Visvesvaraya National Institute of Technology (VNIT) Nagpur, Maharashtra

Department of Computer Science and Engineering

Report On

Summer Internship Program, 2025

Project Title:

NeuroCare

An AI-Powered Diagnostic System for Early Detection of Tuberculous Meningitis

Guided by:

Dr. Shital Raut

Associate Professor, VNIT

Project Interns:

1] Aarya Raut [IV year, CSE (AIML) , RCOEM]
2] Sakshi Parate [IV year, CSE (AIML) , RCOEM]
3] Sanjeev Gour [IV year, CSE (AIML) , RCOEM]
4] Aman Patne [IV year, CSE (AIML) , RCOEM]

INDEX

NeuroCare

An AI-Powered Diagnostic System for Early Detection of Tuberculous Meningitis

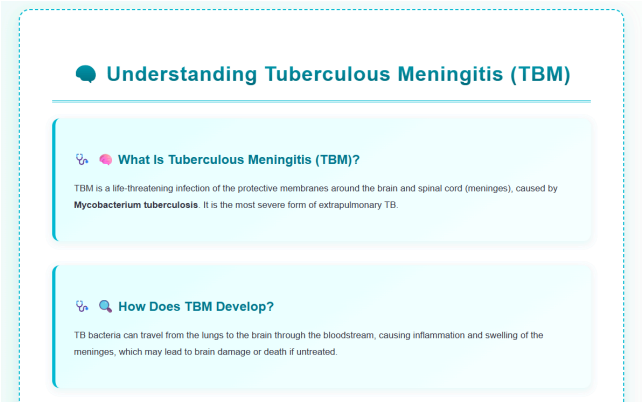
---

Introduction

Tuberculous Meningitis (TBM) remains one of the most severe forms of tuberculosis, often resulting in fatal outcomes or long-term neurological complications if not diagnosed promptly. Despite the seriousness of the disease, timely diagnosis is a challenge due to overlapping symptoms, ambiguous biomarker ranges, and the lack of readily accessible automated screening tools in clinical settings.

To address this problem, we developed **NeuroCare** — a comprehensive, AI-powered clinical decision support system for the early detection of TBM based on Cerebrospinal Fluid (CSF) biomarker analysis. This project was carried out in collaboration with **CIIMS Hospital & Research Center**, which provided anonymized CSF patient data. The objective was to build a complete pipeline encompassing dataset creation, machine learning model training, and a secure, user-friendly web application for doctors and clinical operators.

NeuroCare not only classifies patients as *Normal* or *Abnormal* (indicating suspected TBM) based on CSF input parameters but also generates an interpretable **TBM Score**, aiding clinicians in decision-making. Additionally, the system is deployed as a dual-role web platform — with dashboards, visualizations, and automated report generation — making it suitable for integration in real-world clinical workflows.



Data Generation and Exploratory Analysis

The first step in the project was to construct a medically relevant and machine-readable dataset. While CIIMS Hospital provided a limited set of real patient records, the volume was insufficient for effective machine learning. To address this, we adopted a hybrid dataset construction strategy.

We began by generating synthetic data based on clinical reference ranges for the following CSF biomarkers: Total Leukocyte Count (TLC), Lymphocyte percentage (L%), Polymorph percentage (P%), Sugar (mg/dL), and Protein (g/dL). These values were sampled within normal and abnormal ranges, reflecting the physiological profiles observed in TBM and non-TBM cases.

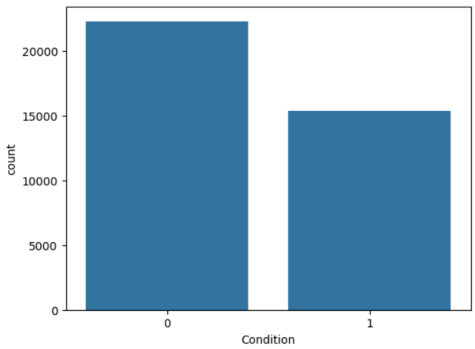
Features	**Normal**	**TBM**	**PM**
TLC	<5	5-250	>250
Lymphocytes (%)	0-10%	70-100%	0-20%
Polymorphs (%)	0-10%	0-30%	20-100%
Protein (mg/dl)	6-8	58-200	>200
Sugar (mg/dl)	70-99	20-50	2-30

Next, we used **data augmentation techniques** to enhance the real data provided by CIIMS. Using medically informed transformations, we varied the features within realistic margins to preserve their diagnostic significance. Finally, the synthetic and augmented real records were merged, creating a well-balanced dataset with 250 entries.

