International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: http://ijwos.com
Vol.02 No.11.
OPEN ACCESS
E-ISSN : 3049-2424
DOI:

Poverty Tracker System for Real-Time Poverty Analysis

Nayan M M[1], Rajashekar G C[2]

[1] *Student, Department of Master of Computer Applications, GM University, Davangere, Karnataka.*
[2] *Assistant Professor, Department of Master of Computer Applications, GM University, Davangere, Karnataka.*

Article Info

Article History:

Published:11 Nov 2025

Publication Issue:
Volume 2, Issue 11
November-2025

Page Number:
241-244

Corresponding Author:
Nayan M M

Abstract:

This paper presents Poverty Tracker, an intelligent and interactive web-based framework designed to monitor, predict, and visualize poverty-related data in real time. Traditional poverty-tracking methods rely on manual surveys and static reports, resulting in delayed and error-prone outcomes. The proposed system leverages React, Vite, and AI-based image analysis to provide dynamic poverty visualization, predictive insights, and personalized recommendations. Modules such as Dashboard, Calculator, Compare, Stories, Schemes, and Scan collectively enhance analytical depth. By integrating Recharts for data visualization and React-Leaflet for geographical mapping, the system enables users and policymakers to analyze poverty trends efficiently. The Poverty Tracker promotes data transparency, community engagement, and intelligent poverty assessment, bridging the gap between socioeconomic data and actionable insights.

Keywords: Poverty Analysis, Web Application, React, Data Visualization, AI Image Analysis, Socioeconomic Prediction.

1. INTRODUCTION

The global fight against poverty demands intelligent digital tools capable of transforming raw socioeconomic data into actionable insight. Traditional tracking systems depend heavily on manual data collection, static census reports, and infrequent surveys that fail to capture real-time economic fluctuations.

The Poverty Tracker system, developed by GM University, Davangere, addresses these limitations through a web-based, modular, and data-driven architecture. It provides a platform where individuals, NGOs, and policymakers can analyze poverty levels, predict regional trends, and access government schemes relevant to poverty alleviation.

As shown in Fig. 1, the system architecture is divided into six interactive modules: Dashboard, Calculator, Compare, Stories, Schemes, and Scan. Each module functions independently but communicates through a unified data model using client-side state management.

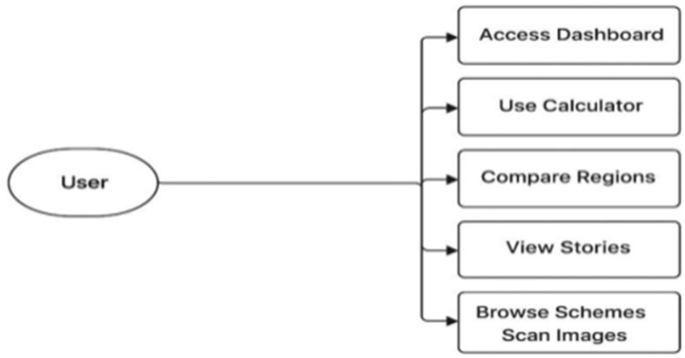

Fig 1: Use case diagram

2. SYSTEM DESIGN AND ARCHITECTURE

The architecture of Poverty Tracker integrates data visualization, mapping, and AI-based image analysis. Fig. 2 depicts the Data Flow Diagram, outlining the movement of data between the modules. The user inputs income, expenses, and regional details; the system processes them to calculate poverty levels and visualize trends.

The Entity–Relationship (E-R) Diagram in Fig. 3 represents the logical database structure that connects entities such as Locations, Schemes, and Stories [1].

