ENS 491-492 – Graduation Project

Final Report

Project Title: Automated Enterprise Resource Planning (ERP) Systems: Advancing Intelligence and Adaptability

Group Members:

#22424 Mehmet Barkın Palabıyık

#30299 Metin Gündüz

#22481 Hasan Oğul Çınar

#22425 Furkan Keskin

#30873 Ömer Fatih Tarım

#30996 Hüseyin Samed Dağcı

#30441 Egi Cekici

31037 Atacan Dilber

Supervisor(s): Mohaned Chraiti

Date:01/01/2026

Sabancı Üniversitesi

1. EXECUTIVE SUMMARY

Enterprise Resource Planning (ERP) systems are vital towards a system that incorporates the basic operations of an organization which in this case includes sales, procurement, inventory, finance, and project management. Although ERP systems are important strategically, they are still challenging to set up and apply where small and medium-sized businesses (SMEs) are concerned. These challenges come about due to the lack of fit between the complex structure of the technical systems and the real-world business processes, decision logic, and performance measurement requirements. This project will fill this gap by suggesting an interdisciplinary solution that incorporates the Industrial Engineering (IE) approach with computer science (CS)-based artificial intelligence techniques.

The project as an Industrial Engineering model displays real-life business processes in four sectors that have been used to represent the project - retail, warehouse/logistics, construction, and architecture. Business processes have been analyzed and documented formally, with Business Process Model and Notation (BPMN) and identification of crucial points of operation and decision and development of structured decision logic and key performance indicators (KPIs). These deliverables converted tacit business knowledge to explicit structured forms that can be used to make uniform and transparent decisions.

An AI-based conversational ERP decision-support system was built on this organized business foundation, by the Computer Science aspect of the project. The system enables the user to state his business requirements in natural language and gets context-aware ERP guidance. A Retrieval-Augmented Generation (RAG) model was deployed to make sure that the answers are based on the sector-specific ERP knowledge instead of the outputs of the general language models. The system facilitates multi-session discussions by storing persistent and real-time and streaming responses by a web-based interface.

The findings reveal that the combination of structured knowledge of IE and AI conversational systems may greatly enhance the usability, accessibility, and quality of decision making in ERP. Instead of automating the configuration of ERP, the system is used as a transparency tool which provides the gap between business requirements and the capabilities of the ERP. This model minimizes the dependence on external consultancy systems, improves the level of understanding among the users and provides the future intelligent ERP advisory products system with scalability.

2. PROBLEM STATEMENT

Enterprise Resource Planning (ERP) systems are made to combine and streamline business procedures like sales, purchase, inventory, accounting and project implementation. Even though ERP systems are claimed to enhance consistency of data, operational visibility and coordination, their use and implementation is extremely difficult (Davenport, 1998). These issues are manifest in the small and medium-sized enterprise (SME) in particular, where very little technical knowledge, lack of time, and money are major obstacles to the successful adoption of ERP (Monk and Wagner, 2013).

Traditional methods through which ERP is implemented place a lot of emphasis on technical configuration, pre-defined workflow and large documentation of the system. Whereas structural completeness is ensured, such approaches fail to capture the real business processes, operational decision-making practices and performance priorities of the business organizations (Laudon and Laudon, 2020). Consequently, users find it hard to transform the operational requirements as perceived in the real world into the patterns of ERP systems and thus end up using the system functionality sparingly and employing external consultants extensively. ERP configuration is a multi-criteria decision problem where there is a trade-off between the cost, efficiency, flexibility, availability of data, and the capability of the organization (Turban, Sharda, and Delen, 2017, an industrial engineering approach). Nonetheless, these decision structures and focus on transparent decision-based support are not explicitly modeled and are not available in most ERP systems.

The current achievements in artificial intelligence and large language models have offered new possibilities to enhance the user interactivity and usability of ERP. However, a significant number of currently available AI-based ERP products are based on rule automation or general conversation interfaces that are not grounded in domains and do not have contextual awareness (Walton and Xu, 2021). Such systems are normally not bundled with structured business process knowledge and as such, they fail to offer explainable and context-sensitive advice that can be linked to actual operational workflows.

Driven by these restrictions, this project will fill the root cause of the disconnects between business operations, the complexity of the ERP system and user decision-making. The fundamental issue to be explored in this research is whether there exists a unified model by which business processes, logic of decisions and performance criteria are translated in intelligible and easily accessible ERP guidelines. The project will integrate the development of a transparent, context-responsive, and user-friendly ERP decision-support model through a combination of Industrial Engineering process modelling, decision-making and performance measures of the system through the application of artificial intelligence methods and principles of Computer Science. The final vision is to make ERP more accessible and effective, especially among SMEs, without compromising human judgment or making decisions that lack explainability and have a sound process basis.

2.1. Objectives/Tasks

The main goal of the project is to create an interdisciplinary ERP decision support model that will connect the real-life business processes to intelligent and AI-based advice. To meet this general aim, the project was designed based on a series of complementary goals that incorporated the Industrial Engineering (IE) and Computer Science (CS) lenses.

Industrial Engineering In the perspective of Industrial Engineering, the initial aim was to formally study and model actual business activities in four real-life sectors including retail, warehouse/logistics, construction, and architecture. It was done through the creation of the standardized Business Process Model and Notation (BPMN) diagrams to model end-to-end workflows, roles, activities, and information flows in a standard and analyzable form (Dumas et al., 2018; OMG, 2014). The aim was to develop transparent and reusable process models that are realistic to the operations and aid systematized analysis.

The second goal was to define those essential decision points that are inherent in these business processes and to formalize the decision logic behind. Each decision point had its data inputs needed, validation rules and exception-handling mechanisms. This goal was to change implicit managerial judgment into explicit and structured decision logic, thus making the decision-making process consistent and explainable (Turban, Sharda, and Delen, 2017).

The third goal was to have industry specific frameworks of performance measurements through the identification of key performance indicators (KPIs) that would be directly related to the modeled processes and decision points. These KPIs were intended to convert the data generated by ERP into managerial actionable information, which would be used to assist performance measurement and improvement (Kaplan and Norton, 1996). The desired outcome was to make sure that ERP systems are not just repositories of information, but also good decision support tools.

In the framework of the structured IE outputs, the Computer Science objectives concerned the creation of an intelligent system that could use this base of knowledge. One of them was to develop and deploy an AI-based conversational ERP assistant that enables users to state business requirements in natural language as opposed to technical ERP language. The anticipated result was enhanced accessibility and usability of non-technical user guidance of ERP.

