WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Scheme Finder: AI-Powered Government Scheme Eligibility and Recommendation System

Ms. Sasikala P [1], Mrs. Sabeetha M [2] , Akileshwaran N[3], Aksatha R[4], Ananthakrishnan M[5], Mukilan M [6]

[1] (Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India)
[2] (Assistant professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India)
[3-6] (Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India)
[a] Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India
[*] Corresponding Author: akileshwarann61@gmail.com
* *Corresponding author.* Akileshwaran N, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India.

A B S T R A C T:

Government welfare schemes provide financial assistance, education support, healthcare coverage, housing, employment and social security to citizens. In practice, however, information about these schemes is scattered across many central and state portals, eligibility conditions are written in complex administrative language, and most citizens receive no personalized guidance on what they qualify for or which documents they must prepare. As a result, eligible citizens frequently miss benefits they are entitled to. To address this problem, this paper presents Scheme Finder, an AI-powered government scheme eligibility and recommendation system. Scheme Finder integrates: (1) a conversational chatbot that collects personal details in natural language and answers scheme-related questions; (2) an automated eligibility engine that evaluates age, income, occupation, state, education and other criteria against structured scheme rules using deterministic, rule-based logic instead of relying solely on AI-generated decisions; (3) a personalized recommendation module that ranks suitable schemes and supports side-by-side scheme comparison; (4) a document management module offering secure uploads, encrypted storage, optical character recognition (OCR) based text extraction and document readiness checks; and (5) direct Apply Now links to official government application portals. The platform is protected through user registration, password hashing, token-based authentication and per-user access control, and is supported by input validation, graceful error handling and automated testing of the eligibility, recommendation and authentication logic. Delivered through a responsive web interface, Scheme Finder combines AI-powered interaction, reliable eligibility analysis, document assistance and official application links in a single platform, helping citizens discover suitable welfare schemes and understand how to apply for them.

Keywords: Government Schemes, Eligibility Checking, Recommendation System, AI Chatbot, Rule-Based Reasoning, Optical Character Recognition, Token-Based Authentication, Document Management.

1. Introduction

Welfare schemes are among the principal instruments through which governments deliver social and economic support to citizens. Central and state governments in India together operate a very large number of schemes spanning scholarships, agriculture, health insurance, housing, skill development, pensions, entrepreneurship, and women and child welfare. Each scheme is defined by its own benefits, eligibility conditions, documentation requirements and application procedure, and these details are published by different ministries and departments on separate portals, circulars and notifications. For an individual citizen, identifying the few schemes that are actually relevant to his or her circumstances is therefore a demanding exercise in searching, interpretation and cross-checking.

1.1 Background and Problem Formulation

Conventionally, a citizen who wishes to avail a welfare benefit must first learn that a scheme exists, typically through word of mouth, local officials or common service centres. The citizen must then read the scheme guidelines to decide whether he or she is eligible, collect the necessary supporting documents, and locate the correct official portal for applying. This process exhibits four recurring problems: (1) Scattered Information: scheme details are distributed across many portals with inconsistent structure and terminology; (2) Complex Eligibility Criteria: conditions involving age bands, income ceilings, occupation, state of residence, education level, social category and gender are expressed in lengthy administrative text that is difficult for non-experts to evaluate; (3) Lack of Personalized Guidance: portals mostly present generic lists and keyword search and do not tell a particular citizen which schemes suit his or her profile; and (4) Document Uncertainty: citizens are often

unsure which documents are mandatory, whether the documents they hold are legible and acceptable, and what must be arranged before applying.

Manually checking scheme details, required documents and application procedures is time-consuming and may cause eligible citizens to miss important benefits. A centralized AI-powered system is therefore needed to analyze user information, identify suitable schemes, explain eligibility requirements, assist with document preparation and provide direct links to official government application portals.

