



# Hacettepe University

Computer Engineering Department

**BBM479/480 End of Project Report**

## Project Details

<b>Title</b>	Collaborative AI Healthcare Solution Development
<b>Supervisor</b>	İlyas Çiçekli

## Group Members

	<b>Full Name</b>	<b>Student ID</b>
1	Gökçenaz Akyol	2200765030
2	Evren Çağılıcı	21945977
3	Utku Bora	21992871
4		

## **Abstract of the Project ( / 10 Points)**

In the contemporary healthcare environment, several persistent challenges affect both patients and healthcare professionals. Patients often struggle to determine which specialist to consult based on their symptoms, leading to delays in appropriate care and potentially resulting in ineffective or incorrect treatment. Healthcare professionals face the daunting task of staying updated with the extensive and continually expanding body of medical literature, making it difficult to keep abreast of the latest developments and increasing the complexity involved in prescribing the correct medication.

To address these issues, a comprehensive AI-based system was developed in a user-friendly website format. This system offers tailored access and functionalities based on the user type, including separate logins for doctors, patients, and administrators. Three models—Literature Review Model, Drug Recommendation Model, and Disease Detection Model—were meticulously trained to facilitate key aspects of healthcare, leveraging advanced artificial intelligence and machine learning techniques to enhance healthcare delivery and support medical practice.

The Literature Review Model assists healthcare professionals in efficiently accessing and reviewing relevant medical literature, utilizing a specialized dataset compiled from textbooks and medical literature, and supported by web scraping for continuous updates. The Drug Recommendation Model aids doctors in selecting appropriate medications by providing evidence-based recommendations tailored to individual patient needs, reducing uncertainty and enhancing the quality of patient care. The Disease Detection Model, leveraging the BERT architecture, accurately identifies diseases based on symptoms, aiding both patients and doctors in diagnostic support. The dataset for this model includes extensive combinations of symptoms and their corresponding diseases. The system's integration into a chatbot format further enhances its accessibility and usability for end-users. The project's expected impact includes improved healthcare accessibility and effectiveness, reduced medication errors, and enhanced decision-making for both patients and healthcare professionals. Future directions may involve expanding the models' capabilities and integrating additional functionalities to further support comprehensive healthcare solutions.

## Introduction, Problem Definition & Literature Review ( / 20 Points)

In the modern healthcare environment, several persistent challenges continue to affect both patients and healthcare professionals. Patients often struggle to determine which specialist to consult based on their symptoms, leading to delays in receiving appropriate care and potentially resulting in ineffective or incorrect treatment. This confusion can result in a significant barrier to timely and effective healthcare.

Healthcare professionals, on the other hand, face the daunting task of staying updated with the extensive and continually expanding body of medical literature. The sheer volume of new research and advancements can be overwhelming, making it difficult for practitioners to keep abreast of the latest developments. Additionally, the complexity involved in prescribing the correct medication for various diseases adds another layer of difficulty. Doctors often struggle with deciding on the most suitable medication, which can compromise the quality of patient care and increase the risk of medication errors.

To address these pressing issues, a comprehensive system has been developed in a user-friendly website format. This system offers separate logins for doctors, patients, and administrators, providing tailored access and functionalities based on the user type. Depending on their login, users can access different models specifically designed to assist them. These models include a Literature Review Model, a Drug Recommendation Model, and a Disease Detection Model, each meticulously trained to facilitate key aspects of healthcare and improve overall health outcomes. By leveraging advanced artificial intelligence and machine learning techniques, this system aims to enhance healthcare delivery, support medical practice, and ultimately make healthcare more accessible and effective for everyone involved.

