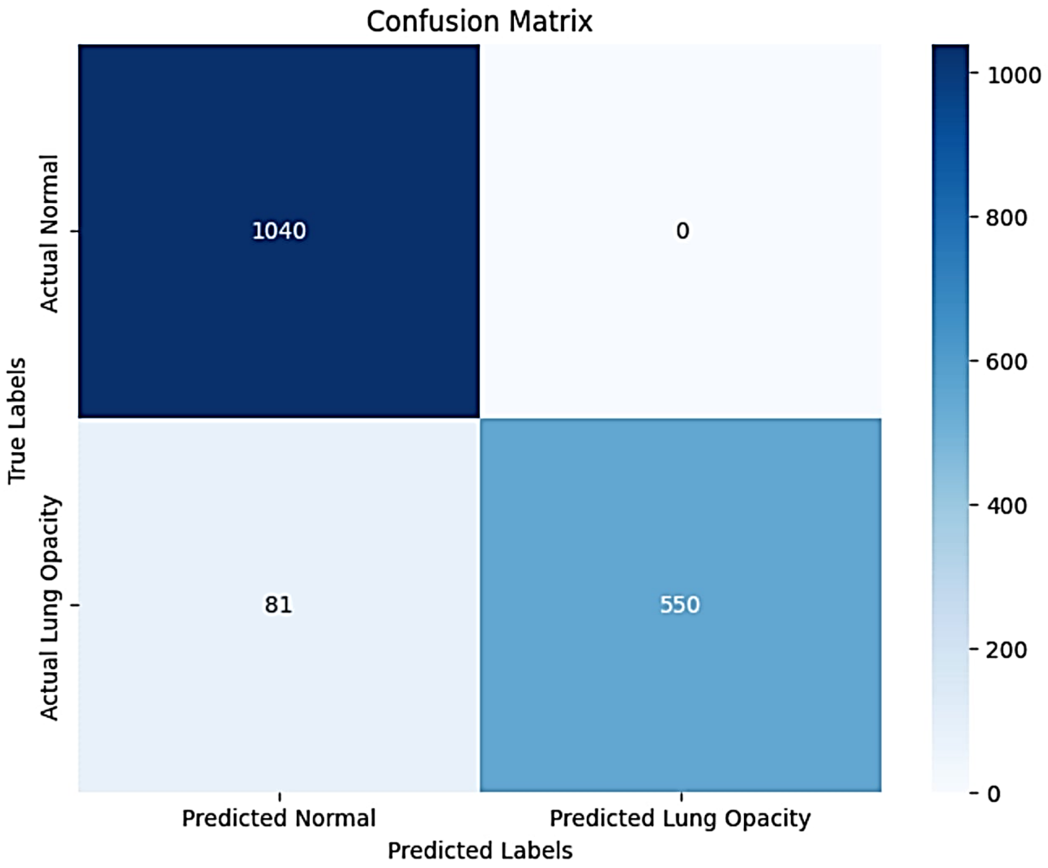


Figure 4.4: Confusion Matrix of ResNet-UNet Model for Pneumonia Detection

It is observed that out of 1,671 total samples, the model shows a strong preference for accuracy in the Normal category, correctly identifying all 1,040 True Negatives with zero false alarms (False Positives). However, it struggles more with sensitivity; while it correctly identified 550 True Positives for lung opacity, it missed 81 cases (False Negatives), labeling them as Normal when they actually contained opacities. This results in an overall accuracy of 95.15%, though the presence of those 81 missed diagnoses suggests the model may need further tuning to ensure it doesn't overlook potentially ill patients.

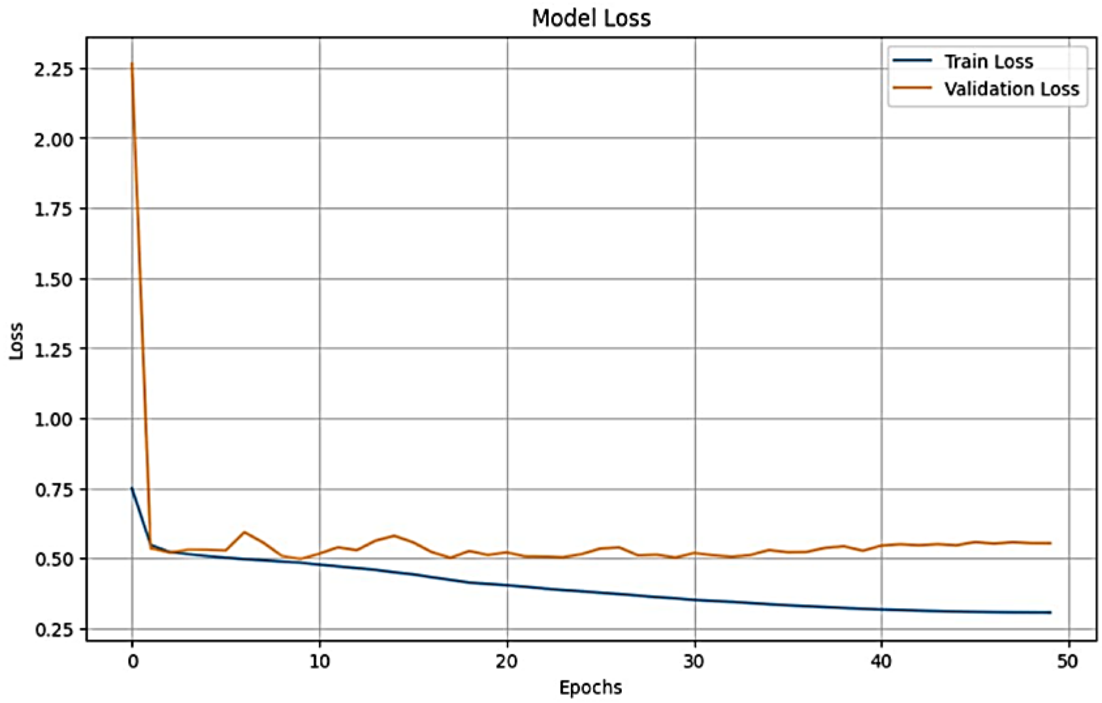


Figure 4.5: Model loss graph of ResNet-UNet Model for Pneumonia Detection

The model loss graph illustrates the training and validation loss curves of the ResNet-UNet model during the training process. It can be observed that the training loss decreases steadily from the initial epochs and begins to stabilize around epoch 20, indicating that the model is effectively learning meaningful features from the chest X-ray images. The validation loss initially shows some fluctuations but gradually stabilizes after approximately epoch 15. Although a slight gap between training and validation loss is visible in later epochs, the difference is not significantly large, suggesting that the model does not suffer from severe overfitting. The smooth convergence of both curves demonstrates stable learning behavior and good generalization capability.