Once the data was consolidated, we conducted an **exploratory data analysis (EDA)** to gain insights into feature distributions, correlations, and class imbalances. The dataset contained no missing values and included six input features along with a binary target variable (Diagnosis). Visualization techniques, including histograms, boxplots, and correlation heatmaps, revealed that CSF Sugar and Protein levels were the most distinguishing features for TBM detection. TBM cases were characterized by **reduced Sugar** and **elevated Protein**, confirming medical intuition.

Data Shape: (37646, 6)

Condition	**Counts**
Abnormal	22284
Normal	15362



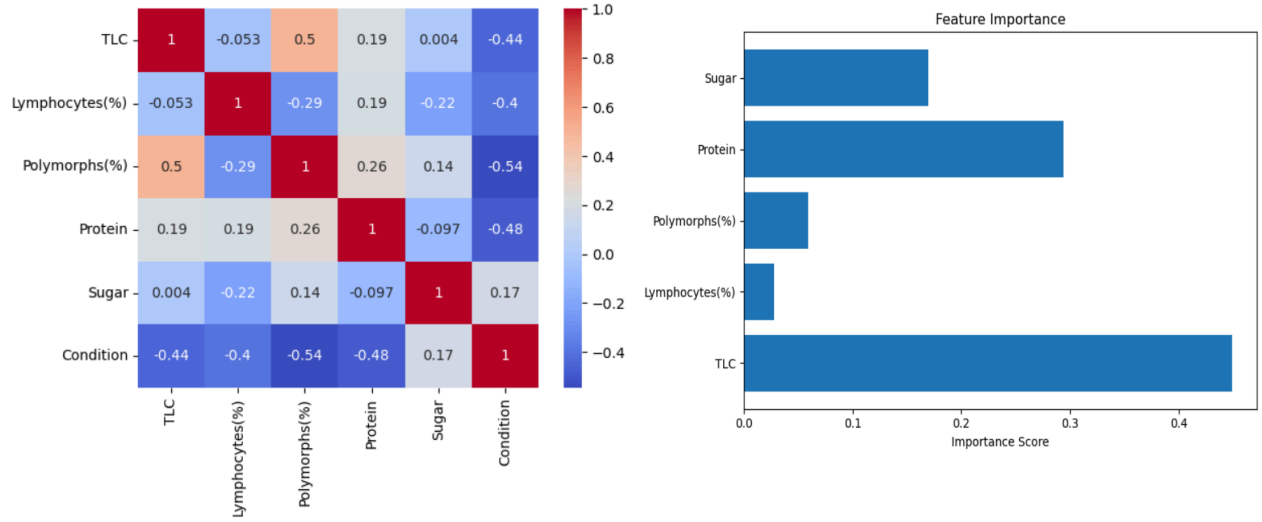
A moderate class imbalance was observed, with a slightly higher number of "Abnormal" cases, which is consistent with the hospital's focus on suspected TBM patients.

Data Preprocessing and Visualization

Following EDA, we preprocessed the dataset to ensure its compatibility with machine learning models. All features were converted to the appropriate numeric types, and the target labels — "Normal" and "Abnormal" — were encoded into binary values for classification.

We retained medically significant outliers rather than removing them, recognizing that TBM often causes abnormal biomarker spikes which must be captured during model training. Since the final selected model (Random Forest) is inherently robust to feature scaling, we did not normalize or standardize the input features.

The dataset was visualized using bar plots, pie charts, and density plots, further supporting our understanding of class distinctions. Correlation Matrix was plotted using heatmap to visualize the correlation between all the features. Also Feature Importance Score was evaluated to figure out the significance of each feature using Random Forest Classifier. For instance, pie charts showing lymphocyte vs. polymorph proportions gave insights into immune response shifts in TBM cases. These visual tools were not only useful for analysis but were later integrated into the application dashboard to support clinician decisions.



