A Low-Code LLM-Based Conversational AI Assistant for ERP Systems: Research, Enhancements, and Implementation Considerations for an MTECH Project

1. Introduction to Conversational AI in ERP Systems

1.1. Overview of Enterprise Resource Planning (ERP) Systems

Enterprise Resource Planning (ERP) systems are foundational software solutions meticulously designed to integrate and manage core business processes across an organization's diverse departments. These systems centralize critical functions such as finance, human resources, supply chain management, manufacturing, and customer relationship management onto a singular, unified platform.[1] The overarching objective of ERP systems is to streamline operations, bolster efficiency, and provide a consistent, centralized view of business data, which is indispensable for informed decision-making.[1] The ERP market itself is a significant sector, projected to reach approximately $52 billion in 2024, underscoring its pervasive and indispensable role in contemporary enterprises.[2]

As business processes continue to escalate in complexity, the capabilities of traditional ERP systems are increasingly found to be insufficient.[2] This limitation extends beyond mere data storage capacity; it pertains to the agility with which data can be utilized and the accessibility required for proactive decision-making and rapid operational responses. The traditional design paradigm of ERP systems, while effective for integration and centralization, struggles to meet the dynamic, real-time, and intuitive interaction demands of modern business environments. This indicates a fundamental shift in the requirements for ERP systems, evolving from being primarily systems of record and process execution to becoming intelligent systems capable of

providing deep insights and proactive assistance. The integration of advanced Artificial Intelligence (AI), particularly conversational AI, is therefore not merely an optional enhancement but a necessary evolutionary step to ensure ERP's continued relevance and value proposition. The proposed MTECH project directly addresses this critical need by introducing an intelligent, accessible layer designed to transform how users interact with and leverage their ERP system.

1.2. The Evolution and Impact of Conversational AI in Enterprise

Conversational AI encompasses a broad spectrum of systems engineered to facilitate human-like interaction through natural language. This category includes widely adopted forms such as AI chatbots, commonly deployed in e-commerce, customer support, and for basic healthcare tasks; voice bots and assistants, prevalent in automotive systems, banking, and smart home devices; and more sophisticated Interactive Voice Assistants (IVAs), which find application in legal services, retail, and complex customer service scenarios.[3] These systems have undergone substantial evolution, transitioning from rudimentary rule-based interactions to highly advanced models that leverage Natural Language Processing (NLP), generative AI, and Large Language Models (LLMs) to achieve remarkably natural and contextually aware conversations.[3]

Within the enterprise context, the integration of AI capabilities into ERP systems is experiencing rapid acceleration. This is driven by the compelling promise of enhanced functionality, significant operational efficiency gains, and an overall boost in business intelligence.[4] The impact of conversational AI in enterprise environments is profound and multifaceted, yielding tangible benefits such as improved predictive analytics, intelligent automation of routine tasks, and the delivery of highly personalized user experiences.[4] Empirical studies provide compelling evidence of these improvements, indicating that businesses adopting AI-driven ERP solutions have observed over a 30% increase in user satisfaction and a 25% boost in productivity, largely attributable to the enhanced personalization of interfaces and streamlined interactions.[4] This collective evidence suggests that conversational AI is far more than a mere interface enhancement; it functions as a fundamental enabler of digital transformation within the ERP domain. By simplifying complex interactions and automating tedious processes, it liberates human capital to focus on more strategic tasks and fosters a culture of data-driven decision-making. This implies that the proposed assistant is not just a convenient tool but a critical component for unlocking new levels of operational

efficiency, user engagement, and data leverage within an ERP environment, positioning the project as a significant and timely contribution to modern enterprise computing.

1.3. Project Context: Low-Code LLM-Based Conversational AI for ERP

The proposed MTECH project aims to develop a conversational AI assistant specifically tailored for ERP systems. This initiative strategically employs a low-code development approach using Oracle APEX and integrates the ChatGPT API for its core conversational intelligence. This deliberate combination directly addresses the escalating demand for rapid application development (RAD) and the increasing sophistication of AI-powered interactions in enterprise environments. Low-code development platforms (LCDPs), exemplified by Oracle APEX, empower "citizen developers" to construct robust software systems with minimal manual coding, thereby significantly accelerating development cycles and reducing associated costs.[6] The integration of advanced Large Language Models (LLMs) like ChatGPT provides unparalleled Natural Language Understanding (NLU) and Natural Language Generation (NLG) capabilities, which are crucial for creating intuitive, human-like conversational interfaces.[3]

The synergy between low-code platforms and LLMs is a powerful one for agile ERP innovation. Low-code platforms are consistently highlighted for their capacity to accelerate development [6] and facilitate "rapid prototyping and iteration".[7] Concurrently, LLMs are demonstrated to provide advanced conversational capabilities, automate complex tasks, and simplify processes.[8] The explicit leverage of both technologies in this project points to a potent synergistic relationship: low-code provides the agility and speed necessary for developing enterprise solutions, while LLMs inject the intelligence and natural interaction required for modern user experiences. This combination signifies a crucial shift from traditional, often lengthy and resource-intensive ERP customization cycles [10] to a more agile, iterative development paradigm for AI-driven features. For the MTECH project, this implies that the student can not only demonstrate technical proficiency in integrating complex AI models but also showcase a keen understanding of modern software development methodologies that prioritize speed, flexibility, and direct business user involvement. The project can effectively serve as a proof-of-concept for how specialized, intelligent extensions can be rapidly built, deployed, and refined within existing enterprise

resource planning frameworks.