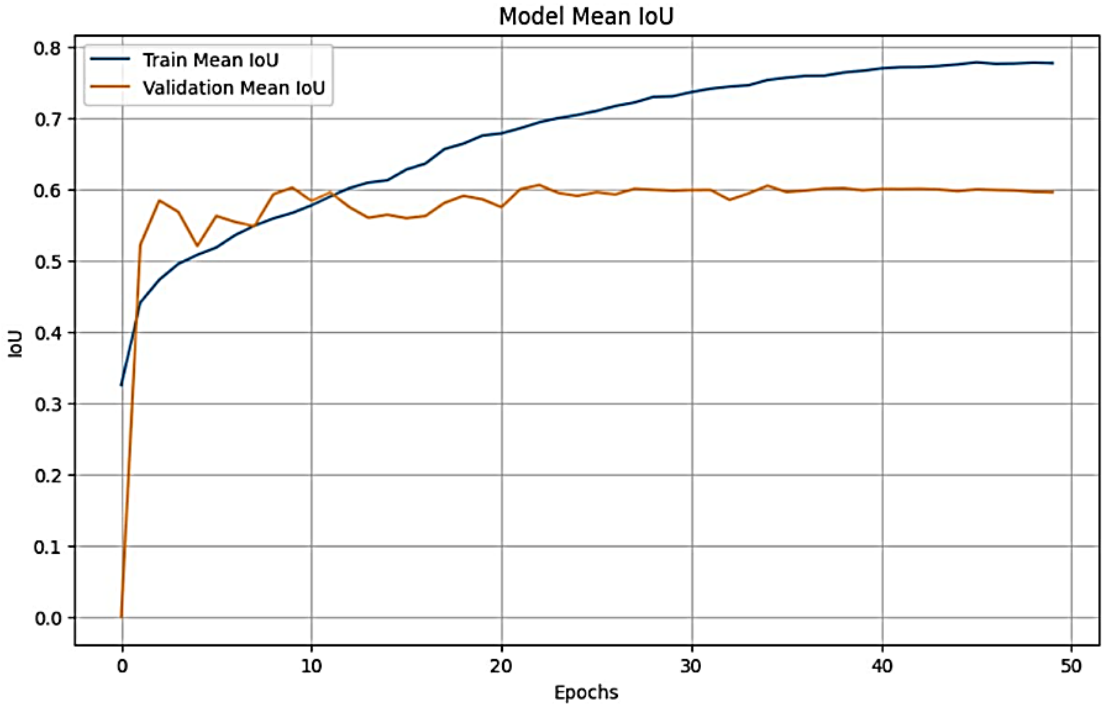


Figure 4.6: Model Mean IoU graph of ResNet-UNet Model for Pneumonia Detection

The Mean Intersection over Union (Mean IoU) performance of the ResNet-UNet model is displayed across training and validation datasets. The training Mean IoU increases consistently with each epoch and shows strong improvement during the early training phase. Stable and good segmentation performance is observed after approximately epoch 25, where both training and validation IoU curves begin to converge and remain nearly constant. The validation Mean IoU also increases in the initial epochs and later stabilizes, maintaining consistent performance throughout the remaining training process. This stability indicates that the model has successfully converged and is capable of accurately localizing infected lung regions.

Table 4.4: Performance Classification of ResNet-UNet for Pneumonia Detection

Model	Metrices			
	Accuracy	Precision	Recall	F1 Score
ResNet-UNet	0.95	0.96	0.95	0.95

The performance classification table shows the performance evaluation of the ResNet-UNet model for pneumonia detection. The model achieved an accuracy of 95%, indicating high overall correctness in classification. With a precision of 96%, the system produces very few false positive predictions, while a recall of 95% demonstrates its strong ability to correctly identify infected cases. The F1-score of 95% confirms a balanced performance between precision and recall. Overall, these results indicate that the model is reliable and suitable for medical diagnosis support.

4.2.5.2 TB Detection Model:

While testing and evaluating the models made for the TB detection system, there were a lot of fluctuation that have been seen in a lot of models. But after testing regularly and repeatedly, ResNet50 model gave much more consistent and better result than ResNet-UNet, DenseNet121 and other models.

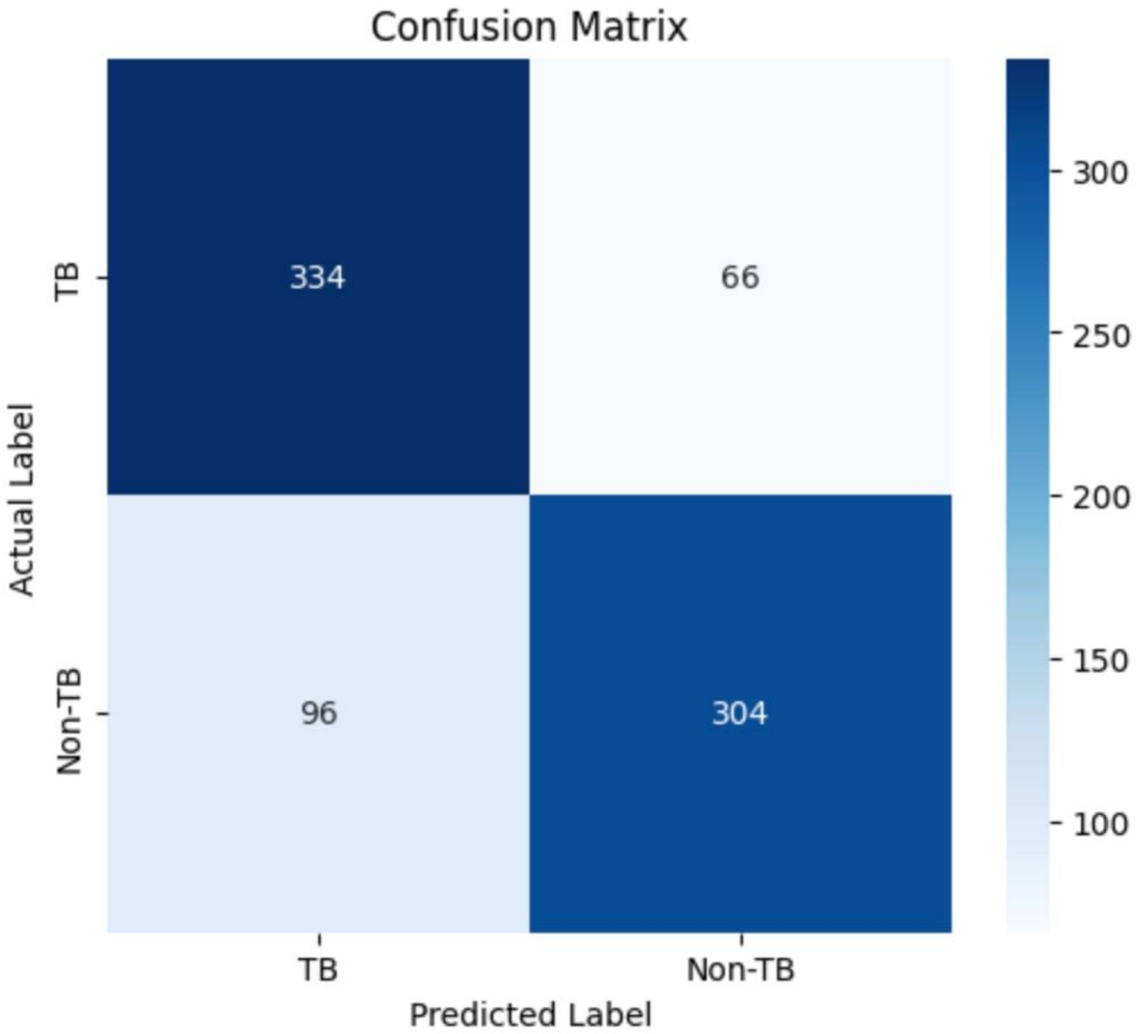


Figure 4.7: Confusion Matrix of ResNet50 Model for TB Detection

This confusion matrix illustrates the performance of a classification model intended to distinguish between TB and Non-TB cases across 800 total samples. The diagonal cells represent correct predictions: the model successfully identified 334 True Positives (actual

TB cases correctly labeled as TB) and 304 True Negatives (actual Non-TB cases correctly labeled as Non-TB), resulting in an overall accuracy of 79.75%. However, the off-diagonal cells reveal specific errors: there are 66 False Negatives, where patients with TB were missed by the model, and 96 False Positives, where healthy individuals were incorrectly flagged as having TB. In a medical context, the 66 False Negatives are particularly critical, as they represent missed diagnoses of a contagious disease, while the 96 False Positives might lead to unnecessary further testing or anxiety for those patients.