1.2 System Contributions

To resolve these deficiencies, this work develops Scheme Finder, a full-stack, web-based system that guides a citizen from an initial conversation to the official application page. The principal contributions of this work are summarized as follows:

- AI-Powered Conversational Interface: A chatbot that interacts with users in natural language, collects personal details, answers scheme-related questions and guides users through the scheme discovery process.
- Reliable Rule-Based Eligibility Engine: Automated evaluation of age, income, occupation, state, education and other criteria against structured scheme rules, producing consistent and explainable decisions instead of relying solely on AI-generated judgments.
- Personalized Recommendation and Comparison: Ranking of suitable schemes according to the user profile and eligibility results, with side-by-side comparison of benefits, eligibility conditions and required documents.
- Document Management and OCR: Secure document upload, encrypted storage, text extraction from supported documents and document readiness checks against the requirements of each scheme.
- Direct Official Application Links: An Apply Now option that redirects users to the official government application portal of the selected scheme.
- Secure and Robust Platform: Token-based authentication, protection of personal data, a managed scheme database, input validation, error handling, automated testing and a responsive user interface.

1.3 Paper Organization

The remainder of this paper is structured as follows: Section 2 reviews related work on scheme information portals, conversational assistants and rule-based decision support. Section 3 details the system architecture, data model and the mathematical formulation of eligibility and recommendation scoring. Section 4 elaborates on the functional implementation of the core modules. Section 5 analyzes the security architecture and error-handling strategy. Section 6 presents the testing methodology, rule evaluation and discussion. Section 7 concludes the paper and outlines future enhancements.

2. Related Work and Literature Survey

Digital delivery of public services has been widely studied under the broader theme of e-government. It is widely recognized that making a service available online does not by itself ensure its use: citizens must be aware of the service, understand whether they qualify for it and be able to complete the associated procedure. Information that is scattered across departments and written in administrative language is a recurring barrier to awareness and to the take-up of welfare benefits, which motivates the use of personalized, interactive decision-support tools.

2.1 Existing Scheme Information Portals

National and state-level portals such as the Government of India's myScheme platform provide searchable repositories of schemes with summaries of benefits and eligibility, together with links to the relevant application pages. Department-specific portals, such as those for scholarships or agriculture, handle application submission for individual programmes. These platforms are authoritative sources of information; however, they are organized around schemes rather than around the citizen. Users are expected to know which category or department to search, to interpret eligibility text themselves and to prepare documents without automated checking. Offline intermediaries such as common service centres and local agents partly fill this gap, but they depend on individual availability and expertise. Consequently, a layer of personalized, conversational guidance built on top of official sources remains largely missing.

2.2 Conversational AI and Rule-Based Decision Support

Large language models have made natural-language interaction with information systems practical, enabling assistants that converse fluently, extract structured information from free text and explain complex content in simple language (Brown et al., 2020). Retrieval-augmented generation further grounds model responses in an external knowledge base rather than in the model parameters alone (Lewis et al., 2020). Nevertheless, language models can produce plausible but incorrect statements, which is unacceptable when the output determines whether a citizen believes he or she qualifies for a benefit.

Rule-based decision support, in contrast, evaluates explicit criteria deterministically, yields identical results for identical inputs and can justify each outcome by pointing to the criterion that was met or failed. Scheme Finder therefore adopts a hybrid design: the language model is used for conversation, information extraction and explanation, whereas the eligibility verdict itself is computed by a rule-based engine operating on a structured scheme database.

2.3 Comparative Feature Analysis

Table 1 provides an analytical comparison of Scheme Finder against government scheme portals, generic AI chatbots and manual assistance through agents or service centres across eight functional dimensions.

Table 1 - Comparative Feature Matrix of Scheme Discovery Approaches.

