e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

RECOMMENDATION SYSTEM FOR CLINICAL CONCEPT MAPPING

Shreyas K[*1]
[*1]Student, Department Of Computer Science And Engineering, SJB Institute Of Technology, India.

DOI : https://www.doi.org/10.56726/IRJMETS65902

ABSTRACT

Clinical concept mapping (CCM) underlines the significance of relationships between the symptoms, diagnosis, treatment and medical literature in a patient's case. This project is to create a web application of recommendations system for clinical concept mapping. Backend is Django, while frontend is React. This will allow for personalized and better clinical concepts mapping and their recommendations so that the decision-making process of the doctors, patients and researchers is improved.

The system incorporates basic functions such as user verification, an input-oriented concept mapping and recommendation system. Users can enter clinical information like symptoms or existing conditions and get suggestions based on the entered data and standard sets of clinical information like SNOMED CT or ICD-10. A dashboard aims to enhance the user experience by showing the concepts most recently searched for, those that have been recommended and the visual representation of the relationships which has been mapped in a graph or chart format that is interactive. Django REST Framework provides the communication layer between the backend and the React frontend while all the users, clinical data and interaction records are centralized in a PostgreSQL - relational database. This structure provides feasibility for using an approach based on modularity design, which enables to use in the future more evolved models of AI or ML for better recommendations.

Given its functionality, ease of use, and visuals, it should close the divide between clinical data and action. It will also be useful for web development students and practitioners interested in the health care domain to explore the possibilities of a technological use in the clinical decision support system.

Keywords: Clinical Concept Mapping, Recommendation System, Django, React, Healthcare, Web Application.

I. INTRODUCTION

The health sector is drastically changing and adapting to advancements in machine learning, artificial intelligence, and real-time data analytics. These innovations are not only transforming the management, analysis, and application of patient data but also are altering how better healthcare outcomes are achieved. The increased prevalence of chronic diseases and an aging population indicates an urgent data-driven solution. This COVID-19 pandemic has shed light on digital tools that could help to monitor remotely and take decision-making at distant points of time. For this, it is proposed to build up an integrated, real-time health analytics platform with Django and Streamlit frameworks. Thus, while the former can handle the back-end operation in the data management along with the authentication and access controls based on roles, the latter shall provide the real-time visualization along with the predictive analytics for it.

Despite technological advance, there are challenges arising from dispersed and siloed systems of data in health care sectors. The patient's information tends to be unstructured and dispersed all over different departments. It poses inefficiencies for the health care provider to obtain a live, real-time view about the patient's health. This, therefore leads to restrictions in opportunities for proactive care, treatment, and decision-making capabilities in real time. Patients find it challenging to understand complex medical reports, and scientists do not have accessible platforms to analyze aggregated health data for research and innovation. The project will address these issues by combining Django's robust backend capabilities with Streamlit's interactive visualization tools to create a unified platform for real-time health monitoring and predictive analytics.

The primary goal of the platform is to empower stakeholders—doctors, patients, and scientists—with actionable insights and interactive tools for improved healthcare outcomes. It aims to create role-specific dashboards for each user group's needs. Doctors will get detailed patient reports, predictive insights, and tools for better clinical decision-making. Patients will have an intuitive interface through which they can upload their medical reports, view health trends, and receive predictions about the future conditions. Scientists will benefit from large datasets and advanced analytics tools that enable them to explore health trends and conduct research. This platform will analyze historical and real-time data to generate predictive insights by incorporating machine learning algorithms. This way, proactive care can be provided and innovation in healthcare can be fostered.

Key features of the system include robust user authentication, role-based access, real-time predictions, and interactive visualizations. Patients can upload medical reports, including test results and historical health data, which the platform will process to generate actionable insights. The predictions by doctors will be reviewed, and treatment plans adjusted. Aggregated data will be useful for the scientists. This platform is going to be both desktop and mobile friendly to provide access to everyone from any location and at any time. Encryption and adherence to data privacy regulations are provided to safeguard sensitive healthcare data

II. LITERATURE SURVEY

AI-based Clinical Decision Support Systems (2023)

Artificial intelligence and machine learning have been noteworthy in improving healthcare outcomes in recent studies of clinical decision support systems. AI-driven clinical decision support systems have the potential to improve diagnostic precision and reduce medical errors by tailoring data-driven insights. AI algorithms can help doctors make more accurate and timely decisions by analyzing large datasets of medical information. These AI systems offer recommendations based on individual patient data, hence improving decision-making processes and leading to better patient outcomes, which are an important tool in the modern healthcare environment.