The other goal was to deploy a Retrieval-Augmented Generation (RAG) pipeline based on generating chatbot replies on sector-specific ERP module knowledge contained in a vector database. Such a strategy was supposed to enhance factual believability, situational relevance, and supportability of system suggestions (Walton and Xu, 2021). Other goals were to institute persistent conversation management through a database to provide multi-session interactions and to institute a modular backend and a web-based frontend architecture so that it would be scalable with maintainability and easy to use.

2.2. Realistic Constraints

An assortment of realistic constraints was observed in this project in terms of time, availability of data, computing capabilities, and the level of academics. These limitations had a lot of impact on the design, the methodology and the ultimate shape of the solution developed.

Time was one of the major constraints. The study was restricted to two semesters in academics, and this necessitated the need to strike a balance between the depth of analysis and practical feasibility. Consequently, this research focused on creating process templates that could be scaled and reused on four industries representing the whole organization over a comprehensive, organization-specific ERP deployment. This strategy enabled the project to capture critical operational trend whilst working within the stipulated time.

Another significant constraint was the data availability. Confidentiality and security issues could not allow access to live organizational ERP data. The project was therefore based on structured data based on open-source ERP documentation and simulated situations portraying the general industry practices. This limitation limited direct validation to real operational settings, but guaranteed data privacy and enabled generalized and realistic decision-support logic to be developed (Laudon and Laudon, 2020).

Computationally, the utilization of large language models and of recycling systems based on vectors raised limitations of resources. The use of high-performance computing infrastructure, i.e. dedicated GPUs, was out of the question given a student project. To overcome this limitation, the system was to use external-hosted language models, which are accessed over application programming interfaces (APIs) and lightweight embedding models, and an effective vector database to retrieve. This design option allowed creating a working and scalable prototype as well as having acceptable performance rates.

There was also a practical limitation of interdisciplinary integration. The conversion of the products of the Industrial Engineering process models, decision matrices, and performance metrics into machine-readable forms demanded ongoing integration of the perspective of the IE and the CS. The variation in disciplinary terminology and levels of abstraction was solved by refinement, through common documentation and through the design of the system in modules.

Lastly, the project had to be in line with the existing engineering and managerial standards, such as the conventions of business process modeling, decision-support concepts, ERP best practices (Dumas et al., 2018; Turban, Sharda, and Delen, 2017). The system was not supposed to be rigidly standardized but to be flexible and adaptive and still be transparent and interpretable. All these limitations made the project to be a realistic, defensible and extendable decision support framework in a real-life environment.

3. METHODOLOGY

This project methodology was developed to enable the structured approach to an interdisciplinary problem of ERP configuration through the combination of both process analysis using the Industrial Engineering approach with artificial intelligence techniques using Computer Science. Instead of adopting the ERP implementation as a technical activity, the methodology was based on the systematic translation of the actual business processes and decision-making logic into an intelligent, contextual decision-support system.

The initial methodological process was the analysis of the business operations and the modeling of the processes in four representative industries, namely: retail, warehouse/logistics, construction and architecture. Business process map Business Process Model and Notation (BPMN 2.0) were used to capture the end-to-end processes, roles, activities, decision points, and information flows in a standard and analyzable format (Dumas et al., 2018; OMG, 2014). These process models formed the basis of knowing how operations depended on each other and where the ERP-relevant points of integration would be.

After the process modeling, some decision points that were critical to the workflow were identified systematically. Decision requirements, data requirements, validation requirements and exception-handling mechanisms were explicitly specified at each decision point. This move converted tacit management decision-making into formal decision reasoning and made decisions transparent and consistent. The decision structures were developed in response to the actual operational limits, including inventory levels, project lifecycle, service level contracts, and budget limits, consistent with several accepted principles of decision support (Turban, Sharda, and Delen, 2017).

Simultaneously, the performance measurement frameworks were created to relate operational processes and decision logic to the measurable outcome. Key performance indicators (KPIs) specific to the industry were established and even connected to the ERP data sources. These KPIs aimed at transforming raw transactional material into action oriented managerial information to aid in monitoring, control, and continual improvement (Kaplan and Norton, 1996). The approach/methodology made sure that ERP systems were not used as passive data storage mechanisms by linking KPIs to process phases and decisions.

Formal knowledge base was then compiled into the structured products of the Industrial Engineering analysis i.e., process models, decision matrices, and KPI definitions and made ready to be integrated into intelligent systems. This knowledge base defined clear mappings between business operations, decision needs, and ERP modules, which articulate a consistent and lively picture of the association of operational requirements to system capabilities.

The construction of this systematic framework and subsequent application of artificial intelligence methods to facilitate user-friendly access to ERP guidance can be seen as the result of this paper. It has also taken a conversational decision support approach whereby users are able to give business needs in natural language. A Retrieval-Augmented Generation (RAG) methodology was applied to guarantee reliability and contextual relevance whereby relevant ERP knowledge is recalled by searching a structured repository and used as contextual input by a language model in generating responses (Walton and Xu, 2021). This method limits the system responses to domain knowledge as well as the potential of inappropriate or guesswork outputs.

Through methodology, transparency, explainability, and human-in-the-loop decision-making were emphasized. The system had been carefully created as a decision support tool and not as a decision maker so that the final responsibility goes to the user. This combined methodology framework made it possible to create an intelligent ERP assistant, rooted in the real business process, compatible with the decision support theory, and usable by non-technical users.

3.1 Overall System Architecture

The Computer Science perspective of this project was implemented as a modular, service-oriented system that connects a web based conversational interface to an AI empowered, ERP knowledge assistant. The architecture was designed; firstly, to keep the user experience as simple as much for normal users who have weak knowledge about the systems, secondly ensure that responses are based on significant ERP module data, and thirdly support multi session conversations with persistent and relevant history.

3.1.1 High-Level Components

Frontend(Web-Client).
A single page web application provides the primary user interface. It presents a multi-sessional chat experience (new chat, select chat, delete chat), renders messages in a more readable format (including formatted/structured bot answers), and supports real-time assisted responses for improved responsiveness and usability.

Backend(Application-Server/API-Layer).
An API acts as the organizer layer between the frontend, persistent storage, and the AI pipeline. So, the backend builds the backbone which provides the communication and infrastructure. It exposes endpoints for:

- Creating and managing chat sessions,
- Adding and retrieving messages for a given session,
- Sending user prompts to the AI pipeline and returning responses via streaming.