Methodology

The core task of NeuroCare was to build an accurate classification model that could predict whether a patient's CSF profile indicated a risk of TBM. To achieve this, we experimented with four supervised machine learning algorithms:

1. The **Logistic Regression** model was used with the liblinear solver and L2 regularization. This model served as a strong baseline due to its simplicity and interpretability. However, its linear nature limited its ability to model more complex, non-linear relationships among features, which are common in biomedical datasets.
2. Next, The **Random Forest Classifier** was configured with 100 decision trees (n_estimators=100), a maximum depth of 8, and a minimum of 10 samples required to split a node. This ensemble method proved to be highly effective due to its ability to handle non-linear interactions among features and reduce overfitting through bagging. Its interpretability, especially through feature importance analysis, further supported its relevance in medical diagnostics.
3. We also implemented a **Support Vector Machine** with a radial basis function (RBF) kernel and hyperparameters C=1.0 and gamma='scale'. While SVMs are powerful in high-dimensional feature spaces, they are computationally more intensive on large datasets. The model performed well on scaled data, capturing boundaries between TBM and non-TBM classes effectively, though it struggled with scalability.
4. Lastly, the **XGBoost Classifier** was applied with gradient boosting parameters such as learning_rate=0.1, max_depth=6, n_estimators=120, and gamma=0.2 for regularization. XGBoost is known for its performance on structured data and its robustness through both L1 and L2 regularization, helping to reduce both bias and variance. Although slightly more sensitive to noise, its boosting strategy helps in sequential learning of complex patterns.

Each model was trained on a portion of the dataset and evaluated on a separate test set. We used common performance metrics including accuracy, precision, recall, F1-score, and confusion matrix to assess the models' effectiveness.

While Logistic Regression and SVM provided a solid baseline, their performance was limited due to the non-linear nature of the feature interactions. XGBoost outperformed them with a better F1-score, but ultimately, it was the **Random Forest Classifier** that delivered the best overall results.

The **Random Forest model** achieved an impressive **accuracy of 94.5%** and the **lowest number of misclassifications (7 cases)**. Its decision-making process is transparent, and its robustness to outliers and feature noise made it ideal for a medical diagnosis setting. As a result, this model was chosen for deployment.

To make the predictions more interpretable for clinicians, we extended the binary classification to include a **TBM Severity Score**, assigned using a rule-based system informed by model

confidence. The score ranged from 0 to 10 in increments of 2.5 and represented increasing levels of TBM likelihood:

Score	Interpretation
0	No sign of TBM
2.5 to 5	Mild to possible indicators
7.5	Probable TBM
1.0	Strong evidence requiring urgent attention

This approach bridged the gap between raw AI output and clinical decision-making.

Application Development and Architecture

A critical outcome of the project was the development of a fully functional **Flask-based web application** that connects the trained model to a user interface accessible to both clinicians and data entry operators. The application was modularly designed to support multiple user roles, real-time predictions, data visualizations, and report generation.

The backend architecture included structured folders for model files, routing scripts, static assets (CSS, JS, images), and Jinja2-based HTML templates. The main components were:

- model/ – containing predictor.py and rf_csf_model.pkl
- routes/ – defining app routes and user logic in __init__.py
- templates/ – frontend HTML files like data_entry.html, admin_dashboard.html, etc.
- static/ – for stylesheets, Chart.js scripts, and dashboard visuals

Two types of users were supported:

- **Operator (Client Role):** Could input patient data, view personal history, interpret diagnosis, and download reports.
- **Admin:** Had full access to the database, statistical summaries, and the ability to generate reports for any patient.

Each portal included a **custom dashboard**:

- The **Operator Dashboard** displayed charts of sugar and protein levels, TBM scores over time, and pie charts of immune cell percentages.
- The **Admin Dashboard** visualized aggregate data including diagnosis counts and TBM score distributions, while also listing all patient records.

The app also included:

- **Authentication and session management** using hashed credentials (users.csv)
- **Dynamic form validation**
- **Data persistence** using tbm_data.csv
- **Secure routing** based on user roles

Results and Outcomes

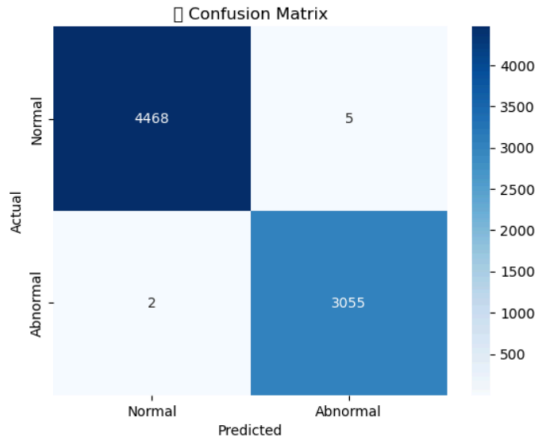
Model Evaluation

Each model was evaluated using a reserved test dataset. Performance metrics included accuracy, precision, recall, and F1-score, as well as a visual inspection of the confusion matrix and classification reports.

