Real Time Inventory Management System powered by Generative User Interface

Omkar Patil[1], Himani Kulkarni[2], Rohit Pottavathini[3], Ishaan Dhole[4], Prof. Krishna Salgaonkar[5]

[1234]Students, Department of Computer Science and Engineering (IOT And Cybersecurity Including Blockchain Technology), Smt. Indira Gandhi College of Engineering, Navi Mumbai, India
[5]Guide, Department of Computer Science and Engineering (IOT And Cybersecurity Including Blockchain Technology), Smt. Indira Gandhi College of Engineering, Navi Mumbai, India

---***---

Abstract - This research explores the developing a real time inventory management system powered by a generative user interface. We are leveraging large language models like GPT4, Claude 3, and Google Gemini that support tool calling or function calling, and integrating it with the modern frontend frameworks like Next js that support streaming React Server Component (RSC), the proposed system enables interaction with the inventory through natural language prompts. We are using PostgreSQL as a choice of database and server actions are used to interact with the database in real time. The system composes and renders appropriate react components based on user prompt, providing a personalized user experience. The research discusses the system's architecture, implementation, and potential impact on inventory management systems. It showcases the potential of Large Language Models (LLMs) and conversational interfaces in enhancing enterprise software user experiences.

***Key Words*:** Inventory Management System, Generative User Interface, Generative AI, Large Language Models, Conversational Interface, Natural Language Processing

1. INTRODUCTION

In the scene of big business programming, the client experience (UX) remains as an essential component impacting the achievement and reception of frameworks. Customary Stock Administration Framework (IMS), albeit famous for their unwavering quality, frequently slack as far as development in client collaboration and connection point plan. Through the user-friendliness of conversational interfaces and advanced generative AI models like GPT-4, Claude 3, and Google Gemini, inventory management systems now have a chance to completely reconsideration how users interrelate with them. These high-tech advances vow to improve client guarantee and efficiency, maybe varying the manner in which organizations pact with their stock and even out tasks. A decisive aspect of any company that deals with physical goods is inventory management. Compelling IMS are essential for keeping up with ideal stock levels, limiting wastage, and ensuring smooth tasks. While customary IMS have been fruitful in giving a unified stage to observing and supervision stock information, they frequently need natural UIs, making it trying for non-specialized staff to successfully explore and cooperate with the product.

Background and Motivation

Recent advancements in AI have resulted in the creation of powerful generative AI models like GPT-4, Claude 3, and Google Gemini. Their capacity to call capabilities through a Programming interface, which empowers these models to consistently create JSON yield incorporating contentions, addresses a critical movement.

While this capability calling ability doesn't straightforwardly produce visual parts, it has empowered engineers to join it with present day web structures like Respond to assemble imaginative UIs called generative UIs. By coordinating LLMs' capability calling capacity with Respond Server Parts, designers can make conversational connection points that powerfully render UI (UI) parts in light of client input, giving a customized and setting mindful experience.

Boosted by the capability of combining LLMs' capability calling capacity with Respond Server Parts to construct generative UIs, this exploration plans to foster an ongoing stock administration framework with such a conversational, generative connection point. This approach intends to change the manner in which clients communicate with stock information, making it more natural and operative analogized with conventional static connection points.

1.1 Objectives and Scope of the Research

The crucial goal is to foster an enduring stock administration framework with a generative UI controlled by enormous language models (LLMs) like GPT-4, Claude 3 group of models, or Google Gemini group of models, coordinated with the Vercel computer-based intelligence sdk [4], Respond Server Components [5], and Next.js system. The goal is to make inventory management more informal.

The critical viewing platform inside the degree are:

1. Planning a conversational connection point for regular language collaboration with stock information.
2. Constructing the generative user interface by combining React Server Components and LLMs' function calling for dynamic user interface rendering based on user input.
3. Planning Vercel man-made knowledge sdk and Next.js to help the generative UI system.
4. Using Next.js server exercises, ensuring continuous data synchronization for best-in-class stock information.
5. Researching likely impact, challenges, and cutoff points inside huge business programming setting. new research directions, potential enhancements, and their transferability to other fields.

