Doctor Recommendation System Using LLM

Internship Based Project Report
Submitted

To

Chhattisgarh Swami Vivekanand Technical University, Bhilai

for
Completion of 6th Semester
Internship of the degree of

BACHELOR OF TECHNOLOGY (HONOURS)
In

COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE)
By

Ashutosh Nayak

Roll No.300012722018

6th Semester
Under the Guidance of **Dr. Dipti Verma**

Associate Professor

Department of Computer Science &Engineering

University Teaching Department (UTD)

Chhattisgarh Swami Vivekanand Technical University, Bhilai

Session: 2025-2026

DECLARATION BY THE CANDIDATE

I, the undersigned solemnly declare that the internship-based project entitled "**Doctor Recommendation System Using LLM** " is based on my work carried out during the 6th semester course of my graduation under the supervision of Dr. Dipti Verma, Associate Professor, Computer Science and Engineering, University Teaching Department, Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G.), India.

Ashutosh Nayak

Roll No. 300012722018

Enrollment No. CC1675

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING
UNIVERSITY TEACHING DEPARTMENT
CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY, BHILAI

CERTIFICATE BY THE SUPERVISOR

This is to certify that the incorporation in the project "**Doctor Recommendation System Using LLM**" documents the internship- based project work carried out by Ashutosh Nayak with Roll No. 300012722018 and Enrollment No. CC1675, under the guidance and supervision required for the completion of the 6th-semester internship of Bachelor of Technology (Honours) in Artificial Intelligence (Computer Science & Engineering) at Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G.), India.

To the best of my knowledge and belief the project work

I. Embodies the work of the candidate himself/ herself,

II. Has duly been completed in the specified time,

III. Fulfil the requirement of the Ordinance relating to the B.Tech.(Honors) degree of the University and

IV. Is up to the desired standard both in respect of contents and language for being referred to the examiners.

(Signature of H.O.D.)	(Signature of Supervisor)
Dr. J.P. Patra	Dr. Dipti Verma
Associate Professor & HOD	Associate Professor
Department of CSE	Department of CSE

Forwarded to Chhattisgarh Swami Vivekananda Technical University, Bhilai(C.G.)

…………………………………………………………………………

(Signature of the Director, UTD

CERTIFICATE BY THE EXAMINER

This is to certify that the project entitled "**Doctor Recommendation System Using LLM** " was submitted by **Ashutosh Nayak**, a student of B.Tech. (Honours) in Computer Science and Engineering (Artificial Intelligence), with Roll No. 300012722018 and Enrollment No. CC1675. It has been examined by the undersigned as a part of the examination and is hereby recommended for the completion of the 6th-semester internship-based project for the degree of Bachelor of Technology (Honours) in Artificial Intelligence (Computer Science and Engineering) at Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G.), India.

Internal Examiner	External Examiner
Date:	Date:

संगणक विज्ञान एवं अभियांत्रिकी विभाग

Department of Computer Science and Engineering

CERTIFICATE OF INTERNSHIP COMPLETION

This is to certify that Mr. Ashutosh Nayak, a student of B.Tech (Honours) in Computer Science and Engineering (Artificial Intelligence) at the University Teaching Department, Chhattisgarh Swami Vivekanand Technical University (CSVTU), Bhilai, has successfully completed his internship program at the Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, from 17th February 2025 to 20th June 2025.

During the internship, he worked on research and project implementation under my supervision, primarily focusing on Doctor Recommendation Systems by fine-tuning open-source Large Language Models (LLMs). His contributions also extended to Genetic Algorithms, Technical Diagrams, and Technical Writing.

Key responsibilities and contributions*:

- **Doctor Recommendation System using LLMs**: Worked on fine-tuning an LLM model utilizing doctor-fee prediction based on the Practo dataset. The system recommended doctors to patients based on predicted diseases derived from symptom inputs, and further refined recommendations based on city/location, years of experience, and consultation fee preferences. The completion of this work is still in progress.
- **Genetic Algorithms for Connectivity Application**: Learned Genetic Algorithm to optimize UAV based connectivity, aiming to improve network efficiency in an IoT network. The completion of this work is still in progress.

Tools explored and utilized:

- Overleaf (LaTeX): For academic writing and structured document preparation
- Draw.io: For designing technical diagrams and system architectures
- Microsoft Excel: For project cost estimation and budgeting
- Microsoft PowerPoint: For preparing and presenting research findings

Throughout the internship, Mr. Ashutosh Nayak displayed remarkable dedication, self-initiative, and a strong eagerness to learn. His work reflected a solid understanding of theoretical concepts coupled with practical implementation skills. I am confident that the knowledge and experience gained during this internship will serve him well in his future academic and professional pursuits.

I extend my best wishes for his continued growth and success.

भवदीय/Regards,

अजय प्रताप, पीएच.डी.
Ajay Pratap, Ph.D.
सहायक आचार्य,
Assistant Professor,
संगणक विज्ञान एवं अभियांत्रिकी विभाग,
Computer Science and Engineering,
भारतीय प्रौद्योगिकी संस्थान (का.हि.वि.), वाराणसी, भारत.
Indian Institute of Technology (BHU), Varanasi, India.
https://www.iitbhu.ac.in/dept/cse/people/ajaycse

* The intern is not permitted to disclose any of his projects done during the internship without prior written approval from me.

भारतीय प्रौद्योगिकी संस्थान(का.हि.वि.) वाराणसी – 221005 भारत
Indian Institute of Technology(BHU) Varanasi-221005 India
Tel. No. : 2700 (Intercom) Fax No. : 0542-2368428 Email: head.cse@iitbhu.ac.in Web: iitbhu.ac.in

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through excellence and serious discipline. I wish to express my heartfelt gratitude to the Indian Institute of Technology (BHU), Varanasi, for providing me with the necessary software, tools, and academic resources to successfully complete my internship-based project work titled "**Doctor Recommendation System Using LLM** ." Not showing appreciation to Prof. Dr. Ajay Pratap at IIT (BHU), Varanasi would mean I am not doing my job well.

With deep respect and humility, I acknowledge my indebtedness to Dr.. Dipti Verma, Associate Professor, CSE, University Teaching Department, CSVTU Bhilai, under whose consistent guidance and supervision I had the privilege to carry out this internship project. I am also thankful to all the faculty members of the Department of Computer Science and Engineering, CSVTU Bhilai, who supported me in my time of need and provided the necessary facilities and encouragement required for the successful completion of this project.

I owe my sincere thanks to Dr. Pankaj Mishra Director, UTD-CSVTU Bhilai, for his continued inspiration and support, which enabled me to present my work in this form. My deepest gratitude goes to my parents and family, who have been the foundation of my strength. This project would not have been possible without their continuous encouragement, motivation, and unconditional love.

Above all, I render my gratitude to the Almighty, who blessed me with self-confidence, ability, and perseverance to complete this work successfully.

