iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82561

SEWS: A Web-Based Sepsis Early Warning System Using Random Forest for Real-Time Risk Prediction

Mrs. Yashaswini R[1], Meghana M D[2], Mohith M[3], Nisarga P[4], Sowjanya S R[5]
[1]Assistant Professor, [2, 3, 4, 5]UG Students, Department of Electronics and Communication Engineering, P.E.S. College of Engineering, Mandya, Karnataka, India

Abstract: Sepsis is a life-threatening condition requiring early detection to reduce mortality. This paper presents a web-based Sepsis Early Warning System (SEWS) using a Random Forest model to predict risk from clinical parameters such as age, vital signs, and laboratory values. A React and TypeScript interface enables healthcare professionals to input data and receive real-time predictions. The Flask backend performs data processing and model inference, while Firebase ensures secure authentication and storage of patient history. The system displays color-coded risk levels, supporting quick clinical decisions and improving usability, accuracy, and scalability in real-world healthcare environments.
Keywords: Sepsis, Machine Learning, Random Forest, Early Warning System, Healthcare, Flask, Firebase, React, Real-time Prediction

I. INTRODUCTION

Sepsis is a critical medical condition caused by the body's extreme response to infection and remains a leading cause of mortality worldwide. Early detection plays a vital role in improving survival rates, as timely intervention can significantly reduce complications. Traditional diagnostic methods rely on clinical judgment and scoring systems, which may not always capture complex patterns in patient data. Machine learning techniques offer a data-driven approach to identifying such patterns and predicting medical outcomes more accurately. In this paper, a Sepsis Early Warning System (SEWS) has been developed using a Random Forest classifier. The system is implemented as a web-based application enabling real-time prediction of sepsis risk. The frontend interface allows users to input patient data, while the backend processes this data and generates predictions. The system also maintains historical records using Firebase, making it useful for continuous monitoring and analysis.

II. LITERATURE SURVEY

Boussina et al. [1] reported a real-world deployment (COMPOSER) of a sepsis deep-learning model in multiple hospitals, finding associations with improved care process measures and reduced mortality. Moor et al. [2] conducted a large international multi-centre retrospective study validating deep-learning sepsis predictors across heterogeneous ICU datasets, emphasizing cross-site validation and dataset shift issues.

Rafiei et al. [3] proposed a hybrid CNN-LSTM model (SSP) integrating vital signs, lab values, and demographics, achieving significant improvements with clinically useful prediction horizons several hours ahead of onset. Shashikumar et al. [4] developed DeepAISE, a recurrent survival analysis framework combining RNNs and interpretable hazard modeling for ICU and ED applications. Zhang et al. [5] introduced a time-aware deep-learning model encoding temporal dynamics of clinical variables, delivering top performance in the PhysioNet sepsis challenge.

III. PROBLEM FORMULATION

Sepsis is a life-threatening medical condition caused by the body's extreme response to infection, often leading to organ failure and high mortality if not detected early. Traditional scoring systems depend on manual interpretation and may fail to capture complex relationships among multiple physiological parameters, resulting in delayed or inaccurate diagnosis. Key issues include lack of real-time predictive capability, difficulty identifying hidden patterns in patient data, dependence on clinician experience, and delayed detection leading to increased mortality. This project focuses on developing a machine learning-based Sepsis Early Warning System that analyzes patient demographic details, vital signs, and laboratory values to predict sepsis risk in real time, providing accurate, reliable, and interpretable results while ensuring secure data handling.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

IV. OBJECTIVES OF THE PROJECT

1) Implement a Random Forest-based machine learning model for predicting sepsis risk.
2) Preprocess and structure clinical data for accurate predictions.
3) Design a React-based web application and develop a Flask backend API for model inference.
4) Integrate Firebase for authentication and secure data storage.
5) Provide real-time risk prediction with clear color-coded visualization.
6) Maintain historical patient data for tracking and analysis.

V. METHODOLOGY OF PROPOSED WORK

A. *System Architecture*

The architecture is designed as a multi-layered framework enabling efficient patient data flow from entry to real-time risk prediction. Four primary layers structure the system: Data Input, Communication, Processing and Machine Learning, and Application layers. The React and TypeScript frontend collects patient information including age, vital signs, and laboratory values. Built-in validation ensures data falls within medically acceptable limits.

The communication layer uses secure HTTP/HTTPS RESTful APIs with JSON data format. The Flask backend acts as the central hub for preprocessing and model inference. Firebase Firestore stores patient records and prediction results, while Firebase Authentication restricts access to authorized users.