To further develop client experiences in enormous business programming, particularly stock organization, the assessment intends to add to generative mimicked knowledge applications and conversational association focuses.

2. LITERATURE REVIEW

2.1 Traditional Inventory Management System (IMS)

Stock organization systems (IMS) have been an indispensable piece of enormous business resource orchestrating (ERP) deals with any consequences regarding associations across various undertakings. The purpose of these product applications is to provide a specialized platform for monitoring and controlling stock information, such as stock levels, request handling, and store network executives. Customers can typically view and control stock-related data through a dashboard interface provided by conventional IMS.

IMS have proven to be solid and reliable arrangements that help businesses improve their functional effectiveness and streamline their stock administration procedures. They offer abilities for revealing, robotized reorder point computations, and continuous stock following. In any case, conventional IMS much of the time need advancement as far as client experience (UX) and interface plan in spite of their practical benefits.

2.2 Challenges and Limitations of Existing IMS

While traditional IMS have been effective in managing inventory data, they face several challenges and limitations:

1. **Client Experience (UX) and Association point Plan:** Most IMS rely upon static, structure-based interfaces that can be difficult to investigate, especially for non-specific clients. These connection points habitually need easy to use and natural plan, bringing about a lofty expectation to learn and adapt and the chance of client dissatisfaction.
2. **Restricted Regular Language Collaboration:** Conventional IMS essentially depend on menu-driven or structure-based connections, which can be awkward and tedious for clients who lean toward normal language correspondence.
3. **Absence of Personalization:** IMS for the most part give a one-size-fits-all methodology, neglecting to take special care of individual client inclinations or context-oriented needs.
4. **Rigid Announcing and Investigation:** Numerous IMS offer restricted detailing capacities, making it trying to create redid reports or gain further bits of knowledge into stock information.

2.3 Emergence of Generative AI Models

The way we interact with technology has changed significantly as a result of the recent rise in generative AI models like GPT-4, the Claude 3 family, and the Google Gemini family. These high-level language models are fit for understanding and creating human-like text, empowering conversational collaborations and regular language handling (NLP) at an extraordinary level.

GPT-4, a multimodal, large-scale model that can produce text outputs from both images and text. GPT-4 demonstrates human-level performance on a variety of professional and academic benchmarks, including passing a simulated bar exam with a score in the top 10% of test takers. Nonetheless, GPT-4 is less skilled than people in some certifiable situations. GPT-4 is a Transformer based model pre-prepared to foresee the following token in a document. [1]

Claude 3 family of large multimodal models is the Claude 3 Opus; the fastest and cheapest models are the Claude 3 Haiku and the Claude 3 Sonnet. All new models have vision capabilities that enable them to process and analyze image data. The Claude 3 family demonstrates strong performance across benchmark evaluations and sets a new standard on measures of reasoning, math, and coding [2]

The Gemini family consists of Ultra, Pro, and Nano sizes, suitable for applications ranging from complex reasoning tasks to on-device memory-constrained use-cases. Our most advanced Gemini Ultra model outperforms the current state of the art in 30 of 32 benchmarks. These benchmarks include becoming the first model to achieve human-expert performance on the well-studied exam benchmark MMLU and improving the current state of the art in all 20 of the multimodal benchmarks we looked at.

Regular language questions and orders can now be utilized to cooperate with frameworks because of these generative computer-based intelligence models, which have introduced another period of conversational points of interaction. The way we communicate with programming may be altered by this restriction.