This separation allows the AI and data layers to evolve without requiring major changes in the user interface.

Persistent Storage (SQLite)
An SQLite database stores chat sessions and message history. The purpose is to maintain continuity across user interactions and enable retrieval of past conversations. This also makes the system usable in real scenarios where users return to earlier configuration discussions and continue iterating requirements.

AI Pipeline (LLM + Retrieval)
The assistant’s answer generation is organized around a Retrieval Augmented Generation (RAG) pattern. Instead of relying solely on the language model’s general knowledge, the system retrieves relevant ERP module information from a vector index and injects it into the prompt context. The LLM then generates responses constrained by that context, improving factual grounding and reducing hallucinations.

Vector Store (FAISS Index)
ERP domain data (sector/module descriptions and metadata) is embedded and indexed in a FAISS vector store. This enables fast similarity search during user queries and provides the “knowledge source” for the assistant.

3.1.2 Data Flow and Runtime Interaction

At runtime, a typical request follows this sequence:

1. **User Interaction (Frontend):** The user selects pre-existing chat sessions or starts a new chat and enters their prompt.

2. **Session Handling (Backend + SQLite):** The backend assigns a unique ID for each session and then stores the user message in SQLite. Finally, it prepares the AI request.

3. **Retrieval (RAG):** The system searches the FAISS database index to retrieve the most relevant ERP module chunks for the given prompt by using top-k retrieval.

4. **Answer Generation (LLM):** The retrieved chunks are composed into a context block. The LLM generates an answer using this context, following a “context-grounded response” policy.

5. **Streaming Response (Backend → Frontend):** The generated answer is delivered to the front end progressively, allowing users to view the response as it is produced and improving the overall responsiveness of the chat experience.

6. **Persistence (SQLite):** Once the response completes, the assistant message is stored and kept in the SQLite under the same session which enables later retrieval and continuation.

This end-to-end flow supports iterative, multi-turn requirement refinement while keeping the system maintainable and extendable.

3.1.3 Architectural Rationale

The overall architecture was not designed around a single ideal model but rather evolved based on practical development needs and limitations encountered during the project. Separating the user interface, backend logic, data storage, and AI-related components helped the team work on different parts of the system independently and made testing and debugging more manageable as the system grew.

Another motivation behind this design was long-term flexibility. Instead of tightly coupling system behavior to a single trained model, the project relies on a retrieval-based mechanism. This allows new ERP modules or sector-specific information to be incorporated by updating datasets and rebuilding the vector index, which is considerably more practical than retraining a language model whenever requirements change.

User interaction also played an important role in architectural decisions. Allowing users to see responses as they are being generated and giving them access to previous conversations made the system easier to use and more engaging during longer configuration discussions.

3.2 AI Model Selection and Language Model Integration

The AI model of the system is decided among the large language models to enable natural language interaction, fluent conversation and contextual reasoning for both ERP related queries and natural responses. Rather than developing a custom model from scratch, the project utilizes an external language model accessed through an API key. This decision was mainly driven by computational and economic limitations to provide needed flexible and scalable solutions within a student project setting.

The selected language model is used as a reasoning and generation engine, while domain specific knowledge is handled distinctly through a professional retrieval mechanism. This separation allows the model to focus on language understanding and response formulation instead of memorizing ERP specific information. As a result, the system remains adaptable and can evolve as new data is introduced.

Prompt design plays a significant role in controlling model behavior. The prompts are structured to clearly define the assistant’s role as an ERP-based helper and to restrict responses to the contextual information provided. If the retrieved data is not sufficient, the model is instructed to avoid speculative answers. This approach helps to reduce false and suspicious outputs and improves the reliability of recommendations.

3.3 Retrieval-Augmented Generation (RAG) Pipeline

The system uses a Retrieval-Augmented Generation (RAG) approach to ensure that chatbot responses are based on ERP-specific knowledge rather than generic language model outputs. ERP module names and the related sectors’ information is first collected from structured datasets and converted into textual form. These texts are divided into manageable chunks and transformed into vector embeddings, which are stored in a vector database to enable efficient similarity-based retrieval.

When a user submits a query, the most relevant information is retrieved from the vector store and included as contextual input to the language model. The model is instructed to generate responses only using this retrieved context and to avoid unsupported assumptions when sufficient information is not available. This approach improves the reliability of answers and helps align recommendations with the curated ERP dataset.

3.4 Conversation Management and Persistence

The system was designed to support mutual and relational conversations rather than treating each user input as an independent request. This is very crucial for ERP-related discussions, where users sometimes re-consider their requirements step by step and may need to revisit earlier recommendations. To enable this, conversations are managed using a session-based structure.

Each chat session and its associated messages are stored permanently in an SQLite database. This enables users to view and consider the previous conversations, continue previous discussions, and maintain the consistency. In addition to holding the past conversations by database, the system also considers and keeps the recent messages when generating responses, which helps the chatbot to stay consistent during an ongoing conversation. This approach keeps conversation persistent while keeping the overall system simple, efficient and lightweight.

3.5 Frontend-Backend Interaction and User Experience

The system contains a web-based chat interface that is designed as a single-page application (SPA), which has a user-friendly and continuous interaction layer between users and the ERP assistant. The frontend will allow people to start new conversations, to change between the already-created chat sessions, and to view past messages in a structured interface. The client side deals with session state whilst the server deals with conversation persistence, which enables users to engage in lengthy and iterative discussions involving ERP business processes with information being properly maintained.

The RESTful API architecture is used to communicate between the frontend and the backend through the use of HTTP. The front end sends POST requests on the HTTP to send user prompts and GET requests to get chat sessions and message history. These endpoints are provided by the backend as a lightweight application server and coordinate the process of request processing, database access and running of the AI pipeline. The layers communicate through the use of JSON-encoded payloads to facilitate data exchange to enable interoperability and easy extensibility.

Assistant responses are also streamed in real-time to the frontend incrementally via a server-side streaming service to enable real-time communication. When the language model produces partial outputs, such tokens are sent to the client and displayed in the chat interface in a stream, and not until the response is completed. This streaming method would greatly enhance the perceived responsiveness and match conversational patterns of interaction that are typically desired in current AI-driven systems.

After completion of response generation, it is saved to both user and assistant into an SQLite database with session based schema to continue a conversation even during subsequent user visits. The backend is an interoperative Retrieval-Augmented Generation (RAG) system which leverages similarity search of a FAISS vector index containing the relevant ERP-related knowledge and infused into the prompt of the large language model accessed via external API. This design permits that frontend interactions are straightforward and user friendly, whereas complicated tasks like retrieval, grounding, persistence and inference are processed in an opaque fashion in the backend infrastructure.

