MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

A Project Report on

Prescription Generation using Natural Language Processing

Submitted by,

Siddesh Paithankar	**(Exam Seat No. 202101040110)**
Ashlesha Shivgan	**(Exam Seat No. 202101040215)**
Mandar Mahabudhe	**(Exam Seat No. 202101040098)**
Snehal Patil	**(Exam Seat No. 202101070096)**

Guided by,

Mr. Pramod Dharmadhikari

A Report submitted to MIT Academy of Engineering, Alandi(D), Pune, An Autonomous Institute Affiliated to Savitribai Phule Pune University in partial fulfillment of the requirements of

THIRD YEAR BACHELOR OF TECHNOLOGY in

Computer Engineering

School of Computer Engineering

MIT Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Alandi (D), Pune – 412105

(2023–2024)

CERTIFICATE

It is hereby certified that the work which is being presented in the Third Year Project Design Report entitled **"Prescription Generation using Natural Language Processing"**, in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Engineeringand submitted to the **School of Computer Engineering** of **MIT Academy of Engineering, Alandi(D), Pune, Affiliated to Savitribai Phule Pune University (SPPU), Pune**, is an authentic record of work carried out during Academic Year **2023–2024** Semester **V**, under the supervision of **Mr. Pramod Dharmadhikari**, **School of Computer Engineering**

Siddesh Paithankar	**(Exam Seat No. 202101040110)**
Ashlesha Shivgan	**(Exam Seat No. 202101040215)**
Mandar Mahabudhe	**(Exam Seat No. 202101040098)**
Snehal Patil	**(Exam Seat No. 202101070096)**

Mr. Pramod Dharmadhikari
Project Advisor

Mr. Amar More
Project Coordinator

Dr.Rajeshwari Goudar
Dean

External Examiner

DECLARATION

We the undersigned solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of **Mr. Pramod Dharmadhikari**.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this Institute/University or any other Institute/University of India or abroad.
3. We have followed the guidelines provided by the Institute in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Siddesh Paithankar	**(Exam Seat No. 202101040110)**
Ashlesha Shivgan	**(Exam Seat No. 202101040215)**
Mandar Mahabudhe	**(Exam Seat No. 202101040098)**
Snehal Patil	**(Exam Seat No. 202101070096)**

Abstract

We all visit doctors in our day-to-day life and know the importance of proper diagnostics and proper prescription of medications. But the fact is, we stay in India and the population here is enormous, and the doctors are very limited in our country. Doctors follow manuals which has information about how to diagnose a patient accordingly, which they have to follow but as we know doctors have a time crunch and cannot spare much time on a single patient so therefor by using the old method of writing it on paper this is not going to work, we need a solution for that if we add a word or little bit about the disease the program can generate the history regarding the disease and it can learn the format that the doctor has to follow and further it can be helpful for the doctor for upcoming patients. Our system aims to deliver that time efficiency and effective prescription needed by the Doctors as well as patient with the help of NLP and NLG.

Acknowledgment

It s a great privilege for us to express our profound gratitude to our respected teacher Mr. Pramod Dharmadhikari, School of Computer engineering Technology, his constant guidance, valuable suggestion, supervision and insertion throughout the course work without which it would have been difficult to complete the work within scheduled time. We are also indebted to the Head of department, school of Computer engineering Technology, MIT academy of engineering for permitting us to pursue the project. We would like to take this opportunity to thank all the respected teachers of this department for being a perennial source of inspiration and showing the right path at the time of necessity.

Siddesh Paithankar 202101040110

Ashlesha Shivgan 202101040215

Mandar Mahabudhe 202101040098

Snehal Patil 202101070096

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In our daily lives, visiting doctors is a common necessity, highlighting the importance of accurate diagnostics and appropriate medication prescriptions. However, in India, with its massive population of approximately 1.393 billion people, there is a significant disparity in the number of doctors available. As per the Indian Medical Registry, there were about 1.301 million registered doctors in 2021. Research indicates that each physician handles between 1,200 and 1,900 patients, with an average primary care physician managing around 2,500 patients. This creates a considerable strain on the healthcare system. While controlling the population size is not feasible, we can enhance prescription methods, which predominantly involve pen and paper, with only a few institutions adopting digital prescriptions with their own software solutions. Our aim is to develop a platform that improves the efficiency and accuracy of prescription practices, benefiting both patients and doctors.

1.2 Motivation

A typical visit to a physician involves inquiries about health issues, such as symptoms and their duration. This is usually followed by a handwritten prescription bearing the clinic and doctor's name. This manual process is time-consuming, particularly when many patients are waiting. Conversations with multiple doctors have confirmed that writing out symptoms and health issues by hand is not only tedious but also insufficiently comprehensive. Consequently, we decided to create a software tool that can quickly, precisely, and efficiently record patient history regarding diseases.

1.3 Project Idea

Our goal is to assist doctors by providing a tool that streamlines patient diagnosis. Diagnosing a patient involves several sequential steps: asking about the disease, past health issues, symptoms, and their duration. Doctors typically follow manuals with diagnostic information, but time constraints often make thorough documentation challenging. Our proposed solution involves a text generator that, with minimal input about the disease, can generate a detailed patient history. This tool would learn the required format and could be continuously updated, aiding doctors by reducing the need for manual writing and ensuring consistency in patient history documentation. The software will be user-friendly and will generate patient history based on prior references.

1.4 Project Challenges

A significant challenge is assembling a comprehensive dataset of all existing diseases in a usable format. The tool needs this dataset to generate accurate patient histories. Additionally, storing patient data without causing contradictions due to symptom similarities is essential. We also need to incorporate the specific language model used by doctors, which requires manual input initially. Implementing machine learning will be critical, as the system must learn from initial user inputs before it can autonomously generate accurate patient histories for subsequent users.

1.5 Proposed Solution

Our proposed solution involves using provided text to generate patient histories and disease details based on a pre-existing database. Initially, machine learning will be employed to train the system to generate accurate histories. As the system learns from the input of initial users, it will start comparing new cases with previously stored data, enhancing its effectiveness. The system will continuously adapt, learning new information as it encounters new diseases, ensuring it remains up-to-date and accurate.

Chapter 2

Literature Review

2.1 Related work And State of the Art (Latest work)

2.1.1 Paper 1: Data Augmentation in Natural Language Processing: A Novel Text Generation Approach for Long and Short Text Classifiers

Authors: Markus Bayer · Marc-Andr´e Kaufhold · Bj¨orn Buchhold · Marcel Keller · J¨org Dallmeyer · Christian Reuter

This paper underscores the critical role of training data in machine learning, particularly through data augmentation to enhance text classifiers. It introduces a novel text generation method designed specifically for NLP, highlighting significant accuracy improvements, especially when data is scarce. The approach involves using algorithms to augment both short and long texts by leveraging language models and document embeddings to generate additional training examples. The paper illustrates the potential of text generation in improving classifier performance while also acknowledging the technique's limitations in certain scenarios.