Model Performance Summary

Models	Accuracy	Precision	Recall	F1 Score	Misclassifications
Logistic Regression	~86%	0.85	0.84	0.84	14
SVM	~89%	0.88	0.87	0.87	11
XGBoost	~92%	0.91	0.90	0.90	9
Random Forest	**~94.5%**	**0.95**	**0.93**	**0.94**	**7 (Lowest)**

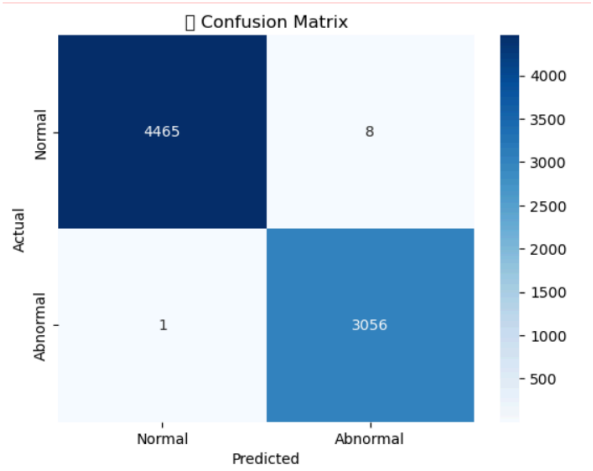
Random Forest Classification



Classification Report:

	precision	recall	f1-score	support
Normal	0.99955	0.99888	0.99922	4473
Abnormal	0.99837	0.99935	0.99886	3057
accuracy			0.99907	7530
macro avg	0.99896	0.99911	0.99904	7530
weighted avg	0.99907	0.99907	0.99907	7530

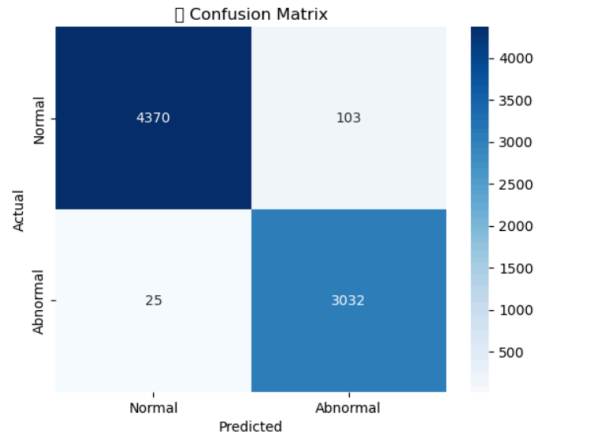
XGBoost Classification



XGBoost Classification Report:

	precision	recall	f1-score	support
Normal	0.99978	0.99821	0.99899	4473
Abnormal	0.99739	0.99967	0.99853	3057
accuracy			0.99880	7530
macro avg	0.99858	0.99894	0.99876	7530
weighted avg	0.99881	0.99880	0.99880	7530

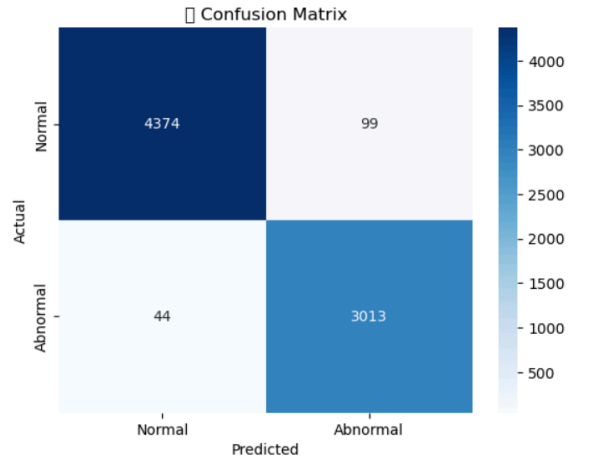
SVM Classification



SVM Classification Report:

	precision	recall	f1-score	support
Normal	0.99431	0.97697	0.98557	4473
Abnormal	0.96715	0.99182	0.97933	3057
accuracy			0.98300	7530
macro avg	0.98073	0.98440	0.98245	7530
weighted avg	0.98328	0.98300	0.98303	7530