A total of three models have been meticulously trained to facilitate key aspects of healthcare:

- 1. Literature Review Model:** The Literature Review Model is designed to assist healthcare professionals in efficiently accessing and reviewing relevant medical literature. A specialized dataset has been created for this model, compiled from textbooks and literature sources specifically used in the field of medicine. This dataset has been trained with an advanced Large Language Model (LLM), enabling the Literature Review Model to analyze vast quantities of medical texts, including research papers, clinical studies, and medical journals. Additionally, the model is supported by web scraping, allowing it to stay current by continuously integrating the latest information from reliable medical websites. This ensures that healthcare professionals, as well as medical students, have access to the most up-to-date research findings and can make informed decisions based on the most current knowledge.
- 2. Drug Recommendation Model:** The Drug Recommendation Model assists doctors in selecting the most appropriate medications for various diseases. This model has been trained with a dataset that ranks medications according to their success rates for different disease types. So, the model provides evidence-based recommendations tailored to individual patient needs. This makes it easier for doctors to make successful drug recommendations, reducing uncertainty and enhancing the quality of patient care. The Drug Recommendation Model thus helps in minimizing medication errors and ensures that treatment decisions are based on the most effective and current pharmaceutical knowledge.
- 3. Disease Detection Model:** The Disease Detection Model is designed to aid both patients and doctors by accurately identifying diseases based on the symptoms presented. This model leverages the BERT (Bidirectional Encoder Representations from Transformers) architecture to understand the input sentences describing symptoms. It then calculates a probability for each potential disease and returns the disease with the highest probability. This system allows both doctors and patients to benefit from accurate and timely diagnostic support. Patients can use the model to gain insights into their symptoms and determine the appropriate specialist to consult,

while doctors can use it to confirm or refine their diagnoses, ensuring more precise and effective treatment plans.

When previous studies in these areas are examined, it is evident that artificial intelligence-based diagnostic models have a higher successful diagnosis rate compared to physicians. According to a study conducted with German physicians with an average of 10 years of medical experience, it was observed that the artificial intelligence model was 29% more successful than physicians in making the correct diagnosis for patients.

Several AI models have been developed to address specific healthcare challenges. ChatGPT, developed by OpenAI, is a prominent general-purpose language model. It is designed to generate human-like text based on the input it receives, leveraging deep learning techniques and the GPT architecture. While ChatGPT is versatile and capable of engaging in a wide range of conversations, its general-purpose nature means it lacks the specialized focus required for medical applications. Consequently, it may not always provide the most accurate or specific medical information, as it is not exclusively trained on medical data.

ClinicalBERT, another significant project in the field of medical AI, is a variation of the BERT (Bidirectional Encoder Representations from Transformers) model. ClinicalBERT has been fine-tuned on clinical notes from electronic health records (EHRs) to improve its understanding of medical language and its ability to predict hospital readmissions. By training on a large dataset of clinical notes, ClinicalBERT can better capture the nuances and specifics of medical terminology, making it more effective in clinical settings compared to general-purpose models. However, ClinicalBERT's focus is primarily on understanding clinical notes and predicting readmissions, rather than providing comprehensive access to medical information or assisting with drug recommendations and disease detection from symptoms.

In contrast, the current project introduces a comprehensive system designed to address these issues through a user-friendly website format. Separate logins are available for doctors, patients, and administrators, each providing tailored access and functionalities. Users can access different models specifically designed to assist them, including a literature review model, a drug recommendation model, and a disease detection model.

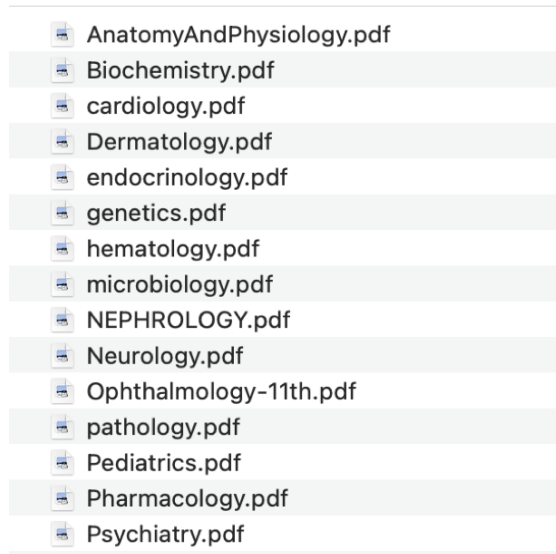
While some open-source codes for drug recommendation and disease detection models based on symptoms exist on the internet, these are neither academic nor comprehensive. Additionally, these models typically remain at the model stage and have not been integrated into a chatbot as done in this project.