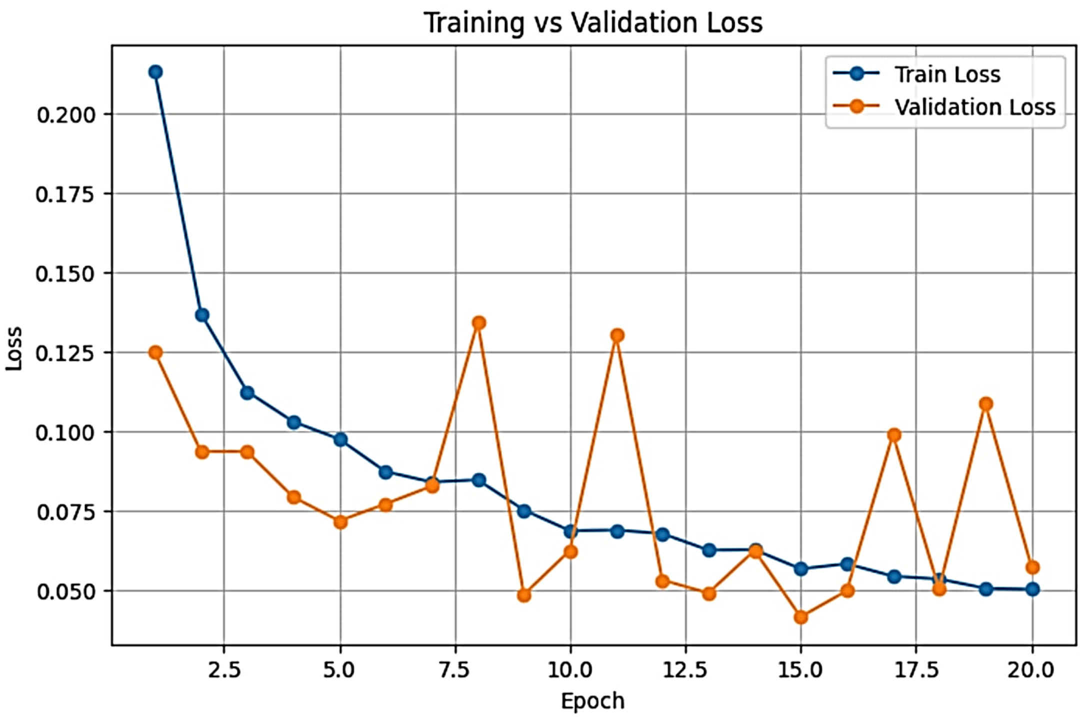


Figure 4.8: Model loss graph of ResNet50 Model for TB Detection

The model loss graph illustrates the training and validation loss curves of the ResNet50 model during the training process. It can be observed that the training loss decreases steadily during the initial epochs, indicating that the model is successfully learning discriminative features from chest X-ray images. After approximately epoch 15–20, the loss curve gradually stabilizes, showing smooth convergence. The validation loss initially exhibits slight fluctuations, which is common in medical image classification due to dataset variability. However, as training progresses, the validation loss also begins to stabilize and follows a trend similar to the training loss. The gap between training and validation loss remains small, suggesting that the model does not suffer from significant overfitting. The smooth and consistent reduction in loss confirms stable learning behavior and good generalization capability of the ResNet50 model for TB detection.

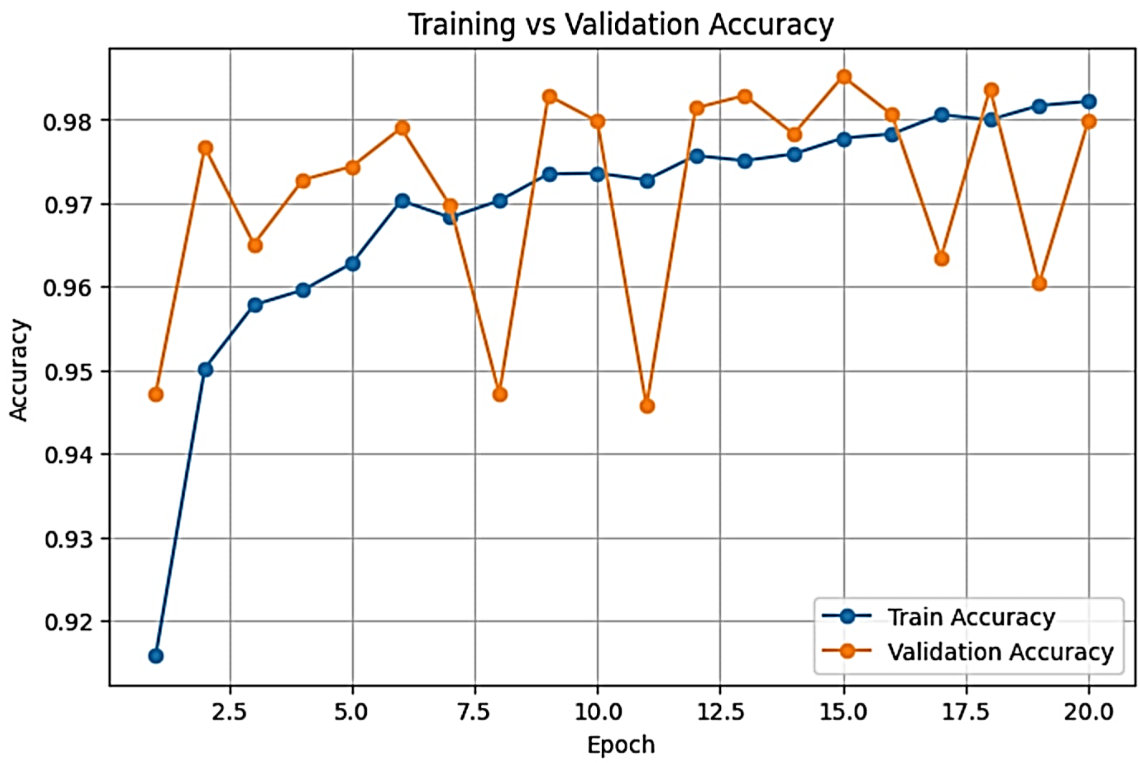


Figure 4.9: Model Mean IoU graph of ResNet50 Model for TB Detection

The Mean Intersection over Union (Mean IoU) graph represents the segmentation/localization performance of the ResNet50 model across training and validation datasets. During the early training phase, the training Mean IoU increases rapidly, indicating that the model is improving its ability to correctly identify and localize TB-infected lung regions. After approximately epoch 20–25, both training and validation IoU curves begin to converge and show minimal fluctuations. The validation IoU follows a similar upward trend and stabilizes at a consistent level, demonstrating reliable performance on unseen data. The close alignment between training and validation IoU curves indicates strong generalization and reduced risk of overfitting.