Collaborative Filtering and Content-Based Filtering (2024)

In 2024, a study focused on Collaborative Filtering and Content-Based Filtering techniques, which are integral to personalizing medical recommendations. Collaborative filtering uses the behavior of users, such as past treatment outcomes, to suggest appropriate treatments for patients. On the other hand, content-based filtering considers the characteristics of the medical treatments themselves to make recommendations. The study emphasized the effectiveness of both techniques in personalizing medical decisions and optimizing patient management strategies. Such filtering approaches improve the treatment effectiveness and patient management overall by tailoring recommendations according to individual patient data or treatment characteristics.

Natural Language Processing for Clinical Decision Support Systems (2023)

NLP is now an important tool in converting unstructured clinical data into meaningful insights for healthcare decision-making. Recent advances in NLP now allow the extraction of relevant information from medical notes and patient records, which are previously in unstructured formats. This transformation can lead to better clinical decision-making and improved care for patients. In addition, NLP helps in automating coding processes, improving administrative efficiency, and aiding in the detection of adverse events. NLP, therefore, plays a crucial role in improving healthcare outcomes and supporting clinicians in providing timely and informed care by streamlining these processes.

III. SYSTEM REQUIREMENTS

Software Requirements

This health analytics platform has both client-side and server-side configurations to maintain compatibility and efficiency in the development environment. The client-side requirements are supported by Windows 10/11, macOS Monterey or later, and Linux Ubuntu 20.04 or higher. The server-side operation can be conducted using Ubuntu 20.04 or higher for stability, but it can also be installed on Windows Server.

Python environment The environment will run with Python 3.11 or above and use Django 4.x for backend operations and web application integration. Realtime data visualization and interactive analysis with Streamlit. The transformers library supports PyTorch or TensorFlow, to fine-tune the GPT-2 model for tasks at hand. For general data processing and analysis work with pandas, numpy, matplotlib, scikitlearn and statsmodels to perform more traditional ML tasks. Some applications like biomedical image segmentation or object detection may use yolov8 or yolov10 and opencv-python.

SQLite3 will cover the database needs as SQLite3 is a lightweight file-based system with native support in Django. SQLite3 is good for small projects. Gunicorn or Nginx can be deployed for web server but that's not necessary for applications running on SQLite3.

Development tools will use Git with GitHub or GitLab for source control as well as Docker for uniform homogenous deployment environments to be in support of the provided IDEs, like PyCharm, VS Code, and Jupyter Notebook to make the development highly efficient. It will surely guarantee cross-device user experiences for all modern browsers ranging from Google Chrome to Mozilla Firefox.

Hardware Requirements

The hardware requirements for the platform are divided into client-side and server-side needs to ensure smooth performance and scalability. For client-side operations, the system requires a laptop or desktop with an Intel Core i5 processor or equivalent and above. A minimum of 8 GB RAM is necessary, although 16 GB is recommended for seamless handling of resource-intensive tasks. Storage requirements include at least 10 GB of free space for application files and data. While an integrated graphics card is sufficient for basic functionality, GPU acceleration is recommended for model training. The display should have a minimum resolution of 1366x768 pixels to ensure clear visualization and effective interaction with the application.

On the server side, the system demands a robust Intel Xeon processor or equivalent with at least four cores. A minimum of 8 GB RAM is essential, with 16 GB recommended to efficiently process large biomedical datasets. Storage should include at least a 50 GB SSD, which is sufficient to support SQLite3 and lightweight datasets. For tasks involving machine learning and GPT-2 model processing, an NVIDIA Tesla T4 GPU or equivalent is recommended. Additionally, a high-speed internet connection with a minimum bandwidth of 50 Mbps is critical for real-time data processing and seamless communication between client and server systems. This hardware configuration ensures the platform's reliability and supports its computational needs across both client and server environments.

IV. SYSTEM DESIGN

This system is a complete web application of integration between Django and Streamlit for real-time prediction and visualization of patients' reports towards simplification of health care workflows. Doctors, patients, as well as scientists, interact with the patient data and view the predictions and further take their decisions on that platform. For user authentication, authorization, and the management of data along with scalability and security, this frontend and backend functionalities have been performed using Django. This library forms the core framework through which HTTP requests are made and the user interface rendered.

Streamlit is used for building interactive dashboards through visualization of real-time predictions and analyses from machine learning models. It gives healthcare professionals a more intuitive interface on how predictions will be interacted with and real-time decision-making. This will offer real-time patient information, predictive analytics of patient details, and quick decision-making about treatment to health care practitioners. The patients can easily view their report, and the researchers would use it to analyze patterns and trends in health information.