2.4 Conversational Interfaces and Their Advantages

UIs known as conversational connection points, visit interfaces, or conversational specialists empower normal language collaboration among people and programming frameworks. The ability of regular language processing (NLP) and natural language understanding (NLU) to decipher client input and provide pertinent responses or actions is influenced by these points of interaction. Conversational affiliation centers offer two or three benefits over normal graphical UIs (GUIs):

1. **Normal Language Collaboration:** Clients can talk with the structure using standard language, making the association more normal and simpler to utilize, especially for non-particular clients.
2. **Relevant Comprehension:** Conversational points of interaction are equipped for appreciating and answering setting, giving reactions that are more appropriate and individualized in light of the specific necessities or inclinations of the client.
3. **Diminished Expectation to absorb information:** With normal language cooperation, clients can sidestep the need to learn complex menu structures or explore through numerous screens, decreasing the expectation to absorb information related with conventional connection points.
4. **Extended Viability**: Conversational marks of collaboration can streamline endeavors by allowing clients to impart their assumptions directly, potentially decreasing the amount of advances expected to completely finish a given liability.
5. **Accessibility:** Conversational interfaces can provide an accessible and inclusive user experience for users who have disabilities or who prefer to interact without using their hands.

2.5 Web Development Trends

The fast advancement of web improvement innovations has prepared for building more refined and easy to use applications. Respond, a JavaScript library created and publicly released by Meta (previously Facebook), has turned into a famous decision for creating web applications because of its part based engineering and proficient delivering.

However, site improvement (Website design enhancement) and introductory burden execution might be impeded by Respond's client-side delivering approach. The meta-framework, which was built on top of React, was introduced by Next.js, a Vercel company, to address these obstacles. Next.js gives features, for instance, server-side conveying and customized static smoothing out, dealing with the show and Site enhancement detectable quality of Answer applications.

In June 2023, Vercel introduced the Vercel AI sdk, an open-source library designed to help developers build conversational, streaming, and chat user interfaces in JavaScript and TypeScript. The Vercel AI sdk supports popular frameworks like React/Next.js, Svelte/SvelteKit, and Vue/Nuxt, as well as Node.js, Serverless, and the Edge Runtime.

In June 2023, Vercel released an open-source library to help developers create conversational, streaming, and talk UIs using JavaScript and TypeScript. The Vue/Nuxt, Smooth/SvelteKit, Reply/Next.js, Node.js, Serverless, and Edge Runtime structures are all maintained by the Vercel artificial intelligence sdk.In Walk 2024, Vercel delivered its man-made knowledge sdk 3.0, which incorporates support for generative client interfaces.This implies that designers can now make applications in view of huge language models (LLMs) like GPT-4, the Claude 3 group of models, or the Google Gemini group of models. This element, which empowers the powerful delivering of visual parts in light of client input, empowers a customized.

By combining the power of React, Next.js, and the Vercel AI sdk, developers can create highly interactive and responsive web applications with generative user interfaces, potentially revolutionizing the way users interact with enterprise software, including inventory management systems.

3. PROPOSED SYSTEM ARCHITECTURE

3.1 Overview of the generative UI-powered IMS

The proposed system architecture aims to integrate cutting-edge gpt-4-turbo-2024-04-09 model with 128k context window by OpenAI with modern web development frameworks like React and Next.js, leveraging the capabilities of the Vercel AI sdk. The architecture can be broadly divided into three main components: the front-end interface, the back-end services, and the data storage layer.