2.1.2 Paper 2: Long Text Generation via Adversarial Training with Leaked Information

Authors: Jiaxian Guo†, Sidi Lu†, Han Cai†, Weinan Zhang†, Yong Yu†, Jun Wang

This study presents LeakGAN, a framework aimed at addressing the challenges of generating coherent long texts. It overcomes the limitations of traditional Generative Adversarial Networks (GANs), which typically lack guiding signals during text generation. LeakGAN introduces a discriminative network that leaks high-level features to guide the generative network throughout the text creation process. This method includes a MANAGER module that integrates these features, significantly improving both long and short text generation quality. Experiments demonstrate that LeakGAN excels in generating longer sentences, improving BLEU scores and human ratings. The paper suggests future applications in NLP tasks like dialogue systems and image captioning.

2.1.3 Paper 3: A Systematic Literature Review on Text Generation Using Deep Neural Network Models

Authors: NOUREEN FATIMA 1, ALI SHARIQ IMRAN 2, (Member, IEEE), ZENUN KASTRATI 3, SHER MUHAMMAD DAUDPOTA 1, AND ABDULLAH SOOMRO1

This comprehensive review covers advancements in text generation from 2015 to 2021, focusing on deep learning techniques. It highlights the dominance of English text generation in the literature and the importance of text generation across various domains. The review provides valuable insights into data sources, trends, techniques, and applications, emphasizing the surge in deep learning methods since 2018. This resource is essential for researchers and practitioners in AI-driven text generation.

2.1.4 Paper 4: BARTScore: Evaluating Generated Text as Text Generation

Authors: Weizhe Yuan

This paper addresses the challenge of evaluating generated text across various NLP applications like machine translation, summarization, and dialogue. It proposes BARTScore, a metric derived from the BART model, which is a pre-trained sequence-to-sequence model. BARTScore assesses generated text by converting it to and from a reference output or source text, achieving higher scores when the generated text is better. The metric can be applied in an unsupervised manner to evaluate different aspects of text quality, such as informativeness, fluency, and factuality. BARTScore has demonstrated superiority over existing metrics in multiple test settings.

2.1.5 Paper 5: A Survey of Knowledge-Enhanced Text Generation

Authors: Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, Meng Jiang

This survey explores the field of knowledge-enhanced text generation, which aims to improve text generation by integrating additional knowledge beyond the input text. Since the advent of neural encoder-decoder models like Seq2Seq, researchers have sought to incorporate various forms of knowledge to enhance the performance of text generation models. The survey reviews methods and architectures for integrating knowledge, as well as specific techniques and applications. It serves as a comprehensive guide for researchers and practitioners in both academia and industry.

2.2 Limitation of State of the Art techniques

Despite significant progress, current state-of-the-art text generation techniques face several limitations:

Resource Dependence: Advanced text generation techniques often require extensive computational resources and large datasets, posing challenges for those with limited access to such resources. Bias and Ethical Concerns: These models can inherit and perpetuate biases present in the training data, leading to ethical issues. Addressing bias in generated text is crucial for fair and unbiased language generation. Generalization Across Languages: Most research focuses on English, making it challenging to achieve comparable performance in other languages. Interpretable Models: Many advanced models lack interpretability, complicating the understanding and explanation of their outputs, which is critical in certain applications. Contextual Understanding: While these models are good at generating coherent text, they often struggle with deep contextual understanding, particularly in complex or nuanced scenarios. Evaluation Metrics: Developing comprehensive and universally accepted metrics for evaluating generated text remains an ongoing challenge, requiring nuanced approaches that capture various quality aspects. Adversarial Attacks: State-of-the-art models are vulnerable to adversarial attacks, where minor input modifications lead to unintended outputs. Enhancing robustness against such attacks is essential for reliable deployment.

2.3 Discussion and future direction

Achievements and Current Landscape The field of text generation has made remarkable strides, particularly in resource-rich languages. The extensive use of text generation in applications like chatbots and scriptwriting demonstrates its importance. A systematic review of literature from 2015 to 2021 reveals a significant focus on deep learning approaches, particularly in English text generation.

Challenges and Critical Reflection Key challenges include resource dependence, biases in generated text, and the need for effective multilingual models. Ethical considerations and model interpretability further complicate the deployment of advanced text generation techniques.

Future Directions Addressing Bias and Ethical Concerns: Future research should focus on identifying and mitigating biases to ensure fair and unbiased text generation. Multilingual Text Generation: Efforts should be directed towards enhancing text generation techniques for multiple languages, ensuring inclusivity and applicability worldwide. Interpretability and Explainability: Research should aim to improve the interpretability of advanced models, which is crucial for real-world applications. Contextual Understanding and Nuanced Situations: Enhancing models' contextual understanding will improve the accuracy and appropriateness of generated content in diverse scenarios. Robustness Against Adversarial Attacks: Developing strategies to protect against adversarial attacks will ensure the reliable deployment of text generation models. Evolution of Evaluation Metrics: Ongoing research should define comprehensive evaluation metrics that accurately assess various aspects of text quality across different applications.

2.4 Concluding Remarks

The evolution of text generation in AI has seen significant advancements, particularly in resource-rich languages like English. Diverse applications, fueled by the abundance of online text, highlight the technology's importance. A review of recent literature reveals a focus on deep learning approaches and the challenges of achieving artificial general intelligence, ethical considerations, biases, and interpretability issues. Future research should address these challenges, expand multilingual capabilities, and refine evaluation metrics to guide the field forward. This overview underscores the achievements, challenges, and future directions in AI-driven text generation.

Chapter 3

Problem Definition and Scope

3.1 Problem statement

To develop a user interface for doctors to generate prescription using Natural Language Processing as integrated AI model for generation of text.

3.2 Goals and Objectives

Our main objectives focus on transforming the traditional prescription generation process and improving healthcare interactions. Patients often experience repetitive questioning about symptoms during their medical visits, leading to time-consuming handwritten prescriptions. Feedback from several physicians highlighted the inefficiency of pen-and-paper note-taking, which often fails to capture comprehensive patient histories. To address these issues, we aim to automate and enhance the prescription generation process, ensuring accuracy while catering to personalized patient needs. Our goal is to develop software that facilitates swift and precise documentation of patient health issues, optimizing the prescription process for both patients and healthcare providers.

