**DOCTOR PRESCRIPTION GENERATOR**

***A thesis***

***Submitted to Department of Computer Science & Engineering***

*in the partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

By

G. NEYNA SRI - 21567T0920

N. NISHITHA - 21567T0943 V.NARESH - 21567T0958 C.SAI PRAKASH - 21567T0910 P. MURALI - 21567T0947

UNDER THE GUIDANCE OF Prof M. SADANANDAM



**DEPARTMENT OF COMPUTER SCIENCE AND**

**ENGINEERING**

**KU COLLEGE OF ENGINEERING & TECHNOLOGY KAKATIYA UNIVERSITY CAMPUS, VIDYARANYAPURI, WARANGAL - 506009, INDIA,**

**JULY 2023.**

**KU COLLEGE OF ENGINEERING & TECHNOLOGY KAKATIYA UNIVERSITY CAMPUS, VIDYARANYAPURI, WARANGAL –506009**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**Certificate**

This is to certify that this thesis entitled “DOCTOR PRESCRIPTION GENERATOR” that is being submitted by G. NEYNA SRI (21567T0920), N. NISHITHA (21567T0943), V. NARESH (21567T0958),

C. SAI PRAKASH (21567T0910), P. MURALI (21567T0947) the partial fulfillment for the award of **Bachelor of Technology** in **Computer Science & Engineering** to the KAKATIYA UNIVERSITY is a record of work carried out during the academic year 2022-2023 under our guidance and supervision.

Supervisor Project Coordinator

*(M. Sadanandam)* ***(Smt. D. Bhagyalaxmi)***

Head of the Department Principal

***(Sri. M. Venugopal Reddy) (Professor. M. Sadanandam)***

Internal Examiner External Examiner

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We declare that the work presented in this project report is original and carried out in the department of Computer Science & Engineering, KU College of Engineering & Technology, Warangal, Telangana and have not been submitted elsewhere for any graduate in part or in full.

|  |  |
| --- | --- |
| G. NEYNA SRI | - 21567T0920 |
| N. NISHITHA | - 21567T0943 |
| V.NARESH | - 21567T0958 |
| C.SAI PRAKASH | - 21567T0910 |
| P. MURALI | - 21567T0947 |

I thank the almighty for giving me the courage and perseverance in completing the project. This project itself is an acknowledgement for all these people who have given me their heartfelt co-operation in making this project success.

I take this opportunity to express my deep and sincere gratitude to the project In- charge **Prof. M. Sadanandam** Department of CSE, for their valuable advice at every stage of this work; without their supervision this project would never have come out in this form.

I am greatly indebted to my project supervisor ***Smt. D. Bhagya Laxmi*** for many hours of dedicated guidance, stimulating and constructive criticism. This project came out in this form.

I am thankful to **Sri. M. Venu Gopal Reddy**, Head of the Department, for providing the excellent facilities, motivational and valuable guidance throughout the project work.

I am thankful to **Prof. M. Sadanandam**, Principal, KU College of Engineering and Technology, for his cooperation and encouragement to complete the project work in time. Last but not least, I would like to express my deep sense of gratitude and earnest thanksgiving to my dear parents for their moral support and heartfelt cooperation in doing the project. I would also like to thank all the teaching and non-teaching staff and our friends, whose direct or indirect help has enabled us to complete this work successfully.

|  |  |
| --- | --- |
| G. NEYNA SRI | - 21567T0920 |
| N. NISHITHA | - 21567T0943 |
| V.NARESH | - 21567T0958 |
| C.SAI PRAKASH | - 21567T0910 |
| P. MURALI | - 21567T0947 |

The advancement of technology has paved the way for innovative solutions in various fields, including healthcare. In this study, we propose an intelligent prescription development system designed to assist a doctor in generating accurate treatment plans and medication based on the symptoms provided by the user. Leveraging machine learning algorithms and natural language processing techniques, the system analyses the user's symptoms inputted through an intuitive interface.

It then cross-references the symptoms with a comprehensive database of medical conditions, medications, and treatment protocols to generate personalized prescriptions. Initially, it conducts symptom clustering and pattern recognition to identify potential medical conditions associated with the presented symptoms.

The user interface of the system is designed to be user-friendly, allowing them to easily input and

modify symptoms, view and edit generated prescriptions.

|  |  |
| --- | --- |
| **CERTIFICATE** | **II** |
| **DECLARATION** | **III** |
| **ACKNOWLEDGEMENT** | **IV** |
| **ABSTRACT** | **Ⅴ** |
| **TABLEOFCONTENTS** | **VI** |
| **LISTOFFIGURES** | **VIII** |
| **LISTOFSCREENSHOTS** | **IX** |
| **LIST OFABBREVIATIONS** | **X** |
| **CHAPTER 1: INTRODUCTION** | **1** |
| 1.1 Introduction | 2 |
| **CHAPTER 2: LITERATURE SURVEY** | **3** |
| 2.1 Literature survey | 4 |
| **CHAPTER 3: SYSTEM ANALYSIS** | **5** |
| 3.1 Existing system | 6 |
| 3.2 Drawbacks | 6 |
| 3.3 Proposed system | 7 |
| 3.4 Advantages | 7 |
| 3.5 System Requirements | 8 |
| 3.6 Feasibility study | 9 |
| **CHAPTER 4: SYSTEM ARCHITECTURE** | **10** |
| 4.1 System architecture | 11 |
| 4.2 Modules | 12 |
| 4.3 UML Diagrams | 12 |
| **CHAPTER 5: SYSTEM IMPLEMENTATION** | **14** |
| 5.1 Python | 15 |
| 5.2 SQL (Structured Query Language) | 25 |
| 5.3 HTML/CSS/JavaScript | 25 |
| 5.4 SQL Alchemy | 26 |
| 5.5 Other Python Libraries | 26 |
| 5.6 Source code | 27 |
| **CHAPTER 6: RESULTS** | **44** |
| 6.1 Screenshots | 45 |

|  |  |
| --- | --- |
| **CHAPTER 7: TESTING** | 47 |
| **7.1 Testing** | 48 |
| **CHAPTER 8: CONCLUSION** | **50** |
| **8.1 Conclusion** | 51 |
| **CHAPTER 9: FUTURE ENHANCEMENTS** | **52** |
| 9.1 **Future enhancements** | 53 |
| **REFERENCES** | **54** |

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name of the Figure** | **Page No.** |
| 1 | System Architecture of the model | 11 |
| 2 | Home page | 45 |
| 3 | Login page | 45 |
| 4 | Prescription page | 45 |
| 5 | Patient History page | 46 |
| 6 | Registration page | 46 |
| 7 | Result page | 46 |

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name of Screenshot** | **Page No** |
| 2 | Home page | 45 |
| 3 | Login page | 45 |
| 4 | Prescription page | 45 |
| 5 | Patient History page | 46 |
| 6 | Registration page | 46 |
| 7 | Result page | 46 |

# CHAPTER-1 INTRODUCTION

## 1.1INTRODUCTION

Medical recommendation systems are an essential part of healthcare technology, aimed at providing accurate and relevant medical advice based on user input, such as symptoms or medical history. These systems leverage various computational techniques to assist both healthcare providers and patients in making informed decisions about treatment options. The integration of machine learning, natural language processing (NLP), and web technologies has significantly enhanced the capabilities and accessibility of these systems.

The integration of advanced computational techniques in medical recommendation systems represents a significant advancement in healthcare technology. By leveraging data preprocessing, feature extraction, machine learning models, and web frameworks, these systems can provide accurate and relevant medical advice, improving patient outcomes and enhancing the efficiency of healthcare delivery. The ongoing research and development in this field continue to refine these technologies, making them more robust and accessible to a broader audience.