All in all, the front-end to back-end interaction was configured to be usable and yet technically sound. The system delivers responsive and scalable user experience with a distinct separation of concerns and a maintainable system architecture by combining restful communication, streaming response, persistent storage and retrieval based AI inference.

4. **RESULTS & DISCUSSION**

This project methodology was developed to enable the structured approach to an interdisciplinary problem of ERP configuration through the combination of both process analysis using the Industrial Engineering approach with artificial intelligence techniques using Computer Science. Instead of adopting the ERP implementation as a technical activity, the methodology was based on the systematic translation of the actual business processes and decision-making logic into an intelligent, contextual decision-support system.

The initial methodological process was the analysis of the business operations and the modeling of the processes in four representative industries, namely: retail, warehouse/logistics, construction and architecture. Business process map Business Process Model and Notation (BPMN 2.0) were used to capture the end-to-end processes, roles, activities, decision points, and information flows in a standard and analyzable format (Dumas et al., 2018; OMG, 2014). These process models formed the basis of knowing how operations depended on each other and where the ERP-relevant points of integration would be.

After the process modeling, some decision points that were critical to the workflow were identified systematically. Decision requirements, data requirements, validation requirements and exception-handling mechanisms were explicitly specified at each decision point. This move converted tacit management decision-making into formal decision reasoning and made decisions transparent and consistent. The decision structures were developed in response to the actual operational limits, including inventory levels, project lifecycle, service level contracts, and budget limits, consistent with several accepted principles of decision support (Turban, Sharda, and Delen, 2017).

Simultaneously, the performance measurement frameworks were created to relate operational processes and decision logic to the measurable outcome. Key performance indicators (KPIs) specific to the industry were established and even connected to the ERP data sources. These KPIs aimed at transforming raw transactional material into action oriented managerial information to aid in monitoring, control, and continual improvement (Kaplan and Norton, 1996). The approach/methodology made sure that ERP systems were not used as passive data storage mechanisms by linking KPIs to process phases and decisions.

Formal knowledge base was then compiled into the structured products of the Industrial Engineering analysis i.e., process models, decision matrices, and KPI definitions and made ready to be integrated into intelligent systems. This knowledge base defined clear mappings between business operations, decision needs, and ERP modules, which articulate a consistent and lively picture of the association of operational requirements to system capabilities.

The construction of this systematic framework and subsequent application of artificial intelligence methods to facilitate user-friendly access to ERP guidance can be seen as the result of this paper. It has also taken a conversational decision support approach whereby users are able to give business needs in natural language. A Retrieval-Augmented Generation (RAG) methodology was applied to guarantee reliability and contextual relevance whereby relevant ERP knowledge is recalled by searching a structured repository and used as contextual input by a language model in generating responses (Walton and Xu, 2021). This method limits the system responses to domain knowledge as well as the potential of inappropriate or guesswork outputs.

Through methodology, transparency, explainability, and human-in-the-loop decision-making were emphasized. The system had been carefully created as a decision support tool and not as a decision maker so that the final responsibility goes to the user. This combined methodology framework made it possible to create an intelligent ERP assistant, rooted in the real business process, compatible with the decision support theory, and usable by non-technical users.

5. IMPACT

The project illustrates that it has significant scientific, technological and socio-economic implications because it solves an established but unsolved problem in enterprise systems: that ERP platforms can be used, understood and readily available to real-world decision-making. The project will add a useful conceptualization to the sphere of ERP by implementing Industrial Engineering techniques in conjunction with decision-support mechanisms based on artificial intelligence, which will help optimize ERP performance without adding complexity to the system.

Scientific/methodological-wise, the project demonstrates that structured business knowledge may be operationalized into intelligent systems. A clear formulation of business processes, decision logic and performance measures will give a practical demonstration of how the traditionally qualitative managerial knowledge may be turned into machine-readable forms. Such a design complies with the established theory of explainable and context-aware AI applications in the enterprise setting and is part of the growing literature on systems to support decision support (Turban, Sharda, and Delen, 2017; Walton and Xu, 2021). Although the project does not present new theoretical models, it shows a good example of the interdisciplinary application of the existing methods in a coherent and replicable way.

The project has technological implications in the fact that it created an AI-based operational knowledge-based conversational ERP decision-support system. In contrast to generic chatbot solutions, the proposed system uses Retrieval-Augmented Generation (RAG) to make sure that the solution generated in the case of recommendations is based on domain-specific ERP information. Such a design enhances reliability, transparency, and trust by the user. Also, the functionality like the ability to have multiple sessions that are persistent and the stream of responses make the system more usable and fit into the expectations of how users can interact with the system. The modular structure also allows extension to other areas, ERP systems, or to sophisticated analytics.

Small and medium-sized enterprises (SMEs) are of socio-economic impact. High consultancy price, lack of internal capacity and ability to match the standardized ERP architectures with the unique business operation are some of the barriers to the adoption of ERP in SMEs (Monk and Wagner, 2013). The proposed system can decrease the reliance on the outside consultant and decrease the barrier to entry into successful ERP use by offering context-sensitive guidelines and making them accessible to users. This enhanced access can result in a greater operational efficiency enhanced informed decision-making and competitiveness among resource constrained organizations.

There is also innovative and commercial potential in the project. The framework devised may become a range of practical products or services, including an ERP advisory system based on AI, an intelligent ERP training and onboarding system or a decision-support layer that is part of the current ERP systems. Its ERP and platform agnostic and modular design can be adapted to various other industries and platforms, which enhances its commercialization or research-oriented development potential.

Regarding Freedom-to-Use (FTU) considerations, there were no clear problems that were detected in the academic scope of the project. The system is based on open-source tools, publicly available documentation and language models which are hosted externally accessed using standard APIs. To deploy the third-party services in the future, the licensing conditions of the third-party services and data protection laws and policies as well as the organizational security policies would have to be examined and discussed thoroughly.

Overall, the contribution of the project is that the difficulties of ERP usability can be resolved based on the interdisciplinary design instead of the enhanced complexity of the system. The project provides the scalability of process engineering rigor and an intelligent conversational interface to support the digital transformation in organizations in a socially relevant manner. Would also have to be adhered to in case actual user data is used.

