IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

An Integrated E-Procurement System for Government Tender

Abhishek Kumar G[1], Dr. K. Santhi[2]

Department Of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore, Tamil Nadu, India[1]

Professor, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore, Tamil Nadu, India[2]

Abstract: The rapid digitization of public sector operations has transformed government procurement processes, but it has also introduced challenges such as information overload, fragmented systems, and limited personalization. Traditional e-procurement platforms mainly function as information repositories and require manual effort for schedule planning, compliance verification, and bid preparation, thereby reducing operational efficiency [1]. Recent advancements in artificial intelligence (AI), particularly large language models (LLMs), have demonstrated strong capabilities in automating decision-support tasks and generating intelligent recommendations in real time [3].

This paper proposes an Integrated AI-Driven E-Procurement System for Government Tenders, a web-based platform designed to streamline the tendering lifecycle through automation and conversational intelligence. Developed using the Flask framework and integrated with the Open Router API (GPT-3.5-turbo), the system enables dynamic generation of customized tender schedules, document insights, compliance guidance, and portal recommendations. Unlike conventional portals that provide static listings, the proposed system incorporates a context-aware AI assistant capable of interpreting natural language queries and generating personalized bidding strategies.

The platform also includes secure user authentication with encrypted password storage using SQLite and automated email notifications upon login. Experimental validation indicates that the system reduces manual effort in procurement planning while improving transparency and decision-making quality. Future enhancements include integration with live government procurement APIs, multilingual support, and automated tender document parsing.

Keywords: E-Procurement, Government Tender, Artificial Intelligence, Large Language Model, Decision Support System, Chatbot, Flask, Open Router.

I. INTRODUCTION

Government procurement plays a crucial role in national economic development, as it governs the acquisition of goods, services, and infrastructure projects through transparent and competitive tendering mechanisms. Traditionally, public procurement processes were paper-based and highly bureaucratic, involving multiple approval stages, physical documentation, and manual bid evaluations. Although e-Government initiatives have digitized many procurement functions, existing e-tendering platforms primarily serve as information dissemination systems rather than intelligent decision-support tools [1].

Modern e-procurement portals such as centralized public procurement websites allow vendors to view tender notices, download documents, and submit bids online. However, vendors still face significant challenges in understanding eligibility requirements, preparing compliance documents, generating schedules, and selecting appropriate portals. These processes often require manual interpretation of lengthy tender documents, resulting in inefficiencies and increased risk of non-compliance [2]. Furthermore, most platforms provide static information and lack conversational assistance or real-time personalization.

Recent advancements in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP) and Large Language Models (LLMs), have introduced new possibilities for automating complex advisory and planning tasks. Transformer-based models such as GPT have demonstrated strong capabilities in contextual understanding, content generation, and decision-support assistance across domains [3]. These technologies enable systems to interpret user queries, generate structured schedules, summarize documents, and provide intelligent recommendations without extensive domain-specific rule programming.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

This paper introduces an **Integrated E-Procurement System for Government Tenders**, a web-based application that leverages AI to enhance procurement planning and vendor support. The system is developed using the Flask framework and integrates the OpenRouter API to access the GPT-3.5-turbo model for intelligent response generation. Unlike conventional tender portals, the proposed system provides five integrated modules: AI Chat, Tender Schedule Generator, Tender Insights, Portal Suggestions, and Compliance Tips. These modules collectively support the entire bidding lifecycle, from initial inquiry to final submission guidance.

The application also incorporates secure user authentication with password hashing and email notification mechanisms to ensure security and accountability. By combining AI-driven advisory features with a modular web architecture, the system aims to reduce administrative burden, improve transparency, and enhance decision-making efficiency in public procurement processes [4].

The remainder of this paper is organized as follows: Section 2 reviews related work in e-governance procurement systems and AI-assisted platforms. Section 3 defines the problem statement. Section 4 describes the proposed system architecture and modules. Section 5 details the system workflow. Section 6 presents implementation results and performance analysis. Section 7 concludes the paper and discusses future enhancements.

II. LITERATURE SURVEY

Several research studies and practical systems have explored the development of electronic tendering and procurement solutions within e-Government frameworks. However, most existing works focus on specific components such as bid evaluation, system interoperability, or decision-support models rather than providing an integrated intelligent procurement assistant.