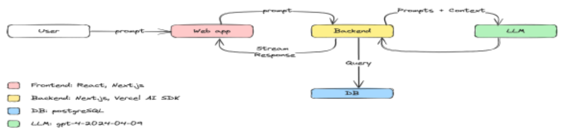


Fig -1: L1 Data flow diagram

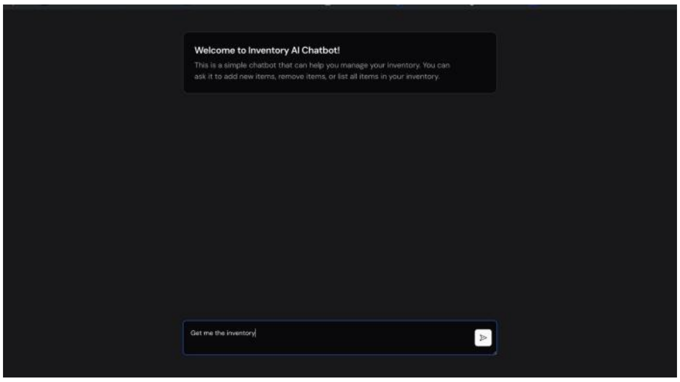
3.2 Integration of Generative AI Model

The focal point of the proposed structure lies in the compromise with generative man-made reasoning models. These models will comprehend and process user queries and commands in natural language, producing appropriate responses and actions related to inventory management tasks.

The Vercel simulated intelligence sdk, which offers a smoothed out interface for integrating different language models into the application, will make it simpler to consolidate the gpt-4-super 2024-04-09 model into the application. The sdk will deal with the correspondence between the model and the front-end interface, making it conceivable to handle client input and create pertinent results continuously.

3.3 Conversational Interface Design

The front-end interface of the proposed system will be built using React and Next.js, leveraging the capabilities of the Vercel AI sdk for creating a conversational, generative user interface (UI). The UI will feature a chat-like interface where users can interact with the system using natural language queries and commands.

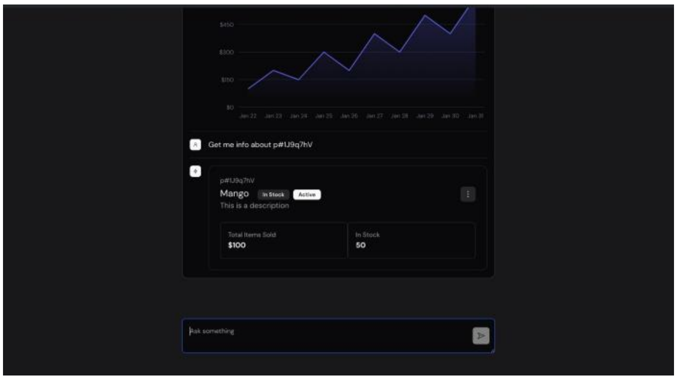


Welcome to Inventory AI Chatbot!

This is a simple chatbot that can help you manage your inventory. You can ask it to add new items, remove items, or list all items in your inventory.

Get me the inventory

Fig -2: Showing a chat-like interface where users can input natural language queries and commands



Get me info about p#1J9q7hV

p#1J9q7hV
Mango In Stock Active
This is a description

Total Items Sold
$100

In Stock
50

Ask something

Fig -3: Dynamically rendering relevant visual components and information based on the user input.

The Vercel AI sdk's generative UI support will enable the dynamic rendering of visual components based on user input and the responses generated by the integrated AI models. This will create a personalized and context-aware experience, where the UI adapts and presents relevant information and actions based on the user's specific needs and requests.

3.2 Real-time Data Synchronization and Updates

To ensure that the inventory management system operates with up-to-date data, the proposed architecture will incorporate real-time data synchronization and updates. This will be achieved through a combination of Next.js server actions and efficient data management techniques using a PostgreSQL[6] database.

Next.js server actions will be utilized to handle real-time updates and modifications to the inventory data, ensuring that any changes made through the conversational interface are immediately reflected in the PostgreSQL database. This will prevent potential data inconsistencies and provide users with accurate and reliable information at all times.

The PostgreSQL database will be designed to support efficient querying and retrieval of inventory data, leveraging its robust features for data integrity, scalability, and performance. This will enable fast response times and seamless integration with the gpt-4-turbo-2024-04-09 model and the front-end interface.

By utilizing a PostgreSQL database as the data storage layer, the proposed system will benefit from a reliable and battle-tested database management system, ensuring data consistency and enabling efficient real-time synchronization of inventory information.