6.ETHICAL ISSUES

The major ethical issue pursued in the project is the issue of artificial intelligence in decision-making in an organization. The system was meant to be a decision support tool and not a decision maker. The design option guarantees that accountability and responsibility are kept with the human users, and that there is no over-dependence on automated recommendations, and that managerial judgment is retained in the key business decisions. This human-in-the-loop solution is consistent with the accepted code of ethics of responsible use of decision supporting systems.

Ethical concerns included data privacy and data security also. No actual organizational data or sensitive personal information has been used in the project. Rather it is based on anonymized, synthetic and publicly available ERP-related data on which the system is developed and assessed. The history of conversations stored in the system is restricted to the interactions between users and assistants and not any confidential business information. There would also be other security measures like access control, encryption and adherence to data protection laws needed in a practical deployment scenario especially in instances where the system would have been linked to live ERP environments.

Another ethical aspect of the project is prejudice and equity. Implicit in business process models and decision rules might be historical or industry-based practices that may be biased. To address such risk, the decision logic applied in the system is well documented and transparent. This transparency enables the users and organizations to examine, adjust or customize the underlying rules and assumptions within their context of operation, and it thus minimizes the chances of blind or unintentional bias.

During the system design, explainability and transparency were considered as primary ethical requirements. System generated recommendations are based on documented business processes, decision criteria, and ERP module mappings. This enables users to have insight into why a certain recommendation is made and how these fits with their operations and not the opaque or black-box behavior commonly linked with AI systems.

There were no ethical concerns regarding the execution of hazardous materials, health and safety hazards, and other technologies that were prohibited by law in the framework of this project. The system can be reviewed as ethically designed and focused on responsibility, transparency, and user control in general. Although the proposed solution does not imply any major cases of ethical breach, ethical duty to continually monitor the data, train users, and control its application would be needed in the real-life context.

7.PROJECT MANAGEMENT

The project was initiated with the objective of developing an interdisciplinary solution to address the complex and usability challenges of Enterprise Resource Planning (ERP) systems. At the outset, a high-level project plan was established that defined the scope, timeline, and responsibilities of both Industrial Engineering (IE) and Computer Science (CS) components. The initial plan emphasized parallel progress, where business process analysis and system development would advance concurrently and inform each other throughout the project lifecycle.

During the early phases, the focus of the IE team was on understanding and modeling representative business operations across four sectors. In parallel, the CS team concentrated on designing the core architecture of an AI-driven ERP assistant. As the project progressed, iterative refinement became a central project management strategy. Early process models, decision structures, and system prototypes were continuously reviewed, discussed, and revised based on internal feedback and emerging technical insights. This iterative approach allowed the team to identify misalignments early and adjust both analytical and technical components accordingly.

One significant change from the initial project plan concerned the role of automation. The original concept included exploring partial automation of ERP configuration tasks. However, during implementation, it became evident that prioritizing transparency, explainability, and user understanding was more critical than automating system actions. As a result, the project scope was refined to focus on decision support rather than execution. This change required reallocating effort toward improving conversational flow, decision logic clarity, and system usability, demonstrating adaptive project management in response to practical and ethical considerations.

Another important adjustment involved system features related to user interaction. While the initial plan envisioned a basic chatbot interface, development experience highlighted the importance of conversation continuity and user experience in ERP-related discussions. Consequently, persistent conversation management and real-time response streaming were incorporated into the system. These features were not part of the original plan but were added to better reflect real-world usage scenarios and enhance system effectiveness.

Interdisciplinary coordination was a central challenge throughout the project. Differences in terminology, abstraction levels, and problem-solving approaches between IE and CS disciplines required deliberate communication and alignment. This challenge was addressed through regular coordination meetings, shared documentation, and the use of visual artifacts such as process diagrams and system flow representations. These practices facilitated a shared understanding of project goals and ensured that analytical outputs could be effectively translated into technical implementations.

From a resource management perspective, the project operated within the constraints of an academic environment. The use of open-source tools, publicly available documentation, and externally hosted AI services enabled the team to manage computational and financial limitations effectively. Time management was achieved by prioritizing core functionalities and deferring non-essential features, ensuring that the project objectives were met within the two-semester timeframe.

Overall, the project management experience highlighted the importance of flexibility, iterative development, and clear interdisciplinary communication in complex, technology-driven projects. The lessons learned emphasize that successful integration of engineering analysis and intelligent systems requires not only technical competence but also adaptive planning, continuous feedback, and a shared understanding of objectives and limitations.

8. CONCLUSION AND FUTURE WORK

This project aimed to tackle one of the long-standing and basic challenges in Enterprise Resource Planning (ERP) systems, which is the lack of connection between complex system designs and the actual business operations, decision-making, and performance assessment requirements. The project, by integrating Industrial Engineering (IE)-based approaches to artificial intelligence systems with Computer Science (CS)-based approaches, shows that ERP usability and effectiveness may be significantly enhanced by interdisciplinary design, as opposed to technical complexity.

In terms of Industrial Engineering, the project was successful in terms of converting the actual practices in operation into explicit representations. The evolution of normative business process models, formal decision logic, and industry-specific performance indicators offered a strong and well-integrated body of knowledge that covers the actual workings and decision-making process of organizations. Such outputs converted tacit managerial knowledge into explicit and comprehensible forms and allowed uniform and commendable decision support. That contribution proves the importance of process modeling and decision analysis in the process of closing the gap between business reality and enterprise systems.

Computer Science-wise, the project has shown the potential of artificial intelligence to be implemented in an enterprise-related and responsible way. The created AI-based conversational ERP assistant offers the user easy to access, context-sensitive advice based on systematic ERP knowledge. The system has been developed using a Retrieval-Augmented Generation (RAG) model, which guarantees that the responses are found on the domain-specific information and not the uncontrollable language model generation results. The conversational interaction, continuity between sessions, and delivery of service in real-time further increase usability without compromising transparency and user control.

The main product of the work is the conscious formation of the system as the decision-support tool and not an automated decision-maker. This architecture decision is based on the moral aspects and engineering limits in that the human judgment should be in the center, but smart advice should be taken. The project thus adds a pragmatic framework that is a balance between analytical rigour, technical feasibility and responsible use of AI.

The project has several limitations, although it has achieved its success. The system was designed as a prototype in an academic context and had not been put to test in a live organization setting. Simulated and publicly available ERP-related data will restrict the possibility of empirically validating that the system is effective in the real-life context. Furthermore, system assessment was based on functional and qualitative concerns and not on large-scale user testing or quantitative performance indicators.