Table 4.5: Performance Classification of ResNet50 for TB Detection

Model	Metrices			
	Accuracy	Precision	Recall	F1 Score
ResNet50	0.9783	0.9437	0.9954	0.9689

The performance classification table shows the performance evaluation of the ResNet50 model for tuberculosis detection. The model achieved an accuracy of 97.83%, indicating a very high overall correctness in distinguishing between TB and Non-TB cases. With a precision of 94.37%, the system has a low rate of false positive predictions, meaning that when it flags an X-ray as showing TB, it is correct most of the time. The recall of 99.54% is exceptionally high, demonstrating the model's strong ability to correctly identify actual TB-infected cases and minimize missed diagnoses (false negatives). The F1-score of 96.89%, which is the harmonic mean of precision and recall, confirms an excellent and balanced performance between the two metrics. Overall, these results indicate that the model is highly reliable and well-suited for medical diagnosis support, with a particular strength in not missing positive cases of tuberculosis.

4.2.6 User Profile and Security Implementation

In addition to the AI-based disease detection system, the application includes comprehensive user profile management and security features to enhance usability and protect user data:

- **User Profile Management:** Users can view and update personal information, including name and password, ensuring that the account remains accurate and secure.
- **Disease History:** The application maintains a history of previous X-ray analyses, allowing users to track their pulmonary health over time.
- **Secure Login with OTP:** A two-factor authentication (2FA) system using OTP sent via email is implemented to secure the login process and prevent unauthorized access.
- **Integration with Backend and Database:** All user data, including login credentials, profile details, and disease history, are securely stored in the MySQL database. The backend developed in Spring Boot handles login, signup, and update requests while ensuring data integrity and privacy.

These features complement the AI detection models and improve the overall user experience, making the system not only accurate in disease diagnosis but also reliable, secure, and user-friendly.

CHAPTER 5
CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Conclusion

Thus, the project successfully designed and developed a mobile-based lungs disease detection system for the detection and localization of lung diseases Pneumonia and Tuberculosis, using chest X-ray images and deep learning techniques. The system integrates a Flutter-based mobile frontend, a Spring Boot backend service, a MySQL database for data management, and CNN-based artificial intelligence models to deliver an end-to-end diagnostic support solution.

The mobile application provides a structured workflow that includes user registration and authentication, image upload functionality, automated image preprocessing, AI-based disease prediction, probability score generation, and visualization of infected regions through explainable AI techniques. The backend system efficiently handles model inference and data processing, ensuring reliable and timely responses to user requests.

The implemented deep learning models, including ResNet-UNet for pneumonia detection and ResNet50 for tuberculosis classification, demonstrated strong performance in terms of accuracy, precision, recall, and F1-score. Training and validation performance curves confirmed stable convergence and acceptable generalization capability. The inclusion of heatmap-based visualization enhances interpretability, making the system more transparent and suitable for medical decision support.

As a mobile application, the system emphasizes accessibility, usability, and scalability. It is designed to support healthcare professionals by providing rapid preliminary analysis of chest X-ray images, particularly in remote and resource-limited settings where radiological expertise may not always be available. Overall, the project demonstrates the practical integration of artificial intelligence into mobile health technology to improve early detection and assist clinical evaluation.

5.2 Limitations and Future Recommendations

Despite the successful implementation and promising results, the current system has certain limitations that provide opportunities for future enhancement and research.

5.2.1 Limitations

- **Dataset Constraints:** Although the models were trained on publicly available datasets, the diversity of data remains limited. The datasets primarily represent specific populations and may not fully capture the variability in chest X-ray images across different demographics, geographic regions, and imaging equipment.
- **Limited Disease Coverage:** The current system focuses only on pneumonia and tuberculosis detection. Other significant lung conditions such as lung cancer, COVID-19, pneumothorax, and pulmonary fibrosis are not included in the current scope.
- **Image Quality Dependency:** The system's performance is dependent on the quality of input images. Poorly captured X-ray photographs, improper positioning, or low-resolution images may affect diagnostic accuracy.
- **Regulatory Compliance:** The system has not been evaluated against medical device regulations and standards required for clinical deployment.
- **Clinical Validation:** While the models demonstrate strong technical performance, extensive clinical validation with real-world patient data and collaboration with medical professionals is needed before deployment in actual healthcare settings.

5.2.2 Future Recommendations

- **Dataset Expansion:** Future work should focus on curating more diverse and comprehensive datasets that include chest X-rays from multiple healthcare institutions, different populations, and various imaging protocols to improve model generalization.
- **Multi-Disease Detection:** Extend the system to detect a wider range of pulmonary conditions including lung cancer, COVID-19, pneumothorax, pleural effusion, and chronic obstructive pulmonary disease (COPD), transforming the application into a comprehensive chest X-ray analysis tool.

- **Integration with Medical Equipment:** By regular optimization and more upgradation in the systems, this disease detection system can be integrated with the medical equipment like X-ray machines so that with the X-ray image the lungs X-ray report can contain the localization with the disease possibility printed on image.
- **Clinical Collaboration and Validation:** Partner with hospitals, radiology departments, and medical colleges to conduct rigorous clinical validation studies, gather expert feedback, and refine the system based on real-world requirements.
- **Regulatory Approvals:** Work towards obtaining necessary regulatory clearances (like CE marking) to enable clinical deployment as a legitimate medical device.
- **Multilingual Support:** Expand the application's language support to reach non-English speaking populations, particularly in rural and remote areas where language barriers may exist.

REFERENCES

[1] S. Kakchapati, "Tuberculosis in Nepal: Status, Achievement and Challenges," 25 March 2024. [Online]. Available: https://herdint.com/tuberculosis-in-nepal-status-achievement-and-challenges/.

[2] S. Basnet, S. Shrestha, S. Tawanabasu, S. Budathoki, A. Pokhrel, S. Shrestha, Shrestha and Divesh, "Understanding the current childhood pneumonia management practices in Nepal," OUCRU Nepal, 2024. [Online]. Available: https://www.oucru.org/project/understanding-the-current-childhood-pneumonia-management-practices-in-nepal.

[3] S. Jaeger, S. Candemir, S. Antani, Y. Wáng, P. Xue and G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative Imaging in Medicine and Surgery*, vol. 4, no. 6, pp. 475-477, 2014.

[4] S. Candemir, S. Jaeger, K. Palaniappan, R. K. Singh, J. P. Musco, R. K. Singh, Z. Xue, A. Karargyris, S. Antani and G. Thoma, "Lung segmentation in chest radiographs using anatomical atlases with non-rigid registration," *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, pp. 577-590, 2014.