## Methodology ( / 25 Points)

### Literature Review Model

#### A. Data Set

In developing this model, it was identified that there was a significant lack of datasets focusing on medical literature. While several datasets exist that cover symptoms, sickness predictions, and clinical notes about patients, there was no comprehensive dataset available that covered generic medical knowledge and literature.



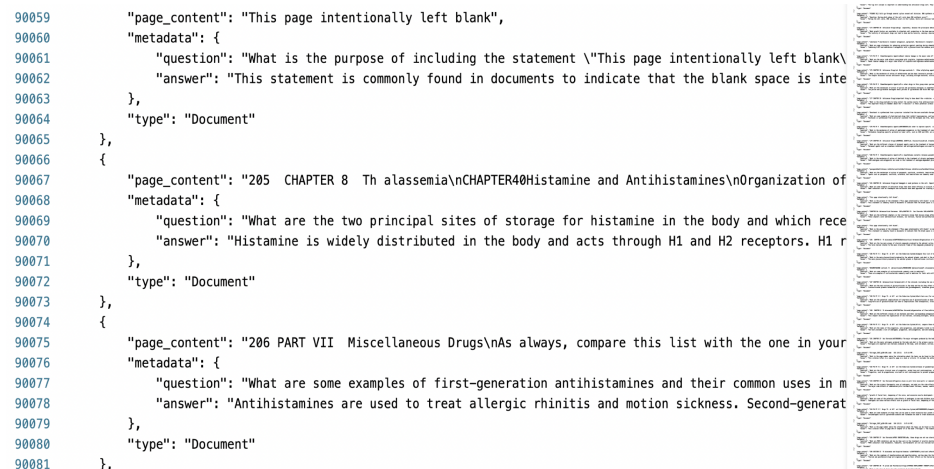
**Fig. 1. Resources to create a new dataset**

To address this gap, a new dataset was created using approximately 15 medical books covering various subjects including cardiology, hematology, and other key medical fields as seen in Figure 1. These books were selected to provide a wide-ranging base of medical knowledge that is essential for training the chatbot model to answer diverse medical questions accurately.

The process of creating this dataset involved several steps, starting with the division of each medical book into manageable chunks using the Langchain library. Once the documents were chunked, the next step was to generate questions and answers based on the content of each chunk. This was accomplished using the OpenAI library, which provides access to advanced language models capable of understanding and generating human-like text. The process involved the following steps:

- **Setting Parameters:** Specific parameters such as the model name (e.g., GPT-3), temperature (which controls the randomness of the output), and other relevant settings were configured to ensure the quality and relevance of the generated questions and answers.
- **Preparing the Request:** Each chunk of text was used as input to the OpenAI model, with prompts designed to elicit useful question-answer pairs. These prompts were created to guide the model in generating responses that are directly relevant to the content of the chunk.
- **Sending the Request:** Requests were sent to the OpenAI API, which processed the input and generated the corresponding questions and answers. This step involved communicating with the API over the internet, where the text chunks and prompts were transmitted to the OpenAI servers for processing.