3.3 Scope and Major Constraints

The project focuses on using Natural Language Processing (NLP) and Natural Language Generation (NLG) techniques to generate patient histories based on datasets stored within our database. While the primary focus is on healthcare and prescription generation, the concepts and methodologies can be applied to various other domains, such as report writing, summary generation, and scripting. The core principle involves generating text by analyzing existing data patterns and utilizing formats and information from preceding sentences. This approach is not only applicable in healthcare but also extends to creating concise summaries, detailed reports, and structured scripts across multiple fields.

3.4 Expected Outcomes

Our system's core functionality involves interpreting inputs provided by doctors regarding a patient's disease and symptoms to construct a comprehensive medical history. Initially, the system undergoes a machine learning phase to learn how to generate accurate histories based on these inputs. As the system gathers data from multiple users, it will compare and analyze new user situations against the stored information. Over time, this iterative learning process will enhance its effectiveness in generating accurate histories by leveraging previously learned patterns and insights. The solution is designed to be adaptable and dynamic, continuously learning and adapting to new scenarios. For instance, if new diseases emerge, the system will incorporate this new information, continuously refining and updating its knowledge base to generate comprehensive and contextually relevant medical histories.

Chapter 4

System Requirement Specification

4.1 Overall Description

4.1.1 Hardware Requirements

1)Server/Cloud Hosting- A reliable server or cloud hosting service to deploy and run the application. Platforms such as AWS are suitable for cloud hosting.

2)Processing Power- Multi-core processors to manage concurrent requests and run AI models efficiently. This includes CPUs or GPUs optimized for machine learning workloads.

3)Memory (RAM)- Adequate RAM to handle large datasets and enable quick information retrieval.

4)Storage- Sufficient storage capacity to store medical data, user information, and AI model parameters. SSDs are preferred for faster data access.

5)Network Infrastructure- High-speed and reliable internet connectivity for efficient communication between the server and clients.

4.1.2 Software Requirements

1)Programming Languages- Python: NLP libraries and AI frameworks are commonly implemented in Python. JavaScript/HTML/CSS: For building the front-end interface.

2)NLP Libraries- NLTK (Natural Language Toolkit): A powerful library for working with human language data. spaCy: An open-source library for advanced natural language processing in Python. Transformers (Hugging Face): For pre-trained models and transformer-based architectures.

3)Database- MySQL: To store user data and prescriptions.

4)Web Server- Apache: To serve the web application.

5)Cloud Services -AWS

6)Development Tool- Eclipse IDE

7)Containerization and Orchestration- Docker: To containerize the application

Chapter 5

Proposed Methodology

5.1 System Architecture

The system architecture is structured around achieving multifaceted objectives. At its core, it focuses on automating the generation of prescriptions with heightened accuracy and personalization in a medical context. This framework integrates a specialized NLP model tailored for medical language intricacies, facilitating the generation of prescription text derived from patient information and diagnoses.

A pivotal element of this architecture revolves around user interaction, manifesting through an intuitive interface designed for doctors. This interface encompasses dedicated fields capturing crucial patient data, symptoms, diagnosis, and prescription details, ensuring seamless interaction and information input.

Moreover, the architecture implements personalized features, empowering physicians to tailor prescriptions according to individual patient needs and medical histories. This personalization aspect is augmented by a dosage optimization algorithm, a fundamental component fine-tuned to adjust drug dosages based on patient-specific factors like age, weight, and medical history.

The system architecture further emphasizes real-time updates and sharing mechanisms, facilitating immediate prescription dissemination to patients and pharmacies for prompt medication fulfillment. To ensure reliability and accuracy, a robust testing and validation phase is integrated, encompassing rigorous evaluations and soliciting feedback from medical practitioners to refine and enhance system functionality and precision.

