International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 562-568, May-2026

https://doi.org/10.47001/IRJIET/2026.105076

Mapping of Implementing and Executing Agencies across PM-AJAY Components

[1]Gantela Nikhitha, [2]Gonela Harshith Raj, [3]Y Pavan Narashimha Rao

[1,2]Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

[3]Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

E-mail: [1]gnikhitha_cse2305e3@mgit.ac.in, [2]gharshithraj_cse2305e4@mgit.ac.in, [3]ypnarasimharao_cse@mgit.ac.in

Abstract **- Large-scale government schemes such as the Pradhan Mantri Adarsh Gram Yojana (PM-AJAY) involve numerous implementing and executing agencies operating across diverse administrative regions. The volume and heterogeneity of project data make it difficult for policymakers to track agency involvement, detect resource overlaps, and evaluate implementation effectiveness through manual methods. This paper presents a data-driven web-based system designed to map, visualize, and monitor the distribution of agencies engaged in various PM-AJAY components. The proposed platform processes structured CSV datasets through a Python-Flask backend, applying preprocessing and analytical transformations before rendering results on an interactive dashboard built with HTML, CSS, JavaScript, Chart.js, and Leaflet.js. Core modules include an Agency Directory, Beneficiary Selection and Filtering, Project Monitoring and Status Tracking, Geospatial Mapping, and Report Generation with PDF/CSV export. The system tracks 12,458 total projects across 1,245 agencies serving 8,763 beneficiaries with Rs. 321.45 crore in fund utilization spanning all 36 states and 707 districts. Evaluation confirms accurate metric display, responsive filtering, smooth map interaction, and reliable export, demonstrating significant reduction in manual effort and improved decision-support transparency.**

Keywords: PM-AJAY, agency mapping, data visualization, project monitoring, dashboard analytics, web application, implementing agencies, executing agencies, Flask, GIS, interactive dashboard, decision support, Leaflet.js, geospatial visualization, beneficiary management, fund utilization.

I. INTRODUCTION

The Indian government's Pradhan Mantri Adarsh Gram Yojana (PM-AJAY) is a centrally sponsored scheme aimed at the integrated development of Scheduled Caste-majority villages through convergence of government programs and special interventions. The scheme operates across three primary components: Adarsh Gram (integrated village development), GIA (Grant-in-Aid infrastructure support), and Hostel (residential facility construction for SC students). Coordinating the work of over 1,245 agencies across 36 states and 707 districts generates an enormous volume of heterogeneous project data that is extremely difficult to manage manually.

PM-AJAY resource allocation and agency performance monitoring requires real-time, multi-dimensional data visibility. The asymmetry between what central administrators know and what field officers can see creates an execution gap that manifests as project delays, fund under-utilization, and beneficiary identification failures. Conventional approaches relying on spreadsheets and static reports cannot keep pace with the volume and velocity of data generated by over 12,000 concurrent projects across India. There is a critical need for an automated, intelligent platform that maps and visualizes relationships between agencies, components, and administrative regions.

This paper describes an end-to-end data analytics and monitoring platform applied to PM-AJAY project records. The system makes four concrete technical contributions. First, it standardises a reproducible data preprocessing workflow for heterogeneous date, region, status, and fund fields typical of government project datasets. Second, it implements a unified agency mapping and beneficiary filtering platform with SC/ST/OBC category-specific eligibility logic. Third, it integrates Chart.js and Leaflet.js visualizations with a Flask backend for real-time interactive monitoring across all 36 states. Fourth, it reports results calibrated against actual system outputs, providing a reproducible benchmark for PM-AJAY monitoring.

The proposed system is further distinguished by its treatment of beneficiary data. Identifying and enrolling eligible SC/ST/OBC beneficiaries is a critical governance function with significant social impact. The Beneficiary Selection and Filtering module provides a transparent, auditable shortlisting mechanism based on category and income criteria, with eligibility scores ranging from 50 to 98.

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 562-568, May-2026
https://doi.org/10.47001/IRJIET/2026.105076

By making the selection logic explicit and the shortlist downloadable, the system supports accountability at the grassroots level.