4. IMPLEMENTATION DETAILS

In this section, we delve into the technical aspects of developing the IMS using Next.js, Vercel AI SDK, and the gpt-4-turbo-2024-04-09 model by OpenAI, including front-end and back-end development, as well as database management and integration.

4.1 Front-end Development

The front-end of the generative UI-powered Inventory Management System (IMS) is built using React and Next.js, leveraging the capabilities of the Vercel AI SDK. The choice of React and Next.js provides a robust and efficient foundation for building modern web applications with server-side rendering and static site generation capabilities.

React, a popular JavaScript library for building user interfaces, enables the development of reusable UI components, ensuring a modular and maintainable codebase. Next.js, a meta-framework built on top of React, enhances the development experience by providing features such as server-side rendering, static site generation, file-based routing, and API routes.

The front-end implementation includes the following key components:

1. **Conversational Interface**: A chat-like interface is implemented using React components, allowing users to interact with the system through natural language queries and commands. This interface dynamically renders UI elements based on the responses generated by the integrated generative AI model (gpt-4-turbo-2024-04-09).
2. **UI State Management**: The Vercel AI SDK introduces the concept of UI state, which is used to manage the rendering of UI components on the client-side. The UI state is updated based on the AI model's responses, ensuring a seamless and responsive user experience.
3. **Real-time Updates**: Next.js server actions are utilized to handle real-time updates and modifications to the inventory data. These server actions are triggered by user interactions with the conversational interface, enabling immediate synchronization with the back-end services and the PostgreSQL database.
4. **Routing and Navigation**: Next.js file-based routing system is utilized to define the application's routing structure, enabling seamless navigation between different pages or components of the IMS.
5. **Performance Optimization**: Techniques such as code splitting, lazy loading, and memoization can be employed to enhance the performance of the front-end application, ensuring a smooth and responsive user experience, even when dealing with large datasets or complex UI components.

4.2 Back-end Development

The back-end of the generative UI powered IMS is built using Next.js and the Vercel AI SDK. Next.js provides a robust server-side rendering (SSR) and API routing capabilities, while the Vercel AI SDK facilitates the integration of the gpt-4-turbo-2024-04-09 generative AI model.

The back-end implementation includes the following key components:

1. **AI Integration**: The gpt-4-turbo-2024-04-09 model is integrated through the Vercel AI SDK, which handles the communication between the front-end interface and the AI model. The SDK enables processing of user input and the generation of relevant outputs, including structured data and dynamic UI components.
2. **AI State Management**: The Vercel AI SDK introduces the concept of AI state, which represents the context of the conversation between the user and the AI model. This state is maintained on the server-side and updated based on user input and the AI model's responses, ensuring consistent and accurate conversation history.
3. **API Routes**: Next.js API routes are utilized to expose server-side functionality and interact with external services or databases. These routes can handle incoming requests, process data, and respond with the necessary information or actions.
4. **Database Integration**: The PostgreSQL database is integrated into the back-end to store and retrieve inventory data. Next.js server-side capabilities allow for efficient communication with the database, enabling real-time synchronization and updates of inventory information.

4.3 Database Management and Integration

The generative UI-powered IMS leverages a PostgreSQL database for efficient data storage and retrieval. PostgreSQL is a robust and feature-rich open-source database management system known for its reliability, data integrity, and performance.

The database integration includes the following key aspects:

1. **Database Schema Design**: A well-designed database schema is crucial for efficient data storage and retrieval. The schema should accurately represent the inventory data, including tables for items, categories, suppliers, sales records, and any other relevant information.
2. **Data Modeling**: Appropriate data modeling techniques, such as normalization and denormalization, are employed to ensure data integrity and optimize query performance.
3. **Query Optimization**: SQL queries are optimized for efficient data retrieval and modification, leveraging indexing, partitioning, and other performance-enhancing techniques as needed.
4. **Real-time Data Synchronization**: Next.js server actions are utilized to handle real-time updates and modifications to the inventory data, ensuring that changes made through the conversational interface

are immediately reflected in the PostgreSQL database. This synchronization is crucial for maintaining data consistency and providing users with accurate and up-to-date information.