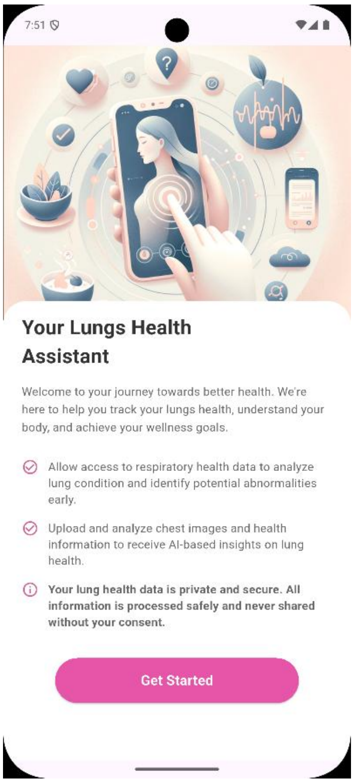
[5] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren and A. Y. Ng, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," *arXiv:1711.05225*, 2017.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

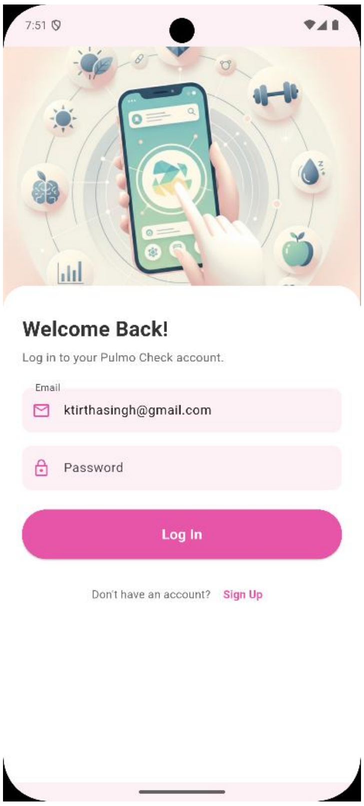
[7] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