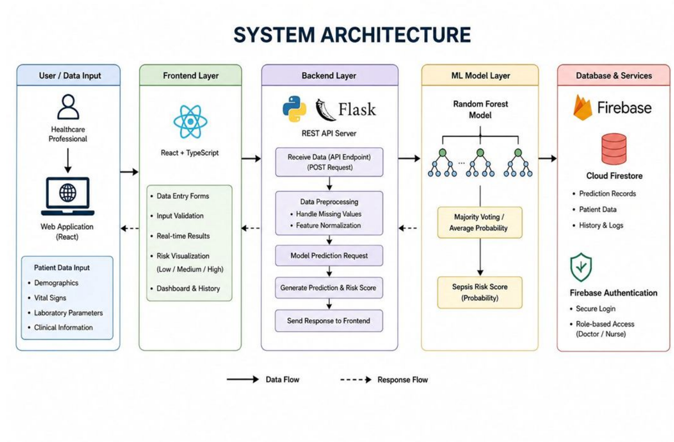


Fig. 1: System Architecture of SEWS

B. *Block Diagram*

The system workflow begins with data collection from synthetic and real ICU datasets. Data passes through preprocessing (cleaning, imputation, feature selection), then a Random Forest classifier is trained and validated using precision, recall, F1-score, and AUC. The trained model is integrated into the Flask backend, which bridges the ML model and the React frontend. Firebase services handle authentication and prediction logging.

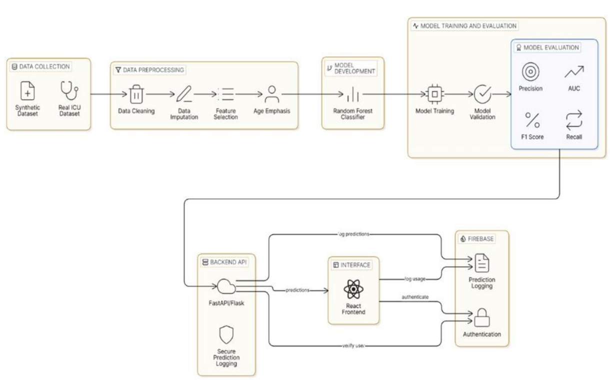


Fig. 2: Block Diagram of SEWS

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

C. Flow Chart

The flowchart illustrates the complete working procedure from patient data collection to risk display. Patient data (age, vitals, laboratory parameters) is first collected, then preprocessed (missing values handled, features normalized and scaled). The processed data is fed to the Random Forest classifier, the prediction is stored in Firebase, the web application fetches the result, and finally the risk score is displayed on the dashboard.

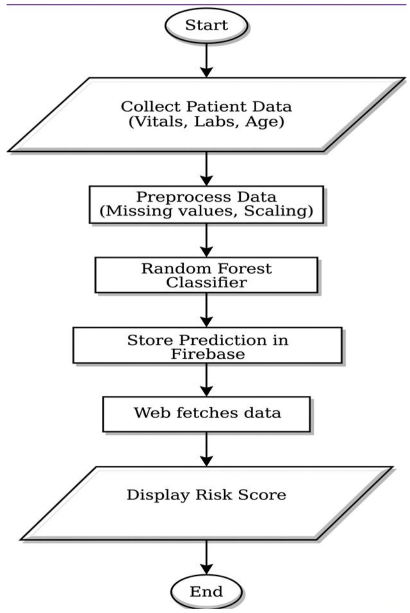


Fig. 3: Flow Chart of SEWS

D. Working Procedure and Observation Table

The web dashboard (React + TypeScript) allows users to enter patient clinical information. The preprocessing pipeline handles missing values, performs feature normalization and scaling, selects important clinical parameters, and structures data for model input. The trained Random Forest model deployed in the Flask backend returns predictions instantly to the dashboard, stores results in Firebase Firestore with timestamps, and logs prediction history for future analysis.

Parameter	Component Used	Observed Value / Action	Output / Display
Patient Data Entry	React Web Interface	Accepts patient age, vitals, and laboratory parameters	Patient information displayed on dashboard
Data Validation	Frontend & Backend Validation	Checks missing values and invalid inputs	Clean and structured data prepared
Data Preprocessing	Flask Backend	Normalization and feature scaling performed	Processed data ready for prediction
Sepsis Prediction	Random Forest Model	Analyzes clinical parameters and computes risk score	Predicted sepsis risk generated
Data Transmission	REST API	Sends and receives data securely between frontend and backend	Real-time prediction response
Data Storage	Firebase Firestore	Stores prediction logs with timestamp and user details	Continuous prediction history maintained
User Authentication	Firebase Authentication	Verifies authorized users before access	Secure login and access control
Risk Monitoring	Web Dashboard	Displays prediction using risk indicators	Low / Medium / High risk visualization
Alert Generation	Backend Logic	Detects high-risk sepsis cases	Alert displayed for immediate attention

