WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

JanSetu AI: An AI-Assisted Web Platform for Personalised Discovery of Indian Government Welfare Schemes

Vaishnavi S[1], Vaishnihaa S[1], Varshini R[1], Vasantha Mani S[1], Ms. S. Abishree[1,*]

[1]Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology (An Autonomous Institution), Coimbatore – 641062, Tamil Nadu, India. *Project guide, Assistant Professor, Department of CSE.

ABSTRACT :

Central and state governments in India run a large number of welfare schemes, scholarships and subsidies, yet many eligible citizens never use them because scheme information is spread over many portals and written in eligibility language that is hard to interpret. This paper presents JanSetu AI, a web platform that lets a citizen describe their circumstances once and then see the government schemes that may apply to them. The user completes a three-step profile form (age and gender; education, occupation and category; income and location). A rule-based weighted matching engine compares the profile with structured eligibility criteria stored for each scheme (age range, occupation, income bracket, state of domicile and gender) and ranks the schemes by an indicative match score with human-readable match reasons. Each scheme page shows benefits, required documents, an application roadmap with an interactive document checklist, and a link to the official portal. The system is implemented as a React single-page application with a Node.js/Express REST back end, MongoDB storage, JWT-based authentication and PBKDF2 password hashing, and is deployed on Netlify. Results are presented as functional outcomes of the working prototype; no statistical accuracy claim is made. The platform also includes a conversational assistant to help users understand scheme-related information and eligibility requirements. The responses are intended to support scheme discovery and should be verified against official government sources.

Keywords: Government schemes, eligibility matching, recommendation, scheme discovery, e-governance, digital inclusion, React, Node.js, MongoDB

1. Introduction

Governments provide schemes, scholarships, subsidies and welfare programmes to improve education, employment, healthcare, entrepreneurship and financial security. In India these programmes are issued by many central and state departments, and the information about each of them is published on separate departmental sites, scholarship portals, public-service sites and official announcements. A citizen therefore has to know which department is responsible, visit several websites, and read lengthy eligibility conditions before even deciding whether to apply. People with limited digital literacy, first-time users and those who do not know a scheme by name are the most likely to miss benefits they qualify for.

Centralised portals such as myScheme and the National Portal of India have made discovery easier, and the National Scholarship Portal consolidates scholarship services. Users may still need help understanding conditions, comparing schemes and finding the practical next step for a particular programme. JanSetu AI (the name combines the Hindi words for "people" and "bridge") is built to narrow that gap: a single interface that collects a small profile, shows potentially relevant schemes in order of fit, explains why each was matched, and guides the user to the official application channel.

The objectives of the work are: (1) to simplify scheme discovery through one web interface; (2) to support personalised discovery from the user's age, gender, education, occupation, income and location; (3) to present benefits, eligibility, required documents and application steps in a structured form; (4) to reduce the effort and time needed to search several portals; and (5) to keep the interface usable for people with different levels of digital knowledge. JanSetu AI is a discovery aid and does not replace official portals; final eligibility and approval always rest with the responsible authority.

2. Literature Review

2.1 Existing government scheme portals

The National Portal of India provides government information and includes facilities for searching schemes. The myScheme platform, a Government of India initiative under Digital India, goes further: users enter personal and demographic details and receive a list of schemes with eligibility, benefits and application procedures. It demonstrates the value of eligibility-based discovery and of presenting benefits through one user-oriented interface [2,3,6]. The National Scholarship Portal brings many scholarship schemes together and supports application, verification and disbursement, showing how education-related assistance can be organised centrally and why scheme-specific conditions such as academic qualification, category and income must be shown clearly [4].

2.2 Eligibility-based discovery and recommendation

Eligibility-based discovery stores scheme information as structured fields and compares a user's details with each scheme's conditions. Recommender systems more generally help users find items relevant to their needs and can be rule-based, content-based or data-driven [7]. For public schemes, the quality of recommendations depends on the freshness of scheme data, the relevance of the matching logic and the accuracy of the details the user enters. Because schemes contain exceptions and documentation requirements, a displayed match should be treated as guidance, not as a guarantee of eligibility.

2.3 Challenges and research gap

The literature and the portals themselves point to recurring difficulties: information scattered across sources, complex and differing eligibility conditions, differing application procedures, frequently changing details, limited digital literacy, and keyword search that returns related but unsuitable results. Although myScheme already provides personalised discovery, users may still need help understanding conditions, comparing opportunities and identifying the official next step. JanSetu AI addresses this gap by combining profile-based eligibility matching, transparent match explanations, structured scheme details, an application roadmap and a document checklist. The platform also provides conversational assistance to help users understand scheme-related information. It complements existing government portals by directing users to official sources for verification and application.