Kayed and Colomb [1] discussed the evolution of electronic tendering within Business-to-Business (B2B) e-commerce environments, highlighting the importance of digitizing procurement workflows to improve transparency and efficiency. Their work emphasized the transformation from manual paper-based tendering to web-enabled platforms but noted that backend decision processes often remained partially manual. This study laid the foundation for understanding e-tendering as a structured electronic workflow rather than merely an online notice board.

Kerridge et al. [2] proposed a Virtual Tendering and Bidding system for the construction sector that supports electronic submission and management of bids. Their system automated portions of the bidding lifecycle and demonstrated improved coordination among stakeholders. However, the model was industry-specific and lacked intelligent advisory capabilities such as automated schedule planning or conversational assistance.

Ahmad et al. [8] introduced a Decision Support System (DSS) for tender evaluation by integrating statistical single-criteria models with Multi-Criteria Decision Making (MCDM) techniques. Their research focused on improving fairness and objectivity in bid evaluation processes. While the system enhanced evaluation accuracy, it primarily concentrated on backend scoring mechanisms and did not address vendor-side planning or real-time guidance.

Brown et al. [3] demonstrated the effectiveness of large language models in performing few-shot learning tasks, enabling systems to generate structured content, summaries, and contextual responses without domain-specific training. This breakthrough significantly expanded the application of AI in governance systems, including document analysis and automated advisory support. The adoption of LLMs provides opportunities to move beyond rule-based procurement systems toward intelligent conversational assistants.

Open Router Documentation [4] presents a unified API framework that allows seamless integration of multiple large language models into web applications. This architecture simplifies the deployment of AI-driven services within lightweight frameworks such as Flask. By leveraging such APIs, modern e-procurement systems can dynamically generate insights, compliance recommendations, and personalized schedules without maintaining complex in-house AI infrastructure.

The existing literature indicates substantial progress in digitizing tender workflows and enhancing bid evaluation mechanisms. However, there remains a research gap in integrating AI-based conversational assistance, schedule automation, compliance guidance, and portal recommendations into a single unified procurement platform. The proposed system addresses this gap by combining decision-support intelligence with an interactive web-based architecture.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

III. PROBLEM STATEMENT

Despite significant advancements in e-Government initiatives, existing e-procurement systems still face multiple operational and usability challenges. Most government tender portals primarily function as centralized repositories for publishing notices and documents, requiring vendors to manually interpret eligibility criteria, financial conditions, and technical specifications. This often leads to confusion, misinterpretation, and increased risk of non-compliance during bid submission [1].

One major challenge is **information overload**. Vendors must navigate through numerous tender notifications, corrigenda, policy updates, and regulatory guidelines to identify relevant opportunities. The absence of intelligent filtering and contextual assistance makes the selection process time-consuming and inefficient. Additionally, most platforms provide static PDF documents without offering structured summaries or actionable insights.

Another critical issue is the **lack of personalization**. Traditional systems do not consider vendor capabilities, budget constraints, project timelines, or previous bidding history when presenting recommendations. As highlighted in decision-support research, procurement systems require intelligent mechanisms that can evaluate multiple parameters dynamically to assist stakeholders effectively [8]. However, such intelligence is rarely integrated into publicly accessible tender portals.

Furthermore, the tendering lifecycle involves multiple fragmented tools, including procurement portals, document preparation software, digital signature utilities, and banking platforms for fee payments. The need to switch between these systems results in a disjointed user experience and increased administrative workload. Existing platforms also lack conversational interfaces that allow users to clarify doubts in real time.

Therefore, there is a clear need for a unified, AI-driven e-procurement platform capable of understanding natural language queries, generating personalized tender schedules, offering compliance guidance, and recommending relevant procurement portals within a single integrated interface. The objective of this project is to design and implement such a system, demonstrating how large language models can enhance transparency, efficiency, and decision-making in government tender management.

IV. PROPOSED SYSTEM

4.1 OVERVIEW

The proposed system is a web-based AI-powered E-Procurement platform designed to assist vendors in managing government tenders efficiently. It integrates traditional tender portal features with intelligent advisory support to simplify the bidding process. The system is developed using Python Flask for backend processing and HTML, CSS, and Bootstrap for the frontend interface. It includes five main modules: AI Chat, Tender Schedule Generator, Tender Insights, Portal Suggestions, and Compliance Tips. These modules help users understand tender requirements, generate schedules, receive compliance guidance, and access relevant procurement portals.