- **Receiving and Storing Responses:** The generated question-answer pairs were received from the OpenAI API. These pairs were stored along with their original text chunks as metadata in both JSON and CSV formats. This structured storage approach ensured that the data was well-organized and easily accessible for further training and development of the chatbot model.



```

90059     "page_content": "This page intentionally left blank",
90060     "metadata": {
90061         "question": "What is the purpose of including the statement \"This page intentionally left blank\"",
90062         "answer": "This statement is commonly found in documents to indicate that the blank space is intentional."
90063     },
90064     "type": "Document"
90065 },
90066 {
90067     "page_content": "205 CHAPTER 8 Thalassemia\nCHAPTER40Histamine and Antihistamines\nOrganization of",
90068     "metadata": {
90069         "question": "What are the two principal sites of storage for histamine in the body and which receptors",
90070         "answer": "Histamine is widely distributed in the body and acts through H1 and H2 receptors. H1 receptors",
90071     },
90072     "type": "Document"
90073 },
90074 {
90075     "page_content": "206 PART VII Miscellaneous Drugs\nAs always, compare this list with the one in your",
90076     "metadata": {
90077         "question": "What are some examples of first-generation antihistamines and their common uses in medicine?",
90078         "answer": "Antihistamines are used to treat allergic rhinitis and motion sickness. Second-generation",
90079     },
90080     "type": "Document"
90081 },

```

**Fig. 2. Examples from the dataset**

In total, approximately 13,000 question and answer pairs were generated through this process. These pairs (example shown in Figure 2) provide a comprehensive dataset that covers a wide range of medical topics, ensuring that the chatbot model has access to a broad base of knowledge.

## B. Model

This project involves training a conversational AI model for medical applications using the DialoGPT [2] model from Hugging Face. The implementation is carried out in a Jupyter notebook and follows a structured approach starting from environment setup and data preprocessing to model training and evaluation. The goal is to create a model capable of answering medical questions accurately, making it a valuable tool for healthcare professionals and researchers.

The preprocessing step involves loading a dataset of medical documents, extracting relevant fields, and preparing the data for training. The dataset, loaded from a JSON file, contains metadata fields for questions and answers. Unnecessary columns are dropped, and data types are converted to string format for consistency. The dataset is then split into training and validation sets using an 80-20 split to evaluate the model's performance and prevent overfitting.

For the model setup, the project uses the "microsoft/DialoGPT-small" variant of DialoGPT, known for its conversational capabilities. The training parameters are carefully selected to optimize the model's performance. The batch size is set to 2 per GPU, with a total of 3 epochs. Gradient accumulation steps are set to 1, allowing for effective gradient updates with small batch sizes. The learning rate is configured at 5e-5, with the AdamW optimizer used for parameter updates. Weight decay is set to 0.0, and the AdamW epsilon is 1e-8 to maintain numerical stability.

The training process involves defining a custom dataset class to manage data loading, batching, and caching. Input data is prepared by encoding and padding the sequences, and the model is trained in

batches with the loss calculated and backpropagated. Gradient clipping is applied to prevent exploding gradients. Training progress is logged, and model checkpoints are saved periodically to allow for recovery in case of interruptions.

The evaluation phase assesses the model's performance calculating metrics such as perplexity, euclidean distance and relevancy score to gauge its effectiveness.

Finally, the trained model is saved and uploaded to the Hugging Face Hub for easy access and sharing.

## C. Other Key Processes

The project utilizes a web scraping function, which is designed to extract relevant information from web pages. This function takes a query string, constructs a Google search URL, and retrieves the corresponding web page content using the requests library. The content is then parsed with BeautifulSoup to extract useful links and text elements. Specifically, the function searches for Wikipedia links to prioritize structured and reliable information. If a Wikipedia link is found, the content is further parsed to extract the article's title and the first paragraph, providing a concise summary of the topic.

## Drug Recommendation Model

### A. Data Set

condition	rating	drugName		
Bipolar Disorde	10	Aripiprazole		
Depression	10	L-methylfolate		
Bipolar Disorde	10	Lamotrigine		
Chronic Myelogenous Leukemia	10	Nilotinib		
Insomnia	10	Trazodone		
Rheumatoid Arthritis	10	Etanercept		
Hirsutism	10	Eflornithine		
ADHD	10	Daytrana		

**Fig. 3. Examples from the dataset**

The dataset that model used consists of 161,528 entries, detailing information about various drugs and their ratings for different medical conditions. It includes five columns: *condition*, *rating*, *drugName*, and two unnamed columns (as shown in Figure 3) that are mostly empty. The *condition* column records the medical conditions for which the drugs are prescribed, while the *rating* column provides a numerical rating for the effectiveness of each drug. The *drugName* column lists the names of the drugs. The dataset reveals valuable insights into how different drugs are perceived in terms of their efficacy for various conditions.