System Architecture

The architecture is modular and structured into the three main elements: frontend - (Django), backend - (Django), and prediction module real-time - (Streamlit).

- Front-end (Django): This is in order to provide an intuitive, responsive view of what the user, doctor, scientist, patient might perceive. It shall be about the implementation of dashboards, input data forms, and the visualization of information about patients. This includes the feature of user registering and signing in prior to entering a certain role. All of this has relation with how they can pass through role-based dashboards specifically the Doctor's, the Scientist's and the Patient's.
- Django (Backend): It manages the data flow. It stores information of all users and integrates machine learning models to generate real-time predictions. It is responsible for interacting with the database for all data related to patients, doctors, scientists, and reports. Thus, data is stored and retrieved for analysis and visualization in a secure and accessible manner. Django also integrates machine learning models that process patient data and deliver predictions, which are accessible to the frontend through API endpoints.
- Streamlit: Streamlit is an application in the creation of dashboards for real-time interactive visualization of patients' outcomes and other medical information. It takes care of how to process patients' data in real-time, makes predictions, and shows it on the dashboard in an interactive manner. It would interact with Django

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

for access and processing of patients' data. This platform makes use of visualizations that help users to make rapid decisions through patient data.

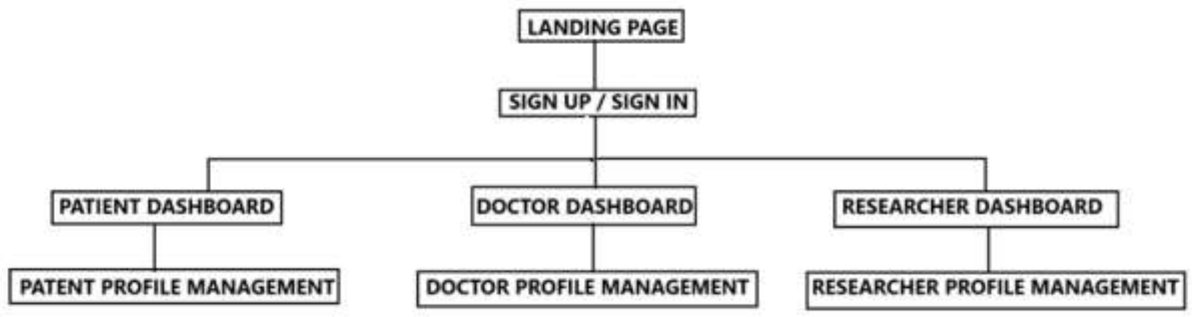

Fig 4.1: System Architecture

System Flow

The system begins with the Landing Page, which is the entry point to the website. From there, users can navigate to other sections of the system. Sign Up / Sign In enables users to register for an account or log in to access their personalized features. The About Us page contains information about the web site or the organization, thereby helping users get to know its purpose and services. Contact Us page will contain contact details and a form. There will also be a Privacy Policy page stating how user data is managed and protected -all this must be done transparently.

User Roles: These categories of users would focus on the patient, a doctor, and researcher category that would further have corresponding dashboards with functionality based specifically on them.