Functional Dimension	Scheme Finder (Proposed)	Government Portals	Generic AI Chatbots	Agents / Service Centres
Conversational guidance	Yes (natural language)	No (search and filters)	Yes	In person
Personalized recommendations	Yes (ranked by profile)	Partial	Unreliable	Depends on agent
Automated eligibility checking	Yes (rule-based)	Partial	Not guaranteed	Manual
Scheme comparison	Yes (benefits, criteria, documents)	Limited	Text only	Verbal
Document upload and OCR	Yes (encrypted storage)	No	No	Manual verification
Document readiness check	Yes	No	No	Manual
Official application link	Yes (Apply Now)	Yes	Not reliable	Assisted
Verified scheme data	Yes (managed database)	Yes	May be outdated	Varies

3. System Architecture and Design Methodology

Scheme Finder is architected as a decoupled, three-tier client-server system designed for reliability, data protection and ease of maintenance. The architecture comprises a Presentation Layer, an Application and Intelligence Layer, and a Data Persistence Layer.

3.1 Architectural Decomposition

The Client Tier is a responsive web interface built with React, HTML5 and CSS3. It provides a chat window, profile forms, scheme result and comparison views, a document upload panel and Apply Now actions, and adapts to desktop, tablet and mobile screens. Client-server communication is mediated through a centralized API client that attaches the authentication token to every request and processes standardized JSON responses and error states.

The Application Tier is implemented with Node.js and the Express framework and is organized into modular route controllers for authentication (/auth), user profile (/profile), chatbot interaction (/chat), eligibility evaluation (/eligibility), recommendations (/recommendations), scheme comparison (/compare), scheme management (/schemes) and document handling (/documents). Incoming requests traverse a middleware pipeline that performs CORS validation, body parsing, input validation, token verification and ownership checks before reaching the business logic. The chatbot controller communicates with an external large language model service for natural-language understanding and response generation, while the eligibility and recommendation services operate purely on structured data and rule logic. The document service integrates an OCR engine (Smith, 2007) to extract text from uploaded files.

The Data Tier stores user accounts, profiles, schemes and the metadata of uploaded documents in a MongoDB database, while the encrypted document files are kept in a protected storage area that is not exposed through public directories.

3.2 Data Modeling and Scheme Representation

The persistence layer is structured around four principal collections. The users collection stores credentials (as password hashes), contact information and profile attributes such as age, gender, annual income, occupation, state, education level, social category and disability status. The schemes collection stores, for every scheme, its name, sponsoring department, level (central or state), category, benefits description, structured eligibility rules, list of required documents and official application URL. The documents collection stores the metadata of each uploaded file, including the owner, document type, encrypted storage reference, extracted text and verification status. The chat sessions collection records the conversation history and the profile information gathered during the dialogue.

Eligibility rules are stored as structured conditions (field, operator, value) rather than free text, so that the same representation can be evaluated by the engine, displayed to the user and updated without modifying program code. Indexes are placed on frequently queried fields such as the user identifier, scheme category and state.

3.3 Mathematical Formulation of Eligibility and Recommendation

To make eligibility decisions objective and reproducible, Scheme Finder formulates two measures for a user profile u and a scheme s: the Eligibility Decision E(u, s) and the Recommendation Score R(u, s).

Each scheme defines a set of n eligibility criteria c_1, ..., c_n, where c_j(u) evaluates to 1 if the profile u satisfies the j-th condition and to 0 otherwise. Mandatory criteria determine eligibility, as defined in Equation (1):

$$E(u, s) = \prod c_j(u),\ j \in M \quad (1)$$

where M is the set of mandatory criteria of scheme s. E(u, s) = 1 means that the user is eligible, whereas E(u, s) = 0 means that at least one mandatory condition is not satisfied; in that case the engine reports exactly which criteria failed. For eligible schemes, the Recommendation Score R(u, s) ∈ [0, 100], defined in Equation (2), ranks schemes by combining profile match, benefit relevance and document readiness:

$$R(u, s) = 100 \times [\ w_m \times (\Sigma\, c_j(u) / n) + w_b \times B(u, s) + w_d \times (D_a / D_r)\ ] \quad (2)$$

where Σ c_j(u) / n is the fraction of all (mandatory and optional) criteria satisfied, B(u, s) ∈ [0, 1] is the relevance of the benefit category of the scheme to the needs expressed by the user, D_a is the number of required documents already available and verified, D_r is the total number of documents required by the scheme, and w_m, w_b and w_d are non-negative weights with w_m + w_b + w_d = 1. Because E(u, s) is computed solely from explicit rules, identical inputs always yield identical eligibility outcomes, and the language model has no influence on the verdict.

4. Functional Implementation and Core Modules

Scheme Finder delivers five functional modules: the Conversational Assistant, the Eligibility Engine, the Recommendation and Comparison module, the Document Management and OCR module, and the Scheme Database and Application Link module.

4.1 Conversational Assistant and Profile Collection

After registering and logging in, the user interacts with the AI chatbot in natural language. The assistant asks for the information required to evaluate eligibility, such as age, annual income, occupation, state of residence, education, social category and gender, one step at a time, extracts the structured values from the replies of the user and stores them in the user profile. Missing or ambiguous answers trigger clarifying follow-up questions instead of assumptions.

Beyond data collection, the assistant answers scheme-related questions, for example about benefits, required documents or the application procedure, using information retrieved from the scheme database so that its answers remain consistent with the verified scheme details. The assistant also guides users through the discovery process by proposing next steps such as viewing recommendations, comparing schemes or uploading documents.

4.2 Automated Eligibility Checking

The Eligibility Engine retrieves the structured rules of each candidate scheme and evaluates them against the user profile as formulated in Equation (1). Supported rule types include numerical ranges (age limits), numerical ceilings (income limits), set membership (state, occupation, category), categorical equality (gender, education level) and Boolean flags (disability, student status). For every scheme, the engine returns the eligibility verdict together with the lists of satisfied and unsatisfied criteria, so that the user can understand why a scheme is or is not suitable and which condition prevents eligibility.

Because the verdict is computed deterministically, the same profile always receives the same result, independent of the wording of the conversation. Missing profile fields are handled explicitly: a scheme whose rules depend on an unknown field is marked as requiring more information, and the chatbot asks the corresponding question.

4.3 Personalized Recommendation and Scheme Comparison

Schemes for which the user is eligible are ranked by the Recommendation Score of Equation (2), and the ranked list is presented with a short summary of the benefit, the matching criteria and the status of the required documents. Users can select two or more schemes and open a comparison view that places benefits, eligibility conditions, required documents and application links side by side, enabling an informed choice without opening multiple portals.

4.4 Document Management and OCR

The Document Module allows users to upload supporting documents such as identity proof, income certificate, community certificate, educational certificates and bank details in supported formats. Each file is validated for type and size, encrypted before being written to storage and linked to the account of its owner. An OCR engine extracts text from supported documents, and the extracted text is used to confirm that the document is legible and of the expected type and to cross-check profile fields.

A document readiness check then compares the documents uploaded by the user with the list required by a selected scheme and reports which documents are available, which are missing and which need to be uploaded again, so that the user knows exactly what to prepare before applying.

4.5 Scheme Database Management and Official Application Links

The Scheme Database holds, for each scheme, its benefits, structured eligibility rules, required documents and official application link. Scheme records can be added, updated or deactivated by authorized maintainers, which immediately affects eligibility checking and recommendations without any change to program code. When a user decides to proceed, the Apply Now option redirects to the official government portal of the scheme. Scheme Finder does not submit applications on behalf of the user; it prepares the user by explaining eligibility, document readiness and the procedure, and then hands over to the official channel.

5. Security Architecture and Deployment Strategy

Because Scheme Finder processes personal, financial and identity-related information, security is treated as an intrinsic property of the platform across the authentication, authorization, storage and input-handling tiers.