2. Foundations of Conversational AI and Large Language Models (LLMs)

2.1. Understanding Conversational AI: Types, Architecture, and Core Components

Conversational AI encompasses a diverse spectrum of systems designed to facilitate human-like interaction through natural language. These systems are broadly categorized into AI Chatbots, commonly employed for e-commerce, customer support, and basic healthcare tasks; Voice Bots/Assistants, which are prevalent in automotive systems, banking, and smart home devices; and more sophisticated Interactive Voice Assistants (IVAs), applied in legal services, retail, and complex customer service scenarios.[3] Modern conversational AI systems have evolved beyond rudimentary rule-based interactions, increasingly relying on advanced technologies such as Natural Language Processing (NLP), generative AI, and Large Language Models (LLMs) to achieve highly natural and contextually relevant conversation flows.[3]

The core architectural components of an ERP AI Chatbot are typically structured to ensure seamless interaction and data processing. These generally include a User Interaction layer, which enables natural language communication; a Bot Connector, responsible for linking user queries to the ERP system; the central Bot Logic, which processes requests and determines appropriate responses; robust Data Handling mechanisms; and a comprehensive Knowledge Base.[1] Key underlying technologies that power these systems comprise Machine Learning (ML) for continuous improvement and adaptation, Natural Language Processing (NLP) for understanding and generating human language, robust Integration Technologies for connecting to diverse ERP data sources, and sophisticated Dialog Management systems for maintaining conversational context across multiple turns.[1]

The collective understanding of these components underscores that a truly effective ERP conversational AI assistant must extend far beyond simple, one-off question-and-answer interactions. It necessitates sophisticated context retention, the

ability to comprehend nuanced and multi-turn requests, and seamless, real-time integration with the ERP's underlying data and complex business processes. The success of such a project will therefore depend not merely on establishing a connection to ChatGPT but on how effectively it manages conversational state, interprets complex user intent, and accurately translates natural language requests into actionable ERP commands. This will require meticulous design of the "Bot Logic" and "Dialog Management" components to ensure a truly intelligent and adaptive user experience.

The following table provides a comparison of various conversational AI types, highlighting their features and limitations in the context of ERP integration:

Type	Highlight Feature	Limitations	Suitability for ERP Integration
:---	:---	:---	:---
AI Chatbots	Easily adaptable; Human-like conversations via NLP, GenAI, LLMs [3]	Doesn't have voice activation features [3]	High; Ideal for text-based data entry, report generation, and automated queries across various ERP modules [1]
Voice Bots/Assistants	Convenient for multitasking [3]	Depends heavily on accurate speech recognition [3]	Moderate to High; Useful for hands-free operations in manufacturing or inventory, but requires robust speech-to-text integration [1]
Interactive Voice Assistants (IVAs)	Integrations to contact center systems; Complex customer service [3]	Highest development cost [3]	Moderate; Best for complex, multi-step workflows requiring deep integration with contact centers or specialized ERP functions, but cost-intensive [3]

Table 1: Comparison of Conversational AI Types for ERP

2.2. The Transformative Role and Benefits of Large Language Models (LLMs)

Large Language Models (LLMs) represent advanced deep learning models, typically built upon transformer architectures, trained on extensive datasets of text and code. This comprehensive training enables them to understand context, retain long-term dependencies within conversations, and proficiently manage complex sentence structures.[9] LLMs are fundamentally revolutionizing the interaction between humans and Artificial Intelligence by imitating natural, human-like language and providing thoughtful, intelligent responses that extend far beyond pre-programmed scripts.[8]

The integration of LLMs into business operations yields extensive and impactful benefits. These include a significantly improved customer experience through 24/7 natural language support, automated content creation (e.g., blog posts, product descriptions, ad copy), sophisticated data analysis capabilities (such as summarizing large documents and extracting executive summaries), cost-efficient operations achieved by automating repetitive and time-consuming tasks, and simplified processes through advanced prompt engineering.[9] For enterprise applications, LLMs are particularly well-suited due to their advanced reasoning capabilities, contextual understanding, and inherent ability to integrate seamlessly with existing workflows via robust APIs.[9]

The core value proposition of LLMs in this context is their ability to serve as an intelligent processing layer that can interpret complex user intent expressed in natural language and translate it into specific, actionable commands within the ERP system. This effectively positions the LLM as the "brain" of the conversational assistant. This capability elevates the system beyond simple data retrieval to enabling proactive insights, sophisticated data analysis, and true task automation, fundamentally transforming how users interact with and leverage ERP data and functionalities. The MTECH project should therefore emphasize how ChatGPT's advanced capabilities will enable this intelligent automation, rather than merely focusing on basic conversational interaction.

The following table summarizes the key benefits that LLMs bring to enterprise conversational AI:

Benefit Category	Specific Benefit	Detailed Explanation/Impact in ERP Context
User Experience	Better User Experience	LLMs enable human-like language, providing intelligent responses without technical jargon, simplifying navigation of complex ERP software.[1]
	Personalized Analysis Tools	LLMs allow customization of data summaries and enable users to ask specific questions about conversational data, leading to tailored insights.[8]
	Improved Customer Understanding	Facilitates analysis of aggregated conversational data, informing strategic decisions and reducing human analysis errors.[8]
Automation	Accelerated Handling of Repetitive Tasks	Automates routine tasks like data entry, report generation, and approval workflows, freeing human teams for strategic work.[1]
	Automated Content Creation	Generates business content (e.g., product descriptions, ad copy), saving time and resources.[9]
Data Analysis	Smart Data Analysis	Examines unstructured data, summarizes large documents, and provides executive summaries for faster decision-making.[9]
	Smarter Decision-Making with Real-Time Insights	Provides instant access to live ERP data, enabling informed decisions without sifting through complex reports.[11]

Cost Efficiency	Deliver Cost and Time Savings	Automates manual tasks like Quality Assurance, processing vast amounts of data quickly, leading to significant cost and time savings.[8]
Adaptability	Modify Tools Quickly	LLMs can be fine-tuned to adapt to changing trends, process new information, and perform new tasks, ensuring systems remain cutting-edge.[8]

Table 2: Key Benefits of LLMs in Enterprise Conversational AI

2.3. LLMs in Enterprise Applications: Enhancing User Experience and Automation

Large Language Models significantly enhance the overall user experience within enterprise applications by simplifying the navigation of complex software through an intuitive, conversational interface.[1] They are capable of providing highly personalized support, remembering user preferences, and offering contextually relevant responses, which leads to smoother and more efficient operations.[11] This improved interaction also facilitates a deeper understanding for end-users by enabling the analysis of aggregated conversational data. This analysis, in turn, informs strategic decisions related to training, branding, and customer satisfaction, while simultaneously reducing human analysis errors.[8]