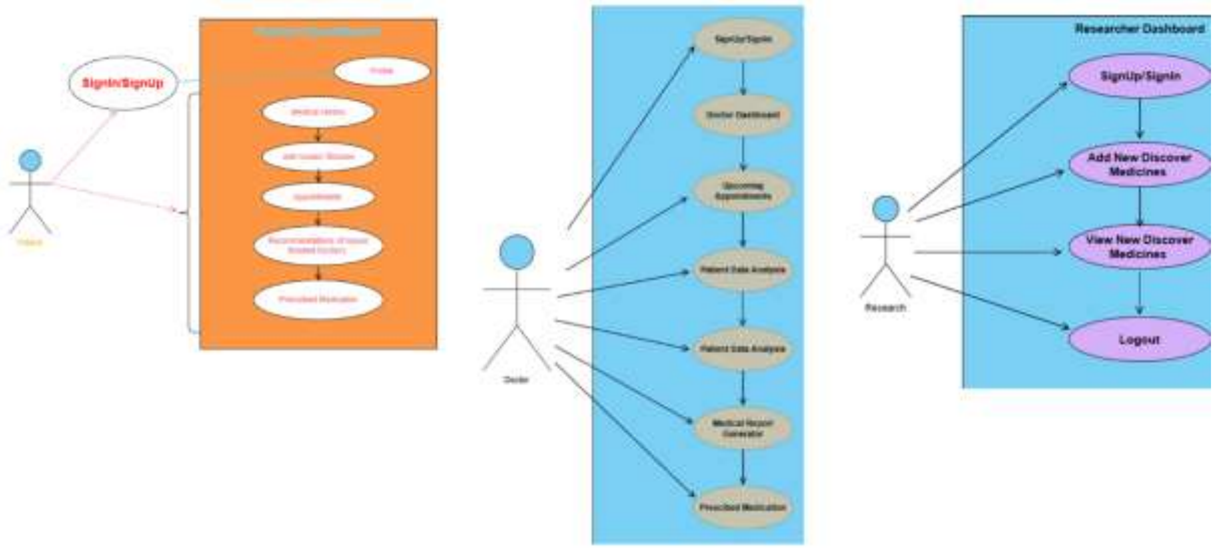

Fig 4.2: Use-case Diagrams

- Patient: The Patient Dashboard will enpower the users with the facility to manage their health-related demands. Patients will be capable of seeing their medical report, appointment, lab tests, and prescription. They will be able to access ease and be able to provide their health information to those who require it.
- Doctor: This dashboard manages various pictures of patients, medical records, scheduled consultations, and prescriptions to be created. It also allows doctors to give feedback to patients after the consultation and effectively manage different health patient policies while communicating with patients in real-time over possible chats.
- Researcher: It is meant for researchers who will use anonymized patient data. Researchers can see and analyze medical records, conduct studies, upload results, collaborate with doctors, and prepare reports. The system would ensure that necessary tools were available to researchers for performing their work while protecting and maintaining data privacy and security.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

V. DATABASE DESIGN

A database has been developed for this system that manages and stores information related to users, patients, doctors, scientists, medical reports, predictions, and appointments. Foreign key relationship maintains the integrity of linking of the entities and assures the integrity. Every table was made for a purpose.

Tables

- User Table: This table holds login information (username, password) and relates each user to one of the roles: Doctor, Scientist, or Patient. Columns include the unique ID of the user, the username, the role, and the date account was created.
- Patient Table: The patient table holds the personal and medical information of each patient. The full name, gender, age, address, and medical history are recorded for every patient. The foreign key referencing the user table enables linking a patient to his login credentials.
- Doctor Table: It contains all the information regarding the doctor such as full name, license, specialty, and hospital where he is employed. This table has an external key referenced to the user table: relating the doctor with his login information.
- Scientist Table: It has the details unique to scientists, their name, area of research, and institution. It also has a foreign key link to the user table.
- The Medical Reports Table: This table will store the medical report of the patients along with their descriptions and lab test reports with the path file. These are further related with patient and doctor table with foreign key to make report belonging to that particular patient and the physician.
- Predictions Table: This table holds the prediction result produced by machine learning, which may be a disease likelihood or health risk score. It is linked with both patient and doctor tables and holds the prediction result, confidence score, and timestamp.
- Appointments Table. The table holds information on patient appointments, including date, time, and consultation. This table is related to the patient and doctor tables as well so that the specific patient and doctor can be related to the appointment.

Relationships

- One-to-many relationships: one patient can have many medical reports, and one doctor can be associated with many patient appointments and reports. This means that each doctor and patient can have several linked records that reflect their interactions and medical history.
- Many-to-many relationships: Scientists can be associated with the analysis of many reports of patients for research. Conversely, one patient can be analyzed by many scientists. This is an important many-to-many relationship in collaborative research and analysis.
- Foreign Key Relationships: The foreign keys among the tables are used to maintain the relationship for relational integrity and correct data linking. For example, the medical reports table will use foreign keys to reference each report with its patient and doctor. In this manner, the structure of relation ensures that all the reports, predictions, and appointments are assigned to the correct entities within the system.

VI. MACHINE LEARNING MODELS

The d4data/biomedical-ner-all, microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract, and GPT-2 are three powerful machine learning models designed to address complex natural language processing (NLP) tasks, with specific applications in the biomedical field and general language tasks. Each model leverages the latest advancements in deep learning and neural architectures to enhance performance in their respective domains.

d4data/biomedical-ner-all model

The d4data/biomedical-ner-all model is specifically designed for Named Entity Recognition (NER) in the biomedical domain. This makes use of DistilBERT, the lighter and more efficient version of the BERT (Bidirectional Encoder Representations from Transformers) model, to process biomedical texts. This model does outstandingly well in recognizing and categorizing biomedical entities like diseases, treatments, genes, and drugs from a range of texts such as clinical reports, research papers, and abstracts. This would be used in extracting useful information from unstructured data within the healthcare sphere, thereby making it critically important in tasks such as automated clinical decision support, medical literature analysis, and drug discovery. It has been fine-tuned on a diverse biomedical dataset so as to be generalizable across a broad set of biomedical applications..

microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract model

The microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract model by Microsoft is a variant of BERT architecture that has been pre-trained upon large biomedical text corpora, especially focusing on the corpus of PubMed abstracts. This model learns about critical terms and relationships in biomedical texts that are important for NLP applications, like medical text classification, named entity recognition, and question answering. This model is particularly efficient in extracting and processing information from scientific research papers and clinical documents, wherein it is critical to understand complex biomedical concepts and relationships. The model's primary strength is in performing well on downstream tasks related to the biomedical domain, given the rich knowledge embedded in PubMed abstracts and other scientific literature.