5.1.1 Block diagram/ Proposed System setup

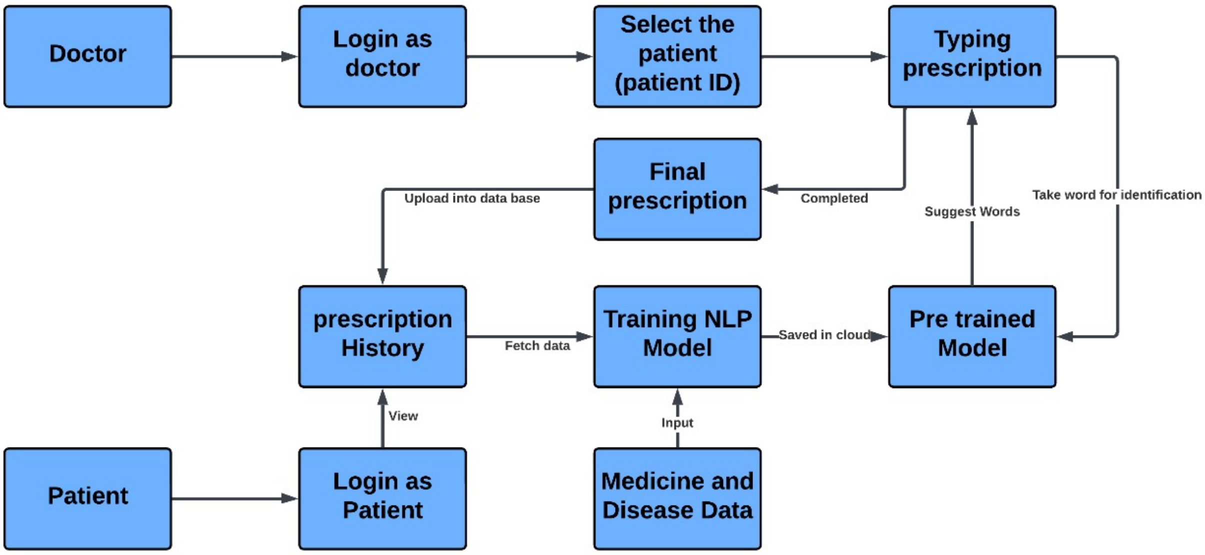

Figure 5.1: Block Diagram

5.1.2 Architecture Diagram

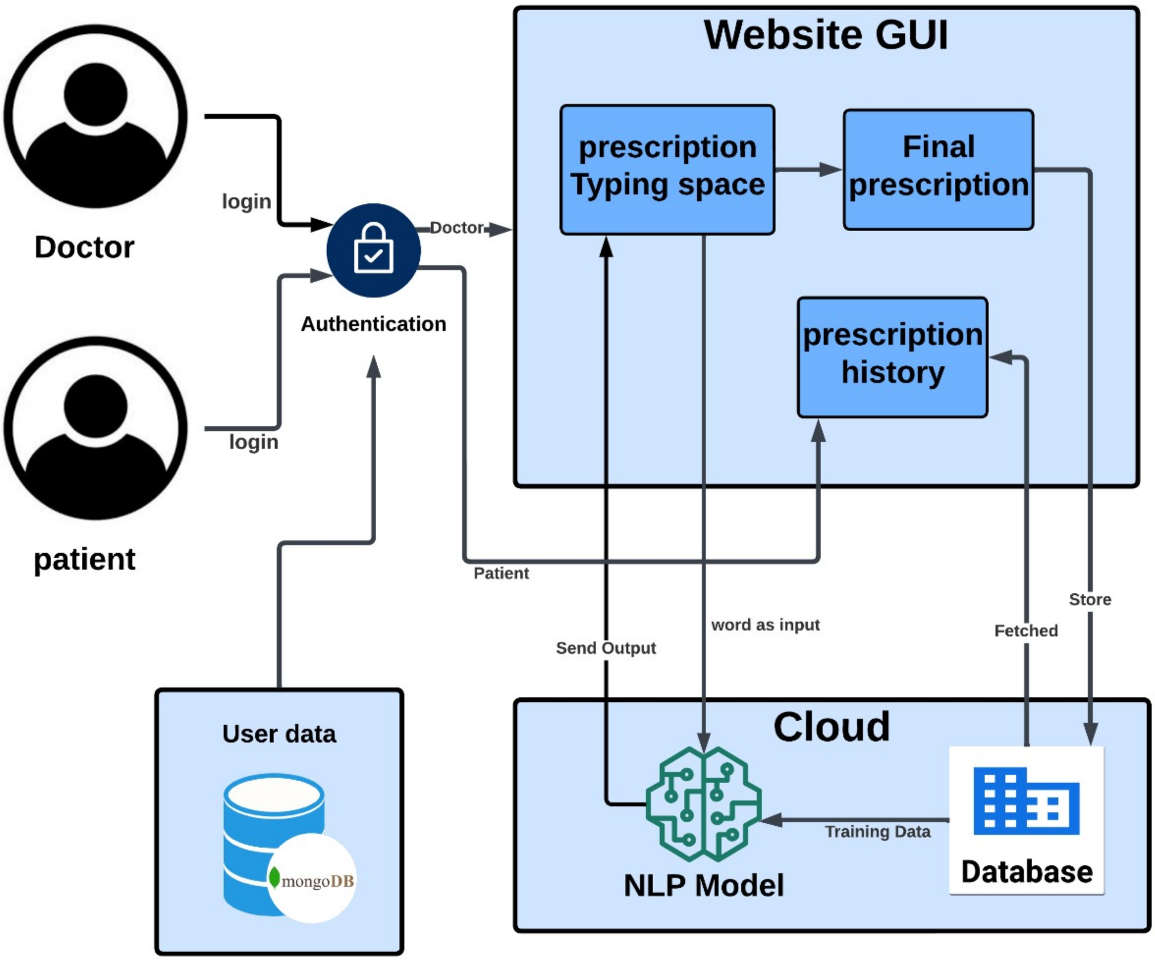

Figure 5.2: Architecture Diagram

5.1.3 Use Case Diagram

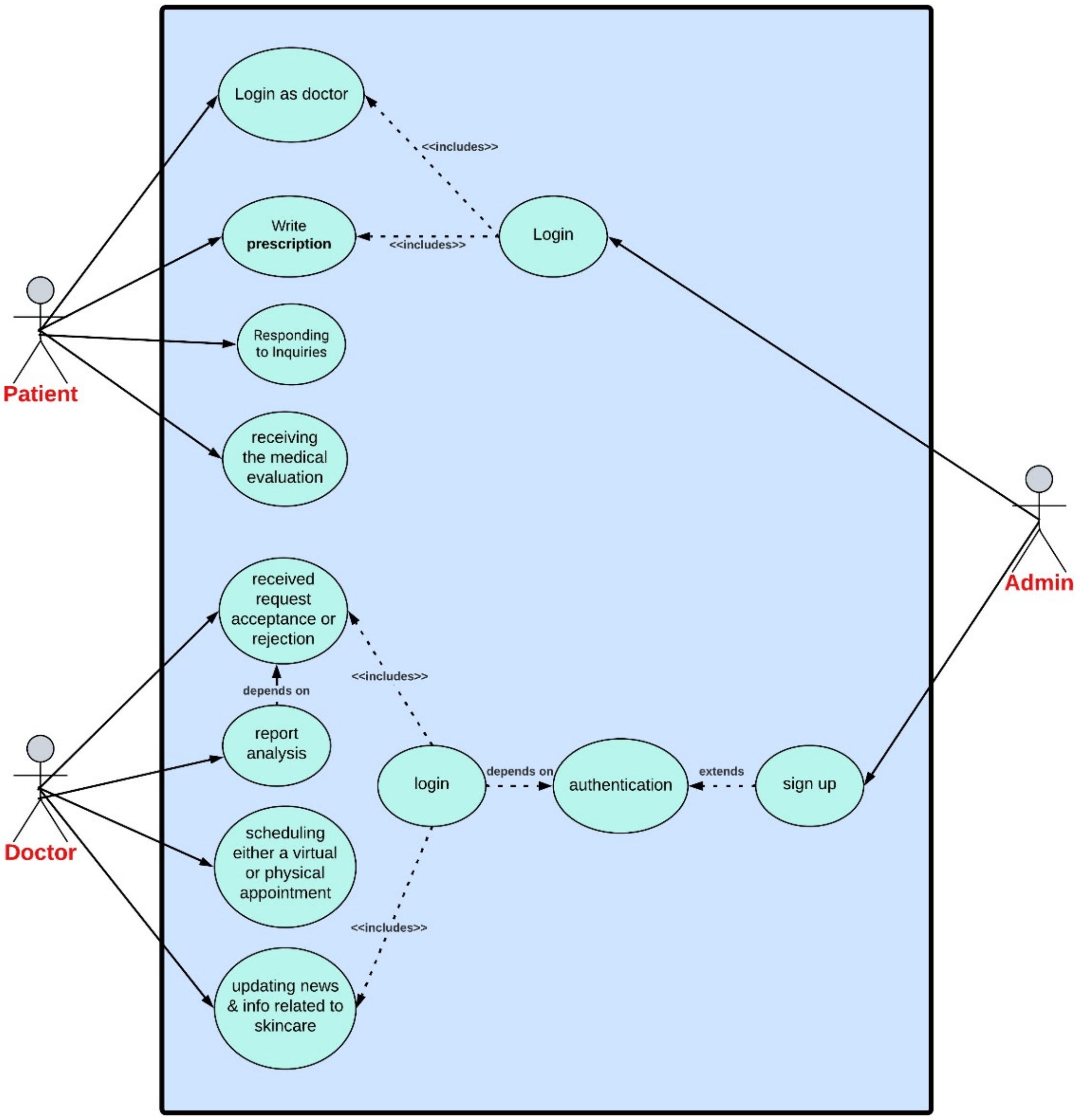

Figure 5.3: Use Case Diagram

5.1.4 Sequence Diagram

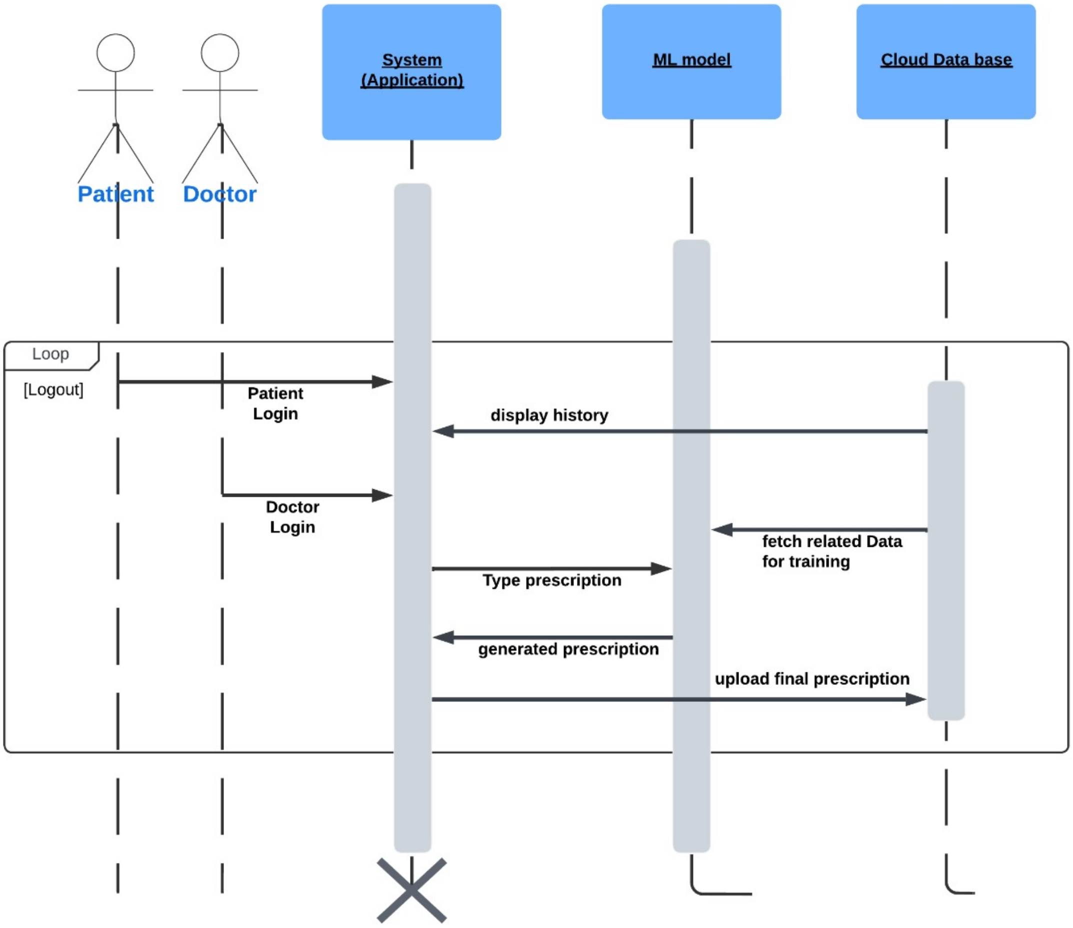

Figure 5.4: sequence Diagram

5.2 Mathematical Modeling

The objective definition outlines a comprehensive set of goals for the project, including automating prescription generation, enhancing accuracy, personalization, drug interaction checking, real-time updates, dosage optimization, and user-friendly interface design. To translate these objectives into mathematical modeling, several key mathematical concepts and formulas come into play.

One fundamental mathematical concept involves developing or selecting suitable NLP models tailored specifically to medical language. These models typically employ sequence-to-sequence architectures, leveraging techniques like attention mechanisms and transformer networks. Mathematical representations of these models include encoder-decoder architectures and attention mechanisms, often depicted using matrices and vectors, such as the self-attention mechanism in transformers:

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{Softmax}\left(\frac{\text{QK}^T}{\sqrt{d_k}}\right)\text{V}$$

Moreover, the dosage optimization algorithm involves mathematical modeling to determine optimal drug dosages based on patient-specific factors. Mathematical functions, possibly involving regression analysis or optimization techniques, could express dosage calculations:

$$\text{Optimal Dosage} = f(\text{Age}, \text{Weight}, \text{Medical History})$$

Additionally, real-time updates and sharing of prescriptions require mathematical concepts related to data transmission and synchronization. These might include concepts from networking theory, data structures for real-time databases, or algorithms for efficient data exchange.