Beyond enhancing user experience, LLMs are powerful drivers of automation. They can automate repetitive and time-consuming tasks such as report generation, data entry, and approval workflows, thereby boosting overall productivity and significantly reducing operational costs.[1] Furthermore, LLMs enable smarter decision-making by providing real-time insights derived directly from ERP data.[11] Their adaptability is another key advantage, as they can be quickly modified or fine-tuned to adapt to changing trends, process new information, and perform new tasks, ensuring the system remains cutting-edge.[8]

Traditional ERP interaction often requires users to actively search for data, manually generate reports, or explicitly initiate processes. However, the capabilities of LLMs enable "proactive alerts and notifications" [11] and allow the system to "anticipate issues

or opportunities".[11] They also facilitate "smarter decision-making with real-time insights" [11], indicating a move beyond retrospective analysis. This capability represents a fundamental shift in the ERP user experience from a reactive model of data retrieval and manual task execution to a proactive, intelligent assistance paradigm. The conversational AI assistant, powered by LLMs, can evolve into a vigilant operational partner that not only responds to direct queries but also anticipates needs, flags anomalies, identifies trends, and suggests optimal actions before issues escalate. This significantly increases the strategic value and utility of the ERP system. The MTECH project should aim to incorporate such proactive features where technically feasible, demonstrating a forward-looking approach to enterprise AI.

3. Low-Code Development with Oracle APEX

3.1. Introduction to Low-Code Development Platforms (LCDPs) and their Advantages

Low-code development platforms (LCDPs) represent a transformative shift in software development, offering intuitive visual environments that drastically reduce the manual coding required to build applications.[6] These platforms empower a new category of "citizen developers"—individuals with minimal or no traditional programming background—to rapidly create and deploy fully functional software applications.[6] LCDPs achieve this acceleration through user-friendly graphical interfaces, drag-and-drop functionalities, visual workflow designers, and the provision of pre-built components and connectors.[6]

The advantages of low-code platforms are multifaceted and highly beneficial for enterprise contexts. They include a significant reduction in development time, with platforms like Power Automate demonstrating an approximately 87% reduction compared to traditional development approaches.[6] Furthermore, LCDPs lead to a lower total cost of ownership, accelerate digital transformation initiatives, and increase the accessibility of technology for organizations of all sizes, from large enterprises to Small and Medium-sized Enterprises (SMEs).[6] These platforms also facilitate the reuse of existing artifacts and ensure that developed services align

closely with evolving business requirements.[6]

The strategic choice of a low-code platform like Oracle APEX is not merely a convenience but a significant advantage for developing and iteratively refining a complex AI-driven ERP assistant within a defined project timeline. Low-code platforms are consistently lauded for their ability to "dramatically reduce the number of lines of code" and "accelerate the development process" [6], enabling "rapid prototyping and iteration of integration solutions".[7] This stands in stark contrast to the traditional complexity and resource-intensiveness associated with ERP customization.[10] By leveraging low-code, the student can allocate more focus and resources to the intricate aspects of AI logic, data integration, and conversational design, rather than being encumbered by extensive boilerplate code. This also implies that low-code platforms are effectively democratizing access to complex enterprise AI development, making such ambitious projects feasible for individual students or smaller development teams.

3.2. Oracle APEX: AI Capabilities for Application Development

Oracle APEX (Application Express) is a robust, web-based development platform that facilitates the creation of scalable web applications directly on the Oracle Database.[12] It offers a rich suite of built-in AI capabilities, empowering developers to build intelligent, context-aware applications with relative ease.[13]

Key AI features and functionalities available within Oracle APEX that are particularly relevant for conversational AI development include:

- **AI-Assisted Development (GenDev):** The APEX AI Assistant introduces new capabilities for Generative Development. It assists in generating application blueprints from natural language descriptions of desired pages, data, and features. Moreover, it aids in code authoring for SQL queries, HTML, CSS, JavaScript, and PL/SQL, and helps in debugging across APEX code editors.[13] This significantly simplifies the development process and enhances developer productivity.[13]
- **AI-Powered Apps:** APEX provides direct support for configuring AI-powered conversational experiences through "Conversational AI Dialogs." Developers can easily define system prompts, welcome messages, and customize the appearance of the AI assistant. Crucially, it enables the definition of "AI Prompts and RAG Data

Sources," where responses can be dynamically enhanced by sending additional context data derived from SQL queries, function outputs, or static text. These Retrieval-Augmented Generation (RAG) sources are re-evaluated with each user prompt, ensuring responses remain highly relevant and up-to-date.[13]

- **Vector Search Support:** APEX simplifies the integration of Oracle Database 23ai's vector search capabilities into applications. This empowers end-users to find the most relevant results through semantic similarity searching, enhancing the ability to retrieve information from unstructured data within the ERP context.[13]
- **Generative AI Services Integration:** APEX offers native integrations with leading Generative AI services, including OCI Generative AI, OpenAI (which encompasses ChatGPT), and Cohere. Developers also have the flexibility to connect to other Generative AI services via standard REST APIs within their APEX applications.[13] Additionally, APEX supports ONNX models loaded into Oracle Database 23ai for performing vector embeddings directly within the database.[13]
- **Select AI:** This feature, available in Autonomous Database, leverages generative AI and Large Language Models to translate natural language user input directly into Oracle SQL queries, further simplifying data interaction.[13]

The capabilities of APEX extend far beyond basic visual development. It explicitly includes features like GenDev, AI-powered apps with RAG support, vector search, and native integrations with major LLM providers like OpenAI.[13] This comprehensive suite indicates that APEX is not merely a generic low-code platform but a mature, purpose-built environment explicitly designed to facilitate and accelerate AI integration, particularly for enterprise applications. For the MTECH project, this means Oracle APEX provides a robust, pre-built infrastructure that significantly simplifies the integration of ChatGPT and the management of ERP data. The student will not need to build complex RAG pipelines or implement vector search functionalities from scratch, as APEX offers native or highly streamlined support for these. This dramatically reduces the technical overhead and allows for the development of a more sophisticated and feature-rich AI assistant than might be possible with a purely custom-coded approach, enabling the student to focus on the ERP-specific business logic and the nuances of conversational interaction.