The application connects to the Open Router API to generate AI-based responses dynamically. User authentication is secured using password hashing, and credentials are stored in an SQLite database. Email notifications are sent upon successful login to ensure security. The overall goal of the system is to provide a simple, centralized, and intelligent platform that enhances efficiency and decision-making in government tender management.

4.2 SYSTEM ARCHITECTURE

The system architecture follows a client–server model where the user interacts with the application through a web browser. The browser sends HTTP requests to the Flask backend server, which processes user inputs, manages sessions, and coordinates communication with external services.

At the frontend level, the system uses HTML templates styled with CSS and Bootstrap to provide a responsive and user-friendly interface. JavaScript is used for basic dynamic features such as form validation and interactive elements. All user inputs from modules such as Chat, Schedule Generator, and Insights are submitted to the Flask backend through defined routes.

The backend, developed using the Flask framework, acts as the central processing unit of the system. It handles user authentication, session management, form processing, and prompt construction for AI requests. When a user submits a query or form, the backend builds a structured prompt and sends it to the Open Router API.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

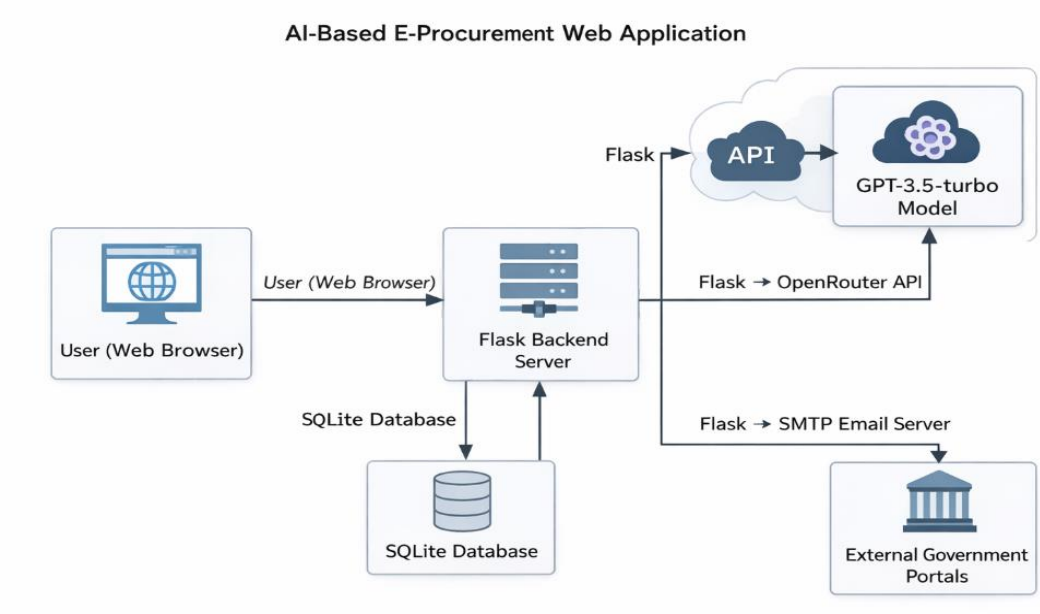

Figure 1

The AI Integration layer connects to the Open Router API, which accesses the GPT-3.5-turbo language model. The model processes the prompt and generates a context-aware response. This response is returned to the Flask server and rendered back to the user interface.

The system also includes an SQLite database to store user credentials such as username, email, and hashed passwords. This ensures lightweight and efficient data storage. Additionally, an SMTP email server is integrated to send login notifications to users for security purposes.

External Government Portals form another component of the architecture. Based on user input, the system generates and displays relevant portal links (such as GeM, CPPP, or state portals), allowing users to access official tender platforms directly.

Overall, the architecture ensures modularity, scalability, and secure communication between the frontend interface, backend processing unit, AI services, database, and external procurement portals.

4.3 MODULES DESCRIPTION

The proposed system is divided into functional modules to ensure modularity, scalability, and ease of maintenance. Each module is designed to support a specific stage of the government tendering lifecycle while integrating intelligent AI-driven assistance.

4.3.1 User Authentication Module

The User Authentication module ensures secure access to the system. It provides registration and login functionalities for vendors and users. During registration, users must provide a username, email address, and password. The password is securely hashed using standard cryptographic hashing techniques before being stored in the SQLite database, thereby preventing unauthorized access in case of data exposure.