3. Proposed System

Existing practice requires users to search multiple sites, read each scheme's rules, and decide manually which apply. Its main drawbacks are scattered information, a time-consuming search, difficulty interpreting eligibility criteria, limited personalisation, low awareness of available benefits, difficulty comparing schemes, digital-literacy barriers, the risk of relying on outdated information, and repeated effort when searching for several family members or needs.

JanSetu AI replaces this with a single guided workflow: sign in, complete a short profile, view ranked schemes with reasons, open a scheme to see its details and roadmap, and continue to the official portal. The platform collects only the details needed for matching, and displays a notice on the results page that scores are indicative.

4. Methodology

4.1 System architecture and technology stack

The system follows a client–server design. The front end is a React single-page application (HTML5, CSS3, JavaScript) with a dark glassmorphism interface, responsive layout and client-side routing. The back end is a Node.js/Express REST service with separate authentication and scheme routes and a health-check endpoint. User accounts and scheme records are stored in MongoDB through Mongoose models. The front end is hosted on Netlify. The platform also includes a conversational assistance component for scheme-related queries, subject to the capabilities of the implemented chatbot service. Table 1 summarises the stack.

Table 1. Technology stack of JanSetu AI

Layer	Technologies / purpose
Front end	React.js, react-router-dom (protected routes), HTML5, CSS3, JavaScript, responsive design
Back end	Node.js, Express, CORS, dotenv; RESTful API (/api/auth, /api/schemes, /api/health)
Database	MongoDB with Mongoose: user accounts, demographic profile, saved schemes, scheme records and eligibility criteria
Security	JSON Web Tokens (7-day expiry) for sessions; PBKDF2-SHA256 (100,000 iterations) with a per-user random salt for passwords; token-verification middleware
Deployment and tools	Netlify, Git/GitHub, VS Code, MongoDB Compass, Chrome

4.2 User workflow

Figure 1 shows the end-to-end user and administrator workflow. A user opens the website, registers or logs in, and completes email verification if required by the authentication flow. Protected pages are accessible only to authenticated users. The user then completes the three-step scheme-finder form: (1) age and gender; (2) education, occupation and special category; and (3) income and location. The application validates the submitted information and compares it with the stored scheme criteria. Matching results are displayed with indicative scores and scheme details. Users can open a scheme page, review its benefits and application roadmap, track required documents through the checklist, and follow the official application link. The conversational assistant provides additional support for scheme-related questions. Administrative functions, where implemented, support scheme-data maintenance and user management.

The prototype is deployed and its main screens are shown in Figures 2–5.

