International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

AI-Enhanced Interactive Portfolio using Next.js, OpenRouter, and Qdrant

Swathi Damodharan[1], Shakila Siddavatam[2]

[1]Master's Student, Department of Computer Science, Abeda Inamdar Senior college, India
[2]Head of Department, Department of Computer Science, Abeda Inamdar Senior college, India

Abstract
In today's competitive job market, personal portfolio websites have become essential tools for students and professionals to showcase their skills, projects, and achievements. However, most existing portfolio platforms are static and lack intelligent features that can personalize content or provide insights aligned with individual career goals. This project proposes the development of an AI-enhanced personal portfolio website that leverages Natural Language Processing (NLP) techniques to make portfolios more dynamic, interactive, and informative.
The system integrates NLP to automatically extract key information from resumes, generate professional summaries, identify skill gaps, and offer personalized recommendations. Additionally, a built-in chatbot provides instant responses to portfolio queries, enhancing engagement and accessibility for recruiters. The project uses Python-based NLP libraries such as SpaCy and transformers for text processing, and includes modules for skill analysis using similarity measures and topic modeling.
A review of existing research and tools reveals significant gaps in user interactivity, intelligent automation, and real-time feedback in portfolio platforms. By addressing these limitations, the proposed solution stands out by offering a smart, self-improving, and career-oriented digital presence tailored to the needs of modern tech professionals. This portfolio system has potential applications in recruitment, academic profiling, and personal branding, offering a new standard for AI-integrated digital portfolios.

Keywords: Artificial Intelligence, Natural Language Processing, AI Chatbot, Semantic Search, Vector Database

1. Introduction

1.1. Problem Statement

In recent years, personal portfolio websites have become essential tools for students, developers, and professionals to present their skills, achievements, and experiences. However, the majority of existing portfolio systems are static in nature and offer limite-d interactivity. Users can only view predefined content without the ability to engage or inquire dynamically about specific information.

1.2. Significance

This lack of interactivity reduces user engagement and fails to deliver personalized experiences. Visitors who wish to learn about a developer's specific skill set, academic background, or project details must manually navigate through multiple sections, which can be time-consuming and inefficient.
With the advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies, it is now possible to design intelligent systems that understand user queries and respond

contextually. However, these capabilities have not yet been widely applied in personal portfolio systems. Therefore, the research problem addressed in this project is:

> "To design and develop an AI-powered interactive portfolio system that uses natural language understanding to provide personalized, context-aware responses to user queries."

This system aims to integrate AI-driven conversational capabilities into a web-based portfolio using Next.js for the front-end framework, OpenRouter API for AI interaction, and Qdrant as a vector database for efficient semantic information retrieval.

1.3. Proposed Solution

The proposed system transforms the conventional static portfolio into an AI-driven intelligent assistant, allowing users to interact conversationally and explore information in a more natural and meaningful way.

2. Literature Review

The rapid advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP) has significantly transformed the way users interact with web-based systems. Traditional websites rely on static content and manual navigation, which limits user engagement and accessibility. Research in NLP and conversational systems has focused on enabling machines to understand and respond to human language more naturally and effectively.

Jurafsky and Martin [1] provide a comprehensive foundation for understanding NLP techniques such as tokenization, language modeling, and semantic analysis, which are essential for building intelligent conversational systems. Their work highlights how NLP enables systems to interpret user intent rather than relying solely on keyword matching. This forms the theoretical basis for conversational interfaces used in modern AI-driven applications.

The emergence of Large Language Models (LLMs) has further enhanced conversational AI capabilities. Brown et al. [2] demonstrated that large-scale transformer-based models can perform complex language tasks using few-shot learning, significantly improving response quality and contextual understanding. Similarly, Vaswani et al. [3] introduced the transformer architecture, which forms the backbone of most modern LLMs and enables efficient processing of long textual inputs.

Recent studies have emphasized the importance of combining language models with external knowledge sources to improve accuracy. Lewis et al. [4] proposed the concept of Retrieval-Augmented Generation (RAG), where relevant information is retrieved from a knowledge base before generating responses. This approach reduces hallucinations and ensures that responses are grounded in factual data. The methodology adopted in this project aligns with this concept by retrieving portfolio-related data before generating responses.