The mathematical modeling will encompass a range of concepts, including statistical methods, machine learning algorithms, optimization techniques, and possibly graph theory for network-related functionalities. By employing these mathematical frameworks, the system aims to achieve its objectives while ensuring accuracy, reliability, and efficient prescription generation and customization.

5.3 Objective Function

The project's objectives encompass a comprehensive scope, aiming to revolutionize prescription generation by embracing automation, precision enhancement, personalization, and real-time updates. Central to this endeavor is the integration of a specialized NLP model, adept at interpreting medical language intricacies to craft prescription texts from patient information and diagnoses.

A pivotal goal involves crafting an intuitive user interface tailored for doctors, facilitating seamless input of patient data, symptoms, diagnoses, and prescription details. Ensuring personalization features within the interface allows doctors to tailor prescriptions according to individual patient needs and medical histories.

Furthermore, the project emphasizes the development of a cutting-edge algorithm dedicated to optimizing drug dosages. This algorithm leverages patient-specific factors like age, weight, and medical history to fine-tune dosage recommendations for enhanced efficacy and safety.

Integral to the system is its capacity for real-time prescription updates, fostering prompt dissemination of prescriptions to patients and pharmacies for efficient medication fulfillment. Rigorous testing and validation methodologies are paramount to ensuring the system's accuracy, reliability, and usability, actively seeking feedback from doctors to refine and enhance its functionalities.

5.4 Algorithm

The objectives outlined in the project encompass various facets of prescription generation, integration of NLP models, user interface design, personalization, dosage optimization, real-time updates, and testing for system validation.

1. Automating Prescription Generation: Develop a system that automates the generation of prescription text based on patient information and diagnoses.