The following table details Oracle APEX's AI capabilities relevant for conversational AI development:

Capability Category	Specific Feature	Brief Description	Direct Relevance to Conversational

			AI/ERP Assistant Development
AI-Assisted Development (GenDev)	Create Apps and Model Data using Natural Language	Generates application blueprints from natural language descriptions of desired pages, data, and features.[13]	Accelerates initial setup and scaffolding for the conversational assistant's interface and data models within APEX.
	Code Authoring and Debugging	Assists with SQL, HTML, CSS, JavaScript, and PL/SQL code generation, optimization, explanation, and debugging.[13]	Streamlines the development of backend logic and UI components that interact with the LLM and ERP data.
AI-Powered Apps	Conversational AI Dialogs	Easily configures AI-powered conversational experiences with customizable prompts and appearance.[13]	Provides a native, low-code way to embed the chatbot interface directly into APEX pages.
	AI Prompts and RAG Data Sources	Defines AI Assistant prompts and enhances responses with context from SQL queries, function output, or static text, re-evaluated per prompt.[13]	Crucial for grounding LLM responses in real-time, specific ERP data, preventing hallucinations and ensuring accuracy.
	Vector Search Support	Simplifies adding Oracle Database 23ai vector search for semantic similarity searching.[13]	Enables efficient retrieval of relevant information from unstructured ERP documents or knowledge bases to augment LLM responses.

Generative AI Integration	Native Integrations	Offers direct connections to OCI Generative AI, OpenAI (ChatGPT), and Cohere, plus REST API support for others.[13]	Simplifies the technical connection to the ChatGPT API, reducing integration effort.
	ONNX Model Support	Supports ONNX models loaded into Oracle Database 23ai for vector embeddings.[13]	Provides flexibility for advanced AI features, such as custom embeddings for specialized ERP data.
Data Interaction	Select AI	Translates natural language user input directly into Oracle SQL queries.[13]	Allows users to query ERP data using natural language, making data access more intuitive.

Table 3: Oracle APEX AI Capabilities for Conversational AI Development

3.3. Practical Integration of ChatGPT API with Oracle APEX

Integrating the ChatGPT API into an Oracle APEX application is a practical and well-supported process, leveraging APEX's robust capabilities for consuming external REST APIs.[12]

The general steps involved in this integration are as follows:

1. **Obtaining an OpenAI API Key:** The foundational step requires creating an OpenAI account and generating a unique API key. This key serves as a digital passport, authenticating access to the ChatGPT API for use within applications.[16]
2. **Understanding OpenAI API Structure and Response:** Developers should familiarize themselves with the OpenAI Playground to experiment with API parameters such as model, prompt, temperature, and max_tokens. This understanding is crucial for crafting effective requests that elicit desired responses from the LLM.[17]
3. **Configuring REST Data Sources in APEX:** Oracle APEX's "REST Data Sources" are shared components specifically designed for integrating REST services or

generic JSON data feeds. This involves setting up Web Credentials to store API keys for authentication (e.g., using the Bearer <your API key> scheme) and defining the appropriate ChatGPT API endpoint (e.g., https://api.openai.com/v1/completions).[16]

4. **Making API Calls (PL/SQL or Dynamic Actions):** Within the APEX application, the ChatGPT API can be invoked. This is commonly achieved through custom PL/SQL packages that make POST requests to the OpenAI endpoint, passing the user's natural language prompt and receiving the AI's generated response.[17] Alternatively, APEX offers "Conversational AI Dialogs" as a dedicated Dynamic Action, simplifying the configuration of AI-powered conversational experiences.[13]
5. **Handling and Displaying Responses:** The AI's response, typically in JSON format, is returned to the APEX application. This response can then be parsed and displayed to the user within the application's interface.[15]

While the technical integration process via REST APIs is relatively straightforward within Oracle APEX, the *effectiveness* and *utility* of the conversational AI assistant in an ERP context hinge critically on sophisticated prompt engineering and the ability to feed relevant, real-time ERP data as context to ChatGPT (a process known as Retrieval-Augmented Generation - RAG). The importance of a well-crafted prompt cannot be overstated, as it sets the stage and provides context for the AI; an ambiguous or poorly defined prompt can lead to unclear or irrelevant outputs.[17] Furthermore, APEX's "AI Prompts and RAG Data Sources" feature is designed to enhance responses by sending additional context data from SQL queries, function outputs, or static text.[13] Therefore, the MTECH project should dedicate significant effort to designing prompts that elicit accurate, precise, and useful responses from ChatGPT based on specific ERP queries. Concurrently, implementing robust RAG mechanisms within APEX is paramount to ensure that ChatGPT's responses are grounded in the most current and relevant ERP data, transforming the LLM into an "expert" on the specific ERP instance rather than a general knowledge model. This emphasis moves the project beyond a mere API call to a truly intelligent and context-aware system.

4. Analysis of Existing ERP AI Chatbot Systems

4.1. Current Landscape and Capabilities of AI-Driven ERP Solutions

The current enterprise landscape is characterized by a significant and accelerating trend towards integrating Artificial Intelligence into ERP systems. Virtually all major ERP vendors now assert that they have integrated AI capabilities into their offerings.[2] These intelligent ERP solutions are designed to analyze vast amounts of complex data, optimize intricate business processes, predict potential issues before they arise, and ultimately enhance strategic decision-making across the organization.[5] AI-powered features are rapidly becoming standard, extending beyond basic monitoring functionalities to include more advanced, and even autonomous, decision-making capabilities in critical areas such as production scheduling, quality control, and supply chain optimization.[5]

Existing ERP AI chatbots offer a robust set of core capabilities. These include intuitive user interaction through natural language (text or voice), automation of routine tasks (e.g., data entry, report generation, system updates), effective issue resolution, generation of data-driven insights, personalized user interaction based on preferences, enhanced accessibility and overall user experience, and continuous learning and adaptation over time.[1] Furthermore, these chatbots provide round-the-clock availability, offer real-time data access, and can proactively send alerts and notifications, anticipating issues or opportunities.[11]

The current trend indicates a profound maturation in the application of AI within ERP, moving from merely automating manual, rule-based processes to actively supporting or even making complex, high-stakes operational decisions. For instance, AI-powered ERP systems can monitor for unusual activity, such as potential fraud, and proactively alert compliance teams.[2] This demonstrates a significant leap beyond the more commonly cited benefits of simply automating repetitive tasks like data entry or report generation.[1] For the MTECH project, this implies that the proposed assistant should aspire to more than just answering questions or executing simple commands; it should aim to provide actionable recommendations, generate predictive insights, or even trigger automated processes based on real-time ERP data analysis. This demonstrates a higher level of intelligence and utility, positioning the project at the forefront of AI-driven enterprise solutions.