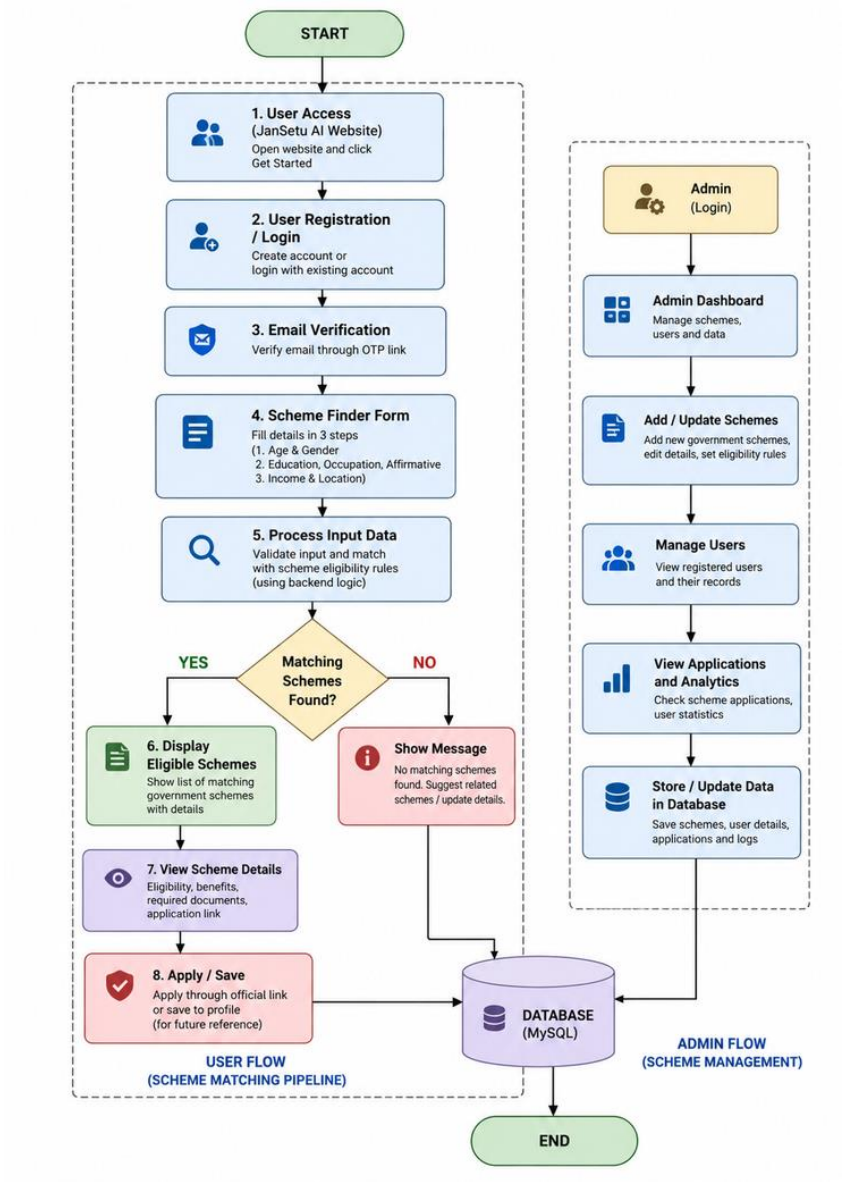


Fig. 1. Flowchart of JanSetu AI (user flow and administrator flow)

4.3 Data model

Each user document stores a lower-cased unique email, password hash and salt, creation date, a profile (age, gender, education level, occupation, income bracket, state, area type, special tags and a land-owner flag) and a list of saved scheme identifiers. Each scheme carries its name, description, department, category, level (central or state), benefits, required documents, application link and a structured eligibilityCriteria object containing minimum and maximum age, permitted occupations, income brackets, states and gender. On the client, each account's profile is kept under an account-specific key so that data of different users on the same browser stay separate.

4.4 Eligibility matching algorithm

The matching engine is a transparent rule-based scoring method rather than a trained machine-learning model; "AI" in the platform's name refers to this intelligent matching and the planned assistant features. For each scheme the engine evaluates five criteria and adds a weight when the user's profile satisfies the criterion (Table 2). The total (out of 100) is converted to a percentage, bounded between 38 and 98, and each satisfied criterion adds a plain-language reason (for example, "Age 24 falls within the eligible range"). Schemes are sorted in descending order of score and labelled "Potentially Eligible" or, at 80 or above, "Potentially Eligible (High Match)". Whenever the profile changes, the context provider recomputes the matches.

Table 2. Criteria and weights used by the matching engine

No.	Criterion	Weight	Rule
1	Age	20	minAge ≤ age ≤ maxAge
2	Occupation	30	scheme lists the user's occupation or "all"
3	Income bracket	20	user's bracket is among the scheme's permitted brackets
4	Jurisdiction	20	scheme is pan-India ("ALL") or lists the user's state
5	Gender	10	scheme is open to all or lists the user's gender

The steps of the overall algorithm are: (i) input acquisition; (ii) preprocessing and validation of mandatory and malformed fields; (iii) matching against scheme criteria; (iv) retrieval of scheme details from the data store; (v) search and filtering; (vi) rendering of results and navigation; (vii) frontend–backend communication through the REST API; (viii) presentation of results with official links for verification; (ix) error handling and repeated searches; and (x) deployment and availability testing.

4.5 Search, filtering and authentication

A public search endpoint filters schemes by category, level (central or state), state and a case-insensitive keyword matched against name, description and department, using MongoDB queries. Registration requires a valid email and a password of at least six characters, rejects duplicates (HTTP 409) and returns a signed token. Login recomputes the PBKDF2 hash with the stored salt and returns identical error messages for unknown email and wrong password to avoid disclosing which accounts exist. A middleware verifies the bearer token on protected routes (HTTP 403 when absent, 401 when invalid or expired).