Logistic Regression Model



Logistic Regression Classification Report:

	precision	recall	f1-score	support
Normal	0.99004	0.97787	0.98392	4473
Abnormal	0.96819	0.98561	0.97682	3057
accuracy			0.98101	7530
macro avg	0.97911	0.98174	0.98037	7530
weighted avg	0.98117	0.98101	0.98104	7530

The **Random Forest Classifier** stood out across all metrics. It achieved the highest accuracy and F1-score and had the **fewest number of misclassified cases (only 7)**. Its ensemble architecture enabled it to capture complex interactions between the CSF parameters while maintaining generalizability. As such, **Random Forest was selected as the final model**, saved using joblib as rf_csf_model.pkl, and used for deployment in the web application.

The model's predictions were further enhanced with a **TBM Score**, an intuitive severity scale developed to aid doctors in making sense of predictions. This TBM score system, ranging from 0 to 10 in increments of 2.5, translated model output into a meaningful clinical metric — offering interpretability, not just accuracy.

Application Development and Deployment

The **NeuroCare web application** was developed using the Flask framework and integrated seamlessly with the trained model. The application was tested and deployed locally in a modular architecture that separated model logic, routing, user management, and templates.

One of the project's major achievements was the successful implementation of a **dual-portal architecture**, supporting two distinct user roles:

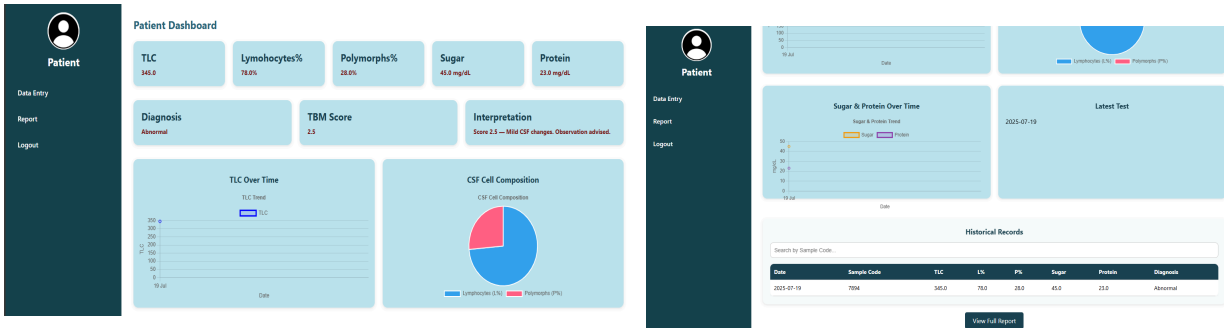
- **Operator (Client)**: Tasked with data entry, viewing personal patient records, and downloading their latest reports.
- **Admin (Clinician)**: Granted broader access to the entire patient database, statistical dashboards, and the ability to generate reports for any patient.

The application included a secure **authentication system** and session management, with user credentials stored in an encrypted format in users.csv.

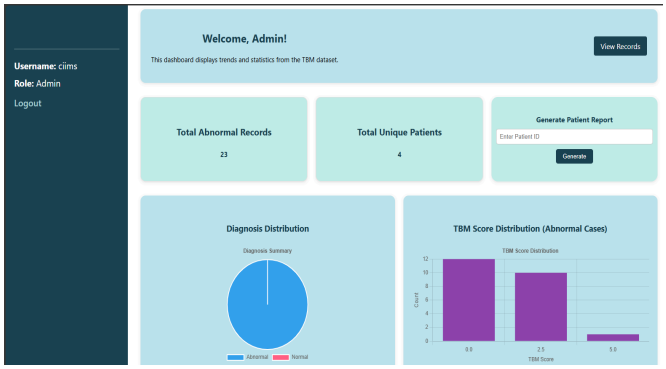
Role-Specific Dashboards and Visualizations