4.2. Common Use Cases and Functional Areas

ERP AI chatbots are proving transformative across a wide array of business operations and functional areas. Their common use cases span various departments, demonstrating the broad applicability of this technology:

- **Customer Support:** These chatbots handle diverse customer questions, provide real-time order status updates, retrieve payment information, and offer detailed product data.[1]
- **Internal Process Automation:** They automate mundane yet critical tasks such as data entry, report generation, inventory management, and approval workflows.[1] This automation significantly frees up human teams to concentrate on higher-priority, strategic activities like problem-solving and innovation.[11]
- **Financial Management:** AI chatbots enhance financial management practices, assist in fraud detection by analyzing large datasets, and automate tasks like invoice processing and reconciliation.[1]
- **Supply Chain & Inventory Management:** They streamline inventory-related processes by automating inquiries, generating reorder requests, and tracking shipments in real-time.[1]
- **HR Management:** Conversational AI facilitates seamless employee onboarding processes and optimizes various HR management activities.[1]
- **Sales & Marketing:** These systems augment sales and marketing efforts by qualifying leads, providing personalized product recommendations, and facilitating sales transactions, thereby accelerating the sales cycle.[1]
- **Analytics:** Chatbots summarize complex unstructured data, enable natural language queries for quick insights (e.g., "What was our top-performing blog last month?"), and automate the creation of visual dashboards and performance reports.[9]

The research consistently lists diverse applications of conversational AI across nearly every major ERP function: finance, HR, supply chain, sales, customer support, and inventory.[1] This extensive list demonstrates that the utility and benefits of conversational AI are not confined to a single departmental silo but are broadly applicable across the integrated ERP system, enhancing various business functions. For the MTECH project, this broad applicability suggests that while the chosen initial scope for the assistant should be carefully defined (e.g., focusing on a specific, high-impact functional area like inventory inquiries or basic HR requests), the underlying architectural design should inherently consider the potential for future expansion and integration with other ERP domains. This approach allows for demonstrating not only the successful implementation of a focused solution but also

a comprehensive understanding of the system's broader potential and scalability within an enterprise context.

4.3. Identified Gaps and Opportunities for Enhancement

While existing AI-driven ERP systems offer significant benefits, their integration and long-term effectiveness are not without challenges. These include pervasive data quality issues, which can undermine the reliability of AI outputs, and inherent resistance to change within organizational culture, which can impede adoption.[4] Furthermore, LLM-driven systems introduce their own set of pitfalls, such as the initial lack of clear objectives, insufficient data preparation, misapplication of LLMs to unsuitable use cases, neglecting scalability requirements, underestimating the complexities of integration with legacy systems, overlooking the critical need for team training, neglecting ongoing monitoring of model performance, and failing to proactively address inherent model biases.[19]

Based on these challenges and the capabilities of modern LLMs, several key opportunities for enhancement emerge for a low-code LLM-based ERP assistant:

- **Deeper Contextual Reasoning:** While current chatbots can follow conversation context to some extent [11], LLMs can still struggle significantly with multi-turn, underspecified conversations, often making premature assumptions and failing to recover from "wrong turns".[20] Enhancing the system's ability to maintain and leverage deep, persistent conversational context is crucial for truly human-like interactions.
- **Robust Human-in-the-Loop (HITL) Mechanisms:** Integrating HITL interactions is essential to ensure precise human control and oversight, particularly for sensitive operations, complex decision-making, or scenarios where the LLM's confidence is low.[21] This balance between automation and human intervention is critical for enterprise-grade reliability.
- **Comprehensive Data Governance:** Proactively addressing data quality, consistency, and security issues throughout the integration and operational lifecycle is paramount.[22] Poorly structured or biased datasets can drastically undermine the effectiveness of an LLM agent.[19]
- **Proactive Intelligence and Anomaly Detection:** Moving beyond reactive responses to anticipatory alerts, predictive insights, and automated anomaly detection based on real-time ERP data analysis.[2] This transforms the assistant

from a query responder to a vigilant operational partner.

- **Optimized Scalability and Performance:** Designing the system with scalability in mind to handle increasing data loads and concurrent user interactions efficiently, while mitigating API limitations such as rate limits and feature restrictions.[19]
- **Ethical AI and Bias Mitigation:** Implementing proactive strategies for identifying, testing, and mitigating biases in LLM outputs to ensure fairness, prevent misinformation, and maintain user trust.[19] Failing to address bias can lead to ethical and reputational risks, as LLMs might generate polarized content or reproduce stereotypes.[23] This requires continuous monitoring and potentially retraining models with more representative data.[19]

The consistent highlighting of LLM limitations, particularly concerning their performance in complex multi-turn conversations [20] and their susceptibility to biases [19], underscores the criticality of addressing these for enterprise-grade reliability. While LLMs offer immense potential, their deployment in sensitive ERP environments demands careful consideration of these challenges. For an MTECH project, this implies that a successful implementation will not merely demonstrate basic functionality but will also incorporate mechanisms and design principles to mitigate these known limitations, ensuring the assistant is robust, trustworthy, and truly valuable in a real-world enterprise setting. This includes robust output filtering and sanitization, strong data handling and privacy controls, and a "red teaming" approach to test for vulnerabilities.[23]

5. Technical Challenges and Security Considerations

5.1. Data Integration Complexities

Integrating data into ERP systems, especially when incorporating new technologies like LLMs, presents several layers of complexity. Enterprise data ecosystems are inherently intricate due to the variety of data formats and processes they employ.