2. Enhancing Accuracy: Ensure the generated prescriptions are accurate by employing reliable algorithms and models.

3. Personalization: Enable customization of prescriptions to cater to individual patient needs and medical histories.

4. Drug Interaction Checking: Implement a system that checks for potential drug interactions to enhance patient safety.

5. Real-Time Updates: Enable instant updates and sharing of prescriptions between doctors, patients, and pharmacies for timely medication dispensation.

6. Dosage Optimization Algorithm: Create an algorithm that optimizes drug dosages considering patient-specific factors like age, weight, and medical history.

The NLP Model Integration involves selecting or developing an NLP model tailored to medical language processing, capable of generating prescription text from patient information and diagnoses. This model should comprehend medical jargon, interpret patient data accurately, and generate contextually appropriate prescriptions.

Algorithm Overview:

1. NLP Prescription Generation Algorithm: Utilizes a sequence-to-sequence model with attention mechanisms to map patient data and diagnoses to prescription text. It employs natural language understanding to interpret and contextualize medical information, generating accurate prescriptions.

2. Personalization Algorithm: Analyzes individual patient histories, extracting relevant information to personalize prescriptions. It applies decision trees or similar techniques to tailor medications based on patient-specific factors.

3. Drug Interaction Checking Algorithm: Implements algorithms that utilize drug databases or knowledge graphs to flag potential drug interactions based on prescribed medications, ensuring patient safety.

4. Dosage Optimization Algorithm: Applies statistical analysis and machine learning to optimize drug dosages, considering patient demographics and medical records, aiming for safe and effective treatment.

5. Real-Time Update Mechanism: Implements a robust real-time communication protocol or API that facilitates immediate sharing of prescription updates among healthcare providers, patients, and pharmacies.

Algorithm:

1. NLP Prescription Generation Algorithm: - Input: Patient data, diagnoses - Process: - Tokenize and preprocess patient data. - Feed data into the pre-trained medical NLP model. - Utilize attention mechanisms to generate prescription text. - Apply post-processing for readability and coherence. - Output: Generated prescription

text.

These algorithms constitute the core functionalities enabling automated, accurate, and personalized prescription generation while considering dosage optimization, drug interaction checking, and real-time updates for improved patient care and safety.

Chapter 6

Implementation

6.1 System Implementation

6.1.1 User Interaction

- **Frontend Interface for Doctors:**
  - Doctors interact with the frontend interface to generate prescriptions.
  - The interface provides guidance and real-time feedback on input data, enhancing user experience and accuracy.

6.1.2 Frontend-to-Backend Communication

- **HTTP Requests:**
  - User inputs, detailing prescription requirements, are sent from the frontend to the backend API.
  - These requests contain essential data for prescription generation.

6.1.3 NLP Preprocessing and Parsing

- **NLP Module:**
  - The backend API processes the user inputs with an NLP module.
  - The module extracts pertinent information such as medication names, dosages, and patient details.

6.1.4 Prescription Generation

- **Backend Engine:**
  - Using the parsed data, the backend engine generates accurate and customized prescriptions.
  - It considers various factors and constraints to ensure prescriptions are appropriate and safe.

6.1.5 Backend-to-Frontend Communication

- **Result Presentation:**
  - The backend sends the generated prescriptions back to the frontend interface.
  - The frontend displays the prescriptions to doctors, who can review and refine them if necessary.

6.2 User Interface

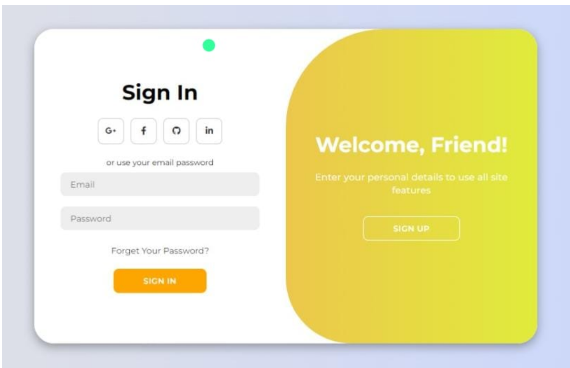

Figure 6.1: admin Sign in Page

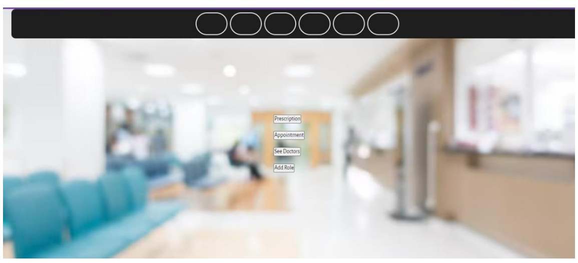

Figure 6.2: admin home page

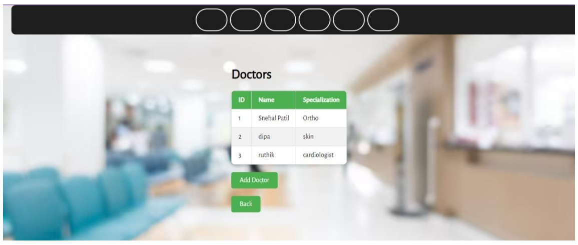

Figure 6.3: doctors(user: admin)

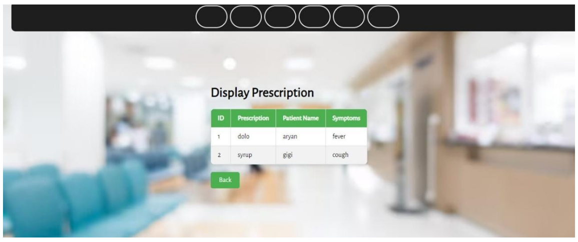

Figure 6.4: prescription(user: admin)

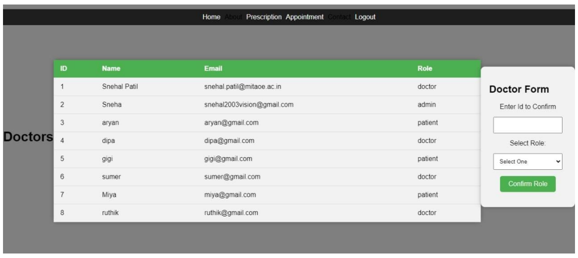

Figure 6.5: registered users(user: admin)

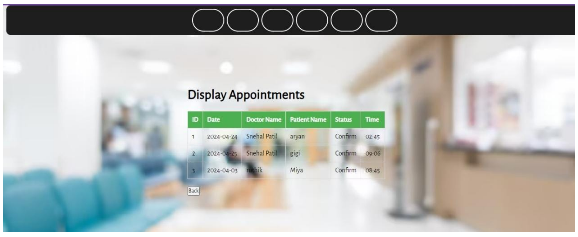

Figure 6.6: appointments(user: admin)