During login, user credentials are validated against the stored records. Upon successful authentication, a session is created to maintain user state throughout system interaction. Additionally, an automated email notification is sent to the registered email address via SMTP, enhancing security awareness. Secure authentication mechanisms are critical in e-Government systems to ensure data privacy and integrity [1].

4.3.2 AI Chat Module

The AI Chat module functions as an intelligent conversational assistant that allows users to ask procurement-related queries in natural language. Users can seek clarification regarding tender terminology, eligibility criteria, financial requirements, or submission procedures. The backend constructs a structured system prompt defining the AI's role as a Government Procurement Expert and forwards the query to the Open Router API.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

The GPT-based language model processes the input and generates context-aware responses. Large Language Models have demonstrated strong capabilities in understanding domain-specific queries and producing structured, informative outputs without extensive domain-specific training [2]. This module significantly reduces dependency on manual interpretation of tender documents and enhances user accessibility.

4.3.3 Tender Schedule Generator Module

The Tender Schedule Generator assists vendors in planning the tender submission timeline. Users input details such as tender title, category, start date, submission deadline, and estimated preparation time. Based on these inputs, the system generates a structured timeline that includes important stages such as Notice Inviting Tender (NIT) review, pre-bid meetings, document preparation, technical evaluation, financial bid opening, and final submission.

The AI dynamically generates step-by-step schedules tailored to the project duration and complexity. This automation reduces planning errors and ensures timely compliance with submission deadlines. By leveraging AI-based text generation, the system transforms raw input parameters into actionable procurement plans.

4.3.4 Tender Insights Module

The Tender Insights module provides structured summaries of technical requirements, financial eligibility, compliance rules, and common pitfalls associated with a specific tender category. Users can enter a tender ID or project type (e.g., Road Construction, IT Hardware Supply), and the AI generates detailed insights.

Traditional e-tender systems typically present information in lengthy PDF documents, requiring manual review. Decision-support systems in procurement have been shown to enhance clarity and objectivity in evaluation and interpretation processes [3]. This module applies similar principles on the vendor side, enabling better preparation and reducing the likelihood of disqualification due to incomplete documentation.

4.3.5 Portal Suggestions Module

Government tenders are distributed across multiple platforms such as national, state, and sector-specific portals. The Portal Suggestions module analyses user input such as region and project type to recommend appropriate procurement portals. It also provides brief guidance on how to use the suggested platform effectively.

This feature reduces the fragmentation commonly experienced by vendors who must manually search across various government websites. By centralizing portal guidance within the system, the module improves accessibility and navigation efficiency.

4.3.6 Compliance Tips Module

The Compliance Tips module generates practical advisory tips for successful bid submission. Upon activation, the AI produces concise yet valuable recommendations such as ensuring valid Digital Signature Certificates (DSC), verifying document formats, checking Earnest Money Deposit (EMD) requirements, and adhering to submission deadlines. Compliance-related errors are among the most common causes of bid rejection. Providing proactive AI-generated tips enhances vendor preparedness and promotes transparency in procurement processes.

Overall, the modular architecture ensures that each component operates independently while contributing to a unified AI-driven procurement assistance platform. The integration of secure authentication, intelligent conversation, schedule automation, document insights, and compliance guidance makes the system a comprehensive solution for modern e-Government tender management.

V. SYSTEM WORKFLOW

The System Workflow describes the sequence of operations performed from user access to AI response generation and output rendering. The workflow ensures smooth interaction between the frontend interface, backend processing, AI services, and database components.

5.1 Overall Operational Flow

The workflow begins when a user accesses the web application through a browser. The user is directed to the login page, where new users can register and existing users can log in. During registration, user credentials are securely stored in the SQLite database after password hashing. Upon successful login, a session is created and an email notification is sent via the SMTP server for security verification [1].

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

After authentication, the user is redirected to the homepage, which displays the available modules: AI Chat, Tender Schedule Generator, Tender Insights, Portal Suggestions, and Compliance Tips. The user selects a module based on their requirement.

When a user submits a query or form input (for example, generating a schedule or requesting tender insights), the frontend sends an HTTP POST request to the Flask backend. The backend processes the input, constructs a structured prompt, and forwards it to the Open Router API.