By integrating the PostgreSQL database into the generative UI powered IMS, the application benefits from a reliable and scalable data storage solution, enabling efficient data management, query processing, and real-time synchronization of inventory information.

5. IMPLICATIONS AND FUTURE WORK

5.1 Potential Impact on Inventory Management Practices

The generative UI powered Inventory Management System (IMS) has the potential to revolutionize inventory management practices across various industries. By leveraging the power of generative AI models and conversational interfaces, this system can significantly improve user experiences and streamline inventory management processes. The following implications highlight the potential impact of the proposed system:

1. **Enhanced User Experience**: The conversational interface and natural language interaction capabilities of the system can greatly enhance the user experience for inventory management tasks. Users can interact with the system using familiar, natural language queries and commands, eliminating the need for extensive training or navigating complex menu structures.
2. **Increased Efficiency and Productivity**: By automating and simplifying various inventory management tasks through natural language commands, the system can significantly increase efficiency and productivity. Users can quickly access inventory information, generate reports, update stock levels, and perform other actions without the need for extensive manual input or navigation.
3. **Real-time Data Accessibility**: The integration of real-time data synchronization enables users to access the most up-to-date inventory information at all times. This can greatly enhance decision-making processes and ensure that businesses operate with accurate and reliable data.
4. **Scalability and Flexibility**: The modular architecture of the system, built on modern web development frameworks like React and Next.js, provides scalability and flexibility. As the business grows or requirements change, the system can be easily extended or modified to accommodate new features or integrations.
5. **Democratization of Inventory Management**: The conversational interface and natural language capabilities can make inventory management more accessible to a broader range of users, including those with limited technical expertise. This democratization can empower businesses to involve more stakeholders in inventory management processes, fostering collaboration and data-driven decision-making.

5.2 Challenges and Limitations of the Proposed System

While the generative UI powered IMS offers numerous benefits, it is essential to acknowledge and address potential challenges and limitations that may arise during its implementation and adoption:

1. **Data Quality and Consistency**: Ensuring the accuracy and consistency of the data fed into the generative AI models is crucial. Inaccurate or incomplete data can lead to erroneous outputs and undermine the system's reliability.
2. **Integration with Existing Systems**: Integrating the proposed system with existing enterprise resource planning (ERP) systems, inventory management software, or other business applications may pose challenges. Seamless data exchange and compatibility must be carefully addressed.
3. **User Adoption and Change Management**: Introducing a new system with a conversational interface and generative AI capabilities may require substantial user training and change management efforts. Users may initially resist adopting the new system, particularly if they are accustomed to traditional inventory management interfaces.
4. **Privacy and Security Considerations**: As the system handles sensitive inventory data, it is essential to implement robust security measures and adhere to data privacy regulations. Potential vulnerabilities and risks associated with natural language processing and generative AI models must be carefully evaluated and mitigated.
5. **Scalability and Performance**: While the proposed architecture is designed to be scalable, ensuring optimal performance under high-traffic or resource-intensive scenarios may require additional optimization efforts, such as load balancing, caching strategies, and infrastructure scaling.
6. **Explainability and Transparency**: Generative AI models are often perceived as "black boxes," making it challenging to explain their decision-making processes or outputs. Addressing the need for transparency and explainability in the context of inventory management may be necessary for user trust and accountability.

5.3 Future Enhancements and Research Directions

The generative UI powered IMS presents numerous opportunities for future enhancements and research directions, as the field of generative AI and conversational interfaces continues to evolve rapidly. Some potential areas for future work include:

1. **Multi-modal Interaction**: Extending the conversational interface to support multi-modal interactions, such as voice commands, gesture recognition, or augmented reality (AR) overlays, can further enhance the user experience and accessibility of the system.
2. **Advanced Analytics and Forecasting**: Integrating advanced analytics and forecasting capabilities into the system can provide valuable insights into inventory trends, demand patterns, and supply chain optimization. Machine learning models could be employed to analyze historical data and generate accurate forecasts, enabling proactive inventory management strategies.