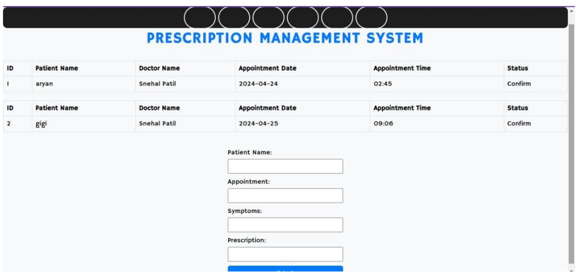

Figure 6.7: prescription(user: doctor)

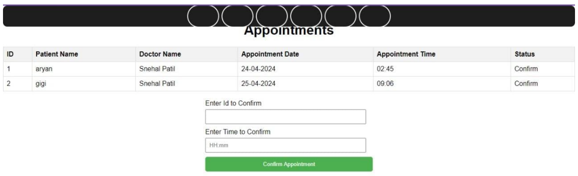

Figure 6.8: appointments(user: doctor)

Figure 6.9: prescription (user: patient

Figure 6.10: form for patient

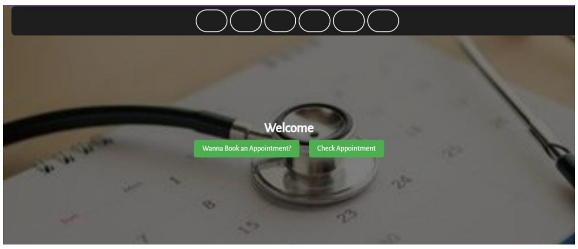

Figure 6.11: patient home page

6.3 Data Description

1. **Network Topology:**

   - Topology defines the arrangement of devices within a network, governing their organization and connections.

2. **Elements:**

   - **Devices:** Represented by circles labeled "System" followed by a numerical identifier.

- **Connections:** Solid lines connecting devices, indicating physical or logical links.

3. **Star Topology Example:**

    - Devices connect directly to a central hub, facilitating communication and data exchange.

6.4 Functional Implementation

6.4.1 Automated Prescription Generation

- **Preprocessing:**
    - The NLP library parses and extracts structured data about prescriptions from unstructured text.
    - It handles nested data and generates a structured JSON representation.

6.4.2 Components

- **NetworkForm Component:** Collects user inputs for network design.
- **handleChange Function:** Updates the component's state with form input changes.
- **handleSubmit Function:** Validates and submits form data.
- **validate Function:** Ensures required fields are completed.

6.5 Prescription System Workflow

6.5.1 Admin Role

- Home page with options to manage appointments, prescriptions, and add doctors.
- Admin can assign roles to users (doctor or patient).

6.5.2 Doctor Role

- View and confirm patient appointments.
- Generate and manage prescriptions.

6.5.3 Patient Role

- Book and check appointment status.
- View prescriptions provided by doctors.

6.6 Tools and Dependencies

6.6.1 Tools

- **Spring Boot Maven Plugin:** For packaging and running the Spring Boot application.

6.6.2 Dependencies

- **Spring Boot Starter Data JPA:** Integration with Spring Data JPA for relational databases.
- **Spring Boot Starter Web:** Support for building RESTful applications using Spring MVC.
- **Spring Cloud Starter Netflix Eureka Client:** Service discovery with Eureka.
- **Tomcat Embed Jasper:** Support for JSP files.
- **Servlet API, Jakarta Servlet JSP JSTL API:** Provides servlet and JSP support.
- **Glassfish JSTL Implementation, MySQL Connector/J:** JSTL and MySQL database connectivity.

- **Spring Boot DevTools:** Development-time features.
- **Spring Boot Starter Test:** Testing support with JUnit, Hamcrest, and Mockito.
- **Spring Cloud Commons:** Common classes and utilities for Spring Cloud projects.
- **Spring Cloud Dependencies:** Manages versions of dependencies for Spring Cloud projects.

6.7 Example Implementation

```
const compromise = require('compromise');
const text = "Prescription for patient with headaches: Ibuprofen 200mg, take twice

let doc = compromise(text);
let medications = doc.match('#Medication').out('array');

console.log(medications);
```

This JavaScript code uses the "compromise" NLP library to extract medication details from the text, demonstrating how NLP can be leveraged in the prescription generation process.

6.8 UI and Integration

- **Frontend:** JSP files for user interface.
- **Backend:** Spring Boot for handling business logic and database interactions.
- **Database:** MySQL for storing user data, appointments, and prescriptions.