5. Results and Discussion

5.1 Implemented functionality

The prototype is deployed and its main screens are shown in Figures 2–5 The prototype is deployed and its main screens are shown in Figures 2–5. The home page introduces the platform and guides users towards scheme discovery. The scheme finder presents a three-step profile form for collecting relevant demographic and socioeconomic information. The results page displays an indicative qualification notice, category filters, a search option and scheme cards containing match percentages, descriptions and key eligibility details. Users can access additional information through the Quick View and View Details options. Each scheme detail page presents benefits, an application roadmap and an interactive document checklist, together with a link to the relevant official portal where available. The conversational assistant provides additional support for scheme-related queries. These features describe the functionality represented in the project materials; their accuracy and effectiveness require further systematic evaluation.

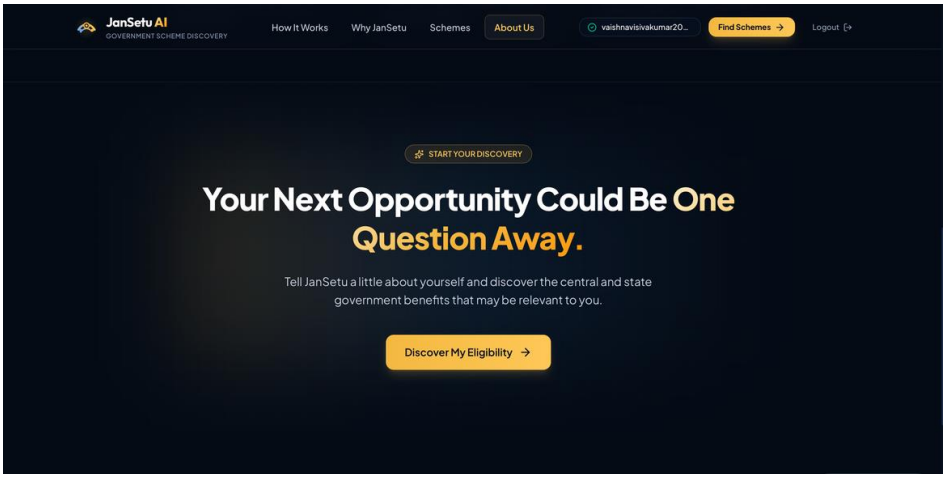


Fig. 2. Home page of JanSetu AI

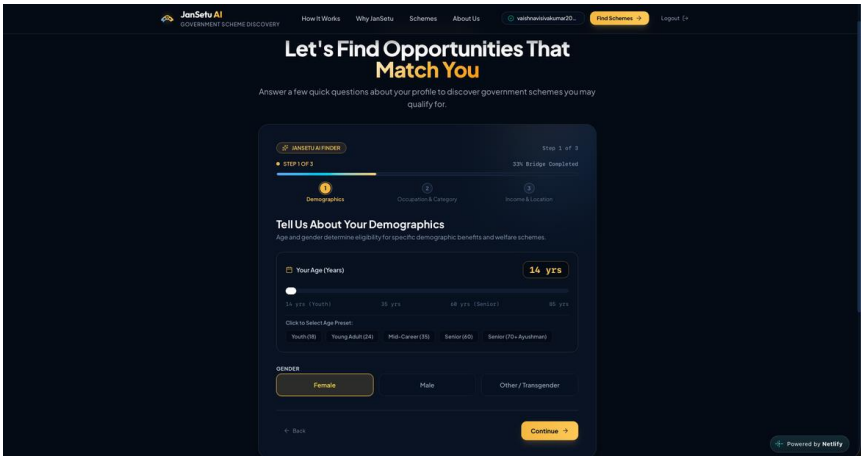


Fig. 3. Scheme finder form (step 1 of 3: age and gender)