These shortcomings are the indicators of the bright perspectives of future work. There is a natural extension to the integration of the decision support system with live ERP platforms that allow interacting with actual organizational data and configuration processes. Increasing the knowledge base into other industries, ERP modules, and organization roles would also be able to improve the applicability of the system. Progressive analytics and learning may be included in future studies as well to offer proactive suggestions and proposals, which may be based on the patterns observed and historical information. In addition, systematic user studies would enable the judgment of usability and decision quality as well as user trust.

On a larger scale, this project underscores a paradigm of enterprise system design based on the need to focus on alignment of business processes, decision logic and intelligent technologies. With organizations still actively working towards digital transformation, solutions with an ability to merge engineering rigor with explainable and user-friendly AI will gain more significance. The model created during this work provides the scalable and flexible base of such solutions and shows that interdisciplinary cooperation may deliver effective and tangible results in a complicated business setting.

9. APPENDIX

Appendix A: Integrated Business Process and Operations Framework

This appendix details the standardized approach used by the Industrial Engineering team to model business operations across four target sectors: retail, warehouse/logistics, construction, and architecture. These models represent a unified framework that served as the foundational business setting for the AI-driven ERP assistant

All business processes were developed using the **Business Process Model and Notation (BPMN 2.0)** standard to ensure technical clarity and compatibility.

- **Workflow Mapping:** Every sector follows a systematic end-to-end workflow, defining activities, roles, and information flow.
- **Standardized Elements:** The models utilize tasks for operational activities and gateways for decision logic to maintain consistency across different business environments.

A.2. Common Operational Lifecycle

The modeled processes follow a synchronized lifecycle, allowing the ERP system to provide consistent advice across various modules.

Phase	Core Activities Across Sectors	Related ERP Modules
Initiation	Lead qualification, project bidding, or sales order reception.	Sales, CRM, Project Management.
Processing	Material procurement, design phases, or inventory picking.	Purchase, Inventory, Manufacturing.
Control	Quality inspections, budget approvals, and exception handling.	Accounting, Quality Control.
Closure	Final delivery, project handover, and invoicing.	Invoicing, Accounting.

A.3. Decision Logic and Knowledge Base Integration

The process models serve as the primary source of truth for the AI assistant.

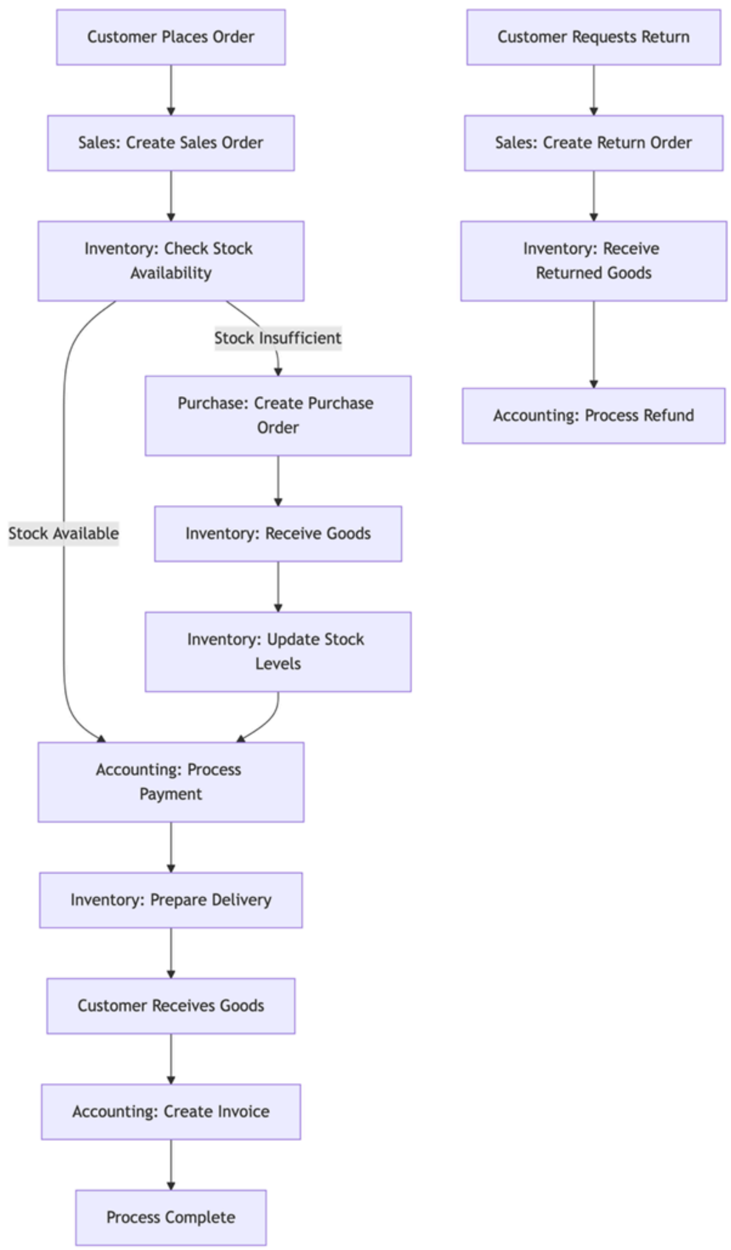
- **Contextual Grounding:** By mapping industry-specific tasks to modular ERP functions, the system provides advice grounded in specific operational contexts.

- **Structured Decision Points:** Critical nodes were identified where data requirements, validation rules, and exception handling are highest.

A.4. Industry-Specific Process Visuals (BPMN Gallery)

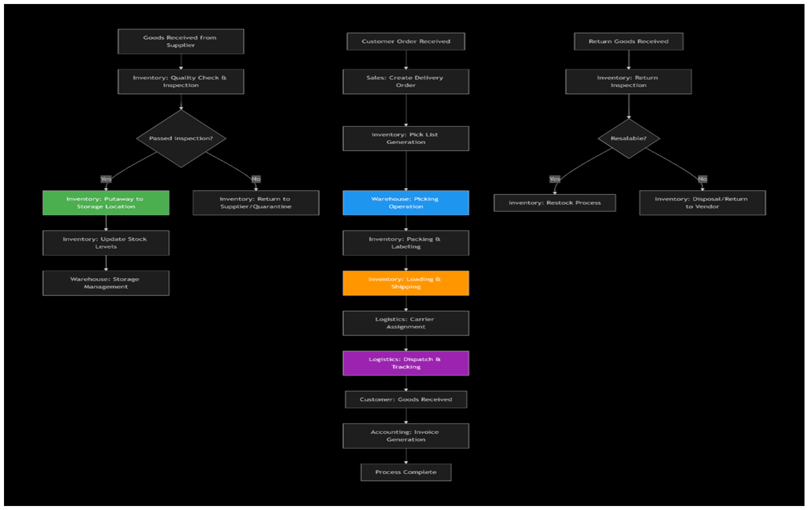
The following figures represent the detailed BPMN 2.0 diagrams developed for each industry. These visual aids define the process boundaries and integration points that directly inform the ERP configuration decisions.

