IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131068

VitalSense AI: An AI and IoT-Based Health Monitoring System with Nutrition and Activity Support

Harish P Gungi[1], Abhishek S[2], Shambulinga[3], Karan Shankar[4], Dr. Ranjith Kumar[5]

Dept of CSE, Don Bosco Institute of Technology Bengaluru, India[1-4]

Associate Professor, Dept. of CSE, Don Bosco Institute of Technology Bangalore, India[5]

Abstract: VitalSense AI is a modular web-based prototype that combines vital-sign logging, nutrition and activity support, and health-related data visualization. The application includes heart rate, body temperature, and blood oxygen saturation (SpO2) monitoring through simulated readings and an API prepared to accept IoT data. Its existing modules include a Logistic Regression demonstration model trained on synthetic data, food and activity logs, recommendations, history, and PDF reporting. The expanded prototype adds a daily wellness planner, food-substitution suggestions, a digital health summary, sleep logging, locally stored reminders, statistical flags for unusual readings, browser-dependent voice input, CSV wearable-data import, a simple goal-progress estimate, text-file report extraction, and ag-gregate research analytics. These added functions are lightweight utilities and are not all independently validated; browser support, compatible input files, and runtime testing affect their operation. The application uses Flask, SQLite, NumPy, scikit-learn, HTML, CSS, JavaScript, and ReportLab. It is an academic prototype, not a clinically validated diagnostic system.

Index Terms: Internet of Things, health monitoring, machine learning, Logistic Regression, SpO2, nutrition tracking, activity monitoring, Flask, SQLite.

I. INTRODUCTION

Regular observation of basic health parameters can help people become more aware of changes in their physical con-dition. Wearable devices and Internet of Things (IoT) systems have made it possible to collect physiological measurements continuously and present them through digital interfaces. Re-cent research has explored IoT-based health monitoring using wearable sensors, web or mobile dashboards, and machine-learning techniques for identifying unusual patterns [1]–[3].

Although individual health applications are widely avail-able, many student-level systems focus on only one part of the problem, such as sensor monitoring or calorie tracking. A user may therefore need separate applications for vital signs, food records, activity information, recommendations, and reports. VitalSense AI was developed with the goal of bringing these functions into one lightweight platform.

The system collects or simulates three primary physiological values: heart rate, temperature, and SpO_2. A prototype Logistic Regression classifier produces a normal/risk indication and probability output; its training data are synthetic, so the output is not a clinically meaningful risk estimate. SQLite stores local records. Beyond vital-sign monitoring, the application provides food and activity logging, recommendations, PDF reports, wellness planning, sleep logs, reminders, data-import utilities, and summary analytics.

The present implementation deliberately separates simula-tion from real-device operation. During development, simu-lated readings are used so that the complete software workflow can be tested without requiring physical sensors. The real-device mode provides an API interface through which an ESP32 or another IoT controller can send sensor readings in JSON form in a later hardware stage.

II. RELATED WORK

IoT has become an important part of remote and continuous healthcare monitoring. Research on smart non-invasive health monitoring has combined wearable sensors, IoT connectivity, centralized data handling, and machine learning to support anomaly detection and user-facing health information [1]. Other work has combined IoT and

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131068

cloud platforms with machine learning for monitoring physiological parameters and supporting remote access to health information [2].

Wearable health monitoring systems have also demonstrated the practical value of collecting parameters such as heart rate, temperature, and oxygen saturation. A 2024 IEEE con-ference study described an IoT-based wearable system using sensors for oxygen level, heart rate, and body temperature and included a simulator to reproduce real-world scenarios [3]. Multi-physiological monitoring systems have similarly combined sensors, microcontrollers, wireless communication, and software visualization [4].

Nutrition tracking is another active area of research. Image-based food recognition systems have used deep learning and object detection to identify food and estimate calories [5], [6]. These approaches demonstrate the potential of computer vision for dietary tracking, although accurate portion estimation and food diversity remain challenging.

VitalSense AI differs from these focused systems mainly in its integration strategy. It provides a single modular dashboard and lightweight utilities rather than claiming a new specialized clinical model. Its distinguishing contribution is the combi-nation of vital-sign logging, nutrition and activity support, sleep and reminder records, basic trend/anomaly summaries, and report-oriented functions in one academic prototype. The food classifier and health-risk model remain demonstration components.

III. PROBLEM STATEMENT

Existing health-monitoring applications often separate phys-iological monitoring, nutrition tracking, activity logging, and report generation. For a student or general user, this separation can make it difficult to obtain a simple overall view of daily health-related information. There is a need for an integrated and easy-to-use prototype that can collect health readings, perform basic machine-learning analysis, store historical data, and provide supporting nutrition, activity, recommendation, and reporting features.

The project therefore addresses the following problem: how can IoT-oriented physiological data and a lightweight machine-learning model be combined with nutrition, activity, and re-porting functions in a single web-based platform while keeping the architecture simple enough for academic development and future hardware integration?

IV. PROPOSED SYSTEM

VitalSense AI is designed as a modular web application. The overall workflow is shown conceptually in Fig. 1.