One significant challenge lies in ensuring the accuracy, completeness, and

consistency of integrated data, which directly impacts the reliability of the ERP system's outputs. This necessitates the establishment of robust data governance policies that define data standards, cleanliness, and maintenance procedures. The use of data quality tools to cleanse and deduplicate data prior to integration is also crucial.[22] Mapping data fields from various sources to the ERP system can be complex and time-consuming, requiring a deep understanding of both source and target data structures. Advanced ETL (Extract, Transform, Load) tools with graphical interfaces can simplify this process, and engaging data experts early in the project is advisable to define clear mapping rules.[22]

Technical hurdles in integration are also prominent. Many APIs, including those for LLMs, impose rate limits on the number of requests allowed within a specific timeframe, which can significantly slow down data synchronization, particularly with large datasets. Furthermore, not all ERP functionalities may be exposed via APIs, meaning complex processes might not be fully supported, necessitating workarounds or manual inputs. Continuous monitoring and adaptation to API updates or deprecations are also required to ensure uninterrupted operations.[22] Scalability and performance bottlenecks pose another challenge. As data volume increases, both the ERP system and the integration infrastructure are susceptible to overload, leading to slower response times and decreased user satisfaction. A high number of concurrent users and processes can exacerbate these issues, causing delays and impacting critical business operations. Insufficient computing resources allocated to the ERP system or integration components can hinder performance, often requiring optimization of the ERP system's architecture, database, or integration logic.[22]

5.2. Security Risks and Best Practices for LLM-Based Systems

The integration of LLMs into ERP systems introduces a unique set of security risks that must be meticulously addressed. The Open Web Application Security Project (OWASP) has identified a "Top 10" list of security risks specifically for LLM applications. These include:

- **Prompt Injection (LLM01):** Attackers can insert malicious input into the model's prompt to manipulate its behavior, leading to harmful or inappropriate content generation.[23]
- **Insecure Output Handling (LLM02):** Without proper controls, LLM-generated content can lead to web application attacks (e.g., XSS, CSRF, remote code

execution) if processed by a receiving application.[23]
- **Training Data Poisoning (LLM03):** Manipulating the model's training data can bias its output or functionality, introducing misleading or false information.[23]
- **Model Denial of Service (DoS) (LLM04):** Overwhelming the model with a high volume of requests can cause it to slow down or crash, rendering the application unavailable.[23]
- **Supply Chain Vulnerabilities (LLM05):** Risks can arise from insecure components and dependencies within the LLM application's supply chain.[23]
- **Sensitive Information Disclosure (LLM06):** The LLM application might inadvertently reveal sensitive data, such as personal or proprietary business information, in its output.[23]
- **Insecure Plugin Design (LLM07):** Vulnerable plugins can be exploited to inject malicious code, alter functionality, or gain unauthorized access.[23]
- **Excessive Agency (LLM08):** Granting LLM systems too much functionality, permissions, or autonomy can lead to unintended or harmful actions in response to ambiguous outputs.[23]
- **Overreliance (LLM09):** Undue dependence on the accuracy and appropriateness of LLM outputs can result in security breaches, misinformation, and reputational damage if content is incorrect.[23]
- **Model Theft (LLM10):** Unauthorized access, copying, or exfiltration of proprietary LLM models can lead to economic losses and compromised competitive advantage.[23]

To mitigate these risks, several security best practices are essential for LLM applications:

- **Sanitize Inputs:** Scrutinize and clean user-provided data to prevent malicious content from influencing LLM responses, including prompt injections and personal data.[23]
- **Data Minimization and Encryption:** Only collect and process data that is absolutely necessary, and apply encryption to both stored and in-transit data to prevent unauthorized access.[23]
- **Implementing Access Control and Auditing:** Determine who has access to the LLM model and what actions they can perform through user roles and permissions. Regularly review model activities to ensure adherence to security measures.[23]
- **Secure Training Data:** Implement strict access controls, encrypt data, and regularly audit data handling processes for training data to ensure reliability.[23]
- **API Security:** Protect Application Programming Interfaces from unauthorized access and misuse through access controls, data encryption, and continuous

monitoring.[23]

- **Federated Learning and Red Teaming:** Employ federated learning for secure and distributed training of LLMs with sensitive datasets, and regularly test systems using "red teams" to simulate attacks and identify weaknesses.[23]
- **Output Risk Scoring and Fine-Grained Prompt Control:** Implement mechanisms to evaluate generated outputs for sensitivity or harmfulness, and design templates or guided input frameworks to limit exposure to harmful prompt engineering.[23]
- **Synthetic Monitoring:** Continuously feed test prompts to probe for signs of malicious behavior or unintended responses, enabling proactive anomaly detection.[23]

These measures are critical to ensure the operational integrity, user trust, and regulatory compliance of an LLM-based ERP conversational AI assistant.[24]

6. Academic Literature Review

6.1. Review of IEEE Papers

The integration of conversational AI and Large Language Models (LLMs) within Enterprise Resource Planning (ERP) systems, particularly through low-code platforms, is a burgeoning area of research. An IEEE survey on LLM-based AI Chatbots highlights the significant advancements driven by the advent of OpenAI's ChatGPT, which has set new standards in the AI community.[25] This survey provides a comprehensive overview of the evolution and deployment of LLM-based chatbots across various sectors, recognizing them as tools for generating new knowledge and exploring their diverse applications.[25] It also discusses open challenges related to training data and potential misuse of generated knowledge, offering a future outlook to enhance efficiency and reliability.[25]

Another relevant IEEE paper focuses on low-code development platforms (LCDPs), describing them as user-friendly visual environments that enable "citizen developers" to build software systems without extensive programming knowledge.[6] This research

presents a technical survey and a conceptual comparative framework for different LCDPs, aiming to facilitate understanding and selection based on user requirements.[6] It emphasizes that low-code tools are widely adopted for basic Business Process Management (BPM) tasks like process modeling and workflow automation, particularly attractive to SMEs due to reduced total cost of ownership and empowerment of business users in digital transformation.[6] The paper notes significant reductions in development time (e.g., 87% for Power Automate) and integration effort (up to 65% for Logic Apps) compared to traditional methods.[6] This underscores the efficiency gains that low-code platforms bring to complex enterprise integrations.