## B. Model

The dataset is loaded from a CSV file using pandas, and the drugName column is encoded into numerical labels through the LabelEncoder from the sklearn library. These encoded labels are subsequently saved to a CSV file for future reference.

Following this, the tokenizer and model are loaded from the Hugging Face Hub, specifically utilizing the RLHFlow/ArmoRM-Llama3-8B-v0.1 model. The dataset is converted into a datasets.Dataset object, focusing on the columns condition and Drug\_Name\_encoded. A tokenization function is defined to process the condition column with appropriate padding and truncation. This function is applied to the entire dataset using the map method, ensuring that all rows are tokenized consistently. Finally, the dataset is split into training and testing sets with a 70-30 ratio, ensuring a balanced division of data for model training and validation.

For the training phase, TrainingArguments from the transformers library are defined with specific parameters tailored to the training process. These parameters include the directory for output storage (output\_dir), the number of training epochs (num\_train\_epochs), batch size per device (per\_device\_train\_batch\_size), and the evaluation strategy (evaluation\_strategy).

The Trainer class from the transformers library is then initialized. This class is provided with the model, the defined training arguments, and the train and test datasets. During training, the model is iteratively updated to minimize the loss function, with evaluations conducted at the end of each epoch to monitor performance improvements.

## Disease Detection Model

### A. Data Set

I have been having migraines and headaches. I can't sleep. My whole body is shaking and shivering. I feel dizzy sometimes.	308
I have asthma and I get wheezing and breathing problems. I also have fevers, headaches, and I feel tired all the time.	35
Signs and symptoms of primary ovarian insufficiency are similar to those of menopause or estrogen deficiency. They include: Irregular or skipped periods,...	798
cough,high_fever,breathlessness,family_history,mucoid_sputum	149
chills,vomiting,high_fever,sweating,headache,nausea,diarrhoea,muscle_pain	596

**Fig. 4. Examples from the dataset**

The "Symptom-Disease Dataset" by duxprajapati, available on Hugging Face, is a comprehensive resource designed for text classification tasks in the medical field. The dataset comprises two main CSV files: the training dataset (1.58 MB) and the test dataset (399 kB). It maps various symptoms to corresponding diseases, facilitating the development of machine learning models aimed at diagnosing medical conditions based on symptom input. Each row in the dataset represents a unique combination of symptoms associated with a specific disease. For example, some entries include combinations like fatigue, nausea, and yellowing of eyes, or vomiting, anxiety, and blurred vision.



## B. Model

The dataset utilized for this model was acquired using the *load\_dataset* function from the *datasets* library, which facilitated the retrieval of both the training and test splits. Following the acquisition, the datasets were transformed into Pandas DataFrames to enable more straightforward manipulation and analysis. Subsequently, the textual data representing symptoms and the corresponding disease labels were extracted into separate lists.

To prepare the labels for the model, a *LabelEncoder* from the scikit-learn library was employed. This encoder transformed the categorical disease labels into numerical format, rendering them suitable for processing by the model. The BERT tokenizer, *BertTokenizer*, was then utilized to tokenize the text data. This tokenization process involved converting the text into tokens comprehensible by the BERT model.

The tokenized inputs, specifically the *input\_ids* and *attention\_masks*, were partitioned into training and test sets using the *train\_test\_split* function from scikit-learn. This partitioning ensured that 80% of the data was allocated for training purposes, while the remaining 20% was reserved for testing. The *TFBertForSequenceClassification* model from the transformers library was then instantiated with the pre-trained BERT base model, specifying the number of labels corresponding to the encoded diseases.

