International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Powered Palliative Care Management System: Integrating Large Language Models for Clinical Translation and Automated Condition Assessment

Chris Paulose*, Josekutty Raju#, Midhun M P &, Nijin Saji@, Ojus Thomas Lee+
Department of Computer Science and Engineering, College of Engineering Kidangoor, Kottayam, Kerala, India
*chrispaulosecp7@gmail.com, #josekuttyraju3@gmail.com, &midhunmp493@gmail.com, @nijinsaji04@gmail.com, +ojustl@ce-kgr.org

***Abstract*—Delivering effective palliative care in distributed home settings presents significant challenges, particularly regarding accurate clinical documentation and patient monitoring by field workers (e.g., Asha workers) who may lack formal medical training or proficiency in professional medical English. We present the design and implementation of an AI-powered Palliative Care Assistance System built on the Django web framework. The platform seamlessly integrates Large Language Models (a local MedGemma 4B instance and Google Gemini Vision) to automate clinical translation, converting layman terminology and regional languages (Malayalam) into standardized medical English. Furthermore, the system employs vision-capable LLMs to generate structured summaries of medical images (X-Rays, Lab Reports, Prescriptions) and evaluates patient visit histories to dynamically assess condition severity (Stable, Moderate, Severe). Supported by a robust backend managing bilingual visit logs, material allocations, and role-based portals, the system significantly reduces documentation overhead and enhances clinical decision-making for doctors and headnurses.**

***Index Terms*—palliative care, large language models, generative AI, medical translation, clinical assessment, Django, health informatics, bilingual systems**

I. Introduction

The shift towards home-based palliative care has improved patient comfort but has introduced a gap in clinical data fidelity. Field workers are often tasked with recording complex symptoms, assessing vitals, and capturing medical images (prescriptions, wound photography, and lab reports) under suboptimal conditions [1]. Without formal medical vocabulary, their notes are frequently colloquial or recorded in local languages, leading to communication bottlenecks when reviewed by assigned doctors.

Prior systems have largely focused on standard operational aspects, such as scheduling and basic electronic health records (EHR) [2]. While predictive models have been utilized for mortality or specific disease trajectories [3], the application of general-purpose Large Language Models (LLMs) to sanitize, translate, and structure unstructured field data in real-time remains underexplored in palliative contexts.

Our proposed system directly addresses these challenges by embedding AI capabilities into the core data ingestion pipeline. Utilizing a local MedGemma 4B model for all clinical text processing ensures patient data privacy, while a dynamic fallback to Google Gemini maintains high availability during localized hardware bottlenecks.

Contributions:

1) **AI-Driven Medical Translation:** A real-time processing layer that accepts bilingual (Malayalam and English) symptoms and notes, using LLMs to translate and convert informal layman terms into formal medical terminology without diagnosing.
2) **Vision-Enabled Image Summarization:** Custom AI prompt pipelines tailored to specific image classifications (e.g., extracting dosages from prescriptions, identifying abnormality severity from lab reports).
3) **Automated Condition Assessment:** An analytical feature that parses the latest patient vital trends and prior visit logs to render a JSON-structured triage recommendation and reasoning for the physician.
4) **Robust Backend Architecture:** A scalable Django deployment managing a unified relational database for patient telemetry, geographic coordinates, visit images, and hardware-agnostic material inventories.

II. Literature Survey

This section reviews the evolution of digital palliative care solutions, highlighting recent advancements in telehealth, electronic medical records, and AI-driven platforms. Understanding these current implementations provides a necessary foundation for identifying the operational gaps our integrated architecture aims to resolve.

To facilitate communication directly between patients, their families, and multiple healthcare providers involved in specialized palliative care, Sigaard et al. [4] proposed TelePal.dk, a secure and interactive web-based platform. It offers features such as video consultations, shared notes, scheduling tools, and self-assessment modules, enabling a collaborative care environment. Its mixed-method evaluation approach highlights its effectiveness in enhancing coordination among stakeholders while reducing healthcare costs.

In the Tell Us™ system, Dy et al. [5] developed a web-based platform that systematically collects electronic patient-reported outcomes (PROs) to support proactive symptom management. It generates automated alerts for clinicians and provides dashboards for monitoring patient health trends, thereby mitigating the risk of crises.