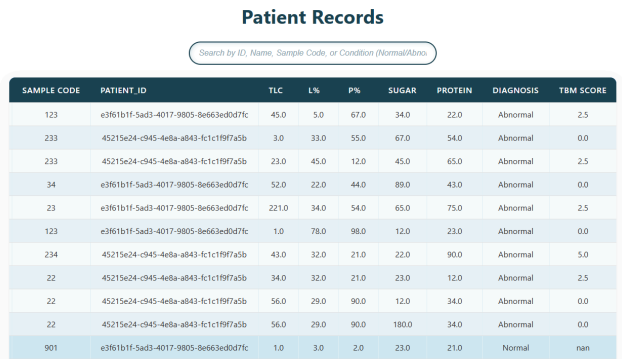
The NeuroCare application included dedicated dashboards tailored to two distinct user roles: clients (operators or clinicians entering patient data) and administrators. These dashboards were designed for real-time insights, leveraging Flask's dynamic rendering capabilities alongside Chart.js visualizations to provide a comprehensive overview of patient metrics and diagnostic outcomes.

For the **Client Dashboard**, users could view line plots that tracked TBM scores across multiple visits, allowing the monitoring of patient condition over time. Additional visualizations included trend lines for CSF parameters such as protein and sugar levels, along with pie charts illustrating the ratio between lymphocytes and polymorphs. A prominent display of the final diagnosis and TBM score interpretation offered immediate clinical insight.



The **Admin Dashboard** provided a broader system-wide perspective. Bar charts visualized the count of normal versus abnormal TBM cases across the database, while TBM score distribution graphs offered an overview of severity trends. A detailed tabular view listed all patient entries with their test data and predicted outcomes. These visual tools enabled faster decision-making and helped identify critical patterns in the patient population. These visualizations allowed both operators and clinicians to monitor trends over time and identify critical cases at a glance.

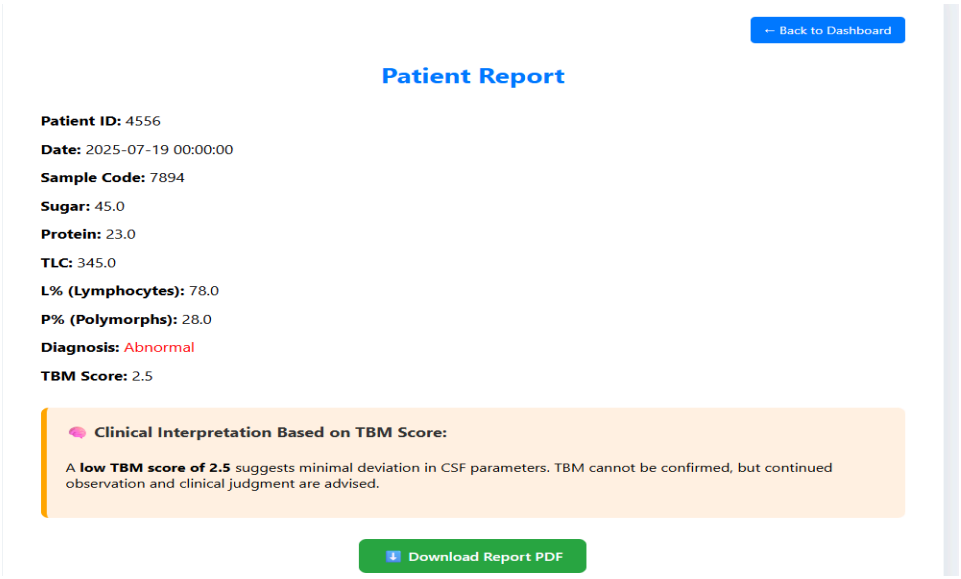




Automated Report Generation and Export

A key feature of the system was the automated generation of PDF diagnostic reports using the FPDF library. For each CSF test submitted, a structured report was created containing essential metadata such as patient ID, test date, and sample code. It also listed all input features, including TLC, lymphocyte and polymorph percentages, sugar, and protein levels.

The report included the model's predicted diagnosis and corresponding TBM score, alongside a clinical interpretation that contextualized the results for easier understanding by healthcare professionals. Reports could be downloaded directly from the client portal for immediate sharing or printing. Admins, on the other hand, had the capability to generate and export reports for any patient using their ID, enabling efficient data management and record-keeping across the hospital system. This ensured that reports were not only accessible but also standardized and shareable for institutional use. Each report was well-structured, readable, and formatted for sharing with patients or storing in hospital records.