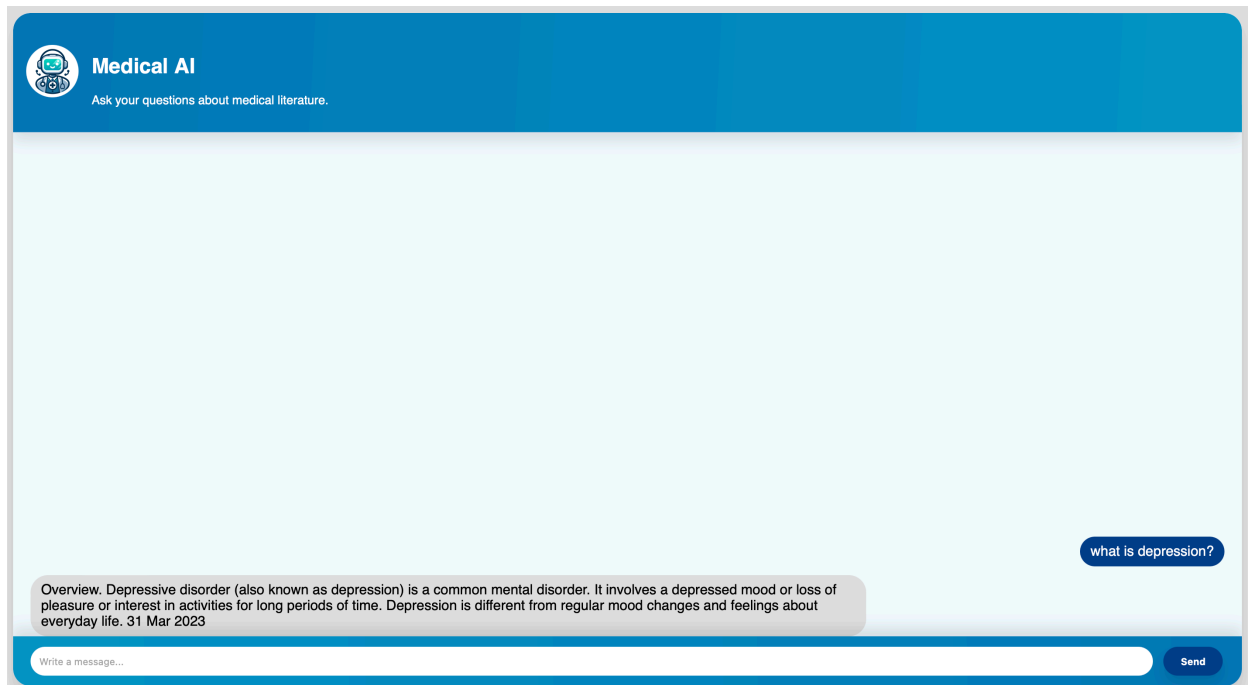
The model was compiled using the Adam optimizer with a learning rate of  $2e-5$  and epsilon of  $1e-08$ . The loss function utilized was *SparseCategoricalCrossentropy* with logits, and the metric chosen for evaluation was *SparseCategoricalAccuracy*. The training process involved fitting the model on the training data for three epochs with a batch size of eight. Validation data, comprising the test split, was used to monitor the model's performance during training.

To facilitate the interpretation of model predictions, a mapping of encoded labels to disease names was maintained. This mapping was inverted to allow for the translation of model output back to human-readable disease names. The trained model was then saved in the 'disease\_classifier\_model' directory, enabling future loading and inference.

## Website

An interface in the form of a website was developed using Flask. To enhance user-friendliness, HTML and CSS codes were also written. Through this website, patients and doctors can register and log in. Subsequently, these users are presented with model options tailored to their specific needs. Doctors are granted access to all three models, whereas patients are restricted to using the model that best suits their requirements. Additionally, medical students can benefit from the site. A chat interface, aligned with the selected model, welcomes the user. The user inputs their question into the chat section, and the model returns the answer accordingly.