A. Health Data Acquisition

The prototype accepts heart rate, temperature, and SpO_2 values. In simulation mode, readings are generated in normal, warning, and risk ranges with controlled probabilities. The interface can update readings automatically and also allows the user to generate a new reading manually.

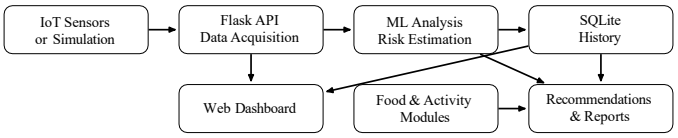


Fig. 1. High-level architecture of VitalSense AI.

The software also contains a real-device mode. In this mode, an external IoT controller can send JSON data to the Flask application. The intended data format contains the three physiological values:

$$D = \{HR, T, SpO_2\}. \quad (1)$$

Here, HR represents heart rate, T represents temperature, and SpO_2 represents blood oxygen saturation.

B. Machine-Learning Risk Analysis

A Logistic Regression classifier is used as the prototype risk-analysis model. Before classification, the three input fea-tures are standardized using a StandardScaler. The model is trained with generated prototype data containing two classes: normal and risk.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131068

The classifier estimates the probability of the risk class:

$$P(y=1|x) = \frac{1}{1+e^{-(\beta 0+\beta T x)}}. \quad (2)$$

The system displays the predicted category and the estimated probability on the dashboard. The probability is intended to communicate the model's output, not to represent a medically validated risk score.

C. *Food and Nutrition Module*

The food module allows the user to upload a food image and records the resulting food information in the database. The current academic prototype uses a lightweight demonstration classifier based on common food names present in the up-loaded filename, with a fallback category for mixed food. The stored nutrition fields include calories, protein, carbohydrates, and fat.

This implementation is intentionally described as a pro-totype rather than as a trained computer-vision model. A future version can replace the demonstration classifier with a trained CNN or an object-detection model. Existing IEEE research shows that deep-learning approaches can support food recognition and calorie estimation from images [5], [6].

D. *Activity Monitoring*

The activity module records the activity name, duration, number of steps, and estimated calories. The information is stored in SQLite and can be used together with health readings to generate simple recommendation messages.

E. *Recommendations and Reporting*

The recommendation module evaluates available measure-ments and produces rule-based messages. The expanded fea-ture page also provides a daily planner based on user-selected step and hydration goals and diet preference, food-substitution suggestions, a digital health summary, and a wellness score based on locally available logs. Sleep duration, self-reported quality and energy, and user-created reminders can be stored.

TABLE I
FUNCTIONAL TESTING OF THE PROTOTYPE

Module	Implementation status / verification note
Simulation and ML	Present in baseline code; model uses synthetic data and needs validation.
Dashboard and SQLite	Existing baseline features for display and local records.
Food and activity	Logging flows are present; food classification is filename-based, not visual AI.
Recommendations and PDF	Rule-based messages and PDF report generation are present in baseline.
IoT API	Endpoint is present; physical sensor connectivity has not been verified.
Planner, food swaps, wellness summary	Implemented as lightweight rule-based utilities; runtime testing required.
Sleep and reminders	Logging routes are present; reminders do not trigger scheduled notifications.
Anomaly flags and goal estimate	Simple statistical/averaging utilities; not clinically validated predictions.
Voice assistant	Depends on browser speech-recognition support and microphone permission.
Wearable import and report reader	CSV/TXT upload paths are present; test with compatible files.
Research analytics	Aggregates locally stored data; data may be simulated or user-entered.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131068

V. IMPLEMENTATION

The prototype is implemented as a Flask web application. Flask handles page rendering and REST-style API endpoints. SQLite provides a lightweight local database for health, food, and activity records. NumPy is used to generate prototype training data, while scikit-learn provides StandardScaler and LogisticRegression. ReportLab is used to generate PDF re-ports.

The health database contains timestamp, heart rate, temper-ature, SpO_2, risk result, probability, and source fields. Separate tables store food and activity records. This separation keeps the system modular and makes it possible to extend the database later.

The main dashboard provides Dashboard, Food & Nutrition, Activity, Recommendations, History, and Reports areas. An additional innovation page groups the expanded utilities: food suggestions and a digital health summary; trends and daily planning; latest IoT API readings and contact/alert utilities; wellness scoring; a rule-based assistant; sleep logs; saved reminders; anomaly flags; browser-dependent voice input; wearable CSV import; goal-progress estimates; TXT report extraction; and aggregate analytics. These are software features in the codebase, but their inclusion does not establish that every feature has been exhaustively tested with real user data or external devices. The simulation/real-device interface and IoT API provide a software integration path; physical sensors have not been verified as connected in the current project context.

The real-device integration follows a simple data path:
Sensor → IoT Controller → Flask API→ ML → SQLite → Dashboard. (3)

An ESP32-based implementation can be added in the future using suitable heart-rate, temperature, and SpO_2 sensors.