Table I: Observation Table

VI. HARDWARE AND SOFTWARE REQUIREMENTS

A. *Hardware Requirements*

A personal computer with powerful GPU processing capability is used for dataset handling, machine learning model implementation, and testing. It supports data pre-processing, model execution, and result visualization. Adequate RAM and storage ensure smooth handling of ICU clinical datasets and organized management of project resources.

B. *Software Requirements*

Python serves as the primary programming language for the machine learning components. Scikit-learn implements the Random Forest classifier, NumPy handles numerical computations, and Pandas supports data cleaning and manipulation. The frontend uses React, TypeScript, and JavaScript for an interactive responsive web application. Flask provides the backend framework for REST API endpoints, and Firebase offers secure authentication and real-time cloud storage through Firestore.

C. *Cost Analysis*

Component	Cost
Domain Name	₹500
Firebase Database	₹499
Total	₹999

Table II: Cost Analysis

VII. DESIGN AND IMPLEMENTATION

A. *Real-Time Patient Data Monitoring*

The web application captures patient demographic details, vital signs, and laboratory parameters through a responsive dashboard. Input validation ensures data accuracy before transmission to the backend. Feature normalization provides standardized inputs to the model. The dashboard provides real-time visualization using color-coded indicators: Low Risk (stable), Medium Risk (requires monitoring), and High Risk (immediate attention required).

B. *Sepsis Prediction and ML Analysis*

The Random Forest model analyzes patient clinical data using an ensemble learning approach. The final prediction is given by:

$$\hat{y} = mode\{h_1(x), h_2(x), \ldots, h_t(x)\}$$

where $\hat{y}$ is the final predicted class, $h_t(x)$ is each decision tree's prediction, and T is the total number of trees. Gini impurity ($Gini = 1 - \Sigma p_i^2$) determines optimal feature splits during training, improving classification performance and reducing overfitting.

C. *Secure Data Storage and System Reliability*

Firebase Authentication restricts access to authorized healthcare professionals only. Prediction results are stored in Firebase Firestore with timestamps, enabling continuous prediction logging, historical patient monitoring, and easy retrieval. During testing, stable communication between all system layers was maintained with no significant data loss or communication failure observed.

D. *Web Interface and User Experience*

The React and TypeScript dashboard includes structured patient data entry forms, real-time prediction display, color-coded risk visualization, and prediction history tracking. Input validation mechanisms reduce incorrect data entry. The responsive design ensures smooth interaction across devices and screen sizes.

VIII. RESULTS AND DISCUSSION

A. *Output Screenshots*

The following screenshots illustrate the operational system. Fig. 4 shows the secure login page of the SEWS portal accessed at sepsisprediction.vercel.app. Fig. 5 shows the patient data entry dashboard displaying a Medium Risk result of 46%. Fig. 6 shows the prediction history table with timestamped entries and color-coded risk levels.