# CHAPTER-2 LITERATURE SURVEY

## 2.1 LITERATURE SURVEY

Medical recommendation systems integrate computational techniques to provide accurate medical advice based on user inputs. Data preparation, including cleaning and feature combination, ensures quality datasets, while CountVectorizer and stemming techniques like PorterStemmer convert text data into numerical formats for machine learning models. Cosine similarity measures the relevance of medical information, aiding in identifying suitable treatments. Flask, a micro web framework, facilitates user interaction by creating interfaces for input and recommendations. MySQL manages user data and session information, ensuring efficient data retrieval. Case studies demonstrate the effectiveness of NLP in improving recommendation accuracy and user satisfaction. User-centered design is crucial for developing intuitive interfaces. These systems enhance healthcare delivery by leveraging advanced techniques, making medical advice more accessible and actionable, with ongoing research continually refining their robustness and accessibility.

# CHAPTER-3 SYSTEM ANALYSIS

## EXISTING SYSTEM

several existing projects leverage similar techniques for medical recommendation systems. One example is IBM Watson Health, which uses natural language processing (NLP) and machine learning to provide personalized treatment recommendations based on patient data. Another is the Mayo Clinic’s clinical decision support system, which integrates electronic health records (EHR) with advanced analytics to recommend treatments. The system described by Zhang et al. (2017) also uses text mining and machine learning to suggest medications based on patient reviews and symptoms. These projects highlight the practical application of machine learning and NLP in enhancing medical recommendations and improving patient care.

## DISADVANTAGES

* + - Data Quality and Completeness
    - Text Preprocessing Limitations
    - Limited Feature Scope
    - Similarity Measure Constraints
    - Scalability Issues
    - User Input Dependency
    - Database Security and Privacy

## PROPOSED SYSTEM

The project proposes developing a medical recommendation system that utilizes machine learning and natural language processing to recommend treatments based on user symptoms. By preprocessing medical data, combining relevant text features, and converting them into numerical vectors, the system uses cosine similarity to identify the most relevant treatments. A Flask-based web interface allows users to input symptoms and receive recommendations, while MySQL manages user data and sessions. This integration of technologies aims to provide accurate, personalized medical advice, enhancing patient care and accessibility to relevant treatment options.

## ADVANATAGES

* + - **Data Preparation:** Ensures high-quality, cleaned datasets.
    - **Comprehensive Feature Combination:** Merges 'Description' and 'Reason' for richer context.
    - **Advanced Text Processing:** Uses stemming for consistent word forms.
    - **Cosine Similarity:** Accurately measures text similarity.
    - **User-Friendly Interface:** Flask enables easy interaction.
    - **Efficient Data Management:** MySQL handles user sessions.
    - **Scalable Solution:** Handles growing datasets effectively.

## SYSTEM REQUIREMENTS

**Hardware**

1. 1GBRAM
2. 80 GB H/Disk
3. Above2GHzProcessor
4. AndroidMobileWithGPRS

## Software

* + - **Python:** Programming language for the core implementation.
    - **Pandas:** Data manipulation and cleaning library.
    - **scikit-learn:** Machine learning library for vectorization and similarity computation.
    - **NLTK:** Natural Language Toolkit for text preprocessing.
    - **Flask:** Web framework for creating the user interface.

## FEASIBILITY STUDY

**Technical Feasibility:** The project leverages well-established technologies like Python, scikit- learn, and Flask, ensuring robust implementation feasibility. However, continuous updates and maintenance are necessary.

**Operational Feasibility:** The system's operational aspects, including data preprocessing, vectorization, and cosine similarity calculations, are feasible with current computational capabilities. Integration with MySQL for data storage and retrieval enhances operational efficiency.

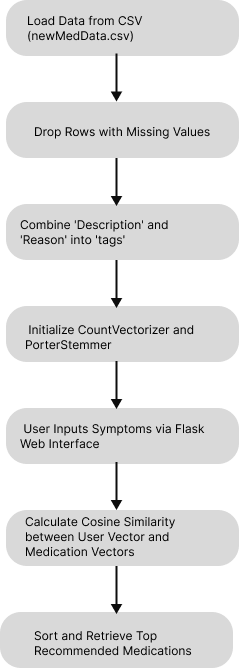
**Economic Feasibility:** The project requires minimal investment in software (Python libraries) and open-source tools (Flask, MySQL), making it economically viable. However, costs may arise from server hosting and maintenance.

**Legal Feasibility**: Compliance with data protection laws (e.g., GDPR, HIPAA) is crucial due to the sensitive nature of medical data. Implementing secure data handling practices and user consent mechanisms ensures legal feasibility.

**Schedule Feasibility:** The development timeline aligns with realistic milestones for each project phase (data preparation, feature extraction, web interface development). Agile methodologies can mitigate schedule risks through iterative development and testing.

# CHAPTER -4 SYSTEM ARCHITECTURE

## SYSTEM ARCHITECTURE



**Fig 4.1:System Architecture of the model**

## MODULES

* + - **pandas:** Used for data manipulation and loading CSV files (pd.read\_csv).
    - **sklearn (scikit-learn):** Provides machine learning algorithms and utilities. Specifically used for:

**CountVectorizer**: Converts text data into numerical vectors.

**cosine\_similarity:** Calculates similarity between vectors.

* + - **nltk (Natural Language Toolkit):** Used for natural language processing tasks, particularly for stemming using PorterStemmer.
    - **Flask:** Micro web framework used for building web applications. Handles user interface and interaction (Flask, render\_template, request).
    - **mysql-connector-python:** Python interface for MySQL database connectivity (mysql.connector).

## UML DIAGRAM

The Unified Modelling Language (UML)is used to specify, visualize,modify, constructand document the artifacts of an object-oriented software intensive system under development.UML offers a standard way to visualize a system's architectural blueprints, including elementssuchas:

Medical Recommendation System

df: DataFrame

cv: CountVectorizer ps: PorterStemmer

load\_data\_from\_csv(filename: str) preprocess\_data()

vectorize\_data() recommend\_based\_on\_tags(user\_tags: str)

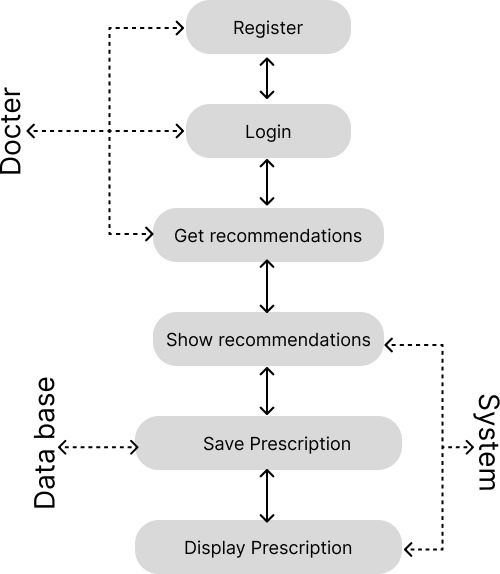
## Attributes:

* df: Represents the DataFrame containing medical data.
* cv: Represents the CountVectorizer object for text vectorization.
* ps: Represents the PorterStemmer object for text preprocessing.

## Methods:

* load\_data\_from\_csv(filename: str): Loads data from a CSV file into df.
* preprocess\_data(): Cleans and preprocesses the data by dropping missing values and combining 'Description' and 'Reason' into 'tags'.
* vectorize\_data(): Uses cv and ps to preprocess and vectorize the 'tags' column.
* recommend\_based\_on\_tags(user\_tags: str): Takes user input, preprocesses it similarly, calculates cosine similarity with existing data vectors, and returns recommended medicines**.**

## Use Case diagram



**Use Cases:**

Register/Login: Users and doctors can register and log in to the system. Input Symptoms: Users can input their symptoms.

Receive Recommendations: Users receive medication recommendations based on their symptoms.

View History: Users can view their past recommendations.