5.1 Authentication and Access Control

New users register with their details and a password, which is never stored in plain text: it is hashed with a salted, adaptive hashing algorithm (bcrypt) before storage (Provos & Mazières, 1999). On successful login the server issues a digitally signed JSON Web Token (JWT) with an expiry time (Jones et al., 2015). The client sends the token in the HTTP Authorization header with every request, and protected routes reject missing, invalid or expired tokens with a standard 401 Unauthorized response.
Access control follows the least-privilege principle formalized in role-based access control models (Sandhu et al., 1996). Ownership checks on every profile and document endpoint ensure that a user can access only his or her own profile and documents; attempts to access the resources of another user are rejected with a 403 Forbidden response.

5.2 User Data Protection and Encrypted Document Storage

Uploaded documents are validated for type and size, renamed with unique identifiers to avoid name collisions and path traversal, encrypted with a symmetric algorithm such as AES (NIST, 2001) before being written to storage, and kept outside publicly served directories. Files are decrypted only when the authenticated owner requests them or when the OCR service needs to process them. Sensitive configuration values such as encryption keys, token secrets and API keys are held in environment variables and excluded from source control, and communication between client and server is carried over HTTPS in deployment.

5.3 Input Validation and Error Handling

All incoming data are validated on the server for type, range and format, and database operations use sanitized or parameterized inputs to mitigate the injection attacks listed among the OWASP Top Ten. Invalid inputs return descriptive but non-revealing error messages.
Failures of dependent services, such as the language model service or the OCR engine, are caught and handled gracefully: the system informs the user, avoids exposing internal details and continues to offer the functions that do not depend on the failed service, for example rule-based eligibility checking with the data already present in the profile.

6. Testing, Results and Discussion

To validate the correctness and reliability of Scheme Finder, automated tests were designed for its important functions, with particular emphasis on the eligibility rules, the recommendation logic and authentication, since these determine the correctness of what citizens are told and the protection of their data.

6.1 Automated Testing Strategy

The test suite combines unit tests for individual rule evaluators and scoring functions, integration tests for API endpoints including the authentication flow and ownership checks, and scenario-based tests using representative user profiles. Table 2 lists the principal test scenarios and the behavior expected of the system.

Table 2 - Principal Automated Test Scenarios and Expected Behavior.

ID	Module	Test Scenario	Expected Behavior
T1	Eligibility	User age below the minimum or above the maximum limit of a scheme	Scheme marked not eligible; failed age criterion reported
T2	Eligibility	Annual income exactly equal to the scheme income limit (boundary value)	Boundary handled consistently according to the rule operator
T3	Eligibility	Profile satisfies all mandatory criteria of a scheme	Scheme marked eligible with the list of matched criteria
T4	Eligibility	A profile field required by a rule is missing	Scheme flagged as needing more information; chatbot asks a follow-up question
T5	Recommendation	Several eligible schemes for one profile	Schemes ranked in descending order of score; identical order on repeated runs
T6	Authentication	Registration with an existing email or an invalid password	Request rejected with a validation message
T7	Authentication	Wrong password at login, or protected route accessed with a missing or expired token	401 Unauthorized returned
T8	Data protection	A user requests the profile or document of another user	403 Forbidden returned; no data disclosed
T9	Documents	Upload of an unsupported file type or an oversized file	Upload rejected with a clear error message
T10	Documents	Readiness check for a scheme with only some documents uploaded	Available and missing documents listed correctly
T11	Resilience	Language model or OCR service unavailable	Graceful error message; rule-based features continue to work

6.2 Eligibility Rule Evaluation

Table 3 summarizes the rule types supported by the eligibility engine, the profile attributes on which they operate and an illustrative scheme condition for each.

Table 3 - Eligibility Rule Types and Evaluation Logic.