APPENDICES

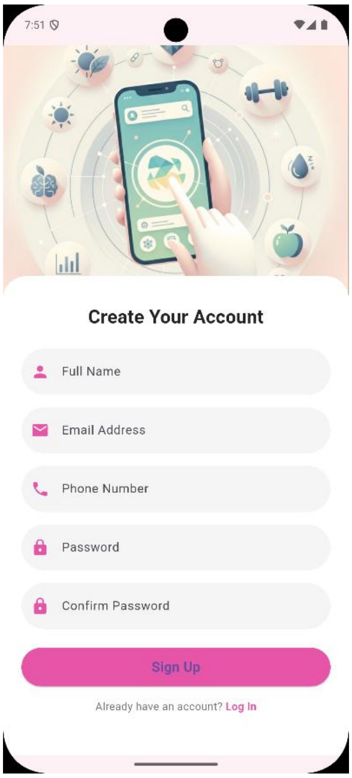
1. Onboarding Screen



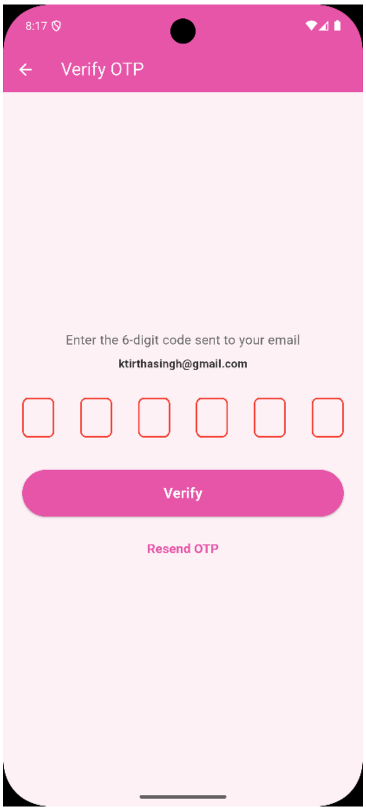
2. Login Screen



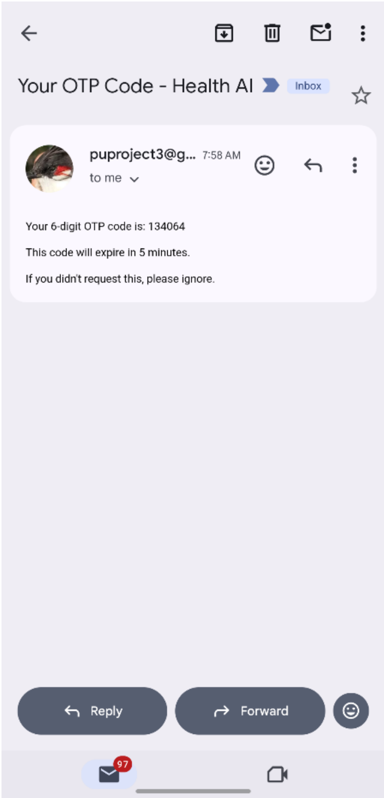
3. Signup Screen



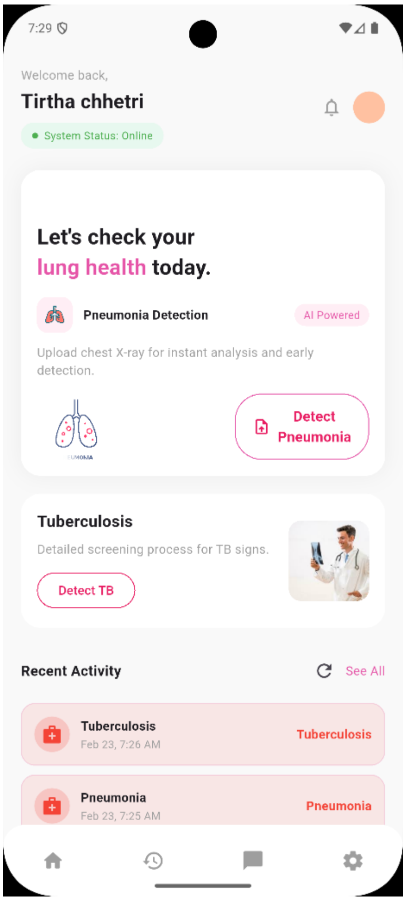
4. OTP Verification Screen



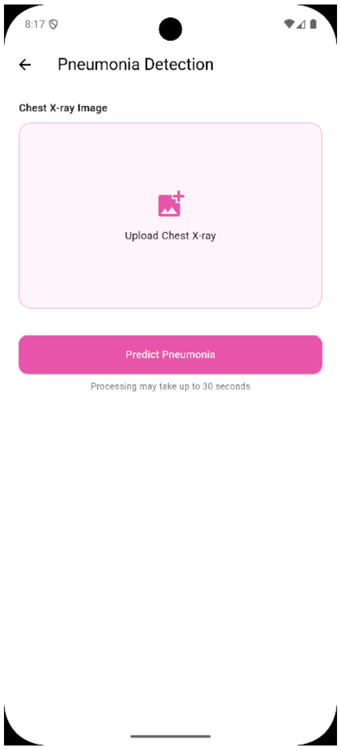
5. OTP Received Screen



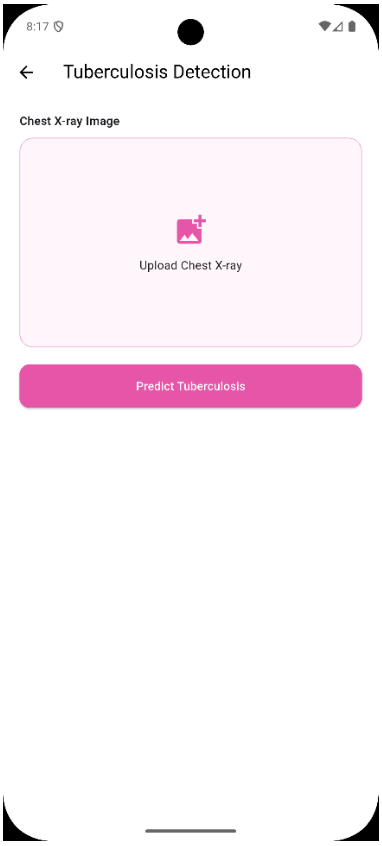
6. Dashboard Screen



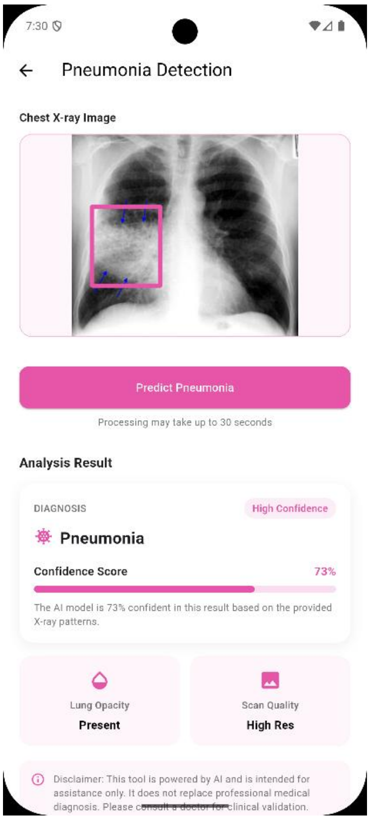
7. X-ray Image Upload Screen for Pneumonia Detection



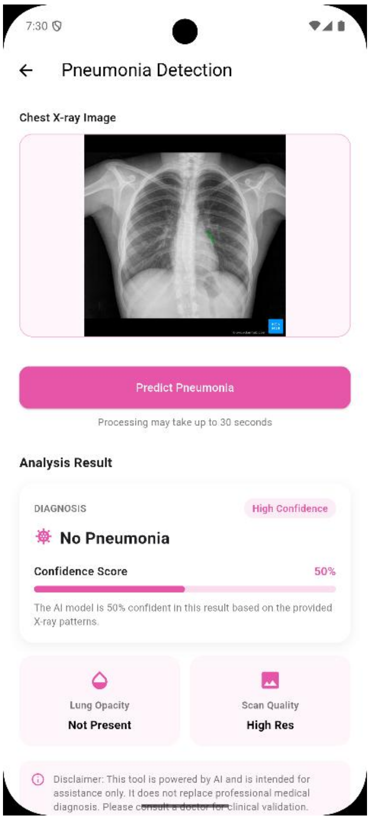
8. X-ray Image Upload Screen for TB Detection