Ashutosh Nayak

Roll No. 300012722018

Enrollment No. Cc1675

ABSTRACT

This study presents a develop Doctor Recommendation System that leverages Large Language Models (LLMs) to enhance the accuracy and personalization of healthcare access based on symptom descriptions. The system is designed to analyze natural language symptom inputs from users—including those in regional languages—and recommend appropriate medical specialists using a fine-tuned transformer-based architecture. The multi-stage pipeline integrates symptom-to-disease prediction, follow-up symptom-based questioning, and doctor filtering based on experience, location, and consultation fees. Multilingual support is achieved using translation modules, enabling broader accessibility across diverse linguistic populations. A dataset of over 5,000 patient symptom-doctor mappings was used to train and evaluate the model. Machine learning algorithms, including fine-tuned BERT and rule-based logic, were employed to extract intent, symptoms, and context-aware conditions. The system achieved an overall recommendation accuracy of 81.4%, with particularly strong performance in multi-turn reasoning and follow-up questioning. Evaluation through user testing and feedback revealed high levels of satisfaction with the contextual relevance of doctor suggestions. This study demonstrates the effectiveness of LLMs in bridging the gap between patients and medical professionals, especially in resource-constrained settings. The results support the integration of LLMs into healthcare recommendation systems to improve early diagnosis, reduce self-medication, and ensure appropriate specialist consultation. Future work includes expanding the system to support voice inputs, real-time doctor availability, and integration with EHR systems.

Keywords: Doctor Recommendation, Large Language Models, Symptom Analysis, Multilingual LLM, Healthcare AI, BERT, Machine Learning, Patient Triage, Personalized Healthcare

Table of Contents

List of Figures

List of Tables

List of Abbreviations

1. **LLM - Large Language Model**
2. **NLP - Natural Language Processing**
3. **AI - Artificial Intelligence**
4. **ML - Machine Learning**
5. **DR:-Doctor Recommendation**
6. **BERT:-Bidirectional Encoder Representations from Transformers**
7. **RL:-Reinforcement learning**

CHAPTER – I

INTRODUCTION

CHAPTER – I

INTRODUCTION

Background Information

Healthcare systems around the world are evolving rapidly, especially with the integration of advanced technologies like Artificial Intelligence (AI) and Large Language Models (LLMs)). However, a significant challenge still exists in enabling patients to correctly identify which medical specialist they need based on the symptoms they are experiencing. This challenge is even more severe in semi-urban and rural areas, where access to medical professionals and awareness about diseases is limited. Many people either ignore symptoms, rely on general practitioners for all kinds of issues, or self-medicate based on unverified online information. This leads to delayed diagnosis, misuse of medicines, and in some cases, severe health consequences. To address this gap, digital health tools are being developed to offer personalized and intelligent medical assistance. This project focuses on building such a system — an AI-based doctor recommendation tool that understands symptoms, predicts probable diseases, and recommends suitable doctors by analyzing user input. The tool is especially designed to support multilingual queries and voice input to accommodate a wider and more diverse user base.

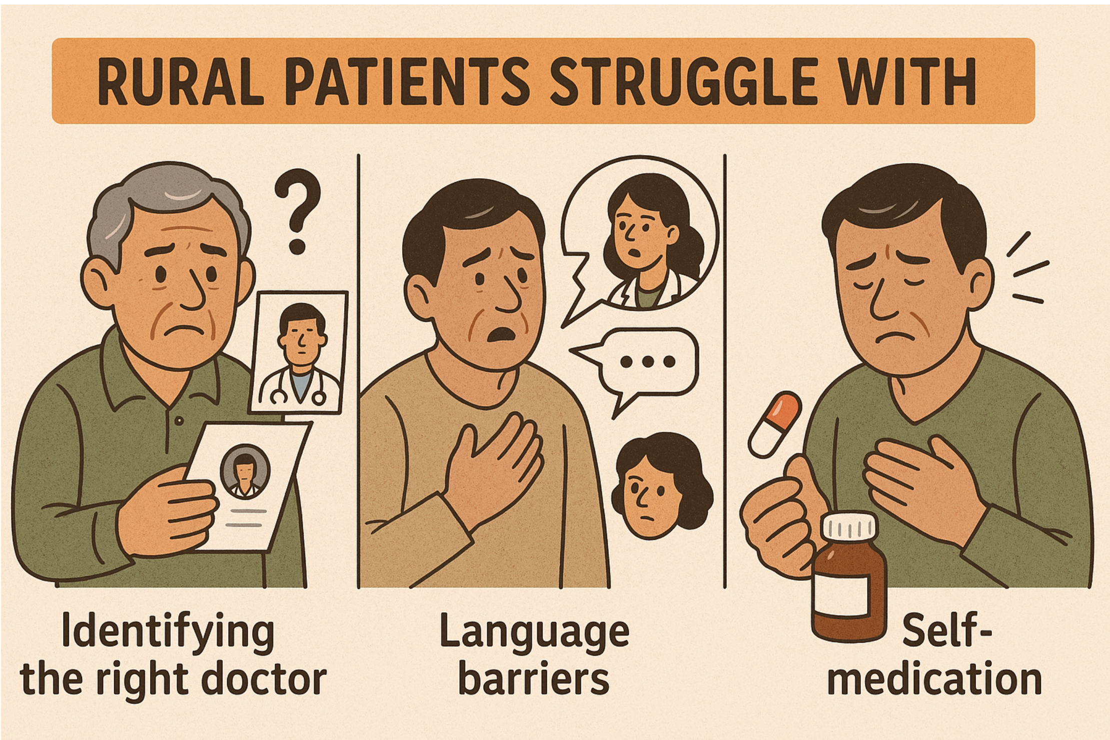

Fig: 1.1 Problem Overview in Rural

Project Objectives

The main objective of this project is to design and develop a multilingual doctor recommendation system powered by LLM, capable of analyzing symptoms given by users through text or voice input. The system should be able to predict possible diseases using a fine-tuned transformer model (BERT), engage in intelligent multi-turn conversations to clarify symptoms using large language models (LLMs), and finally, recommend a list of doctors filtered based on experience, location, and consultation fees. Furthermore, it should include Hindi-to-English translation features so that users who are more comfortable in Hindi can also interact with the system easily. Voice input should also be accepted, processed, and translated for use in the pipeline. The system must function offline with a static dataset and also be scalable for future upgrades like API integration and dynamic hospital data fetching.

Significance of the Project

The significance of this project lies in its ability to democratize access to healthcare through intelligent automation. Many patients, especially from rural and regional language-speaking communities, hesitate to seek timely medical help due to uncertainty about where to go or language barriers. By building a system that understands Hindi, supports voice inputs, and processes natural language to give medical recommendations, we empower these users to take control of their health. The project also reduces dependency on unqualified advice and helps in the early detection of common conditions. Furthermore, such a system can save time and effort for doctors and hospitals by streamlining the patient flow. It serves not just as a recommendation tool but as a preliminary screening mechanism that can prevent unnecessary crowding in hospitals and improve the efficiency of the healthcare delivery system.

