

DeepSeek AI using Next.js

Ms. Sapalika Sahoo (2201298158)

Student, Dept. of CSE,
GIFT Autonomous, Bhubaneswar

Ms. Truptimayee Brahma

(2201298495)

Student, Dept. of CSE IOT,
GIFT Autonomous, Bhubaneswar

Asst. Prof. Pritiprava Mishra

Professor Dept. of CSE, GIFT
GIFT Autonomous, Bhubaneswar

Abstract— This project presents the design and development of an intelligent AI-based chatbot using the capabilities of DeepSeek integrated with the modern web framework Next.js. The primary objective of the system is to create an interactive platform that enables users to communicate with an AI-powered assistant capable of understanding and responding to natural language queries in real time. The chatbot leverages advanced techniques from Artificial Intelligence and Natural Language Processing to interpret user input and generate meaningful, context-aware responses. The DeepSeek API serves as the core processing unit, handling the analysis and generation of responses, while Next.js provides a robust and efficient environment for building a dynamic and responsive user interface. The system is designed to be scalable, efficient, and user-friendly, making it suitable for applications in customer support, education, and information services. The results demonstrate that the developed system is capable of delivering accurate and timely responses, thereby enhancing user interaction and reducing the need for manual intervention.

Keywords—Artificial Intelligence, Natural Language Processing, DeepSeek API, Next.js, Chatbot, Web Application, Real-Time Response.

I. INTRODUCTION

Artificial Intelligence (AI) is transforming the way humans interact with technology by enabling machines to perform tasks that typically require human intelligence. One of the most impactful applications of AI is the development of chatbots, which can simulate human conversation and provide automated, intelligent responses to user queries. These systems are widely used in customer service, education, business automation, and virtual assistance.

With the advancement of AI models, developers can now integrate powerful language processing capabilities into web applications. DeepSeek provides advanced AI models that can understand natural language and generate meaningful responses, making it an ideal choice for building intelligent chatbot systems.

In this project, an AI-based chatbot is developed using DeepSeek API and implemented through Next.js, a modern web development framework. Next.js enables efficient server-side rendering, fast performance, and scalable application development, making it suitable for building dynamic and interactive web applications.

The system allows users to interact with the chatbot through a web interface. The user's input is processed by the backend, sent to the DeepSeek API, and the generated response is displayed in real time. This integration demonstrates how AI can be effectively combined with modern web technologies to create intelligent and user-friendly applications.

The main goal of this project is to design and implement a responsive AI chatbot that enhances user interaction, provides accurate responses, and showcases the practical use of Artificial Intelligence in web development.

II. LITERATURE REVIEW

The development of chatbot systems has evolved significantly over the years, moving from simple rule-based systems to advanced AI-driven conversational agents. Early chatbot systems were primarily based on predefined rules and decision trees, where responses were generated based on specific keywords or patterns. Although these systems were easy to implement, they lacked flexibility and were unable to handle complex or ambiguous user queries effectively.

With the advancement of Artificial Intelligence, modern chatbot systems have incorporated machine learning and deep learning techniques to improve their performance. These systems are capable of learning from large datasets and can generate dynamic responses based on user input, significantly enhancing contextual understanding and accuracy.

A major breakthrough in chatbot development is the use of Natural Language Processing (NLP), which enables machines to understand, interpret, and generate human language. NLP techniques such as tokenization, parsing, and sentiment analysis

allow chatbots to process user queries more effectively. Advanced language models further enhance this capability by providing context-aware responses.

In recent years, the introduction of AI-based APIs has simplified the development of intelligent systems. Platforms like DeepSeek provide powerful pre-trained models that developers can integrate into applications without the need for extensive training or computational resources. From a web development perspective, modern frameworks such as Next.js have made it easier to build scalable, high-performance applications with server-side rendering and efficient routing.

III. SYSTEM ANALYSIS

System analysis is a crucial phase in the development of any software project, as it involves understanding the requirements, identifying system functionalities, and evaluating the feasibility of the proposed solution. The primary goal is to clearly define how the system will operate, what inputs it will accept, and what outputs it will generate.

A. Existing System Analysis

The existing systems for handling user queries are mainly based on manual support or rule-based chatbots. These systems rely on predefined responses and lack the ability to understand complex user inputs, often failing to provide accurate and context-aware responses. Additionally, traditional systems are not scalable and may struggle to handle multiple users simultaneously, requiring significant human intervention.