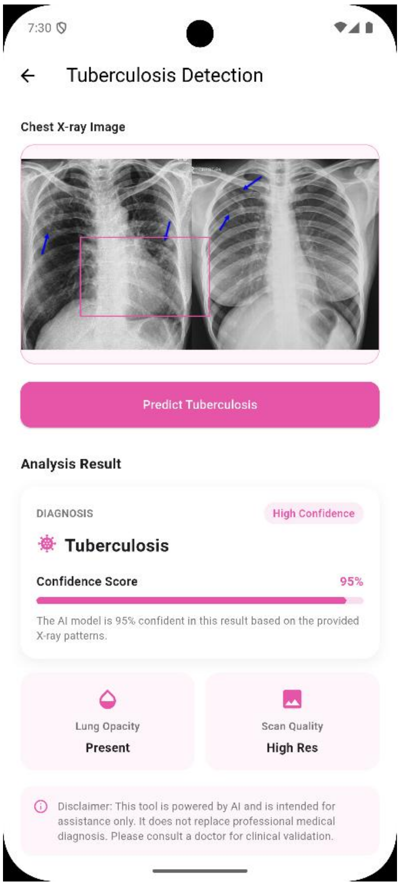
9. Pneumonia Detection



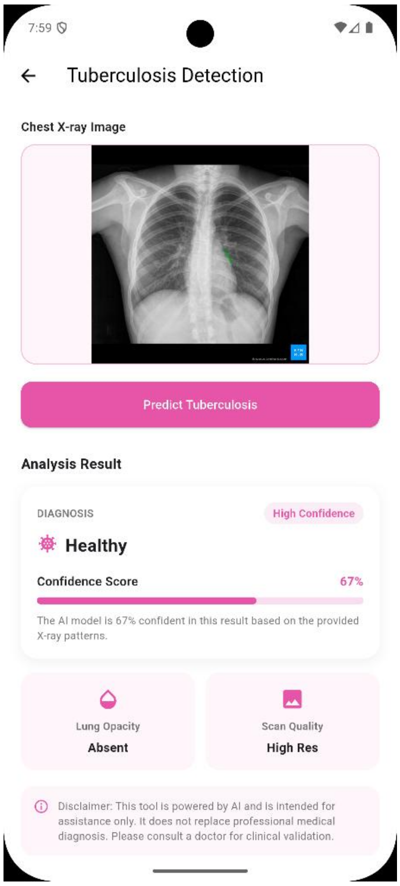
11. No Pneumonia Detection



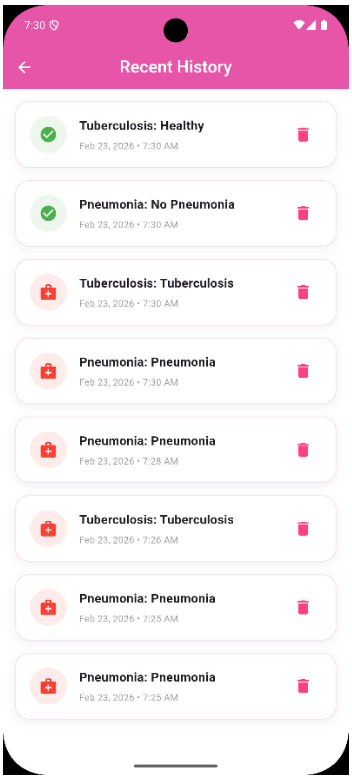
10. TB Detection



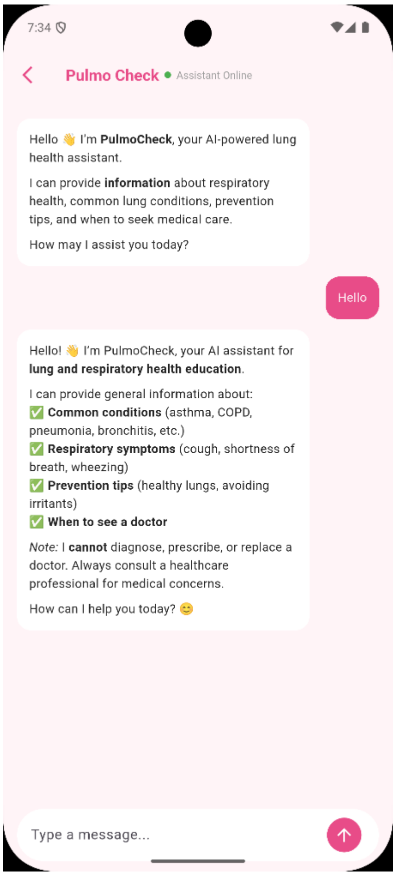
12. Healthy Lungs Detection



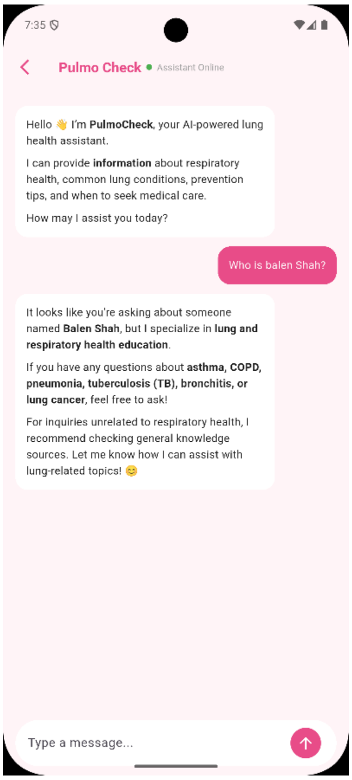
13. Recent History Screen



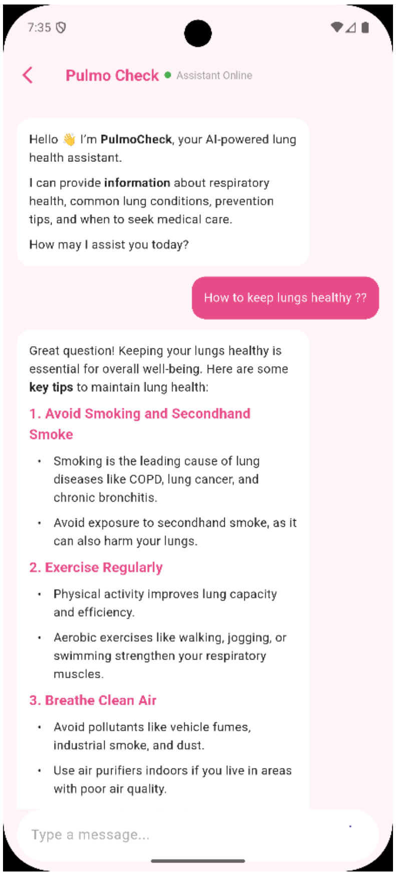
14. Chatbot Response Screen-I



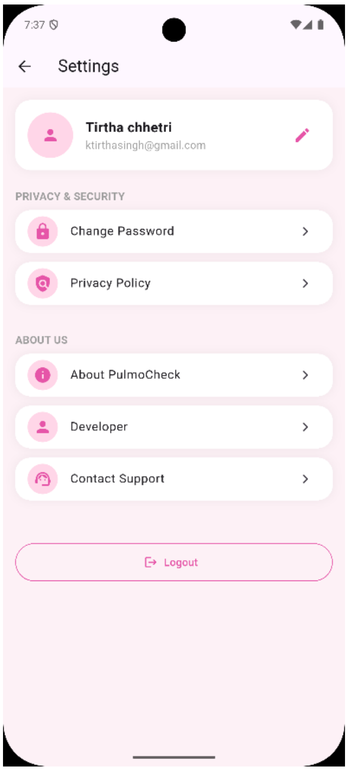
15. Chatbot Response Screen-II



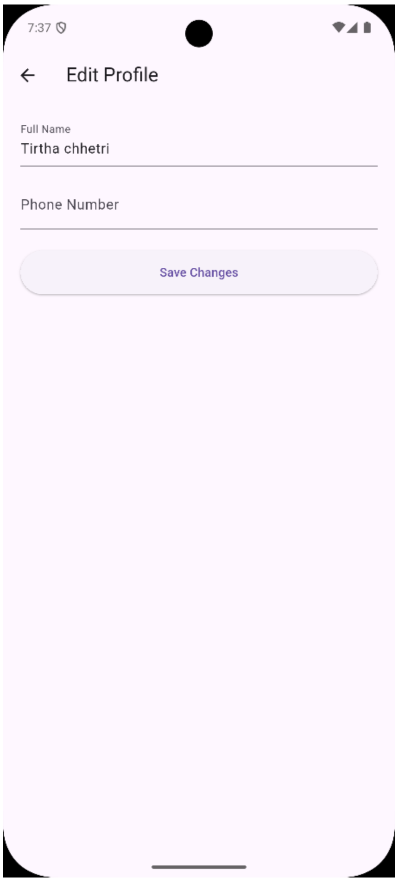
16. Chatbot Response Screen-III



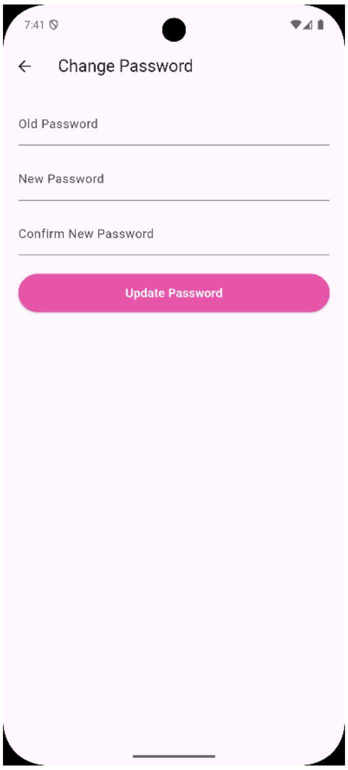
17. Settings Screen



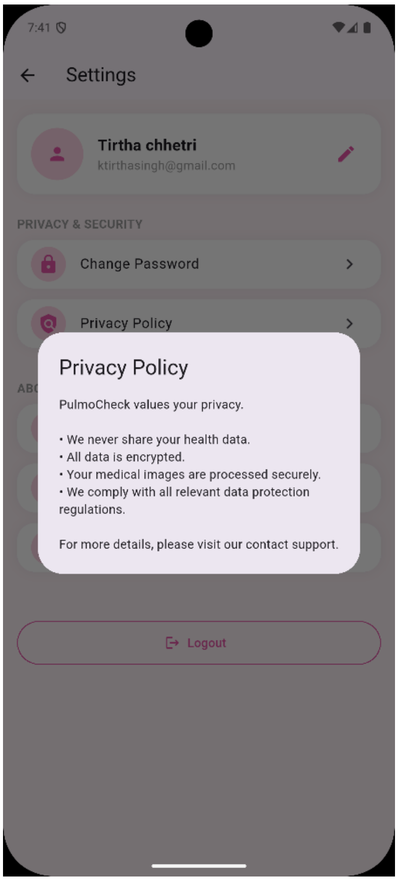
18. Edit Profile Screen



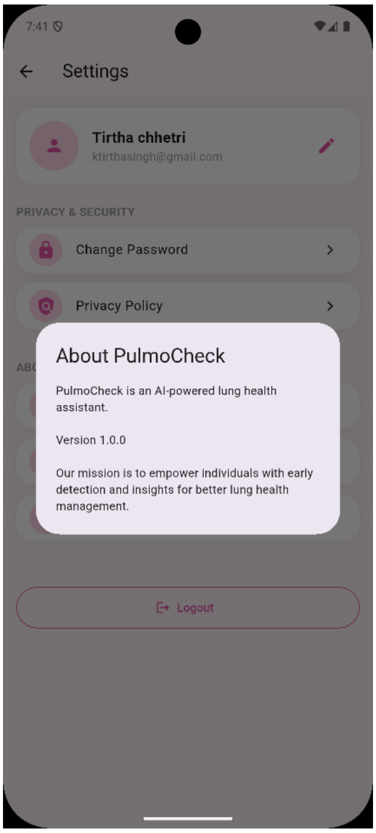
19. Change Password Screen



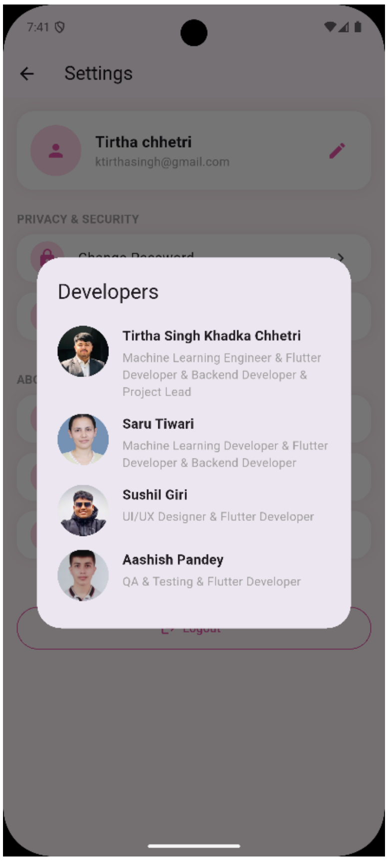
20. Privacy Policy Screen



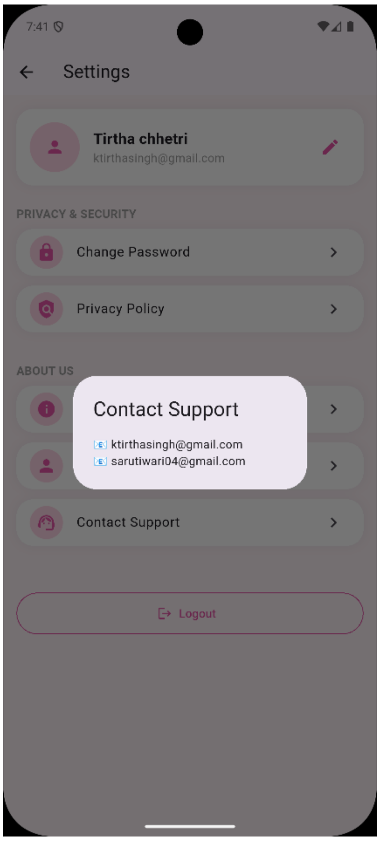
21. About App Screen



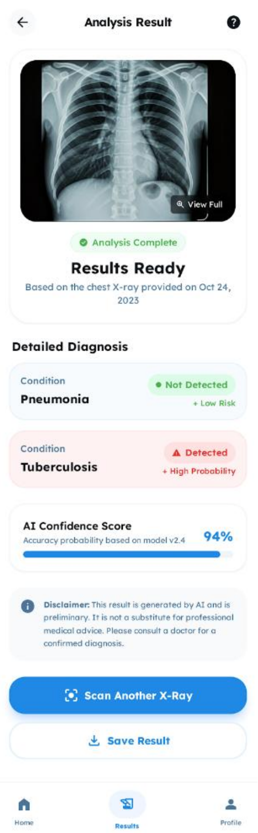
22. Developer Screen



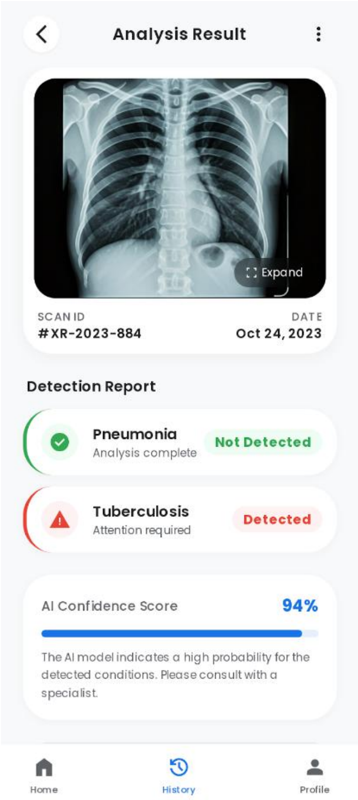
23. Contact Support Screen



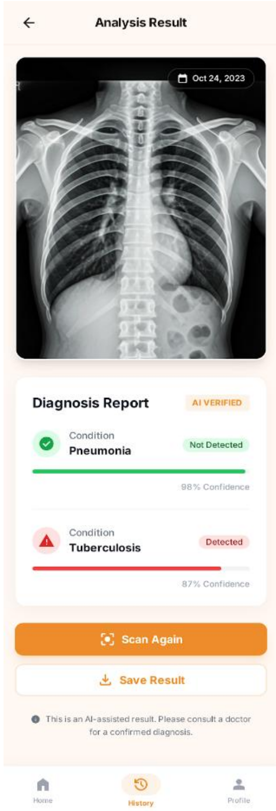
24. Survey Theme 1



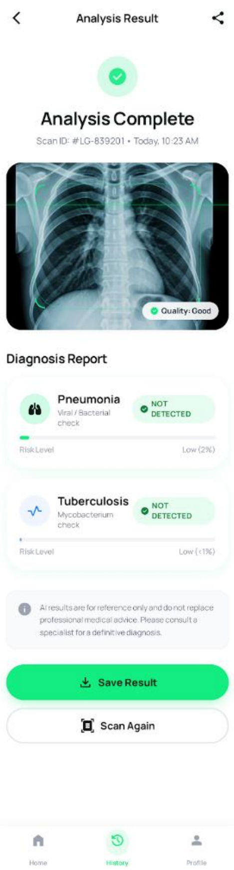
25. Survey Theme 2



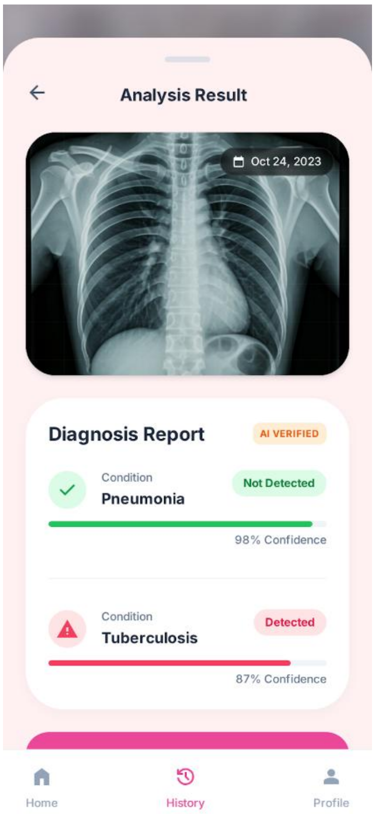
27. Survey Theme 4



26. Survey Theme 3



28. Survey Theme 5



29. Project Work Feedback Form

Pokhara University, School of Engineering

Project Work Feedback Form

Project Title:

Visit Number	Member Name(Only Present)	Date	Feedback	Signature
1	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Jan 25	Integration of chatbot (online)	[signature]
2	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Jan 29	Integration of TB model in app.	[signature]
3	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 04	Finalization of app (online)	[signature]
4	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 12	Full demo process and add login validation	[signature]
5	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 14	Data collection, prepro-cessing & methodology are explained in detail. (online)	[signature]
6	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 15	Language and formatt-ing are clear, making the report easy to understand. (online)	[signature]
7	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 17	Results, analysis and evaluation metrices are presented effectively.	[signature]
8	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 19	Slides are visually clear and well organi-zed. (online)	[signature]
9	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 20	Key points and results are highlighted effectively.	[signature]
10	Ashish Pandey Saru Tiwari Sushil Giri Tirtha Singh Khadka	Feb 22	Diagrams, charts & figures support the explanation well.	[signature]