Remote patient monitoring was uniquely addressed by Castillo Padrós et al. [6] through the HumanITcare platform, which integrates mobile applications, connected devices, and clinician dashboards. It enables real-time data collection and alert-based interventions, preventing hospital readmissions.

Targeting low-resource settings, an open-source EMR system with offline functionality and simplified data entry, known as DataPall, was designed by Shah et al. [7]. It significantly improves efficiency in managing patient records and ensures continuity of care.

A comprehensive framework integrating IoT, 5G, cloud computing, and AI technologies for delivering healthcare services in remote areas was introduced by Humayun et al. [8]. It actively bridges the gap between urban and rural healthcare access through real-time predictive analytics.

Focusing on cancer patients, the usability and effectiveness of the MyPal platform was evaluated by Koumakis et al. [9], showcasing how wearable data and symptom reporting enhance patient engagement and quality of life.

Through the PREPARE study, Alied et al. [10] examined Electronic Palliative Care Coordination Systems (EPaCCS) for documenting patient care preferences, demonstrating their potential in reducing hospitalizations.

For the early detection of deterioration in elderly patients, Gokalp et al. [11] deployed an integrated system combining physiological and behavioral monitoring sensors to support independent living.

Leveraging mobile solutions in developing nations, Ngoma et al. [12] connected specialists with local health workers using the mPalliative Care Link (mPCL) application, fostering scalable healthcare delivery methods.

Patient adherence during complex treatments was notably bolstered by Geerts et al. [13] via the MM E-coach application, which offers active modules for medication management and communication.

Interdisciplinary care and family involvement were identified as critical strategies to improve end-of-life care experiences in a systemic review conducted by Quigley et al. [14].

Tools like PCCS and symptom heatmaps were used by Engel et al. [15] to integrate palliative methodologies strictly into established hospital EMR systems, enhancing robust data continuity.

Analyzing telepalliative care ecosystems, Cormi et al. [16] found that telemedicine in nursing homes is highly effective for clinical triage, albeit occasionally less suited for psychosocial interaction.

Video consultations natively between patients and specialists were streamlined by Balasubramanian et al. [17] utilizing the e-Palliative Homecare (e-PHC) system, which greatly increased accessibility metrics.

Lastly, for autonomous self-care, Chan et al. [18] introduced the SUPPORT+ application, empowering patients to self-report symptoms and receive direct guidance, elevating engagement in advance care planning.

III. System Architecture

The system architecture is structured into highly cohesive, interconnected modules designed to address distinct logistical and clinical requirements in palliative care workflows. Fig. 1 illustrates the holistic module interactions.

A. *Interconnected Dashboards*

The operational interfaces are differentiated by roles to ensure optimal data security and workflow efficiency:

- **Head Nurse Dashboard:** Acts as the administrative nerve center. It allows head nurses to upload and sync field data, register new system assets, review patient statuses historically, and orchestrate scheduling.
- **Doctor Dashboard:** Equips the assigned clinical professional with direct access to patient reviews. The module incorporates an advanced interface enabling direct video conferencing with patients or field volunteers and provides mechanisms to suggest newly adjusted medications rapidly.
- **Patient Dashboard:** Empowers the patient and their guardians. Features include patient profile registration, the ability to review past medical suggestions, and initiating direct requests for material allocation or home visits.

B. *Functional Modules*

Driving the logic between the dashboards are functional processing layers:

- **Patient Visit Module:** Composed of dual subsystems. The *Visit & Symptom Module* handles the management of scheduled visits and integrates geolocation tracking to assist field worker routing. Concurrently, the *AI Module* manages autonomous condition assessments based on synced vitals.
- **Material Allocation Module:** Maintains a strict logistical inventory, covering physical equipment and medications. It dynamically handles stocking and restocking workflows, ensures accurate material count visibility across clinics, and stores a comprehensive allocation history for auditing transparency.
- **Analytical & Report Module:** Aggregates system-wide data to provide meaningful insights. It leverages AI to generate automated summaries and symptom counts, applies advanced filtering for custom demographic reporting, and graphically renders detailed health charts for long-term patient trend analysis.

IV. Implementation

The section gives an insight into the implementation details of AI enabled features of the system. The core novelty of the implementation resides in the `ai_utils.py` subsystem, which leverages prompt engineering and structured JSON outputs across multiple clinical axes.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

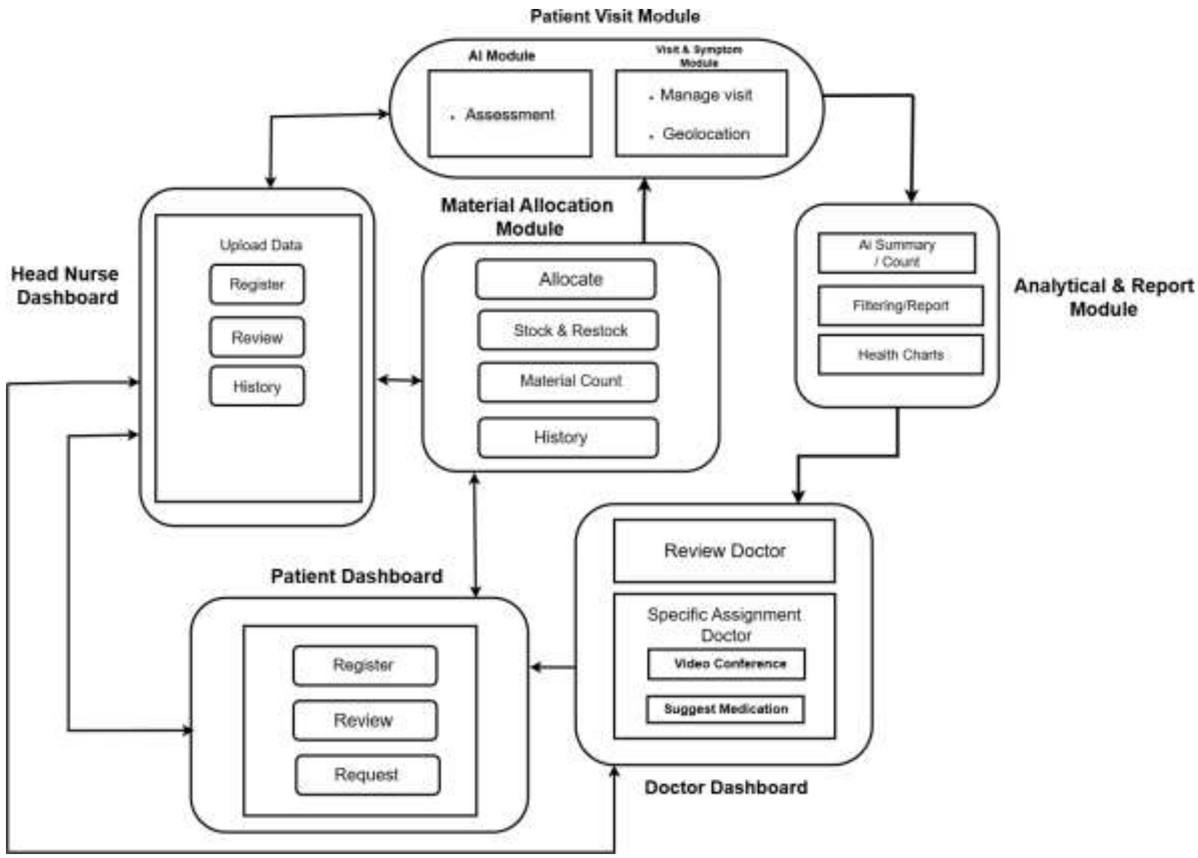

Fig. 1. System architecture illustrating the integrated dashboards and functional modules.

A. Clinical Documentation Translation

When a volunteer submits symptoms recorded in colloquial terms (e.g., "patient complains of stomach pain and throwing up"), the `correct_medical_text_with_ai` function intervenes. Subject to a strict system prompt, the LLM converts the text to formal semantics. Crucially, the prompt instructs the model: *"Do NOT make diagnoses unless clearly implied. Do NOT add new symptoms."* This ensures the generated English remains a structurally sound translation rather than an autonomous medical opinion, mitigating hallucination risks.

B. Automated Condition Assessment

Evaluating patient triage priority traditionally relies on manual chart review. The `analyze_patient_condition` function automates this by constructing a dense context window containing:

- Fixed Patient Profile (Age, Underlying Disease).
- A rolling window of the last three recorded visits.
- The current visit's raw input (Symptoms, Notes, Services Performed).

The LLM is prompted to respond strictly in JSON format containing a `recommendation` ("Stable", "Moderate", or "Severe") and a single-sentence medical `reason`. Upon the Visit object's database commit, an ORM signal updates the Patient's `current_status` state.

TABLE I
AI SUBSYSTEM RESPONSE HANDLERS

Feature Endpoints	LLM System Prompt Constraint
Text Correction	Convert layman/Malayalam to formal medical English; No diagnostics.
Condition Assess	Output strict JSON. Evaluate past 3 visits.
Prescription Vision	Extract Medicines, Instructions, Doctor.
Wound/X-Ray Vision	Identify Affected Part, Severity, Main Cause.
Holistic Analysis	Synthesize images and vitals into global Trend Summary.

C. Holistic Image Summarization

Field workers upload various media types via the `VisitImage` ORM model. The `summarize_image_with_ai` router utilizes Gemini's vision capabilities, dynamically altering the prompt based on the `image_type`:

- **Prescriptions:** Extracts a bulleted list of medicines and dosages.
- **Lab Reports:** Highlights abnormal values and assesses severity.

- **X-Ray / Wound:** Identifies the main cause, affected part, and provides a severity classification (Mild/Moderate/Severe).

This structured output bypasses lengthy paragraphs, delivering actionable bullet points directly to the Asha worker's mobile interface.

D. Trend and Medication Analysis

A unified function (`analyze_patient_images_and_medications`) synthesizes historical text logs and previously generated AI image summaries. The resulting JSON payload populates the patient's dashboard with a high-level "General Summary", a multi-visit "Trend Analysis", and a consolidated "Medications" list. All outputs are cached in the `Patient` record (`ai_general_summary`, etc.) to reduce redundant API calls.

Beyond AI integrations, the platform strictly governs operational logistics through explicit relational tables.

E. Visit and Telemetry Lifecycle

Every registered `Patient` possesses spatial attributes (`latitude`, `longitude`) and tracks the latest recorded vitals (`bp`, `sugar`, `pulse`). A `VisitRequest` can be initiated, subject to head nurse approval, which subsequently provisions a `Visit`. Save-hooks within the Django model automatically bubble up the latest vitals from a discrete visit to the core patient profile, ensuring the doctor's dashboard displays the most recent data in $O(1)$ database access time.

F. Material Allocations

Palliative care heavily relies on equipment (e.g., oxygen concentrators, hospital beds). The `MaterialAllocation` model links specific patients to tracked hardware sourced from the `Inventory` model. Inventories are categorized by source ('Government', 'Sponsorship') ensuring financial accountability and streamlined return logging.

V. Results and Discussion

The system's core capabilities map directly to targeted artificial intelligence processing pipelines designed to minimize field worker cognitive overload.

A. Implementation of Bilingual Clinical Translation

The `correct_medical_text_with_ai` module directly processes text logged by field volunteers—frequently captured in colloquial Malayalam or informal English. The implementation primarily routes this raw text to a locally hosted `MedGemma 4B` model, relying on `gemini-2.5-flash` strictly as a backup. It uses a severely restrictive system prompt: *"Convert informal layman English/Malayalam into formal medical English. Do NOT add new symptoms. Do NOT make diagnoses unless clearly implied."* This ensures the tool acts purely as an advanced clinical translator, preventing the model from hallucinating non-existent medical conditions into the patient's record (see Fig. 2).

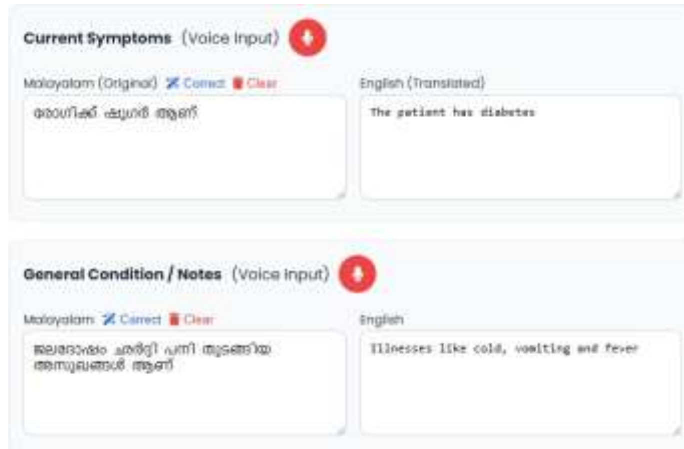

Fig. 2. Demonstration of the Bilingual Clinical Translation feature.

B. Automated Condition Assessment Pipeline

To compute a patient's triaging priority, the backend executes the `analyze_patient_condition` pipeline. Rather than processing isolated events, the Django implementation dynamically aggregates a structured context window containing the patient's fixed demographic profile, an array of their three most recent historical visits, and the current visit's raw symptom string. The LLM is forced to return strict JSON containing two keys: `recommendation` ("Stable", "Moderate", or "Severe") and `reason`. The application layer natively intercepts this response, parses it via Python's `json.loads()`, and utilizes an overridden ORM `Visit.save()` hook to instantly escalate the patient's global `current_status` state.

C. Vision-based Medical Image Summarization

Handling unstructured media is accomplished via the `summarize_image_with_ai` router. When a volunteer uploads a `VisitImage`, the backend detects the uploaded `image_type` and applies a specialized prompt matrix before dispatching the base64-encoded image to the Google Gemini Vision API. For example:

- **Prescriptions:** The LLM is tasked explicitly to extract 'Medicines' (Names and dosages), 'Instructions', and 'Doctor/Specialty'.
- **Lab Reports:** The model bypasses normal metrics and is prompted exclusively to flag 'Abnormal Values' and output a three-tier 'Severity' rating.
- **Wound/X-Ray:** Generates a targeted assessment including 'Main Cause', 'Affected Part', and 'Key Observation'.

This segmented prompt engineering guarantees that head nurses receive dense, immediately valuable bullet points rather than vague conversational paragraphs (see Fig. 3).

D. Holistic Trend and Medication Analytics

To provide a consolidated view, the `analyze_patient_images_and_medications` function synchronizes historical visit texts with previously generated visual summaries. The resulting synthesis

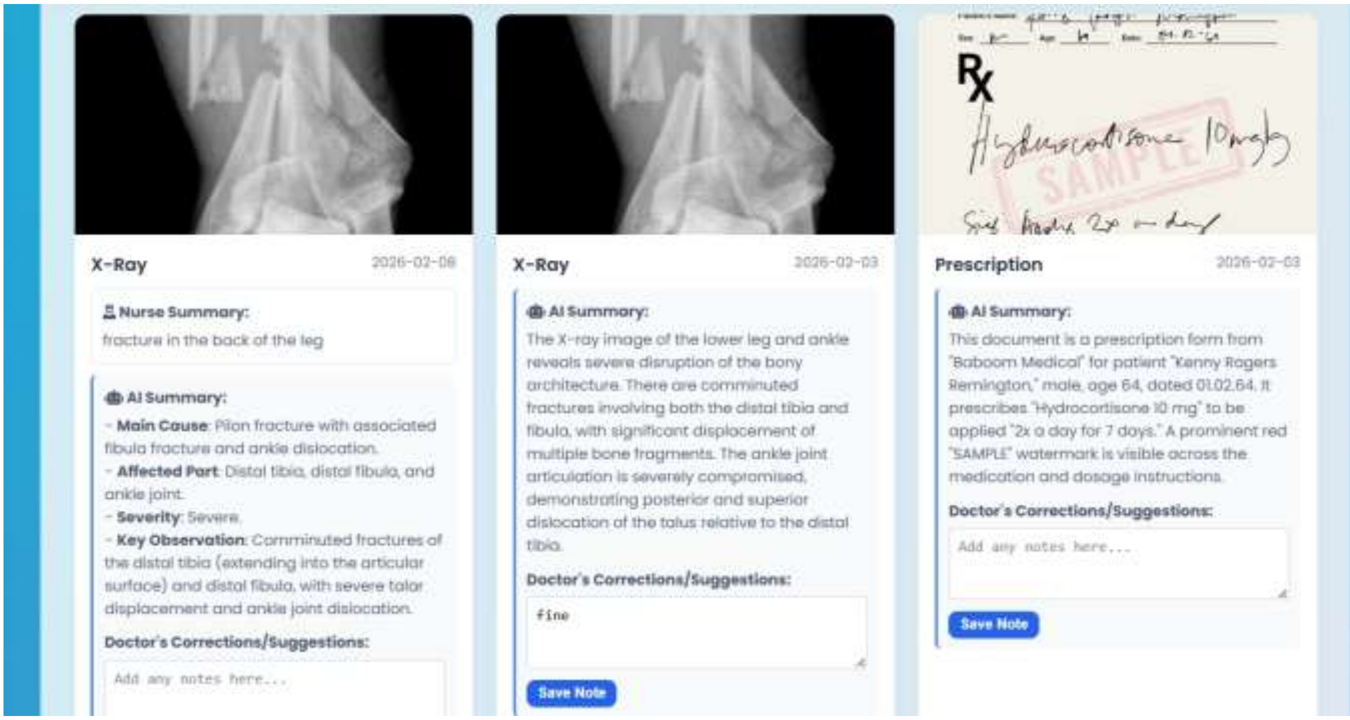

Fig. 3. Examples of the vision-based medical image summarization outputs.

automatically caches four critical attributes on the `Patient` master record: `ai_general_summary`, `ai_image_analysis`, `ai_trend_analysis`, and a parsed `ai_medications` list. Caching these attributes at generation time eliminates the need for redundant API calls, allowing the dashboard frontend to load complex analytical overviews in $O(1)$ database fetches (see Fig. 4).

E. Real-Time Teleconsultation Integration

To facilitate immediate medical intervention, the platform integrates a real-time teleconsultation module directly into the doctor's dashboard. Rather than relying on external, disjointed communication apps, the system securely generates targeted video conferencing sessions using the open-source Jitsi Meet API. When a remote specialist initiates a call, the frontend dynamically hashes the unique `patientId` to spin up a dedicated, encrypted `PalliativeCare_Meeting` room. The system automatically injects the doctor's authenticated credentials and synchronizes the session alongside the patient's EHR overview. This seamless integration prevents fragmentation of patient data, allowing clinical staff to synchronously monitor AI-generated summaries and vital trends while communicating face-to-face with field volunteers or patients.

F. Geospatial Route Planning and Scheduling

Efficient visit scheduling is vital for decentralized healthcare delivery. To minimize travel times, the system incorporates an interactive routing dashboard utilizing Leaflet mappings. When a batch of patients is selected, geographic distances are evaluated via an embedded nearest-neighbor Haversine optimization algorithm, sorting the patient pool dynamically from an anchoring starting coordinate. Rather than relying on simple Euclidean lines, the resulting sequence queries the Open Source Routing Machine (OSRM) API to retrieve high-fidelity geographic polylines, plotting authentic road traversals. Once the route is visually confirmed by the head nurse, a consolidated JSON payload is dispatched to a batch endpoint, sequentially instantiating targeted `Visit` records across the Django backend (see Fig. 5).

VI. Conclusion

The Palliative Care Assistance System successfully illustrates how Large Language Models can be deeply embedded into clinical workflows without overriding physician authority. By restricting the AI's jurisdiction to translation, summarization, and severity flagging, the application eliminates the cognitive load required of field volunteers recording complex medical states in their native languages. The underlying Django architecture reliably manages visit lifecycles, coordinates, and material inventories, establishing a cohesive ecosystem. Future work will focus on further refining the Gemini vision capabilities and optimizing the quantization of our local MedGemma 4B deployments for faster edge inference.

Acknowledgements

The authors thank the development community and healthcare professionals who provided domain-specific insights towards refining the platform.

Fig. 4. Dashboard example cases showing holistic trend and medication analytics.

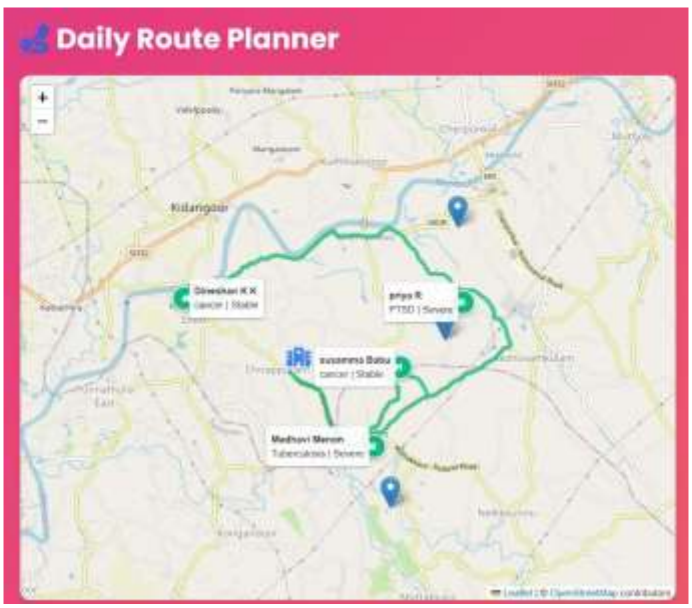

Fig. 5. Visualization of the Geospatial Route Planning interface.

REFERENCES

[1] S. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Smart systems for remote health monitoring: Challenges and status," *Sustainable Health Informatics*, vol. 15, no. 6, pp. 2736–2742, 2021.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] J. Nagi *et al.*, "Support vector machines for clinical anomaly tracking," *IEEE Transactions on Medical Information Technology*, vol. 25, no. 2, pp. 1162–1171, 2018.

[4] J. V. A. Sigaard *et al.*, "Telepalliation platform," *JMIR Research Protocols*, vol. 13, 2024.

[5] S. M. Dy *et al.*, "Tell us™ system," *Journal of Pain and Symptom Management*, vol. 42, no. 4, pp. 526–534, 2011.

[6] M. R. Castillo Padro´s *et al.*, "A smart system for remote monitoring of patients in palliative care (humanitcare platform)," *JMIR Formative Research*, vol. 7, 2023.

[7] K. G. Shah *et al.*, "Datapall emr," *BMC Palliative Care*, vol. 12, no. 1, p. 31, 2013.

[8] M. Humayun *et al.*, "Ict healthcare framework," *Applied Sciences*, vol. 13, no. 11, 2023.

[9] L. Koumakis *et al.*, "Mypal platform," *Frontiers in Digital Health*, vol. 3, 2021.

[10] M. Alied, S. Law-Clucas, M. Allsop, C. Ramsenthaler, P. May, A. Bearne, M. Powell, J. Rosling, R. Kumar, L. Scerri, R. Williams, K. Sleeman, D. Laverty, D. James, J. Verne, K. Saravanakumar, C. Costelloe, J. Droney, and Koffman, "Evaluation of electronic palliative care coordination systems to support advance care planning for people living with life-threatening conditions (prepare): protocol for a multicentre observational study using routinely collected primary and secondary care data in england," *BMJ Open*, vol. 15, no. 3, 2025.

[11] H. Gokalp *et al.*, "Integrated telehealth," *Telemedicine and e-Health*, vol. 24, no. 12, pp. 940–957, 2018.

[12] M. Ngoma *et al.*, "mpalliative care link," *JCO Global Oncology*, vol. 7, pp. 1306–1315, 2021.

[13] P. A. F. Geerts *et al.*, "Mm e-coach," *HemaSphere*, vol. 6, 2022.

[14] D. D. Quigley *et al.*, "End-of-life care review," *American Journal of Hospice and Palliative Medicine*, vol. 38, no. 1, pp. 84–93, 2021.

[15] P. T. Engel *et al.*, "Pccs system," *International Journal of Environmental Research and Public Health*, vol. 17, p. 4977, 2020.

[16] C. Cormi *et al.*, "Telepalliative care strategy in nursing homes," *BMC Palliative Care*, vol. 20, p. 156, 2021.

[17] S. Balasubramanian, M. S. Biji, M. K. Ranjith, and S. S. Abhina, "Patient satisfaction in home care services through e-palliative care: an experience of a tertiary cancer centre from kerala," *Indian Journal of Palliative Care*, vol. 28, pp. 250–255, 2022.

[18] W. L. Chan *et al.*, "Use of mobile app 'support+'," *Palliative Care and Social Practice*, vol. 19, pp. 1–7, 2025.