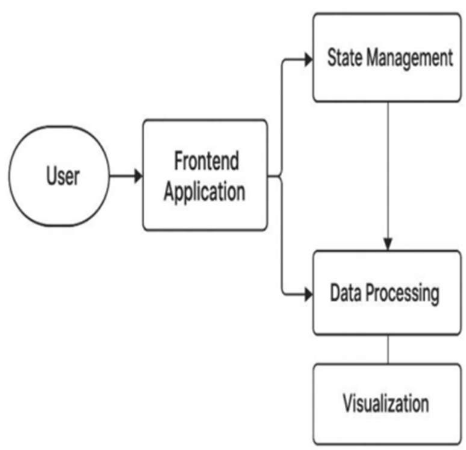

Fig 2 : Data Flow Diagam

Each module serves a distinct function:

- Dashboard visualizes poverty data in charts.
- Calculator estimates individual poverty levels.
- Compare analyzes region-wise disparities.
- Schemes lists welfare programs and eligibility details.
- Scan evaluates uploaded images for poverty indicators using AI models.

3. IMPLEMENTATION AND TESTING

The Poverty Tracker is implemented as a Single Page Application (SPA) using React 19, with Vite 7 as the build tool. The system utilizes Axios for API calls, Recharts for visualization, and React-Leaflet for geographic mapping. It maintains optimal responsiveness through CSS Grid and Flexbox design.

A. System Configuration

The hardware and software specifications are presented in Table I.

Table I - Software and Hardware Configuration

Component	Specification
Processor	Intel Core i3 or higher
RAM	Minimum 4GB
OS	Windows 10/11 or Linux
Framework	React 19, Vite 7
Visualization	Recharts
Mapping	React-I-eaflet
Browser	Chome,firefox,Edge
Testing Tools	Jest, Cypress,Lighthouse

B. Testing and Validation

The system underwent unit, integration, UI, and performance testing [2]. All components were validated for correctness, responsiveness, and speed. Fig. 4 shows the application's Dashboard, which provides visual analytics and real-time poverty indicators.

Fig 4 : Dashboard Interface showing visualized poverty metrics

The test outcomes are summarized in Table II.

Table II - Testing Summary of Poverty Tracker

Testing Type	Result
Unit Testing	100 % Pass Rate
Integration	Smooth Navigation
UI Testing	Fully Responsive
\Performance	Avg Load 2.3s

The performance results demonstrate that the system efficiently handles large datasets and maintains high responsiveness across multiple devices.

4. RESULTS AND DISCUSSION

The implemented system successfully bridges traditional and digital poverty monitoring approaches. It provides real-time analytical dashboards with visual poverty indices, while the Calculator dynamically computes user-specific poverty levels based on financial inputs [3].

The AI-based image analysis module represents a unique innovation, enabling the identification of living conditions from uploaded images, adding objectivity to poverty evaluation. Testing results indicate the framework's scalability and robustness, validating its suitability for NGO, academic, and governmental adoption.The interactive UI and responsive design enhance accessibility, while modular architecture supports scalability for future backend integration.

5. CONCLUSION

The Poverty Tracker system demonstrates how modern web technologies can revolutionize poverty management. By integrating AI, data visualization, and interactive dashboards, it enables evidence-based decision-making for poverty reduction. The system enhances transparency, accuracy, and community engagement, supporting the vision of sustainable social development.

6. FUTURE WORK

Future developments include backend integration with real-time databases, implementation of machine learning algorithms for poverty trend prediction, mobile application support, multilingual accessibility, and API connectivity with government datasets [4][5].

References

[1] A. Banks and E. Porcello, *Learning React: Modern Patterns for Developing React Apps*, O'Reilly Media, 2020.
[2] S. Stefanov, *React Up and Running*, O'Reilly Media, 2016.
[3] E. Brown, *Web Development with Node and Express*, O'Reilly Media, 2019.
[4] I. Sommerville, *Software Engineering*, 10th ed., Pearson Education, 2015.
[5] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed., McGraw-Hill, 2015.
[6] World Bank Poverty Data, https://www.worldbank.org
[7] Government of India Poverty Statistics, https://mohua.gov.in
[8] UNDP Human Development Reports, https://hdr.undp.org