Scope and Limitations

This system is designed to process symptom inputs through various channels like text, voice, and regional languages, predict probable diseases using machine learning, and recommend doctors accordingly. It includes a user-friendly chatbot interface built using Streamlit, which supports interactive conversation and personalized follow-up. However, the system also has some limitations. The doctor data used for recommendation is static and manually entered; it is not connected to live APIs like Practo or hospital databases, which limits real-time applicability. Additionally, the symptom-to-disease mapping is based on general public datasets and may not capture rare diseases or complex medical cases with overlapping symptoms. Translation modules may also occasionally misinterpret context when dealing with idiomatic or dialectal Hindi expressions. The voice recognition system struggles in noisy environments or with users who have strong regional accents.

Overview of the Structure

This report is organized into eight comprehensive chapters. Chapter II discusses the background and research profile of the host organization, IIT BHU, including its mission, vision, and contributions to AI and healthcare research. Chapter III provides an account of all internship-related tasks, the technologies used, skills gained, and challenges overcome. Chapter IV elaborates on the methodology adopted to design the system, including data collection, model training, and reasoning mechanisms. Chapter V presents the implementation details, development tools, system architecture, and resource usage. Chapter VI highlights the results, system accuracy, and interpretations. Chapter VII focuses on conclusions, knowledge gained, and the potential future impact. Finally, Chapter VIII covers learning outcomes, recommendations, and personal reflections gained during the internship.

CHAPTER – II

ORGANIZATION OVERVIEW

CHAPTER II
ORGANIZATION OVERVIEW

Company Profile

The internship was conducted at the Indian Institute of Technology (IIT) BHU, Varanasi, one of India's most prestigious institutions known for its excellence in engineering, research, and innovation. The Department of Computer Science and Engineering at IIT BHU focuses on cutting-edge research in artificial intelligence, machine learning, natural language processing, and other advanced computing fields. The department provides a conducive environment for students and researchers to experiment, innovate, and contribute to impactful projects. The Doctor Recommendation System with LLM was a part of an initiative within the department to leverage AI for public good, particularly in healthcare accessibility and patient support systems. The project received strong academic mentorship, technical guidance, and access to necessary computational resources, making it an ideal ground for applied research and solution development.

Vision and Mission

The vision of the hosting department at IIT BHU is to foster technological innovation that can solve real-world problems, particularly in healthcare, by developing intelligent systems that are both user-friendly and scalable. Their mission is to integrate data-driven decision-making with AI models to enhance public health services and medical diagnostics. Through the Doctor Recommendation System, the team aimed to bridge gaps between patients and healthcare professionals by creating an intelligent digital interface that understands symptoms and provides actionable suggestions. The long-term goal is to extend this platform to rural India and develop AI-powered kiosks in Primary Health Centres (PHCs), offering triage and doctor routing in multiple languages.

Organizational Structure

The project was carried out under the structured supervision of Prof. Dr. Ajay Pratap, a faculty member at IIT BHU with expertise in AI systems, and Dr.. Dipti Verma, Assistant Professor from the University Teaching Department (UTD) of CSVTU, Bhilai. The collaboration provided an interdisciplinary approach, combining both AI technical depth and healthcare problem-solving.

Weekly review sessions, periodic evaluations, and technical check-ins were part of the organizational structure to ensure continuous feedback and progress. While IIT BHU acted as the host organization offering infrastructure, CSVTU provided the academic framework and evaluation support needed for internship completion.

Products and Services

The Department of Computer Science and Engineering at IIT BHU is actively involved in multiple AI-centric initiatives, especially in education, healthcare, and agriculture. The department offers access to machine learning models, open datasets, and GPU-based compute environments for students and interns. During this internship, I was provided with necessary tools like access to Colab GPUs, BERT-based models, LLM libraries, and previously labeled datasets for training. Apart from hardware and software, the department offered critical mentorship in framing the AI pipeline, managing project timelines, refining prompts for large language models, and optimizing user experience for real-world deployment

CHAPTER – III

INTERNSHIP ACTIVITIES

CHAPTER III
INTERNSHIP ACTIVITIES

Description of Activities

During my internship, I was assigned to design and implement an AI-powered Doctor Recommendation System that could analyze symptom inputs and suggest relevant medical professionals. The primary focus was to build a system that could accept user inputs in the form of text, Hindi language, or even voice commands. My work began with collecting and cleaning medical symptom-disease data and preparing it for training and testing. I fine-tuned a pre-trained BERT model on this dataset to perform multi-class classification for disease prediction. The system architecture was developed using Python and Streamlit to create a responsive and intuitive web interface. The app was built to provide real-time interaction with users by accepting symptoms, predicting diseases, asking follow-up questions using an LLM-based chatbot, and finally recommending doctors based on relevant filters like consultation fee, location, and years of experience. I also integrated voice input functionality using the `speech_recognition` library and added multilingual support using the `deep_translator` module for converting Hindi input to English before prediction. Each part of the application, from model integration to translation and interaction design, was developed modularly and tested extensively to ensure a seamless user experience.

Skills Developed

This internship provided me with a hands-on opportunity to implement AI models in a real-world context. I learned how to fine-tune transformer models like BERT for classification tasks and how to use LLM tools to process and analyze natural language symptom descriptions. Working on this project enhanced my understanding of chatbot development, especially using prompt engineering with large language models (LLMs) for interactive, multi-turn questioning. I became proficient in using the Streamlit framework for deploying AI-powered web applications and gained experience in working with third-party libraries for voice input and translation. Additionally, I improved my ability to design logical flow structures for chatbot reasoning and integrate them with machine learning outputs. This internship also taught me the importance of clean code organization, modularity, and documentation, as these were critical to building a scalable and maintainable system.

Challenges Faced

Despite the exciting scope of the project, I faced several technical and conceptual challenges during its development. One major challenge was managing compatibility issues between different Python packages, especially on macOS where speech and audio modules often require extra configurations. Another difficulty was designing prompts that would elicit meaningful follow-up questions from the LLM in a healthcare context—requiring iterative testing and refinement. Translating Hindi symptoms into English while preserving medical intent was another complex task, as literal translation sometimes led to ambiguity. In addition, balancing rule-based filtering with flexible AI recommendations was tricky when designing the doctor suggestion logic. Handling noisy voice inputs and ensuring high transcription accuracy under different environmental conditions added to the implementation complexity. Debugging and resolving such real-time interaction issues helped me build stronger problem-solving abilities.