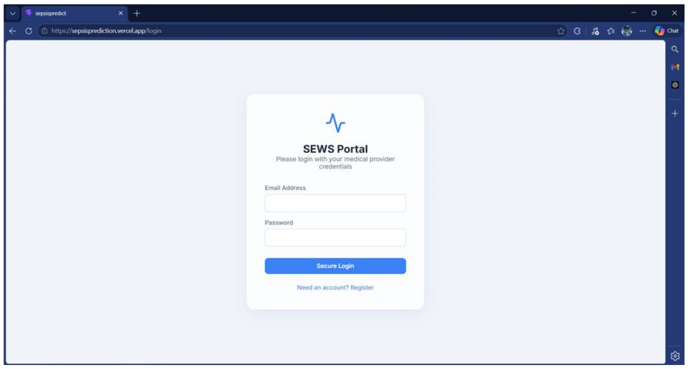


Fig. 4: Login Page of SEWS Portal

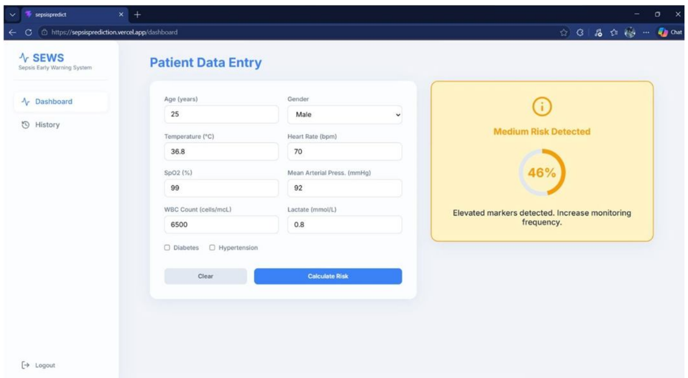


Fig. 5: Dashboard of SEWS Portal

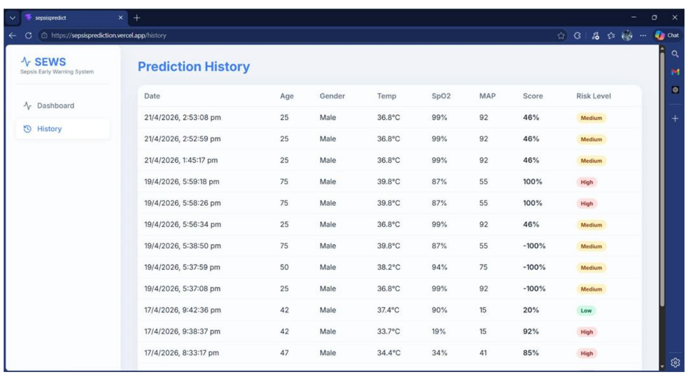


Fig. 6: Prediction History from SEWS

B. Real-Time Prediction and Monitoring

The developed web application successfully performs real-time prediction, processing patient information immediately after submission. The system collects and validates patient information, processes clinical parameters through the backend, generates sepsis risk predictions, and displays results using Low, Medium, and High risk indicators. Firebase Firestore enables continuous storage of timestamped prediction history for longitudinal patient monitoring.

C. Sepsis Prediction Performance

The Random Forest model demonstrated reliable prediction capability analyzing features such as age, heart rate, temperature, SpO2, and laboratory values. The ensemble-based approach improved prediction stability and reduced overfitting. Multiple decision trees improved prediction reliability, generalization capability, and resistance to noisy or incomplete data, confirming the effectiveness of machine learning for early sepsis detection.

D. System Performance and Efficiency

Key observations include fast response time for prediction generation, stable API communication between components, secure Firebase authentication, reliable cloud-based storage of prediction logs, and efficient handling of multiple concurrent requests. The color-coded dashboard interface reduced the complexity of result interpretation and minimized time required for understanding patient conditions.

E. Discussion

Results demonstrate the practical applicability of machine learning in healthcare prediction. The proposed system addresses the limitations of traditional manual sepsis diagnosis by providing automated real-time prediction support. The Random Forest model handled variations in input data while maintaining stable performance. Performance can be further improved using larger clinical datasets and advanced deep learning techniques, and integration with hospital EHR systems could enhance automation and accuracy.