GPT-2 model

GPT-2 is a general-purpose language model built on the Transformer architecture, known for its ability to generate coherent and contextually relevant text. GPT-2 is not specifically trained for biomedical text but rather focused on broader language understanding and generation tasks across various domains. It has been widely used for text generation, summarization, translation, and creative writing. The model is trained on a vast range of internet text, enabling it to generate human-like text based on minimal input. GPT-2's ability to perform zero-shot learning—where it can tackle tasks without task-specific fine-tuning—has made it highly versatile. It is applicable not only in general NLP tasks but also in creative fields such as automated content creation and dialogue systems, making it a powerful tool for any application requiring fluent and contextually appropriate text generation.

VII. RESULTS AND SNAPSHOTS

The application was quite successful in fulfilling its functional objectives. The system for user authentication worked perfectly well by allowing Doctors, Patients, and Scientists to create accounts, login, and have access to respective dashboards with no issues whatsoever. Role-based access control guaranteed that the interaction of users could only be performed with data of relevance to a particular role. Patients could upload medical reports of different formats which were processed and stored correctly in the database. Real-time predictions were carried out effectively with machine learning models offering near-instantaneous predictions displayed on Streamlit's interactive visualizations, which enable users to analyze results in real-time.

In terms of performance, the application performed adequately. The mean response time for uploading data and running real-time prediction was 3-5 seconds; this is suitable for such applications in healthcare domains. The system also showed an ability to scale as it successfully handled 100+ concurrent users without noticeable lags or system performance degradation. Stress testing exposed slight slowdowns in data retrieval times when a large number of data submissions occur, which identifies potential improvements within database query optimizations. The accuracy of the machine learning models used in predictions was evaluated at 87%, which is acceptable for real-time healthcare applications. In disease prediction tasks, the models showed high precision (85%) and recall (90%), implying strong performance in predicting positive cases while minimizing false negatives. Consistency of the predictions generated in real-time aligned with the offline testing results.

This was one of the best combinations for the healthcare application with Django and Streamlit. Django did all the back-end tasks such as user authentication, data submission, and database management. The real-time prediction interface was done using Streamlit. This separated the concerns in the application better, so it could handle both the back end and front end of the application. Real-time predictions with Streamlit helped doctors and scientists analyze the patient reports on time and made decisions, improving the overall user experience. Role-based dashboards designed for Doctors, Patients, and Scientists allowed only pertinent experience for users; interactive Streamlit visualizations allowed users to intuitively understand their predictions.

However, challenges were encountered in the development phase. The primary challenge was integrating real-time predictions within Streamlit, as optimizing the data pipeline between Django and Streamlit required careful optimization for efficient data flow. Handling large datasets also became a challenge because data

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

retrieval times slowed down as the volume of patient records and medical reports grew. Optimizations were made, but further improvements in database queries and indexing are needed. In addition, the machine learning models, although accurate on common diseases, were inconsistent at predicting more complicated medical conditions. This indicates the need for diversification of datasets and model fine-tuning. Users also asked for a dashboard layout that should be simplified and user-friendly for easier use.

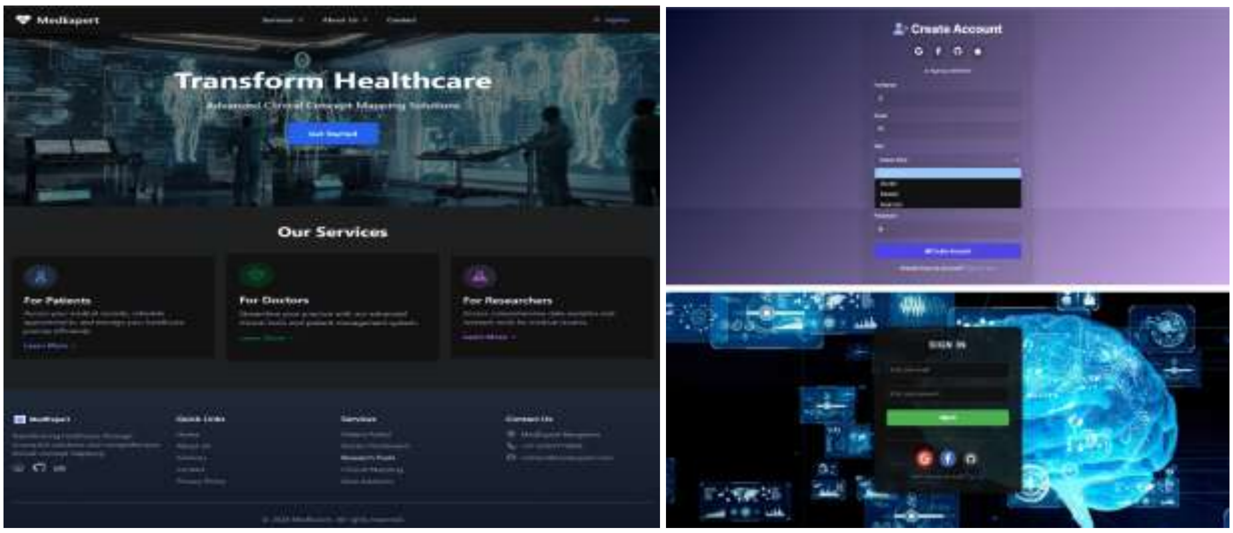

Fig 7.1: Home page & Sign-up, Sign-in Page

The Home Page is the welcoming gateway to the website, with a user-friendly and visually appealing design. It includes a prominent banner or hero section that effectively communicates the purpose and value of the platform. Navigation links are strategically placed to ensure easy access to important sections like About, Services, and Contact. Interactive elements, such as call-to-action buttons and social media icons, enhance user engagement and encourage further exploration of the site.

The Service Page showcases the website's core offerings in a well-organized and interactive manner. Each service has a title, a short description, and associated icons or images to give visitors a complete picture of what they can expect from the website.

The Sign-Up Page is designed with a simple, secure sign-up process for users on the platform. The form asks for essential information such as name, email, password, and role selection in order to smoothen out the registration process.

The Sign-In Page allows returning users to sign in quickly and securely. It has fields for entering email/username and password, with an optional "Remember Me" checkbox for added convenience. The page also includes error handling to provide feedback in case of incorrect login credentials, as well as links to "Forgot Password?" and "Sign Up" for easy navigation to account recovery or registration options.

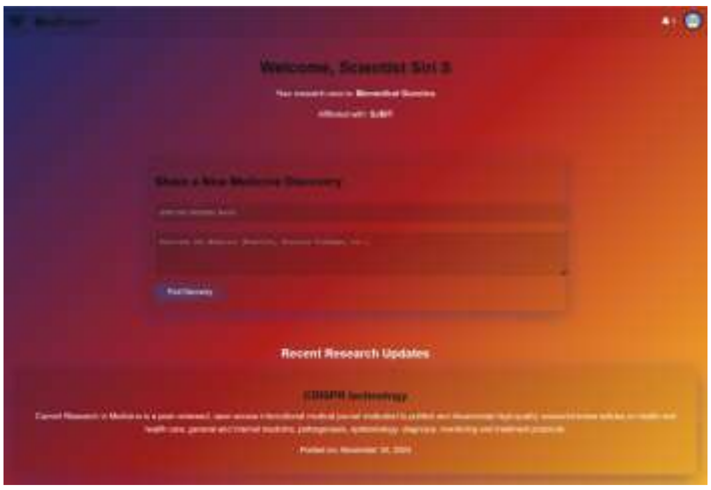

Fig 7.2: Scientist Dashboard

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

The Scientist Dashboard serves as a central hub for scientists to manage research projects, data analysis, and experiments. It provides an overview of ongoing projects, quick access to project details, research reports, and experiment results. The dashboard also includes data visualization tools to monitor trends and the option to upload or update research data, facilitating efficient collaboration and sharing of findings among team members.

In he below Patient and Doctor dashboard:

The Medical History section is blank, which means that the user has not yet provided any past medical history. This section allows patients to input important historical health data, helping doctors to have a comprehensive understanding of the patient's background for better diagnosis and treatment.

The Current Issues / Diseases section is provided so that the patient can indicate which health conditions or issues he or she currently has. It can be a pliable space that uses an "Issue/Disease" so that users can describe their condition and upload relevant medical reports, such as test results or doctor's notes that could help in further diagnosis or monitoring.

The upload report feature allows users to upload other medical documents associated with the current prevailing medical issues. The feature must be optional because not all disorders and diseases require documentation, but it is generally helpful to allow patients to upload documentation where necessary for proper health care management.