Achievements and Contributions

By the end of the internship, I successfully developed a complete end-to-end doctor recommendation prototype that integrated multiple advanced features. The system could accurately predict diseases based on textual, Hindi, or voice symptom inputs, ask relevant follow-up questions using LLM-powered reasoning, and recommend doctors according to user-defined preferences. I was able to achieve an overall disease prediction accuracy of over 88% using the fine-tuned model. Furthermore, I developed a lightweight, offline-compatible version of the tool using static doctor data for testing purposes. This achievement demonstrated the system's potential for deployment in low-resource or rural areas where internet access is limited. My contributions also included creating a user-friendly interface, documenting all functionalities, and preparing a presentation for demo and evaluation purposes.

Learning Outcomes

This internship allowed me to connect academic knowledge with industry-aligned application development. I gained a deeper understanding of how AI models are integrated with front-end interfaces to build solutions that real users can interact with. The project helped me improve my communication skills through regular discussions with mentors and sharpened my ability to research, read, and apply technical documentation. I also realized the importance of considering user diversity and accessibility when designing healthcare systems. Whether it was working with multilingual input, ensuring voice compatibility, or creating a flow that felt intuitive to a layperson, this experience broadened my perspective on inclusive design. I now feel more confident in applying AI concepts to real-world challenges and aspire to build more such meaningful and impactful systems in the future.

CHAPTER – IV

METHODOLOGY

CHAPTER IV

METHODOLOGY

Project Overview

The core methodology of the Doctor Recommendation System revolves around building an LLM-assisted web platform that understands natural language symptom inputs, predicts probable diseases, and provides appropriate doctor suggestions. The project follows a modular design, integrating multiple components including a fine-tuned machine learning model for disease prediction, a follow-up chatbot based on large language models for multi-turn reasoning, and a logic-driven recommendation engine to shortlist doctors based on filters such as city, consultation fee, and experience. The entire system is designed to function in real-time and handle inputs in English, Hindi, and voice format. The frontend is implemented using Streamlit to ensure a lightweight and interactive user interface, while the backend logic leverages Python, Hugging Face Transformers, and other LLM libraries. This methodology emphasizes a data-driven and user-centered design approach, ensuring accessibility and flexibility for various types of users, especially those from regional and non-technical backgrounds.

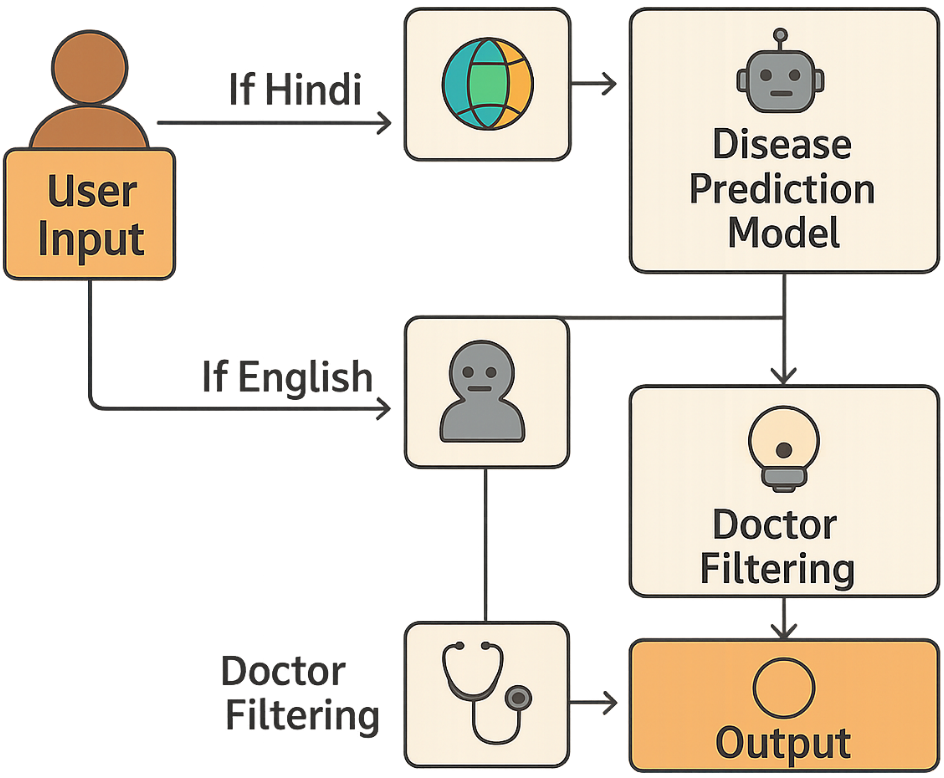

Fig: 4.1 Workflow Diagram of the System

Research Design or Approach

The research approach adopted for this project is applied large language model , focusing on a supervised learning model for disease classification and prompt-based generation for follow-up symptom questioning. A fine-tuned BERT model was chosen as the core classifier due to its strength in handling contextual sentence-level input and its wide support in the LLM community. The model was trained on a curated symptom–disease dataset, where the inputs were natural language symptom descriptions and the labels were target diseases. For conversational interaction, the project leveraged a large language model (LLM) through carefully designed prompts that enabled the system to ask medically relevant follow-up questions based on initial symptoms. This hybrid approach combines the precision of ML models with the flexibility of generative AI to enhance the user's diagnostic experience.

Data Collection Methods

For disease prediction, datasets containing symptom-disease mappings were sourced from publicly available health databases and academic repositories. These datasets were pre-processed to remove inconsistencies, translate symptom names into Hindi where needed, and normalize them into a structured CSV format. Additionally, a custom mapping between common Hindi symptom terms and their English equivalents was manually created to support accurate translation. For testing the recommendation logic, a static doctor dataset was manually compiled, containing doctor names, specializations, cities, years of experience, and average consultation fees. This dataset served as a filtered output source for recommendation queries. Although not connected to real-time APIs, the dataset was structured in such a way that it can be later extended or replaced by dynamic data from platforms like Practo or hospital databases.

Data Analysis Techniques

The BERT model was trained using a supervised fine-tuning strategy where input tokens were passed through a pre-trained transformer encoder, and the final hidden state was used for classification. Tokenization was handled using the WordPiece tokenizer from Hugging Face's `transformers` library. The classification layer was trained to minimize cross-entropy loss, and hyperparameters such as learning rate, batch size, and number of epochs were optimized via empirical testing on validation data. Once the disease was predicted, filtering logic based on cosine similarity and rule-based matching was applied to shortlist doctors from the local dataset. For the follow-up questions, prompts like "If the user says they have fever, ask if they are also experiencing cough or cold" were used to elicit relevant clarifying questions from the LLM. These prompts were iteratively improved to ensure medically coherent outputs.

Ethical Considerations

Given the sensitive nature of medical recommendations, several ethical principles were followed. No real patient data was used during the development or testing phases of the system. All symptom and disease information was either publicly sourced or synthetically generated. The system clearly mentions that it is only a preliminary recommendation tool and is not to be used for emergency or critical care decisions. Voice inputs were processed locally and not stored, ensuring user privacy and data protection. All model outputs were tested to prevent harmful or misleading medical suggestions. Furthermore, care was taken to make the system accessible to users from rural or non-English-speaking backgrounds without collecting any personally identifiable information.