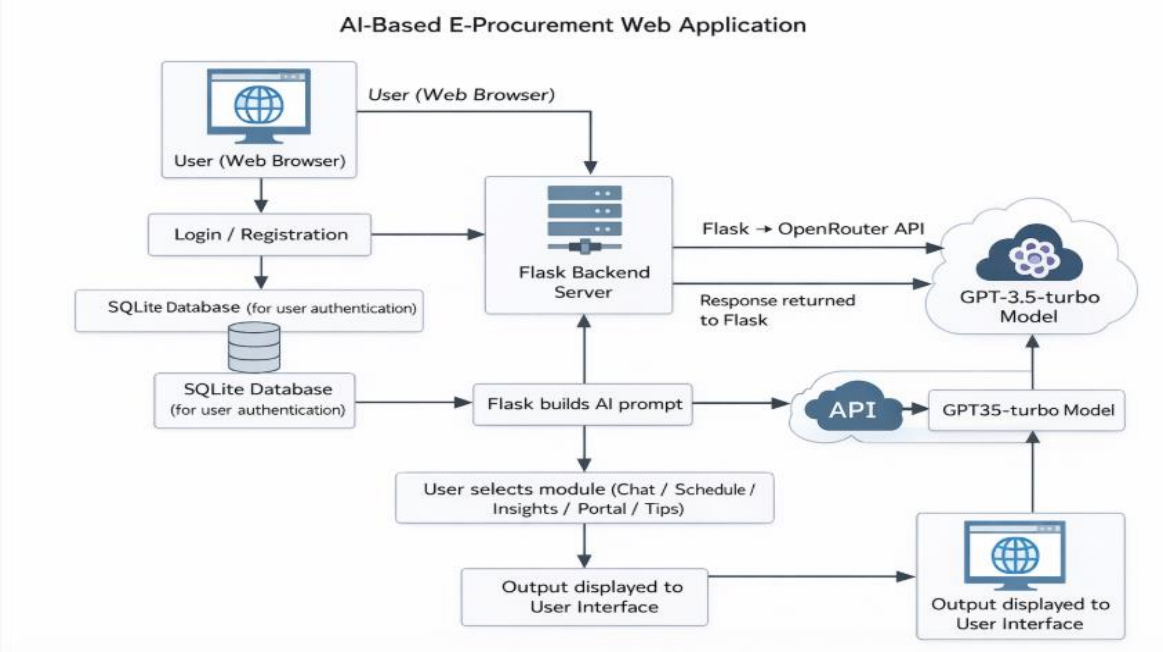

Figure 2

The Open Router API interacts with the GPT-3.5-turbo language model, which generates a context-aware response. Large Language Models can understand structured prompts and generating domain-specific outputs dynamically [2]. The generated response is returned to the Flask server.

The backend may perform additional processing such as formatting the output into bullet points or numbered lists. Finally, the processed result is rendered on the HTML template and displayed to the user in the browser.

The user can continue interacting with different modules during the session. When the user logs out, the session is terminated, ensuring secure closure of the workflow.

5.2 Data Flow

The simplified data flow of the system can be summarized as:

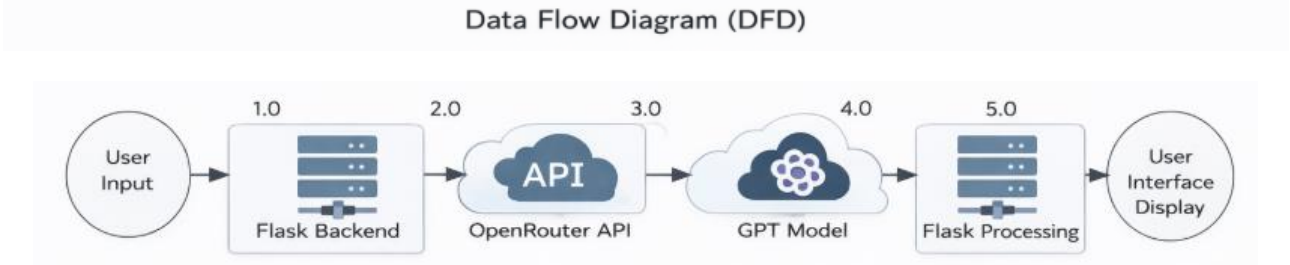

Figure 3

This structured workflow ensures secure communication, modular processing, and real-time AI-generated outputs within the e-procurement environment.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

VI. IMPLEMENTATION AND RESULTS

6.1 TECHNOLOGY STACK

The proposed system is implemented using a lightweight and efficient technology stack suitable for web-based AI applications. The backend is developed using **Python 3** with the **Flask microframework**, which handles routing, session management, and server-side processing. Flask was chosen due to its simplicity, flexibility, and ease of integration with external APIs.