Figure A.1: Retail Industry Operations Model



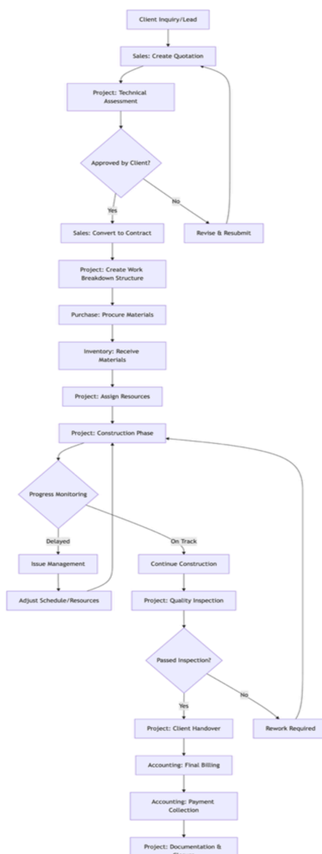
Focus: Multi-channel fulfillment and inventory routing logic.

Figure A.2: Warehouse and Logistics Management Model



Focus: Inbound/outbound logistics and operational accuracy tracking.

Figure A.3: Construction Project Lifecycle Model



Focus: Procurement loops and project milestone management.

Appendix B: Systematic Decision Matrices and Chatbot Logic

This appendix details the structured decision-making framework developed by the Industrial Engineering (IE) team to ground the AI assistant's recommendations in rigorous business logic. It bridges the gap between high-level business processes and technical AI execution.

B.1. Industry-Specific Decision Matrices

The following matrix defines the critical decision points identified across the four sectors. Each point is mapped to specific data requirements and logical rules that the AI assistant uses to generate accurate ERP guidance.

Sector	Decision Point	Required Data Inputs	Logic / Rule Applied	Exception Handling
Retail	Order Routing	Stock levels, distance, SLA	Assign to nearest hub with stock	Trigger inter-warehouse transfer
Warehouse	Quality Approval	PO specs vs. physical state	100% compliance for acceptance	Initiate rejected goods workflow
Construction	Procurement Timing	Project phase, site inventory	Order when stock < 20% of phase need	Escalate to Project Manager
Architecture	Resource Billing	Billable hours, project code	Match hours to specific milestones	Flag exceeding budget limits

B.2. Chatbot Decision Logic Framework

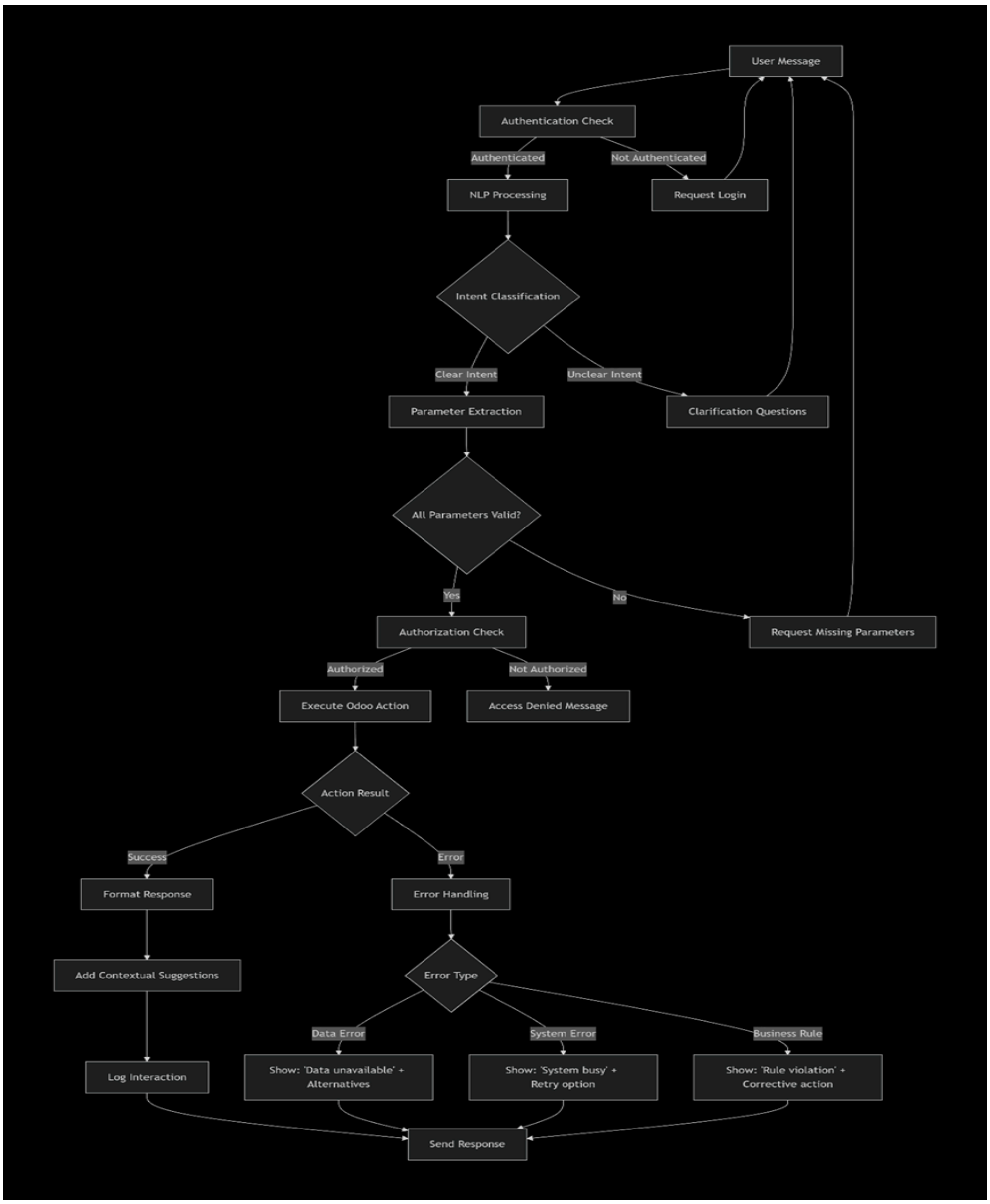
The intelligent assistant does not rely solely on general language model knowledge. Instead, it follows a structured logic path designed by the IE team to ensure responses are "context-grounded".