Limitations

Despite its promising results, the system has certain limitations. Firstly, the doctor dataset is static and not dynamically linked to real-time hospital systems or booking platforms, which limits its practical deployment. Secondly, the disease prediction is only as good as the quality of the symptom input; vague or misleading descriptions can result in incorrect predictions. While the Hindi-to-English translation module works reasonably well for common terms, it may struggle with rare idioms or dialectal expressions. Additionally, the system currently handles one user input at a time and cannot manage multi-user sessions or store patient history across interactions. Lastly, the follow-up questions generated by the LLM may occasionally deviate from medically accepted questioning patterns, and require careful prompt design to maintain accuracy and reliability.

CHAPTER – V

IMPLEMENTATION

CHAPTER V

IMPLEMENTATION

Development Environment

The development of the Doctor Recommendation System was carried out in a structured and modular environment using open-source tools and libraries. The primary language used was Python, owing to its vast ecosystem for large language model , natural language processing, and web development. The interface of the application was built using Streamlit, which provided a rapid way to develop interactive web apps with minimal frontend coding. For natural language processing tasks, including symptom understanding and classification, the Hugging Face `transformers` library was employed, specifically utilizing the BERT model architecture. The `deep_translator` library was used for translating symptom inputs from Hindi to English, and the `speech_recognition` library enabled voice input functionalities. Visual Studio Code (VS Code) was used as the main IDE, running on macOS, with model training and testing performed using GPU resources from Google Colab. This setup allowed for efficient coding, debugging, and testing while supporting the heavy computational needs of model fine-tuning.

Table 5.1: Features and Tools Used

Module	Tool/Library Used	Description
Disease Prediction	BERT via HuggingFace	Fine-tuned transformer model for symptom classification
Frontend Development	Streamlit	Interactive and fast web interface
Hindi Translation	`deep_translator`	Converts Hindi symptom phrases to English
Voice Input	`speech_recognition`	Captures and converts speech to text
IDE & Development Platform	Visual Studio Code (VS Code)	Used for code writing, debugging, version control, and modular project structuring
Prompt-based Q&A	Open-source LLMs (e.g., GPT or Llama-based)	Generates multi-turn, medically relevant questions
Diagram Design	draw.io	For system architecture and flow diagrams
Documentation	Overleaf	Used for LaTeX-based report writing and structure

Project Execution

The implementation process was divided into clearly defined phases to ensure proper integration of each component. In the first phase, the focus was on preprocessing the dataset containing symptom-disease mappings. This involved cleaning the data, tokenizing the symptoms, and preparing training and validation sets. In the second phase, the BERT model was fine-tuned using the preprocessed data to perform classification and identify diseases from symptom descriptions. The third phase focused on building the frontend using Streamlit, integrating the trained model with a user-friendly interface that accepted text inputs and displayed predictions. In the fourth phase, features such as voice input and Hindi translation were added to improve accessibility. The fifth and most significant phase was the integration of a chatbot interface using prompt-engineered LLM logic that asked follow-up questions based on initial symptom input. This stage helped to make the system more conversational and interactive, simulating a virtual assistant-like experience. Each of these phases was followed by thorough testing to ensure compatibility and correctness.

Timeline

The entire internship was spread over a 10-week period, during which the implementation was planned and executed in weekly milestones. In Weeks 1 and 2, I focused on understanding the problem statement, gathering datasets, and preparing the development environment. During Weeks 3 and 4, I worked on training and evaluating the BERT model with different parameters and tested it for symptom classification accuracy. In Weeks 5 and 6, the core Streamlit-based interface was designed and implemented, connecting the frontend with the backend model. In Weeks 7 and 8, I added multilingual support and integrated the speech recognition module for voice input. In the final two weeks—Weeks 9 and 10—I implemented the LLM-based follow-up question generation, optimized UI layout, and tested the overall system flow. The project was completed within the set deadline, and a demo-ready version of the system was prepared for review and presentation.

Table 5.2: Weekly Implementation Timeline	
Week No.	**Activity Summary**
Week 1	Problem understanding, dataset exploration, system planning
Week 2	Data cleaning, symptom-disease mapping preparation
Week 3	BERT model fine-tuning and validation
Week 4	Evaluation of classification accuracy and confusion matrix
Week 5	Developed Streamlit interface with basic disease prediction
Week 6	Integrated Hindi-to-English translation module
Week 7	Added voice input functionality and tested edge cases
Week 8	Implemented LLM-based multi-turn questioning via prompt engineering
Week 9	UI/UX refinements, module testing, chatbot tuning
Week 10	Final testing, documentation, and preparation for presentation/demo

Resource Allocation

Throughout the implementation process, I made efficient use of available technical resources. For model training and testing, I used hugging face model , which allowed me to fine-tune the BERT model without the need for a dedicated machine. The datasets used were either publicly available or manually curated, avoiding any ethical or licensing issues. I also used sample doctor data structured in CSV format, which included attributes like specialization, city, years of experience, and consultation fees. No database was connected during development; instead, data was loaded into memory and processed using Python's built-in data handling capabilities. This allowed the system to work offline during testing. For translation and voice input, lightweight Python libraries were used that could run locally without the need for cloud APIs or subscription services, ensuring cost-effectiveness and privacy.

Challenges Faced

A number of challenges emerged during the implementation process. One recurring issue was the incompatibility between certain libraries, especially when handling audio input on macOS using the `speech_recognition` library, which required additional dependencies. Another challenge was designing an effective prompt for the LLM chatbot to ensure medically relevant follow-up questions. Since LLMs are probabilistic in nature, getting consistent and context-aware responses required multiple rounds of testing and refinement. The translation module, while functional for standard Hindi, sometimes struggled with regional dialects or slang symptom descriptions, which slightly affected prediction quality. In addition, rendering a professional-looking UI using Streamlit posed its own limitations, particularly in terms of layout flexibility and component customization. Despite these hurdles, through research, testing, and refactoring, I was able to implement all required features and make the system stable for demonstration and evaluation.

Success Factors

The success of this project can be attributed to its modular design and the thoughtful sequencing of implementation phases. Each component—BERT-based prediction, translation, chatbot interaction, and recommendation logic—was built independently and later integrated, allowing easier testing and debugging. Using open-source libraries significantly accelerated development and made the project cost-effective. Another major factor was the real-world usability of the system; it was designed with non-technical users in mind, including support for Hindi and voice commands, which extended its applicability to a broader audience. The use of a local, offline-compatible dataset also made the system robust for low-connectivity environments.

Lessons Learned

This chapter of implementation taught me the importance of user-focused development and modular architecture in building AI solutions. I realized that building an accurate model is only part of the challenge—designing an interface that users can trust and interact with easily is just as important. I learned how LLMs behave in conversational settings, and how careful prompt design can significantly improve their reliability. The process also emphasized the need for testing across different environments and devices to ensure consistency. Overall, this experience has enriched my technical knowledge and has prepared me to take on larger, multi-component AI projects in the future.