3. **Intelligent Decision Support**: Exploring the potential of generative AI models to provide intelligent decision support and recommendations for inventory management tasks. This could involve incorporating domain-specific knowledge, constraints, and optimization algorithms to suggest optimal inventory levels, reorder points, or procurement strategies.
4. **Personalization and Adaptive Learning**: Implementing personalization features that allow the system to adapt to individual user preferences, learning patterns, and context-specific requirements. This could involve leveraging machine learning techniques to continuously improve the conversational interface and provide tailored recommendations.
5. **Distributed Inventory Management**: Investigating the applicability of the proposed system in distributed inventory management scenarios, such as supply chain networks or multi-warehouse environments. This could involve exploring techniques for data synchronization, conflict resolution, and collaborative decision-making across multiple locations or stakeholders.
6. **Domain-specific Extensions**: Exploring the potential of extending the generative UI powered IMS to other domains or industries beyond inventory management. This could involve adapting the system's architecture, integrating domain-specific knowledge bases, and developing tailored conversational interfaces for different application areas.
7. **Integration with Internet of Things (IoT) Devices**: Investigating the integration of IoT devices and sensors into the system for real-time inventory tracking, monitoring, and automation. This could involve developing interfaces for IoT data ingestion, analysis, and integration with the generative AI models and conversational interface.

By continuously exploring these future enhancements and research directions, the generative UI powered IMS can evolve to address emerging challenges, leverage advancements in generative AI and conversational interfaces, and provide even more sophisticated and intelligent inventory management solutions.

6. CONCLUSIONS

6.1 Summary of Findings

This research developed a generative UI powered Inventory Management System (IMS) by integrating the gpt-4-turbo-2024-04-09 model, conversational interfaces, and modern web frameworks like React and Next.js. The proposed system aimed to address limitations of traditional IMSs by enabling natural language interaction and dynamic UI rendering based on AI model responses.

The architecture leveraged the Vercel AI SDK, GPT-4 function calling, and a PostgreSQL database for real-time data synchronization. The implementation details covered front-end (React, Next.js), back-end (Vercel AI SDK, Next.js server actions), and database integration.

6.2 Contributions and Significance

This research contributes a conversational interface for natural language interaction with inventory data, addressing traditional IMS limitations. The real-time data synchronization and modular architecture built on React and Next.js provide up-to-date information, scalability, and flexibility.

The generative UI powered IMS demonstrates the potential of generative AI models and conversational interfaces in revolutionizing user experiences in enterprise software and inventory management practices. It paves the way for further advancements in this field, leveraging emerging technologies to improve processes and drive innovation.

REFERENCES

1. Achiam, Josh, et al. "GPT-4 Technical Report." arXiv preprint arXiv:2303.08774 (2023).
2. Anthropic, "The Claude 3 Model Family: Opus, Sonnet, Haiku." https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf (Accessed: Apr 2024)
3. Team, Gemini, et al. "Gemini: A Family of Highly Capable Multimodal Models." arXiv preprint arXiv:2312.11805 (2023).
4. Vercel AI SDK, "Generative UI Docs." [Online], https://sdk.vercel.ai/docs/concepts/ai-rsc (accessed: Apr 2024)
5. Dan Abramov, Lauren Tan, Joseph Savona, and Sebastian Markbåge, “Introducing Zero-Bundle-Size React Server Components” [Online], https://react.dev/blog/2020/12/21/data-fetching-with-react-server-components (2020) (accessed: Apr 2024)
6. Stonebraker, Michael, and Lawrence A. Rowe. "The design of POSTGRES." SIGMOD Record 15.2 (1986): 340-355. DOI: https://doi.org/10.1145/16856.16888.