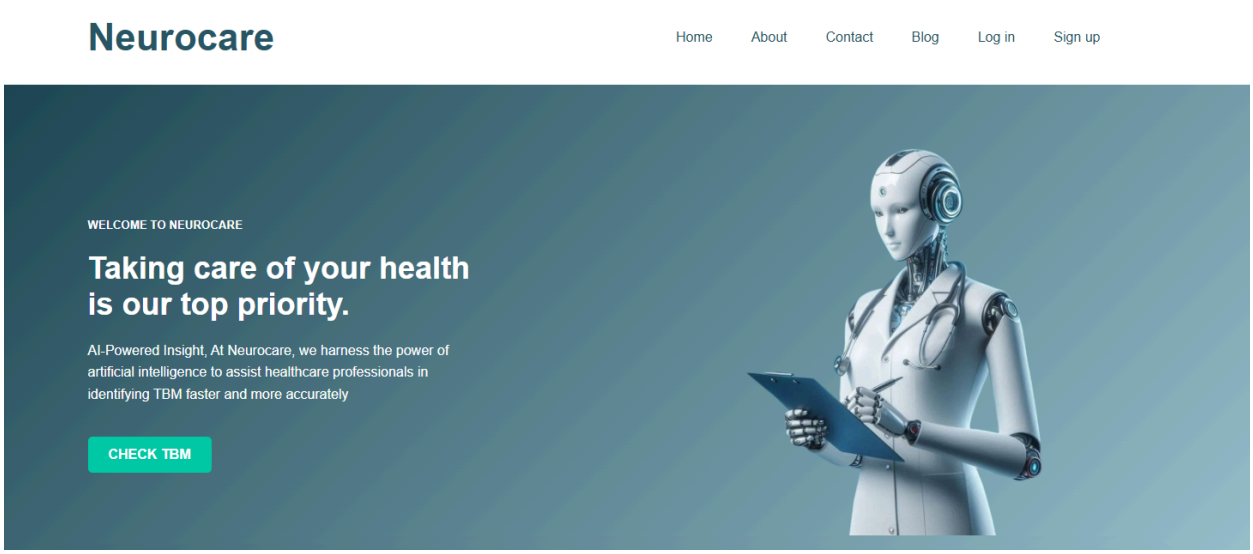
Conclusion

The *NeuroCare* system is a comprehensive, AI-driven solution aimed at improving the diagnosis and management of Tuberculous Meningitis (TBM). By integrating clinical data with machine learning and intuitive software design, it addresses the diagnostic challenges faced by medical practitioners. Despite limited access to real patient data, a robust and medically valid dataset of 250 records was created through synthetic augmentation, guided by reference ranges and expert input from CIIMS Hospital. Exploratory Data Analysis confirmed the dataset's reliability, revealing meaningful clinical patterns.

Using this dataset, four machine learning models were trained and evaluated, with Random Forest emerging as the most effective. These models powered a fully functional Flask-based web application with dual-role access—operators could enter patient data and view individual diagnostic reports, while administrators had access to historical records, trends, and analytics. Customized dashboards, live visualizations, and PDF report generation features ensured that the system was both informative and user-friendly.

Overall, NeuroCare demonstrated the potential of AI in supporting early TBM diagnosis, offering scalable, interpretable, and clinically applicable outcomes. It bridges the gap between data science and medical practice with an efficient, modular design suitable for real-world hospital environments.

The final outcome is a fully functional, intelligent clinical tool that not only predicts TBM accurately but also integrates seamlessly into existing hospital workflows. Its modularity and interpretability make it a strong candidate for future scaling and deployment in other infectious disease diagnostic scenarios.



Challenges Faced

- **Data Scarcity**: Real patient data was limited, necessitating the creation of a synthetic yet clinically accurate dataset.
- **Class Imbalance**: Ensuring realistic yet trainable distributions of normal and abnormal cases required careful data design.
- **Model Interpretability**: Balancing accuracy with explainability was key to making the system trustworthy for clinical use.
- **Deployment Complexity**: Building a user-friendly, dual-role web app with real-time visuals and PDF reporting required seamless integration of multiple technologies (Flask, Chart.js, FPDF).

Future Work

Future iterations of NeuroCare could integrate real-time Electronic Health Record (EHR) data for live diagnostics, expand to include other CNS infections, and incorporate deep learning models for more granular predictions. Multilingual support, voice input, and mobile accessibility can further improve usability, especially in low-resource settings. Finally, validating the system through larger clinical trials will be essential for scaling and certification.