The database used is **SQLite**, a lightweight relational database system that stores user credentials such as username, email, and hashed passwords. SQLite is suitable for small to medium-scale applications and does not require a separate server setup.

For AI integration, the system uses the **Open Router API**, which provides access to the **GPT-3.5-turbo** language model. The backend communicates with the API using the Python requests library to send prompts and receive AI-generated responses dynamically.

The frontend is developed using **HTML5, CSS3, and Bootstrap**, ensuring a responsive and user-friendly interface across devices. Basic JavaScript is used for form validation and interactive elements such as password strength indicators.
Email notifications are implemented using Python's Smtplib library to send login alerts through a configured SMTP server.

This technology stack ensures simplicity, modularity, and efficient performance while supporting AI-driven procurement assistance.

6.2 USER INTERFACE

The user interface

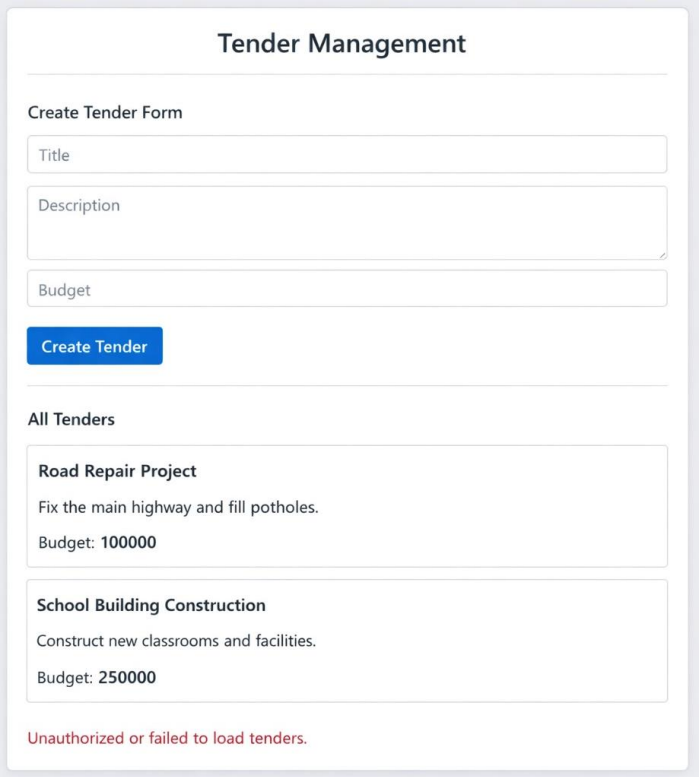

Figure 4

6.3 SYSTEM TESTING AND SAMPLE OUTPUTS

The system was tested using various sample tender scenarios to evaluate functionality, usability, and AI response accuracy. Each module was tested independently to ensure proper data flow and integration between frontend, backend, and AI services.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

For the **AI Chat module**, procurement-related queries such as "What is EMD in government tenders?" were submitted. The system generated accurate and structured explanations describing Earnest Money Deposit, its purpose, and general percentage ranges. The responses were context-aware and relevant to public procurement terminology.

The **Tender Schedule Generator module** was tested using inputs such as "Road Construction Project" with defined start and end dates. The system successfully generated a structured timeline including NIT release, document preparation, pre-bid meetings, technical evaluation, financial bid opening, and final submission stages. The output was displayed in numbered format for clarity.

For the **Tender Insights module**, categories such as "IT Hardware Supply" were entered. The system produced insights covering technical specifications, OEM authorization, warranty clauses, eligibility conditions, and financial compliance requirements. This demonstrated the AI's ability to transform general categories into actionable procurement information.

The **Portal Suggestions module** was tested with inputs specifying region and tender type. The system generated relevant portal recommendations and provided brief guidance for accessing those platforms. The **Compliance Tips module** produced concise advisory tips such as ensuring valid Digital Signature Certificates (DSC) and verifying document uploads before submission.