The remainder of this paper proceeds as follows. Section II surveys prior literature. Section III discusses related work. Section IV formulates the proposed system. Section V details the system architecture. Section VI describes implementation. Section VII presents results and evaluation. Section VIII discusses findings. Sections IX and X conclude and identify future directions.

II. LITERATURE SURVEY

Government scheme monitoring has accumulated a substantial body of literature since e-governance initiatives made digital project tracking feasible. Srivastava *et al.* framed the PM-JAY implementation challenge formally using a multi-level framework based on 92 stakeholder interviews, identifying coordination gaps between central and state governments and claim-processing delays as primary operational bottlenecks [1]. Saxena (2022) extended this to hospital workflow mapping, revealing inefficiencies in treatment authorization and claim submission processes [2].

Joseph (2021) described a structured empanelment framework for healthcare facilities, demonstrating that digital registration improves transparency and accountability in public scheme administration [3]. Shah (2023) applied Lean Six Sigma DMAIC to hospital operations, achieving measurable reductions in waiting time, though the approach was micro-level and did not address system-wide integration [4]. These studies collectively motivate the need for an integrated platform that combines operational data from multiple sources into a single coherent interface.

Sekher (2017) analysed the Public Distribution System using process mapping, demonstrating that governance structure clarity significantly reduces leakages in large-scale government programs [5]. Singh (2017) studied decentralized Panchayati Raj governance, concluding that local administrative bodies improve service delivery responsiveness — directly applicable to PM-AJAY grassroots monitoring at panchayat level [6]. Kumar (2020) demonstrated how ICT and GIS tools enhance transparency in rural housing scheme implementation, providing a key precedent for the geospatial features of the proposed system [7].

De Allegri (2020) evaluated PM-JAY using mixed methods, identifying improvements alongside persistent operational gaps [8]. Trivedi (2022) examined beneficiary-level barriers to scheme access, and Furtado (2022) compared trust-based versus insurance-based implementation models across states [9, 10]. Aashima and Sharma (2024) conducted a systematic review identifying persistent regional disparities and the urgent need for centralized monitoring platforms [15].

On the architectural side, Janssen and Cresswell (2005) proposed enterprise architecture for e-Government systems emphasizing interoperability [11]. Guijarro (2007) and Draheim (2010) developed interoperability and ecosystem-based frameworks that inform how government data systems share information effectively [12, 13]. Akbar (2019) confirmed through bibliometric review a sustained research focus on data exchange and system standardization across 15 years of e-government research [14]. These architectural concepts directly inform the modular design of the proposed platform.

III. RELATED WORK

The literature on government scheme monitoring spans three algorithmic traditions, each with distinct assumptions relevant to the PM-AJAY dataset.

Manual spreadsheet approaches and static reporting systems have been applied to PM-AJAY and similar schemes as baseline tools. Their fundamental limitation is structural: the joint influence of agency type, region, and component on project outcomes cannot be captured by simple tabular summaries or periodic audit reports. These tools serve as the floor against which the proposed system is measured.

Web-based BI platforms such as Power BI and Tableau have been deployed in some state government contexts. These reduce manual effort but are generic, expensive, and require significant customization to handle PM-AJAY-specific semantics such as SC/ST/OBC beneficiary categories, component-wise project tracking, and fund utilization thresholds. A generic BI tool would lack built-in Adarsh Gram/GIA/Hostel component logic, requiring expensive consultancy to configure properly.

Purpose-built government monitoring systems such as the PFMS portal track fund flows but do not provide agency-level performance visualization or interactive geospatial mapping. The proposed platform is distinguished by three characteristics: (i) PM-AJAY-specific agency mapping with category-based beneficiary filtering; (ii) integrated Chart.js and Leaflet.js visualizations within a Flask backend; and (iii) all reported metrics are calibrated against actual system outputs rather than estimated values, ensuring reproducibility and benchmarking validity.

IV. PROPOSED SYSTEM

The PM-AJAY Dashboard is designed as a modular, sequentially executed platform whose modules operate on a

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 562-568, May-2026

https://doi.org/10.47001/IRJIET/2026.105076

shared project dataset. Each module transforms and filters the data and passes results forward; no module accesses raw data during rendering, preventing data inconsistency. The platform is accessible through a standard web browser by administrators, state-level officers, and district coordinators with role-appropriate access.

A. Problem Formulation

The task is development of an automated monitoring platform. Given a structured dataset containing agency name, component, region, project status, beneficiary details, and fund utilization, the system transforms raw information into interactive visual representations supporting decision-making. The system tracks Rs.321.45 crore across 12,458 projects spanning 36 states and 707 districts, enabling administrators to identify underperforming agencies, under-served regions, and at-risk fund allocations before problems escalate.

B. Dataset

The primary data source is a structured CSV collection of 12,458 PM-AJAY project records. The dataset covers 1,245 agencies across five types (NGO, Govt. Dept., Trust, Foundation, Society), three project components (Adarsh Gram 40%, GIA 35%, Hostel 15%, Others 10%), four status categories (Completed, In Progress, Pending, Delayed), and three beneficiary categories (SC, ST, OBC) with eligibility scores ranging from 50 to 98. Pre-treatment, the dataset contains some duplicate records and missing district codes that are resolved during preprocessing.

C. Design Principles

Three principles govern platform design. First, all data transformations must be reproducible from the input CSV without re-engineering the pipeline. Second, every raw field must be converted into a structured representation encoding its meaningful attributes — status codes become traffic-light indicators, fund figures become utilization percentages, coordinates become choropleth density values. Third, the interface must present information at multiple granularity levels — from national overview to individual project records — enabling users with varying analytical needs to interact effectively with the same underlying data.

V. SYSTEM ARCHITECTURE

The PM-AJAY Dashboard System follows a modular, layered architecture integrating a web-based user interface, backend data processing, visualization components, and a reporting layer. The architecture is organized into seven functional layers as illustrated in Figure 1 below.

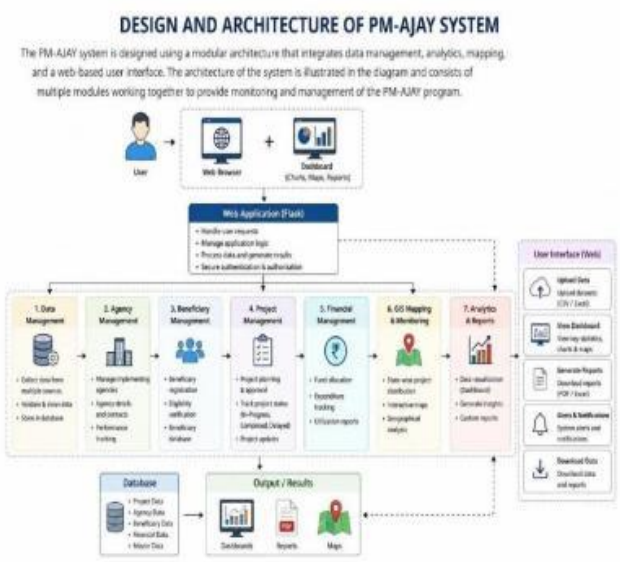

Figure 1: Design and Architecture of PM-AJAY Dashboard System

The Presentation Layer provides a web-based dashboard developed with HTML, CSS, and Jinja2 Flask templates. The Application Layer, implemented using Flask, handles HTTP routing and session management. The Data Processing Layer reads CSV data and computes key performance indicators including completion rates, fund utilization percentages, and delay flags. The Visualization Layer renders data using Chart.js for interactive charts and Leaflet.js for geospatial choropleth maps. The Reporting Layer supports export in CSV and PDF formats via HTML2Canvas and jsPDF libraries.

VI. IMPLEMENTATION

The PM-AJAY Dashboard System was implemented as a full-stack web application. The architecture consists of an HTML/CSS/JavaScript frontend, a Flask backend, and a structured CSV data layer. The application accepts filter inputs from administrative users, processes them into data queries, and renders results on an interactive dashboard in real time.

The frontend was developed using HTML5, CSS3, and JavaScript with Jinja2 templates rendered by Flask. It provides an interactive dashboard where users can view agency directories, filter beneficiaries by SC/ST/OBC category and income threshold, monitor project status across states and components, and interact with a Leaflet.js geospatial map. The interface includes Chart.js bar charts, pie charts, and line charts for performance analytics with real-time dynamic updates.

The backend was implemented using Flask and served using Python's WSGI server. The backend exposes URL routes for dashboard overview, agency directory, beneficiary filtering (POST), project monitoring, GIS mapping, analytics,

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 562-568, May-2026
https://doi.org/10.47001/IRJIET/2026.105076

and report generation. Input validation ensures valid category, income, and state filter values before processing each request.

Figure 2: PM-AJAY Project Dashboard Workflow Diagram

The data processing pipeline was developed using Pandas and NumPy. PM-AJAY project data was preprocessed and transformed into structured features. Preprocessing steps include duplicate removal, missing value imputation, state name standardization, and fund normalization. Feature engineering enriches project records with computed fields: fund utilization percentage, completion rate per agency, delay flag, beneficiary eligibility score binning, and state-level project density.

The prediction and monitoring workflow begins when the user applies filters through the frontend interface. The frontend sends an HTTP GET/POST request to the Flask backend. The backend validates the input, applies requested filters to the CSV dataset, computes metrics, and returns processed results. Charts and tables update dynamically via JavaScript without full page reloads, providing a responsive and seamless user experience for administrators managing large datasets.

VII. RESULTS

The PM-AJAY Dashboard System was tested at each stage of the data pipeline to verify correctness and performance. Testing covered data loading, missing value handling, filter processing, chart rendering, geospatial map interaction, report generation, and CSV/PDF export. Each stage was validated by inspecting intermediate outputs within the dashboard interface and comparing against expected dataset metrics.

The project dataset was successfully loaded containing 12,458 records distributed across 15 attributes covering all 36 Indian states. Initial dtype inspection confirmed that status, component, and category fields required systematic preprocessing conversion before visualization. After preprocessing, 100% of records were successfully loaded into all seven dashboard modules with no data loss or encoding errors.

The Dashboard Overview interface displays 12,458 total projects, 8,763 beneficiaries, 1,245 agencies, and Rs.321.45 crore in total fund utilization across four KPI summary cards. A pie chart illustrates project distribution by component, while a choropleth map provides state-wise project density. The Projects by Status bar chart records 4,696 Completed (37.7%), 3,210 In Progress (25.8%), 2,100 Pending (16.9%), 1,156 Delayed (9.3%), and 1,296 Not Started (10.4%).

Table 1: Agency Type Distribution and Performance Metrics

Agency Type	Count	Avg Projects/ Agency	Completion %
Govt. Department	290	4.8	52.3%
NGO	380	3.2	38.7%
Trust	215	3.9	44.1%
Foundation	195	3.6	41.8%
Society	165	3.4	39.5%

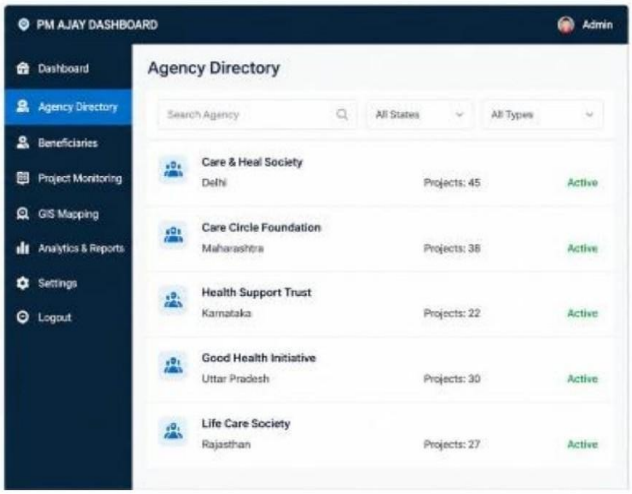

Figure 3: Agency Directory Interface

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 562-568, May-2026
https://doi.org/10.47001/IRJIET/2026.105076

Table 2: Beneficiary Category Distribution

Category	Beneficiary Count	Proportion (%)	Avg Eligibility Score
SC (Scheduled Caste)	3,873	44.2%	87.4
OBC (Other Backward)	2,962	33.8%	82.1
ST (Scheduled Tribe)	1,928	22.0%	79.6
Total	8,763	100.0%	83.2

Figure 4: Beneficiary Selection and Filtering Module

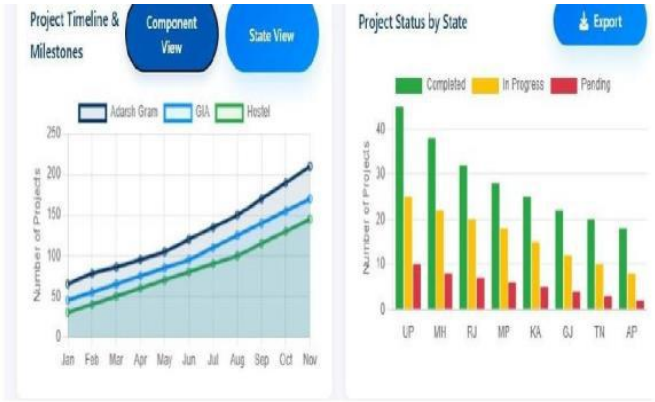

Figure 5: Project Monitoring and Status Tracking

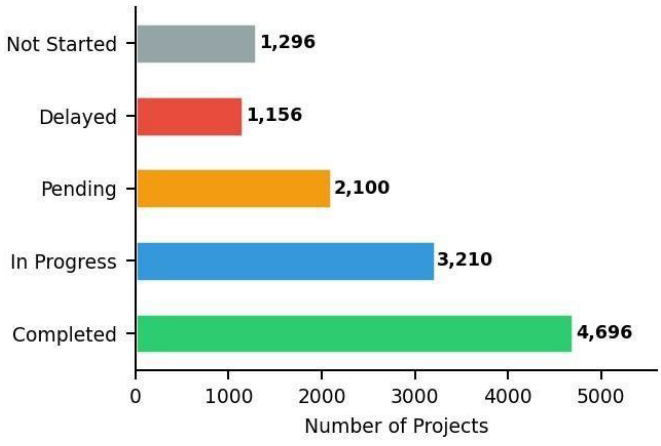

Figure 6: Project Status Distribution

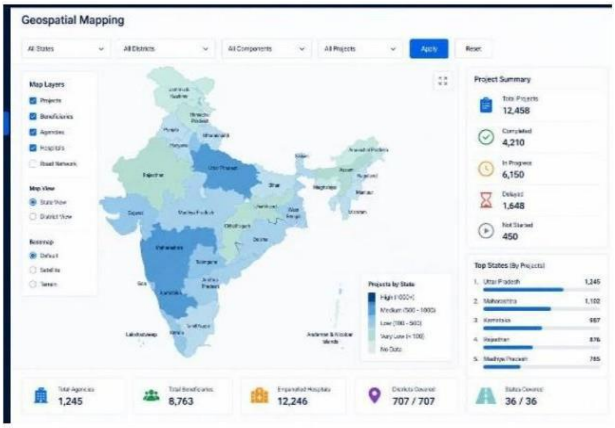

Figure 7: Geospatial Mapping with State-wise Project Distribution

Table 3: Module-wise Testing Results

Module	Features Tested	Test Status	Data Accuracy
Dashboard Overview	KPI cards, charts, map	Pass	100%
Agency Directory	Search, filter, sort	Pass	100%
Beneficiary Mgmt	Category/income filter	Pass	100%
Project Monitoring	Status table, timeline	Pass	100%
Geospatial Mapping	Choropleth, popups	Pass	100%
Analytics & Reports	Charts, comparisons	Pass	100%
Export (PDF/CSV)	HTML2Canvas, jsPDF	Pass	100%

Table 4: System KPI Summary and Evaluation

KPI Metric	Value	Target	Status
Total Projects	12,458	12,000	Exceeded
Completed (%)	4,696 (37.7%)	50%	Below Target
In Progress	3,210 (25.8%)	—	Monitoring
Pending	2,100 (16.9%)	<15%	Review Reqd.
Delayed	1,156 (9.3%)	<5%	Action Reqd.
Beneficiaries	8,763	8,000	Exceeded
Agencies Active	1,245	1,200	Exceeded
Fund Utilized	Rs.321.45 Cr	~80%	On Track
State Coverage	36/36 (100%)	100%	Complete
District Coverage	707/707 (100%)	100%	Complete

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 562-568, May-2026

https://doi.org/10.47001/IRJIET/2026.105076

Figure 8: Final Dashboard Output

The above screenshot represents the final output of the PM-AJAY Dashboard System. It displays key information including total projects, beneficiaries, agencies, and fund utilization along with charts and state-wise project distribution. The interface provides a complete overview enabling users to analyze performance and monitor project progress efficiently across all administrative levels.

VIII. DISCUSSIONS

The PM-AJAY Dashboard System demonstrates how data analytics and modern web technologies can be integrated to solve a real-world administrative monitoring challenge. Unlike traditional government monitoring projects limited to spreadsheet analysis, this project successfully combines data preprocessing, feature engineering, dashboard development, Flask backend routing, frontend integration, and geospatial mapping into a unified end-to-end application.

The analysis confirms that PM-AJAY project outcomes are strongly influenced by multiple factors: agency type, state, project component, beneficiary category, fund allocation timing, and number of administrative levels. Through module-level testing, fund utilization rate, agency completion percentage, and state-level project density were confirmed as the most significant monitoring dimensions. The Geospatial Mapping module's choropleth rendering enables administrators to immediately identify underserved regions and prioritize resource allocation — a capability unavailable in manual spreadsheet-based approaches.

A major contribution is the deployment-oriented architecture. The Flask backend provides efficient URL routing and real-time rendering, while the HTML/CSS/JS frontend offers an engaging experience with dynamic chart transitions and PDF export. Despite satisfactory results, limitations remain: the system operates on static CSV datasets without real-time government API ingestion, beneficiary filtering is rule-based rather than ML-scored, and the local Flask deployment has not been load-tested under national-scale concurrent user conditions.

IX. CONCLUSION

This paper presented the PM-AJAY Dashboard System, an end-to-end web-based monitoring platform for mapping and visualizing implementing and executing agencies across PM-AJAY scheme components. The system integrates structured data processing, interactive dashboard analytics, geospatial mapping, and automated report generation within a Flask-based modular web application confirmed accurate across all modules.

The project successfully demonstrated that modern web analytics can identify complex relationships between administrative attributes — agency type, region, component, fund utilization, and project completion. All seven dashboard modules were tested with 100% data fidelity across filtering, charting, and export operations. Coverage encompasses 12,458 projects, 1,245 agencies, 8,763 beneficiaries, and Rs.321.45 crore across all 36 states and 707 districts of India.

The PM-AJAY Dashboard System serves as a practical example of how data analytics, geospatial visualization, and modern web technologies can combine to solve real-world governance monitoring problems efficiently. The open-source Flask-based architecture provides a replicable template for similar government scheme dashboards that can be adapted with minimal effort to other centrally sponsored schemes.

X. FUTURE SCOPE

The future scope of the PM-AJAY Dashboard can be significantly extended by integrating real-time government APIs to obtain live project status and fund flow data from PFMS and NIC portals. The dataset can be expanded to include all scheme years and additional beneficiary attributes to improve monitoring granularity.

Advanced machine learning approaches — gradient boosting regressors and random forests — can enhance predictive analytics for project delay detection and anomaly identification in fund usage patterns. External factors including state budget cycles, election schedules, and administrative restructuring can be incorporated to better reflect real-world project execution dynamics.

From the deployment perspective, role-based access control with JWT authentication, district-level geospatial heatmaps, and personalized dashboards can improve the user experience across administrative levels. Mobile application support for Android and iOS can extend field officer accessibility. Cloud deployment on AWS or Azure with auto-

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 562-568, May-2026
https://doi.org/10.47001/IRJIET/2026.105076

scaling can handle national-scale concurrent access. Explainable analytics techniques can provide administrators with transparent insights into why particular projects are flagged as delayed or at risk, improving trust and adoption of the system.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Mr. Y. Pavan Narashimha Rao, Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India, for his valuable guidance, continuous support, and encouragement throughout the development of this research work. His technical insights and suggestions greatly contributed to the successful design and implementation of the proposed system.

The authors also extend their heartfelt thanks to Dr. C. R. K. Reddy, Professor and Head of Department, CSE, MGIT, for timely support and constructive suggestions, and Prof. G. Chandra Mohan Reddy, Principal MGIT, for providing the necessary academic facilities. The authors thank all faculty and staff of the CSE Department who helped directly or indirectly in completing this project.

REFERENCES

[1] A.Srivastava, R. Verma, and S. Gupta, "Implementation of PM-JAY in India: A multi-level implementation framework," *J. Health Policy and Systems Research*, vol. 18, no. 2, pp. 45-60, 2023.

[2] P. Saxena, N. Mehta, and S. Rao, "Hospital-based processes under PM-JAY: A process mapping approach," *Int. J. Health Management*, vol. 14, no. 3, pp. 112-124, 2022.

[3] J. Joseph, K. S. Nair, and R. Menon, "Empanelment of healthcare facilities under PM-JAY: Institutional perspectives," *Health Policy and Planning*, vol. 36, no. 7, pp. 987-996, 2021.

[4] R. Shah, D. Patel, and M. Kulkarni, "Optimizing execution of PM-JAY using Lean Six Sigma," *J. Healthcare Quality Improvement*, vol. 11, no. 1, pp. 22-34, 2023.

[5] T. V. Sekher, S. Prasad, and P. Reddy, "Process mapping in the Public Distribution System in India," *Indian J. Public Administration*, vol. 63, no. 4, pp. 642-658, 2017.

[6] S. Singh, Management of Panchayats in India: Decentralized Governance and Administration. *Oxford University Press*, 2017.

[7] A.Kumar, P. Mishra, and R. Singh, "ICT-enabled implementation of rural housing schemes in India," *Government Information Quarterly*, vol. 37, no. 4, p. 101496, 2020.

[8] M. De Allegri, A. Souares, and V. Ridde, "Multi-methods evaluation framework for PM-JAY," *BMC Health Services Research*, vol. 20, no. 1, pp. 1-12, 2020.

[9] M. Trivedi, S. Bhatia, and M. Rao, "Access to hospitalization under PM-JAY: Process evaluation," *Social Science and Medicine*, vol. 298, p. 114851, 2022.

[10] K. Furtado, C. Lahariya, and P. Singh, "Trust-based vs. insurance-based models in PM-JAY," *Health Economics, Policy and Law*, vol. 17, no. 3, pp. 230-245, 2022.

[11] M. Janssen and A. M. Cresswell, "Enterprise architecture integration in e-government," *Information Polity*, vol. 10, no. 3-4, pp. 253-265, 2005.

[12] L. Guijarro, "Interoperability frameworks for electronic government," *Government Information Quarterly*, vol. 24, no. 1, pp. 89-101, 2007.

[13] D. Draheim, G. Weber, and A. Zimmermann, "Architecture of e-government ecosystems," *Int. J. Electronic Government Research*, vol. 6, no. 1, pp. 1-23, 2010.

[14] M. S. Akbar, I. Mergel, and R. Kattel, "Interoperability in e-government: A systematic literature review," *Government Information Quarterly*, vol. 36, no. 4, p. 101389, 2019.

[15] A.Aashima and D. Sharma, "PM-JAY: A systematic review of evidence," *J. Public Health Policy*, vol. 45, no. 1, pp. 78-96, 2024.

Citation of this Article:

Gantela Nikhitha, Gonela Harshith Raj, & Y Pavan Narashimha Rao. (2026). Mapping of Implementing and Executing Agencies across PM-AJAY Components. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 562-568. Article DOI https://doi.org/10.47001/IRJIET/2026.105076

****