Rule Type	Profile Attribute	Evaluation Logic	Illustrative Condition
Numeric range	Age	$\min \leq \text{value} \leq \max$	Applicant aged between 18 and 35 years
Numeric ceiling	Annual income	$\text{value} \leq \text{limit}$	Family income not exceeding a stated annual limit
Set membership	State of residence	$\text{value} \in \text{allowed states}$	Permanent resident of a specified state
Set membership	Occupation	$\text{value} \in \text{allowed occupations}$	Farmer, student or artisan
Categorical equality	Gender / social category	value = required value	Women applicants only
Level comparison	Education	$\text{level} \geq \text{required level}$	Minimum Class 10 pass
Boolean flag	Disability / student status	flag = true	Person with disability
Document presence	Required documents	$\text{required} \subseteq \text{uploaded}$	Identity and income proof available

6.3 Usability and Discussion

Qualitative evaluation of the workflow shows that Scheme Finder replaces a multi-step manual process, namely searching several portals, interpreting eligibility text, identifying documents and locating the application page, with a single guided flow. A citizen provides details once in conversation and obtains a ranked list of schemes, an explanation of which criteria are satisfied, a comparison of selected options, a readiness report for the documents and a direct link to apply.

Separating conversation from decision-making yields two practical benefits: the chatbot can remain flexible and user-friendly, while the eligibility verdict remains consistent, explainable and testable. The system also has limitations: the accuracy of recommendations depends on the completeness and currency of the scheme database, OCR quality depends on the clarity of the uploaded images, and the system provides guidance only and cannot guarantee that an application will be approved by the competent authority.

7. Conclusion and Future Scope

This paper presented Scheme Finder, an AI-powered government scheme eligibility and recommendation system that helps citizens discover welfare schemes suitable for them and understand how to apply. By combining a conversational chatbot, rule-based eligibility checking, personalized recommendation and comparison, secure document management with OCR, and direct links to official application portals, Scheme Finder brings the scattered process of scheme discovery into a single platform. Its hybrid design, with AI used for interaction and deterministic rules used for decisions, supports reliable eligibility analysis, while token-based authentication, encrypted document storage, input validation and automated testing support security and robustness.

Future enhancements will explore: (1) Multilingual and Voice Interaction: supporting regional languages, including Tamil, and speech input to serve citizens with limited literacy or English proficiency; (2) Automatic Scheme Updates: integrating with official sources to keep scheme data and eligibility rules current, with change notifications to users; (3) Application Tracking and Reminders: alerts for deadlines, missing documents and application status; and (4) Wider Coverage and Accessibility: extending the scheme database to additional states and departments and offering a mobile application.

Acknowledgements

The authors express sincere gratitude to their guide, Ms. P. Sasikala, Assistant Professor, Department of Computer Science and Engineering, and to the Principal, Dr. N. M. Saravana Kumar, for their constant guidance, encouragement and support, and to the Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, for providing the facilities required to carry out this work.

REFERENCES

[1] Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

[2] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

[3] Smith, R. (2007). An overview of the Tesseract OCR engine. Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR), 629–633.

[4] Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT). RFC 7519, Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc7519

[5] Provos, N., & Mazières, D. (1999). A future-adaptable password scheme. Proceedings of the USENIX Annual Technical Conference.

[6] National Institute of Standards and Technology. (2001). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197.

[7] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. IEEE Computer, 29(2), 38–47.

[8] OWASP Foundation. (n.d.). OWASP Top Ten. https://owasp.org/www-project-top-ten/

[9] Government of India. (n.d.). myScheme: National platform for government schemes. https://www.myscheme.gov.in/

[10] React. (n.d.). React documentation. https://react.dev/

[11] OpenJS Foundation. (n.d.). Node.js documentation. https://nodejs.org/docs/

[12] OpenJS Foundation. (n.d.). Express documentation. https://expressjs.com/

[13] MongoDB, Inc. (n.d.). MongoDB documentation. https://www.mongodb.com/docs/