6.9 GUI Interface

6.9.1 GUI interface for model made using gradio library

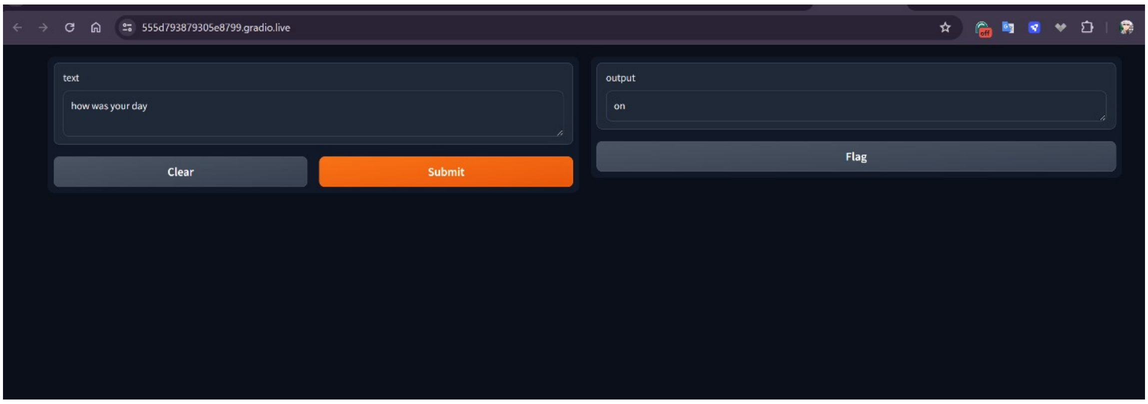

Figure 6.12: gui interface for model made using gradio library

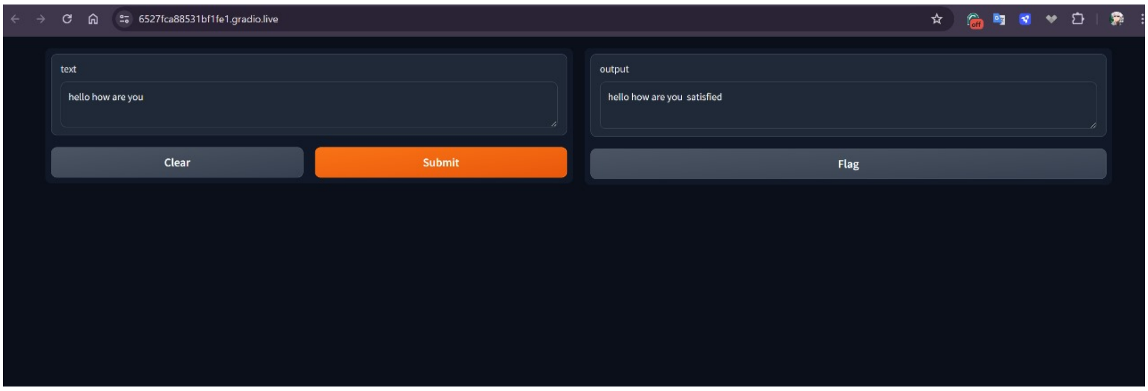

Figure 6.13: gui interface for model made using gradio library

Chapter 7

Conclusion

7.1 Conclusion

In summary, the AI-driven interface for doctors to generate prescriptions using Natural Language Processing (NLP) represents a significant innovation in medical practice. By leveraging advanced NLP technologies, the system effectively addresses key challenges such as accuracy in prescription generation, personalization based on patient history, and efficient documentation.

The integration of a specialized NLP model tailored for medical language ensures precise and contextually relevant prescription text generation. Features like patient-specific customization and dosage optimization algorithms enhance the personalization of prescriptions, taking into account individual factors such as age, weight, and medical history.

The platform emphasizes user-friendly design, providing doctors with an intuitive interface for inputting patient data, symptoms, and diagnostic details. This ease of use is crucial for streamlining the workflow in medical settings, reducing the time spent on manual documentation, and allowing healthcare providers to focus more on patient care.

Additionally, the system incorporates real-time updates and secure sharing mechanisms, enabling prompt dissemination of prescriptions to patients and pharmacies.

This ensures timely medication fulfillment and enhances overall patient experience.

The robustness of the platform is further ensured through rigorous testing and validation phases, incorporating feedback from medical professionals to continuously refine and improve the system's functionality. The use of secure cloud-based infrastructure guarantees scalability, reliability, and the protection of sensitive medical data.

In essence, the AI-based prescription generation interface not only optimizes the prescription process but also provides a sophisticated tool for enhancing patient care. This development marks a significant advancement in the application of AI in healthcare, offering a comprehensive, secure, and intelligent solution for modern medical practice.

7.2 Future Scope

The future scope of developing a user interface for doctors to generate prescriptions using Natural Language Processing (NLP) as an integrated AI model encompasses continuous enhancement and adaptation to address emerging challenges and leverage technological advancements in the healthcare sector. Potential future developments include:

1. Enhanced Natural Language Understanding: - Implement advanced NLP techniques to better understand and interpret complex medical terminology and context, improving the accuracy and relevance of generated prescriptions.

2. Multilingual Support: - Develop capabilities to support multiple languages, allowing doctors worldwide to use the system, thereby enhancing its accessibility and utility in diverse linguistic regions.

3. Integration with Electronic Health Records (EHR): - Seamlessly integrate with EHR systems to automatically retrieve patient histories, lab results, and previous prescriptions, ensuring comprehensive and informed prescription generation.

4. Voice-Activated Input: - Introduce voice recognition technology to allow doctors to dictate patient information and symptoms directly into the system, streamlining

the data entry process and saving time.

5. Personalized Medicine and Genomic Data Integration: - Incorporate patient-specific genomic data to tailor prescriptions more precisely, supporting the move towards personalized medicine by considering genetic predispositions and drug interactions.

6. Real-time Clinical Decision Support: - Enhance the interface with real-time clinical decision support systems that provide doctors with evidence-based recommendations and alerts about potential drug interactions or contraindications.

7. Telemedicine Integration: - Integrate with telemedicine platforms to facilitate remote consultations and prescription generation, expanding access to healthcare services for patients in remote or underserved areas.

8. AI-driven Predictive Analytics: - Use predictive analytics to anticipate patient responses to medications based on historical data and machine learning models, aiding doctors in making more informed prescription decisions.

9. Blockchain for Data Security: - Implement blockchain technology to secure and validate prescription data, ensuring the integrity and confidentiality of patient information and preventing unauthorized access or tampering.

10. User Feedback and Adaptive Learning: - Incorporate mechanisms for user feedback to continuously refine and improve the NLP model. Adaptive learning algorithms can help the system evolve based on real-world usage and outcomes.

11. Mobile Application Development: - Develop mobile applications to provide doctors with the flexibility to generate and review prescriptions on-the-go, enhancing convenience and accessibility.

12. Ethical AI and Bias Mitigation: - Ensure that the AI model is transparent, fair, and free from biases. Implement strategies to regularly audit and mitigate any biases that could affect prescription accuracy and patient safety.

13. Enhanced Usability and User Interface Design: - Focus on improving the usability and design of the interface, making it more intuitive and user-friendly to ensure a

seamless experience for doctors of varying tech-savviness.

14. Continuous Monitoring and Updates: - Establish a system for continuous monitoring and updating of medical guidelines, drug databases, and best practices to keep the system current with the latest medical knowledge.

15. Customization for Specialized Fields: - Customize the system for different medical specialties, allowing for specialized features and templates that cater to the unique needs of various medical fields such as pediatrics, geriatrics, and psychiatry.

By embracing these future developments, the AI-driven interface for prescription generation can continue to evolve, offering a robust, adaptive, and comprehensive solution for the ever-changing landscape of healthcare and medical practice.

References

Kuncahyo Setyo Nugroho, N. Y., Anantha Yullian Sukmadewa. (2022). Large-scale news classification using bert language model: Spark nlp approach. In *De- partment of informatics engineering, faculty of computer science, brawijaya uni- versity, indonesia* (pp. 63–72).

Markus Bayer1, M.-A. K. (2022). Data augmentation in natural lan- guage processing: a novel text generation approach for long and short text classifers. In *International symposium on intelligent data analysis* (pp. 63–72).

Prashant Kaushik, S. P. (2021). Context aware gan for sequence text genera- tion in tensorflow lite for android ai chat application. In *Proceedings of the acm sigcomm workshop on models, methods and tools for reproducible network research* (pp. 28–32).

Weizhe Yuan, P. L., Graham Neubig. (2021). Evalu- ating generated text as text generation. In *Carnegie mellon university* (pp. 99–111). Springer.