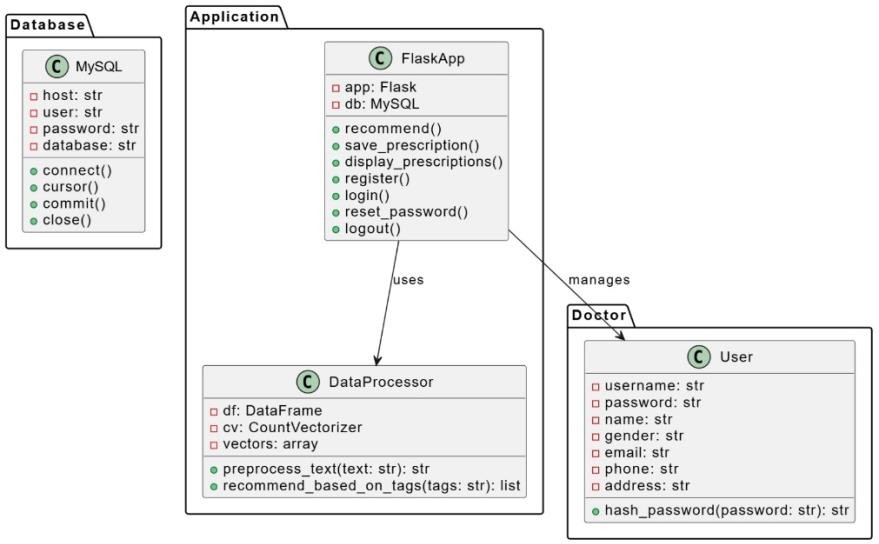
Rate Recommendations: Users can rate the effectiveness of the recommended medications. Consult Doctor: Users can request a consultation with a doctor.

Manage Recommendations: Doctors can update or provide feedback on the recommendation algorithm.

Administer System: Admins manage users, doctors, and system settings. Access Reports: Admins and doctors can access usage and effectiveness reports

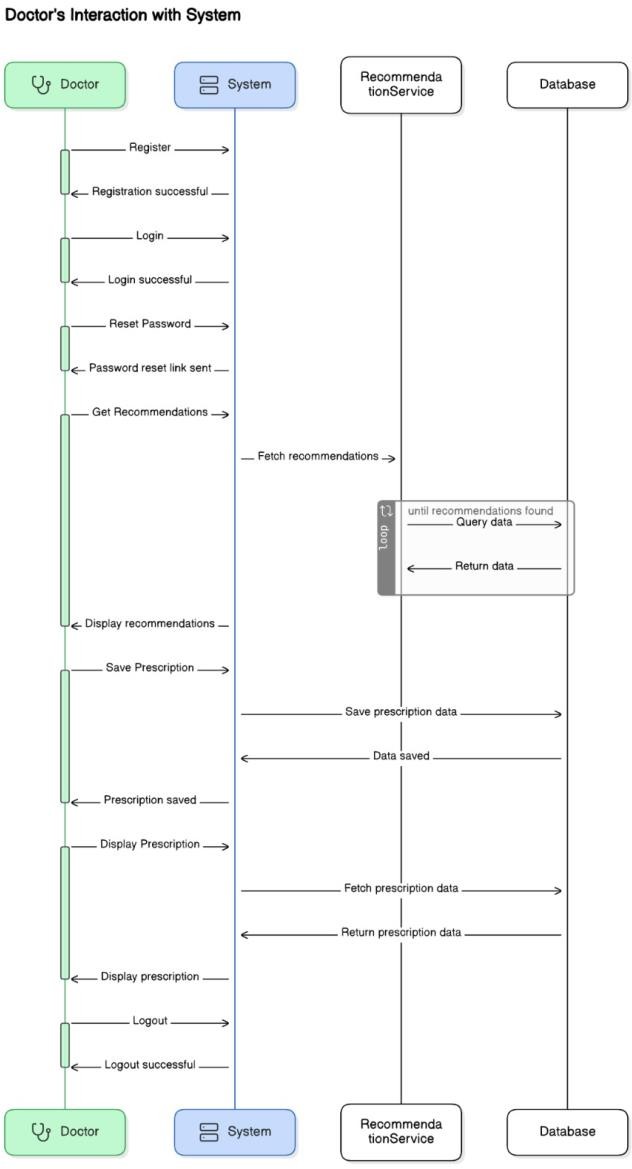
## Class diagram

To create a class diagram for the medical recommendation system, we need to identify the key classes, their attributes, methods, and relationships between them. Below is a textual representation followed by a class diagram:



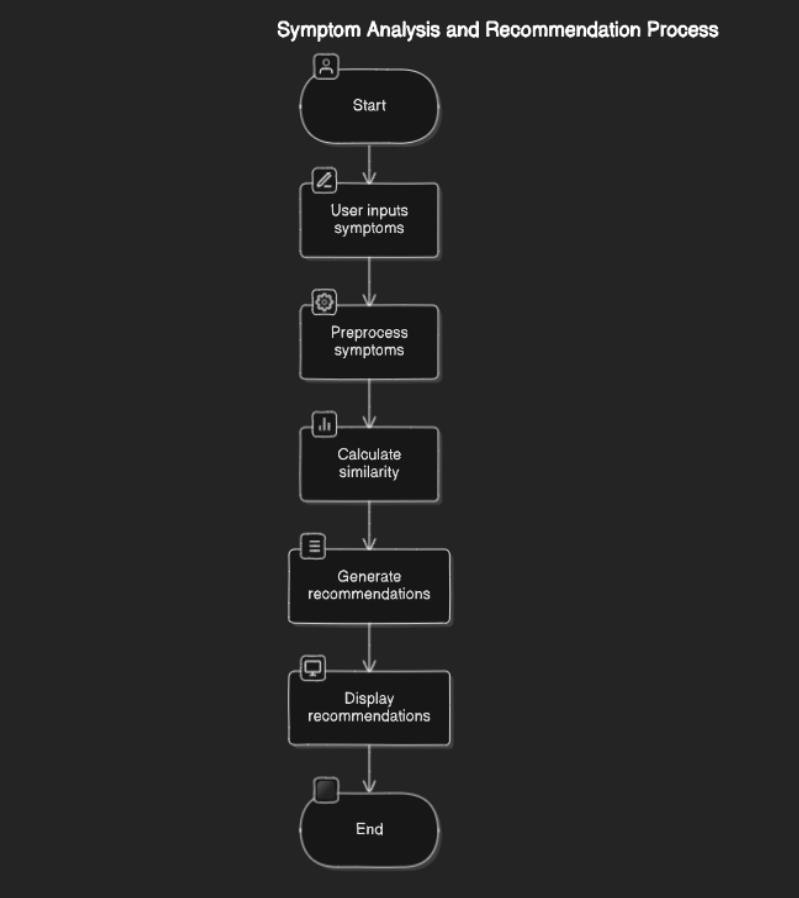
## Sequence diagram

This sequence diagram demonstrates the interactions between the user and system components, highlighting the flow from user login, symptom input, data processing, and displaying recommendations. Use UML tools to create a clear, visual representation that aligns with this textual sequence



## Activity diagram

An activity diagram visually represents the workflow of the medical recommendation system. Below is the description followed by a graphical representation:



# CHAPTER –5 SYSTEM IMPLEMENTATION

The project described utilizes several languages and frameworks to achieve its objectives of building a medical recommendation system with a user-friendly interface and efficient data processing capabilities. Here are the languages and their roles in the project

## Python

**Python** serves as the primary programming language for implementing the project due to its versatility, ease of use, and rich ecosystem of libraries and frameworks relevant to data science, machine learning, and web development.

**Libraries and Frameworks in Python:**

## pandas

Used for data manipulation and analysis. It provides tools for reading and writing data from various file formats, cleaning and preprocessing data, and performing exploratory data analysis (pd.read\_csv, df.dropna, etc.).