Overall, testing confirmed that all modules function correctly, generate relevant AI-based responses, and maintain smooth user interaction.

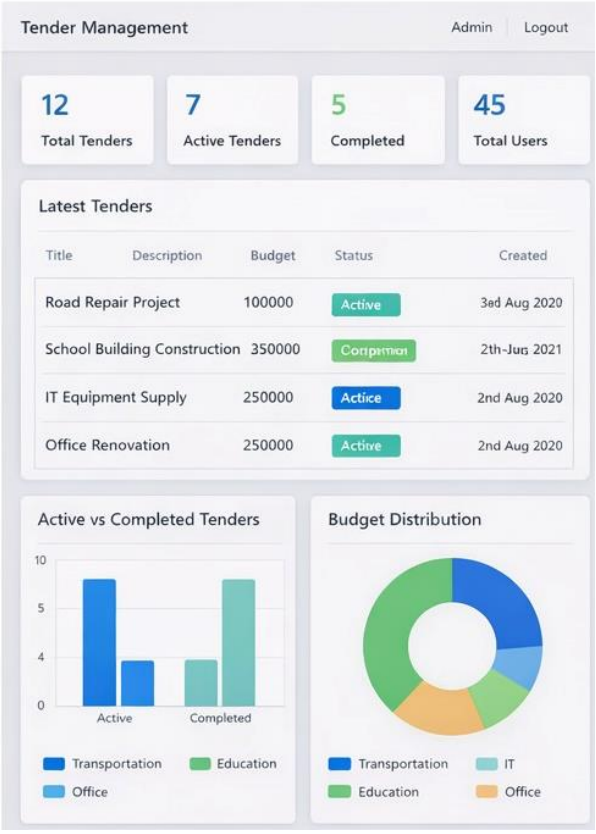

Figure 5

6.4 PERFORMANCE EVALUATION

Performance evaluation focused on response time, system reliability, and database efficiency. The average AI response time ranged between **2 to 4 seconds**, depending on network conditions and API latency. This response time is acceptable for real-time interactive web applications. Database operations using SQLite were executed instantly due to the small-scale architecture and optimized query handling. User authentication, session creation, and email notifications were successfully tested without system errors.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13336

The modular architecture ensured stable communication between system components. Memory usage remained minimal, and the Flask backend handled concurrent user requests effectively in a controlled testing environment. Overall, the system demonstrated reliable performance, stable AI integration, and efficient handling of user interactions.

6.5 DISCUSSION

The implementation results indicate that integrating a large language model into an e-procurement platform significantly enhances usability and decision-support capabilities. Unlike traditional tender portals that primarily provide static document listings, the proposed system delivers dynamic and personalized assistance through conversational AI and automated schedule generation. The effectiveness of large language models in generating structured and context-aware responses has been demonstrated in prior research, highlighting their applicability in decision-support environments [3]. The modular system design improves scalability and future extensibility. AI-generated outputs help reduce manual interpretation effort, minimize compliance errors, and streamline tender preparation processes. Decision-support mechanisms in procurement systems have been shown to improve evaluation clarity and operational efficiency [8]. Similarly, the proposed system extends such support to vendors by offering intelligent insights and planning assistance. The platform also enhances accessibility by simplifying complex procurement terminology, making government tender processes easier to understand for users with varying levels of expertise.

However, certain limitations exist. The system currently relies on externally generated AI responses and does not directly integrate live government tender databases. Additionally, AI-generated content may occasionally require user verification to ensure factual accuracy and compliance with official tender documents. Despite these limitations, the prototype demonstrates the practical feasibility of AI-driven procurement assistance within a web-based architecture.

Overall, the results confirm that combining modular web-based systems with AI-powered advisory mechanisms can improve efficiency, transparency, and user experience in government tender management.

VII. CONCLUSION AND FUTURE SCOPE

This paper presented an Integrated AI-Driven E-Procurement System designed to enhance the efficiency and usability of government tender management. The system combines a Flask-based web architecture with a large language model accessed through the OpenRouter API to provide intelligent procurement assistance. Unlike conventional e-tender platforms that mainly serve as document repositories, the proposed system offers conversational support, automated schedule generation, tender insights, portal recommendations, and compliance guidance within a unified interface.