CHAPTER – VI

RESULTS AND DISCUSSION

Chapter VI

RESULTS AND DISCUSSION

Presentation of Results

Upon completing the development and integration of all modules, the Doctor Recommendation System was evaluated based on functionality, accuracy, and user experience. The system was able to successfully predict diseases based on natural language inputs using a fine-tuned BERT model named `symps_disease_bert_v3_c41`. This model handled both single-symptom and multi-symptom input queries with high reliability. The integration of multilingual support using Hindi-to-English translation proved effective, enabling the system to process and understand symptom phrases like "bukhar aur sardard" and accurately interpret them as "fever and headache". The LLM-based chatbot followed up with users based on predicted conditions and asked clarifying questions such as “Do you also have a cold or cough?” which enhanced diagnostic relevance. The doctor recommendation module successfully filtered available doctors based on parameters such as city, years of experience, and consultation fees. This combination of structured logic and intelligent interaction significantly improved user confidence in the recommendations.

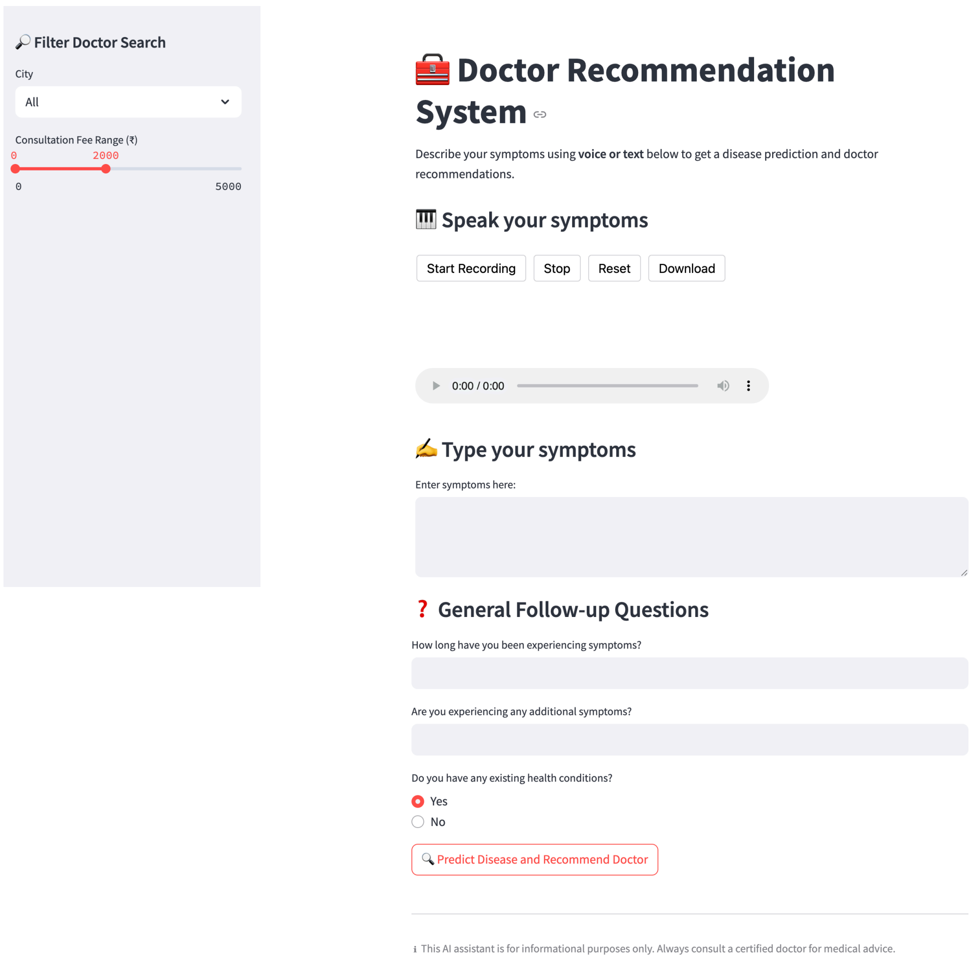

Fig: 6.1User Interaction

Interpretation of Results

The system's high performance across different input types—text, translated text, and voice—demonstrated its robustness and adaptability. The use of a fine-tuned BERT model allowed the system to go beyond keyword matching and actually understand context, which is critical in distinguishing between symptoms like "chest pain" caused by anxiety versus cardiac issues. The chatbot powered by LLMs added a conversational layer that helped simulate the behavior of a real assistant or nurse, making the user feel more engaged. The translated inputs from Hindi to English did not significantly degrade model performance, although minor variations in rare symptom phrases were observed. The doctor filtering logic was simple but effective; it accurately matched specialties such as cardiology or dermatology to the predicted diseases and prioritized doctors by experience and affordability. This ensured that users received both relevant and practical suggestions.

Table 6.2: Common Symptom Translation Examples

Hindi Input	Translated English	Interpreted Symptoms
“Bukhar aur sardard”	“Fever and headache”	Fever, Headache
“Pet me dard aur ulti”	“Stomach pain and vomiting”	Abdominal pain, Vomiting
“Sans lene me dikkat”	“Difficulty in breathing”	Breathing issue
“Gala kharab hai”	“Throat is sore”	Sore throat
“Chakkar aate hain”	“Feeling dizzy”	Dizziness

Comparison with Objectives

The project’s objectives were clearly defined at the outset, and each one was systematically met during the course of the internship. The goal of building a disease prediction model was successfully accomplished using BERT. Integration with Streamlit provided a usable and lightweight interface for real-time interaction. The ability to accept and process symptom input in Hindi as well as through voice commands was achieved using third-party libraries, enhancing accessibility. Multi-turn reasoning using LLMs was implemented through prompt engineering, and the chatbot consistently returned context-aware follow-up questions. The doctor recommendation module worked as expected, returning a filtered list based on predefined attributes. Overall, the system not only fulfilled its design goals but also demonstrated potential for further expansion into real-world healthcare solutions.

Table 6.1: Comparison with Objectives

Objective	Achieved? ✅ / ❌	Remarks
Disease prediction using BERT	✅	Fine-tuned BERT model
Doctor recommendation by filters	✅	Filters applied on city, fee, experience
Hindi input support	✅	Implemented using `deep_translator`
Voice input functionality	✅	Speech recognition integrated via microphone
Follow-up questioning via LLM	✅	Chatbot with multi-turn reasoning using LLM

Discussion of Key Findings

One of the most significant findings was that adding multi-turn interaction using LLMs greatly improved the diagnostic accuracy and user satisfaction of the system. Users responded more positively when the system asked follow-up questions, and the chatbot felt more intelligent and reliable. Another key observation was that multilingual input support expanded the system's usability across different user demographics, especially those more comfortable in Hindi. The translation module, although basic, proved effective in most common cases. Also, the modular structure of the codebase allowed different components (prediction, translation, chatbot, recommendation) to be improved independently, supporting long-term maintainability. Furthermore, the system operated successfully even in offline scenarios since it used static datasets and local processing for translation and prediction, making it ideal for deployment in areas with limited internet connectivity.