6.2. Review of ACM Papers

The Communications of the ACM (CACM) has also explored the transformative impact of AI and LLMs on software development. One article discusses how AI/LLMs are fundamentally reshaping software construction, with over half of developers already using AI-powered coding tools.[26] These tools offer benefits such as secure code suggestions, vulnerability pinpointing, and generation of code from natural language descriptions, freeing developers for more complex tasks.[26] However, the paper also cautions against overreliance on AI/LLMs, highlighting the dangers of security vulnerabilities if developers lack a fundamental understanding of secure coding practices.[26] It emphasizes the importance of secure code training to recognize errors, identify third-party vulnerabilities, and critically review AI-generated code.[26] This perspective is crucial for any MTECH project integrating LLMs, as it highlights the need for human oversight and a balanced approach.

Another ACM opinion piece questions the assumption that LLMs are the inevitable and best future path for AI, particularly for strategic tasks requiring explainable and auditable reasoning.[27] It advocates for exploring alternatives or complementary approaches to LLMs, such as "compact AI" and active learning, which can achieve optimal results with significantly less computational power and provide transparent reasoning.[27] This discussion encourages a critical perspective on LLM capabilities and limitations, suggesting that for specific ERP tasks requiring high accuracy and auditability, a hybrid approach combining LLMs with more deterministic methods might be beneficial.

6.3. Review of ArXiv Papers

ArXiv pre-prints provide further insights into LLM-based conversational AI and its integration challenges. A survey on evaluating LLM-based agents in multi-turn conversational settings reveals that LLMs exhibit significantly lower performance in multi-turn conversations compared to single-turn interactions, with an average drop of 39% across various generation tasks.[20] This performance degradation is attributed to a minor loss in aptitude and a significant increase in unreliability, often manifesting as LLMs making premature assumptions, attempting to generate final solutions too early, and struggling to recover from "wrong turns" in the conversation.[20] This finding is particularly relevant for designing an ERP assistant that needs to handle complex, multi-step user queries.

Another ArXiv paper presents an AI-driven telephone survey system integrating text-to-speech (TTS), LLMs, and speech-to-text (STT) to mimic human-led interviews at scale.[29] This system successfully administered questions, handled clarifications, and navigated branching logic, demonstrating the potential for scalable, consistent data collection.[29] While the AI system's probing for qualitative depth was more limited than human interviewers, overall data quality for structured items approached human-led standards.[29] This case study offers a practical example of LLM integration for conversational purposes, highlighting both successes and remaining limitations in emulating human conversational nuances.

Finally, a survey on LLMs in Artificial Intelligence for IT Operations (AIOps) highlights their application in optimizing processes and improving outcomes, analyzing 183 research papers.[30] This survey examines diverse failure data sources, the evolution of AIOps tasks, various LLM-based methods to address challenges, and evaluation methodologies.[30] This broader context indicates the widespread academic interest in applying LLMs to complex enterprise IT functions, including those related to ERP system management and optimization.

7. Conclusions and Recommendations for the MTECH Project

The development of a Low-Code LLM-Based Conversational AI Assistant for ERP Systems, leveraging Oracle APEX and the ChatGPT API, represents a highly relevant

and impactful MTECH final year project. The analysis demonstrates that traditional ERP systems are evolving from mere record-keeping and process execution tools to intelligent systems of insight and proactive assistance. Conversational AI, powered by LLMs, is a pivotal enabler of this digital transformation, simplifying complex interactions, automating mundane tasks, and fostering data-driven decision-making.

Oracle APEX provides a robust and comprehensive low-code environment, offering built-in AI capabilities such as Generative Development, AI-powered conversational dialogs with Retrieval-Augmented Generation (RAG) support, and native integrations with leading LLMs like OpenAI. This significantly accelerates development and allows the project to focus on sophisticated AI logic and conversational design rather than extensive custom coding. The practical integration of the ChatGPT API into APEX is well-supported through REST Data Sources, making the technical connection feasible.

However, for the project to achieve enterprise-grade reliability and utility, several critical considerations and enhancements are recommended:

1. **Prioritize Contextual Understanding and Dialog Management:** Given that LLMs can struggle with complex, multi-turn conversations, the project should invest heavily in designing the "Bot Logic" and "Dialog Management" components within APEX. This involves meticulous prompt engineering and leveraging APEX's RAG capabilities to feed precise, real-time ERP data as context to ChatGPT, ensuring accurate and relevant responses grounded in the specific ERP instance.
2. **Focus on High-Impact, Actionable Use Cases:** While the potential applications across ERP functions are vast, the project should initially define a clear, high-impact functional area (e.g., inventory inquiries, basic HR requests, or specific financial data retrieval). The design should, however, consider future scalability and cross-functional expansion. The aim should be to move beyond simple data retrieval to providing actionable recommendations, predictive insights, or triggering automated processes based on ERP data analysis.
3. **Implement Robust Data Governance and Security Measures:** ERP systems handle highly sensitive data. Therefore, the project must incorporate strong data governance policies to ensure data accuracy, completeness, and consistency. Crucially, addressing LLM-specific security risks, such as prompt injection, sensitive information disclosure, and potential biases, is paramount. Best practices like input sanitization, data minimization, encryption, access control, and continuous auditing should be integrated into the system's design and implementation.
4. **Incorporate Human-in-the-Loop (HITL) Mechanisms:** For sensitive operations, complex decisions, or scenarios where the LLM's confidence is low,

integrating HITL interactions will ensure precise human oversight and control. This balances automation with human accountability and builds user trust.

5. **Plan for Scalability and Performance Optimization:** Anticipate increasing data loads and concurrent user interactions. The architecture should be designed to handle these efficiently, considering API rate limits and optimizing data retrieval processes within Oracle APEX.

By meticulously addressing these technical and strategic considerations, the MTECH project can successfully demonstrate a sophisticated, intelligent, and reliable conversational AI assistant for ERP systems. This project will not only showcase technical proficiency in integrating cutting-edge AI with low-code platforms but also contribute significantly to the evolving landscape of enterprise computing by transforming how users interact with and derive value from their ERP investments.