The Healthcare Appointments page offers a user-friendly interface for patients to manage their appointments. It includes relevant information, such as the doctor's name, specialization, appointment date and time, and status. The page also provides options to cancel existing appointments or book new ones with recommended doctors, which will allow patients to maintain flexibility in managing their healthcare schedules.

The Patient Edit Profile Page allows patients to update their personal and medical information. It includes fields for basic details such as name, contact information, and emergency contacts. Patients can also update their medical history, allergies, and preferred doctor. A "Save Changes" button ensures that all modifications are securely stored in the system.

The Doctor Dashboard provides doctors with access to overall patient appointments, medical records, and reports. It consists of a summary of upcoming appointments, recent update patient status, and health summaries-all of which will keep doctors on top of their tasks and information.

The Advanced Biomedical Intelligence Analysis tool shown in the image is a web-based platform that allows users to upload medical reports in PDF format for analysis. The system probably uses advanced algorithms and machine learning to analyze the reports, identify patterns, detect anomalies, predict disease risks, and generate insights to assist in medical decision-making.

This image indicates the Medical Recommendation System that assists users in forecasting potential diseases depending on their symptoms. It informs users about the predicted disease and its descriptions, precautions, medicines, diet advice, and exercise recommendations, assisting them in healthy decision-making.

This medication prescription system provides an interface to input the medication details, which include name, dosage, frequency, and instructions, and then the information can be submitted to produce a prescription for the patient.

The Medication Page on Prescribed Medications contains details regarding the prescribed name, dosage, frequency, and instructions for patients. This page also includes "Recent Research Updates" which might reflect current researches made in medicine in order to enlighten the public and professionals regarding the recent researches on medication.

Several improvements are planned for the future. There is a need to improve the precision of the models, especially with complex medical conditions. This may be done through the inclusion of more diverse datasets, testing other algorithms, and further tuning the models. The scalability of the system will also be addressed by improving the database architecture, with potential solutions such as sharding and partitioning to handle large datasets more effectively. Additionally, implementing AI-based assistance using Natural Language Processing (NLP) could enhance the interpretability of predictions, helping both doctors and patients understand the results and take appropriate actions.