VI. RESULTS AND DISCUSSION

The baseline application implements simulation, local stor-age, dashboard display, food/activity logging, recommenda-tions, and PDF reporting. The expanded Python file also contains routes and interface controls for the features listed in Table I. In this review, the expanded code was inspected for its feature routes and descriptions; a complete end-to-end runtime test of every new feature, browser, and imported file type has not been independently established. The table therefore distinguishes existing baseline behavior from features present in code whose runtime behavior still requires local verification. The architecture supports an end-to-end software workflow in which a simulated reading can be passed to the prototype model, stored in SQLite, displayed, and reviewed later. The expanded code broadens the intended workflow with additional local logging and analysis utilities. Because the expanded modules have not all been independently runtime-tested in this review, they should be presented as implemented in code and awaiting verification rather than as experimentally validated results. No quantitative accuracy, clinical performance, or real-sensor performance claim is made.

The current results should not be interpreted as clinical val-idation. The Logistic Regression model is trained on synthetic prototype data rather than a clinically collected dataset. Simi-larly, the current food-recognition component is a demonstra-tion implementation rather than a trained image-recognition model. These limitations are important when interpreting the displayed predictions and nutrition estimates.

VII. LIMITATIONS AND FUTURE WORK

The present version is an academic prototype and has several limitations. Sensor readings may be simulated, and physical sensor connectivity has not been verified. The Logis-tic Regression model is trained on generated data and cannot support claims of medical diagnostic accuracy. Food classifica-tion relies on the uploaded filename rather than visual feature extraction. The assistant and planner use rules rather than a full language model or validated personalization method. Anomaly flags use simple statistical thresholds; wellness scores and goal estimates are illustrative. Reminders are stored locally but are not scheduled notifications. Voice input depends on browser capabilities and permissions, wearable import depends on a compatible CSV schema, and report extraction currently supports TXT files rather than PDF medical reports. The new routes and interface elements require end-to-end testing on the target laptop and browser.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131068

Future work should first verify every new route with pos-itive, negative, and boundary test cases, then connect compatible ESP32-based sensors and validate the data pipeline. The risk model would require representative, ethically sourced, validated data and careful evaluation before any health-related interpretation. Food image recognition could be upgraded with a trained vision model and portion-estimation method. Further engineering work includes scheduled reminders, more robust wearable integrations, broader report formats, authenticated user profiles, secure storage, privacy controls, and optional cloud synchronization. The anomaly and recommendation methods should be evaluated for reliability and should remain clearly separated from medical diagnosis.

VIII. CONCLUSION

VitalSense AI brings together vital-sign logging, a synthetic-data Logistic Regression demonstration, nutrition and activity support, PDF reporting, and a broader set of wellness utilities in a modular Flask application. The expanded code includes sleep logs, locally stored reminders, rule-based planning and assistance, simple anomaly flags, voice-input integration when supported by the browser, CSV wearable import, goal-progress estimates, TXT report extraction, and aggregate analytics. These additions increase the range of workflows that can be demonstrated, but they should be described according to their actual implementation and verification status.

The system remains an academic prototype. The health model is not clinically validated, food classification is not true image recognition, reminders do not trigger scheduled notifications, report extraction is limited to text files, and physical sensor connectivity and full end-to-end operation of the new features require further testing. With documented test results, compatible sensors, stronger data and models, and appropriate privacy safeguards, the platform could be developed further as a personal health-awareness tool.

REFERENCES

[1] P. Jyothi, A. Shankar, A. N. J. Ashwath, P. Gaur, and S. S. Kumar, "Smart Non-Invasive Real-Time Health Monitoring using Machine Learning and IoT," in *2023 IEEE International Symposium on Smart Electronic Systems (iSES)*, Ahmedabad, India, 2023, pp. 437–440, doi: 10.1109/iSES58672.2023.00098.

[2] P. and C. Rana, "IoT and Cloud Based Health Monitoring System Using Machine Learning," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, Uttar Pradesh, India, 2022, pp. 1727–1732, doi: 10.1109/IC3I56241.2022.10072946.

[3] S. Chowdhury, M. M. Rahman, A. R. Dristy, R. S. Yeara, and M.R. Uddin, "An IoT-Based Wearable Healthcare Monitoring Device and Medical Emergency Response System," in *2024 IEEE Students Conference on Engineering and Systems (SCES)*, Prayagraj, India, 2024, doi: 10.1109/SCES61914.2024.10652375.

[4] X. Tang, W. Xu, H. Ye, J. Cai, S. Zhou, and L. Cai, "Health Mon-itoring System Design of IoT-based Multi-Physiological Parameters with STM32," in *2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2024, pp. 409–412, doi: 10.1109/ICCECE61317.2024.10504186.

[5] C. V. Reddy, C. B. S. Teja, B. P. Kumar, C. Srinathreddy, and V.V. S. Tallapragada, "Real-Time Image-based Calorie Estimation using YOLOv4 Object Detection," in *2024 International Conference on Ex-pert Clouds and Applications (ICOECA)*, Bengaluru, India, 2024, doi: 10.1109/ICOECA62351.2024.00137.

[6] V. Anbunathan, K. Ramanthan, and S. Santhosh, "Food Estimation Based on Calories on Images Using Deep Learning Approaches," in *2026 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2026, doi: 10.1109/ICSES66558.2026.11479040.

[7] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License