### Key Features and Capabilities of Pandas

1. **Data Structures**

Pandas primarily operates with two main data structures:

Series: A one-dimensional array-like object that can hold various data types such as integers, floats, and strings. It provides index labels for each element, making it flexible for data alignment.

DataFrame: A two-dimensional labeled data structure resembling a table or spreadsheet. It consists of rows and columns, where each column can hold different types of data. DataFrames allow for efficient data manipulation and analysis, akin to working with SQL tables or Excel spreadsheets.

### Data Import and Export

Pandas facilitates seamless data import from various file formats including CSV, Excel, JSON, SQL databases, and more. For instance, pd.read\_csv() is a popular function used

to read data from CSV files into a DataFrame. Similarly, Pandas provides functions like to\_csv(), to\_excel(), etc., for exporting data after processing.

### Data Cleaning and Preprocessing

Data cleaning is a crucial step in any data analysis workflow to ensure data quality and consistency. Pandas offers robust tools for:

Handling missing data (df.dropna(), df.fillna()) Removing duplicate entries (df.drop\_duplicates())

Filtering and transforming data (df[df['column'] > threshold]) Renaming columns (df.rename()), handling outliers, and more.

### Data Manipulation and Transformation

Pandas provides powerful methods for data manipulation and transformation:

Indexing and selection: Accessing specific rows, columns, or subsets of data (df.loc[], df.iloc[])

Grouping and aggregation: Grouping data based on specified criteria and applying aggregate functions (groupby() with aggregate() or apply())

Merging and joining: Combining multiple DataFrames based on common columns or indices (pd.merge(), df.join())

### Time Series Data Handling

Pandas excels in handling time series data, allowing for efficient manipulation, resampling, and analysis of time-indexed data:

Date/time indexing: Setting date or time columns as DataFrame index (df.set\_index()) Resampling and frequency conversion (resample(), asfreq())

Shifting and lagging data (shift(), diff())

### Exploratory Data Analysis (EDA)

Pandas supports EDA through statistical summary functions and visualization integration:

Descriptive statistics: Computing mean, median, standard deviation, etc., (df.describe(), df.mean(), df.std())

Visualization: Seamless integration with libraries like Matplotlib and Seaborn for creating plots directly from DataFrame objects (df.plot())

### Performance and Efficiency

Pandas is optimized for performance, especially when handling large datasets. It leverages underlying NumPy arrays for fast computation, and operations can be vectorized for efficiency. Additionally, Pandas supports parallelization and optimization techniques to handle big data tasks effectively.

### Applications of Pandas

Pandas finds applications in a wide range of fields:

Data Science: Data preprocessing, feature engineering, and modeling in machine learning workflows.

Finance: Time series analysis, risk management, and financial modeling.

Academia and Research: Analyzing research data, statistical analysis, and generating reports.

Business Analytics: Market research, customer segmentation, and performance analysis.

In conclusion, Pandas' rich functionality, ease of use, and integration with other Python libraries make it a versatile tool for data manipulation and analysis. Its comprehensive capabilities empower data scientists and analysts to explore, clean, transform, and visualize data efficiently, facilitating informed decision-making across various industries.

## scikit-learn (sklearn):

A powerful machine learning library in Python that provides simple and efficient tools for data mining and data analysis. In the project, sklearn.feature\_extraction.text.CountVectorizer is used to convert text data into numerical vectors suitable for machine learning models. Additionally, sklearn.metrics.pairwise.cosine\_similarity is used to calculate similarity scores between vectors.

### Key Features and Capabilities of scikit-learn

1. **Machine Learning Algorithms**

Scikit-learn provides implementations of a vast collection of supervised and unsupervised learning algorithms. These include but are not limited to:

**Supervised Learning**: Algorithms like linear regression, logistic regression, support vector machines (SVM), decision trees, random forests, and neural networks (via integration with TensorFlow or Keras).

Unsupervised Learning: Clustering algorithms such as K-means, hierarchical clustering, and DBSCAN; dimensionality reduction techniques like principal component analysis (PCA) and manifold learning methods.

### Preprocessing and Feature Extraction

Scikit-learn offers a variety of preprocessing techniques to prepare data for machine learning models:

**Data Scaling and Normalization**: Standardizing numerical features (StandardScaler), normalizing data (MinMaxScaler), and handling outliers (RobustScaler).

**Feature Extraction**: Transforming raw data into meaningful features suitable for modeling. For text data, CountVectorizer from sklearn.feature\_extraction.text is utilized. It converts a collection of text documents into a matrix of token counts, representing the frequency of each word (token) in the document.

### Model Evaluation and Metrics

Scikit-learn provides tools to evaluate model performance and select the best models:

**Cross-validation**: cross\_val\_score function for estimating model performance using cross-validation techniques like k-fold cross-validation.

**Metrics**: Comprehensive set of metrics for classification (accuracy, precision, recall, F1-score), regression (mean\_squared\_error, r2\_score), and clustering (silhouette\_score).

### Dimensionality Reduction

Techniques like PCA (PrincipalComponentAnalysis) and manifold learning (Isomap, LocallyLinearEmbedding) help reduce the number of features while preserving important information. This is crucial for dealing with high-dimensional data and improving model efficiency.

### Model Selection and Tuning

Scikit-learn facilitates model selection and hyperparameter tuning:

**Grid Search**: GridSearchCV for exhaustively searching over a specified parameter grid to find the best parameters for a model.

**Pipeline**: Pipeline class for chaining multiple preprocessing steps and estimators into a single object, simplifying the workflow and ensuring reproducibility.

### Integration and Extensibility

Scikit-learn seamlessly integrates with other Python libraries such as NumPy, SciPy, and Pandas, enabling efficient data manipulation, computation, and visualization. It also supports custom extensions and integration with deep learning frameworks like TensorFlow and PyTorch for advanced modeling tasks.

### Applications of scikit-learn

Scikit-learn finds applications across various domains:

* 1. Natural Language Processing (NLP): Text classification, sentiment analysis, and topic modeling using techniques like CountVectorizer and TfidfVectorizer.
  2. Image Processing: Feature extraction, object detection, and image classification using algorithms like SVM and neural networks.
  3. Healthcare: Predictive modeling for disease diagnosis, patient outcome prediction, and personalized treatment recommendations based on medical data.
  4. Finance: Risk assessment, fraud detection, and stock price prediction using machine learning models.
  5. In summary, scikit-learn's comprehensive set of algorithms, preprocessing tools, and evaluation metrics make it a versatile choice for machine learning practitioners. Its ease of use, extensive documentation, and active community support further contribute to its popularity in academia, research, and industry for tackling various machine learning challenges effectively.

## NLTK (Natural Language Toolkit):

NLTK is a leading platform for building Python programs to work with human language data. It is used for natural language processing tasks such as tokenization, stemming (via nltk.stem.PorterStemmer), and other text preprocessing tasks.

### Key Features and Capabilities of NLTK:

1. **Tokenization and Text Preprocessing:**

NLTK offers robust tools for breaking down text into tokens, which are fundamental units for subsequent analysis. Tokenization can handle various languages and text formats, enabling accurate parsing and processing.

### Stemming and Lemmatization:

NLTK includes modules for stemming, such as the PorterStemmer, which reduces words to their base or root form. This process aids in normalizing text and reducing dimensionality in NLP tasks. Lemmatization, which converts words to their dictionary form (lemma), is also supported.

### Part-of-Speech (POS) Tagging:

NLTK provides functions and pre-trained models for assigning grammatical parts-of- speech (e.g., noun, verb, adjective) to words in a sentence. This information is crucial for tasks like syntactic parsing and semantic analysis.

### Named Entity Recognition (NER):

NER identifies and classifies named entities (such as names of persons, organizations, locations) within text. NLTK includes models and algorithms for NER, which is essential for extracting structured information from unstructured text data.

### Parsing and Chunking:

NLTK supports syntactic parsing to analyze the structure of sentences and parse trees. Chunking involves grouping words or tokens into meaningful chunks based on their syntactic structure, aiding in higher-level text understanding.

### Classification and Machine Learning:

NLTK integrates with scikit-learn and other machine learning libraries for tasks like sentiment analysis, text classification, and document clustering. It provides feature extraction methods and utilities to prepare text data for machine learning models.

### WordNet Integration:

NLTK includes interfaces to WordNet, a lexical database of English. WordNet allows NLTK to access synonyms, antonyms, hypernyms, hyponyms, and meronyms, enriching semantic analysis and language understanding tasks.

### Applications of NLTK:

NLTK finds applications across various domains:

* 1. Information Retrieval: Document indexing, search engines, and information extraction.
  2. Sentiment Analysis: Analyzing opinions and sentiments expressed in text data.
  3. Machine Translation: Translating text between languages.
  4. Chatbots and Virtual Assistants: Natural language understanding and response generation.
  5. Academic Research: Linguistics, computational linguistics, and cognitive science.

In conclusion, NLTK's comprehensive set of tools and libraries make it a versatile toolkit for processing and analyzing human language data. Its flexibility

## Flask:

Flask is a lightweight web framework that allows developers to build web applications quickly and with minimal code. In the project, Flask is used to create a web interface where users can input their symptoms and receive medication recommendations. Flask handles routing, request handling (request.form), and rendering HTML templates (render\_template).

### Key Features and Capabilities of Flask:

1. **Minimalistic and Lightweight:**

Flask is minimalist in design, providing only the essential tools needed for web development without imposing strict dependencies. This simplicity allows developers to have more control over the architecture and components of their applications.

### Routing and URL Mapping:

Flask uses decorators to define routes and URL endpoints, making it intuitive to map URLs to specific functions or methods within the application. For example, @app.route('/') defines the root URL endpoint.

### Templating Engine:

Flask integrates with Jinja2, a powerful and flexible templating engine. Jinja2 templates support inheritance, blocks, macros, and filters, facilitating dynamic content generation and HTML rendering based on data from Python code.

### Built-in Development Server:

Flask includes a built-in development server that supports rapid iteration and testing during application development. This server is suitable for local development and debugging purposes.

### RESTful Request Handling:

Flask provides built-in support for handling HTTP methods (GET, POST, PUT, DELETE, etc.) and RESTful API development. This makes it straightforward to create APIs that adhere to REST principles, enabling data exchange and integration with other systems.

### Extension Ecosystem:

Flask offers a rich ecosystem of extensions that extend its functionality. These extensions cover areas such as authentication (Flask-Login, Flask-Security), database integration (Flask-SQLAlchemy, Flask-Migrate), and more complex features like WebSocket support (Flask-SocketIO).

### Deployment and Scalability:

Flask applications can be deployed in various ways, including traditional hosting environments, containerized environments (e.g., Docker), and cloud platforms (e.g., AWS, Heroku). Its lightweight nature and flexibility make Flask suitable for scaling applications as they grow in complexity and traffic.

### Applications of Flask:

Flask is suitable for a wide range of web development tasks and applications:

* 1. Web Applications: Building dynamic web applications with server-side rendering and interactivity.
  2. API Development: Creating RESTful APIs for mobile apps, frontend frameworks (React, Angular), or IoT devices.
  3. Microservices: Developing lightweight and scalable microservices architecture.
  4. Prototyping and MVPs: Rapidly prototyping ideas and building Minimum Viable Products (MVPs) due to its simplicity and quick setup.
  5. In conclusion, Flask's simplicity, flexibility, and extensive ecosystem of extensions make it an excellent choice for developers looking to build web applications and APIs efficiently. Its popularity continues to grow, supported by a vibrant community and comprehensive documentation that facilitate learning and development.

## mysql-connector-python:

This library provides connectivity between Python and MySQL database. In the project, mysql.connector is used to interact with a MySQL database (cursor.execute, cursor.fetchall) for storing and retrieving medical data and user-related information.

## Other Python Libraries:

The project may also utilize other Python libraries for auxiliary tasks such as handling sessions (flask.session), managing configurations (configparser), and ensuring application security (werkzeug for password hashing).

## SQL (Structured Query Language)

Role: SQL is used to interact with and manage the MySQL database, handling data storage, retrieval, and management.

### Contributions:

Database Interaction: SQL queries (SELECT, INSERT, etc.) are used to retrieve medical information (SELECT \* FROM ...) and manage user-related data (e.g., credentials stored in the database).

## HTML/CSS/JavaScript

Role: These front-end technologies are used to create and style the web interface presented to users, ensuring a user-friendly experience.

### Contributions:

**HTML** (Hyper Text Markup Language): Defines the structure and content of web pages, integrating with Flask templates (index.html, result.html) to display recommendations and user interfaces.

**CSS (Cascading Style Sheets):** Used for styling HTML elements, ensuring visual consistency and enhancing user interface design.

**JavaScript:** Enhances interactivity and functionality on the client-side, though not explicitly mentioned, it could be used for additional client-side validation or dynamic behavior in the web interface.

## SQL Alchemy (Optional)

Role: SQLAlchemy could be optionally used as an ORM (Object-Relational Mapping) tool to facilitate database operations within the Flask application, providing a more Pythonic way to interact with the database compared to raw SQL queries.

### Contributions:

ORM: Simplifies database interactions by mapping Python objects to database tables, handling database queries and transactions in an object-oriented manner (SQLAlchemy).

## Other Python Libraries

Apart from the core languages and frameworks mentioned, additional Python libraries may be used for specific functionalities or enhancements in the project:

* + - numpy: Provides support for large, multi-dimensional arrays and matrices, offering mathematical functions to manipulate these arrays (numpy).
    - matplotlib/seaborn: Used for data visualization and plotting, although not explicitly mentioned, these libraries could be used for analyzing data trends or presenting visual insights (matplotlib, seaborn).

## Integration and Collaboration

The combination of these languages and frameworks enables the project to integrate data processing, machine learning, web development, and database management seamlessly. Python's versatility allows for rapid development and integration of various components, while SQL provides robust data management capabilities. HTML/CSS/JavaScript ensure an interactive and visually appealing user interface, enhancing user experience and engagement.

Overall, the project leverages a powerful stack of technologies to deliver a sophisticated medical recommendation system that combines data-driven insights with user-friendly web interaction, enhancing accessibility and usability in healthcare applications.

## 5.6 Source code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Medicine Recommendation System</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/6.5.2/css/all.min.css" integrity="sha512- SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV 7oQPJkl9QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<link href="[https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css](https://cdn.jsdelivr.net/npm/bootstrap%405.3.3/dist/css/bootstrap.min.css)" rel="stylesheet" integrity="sha384-

QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">

<link rel="icon" type="image/x-icon" href="{{ url\_for('static', filename='logo.ico') }}">

<style>

body {

font-family: Arial, sans-serif; background-image:

url('https://ak.picdn.net/shutterstock/videos/1018722751/thumb/1.jpg?ip=x480'); color: #333;

-webkit-backdrop-filter: blur(20px); backdrop-filter: blur(20px);

}

.title {

font-size: 3rem; margin-bottom: 20px; text-align: center; color: #89CFF0;

/\* Blue title color \*/

}

.subtitle {

font-size: 1.5rem; margin-bottom: 30px; text-align: center;

}

.description {

font-size: 1.2rem; line-height: 1.6;

}

.feature {

margin-top: 40px; padding: 20px;

background-color: #1b0a58;

width: 25rem; border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

opacity: 0;

/\* Initially hidden \*/ transform: translateY(20px);

/\* Initial position below \*/

transition: opacity 0.5s ease, transform 0.5s ease;

}

@media (max-width: 576px) {

.container { padding: 20px;

}

.title {

font-size: 2.5rem;

}

.subtitle {

font-size: 1.2rem;

}

}

.bg-image {

background-image: url('https://c4.wallpaperflare.com/wallpaper/834/905/121/net- neuron-connection-wallpaper-preview.jpg');

background-size: cover; background-position: center; position: fixed;

width: 100%;

height: 100%;

z-index: -1;

/\* Behind other content \*/

animation: animateBackground 40s linear infinite;

/\* Background animation \*/

}

@keyframes animateBackground { 0% {

background-position: 0% 50%;

}

50% {

background-position: 100% 50%;

}

100% {

background-position: 0% 50%;

}

}

.light,

.lead {

color: #FFF;

font-weight: 500;

}

.btn-warning {

font-weight: 500;

}

.wrap {

-webkit-backdrop-filter: blur(20px); backdrop-filter: blur(20px);

}

.feature img { width: 100%; height: 15rem; object-fit: cover;

}

.link-offset-3 { color: #c3e7f8;

}

.footer {

margin-top: 80px; width: 100%; background: #1A1A1A;

}

.flexing { display: flex;

margin-bottom: 8rem;

justify-content: space-between;

}

@media (max-width: 768px) {

.flexing {

flex-direction: column;

}

.feature { width: 100%;

}

}

.ext {

padding: 10px;

box-sizing: border-box; border-radius: 10px;

}

</style>

</head>

<body>

<div class="bg-image"></div>

<nav class="navbar navbar-expand-lg bg-body-tertiary bg-dark border-bottom border-body text-light" data-bs-theme="dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">KUCET</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs- target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="/">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/prescription">Prescription</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/register">Register</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/login">Login</a>

</li>

{% if session['user'] %}

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('logout') }}">Logout</a>

</li>

{% endif %}

</ul>

</div>

</div>

</nav>

<div class="wrap">

<div class="container pt-5 light text-center animate animated animate fadeInLeft">

<h1 class="title">Welcome to Medicine Recommendation System</h1>

<p class="lead subtitle">Enhancing Healthcare with Personalized Medicine Suggestions</p>

</div>

<div class="container light mt-5">

<div class="row">

<div class="col-md-6 animate animated animate fadeInDown">

<h2>How It Works</h2>

<p>Our system analyzes symptoms provided by users and suggests tailored medication options, improving decision-making and patient outcomes.</p>

</div>

<div class="col-md-6 animate animated animate fadeInUp">

<h2>Benefits</h2>

<ul>

<li>Receive personalized medicine suggestions based on symptoms</li>

<li>Enhanced decision-making for healthcare professionals</li>

<li>Improved patient care and treatment outcomes</li>

</ul>

</div>

</div>

</div>

<div class="container light mt-5">

<div class="row">

<div class="col-md-6 animate animated animate fadeInDown">

<h2>Get Started</h2>

<p>Interested in trying out our medicine recommendation system? Sign up to receive early access!</p>

<a href="/register" class="btn btn-warning btn-lg">Sign Up</a>

</div>

</div>

</div>

<div class="container mt-5 text-light">

<div class="row">

<div class="col-md-6 mb-4 ext animate animated animate fadeInUp">

<h2>Our Approach</h2>

<p class="description">

We combine machine learning algorithms, natural language processing (NLP), and medical expertise to develop a robust system for recommending medicines based on symptoms.

</p>

</div>

<div class="col-md-6 mb-4 ext animate animated animate fadeInUp">

<h2>Data Privacy</h2>

<p class="description">

Ensuring patient confidentiality and data security is our top priority. We adhere to strict privacy policies and use anonymized data for training our models.

</p>

</div>

<div class="col-md-6 mb-4 ext animate animated animate fadeInUp">

<h2>Collaborations</h2>

<p class="description">

We collaborate with healthcare professionals, researchers, and institutions to continually improve our system's accuracy and applicability in real-world scenarios.

</p>

</div>

<div class="col-md-6 mb-4 ext animate animated animate fadeInUp">

<h2>User Feedback</h2>

<p class="description">

We value user feedback and continuously refine our system based on user experiences and suggestions. Your input drives our innovation.

</p>

</div>

</div>

</div>

<div class="container mt-5 flexing">

<section class="feature bg-dark text-light animate animated animate fadeInDown">

<h2>Importance of Personalized Medicine</h2>

<p class="description">

Personalized medicine revolutionizes treatment by tailoring options to individuals, enhancing healthcare and patient outcomes. Our tool leverages data-driven recommendations, playing a pivotal role in advancing this evolution within healthcare delivery. <img src="{{ url\_for('static', filename='heart.jpg') }}" alt="Healthcare" class="img-fluid mt-3">

</section>

<section class="feature bg-dark text-light animate animated animate fadeInDown">

<h2>Advancements in Healthcare</h2>

<p class="description">

Integrating machine learning and natural language processing in healthcare leads to efficient diagnosis and treatment, revolutionizing patient care standards. Our project contributes to this advancement by facilitating accurate and swift medication recommendations.

</p>

<img src="https://images.unsplash.com/photo-1516549655169- df83a0774514?crop=entropy&cs=tinysrgb&fit=crop&fm=jpg&h=600&ixid=MnwxfDB8MX xyYW5kb218MHx8ZnV0dXJlLG1lZGljYWx8fHx8fHwxNzEzMzU1NTE2&ixlib=rb- 4.0.3&q=80&utm\_campaign=api- credit&utm\_medium=referral&utm\_source=unsplash\_source&w=800" alt="Future Technology" class="img-fluid mt-3">

</section>

<section class="feature bg-dark text-light animate animated animate fadeInDown">

<h2>Future Roadmap</h2>

<p class="description">

Join us in shaping the future of personalized medicine with our state-of-the-art recommendation system. Utilizing cutting-edge technologies, we deliver real-time insights and seamless user experiences, revolutionizing personalized healthcare delivery and advancing patient outcomes. </p>

<img src="{{ url\_for('static', filename='neuron.jpg') }}" alt="Medical Technology" class="img-fluid mt-3">

</section>

</div>

<div class="container light mt-5">

<div class="row">

<div class="col-md-12 animate animated animate fadeInUp">

<h2>Understanding Personalized Medicine</h2>

<p class="description">

Learn about the concept of <a rel="noopener" class="link-offset-3" href="https://[www.ncbi.nlm.nih.gov/pmc/articles/PMC6095726/](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC6095726/)" target="\_blank">personalized medicine</a> and how it is transforming healthcare by tailoring treatment options to individual patients based on their unique characteristics and needs.

</p>

<h2 class="mt-5">Machine Learning in Healthcare</h2>

<p class="description">

Discover how <a rel="noopener" class="link-offset-3" href="https://[www.ibm.com/watson/health/ai-in-healthcare/](http://www.ibm.com/watson/health/ai-in-healthcare/)" target="\_blank">machine learning algorithms</a> are revolutionizing diagnosis, drug discovery, and patient care in the medical field, leading to more efficient and accurate healthcare solutions.

</p>

</div>

</div>

</div>

<div class="container-fluid text-light py-5">

<div class="container" data-bs-theme="dark">

<div class="row justify-content-center">

<div class="col-lg-4 col-md-6 mb-4">

<div class="card border-0 shadow-sm">

<i class="fas fa-dna text-danger fa-4x text-center mt-4"></i>

<div class="card-body text-center">

<h5 class="card-title">Precision Medicine</h5>

<p class="card-text">Discover how precision medicine tailors treatments to individual patients based on genetic, environmental, and lifestyle factors.</p>

<a rel="noopener" href="https://[www.cancer.gov/about-](http://www.cancer.gov/about-) cancer/treatment/types/precision-medicine" class="btn btn-secondary" target="\_blank">Learn More</a>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6 mb-4">

<div class="card border-0 shadow-sm">

<i class="fas fa-hospital-alt text-info fa-4x text-center mt-4"></i>

<div class="card-body text-center">

<h5 class="card-title">Future of Healthcare</h5>

<p class="card-text">Explore how advancements in AI and machine learning are shaping the future of healthcare delivery and personalized treatment.</p>

<a rel="noopener" href="https://[www.who.int/health-topics/precision-](http://www.who.int/health-topics/precision-) medicine" class="btn btn-secondary" target="\_blank">Discover Now</a>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6 mb-4">

<div class="card border-0 shadow-sm">

<i class="fas fa-user-md fa-4x text-success text-center mt-4"></i>

<div class="card-body text-center">

<h5 class="card-title">Patient-Centered Care</h5>

<p class="card-text">Learn how personalized medicine improves patient outcomes and enhances the quality of healthcare delivery.</p>

<a rel="noopener" href="https://[www.mayoclinic.org/precision-](http://www.mayoclinic.org/precision-) medicine" class="btn btn-secondary" target="\_blank">Read More</a>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="container-fluid footer text-light py-5">

<div class="container">

<div class="row">

<div class="col-md-6">

<h2>Our Vision</h2>

<p>

Empowering healthcare professionals with cutting-edge technology to revolutionize personalized medicine and enhance patient care globally.

</p>

</div>

<div class="col-md-6">

<h2>Contact Us</h2>

<p>Have questions or feedback? Feel free to reach out to us!</p>

<a href="<mailto:info@example.com>" class="btn btn-info">Contact</a>

</div>

</div>

</div>

</div>

</div>

<script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js](https://cdn.jsdelivr.net/npm/bootstrap%405.3.3/dist/js/bootstrap.bundle.min.js)" integrity="sha384- YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.9.1/gsap.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script>

// GSAP animations triggered when elements enter viewport

$(document).ready(function() {

$('.animate animated').each(function() { var waypoint = new Waypoint({

element: this, handler: function() {

$(this.element).addClass('animate fadeInUp');

$(this.element).css('opacity', '1');

$(this.element).css('transform', 'translateY(0)');

},

offset: '75&'

});

});

});

gsap.from(".card,.feature", { duration: 1,

y: 50,

opacity: 0,

stagger: 0.2,

delay: 1,

ease: "power3.out",

});

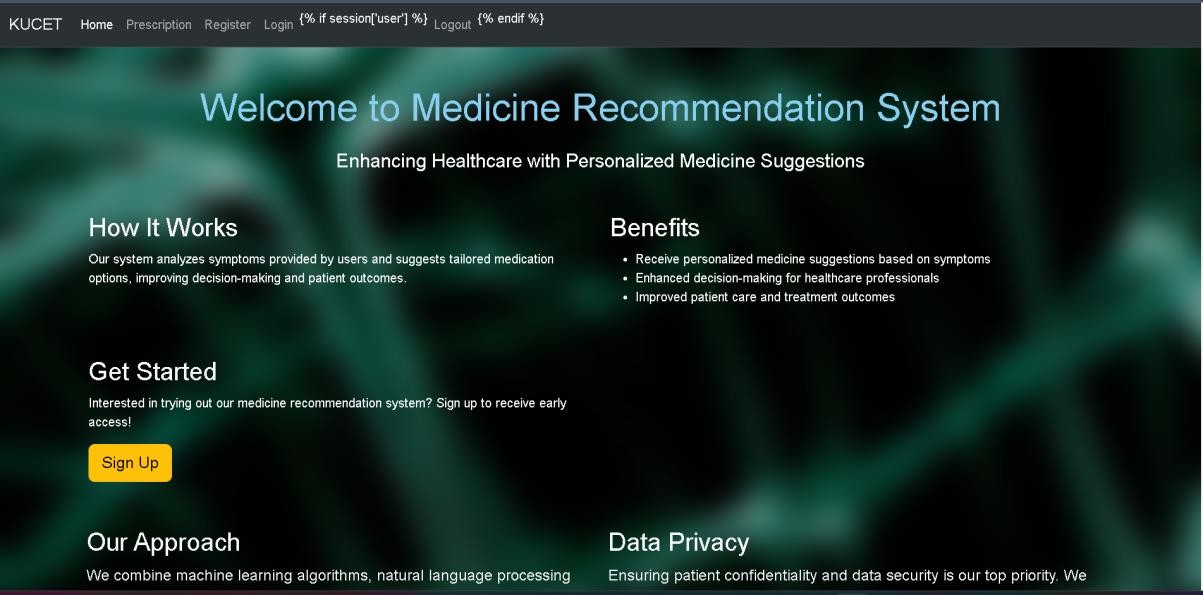
</script>

</body>

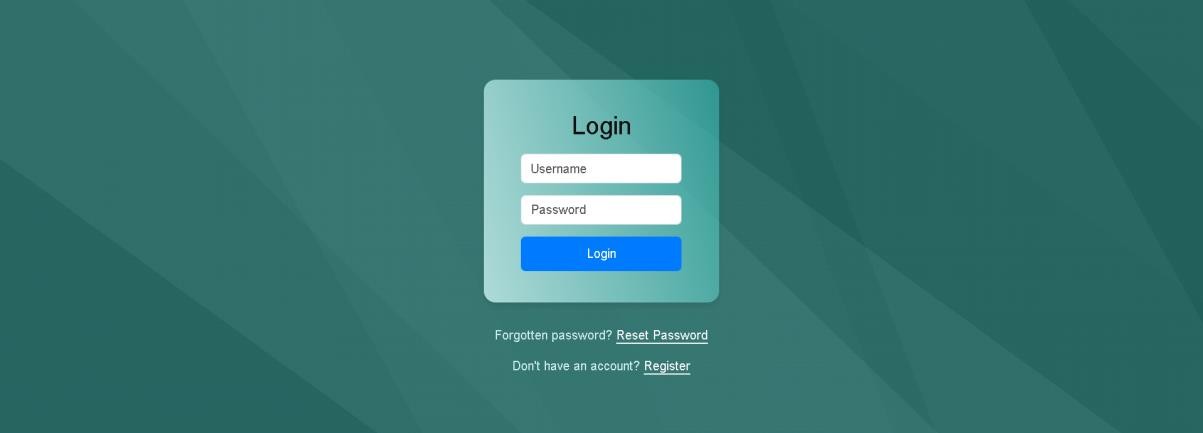
</html>

# CHAPTER 6 RESULTS

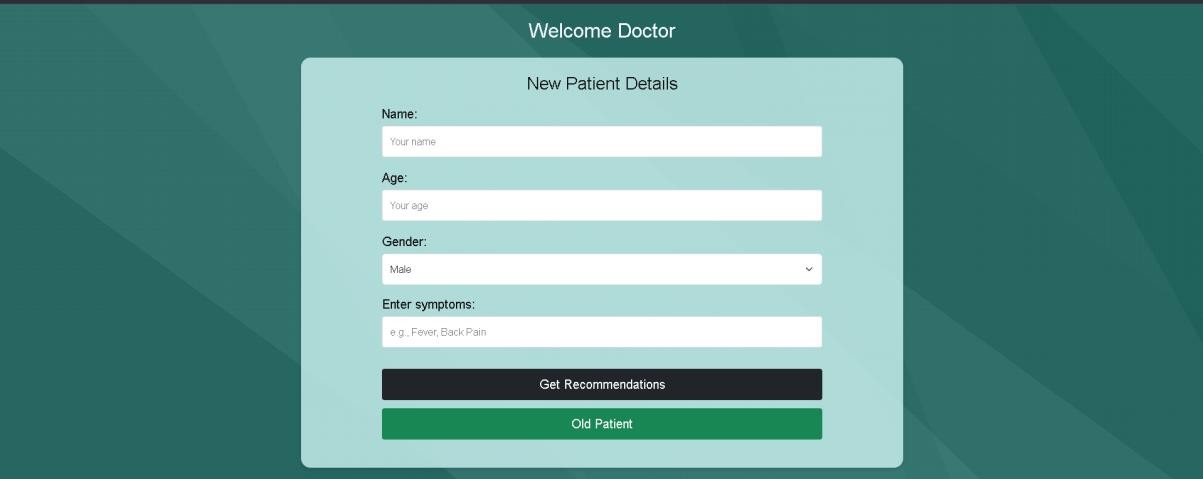
## Home page



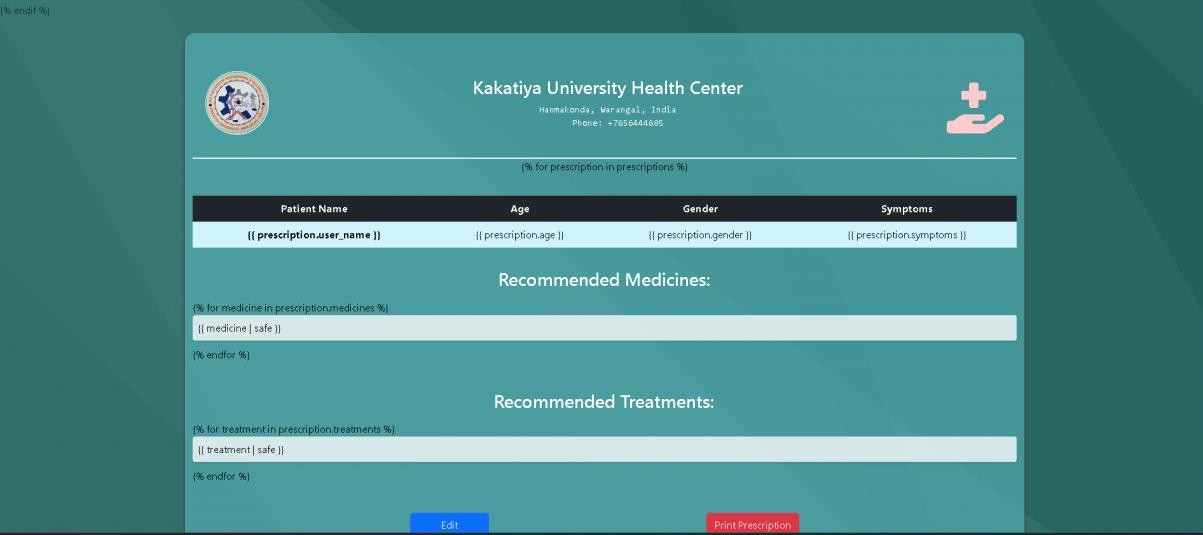
* 1. **Login page**



## Prescription page



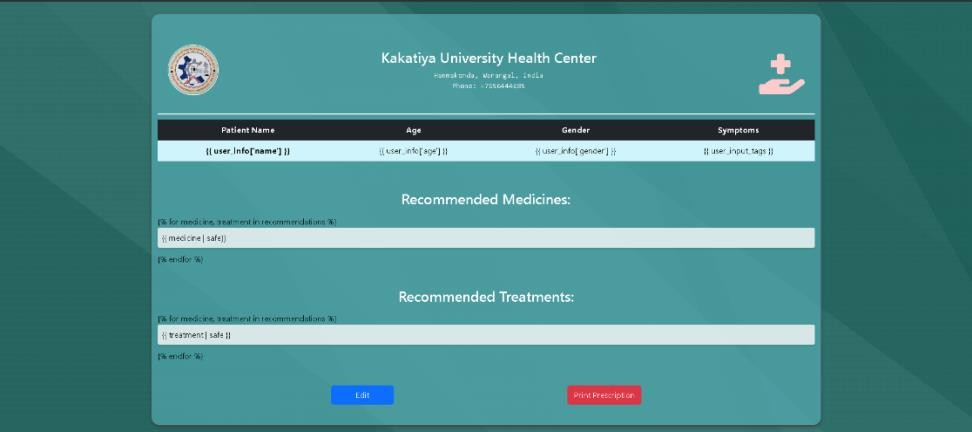
* 1. **Patient History page**



## Registration page



* 1. **Result page**



# CHAPTER 7 TESTING

## TESTING White Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white- box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g.in-circuit testing (ICT).

White-box test design techniques include:

* + - Control flow testing
    - Data flow testing
    - Branch testing
    - Path testing
    - Statement coverage
    - Decision coverage

White-box testing is a method of testing the application at the level of the source code.

The test cases are derived through the use of the design techniques mentioned above: control flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage. White-box testing is the use of these techniques as guide lines to create an error free environment by examining any fragile code. These White-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on. These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment. The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be.

## Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher-level testing, but canal so dominate unit testing as well

**Test procedures**

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to doubt is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

### Testcases

### Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests use arepri marily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output without any knowledge of the test object's internal structure.

### Test design techniques

Typical black -box test design techniques include:

* + - Decision table testing
    - All-pairs testing
    - State transition tables

# CHAPTER 8 CONCLUSION

## 8.1CONCLUSION

project integrates various technologies and methodologies to create an effective medical recommendation system. By leveraging Python's pandas for data manipulation, scikit-learn for machine learning, and NLTK for natural language processing, the project ensures efficient data preprocessing, feature extraction, and similarity computation. The system reads and cleans medical data, transforms text data into numerical vectors using Count Vectorizer, and applies cosine similarity to match user-provided symptoms with relevant medications. Flask is employed to create a user-friendly web interface, allowing users to input symptoms and receive medication recommendations conveniently. Additionally, MySQL database interaction supports user registration, authentication, and personalized experiences.

This project stands out by combining robust data processing and machine learning capabilities with seamless web integration, ensuring both functionality and accessibility. It provides a comprehensive solution that aids users in identifying suitable medications based on their symptoms, enhancing the overall healthcare experience. The system’s modular architecture allows for scalability and further enhancements, ensuring its relevance in evolving medical and technological landscapes. Overall, the project exemplifies the integration of data science and web technologies to deliver a practical and impactful tool in the healthcare domain**.**

# CHAPTER 9: FUTURE ENHANCEMENTS

## 9.1 FUTURE ENHANCEMENTS

Future enhancements for this medical recommendation system can include the integration of more advanced natural language processing techniques, such as deep learning models for better understanding and handling of complex medical terminologies and user inputs. Incorporating real-time data from medical databases and electronic health records (EHRs) can improve the accuracy and relevance of recommendations. Adding a feedback loop where users can rate the effectiveness of the suggested medications can help refine the system over time. Expanding the system to include multi-language support will make it accessible to a broader audience. Additionally, implementing robust security measures and compliance with healthcare regulations like HIPAA will ensure data privacy and protection. Integrating with telemedicine platforms to offer direct consultations with healthcare professionals can further enhance user experience and trust in the system. These improvements will make the recommendation system more comprehensive, user-friendly, and reliable.

## References

DuBois, P. (2008). MySQL (5th ed.). Addison-Wesley Professional.

García, S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. Springer. Grinberg, M. (2018). Flask Web Development (2nd ed.). O'Reilly Media.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. Science, 349(6245), 261-266.

Huang, A. (2008). Similarity measures for text document clustering. Proceedings of the sixth New Zealand computer science research student conference (NZCSRSC).

Little, R. J., & Rubin, D. B. (2019). Statistical Analysis with Missing Data. Wiley.

Patel, V., Shortliffe, E. H., & Stefanelli, M. (2015). Cognitive Informatics in Health and Biomedicine: Understanding and Modeling HealthCare (2nd ed.). Springer.

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.

Zhang, Y., et al. (2017). Drug recommendation towards safe polypharmacy. In Proceedings of the 2017 SIAM International Conference on Data Mining (pp. 423-431). SIAM.