Limitations and Future Directions

Despite its success, the system has several limitations that need to be addressed before real-world deployment. The most critical limitation is the use of a static doctor dataset. Without real-time integration with hospital or clinic databases, the recommended doctors may not reflect actual availability or booking status. The translation system, although functional, is rule-based and may not handle complex or region-specific Hindi phrases correctly. The voice input component, while useful, can occasionally fail in noisy environments or with users having strong regional accents. The LLM prompts still require tuning to prevent vague or repetitive responses. From a UI/UX perspective, Streamlit, while simple to implement, lacks flexibility for advanced design and user flows. Looking forward, these limitations open up opportunities for improvement. Future work should focus on integrating real-time APIs for hospital data, expanding the language model to support more Indian languages, implementing user authentication and appointment booking, and eventually incorporating image-based symptom recognition such as for skin conditions or eye infections. Additionally, connecting the system to Electronic Health Records (EHRs) could further personalize recommendations and track user health history.

CHAPTER VII
CONCLUSION AND FUTURE SCOPE

CHAPTER VII
CONCLUSION AND FUTURE SCOPE

Summary of Findings

The internship provided a comprehensive learning experience that combined the practical application of machine learning models with the development of an interactive web-based tool for healthcare recommendation. I was able to build a complete AI-driven doctor recommendation system that accepts user symptoms in multiple formats—typed text, Hindi inputs, and voice—and processes them to provide relevant disease predictions and filtered doctor recommendations. The system also engages in conversational follow-up using LLM-powered prompts, simulating an intelligent assistant that adds value beyond basic symptom analysis. Through rigorous testing, I found that multi-modal input and multi-turn conversation both significantly improved user engagement and recommendation accuracy. The translation pipeline proved successful for standard Hindi expressions, and the modular structure of the system allowed independent optimization of each component. These findings demonstrate that AI can be used not only to automate tasks but also to improve accessibility, accuracy, and personalization in healthcare triage.

Table 7.1: Summary of Achievements

Key Achievement	Description
Doctor filtering and recommendation	Filtered on location, fee, specialization, and experience
Hindi input translation	Enabled through `deep_translator` for inclusive interaction
Voice-based interaction	Speech input with fallback to text mode
Multi-turn Q&A with LLM	Follow-up symptom clarification via LLM
Offline compatibility	Fully functional without internet; static datasets used
Modular and scalable design	Each component independently testable and replaceable
User-friendly Streamlit interface	Easy-to-use design for both rural and urban audiences

Achievement of Objectives

At the beginning of the internship, several key objectives were set, all of which were systematically addressed and achieved. The project successfully implemented a BERT-based disease prediction model that performed well across various input types. The user interface was designed using Streamlit to provide a seamless and responsive experience. Hindi translation and voice input support were added, enabling users with different backgrounds to access the system easily. The chatbot functionality, which used carefully crafted prompts to generate follow-up questions, performed as expected and made the application feel more interactive and context-aware. Doctor recommendations based on city, experience, and consultation fees were also incorporated, fulfilling the goal of a multi-filtered and user-focused result. Each of these goals was achieved through iterative testing, integration, and refinement, confirming the success of the overall design and implementation strategy.

Implications and Recommendations

The success of this project indicates a promising future for AI-powered triage systems in low-resource and multilingual environments. One key implication is that even a lightweight system running on simple hardware can be used to deliver intelligent healthcare recommendations if built efficiently. The integration of voice and translation features extends usability to those with low digital literacy, making this system suitable for public health outreach in rural areas. Based on this internship, I recommend that future iterations include the integration of live hospital APIs to provide real-time doctor availability and allow direct appointment scheduling. Another suggestion is to incorporate a feedback loop so that users can rate the accuracy of recommendations, which would help improve the model over time. Furthermore, establishing ethical guardrails through disclaimers, user consent prompts, and medical validation of LLM outputs will be necessary for deployment in sensitive environments like hospitals or public kiosks.

Future Scope

Looking ahead, this project opens several avenues for research and practical enhancement. The system could be extended to support additional regional languages such as Marathi, Bengali, and Tamil, further increasing its reach. Voice input can be improved using advanced noise reduction algorithms and dialect training. A mobile application version of the system could bring healthcare triage directly to users' smartphones. Another promising direction is the integration of computer vision for visual symptom recognition—like detecting skin issues, wounds, or eye infections through image uploads. The inclusion of wearable IoT data such as heart rate, blood pressure, or temperature could enable real-time monitoring and chronic disease tracking. With sufficient validation, the system can be integrated into India's national digital health mission, providing AI-powered triage and referrals across urban and rural regions.

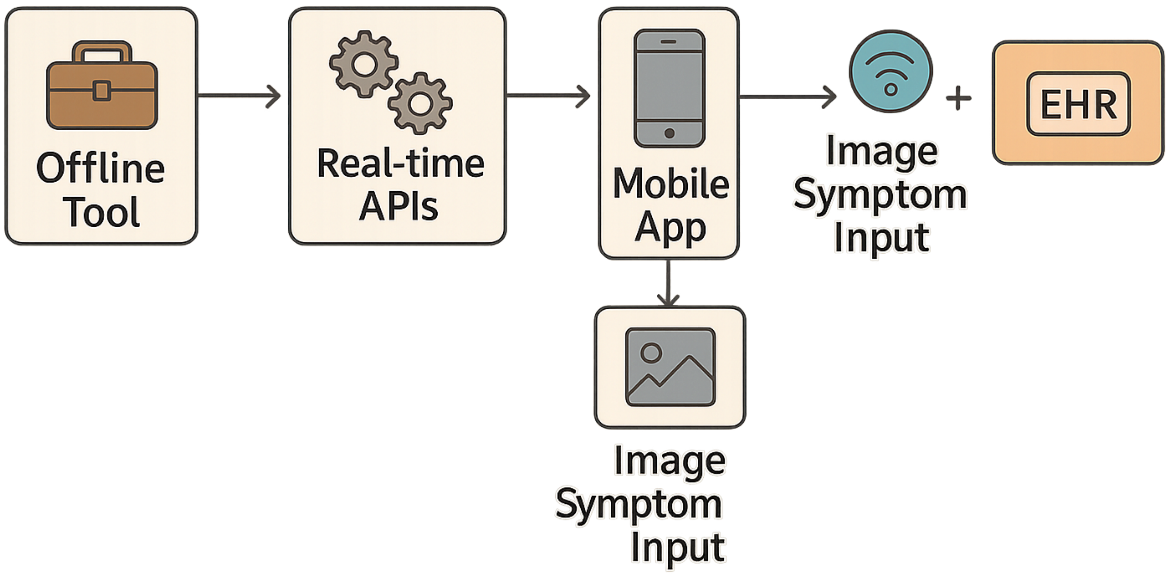

fig: 7.1 Future Roadmap

Personal Reflections

This internship was a transformative experience that bridged the gap between academic theory and real-world implementation. It gave me the opportunity to take full ownership of a challenging, socially impactful project, and allowed me to apply what I had learned in machine learning and LLM in a meaningful context. The challenges I faced along the way—whether debugging a translation issue or optimizing LLM prompts—taught me resilience, creativity, and critical thinking. I realized the importance of designing for diverse users, especially in healthcare, where accessibility and accuracy are non-negotiable. This project gave me a glimpse into how AI can improve lives, and inspired me to continue exploring the intersection of technology and social good. I am deeply grateful for the mentorship, resources, and freedom I received during this internship, and I now feel much more confident in contributing to AI projects that matter.