B. Proposed System

The proposed system is an AI-powered chatbot that utilizes the DeepSeek API to generate intelligent responses. The system is designed using Next.js, providing a fast and efficient platform for web application development. User queries are processed in real time using AI models, with the chatbot understanding natural language input and generating appropriate responses, improving the overall user experience.

IV. SYSTEM REQUIREMENTS

System requirements describe the detailed functions, features, and constraints the system must satisfy to meet user needs. They specify functional requirements (what the system should do) and non-functional requirements (how the system should perform).

A. Functional Requirements

The system shall allow users to enter queries through a web interface, process user input using AI techniques, and communicate with the DeepSeek API. It shall generate and display responses in real time, handle multiple user requests simultaneously, provide a user-friendly interface for interaction, and manage input and output efficiently.

B. Non-Functional Requirements

The system shall maintain high performance and scalability to support growing user demand. It shall ensure data security during API communication, provide a responsive and accessible interface across devices, and maintain reliable uptime and stable performance under concurrent usage.

C. Hardware and Software Requirements

Hardware requirements include a modern processor, minimum 8 GB RAM, and stable internet connectivity. Software requirements include Node.js runtime, Next.js framework, DeepSeek API access, JavaScript programming language, Visual Studio Code editor, and Google Chrome browser for testing.

V. SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data flow of a system to meet the specified requirements. The design focuses on creating a scalable, efficient, and user-friendly system capable of processing user queries and generating real-time responses.

A. System Architecture

The system follows a three-tier architecture consisting of the Presentation Layer (Frontend), the Application Layer (Backend), and the Integration Layer (API). The frontend built with Next.js handles user interaction, the backend manages business logic and routing, and the DeepSeek API processes natural language queries using advanced AI models. DeepSeek's Mixture-of-Experts (MoE) architecture activates 21B of 236B parameters per token, enabling efficient and high-quality responses with a context window of up to 128,000 tokens.

B. ER Diagram

The data model consists of four entities: USER (stores user information), CHAT_SESSION (stores chat session details with User_ID foreign key), MESSAGE (stores each message with Session_ID foreign key, Sender, Message_Text, and Timestamp), and API_RESPONSE (stores AI response generated for each message). Relationships: one User has many Chat Sessions; one Chat Session contains many Messages; one Message generates one API Response.

C. Security Design

The system secures API keys through environment variables, preventing exposure in the client-side code. All API communication occurs over HTTPS. Input sanitization is applied to user queries before forwarding them to the backend, and rate limiting is implemented to prevent abuse of the AI processing resources.

VI. IMPLEMENTATION

Implementation is the phase where the system design is converted into a working application. The AI chatbot is developed by integrating DeepSeek with Next.js to create a responsive and intelligent web-based system involving coding, API integration, and building the user interface.

A. Development Environment

The development environment uses JavaScript as the programming language, Next.js as the framework, Node.js as the runtime, Visual Studio Code as the code editor, the DeepSeek API for AI processing, and Google Chrome for browser testing.

B. Frontend Implementation

The frontend is developed using Next.js to provide an interactive and user-friendly interface. Key features include an input field for user queries, a chat display area showing the conversation history, a send button, and a fully responsive design that adapts to different screen sizes. The frontend captures user input and forwards it to the backend for processing.

C. Backend Implementation

The backend handles the core application logic and API communication. It receives user input from the frontend, processes the request, forwards the query to the DeepSeek API, receives the AI-generated response, and returns it to the frontend for display. Next.js API routes are used to implement the backend endpoints.

D. API Integration

The core functionality of the chatbot is achieved through integration with the DeepSeek API. The integration flow is: (1) user enters query, (2) query is sent to backend, (3) backend sends request to DeepSeek API using the "deepseek-chat" model with a messages array in JSON format, (4) API processes input using AI models, (5) response is returned to backend, and (6) response is forwarded to the frontend for display.

VII. API INTEGRATION

API Integration is the process of connecting the application with external services to enable communication and data exchange. The DeepSeek API acts as the core component providing AI capabilities, allowing the system to understand natural language and respond effectively.

A. Working of API Integration

The flow of operation involves: the user entering a query in the frontend; the request being sent to the backend server; the backend forwarding an API request to DeepSeek; the API processing the input using AI models; the response being generated and returned to the backend; the backend forwarding the response to the frontend; and the user seeing the output displayed in the chat interface.

B. API Request and Response Structure

The system communicates with the API using HTTP POST requests in JSON format, specifying the model "deepseek-chat" and a messages array containing the user role and content. The API response returns a JSON object containing the generated response text, which is extracted and displayed to the user in real time.

C. Security and Challenges

API security is maintained by storing API keys in server-side environment variables and never exposing them to the client. Challenges include dependency on internet connectivity, API rate limiting, managing response latency, and ensuring input validation to prevent prompt injection attacks.

VIII. TESTING

Testing is an important phase in software development that ensures the system works correctly and meets the specified requirements. It helps identify errors, bugs, and performance issues before deployment.

A. Objectives of Testing

The objectives of testing are to verify system functionality, ensure accurate AI responses, detect and fix errors, improve system performance, and ensure user satisfaction with the interface and response quality.

B. Types of Testing

Unit Testing verifies individual components such as the input field and API call functions. Integration Testing verifies the interaction between frontend and backend, ensuring proper API communication. System Testing verifies the complete system end-to-end. User Acceptance Testing (UAT) involves testing by end users to confirm the system meets their expectations.

C. Testing Results

Testing results confirmed that the chatbot responds to user queries in real time with accurate and meaningful responses. Smooth interaction between frontend and backend was verified, proper communication with the DeepSeek API was confirmed,

and the user-friendly interface was validated across multiple devices and screen sizes.

IX. RESULTS AND DISCUSSION

The AI chatbot system was successfully implemented and tested under different conditions. The system was evaluated for its ability to handle user queries, generate responses, and provide a smooth user experience.

A. Performance Analysis

Response time was observed to be fast, typically within a few seconds per query. Accuracy was high for general queries, demonstrating effective use of the DeepSeek AI models. The system handled multiple users effectively, demonstrating good scalability. Stable performance was maintained throughout testing with no critical failures observed.

B. Discussion

The chatbot demonstrates the effective use of modern AI technologies in web applications. The integration of the DeepSeek API allows the system to generate intelligent responses without requiring complex model training. Compared to traditional rule-based systems, the proposed chatbot offers dynamic responses, improved interaction quality, and better scalability. Certain limitations such as dependency on internet connectivity and external API availability still exist but can be addressed in future enhancements.

X. FUTURE ENHANCEMENTS

The current AI chatbot provides a solid foundation for further development. Several additional features can improve system efficiency, scalability, and user experience in future iterations.

Voice-Based Interaction: Integration of speech-to-text and text-to-speech capabilities to enable hands-free communication.
Multilingual Support: Support for multiple languages to improve accessibility for global users.
User Authentication: Addition of login/signup functionality to maintain personalized user data.
Chat History Storage: Storing conversations in a database to allow users to view past chats.
Mobile Application Development: Development of Android/iOS versions to increase accessibility.
Personalization: Learning user preferences to provide customized responses over time.

XI. CONCLUSION

The project "DeepSeek AI using Next.js" was successfully designed and implemented to provide an intelligent and interactive communication system. The main objective of developing a web-based chatbot capable of understanding user queries and generating meaningful responses in real time using DeepSeek integrated with Next.js was fully achieved.

The developed system demonstrates the practical application of Artificial Intelligence and Natural Language Processing in modern web development. The chatbot successfully processes user input, communicates with the DeepSeek API, and generates context-aware responses efficiently. The use of Next.js ensures better performance, scalability, and user experience.

The project highlights the advantages of integrating AI services through APIs, which simplifies development while providing advanced AI functionalities. Overall, the project successfully demonstrates how AI-powered chatbot systems can improve communication, automate user interaction, and enhance user experience, providing a strong foundation for future research and development in intelligent conversational applications.

XII. REFERENCES

- [1] DeepSeek API Documentation. [Online]. Available: <https://api-docs.deepseek.com/>
- [2] Next.js Official Documentation, Vercel. [Online]. Available: <https://nextjs.org/docs>

- [3] React Official Documentation, Meta Platforms Inc. [Online]. Available: <https://react.dev/>
- [4] Node.js Documentation, OpenJS Foundation. [Online]. Available: <https://nodejs.org/>
- [5] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [6] T. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [7] MDN JavaScript Guide. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- [8] W3Schools Next.js Tutorial. [Online]. Available: <https://www.w3schools.com/>
- [9] GeeksforGeeks Next.js Tutorial. [Online]. Available: <https://www.geeksforgeeks.org/>
- [10] OpenAI Research. [Online]. Available: <https://openai.com/research>
- [11] Hugging Face Documentation. [Online]. Available: <https://huggingface.co/docs>
- [12] IEEE Xplore Digital Library. [Online]. Available: <https://ieeexplore.ieee.org/>