The implementation results demonstrate that integrating large language models into e-governance platforms can significantly improve decision-support capabilities and user interaction [3]. The modular design ensures scalability and allows future integration of additional procurement-related services. By simplifying complex tender requirements and generating structured outputs dynamically, the system reduces manual effort and enhances transparency in the bidding process. Furthermore, secure user authentication and email notification mechanisms contribute to system reliability and accountability.

However, the current system operates as a prototype and has certain limitations. It does not directly connect to live government procurement databases, and AI-generated responses may require validation against official tender documents. Despite these constraints, the system successfully demonstrates the feasibility of incorporating AI-based advisory mechanisms into public procurement workflows. Previous studies on e-tendering systems have emphasized automation and workflow optimization [1], and this work extends those concepts by integrating conversational intelligence into the procurement lifecycle.

Future Scope

Several enhancements can be implemented to improve the system further:

- **Real-Time API Integration:** Connecting with live government procurement portals to fetch real-time tender listings.
- **Multi-Language Support:** Enabling regional language interaction to improve accessibility for diverse users.
- **Document Parsing and Summarization:** Allowing users to upload tender PDFs for automatic extraction and summarization of key requirements.
- **Mobile Application Development:** Creating Android and iOS applications for improved accessibility.
- **Advanced AI Models:** Integrating more advanced language models to improve contextual understanding and accuracy.

With these enhancements, the proposed system has the potential to evolve into a comprehensive AI-powered procurement assistant capable of transforming government tender management processes.

DOI: 10.17148/IARJSET.2026.13336

REFERENCES

[1]. A. Kayed and R. Colomb, "Business to Business Electronic Commerce: The Electronic Tendering," in *Business-to-Business Electronic Commerce – Challenges & Solutions*, Idea Group Publishing, Hershey, USA, 2002.

[2]. S. Kerridge, C. Halaris, and G. Mentzas, "Virtual Tendering and Bidding in the Construction Sector," in *Proceedings of EC-Web 2000*, Springer, 2000, pp. 379–388.

[3]. T. Brown et al., "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[4]. OpenRouter Documentation, "Unified API for Large Language Models," [Online]. Available: https://openrouter.ai/docs

[5]. Flask Documentation, "Flask – Web Development, One Drop at a Time," [Online]. Available: https://flask.palletsprojects.com/

[6]. SQLite Documentation, "SQLite: Small. Fast. Reliable," [Online]. Available: https://www.sqlite.org/

[7]. Bootstrap Documentation, "The Most Popular HTML, CSS, and JS Library," [Online]. Available: https://getbootstrap.com/

[8]. F. Ahmad, M. Y. M. Saman, N. M. Mohamad Noor, and A. Othman, "DSS for Tendering Process: Integrating Statistical Single-Criteria Model with MCDM Models," in *Proceedings of IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 863–868.

[9]. R. Du, E. Foo, C. Boyd, and B. Fitzgerald, "Defining Security Services for Electronic Tendering," in *Proceedings of the Australasian Information Security Workshop*, 2004, pp. 43–52.

[10]. N. Mohamad Noor, K. N. Papamichail, and B. Warboys, "Process Modelling for Online Communications in Tendering Processes," in *Proceedings of the 29th Euromicro Conference*, 2003, pp. 17–24.

[11]. S. C. Lai, D. K. W. Chiu, and P. C. K. Hung, "e-Tendering with Web Services: A Case Study on the Tendering Process of Building Construction," in *Proceedings of IEEE International Conference on Services Computing (SCC 2007)*, 2007, pp. 582–588.

[12]. F. Ricci, L. Rokach, and B. Shapira, "Recommender Systems: Techniques, Applications, and Challenges," in *Recommender Systems Handbook*, 3rd ed., Springer, 2022, pp. 1–35.

[13]. J. Smith and L. Johnson, "Artificial Intelligence in Public Procurement: Trends, Challenges and Opportunities," *Journal of GovTech Research*, vol. 60, no. 4, pp. 712–728, 2021.

[14]. A. Fayek, "A Competitive Tendering Strategy Model and Software System Based on Fuzzy Set Theory," in *Proceedings of Intelligent Information Systems (IIS 1997)*, 1997, pp. 236–240.

[15]. Y. Zhuang, S. Fong, and M. L. Shi, "Knowledge-Oriented Negotiation for Agent-Based B2B Electronic Commerce," in *Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 2004)*, 2004, pp. 421–424.