- **Step 1: Contextual Identification:** The system identifies the user's specific industry and operational role.
- **Step 2: Business Rule Retrieval:** The RAG pipeline fetches the relevant decision rules (from Section B.1) from the vector database.
- **Step 3: Logical Validation:** The LLM evaluates the user's query against pre-defined validation rules and thresholds.
- **Step 4: ERP Guidance Generation:** The final advice is produced, linking the decision back to specific ERP modules like Inventory, Sales, or Project Management.

B.3. Logic Visualization

The following diagram illustrates the internal decision-making flow of the chatbot, showing how it transitions from a natural language query to a structured ERP recommendation.

Figure B.1: Chatbot Decision Logic Flowchart



Description: This flowchart represents the hierarchical reasoning process, ensuring that AI prioritizes industry-specific business rules over generic assumptions.

Appendix C: Performance Measurement and KPI Framework

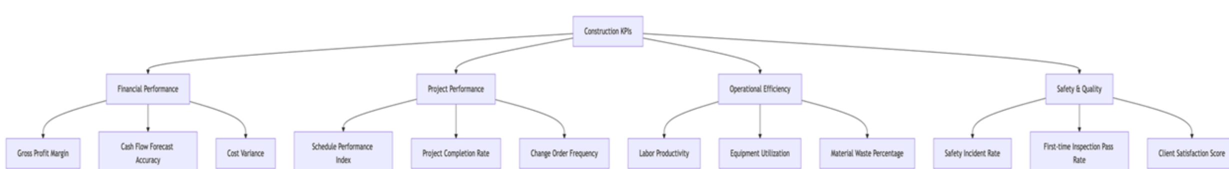
This appendix outlines the metrics established to assess operational effectiveness and the quality of decision-making across the selected industries. These Key Performance Indicators (KPIs) translate raw ERP data into actionable managerial knowledge.

C.1. KPI Matrix and Calculation Logic

The following indicators were selected to monitor the performance of integrated business processes:

Industry	Primary KPI	Calculation Formula	ERP Data Source
Warehouse	Inventory Accuracy	(Physical Stock/System Stock)×100	Inventory Module
Construction	Lead Qualification Rate	(Qualified Leads/Total Leads)×100	CRM Module
Retail	Order Fulfillment Time	Time of Delivery−Time of Order	Sales/Shipping
Architecture	Project Budget Adherence	(Actual Cost/Planned Budget)×100	Accounting

Figure C.1: Sample ERP Performance Dashboard. This visual represents how the raw data from the ERP modules is converted into real-time, actionable managerial information.



C.2. Diagnostic Implementation

These metrics are integrated into the AI-driven assistant as diagnostic tools. By tracking these indicators, the system identifies process bottlenecks and suggests corrective actions, ensuring that operations remain aligned with established industry standards.

Appendix D: ERP Module Integration Map

This appendix demonstrates the functional alignment between the modeled business processes and the specific Enterprise Resource Planning (ERP) modules. This mapping ensures that the AI assistant provides guidance grounded in the actual modular architecture of the system.

Figure D.1: Modular ERP Interface and Integration Map. This visual demonstrates the interconnected nature of the ERP modules used to support industry-specific decision logic.

Process Step	Odoo Module	Description
Lead & Quotation Management	Sales	Client inquiries, quotations, contracts
Project Planning & Execution	Project	WBS, scheduling, resource allocation, tracking
Material Procurement	Purchase	Supplier management, purchase orders
Inventory Management	Inventory	Material storage, tracking, issuance
Resource Management	Project/HR	Labor, equipment, subcontractor management
Quality & Safety	Project	Inspections, compliance, incident reporting
Financial Management	Accounting	Billing, cost tracking, payments
Document Management	Project/Documents	Drawings, permits, certificates, reports

10. REFERENCES

- Davenport, T. H. (1998). Putting the enterprise into the enterprise system. Harvard Business Review, 76(4), 121–131.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of Business Process Management (2nd ed.). Springer.
- Hammer, M., & Champy, J. (2009). Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business.
- ISO 9001:2015. (2015). Quality management systems — Requirements. International Organization for Standardization.
- Kaplan, R. S., & Norton, D. P. (1996). The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press.

- Laudon, K. C., & Laudon, J. P. (2020). Management Information Systems: Managing the Digital Firm (16th ed.). Pearson.
- Monk, E., & Wagner, B. (2013). Concepts in Enterprise Resource Planning (4th ed.). Cengage Learning.
- Object Management Group (OMG). (2014). Business Process Model and Notation (BPMN) Version 2.0.2. https://www.omg.org/spec/BPMN/2.0.2
- Porter, M. E. (1985). Competitive Advantage: Creating and Sustaining Superior Performance. Free Press.
- Sharda, R., Delen, D., & Turban, E. (2020). Business Intelligence, Analytics, and Data Science: A Managerial Perspective (4th ed.). Pearson.
- Slack, N., Brandon-Jones, A., & Burgess, N. (2022). Operations Management (10th ed.). Pearson.
- Turban, E., Sharda, R., & Delen, D. (2017). Decision Support and Business Intelligence Systems (10th ed.). Pearson.
- Weske, M. (2019). Business Process Management: Concepts, Languages, Architectures (3rd ed.). Springer.
- Singh, S. 2009. “Adaptive User Interfaces for ERP Systems.” In Proceedings of the International Conference on Information Systems (ICIS). Association for Information Systems.
- Svensson, G., N. Høgevold, and M. R. S. Moghadam. 2021. “ERP Implementation in Small Enterprises: A Resource Constraint Analysis.” Journal of Small Business and Enterprise Development 28 (1): 49–67. https://doi.org/10.1108/JSBED-02-2020-0041.
- Software Path. (2022). ERP System Report: 2022 Edition. Retrieved from https://softwarepath.com/guides/erp-software
- Walton, L., & Xu, J. (2021). Toward intelligent ERP: The convergence of AI and enterprise systems. *IEEE Access, 9*, 153240–153251. https://doi.org/10.1109/ACCESS.2021.3126816