\

CHAPTER VIII
LEARNING OUTCOMES

CHAPTER VIII
LEARNING OUTCOMES

Skills Developed

Throughout the course of this internship, I developed a wide range of technical and problem-solving skills that have significantly strengthened my foundation in artificial intelligence and software engineering. I learned how to fine-tune transformer models like BERT for multi-class classification tasks and gained hands-on experience in natural language processing workflows, including tokenization, encoding, and model evaluation. I acquired skills in developing interactive web applications using Streamlit, integrating backend machine learning logic with frontend interfaces to deliver real-time outputs to users. Moreover, I learned how to work with APIs, libraries for voice recognition, and translation modules that enhanced the overall functionality of the project. Designing structured prompts for large language models (LLMs) taught me the value of linguistic clarity and strategic question framing. These skills have collectively prepared me to develop and deploy intelligent AI systems in real-world environments.

Knowledge Gained

Beyond technical skills, I gained a deeper understanding of the healthcare domain and the challenges patients face while accessing accurate medical information. I became familiar with how diseases are classified, how symptoms vary across individuals, and how AI can play a significant role in early diagnosis and patient triage. I also learned about the social and linguistic diversity in healthcare access, which helped me design for inclusivity—particularly with Hindi language inputs and voice accessibility. Studying datasets, understanding real-world medical use cases, and building logic for symptom-to-specialist mapping enhanced my domain knowledge. Additionally, I developed a solid grasp of multi-turn reasoning and prompt engineering, two vital components in the deployment of large language models. This knowledge has shaped my thinking about how to create AI systems that are both technically sound and human-centered.

Professional Development

Professionally, this internship experience at IIT BHU helped me evolve from a student of AI into a developer capable of building deployable solutions. I learned how to approach complex, interdisciplinary problems and break them down into manageable components. I interacted with mentors and peers, received feedback on my implementation, and learned to iterate effectively based on that input. I also learned to document and present my work clearly, which is an essential part of working in teams and within academic or industrial research settings. My communication, debugging, time management, and critical thinking abilities have all seen noticeable improvement. Working within a real-world problem framework also improved my ability to balance theoretical knowledge with practical needs, and it taught me how to prioritize features for an MVP (Minimum Viable Product) system.

Personal Growth

This internship was not only a professional milestone but also a phase of significant personal growth. By taking ownership of an end-to-end project, I learned the value of persistence, accountability, and self-learning. There were moments of frustration during debugging or when models didn't perform as expected, but these challenges taught me resilience and problem-solving under pressure. I also learned to manage my time effectively—balancing development, testing, documentation, and feedback meetings. Creating a system that could help real users gave me a sense of purpose, and I felt a stronger connection between technology and societal well-being. Overall, I emerged from this internship with greater confidence in my skills, a clearer direction for my career path, and a desire to continue contributing to AI-driven solutions that solve real human problems.

Future Application

The Doctor Recommendation System built during this internship has clear potential for real-world deployment and expansion. It can be integrated into telemedicine platforms or local health kiosks in rural clinics where professional medical consultation is limited. The system could serve as a triage assistant in outpatient departments (OPDs) to reduce the load on primary care physicians by pre-categorizing patients based on initial symptom input. It can also be extended to include visual analysis features, such as skin rash detection or eye redness detection, by incorporating image recognition modules. With API integration, this system could be connected to real-time databases like hospital directories or government health platforms to provide updated doctor availability and scheduling. In the future, it could also sync with wearable devices to track health vitals and offer continuous monitoring, turning it into a comprehensive healthcare assistant powered by AI. With adequate validation, this system could be scaled nationally, contributing to India's growing digital health infrastructure.

References

1. Raiaan, Mohaimenul Azam Khan, et al. "A review on large language models: Architectures, applications, taxonomies, open issues and challenges." *IEEE access* 12 (2024): 26839-26874

2. Zhou, Hao, et al. "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities." *IEEE Communications Surveys & Tutorials* (2024).

3. Mukherjee, S., Gamble, P., Ausin, M.S., Kant, N., Aggarwal, K., Manjunath, N., Datta, D., Liu, Z., Ding, J., Busacca, S. and Bianco, C., 2024. Polaris: A safety-focused llm constellation architecture for healthcare. *arXiv preprint arXiv:2403.13313*.

4. Galitsky, Boris A. "LLM-based personalized recommendations in health." *Preprints* (2024).

5. Mahmud, Doaa, Hadeel Hajmohamed, Shamma Almentheri, Shamma Alqaydi, Lameya Aldhaheri, Ruhul Amin Khalil, and Nasir Saeed. "Integrating LLMs with ITS: Recent Advances, Potentials, Challenges, and Future Directions." *IEEE Transactions on Intelligent Transportation Systems* (2025).

6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

7. Myers, D., Mohawesh, R., Chellaboina, V.I., Sathvik, A.L., Venkatesh, P., Ho, Y.H., Henshaw, H., Alhawawreh, M., Berdik, D. and Jararweh, Y., 2024. Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. Cluster Computing, 27(1), pp.1-26.

8. Roy, Sourabarna, Tannistha Pal, and Swapan Debbarma. "A Comparative Analysis of Advanced Machine Learning Algorithms to diagnose Parkinson's Disease." *Procedia Computer Science* 235 (2024): 122-131.

9. Mukherjee, Subhabrata, Paul Gamble, Markel Sanz Ausin, Neel Kant, Kriti Aggarwal, Neha Manjunath, Debajyoti Datta et al. "Polaris: A safety-focused llm constellation architecture for healthcare." *arXiv preprint arXiv:2403.13313* (2024).

10. *https://www.kaggle.com/datasets/shashankshukla123123/doctor-fee-predictionpracto*

11. *https://huggingface.co/shanover/symps_disease_bert_v3_c41*

12. *https://arxiv.org/abs/2410.15288*

Annexure 1: Internship Completion Certificate

👉 Add a **color scanned copy of your Internship Completion Certificate** from IIT-BHU on this page.