Works cited

1. ERP AI Chatbots: Transform Business Operations & Customer Service, accessed on July 22, 2025, https://www.jellyfishtechnologies.com/erp-ai-chatbots-transform-business-operations/
2. Top 4 ERP AI Use Cases & Case Studies in 2025 - Research AIMultiple, accessed on July 22, 2025, https://research.aimultiple.com/erp-ai/
3. Types of Conversational AI: GenAI Chatbots to IVR in 2025 - Research AIMultiple, accessed on July 22, 2025, https://research.aimultiple.com/types-of-conversational-ai/
4. (PDF) Integration of Artificial Intelligence (AI) in Enterprise Resource Planning (ERP) Systems: Opportunities, Challenges, and Implications - ResearchGate, accessed on July 22, 2025, https://www.researchgate.net/publication/387667312_Integration_of_Artificial_Intelligence_AI_in_Enterprise_Resource_Planning_ERP_Systems_Opportunities_Challenges_and_Implications
5. AI in ERP: The Next Wave of Intelligent ERP Systems for 2025, accessed on July 22, 2025, https://www.top10erp.org/blog/ai-in-erp
6. Supporting the understanding and comparison of low-code development platforms | Request PDF - ResearchGate, accessed on July 22, 2025, https://www.researchgate.net/publication/344842798_Supporting_the_understanding_and_comparison_of_low-code_development_platforms
7. (PDF) Low-Code Integration Platforms: Revolutionizing Enterprise Architecture Through AI-Driven Development - ResearchGate, accessed on July 22, 2025, https://www.researchgate.net/publication/388841003_Low-Code_Integration_Platforms_Revolutionizing_Enterprise_Architecture_Through_AI-Driven_Development
8. Top 6 benefits of integrating LLMs for Conversation Intelligence ..., accessed on

July 22, 2025, https://www.assemblyai.com/blog/top-benefits-integrating-llms-conversation-intelligence-platforms
9. What Are LLMs? Benefits, Use Cases, & Top Models in 2025, accessed on July 22, 2025, https://graffersid.com/what-are-llms-benefits-use-cases-top-models-in-2025/
10. (PDF) Bridging the Gap: Low-Code Platforms and the Future of ERP Customization, accessed on July 22, 2025, https://www.researchgate.net/publication/392399697_Bridging_the_Gap_Low-Code_Platforms_and_the_Future_of_ERP_Customization
11. ERP AI Chatbot Development: Full Guide for 2025 | SapientPro, accessed on July 22, 2025, https://sapient.pro/blog/erp-ai-chatbot-development-guide
12. OpenAI ChatGPT and Oracle APEX Generating Code - YouTube, accessed on July 22, 2025, https://www.youtube.com/watch?v=dMKZz15RJhc
13. Oracle APEX + AI - Oracle APEX, accessed on July 22, 2025, https://apex.oracle.com/en/platform/ai/
14. Create APEX Apps with AI | Oracle, accessed on July 22, 2025, https://www.oracle.com/artificial-intelligence/create-apex-apps-with-ai/
15. ChatGPT API - Oracle APEX, accessed on July 22, 2025, https://apex.oracle.com/pls/apex/r/abcsolar/fasterrocket/chatgpt-api?session=477645213816
16. Do it yourself — Building a ChatGPT Application with Oracle APEX | by Sonja Meyer, accessed on July 22, 2025, https://medium.com/@sonja.meyer/do-it-yourself-building-a-chatgpt-application-with-oracle-apex-627092b19f2f
17. Exploring the Synergy of OpenAI's ChatGPT and Oracle APEX: Transforming Oracle Applications - Insight - Summer 2023 - OATUG, accessed on July 22, 2025, https://www.oatug.org/insight-summer2023/features-archive/exploring-the-synergy
18. AI-driven cloud ERP: The next frontier in predictive financial management - ResearchGate, accessed on July 22, 2025, https://www.researchgate.net/publication/391437445_AI-driven_cloud_ERP_The_next_frontier_in_predictive_financial_management
19. 12 common pitfalls in LLM agent integration (and how to avoid them), accessed on July 22, 2025, https://www.barrage.net/blog/technology/12-pitfalls-in-llm-integration-and-how-to-avoid-them
20. LLMs Get Lost In Multi-Turn Conversation - arXiv, accessed on July 22, 2025, https://arxiv.org/pdf/2505.06120
21. (PDF) AI-Driven ERP Systems - ResearchGate, accessed on July 22, 2025, https://www.researchgate.net/publication/390267252_AI-Driven_ERP_Systems
22. Complexities of ERP Data Integration in Modern Businesses, accessed on July 22, 2025, https://blog.skyvia.com/overcoming-erp-data-integration-challenges/
23. LLM Security: Top 10 Risks and 7 Security Best Practices - Exabeam, accessed on July 22, 2025,

https://www.exabeam.com/explainers/ai-cyber-security/llm-security-top-10-risks-and-7-security-best-practices/
24. Comprehensive Guide to Large Language Model (LLM) Security | Lakera – Protecting AI teams that disrupt the world., accessed on July 22, 2025, https://www.lakera.ai/blog/llm-security
25. [2406.16937] A Complete Survey on LLM-based AI Chatbots - arXiv, accessed on July 22, 2025, https://arxiv.org/abs/2406.16937
26. How AI/LLMs Can Help, Hinder Developers - Communications of the ACM, accessed on July 22, 2025, https://cacm.acm.org/blogcacm/how-ai-llms-can-help-hinder-developers/
27. The Case for Compact AI - Communications of the ACM, accessed on July 22, 2025, https://cacm.acm.org/opinion/the-case-for-compact-ai/
28. [2503.22458] Evaluating LLM-based Agents for Multi-Turn Conversations: A Survey - arXiv, accessed on July 22, 2025, https://arxiv.org/abs/2503.22458
29. Telephone Surveys Meet Conversational AI: Evaluating a LLM-Based Telephone Survey System at Scale - arXiv, accessed on July 22, 2025, https://arxiv.org/html/2502.20140v2
30. [2507.12472] A Survey of AIOps in the Era of Large Language Models - arXiv, accessed on July 22, 2025, https://arxiv.org/abs/2507.12472