IX. CONCLUSION

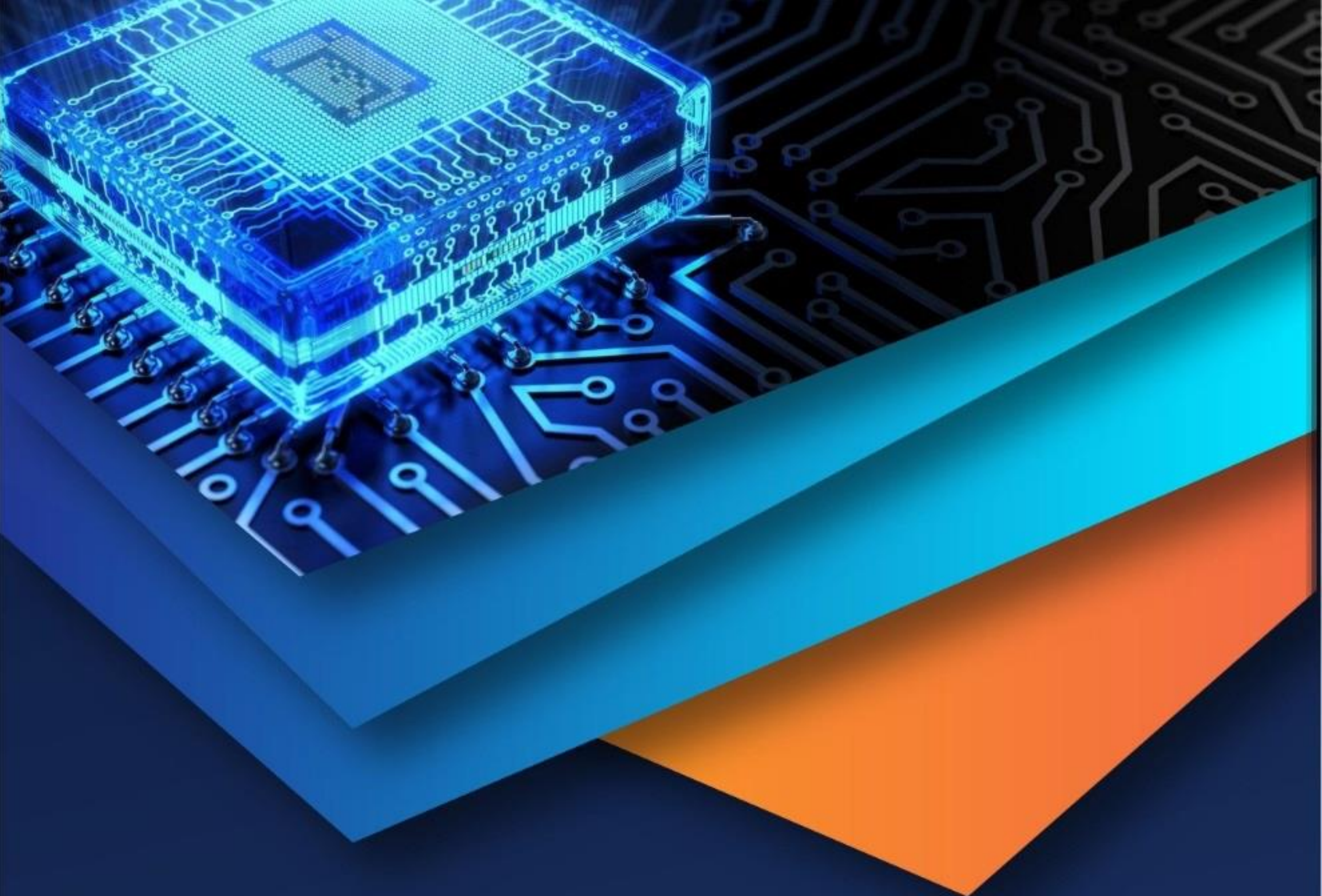
The Sepsis Early Warning System effectively applies machine learning and web technologies for early sepsis detection. By integrating a Random Forest classifier with a React frontend and Flask backend, the system analyzes patient clinical data and delivers real-time risk predictions. Firebase ensures secure data handling and historical tracking. The color-coded visualization (Low, Medium, High) enhances usability and allows healthcare professionals to interpret results quickly. The system is reliable, efficient, and scalable, making it suitable for practical healthcare and clinical decision support applications.

X. FUTURE WORK

1) Integration with real-time hospital Electronic Health Record (EHR) systems.
2) Use of advanced deep learning models for improved prediction accuracy.
3) Development of a mobile application for increased accessibility.
4) Incorporation of additional clinical parameters and larger datasets.
5) Deployment in real clinical environments for validation and testing.
6) Implementation of alert notification systems for critical cases.
7) Continuous model retraining using real-world data for better performance.

REFERENCES

[1] A. Boussina, H. Khurana, A. D. Bedoya et al., "Impact of a deep learning sepsis prediction model on clinical outcomes: a multi-site before-after implementation study," npj Digital Medicine, Vol. 7, 52, 2024.
[2] M. Moor, B. Rieck, M. Horn et al., "Predicting sepsis using deep learning across international, multi-centre data," Nature Medicine, Vol. 29, pp. 1991-2002, 2023.
[3] A. Rafiei, A. Ebrahimi et al., "SSP: Early prediction of sepsis using a fully connected LSTM-CNN model," Computers in Biology and Medicine, Vol. 137, 104795, 2021.
[4] S. P. Shashikumar, C. S. Josef, A. Sharma, and S. Nemati, "DeepAISE - An interpretable and recurrent neural survival model for early prediction of sepsis," Artificial Intelligence in Medicine, Vol. 113, 102036, 2021.
[5] Z. Zhang, C. Yan, and H. Qiu, "An interpretable deep-learning model for early prediction of sepsis," Patterns (Cell Press), Vol. 1, Issue 8, 100142, 2020.
[6] S. Bomrah et al., "A scoping review of machine learning for sepsis prediction," Critical Care, Vol. 28, Issue 1, pp. 68, 2024.
[7] Y. Tang et al., "A time-series driven model for early sepsis prediction based on CNN-Transformer and LSTM-Transformer," BMC Medical Research Methodology, Vol. 24, Issue 1, pp. 15, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY