IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

E-Grampanchayat: AI-Powered Smart Village Administration System

Kapse Tejas Arun[1], Jadhav Ashitosh Suresh[2], Todmal Sanket Ganesh[3], Gangawane Seema Balasaheb[4], P. P. Chavan[5], P. A. Sawant[6], Dr. H. B. Jadhav[7]
Student, Department of Computer Engineering[1234]
Faculty, Department of Computer Engineering[567]
Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, India

Abstract : *Gram Panchayats are the first point of contact between rural citizens and the government, yet many of their services, such as certificates, tax collection and complaint handling, still depend on paper registers, repeated visits to the panchayat office and the availability of a single village official. This paper presents the design of E-Grampanchayat, an AI-powered smart village administration system built with Python. The system provides a multilingual citizen portal for certificate applications, house and water tax payment, grievance registration and scheme information, together with an administrative dashboard for the Sarpanch and the Gram Sevak. Lightweight artificial intelligence components based on natural language processing (NLTK/spaCy) and machine learning (scikit-learn) classify citizen requests, assign grievances a category and priority, route them to the responsible official, and answer common questions through a chatbot in Marathi, Hindi and English. The system is organised as a linear five-stage pipeline that is easy to understand, extend and run on modest hardware. The paper describes the architecture, technology stack, request-processing flow, applications, advantages and limitations, a privacy and security analysis and a planned evaluation, and outlines future work including voice interaction, DigiLocker integration, IoT-based village services and predictive analytics.*

Keywords: E-Governance, Gram Panchayat, Smart Village, Artificial Intelligence, Natural Language Processing, Chatbot, Python, Flask

I. INTRODUCTION

Local self-government is the foundation of rural administration in India. About 69 percent of the population lived in rural areas at the 2011 Census [1], and the Gram Panchayat, the lowest tier of local government under the 73rd Constitutional Amendment [2], is the institution through which villagers obtain birth and death certificates, pay local taxes, raise complaints about water supply or street lighting, and learn about welfare schemes. Despite national initiatives such as Digital India [3] and growing international attention to digital government [4], many of these everyday services in small villages are still delivered through paper registers and in-person visits.

A citizen who needs a certificate may have to travel to the panchayat office more than once, wait for the Gram Sevak to be available and return again to collect the document. Complaints are often made verbally or on paper, leave no record and cannot be tracked. Elected members and officials, in turn, have little consolidated data on pending applications, tax dues or recurring problems in their wards, and information about government schemes often reaches families late or incompletely.

This paper presents E-Grampanchayat, an AI-powered smart village administration system implemented in Python. The system combines a web-based citizen portal, a rule-based workflow engine for panchayat services and lightweight natural language processing for request understanding, grievance classification and a multilingual help-desk chatbot. It is designed for the conditions of a small village: modest hardware, limited connectivity and users who are more

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

comfortable in Marathi than in English. The design is open and modular, so that students, developers and panchayat staff can study, extend and adapt it to their own needs.

The remainder of the paper is organised as follows. Section II reviews related work. Section III states the problem and objectives. Section IV describes the system architecture and methodology. Section V lists the technology stack, Section VI presents the supported features, and Section VII discusses applications. Sections VIII and IX discuss advantages, limitations and future scope, and Section X concludes the paper.

II. LITERATURE REVIEW

E-governance research has long argued that technology alone does not improve public service delivery unless it is matched to local processes and users. Case studies of Indian e-government projects describe both the gains in transparency and the difficulty of sustaining them at the local level [5], and Heeks describes the design–reality gap, the distance between what a system assumes and the conditions in which it is actually used [6]. A village administration system therefore has to be designed around the real constraints of a panchayat office.

At the national level, the eGramSwaraj portal of the Ministry of Panchayati Raj supports panchayats in planning, progress reporting and accounting [7], and state portals such as Maharashtra's Aaple Sarkar offer notified public services online [8] under the Maharashtra Right to Public Services Act, 2015 [9]. These platforms are valuable, but, as far as their public documentation shows, they are oriented towards departmental and statewide workflows. They are not presented as a village-level help desk that a citizen can talk to in the local language, and they do not by themselves provide the request classification, routing and ward-level analytics that a small panchayat office needs.

Chatbots have become a common interface for public information services, and Adamopoulou and Moussiades survey their history, technology and applications [10]. For the language-understanding layer, NLTK [11] and spaCy [12] provide tokenisation, tagging and entity recognition, while scikit-learn [13] offers compact text classifiers, such as TF-IDF features with logistic regression, that train in seconds on an ordinary laptop. Pre-trained transformers such as BERT [14] handle paraphrases better at a higher computational cost, and Whisper [15] extends speech recognition to many languages, including Hindi and Marathi. On the application side, Flask [16] is a small web framework suitable for prototypes, and Tesseract [17] is an open OCR engine that can read printed documents for verification.

Two observations from the literature shape this work. Speech technologies can perform unevenly across speaker groups [18], so a village system must offer text and assisted alternatives rather than voice alone. Language resources for Indian languages are also improving quickly through initiatives such as BHASHINI [19], which favours a modular design in which language components can be swapped. These observations motivate an open, modular and easily extensible village administration system such as the one proposed here.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Panchayat services still require citizens to be physically present during office hours, which costs wages and travel for farmers and daily-wage workers. Applications and complaints are not tracked, so citizens cannot tell whether their request is pending, approved or lost. Records are scattered across registers, which makes tax dues, certificate history and complaint patterns hard to see. Many residents prefer to work in Marathi, have limited digital literacy and are unaware of the schemes they qualify for. Finally, panchayat offices have very few staff, so manual sorting and follow-up of requests consumes time that could be spent on development work.

B. Objectives

- To build a citizen portal for certificate applications, tax payment and grievance registration, with a trackable ID for every request.
- To classify and route citizen grievances automatically using simple NLP and machine-learning techniques.
- To provide a multilingual (Marathi, Hindi and English) help-desk chatbot for services and scheme information.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

- To give the Sarpanch and Gram Sevak a dashboard of pending work, dues and grievance trends.
- To protect citizen data through role-based access, audit logs and minimal data collection.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall Architecture

The proposed system follows a linear pipeline in which each stage hands its output to the next, from a citizen's request to a notification and a stored record, as shown in Fig. 1. The five stages are the citizen interface, request understanding, AI services, workflow and records, and notification and output. This modular design allows any stage to be replaced, for example swapping the intent classifier, without affecting the others.

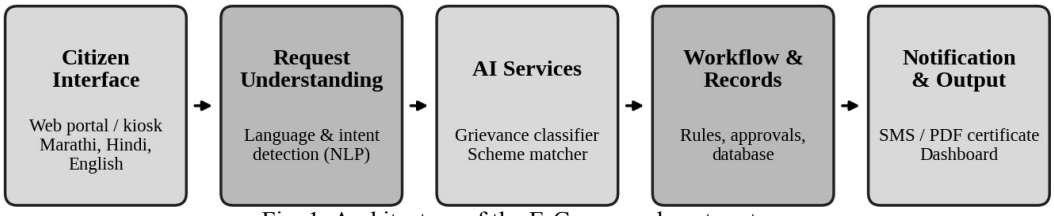


Fig. 1. Architecture of the E-Grampanchayat system

B. Process Flow

The operation of the system can be summarised in five steps.

- Submit: the citizen logs in with a mobile number and one-time password, then selects a service or types a request in the preferred language.
- Understand: free-text requests are normalised and matched against known intents; the language is detected from the script (Devanagari or Latin).
- Process: the matched handler runs. Grievances receive a category, priority and responsible official, and certificate applications are checked for required fields and documents.
- Approve: the Gram Sevak or Sarpanch reviews the queue and approves the application or returns it with a reason.
- Notify: the citizen receives status updates by SMS and on screen, and approved certificates are generated as PDF files with a QR code for verification.

C. Registration and Authentication

Citizens register with a mobile number verified by a one-time password and are linked to a household record so that family members and dues can be viewed together. The system defines five roles: citizen, ward member, Gram Sevak, Sarpanch and administrator. Passwords of staff accounts are stored only as salted hashes, and the system does not collect an identity number unless a service strictly requires it.

D. NLP Processing and Intent Matching

A free-text request is converted to lower case, normalised for Devanagari and Latin spelling variants, tokenised and cleaned of stop words using NLTK or spaCy together with hand-maintained Marathi and Hindi stop-word lists. Keywords and phrases are compared with a table of predefined intents such as certificate, tax, complaint, scheme, status and notice, and any extracted argument (for example the certificate type) is forwarded to the handler. Grievance text is additionally classified into categories such as water supply, street lighting, drainage and sanitation, roads, electricity and others, using TF-IDF features with a linear classifier from scikit-learn. Keyword rules raise the priority of urgent complaints, for example those mentioning no water, fire or accident. If the classifier's confidence is below a threshold, the request is placed in a human help-desk queue instead of being guessed.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

E. Workflow, Records and Output

Each intent is bound to an ordinary Python function registered in a dictionary, so new services are added without changing the rest of the code. Applications, dues, complaints and notices are stored in a relational database, and every status change is written to an audit log with the user, time and reason. Tax amounts are computed from a rate table maintained by the panchayat, and payment is designed around a UPI-compatible gateway, with a test mode used during development. Approved certificates are rendered as PDF files carrying a QR code that leads to a verification page, and the same data feeds the dashboard used by the Sarpanch and Gram Sevak.

V. TECHNOLOGY STACK

The system is implemented with freely available tools, summarised in Table I.

TABLE I: TECHNOLOGIES USED IN THE PROPOSED SYSTEM

Component	Technology	Purpose
Core language	Python 3	Main programming language
Web framework	Flask	Routes, forms and REST APIs
Front end	HTML, CSS, Bootstrap, JavaScript	Responsive citizen and admin screens
Database	SQLite (prototype) / MySQL	Citizens, applications, dues, grievances
NLP	NLTK / spaCy	Tokenisation and intent parsing
Machine learning	scikit-learn	Grievance classification, scheme matching
Documents	ReportLab / qrcode	Certificate PDFs with QR verification
Notifications	SMS gateway API	One-time passwords and status messages

VI. FEATURES AND SUPPORTED REQUESTS

The system currently provides the capabilities listed in Table II. Every feature can be reached from a menu, and the main ones can also be requested in free text through the chatbot.

TABLE II: KEY FEATURES AND EXAMPLE REQUESTS

Feature	Description	Example request
Registration	Mobile OTP login and household record	“Register my family”
Certificates	Birth, death, residence and other certificate applications with status tracking	“Apply for birth certificate”
Tax and fees	House and water dues lookup, online payment and receipt	“Pay my house tax”
Grievances	Complaint with category, priority, ticket ID and status	“Street light not working”
Scheme information	Eligibility questions and matched schemes	“Am I eligible for a housing scheme”
Feature	Description	Example request
Help-desk chatbot	Answers common questions in Marathi, Hindi and English	“Which documents are needed”
Notices	Gram Sabha dates, public notices and alerts	“Show latest notices”
Admin dashboard	Pending queues, dues and grievance trends for officials	Pending applications by ward

The scheme module is rule based. It asks a few eligibility questions, such as household type, land holding and housing status, and lists schemes whose published conditions match, for example the rural housing scheme PMAY-G [20] or rural employment guarantee work. The rules are stored as data, so they can be updated when scheme conditions change.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

VII. APPLICATIONS

- Digital service delivery: certificates, dues and receipts without repeated office visits.
- Grievance management: every complaint receives an ID, an owner and a visible status.
- Scheme awareness: families can check which schemes they may qualify for.
- Transparent administration: public notices, Gram Sabha dates and tax collection summaries in one place.
- Local alerts: messages on water cut-offs, health camps and weather warnings.
- Teaching reference: a small, readable e-governance system for students and developers.

VIII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Free and open source, with no vendor lock-in.
- Saves citizens' travel time and gives them a trackable record of every request.
- Reduces routine sorting work for panchayat staff, while decisions stay with officials.
- Easy to extend with new services, certificate types and schemes.
- Lightweight, and runs on an ordinary computer in the panchayat office.

B. Limitations

- Online use needs a mobile network, and rural connectivity can be unreliable.
- Request understanding is limited to predefined intents and the categories in the training data.
- Existing paper records must be digitised before the dashboard becomes useful.
- Certificates issued by the prototype have no legal standing until it is integrated with the official approval and signature process.

Quantitative evaluation of classification accuracy, response time, service turnaround and usability with real users is planned as part of further work.

C. Comparison with Existing Approaches

Table III summarises, in qualitative terms, how the proposed system differs from the manual process and from existing government portals. The comparison is based on the publicly documented purpose of those portals [7], [8], not on measured benchmarks, and it is intended only to clarify the design space in which this work sits.

TABLE III: COMPARISON OF THE PROPOSED SYSTEM WITH EXISTING APPROACHES

Aspect	Manual process	Existing portals	Proposed system
Service access	Visit office in working hours	Online, mostly statewide forms	Online at village level, plus kiosk at the office
Language	Local language, spoken	Forms in regional language or English	Marathi, Hindi, English with chatbot
Tracking	Register entry, if any	Tracking for notified services	Ticket ID for every request
Grievances	Verbal or on paper	Separate public grievance channels	AI category, priority and routing
Local analytics	Manual registers	Aggregated departmental reporting	Ward-level panchayat dashboard
Cost and hardware	Staff time	Government-run infrastructure	Free libraries, ordinary PC

The principal trade-off is local fit versus statutory integration. National and state portals are connected to official databases and legally recognised processes, which a student-built system cannot replace. The proposed system is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

therefore best seen as a village-level front end and workflow layer that can later be connected to those platforms, and its value lies in local language access, transparency and the use of local data.

D. Request Processing Walk-through

To make the processing flow concrete, Table IV traces four representative requests through the pipeline. The traces illustrate the design of the system and are not measured results.

TABLE IV: EXAMPLE REQUEST TRACES THROUGH THE PIPELINE

Citizen request	Intent and argument	Action	System reply
I need a birth certificate	certificate; type = birth	Opens the birth certificate form	Lists the details and documents required
Pay my house tax	tax; type = house	Fetches dues from the household record	Shows the amount due and a payment option
No water in ward 3 since morning	complaint; category = water; urgent	Creates a ticket and assigns the water official	Ticket number and expected follow-up
Am I eligible for a housing scheme	scheme; type = housing	Asks eligibility questions	Lists matching schemes and documents

In every case the same sequence applies. The language is detected first, the text is normalised, the intent keywords select the handler, and whatever remains becomes its argument. If no intent matches or the confidence is low, the citizen is told that the request has been passed to the help desk and the request enters the human queue.

E. Implementation Challenges and Design Considerations

Several practical issues shaped the design, and they are worth recording because they are common to rural digital services.

- Language and script variation: Marathi may be typed in Devanagari or in Roman letters, and sentences often mix Marathi, Hindi and English. Normalisation and keyword lists in both scripts handle most cases without a large model.
- Limited labelled data: no public dataset of village grievances in Marathi was assumed, so the classifier starts from a small hand-labelled set and keyword rules, and it improves as real complaints are labelled by staff.
- Connectivity: pages are kept light, forms retry on failure, and SMS is used for confirmations so that a weak connection does not lose a request.
- Digital literacy: icons, large buttons and an assisted kiosk at the panchayat office let a helper submit a request on behalf of a citizen.
- Accountability: the AI only suggests a category, priority or owner; approvals and rejections are always made by a named official and logged.
- Intent ambiguity: requests such as “tax receipt” and “pay tax” share keywords. Giving each intent a priority and preferring the longest matching phrase resolves most conflicts without a statistical model.

F. Extensibility and Maintainability

Because each service is an ordinary Python function registered with its keyword set, a new capability, such as a licence application or a notice for a village event, is added by writing one handler and one form. Tax rates, certificate fields and scheme rules live in configuration data rather than in code. The same separation makes the system easy to test: handlers can be exercised with typed text, labelled grievances can be replayed to check the classifier, and the SMS and payment services can be replaced by stubs. Logging every request, matched intent and response time also supplies the raw data for the evaluation described below.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

G. Accessibility Considerations

A web portal does not reach everyone in a village. The interface is therefore text first, with large fonts, icons and the local language, and an assisted mode at the panchayat office or a Common Service Centre lets a helper act with the citizen's consent. Voice input would help low-literacy users, but because speech systems can misrecognise some accents and speakers more often than others [18], voice must be an addition to, and never a replacement for, text and assisted access. Women, the elderly and persons with disabilities should be included in the usability testing.

H. Security and Privacy Considerations

A village administration system holds personal data and can change official records, so security deserves explicit attention. Access is controlled by roles, so that a ward member sees only the grievances of the ward and a citizen sees only their own household. Inputs are validated and database queries are parameterised to guard against the common web risks listed by OWASP [21], connections should use HTTPS, and one-time-password requests are rate limited. Every status change is written to an audit log. Following the principles of the Digital Personal Data Protection Act, 2023 [22], the system collects only the data a service needs, states the purpose, and avoids storing identity numbers that are not required. The set of callable handlers is an explicit allow-list, and the recognised text is never passed to a shell or evaluated as code. Finally, no certificate, payment or approval is ever completed by the AI components alone; an official confirms each legally significant action.

I. Planned Evaluation Methodology

The evaluation planned for the next stage of the work is summarised in Table V. Each metric will be measured on a fixed list of requests covering every feature in Table II, written in Marathi, Hindi and English and in both scripts, and the system will be tested by panchayat staff and residents of different ages.

TABLE V
METRICS PLANNED FOR EVALUATING THE SYSTEM

Metric	Definition	How it is measured
Intent accuracy	Correct intents / total requests	Labelled request list
Grievance category F1	Macro-average of 2PR / (P + R)	Labelled complaint set
Task completion rate	Requests completed as intended	Observed outcome per request
Response time	Submission to first system reply	Timestamps in the program log
Service turnaround	Application to approved certificate	Audit log against the manual process
Usability	Perceived ease of use	System Usability Scale [23]

Here P and R are the precision and recall of each grievance category, and the F1 score is averaged over categories so that rare categories count equally. Reporting these figures for each language and script will show where the current pipeline is weakest and will give a baseline against which later extensions can be compared.

IX. FUTURE SCOPE

- Voice and multilingual interaction: speech input and spoken replies in Marathi and Hindi.
- Identity and document integration: DigiLocker access and OCR-assisted document checking.
- IoT village services: water tank, pump and street-light monitoring feeding the grievance system.
- Predictive analytics: early warning of tax defaults and recurring local problems.

These directions are developed below. Fig. 2 shows how the planned components would extend the present five-stage pipeline while keeping its modular, stage-by-stage structure.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

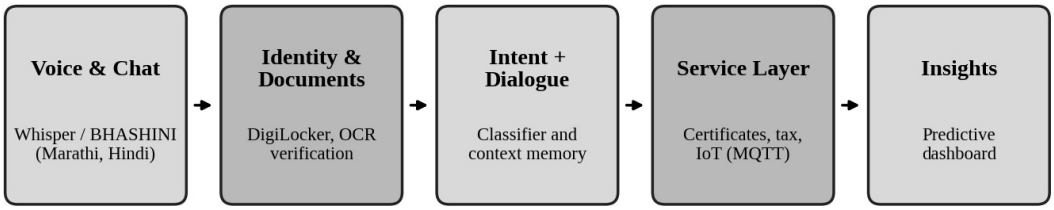


Fig. 2. Proposed extended architecture with voice, identity integration, dialogue memory and insights

A. Voice and Multilingual Support

Adding speech input would let residents who cannot type or read comfortably use the system, and it builds on the authors' earlier voice command assistant [24]. Whisper [15] and the services of BHASHINI [19] cover Hindi and Marathi, but accuracy differs by accent and noise [18], so the evaluation must include a diverse set of speakers, and every spoken request should also be shown as text for confirmation before it is acted upon. The availability of local speech synthesis voices must be verified on each target platform.

B. Identity and Document Integration

Integration with DigiLocker would allow citizens to attach official documents without uploading photographs, and OCR with Tesseract [17] could read printed documents to pre-fill forms and flag mismatches. These aids should only assist the Gram Sevak, who remains responsible for verification. Any use of identity services must follow their legal conditions and the data-protection principles discussed in Section VIII.

C. IoT Integration

Sensors on overhead water tanks, borewell pumps and solar street lights could publish readings through the lightweight MQTT protocol [25] to a Raspberry Pi or ESP32 gateway in the panchayat office. A low water level or a failed light would then open a grievance automatically with the correct category and ward, so that problems are noticed before residents complain. Because these actions affect the physical world, remote control of devices should require confirmation by an authorised official.

D. Predictive Analytics and Learned Intent Models

Stored data on dues, payments and complaints can support simple forecasts, for example which households are likely to pay late or which wards report repeated drainage problems, helping the panchayat plan its work. The keyword table could also be supplemented by a classifier built on a pre-trained transformer such as BERT [14], which tolerates paraphrases that a fixed keyword list would miss. The rule-based matcher should be kept as a fallback because it preserves the transparency that motivates this work.

E. Offline-First and Low-Bandwidth Operation

A progressive web application that stores drafts on the device and submits them when the connection returns, together with SMS or interactive voice fallbacks for status checks, would make the system usable in areas with weak coverage. A local server in the panchayat office could also keep operating when the wider network is down and synchronise later.

F. Generative Language Models

Open-ended questions are the weakest point of a rule-based help desk. A generative language model could be called as an optional fallback when no intent matches, but its answers can be wrong or invented, the call usually needs a network connection and sends the question to an external service, and legal or financial information must be correct. For these reasons it should remain optional, be clearly labelled as such, draw only on approved panchayat documents, and never trigger an action or approval.