## Results & Discussion ( / 30 Points)



The code provided for evaluation of Literature Review Model performs several key functions necessary for evaluating a language model. It begins by importing essential libraries, including `sentence_transformers` for sentence embeddings and `openai` for language model interactions. The function `load_similarity_model` initializes a `SentenceTransformer` model, which is later used for encoding sentences into embeddings. The function `compute_similarity` calculates the cosine similarity between the embeddings of a reference sentence and a ground truth answer, while `compute_euc_dist` measures the Euclidean distance between these embeddings. The `compute_perplexity` function determines the perplexity of both the reference and ground truth embeddings, providing insight into the model's predictive capabilities. Additionally, `compute_relevancy` and the associated asynchronous function `is_relevant` assess the relevance of a given response to a prompt using the OpenAI language model. Another asynchronous function, `get_ground_truth_answer`, fetches the expected answer from the language model. The `evaluate_answer` function synthesizes these components to evaluate a provided answer against a ground truth by computing all relevant metrics.

The ground truth answers were obtained from the ChatGPT model, providing a reliable reference for evaluation. The similarity between the provided answers and the ground truth was calculated using cosine similarity, resulting in an average similarity score of 0.6154. This metric indicates the degree of semantic similarity between the provided answers and the ground truth. The Euclidean distance between the embeddings of the provided answers and the ground truth was also measured, yielding an average distance of 2.5342. This distance metric reflects how far apart the provided answers are from the ground truth in the high-dimensional embedding space. Perplexity scores were computed to assess the uncertainty of the language model when predicting the provided answers and the ground truth. The average perplexity score was found to be 0.9984, indicating the model's effectiveness in generating predictions. Relevancy of the provided answers was determined using a binary classification approach, facilitated by the OpenAI API. This metric evaluated whether the provided answers contained the correct responses to the given prompts.

The evaluation of the Disease Detection Model was conducted meticulously to ensure a comprehensive understanding of its performance. During each epoch, the model's performance was monitored using the validation data, which consisted of the test split. This approach ensured that the model's ability to generalize to unseen data was consistently evaluated throughout the training process.

The results from the training epochs indicated significant improvements in both the loss and accuracy metrics. In the first epoch, the model achieved a loss of 4.3195 and an accuracy of 37.40%, with a validation loss of 2.6444 and a validation accuracy of 80.20%. By the second epoch, the loss had decreased to 1.9594, and the accuracy had increased to 80.32%, while the validation loss further reduced to 1.3368, with a validation accuracy of 83.39%. In the third and final epoch, the model attained a loss of 1.2851 and an accuracy of 83.33%, with a validation loss of 1.2470 and a validation accuracy of 84.10%.

These results demonstrate that the model's performance improved consistently with each epoch, indicating effective learning and adaptation to the training data. The high validation accuracy suggests that the model is capable of generalizing well to new, unseen data, thereby making it a reliable tool for disease diagnosis based on symptom inputs.

## **The Impact and Future Directions ( / 15 Points)**

Several qualitative and quantitative updates can be envisioned to further enhance the system's capabilities. Expanding the dataset to include more diverse and comprehensive medical literature, drug information, and symptom-disease mappings would improve the models' accuracy and applicability across various medical fields. Incorporating advanced machine learning techniques and regularly updating the models with the latest data will ensure continued relevance and effectiveness.

The project could also explore integration with electronic health record (EHR) systems to provide real-time, personalized recommendations based on patient history and current health status.

Additionally, extending the system's language support to cater to non-English speaking users would broaden its accessibility and impact globally. Currently, the chatbot may be limited to specific languages, which restricts its accessibility to non-English speaking users. Developing multilingual support would make the tool more inclusive, allowing it to serve a global audience of patients, medical professionals and students.

Future research and development efforts could focus on enhancing the user interface and experience, ensuring that the system remains intuitive and easy to navigate for all users. Collaborative partnerships with medical institutions and research organizations could facilitate the collection of high-quality data and the validation of model performance in real-world settings.

In conclusion, the developed system has the potential to transform healthcare delivery by leveraging artificial intelligence and machine learning to support medical practice. The next steps in this project involve expanding datasets, improving model accuracy, integrating with EHR systems, and enhancing user accessibility. By continuously evolving, the system can further contribute to making healthcare more efficient, accurate, and accessible for all stakeholders involved.