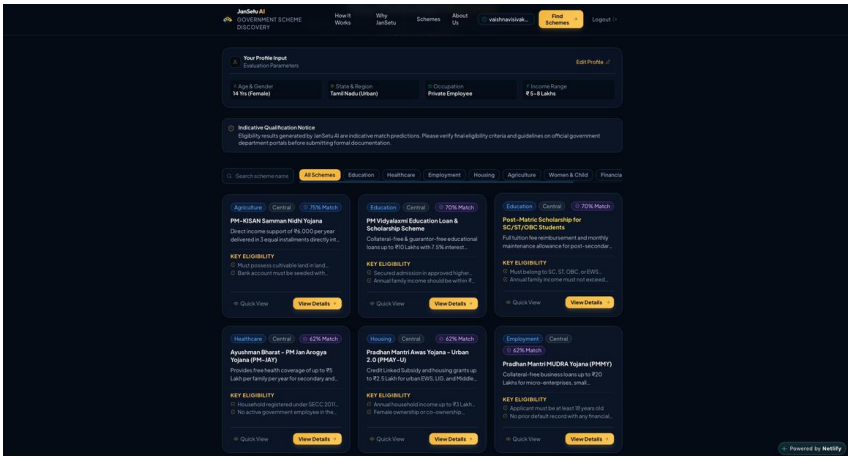


Fig. 4. Eligible schemes with match percentage and key eligibility

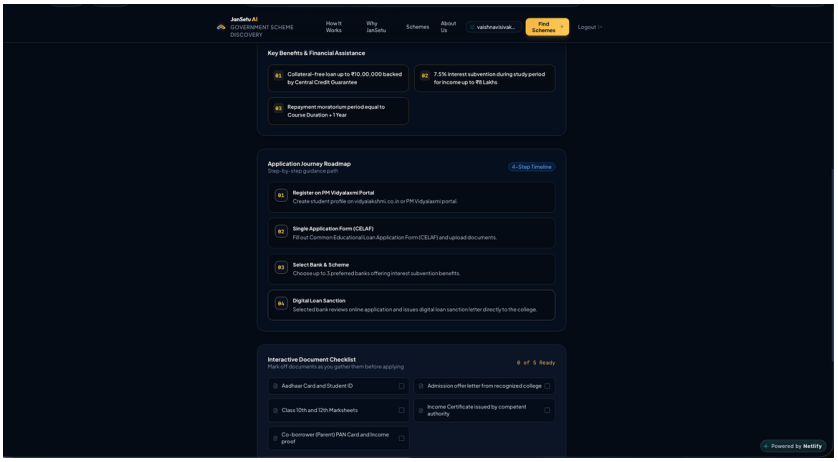


Fig. 5. Application roadmap and interactive document checklist for a selected scheme

5.2 Discussion

The prototype shows that a short structured profile and per-scheme criteria are enough to turn a long search across portals into a ranked list, and that showing why a scheme matched, what it offers, which documents are needed and where to apply gives the user a concrete next step. Presenting these elements together addresses the drawbacks identified for the existing approach: scattered information, difficulty interpreting criteria and weak personalisation. Account-isolated profiles and token-based sessions protect users' personal details.

The work also has limitations that should be stated plainly. First, no quantitative evaluation (for example precision of matches or user-study scores) has yet been carried out, so the paper claims functional feasibility rather than measured accuracy. Second, the engine is additive: a scheme that fails a hard condition, such as an age limit, still receives partial credit and is listed with a lower score, and the lower bound of 38% means every scheme is shown as

potentially eligible. Hard-exclusion rules would make the ranking stricter. Third, when a field is missing the engine falls back to default values (age 22, student, low income, a default state), which may produce misleading matches if the form is not completed. Fourth, the quality of results depends on the completeness and currency of the scheme data, which currently has to be maintained manually. For these reasons results are labelled indicative and users are directed to verify on official portals.

6. Conclusion and Future Work

JanSetu AI is a web platform that helps citizens discover government schemes relevant to their profile through a single interface. It combines a three-step profile form, a transparent weighted eligibility-matching engine, structured scheme details, an application roadmap with a document checklist and links to official portals, on a secured React–Express–MongoDB stack. It reduces the effort of searching several sources and supports informed decisions while leaving final eligibility to the authorities.

Future improvements include implementing hard-exclusion rules for mandatory eligibility conditions, handling missing profile information without misleading defaults, conducting formal evaluations using verified test profiles and user feedback, adding regional-language support, improving chatbot reliability and source attribution, and maintaining up-to-date scheme information through authorised sources. Further work may also focus on accessibility, mobile usability, privacy protection and security improvements.

REFERENCES :

1. JanSetu AI – AI-Powered Government Scheme Discovery Platform, project website. Available: https://jansetu-aischeme.netlify.app/
2. Government of India, "myScheme – National Platform for Government Schemes." Available: https://www.myscheme.gov.in/
3. Government of India, "National Portal of India – Government Schemes." Available: https://www.india.gov.in/my-government/schemes
4. Ministry of Education, Government of India, "National Scholarship Portal." Available: https://scholarships.gov.in/
5. MDN Web Docs, "Web Development Documentation." Available: https://developer.mozilla.org/
6. Digital India, Ministry of Electronics and Information Technology, Government of India, "myScheme Initiative." Available: https://www.digitalindia.gov.in/initiative/myscheme/
7. F. Ricci, L. Rokach and B. Shapira (Eds.), Recommender Systems Handbook, 2nd ed., Springer, 2015.