G. Scaling Beyond One Panchayat

With per-panchayat configuration of tax rates, forms, schemes and language settings, the same code could serve many villages, with block and district officers viewing consolidated dashboards without access to personal records. Hosting could be on a local server or on government cloud infrastructure, depending on policy.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

TABLE VI
PROPOSED ROADMAP OF ENHANCEMENTS

Phase	Enhancement	Technology	Expected benefit
1	Voice help desk	Whisper or BHASHINI	Access for low-literacy users
2	Identity and documents	DigiLocker, Tesseract OCR	Fewer visits, faster checks
3	IoT village services	MQTT, ESP32 / Raspberry Pi	Early fault alerts
4	Predictive analytics	scikit-learn, BERT classifier	Better local planning

X. CONCLUSION

This paper presented E-Grampanchayat, an AI-powered smart village administration system built with Python. By chaining a citizen portal, request understanding, lightweight AI services, a rule-based workflow and a notification layer, the system lets villagers apply for certificates, pay dues, register complaints and learn about schemes in their own language, and gives panchayat officials a clear view of pending work. The project shows that a useful village-level e-governance system is within reach using open and free tools, and it lays the groundwork for more accessible, transparent and responsive local administration.

The main contribution of the work is a clear, modular reference design. Because every stage of the pipeline is a separate component with a narrow interface, students and developers can replace the classifier, add a service or change a language without touching the rest of the code. The discussion of limitations, privacy and evaluation is intended to make clear what the present system can and cannot do, and to give later work a fair basis for comparison.

Future work will concentrate on measuring classification accuracy, response time and service turnaround with real users, on adding voice interaction and integration with official identity and document services, and on piloting the system in a Gram Panchayat so that its design can be corrected against the realities of village use.

ACKNOWLEDGMENT

The authors thank the Department of Computer Engineering, Adsul's Technical Campus, Ahilyanagar, for providing the facilities and guidance that made this project possible.

REFERENCES

1. Office of the Registrar General and Census Commissioner, India, Census of India 2011: Rural–Urban Distribution of Population. Government of India, 2011.
2. Government of India, The Constitution (Seventy-third Amendment) Act, 1992.
3. Ministry of Electronics and Information Technology, Government of India, "Digital India programme." [Online]. Available: https://www.digitalindia.gov.in/
4. United Nations Department of Economic and Social Affairs, E-Government Survey 2022: The Future of Digital Government. New York: United Nations, 2022.
5. S. Bhatnagar, E-Government: From Vision to Implementation – A Practical Guide with Case Studies. New Delhi: Sage Publications, 2004.
6. R. Heeks, Implementing and Managing eGovernment: An International Text. London: Sage Publications, 2006.
7. Ministry of Panchayati Raj, Government of India, "eGramSwaraj." [Online]. Available: https://egramswaraj.gov.in/
8. Government of Maharashtra, "Aaple Sarkar: Right to Public Services portal." [Online]. Available: https://aaplesarkar.mahaonline.gov.in/
9. Government of Maharashtra, The Maharashtra Right to Public Services Act, 2015.
10. E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," Mach. Learn. Appl.,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

vol. 2, Art. 100006, 2020.
11. S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. Sebastopol, CA, USA: O'Reilly Media, 2009.
12. M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.
13. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
14. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
15. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in Proc. ICML, 2023.
16. M. Grinberg, Flask Web Development, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
17. R. Smith, "An overview of the Tesseract OCR engine," in Proc. Int. Conf. Document Analysis and Recognition (ICDAR), 2007, pp. 629–633.
18. A. Koenecke et al., "Racial disparities in automated speech recognition," Proc. Natl. Acad. Sci. U.S.A., vol. 117, no. 14, pp. 7684–7689, 2020.
19. Digital India BHASHINI Division, "BHASHINI: National Language Translation Mission." [Online]. Available: https://bhashini.gov.in/
20. Ministry of Rural Development, Government of India, "Pradhan Mantri Awaas Yojana – Gramin." [Online]. Available: https://pmayg.nic.in/
21. OWASP Foundation, "OWASP Top Ten web application security risks." [Online]. Available: https://owasp.org/www-project-top-ten/
22. Government of India, The Digital Personal Data Protection Act, 2023.
23. J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry. London, U.K.: Taylor & Francis, 1996, pp. 189–194.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40882