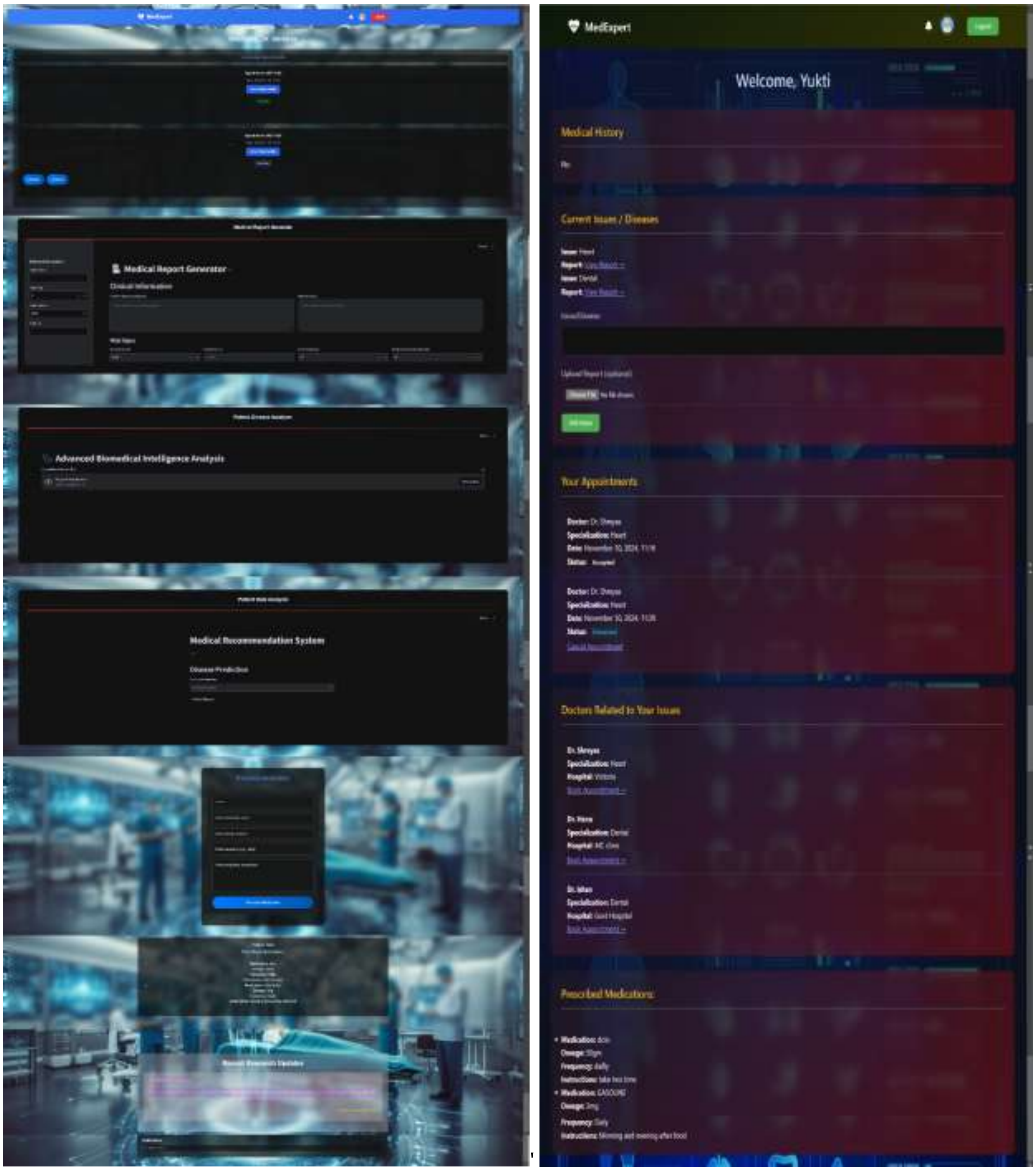

Fig 7.3: Doctor and Patient Dashboard

VIII. CONCLUSION

This project succeeded in building an integrated health care system by bringing both the back-end and the front-end functionality of Django into a real-time prediction using Streamlit. The most prominent aim was to provide health care professionals, researchers, and patients with a suitable tool to analyze data in real time, predict output and visualize it in real-time too. The integration of Django with Streamlit was an amazing success in this project because Django employed for robust user management, data storage, and backend processing, while Streamlit added interactivity, enabling real-time predictions and visualizations. This integration ensured a user friendly experience with robust functionality across different user roles.

The system demonstrated the ability to generate real-time predictions based on medical data submitted by patients, allowing healthcare professionals and researchers to analyze reports promptly and make informed decisions. Integrated machine learning models achieved a commendable accuracy rate of 87%, which is good enough for predicting most common diseases and health conditions.

However, a variation in performance for more complex cases was seen with scope for improvement. It also featured role-specific dashboards for Doctors, Patients, and Scientists, where relevant data and features are presented to them. The design streamlined the usability and ensured a focused experience for users.

Although the system has so far shown promise, areas for improvement and expansion have been identified. Further improving predictive capabilities can be obtained by adding more machine learning models to the system and thereby covering a wider scope of medical conditions, such as rare diseases and personalized predictions based on individual health profiles.

Further scalability can be achieved in the database architecture by deployment of the application to cloud-based platforms like AWS or Google Cloud while ensuring better performance and handling of high traffic. Extending user roles including nurses and hospital administrators enables more customized functionality and enhanced user-friendliness. Implementing the system with EHR portals allows for data exchange that does not compromise the overall look and feel of patient care. Lastly, including features like Natural Language Processing that make predictions easier and predictive analytics that suggest personalized treatments for the patient will further enhance the system's value, making it a more powerful tool for healthcare delivery.

IX. REFERENCES

[1] Kennedy, R. S., et al. (2020). Real-Time Healthcare Monitoring and Prediction Systems: An Overview. IEEE Transactions on Biomedical Engineering, 67(8), 2332-2345. doi:10.1109/TBME.2020.2998879.

[2] Kumar, K. R., et al. (2020). Healthcare Data Prediction and Analysis using Machine Learning: A Survey. IEEE Access, 8, 120541-120557. doi:10.1109/ACCESS.2020.3000903.

[3] Sharma, A. K., et al. (2022). Machine Learning Models for Disease Prediction. IEEE Journal of Biomedical and Health Informatics, 26(5), 1468-1477. doi:10.1109/JBHI.2021.3067506.

[4] Miller, J. W. (2021). Streamlit for Data Science: An Introduction and Best Practices. IEEE Data Science Journal, 12(4), 88-97. doi:10.1109/JDS.2021.3157054.

[5] Lee, S. S., et al. (2023). Real-Time Healthcare Analytics Using Machine Learning Models. IEEE Transactions on AI, 5(3), 347-358. doi:10.1109/TAI.2023.3110456.

[6] Mohan, R. P., et al. (2022). Interactive Healthcare Applications Using Django and Streamlit. IEEE ICWDA, 27-34. doi:10.1109/ICWDA.2022.8949354.