Vector databases have gained popularity for storing and retrieving semantic representations of text. Qdrant [7] provides efficient storage and similarity search mechanisms for embeddings, enabling semantic search across large datasets. Vector-based retrieval allows systems to identify relevant content even when user queries are phrased differently, which is crucial for intelligent information retrieval in conversational systems.

Cloud-based AI services have simplified access to advanced NLP models. OpenAI's API [5] and OpenRouter [6] provide scalable interfaces for integrating LLMs into applications, allowing developers to build AI-powered systems without training models from scratch. These services have been widely adopted in chatbot systems, virtual assistants, and intelligent web applications.

On the frontend side, modern web frameworks play a critical role in delivering responsive and user-friendly interfaces. Next.js [8] and React [9] enable component-based development, fast rendering, and seamless integration with backend APIs. Tailwind CSS [10] further simplifies UI design by providing utility-based styling, resulting in clean and consistent layouts. These technologies have been widely used in AI-enabled web applications due to their flexibility and performance.

Version control and deployment platforms such as GitHub [11] and Vercel [12] support collaborative development and efficient deployment of web applications. They enable continuous integration, code management, and cloud hosting, making them suitable for research and production environments.

From the existing literature, it is evident that while significant work has been done in NLP, conversational AI, semantic search, and AI-powered web systems, limited research focuses on applying these technologies to personal portfolio websites. Most portfolios remain static and lack intelligent interaction. This project bridges this gap by integrating NLP, LLMs, and vector databases into a personal portfolio platform, providing an interactive and intelligent user experience.

3. Methodology

3.1. Design of Research

The research methodology adopted for this project follows a design and development–based research approach, focusing on the practical implementation and evaluation of an AI-enhanced web system. The methodology integrates software engineering principles with artificial intelligence and natural language processing techniques to design, develop, and validate an intelligent personal portfolio website

3.2. Information Gathering

Information gathering is a crucial initial phase in the development of the AI-Enhanced Personal Portfolio Website. This phase involves collecting all relevant data, requirements, and technical knowledge necessary for designing and implementing the system effectively.

In this project, information was gathered from multiple sources, including academic research papers, technical documentation, and online resources related to Artificial Intelligence, Natural Language Processing, conversational AI, and semantic search. Documentation from technologies such as OpenAI/OpenRouter, Qdrant, Next.js, and React was studied to understand their functionalities, integration methods, and best practices.

Additionally, portfolio-related data such as personal details, educational background, skills, projects, and achievements were collected manually. This data serves as the core knowledge base for the AI system. User requirements were also analyzed to identify the need for an interactive and intelligent portfolio interface rather than a traditional static website.

The information gathered during this phase helped define the problem statement, identify the research gap, select appropriate technologies, and design the system architecture. Overall, the information gathering process ensured that the project was built on accurate, relevant, and reliable data, leading to an effective and well-structured AI-enhanced portfolio system.

3.3. Architecture of the system

The architecture of the AI-enhanced personal portfolio website follows a layered design approach that integrates modern web technologies with artificial intelligence components. The system consists of a user interface layer developed using Next.js and React, which provides a responsive web interface and an interactive chat feature for user interaction. User queries are sent to a backend API layer implemented using Node.js, which manages request processing and communication between system components. The

AI and NLP layer utilizes Large Language Models accessed through OpenAI/OpenRouter APIs to understand natural language queries and generate responses. Portfolio data is stored as embeddings in a Qdrant vector database, enabling semantic similarity search for accurate information retrieval. The retrieved data is then used by the AI model to produce context-aware responses, which are returned to the frontend and displayed to the user. This modular architecture ensures scalability, maintainability, and efficient integration of AI-driven functionalities.

4. Design and Implementation

4.1. System Architecture:

The system architecture of the AI-enhanced personal portfolio website is designed using a layered and modular approach to ensure scalability, maintainability, and efficient integration of artificial intelligence components. The design separates the system into distinct layers, including the user interface, backend processing, AI and NLP services, and data storage, allowing each component to operate independently while maintaining seamless communication.

The user interface layer is implemented using Next.js and React, providing a responsive and interactive web interface. This layer includes portfolio sections and an AI-powered chat component that enables users to interact with the system using natural language queries. Tailwind CSS is used to create a clean and consistent user interface across different devices.

The backend layer is implemented using Node.js with Next.js API routes, which handle user requests, perform input validation, and manage communication with external AI services and the vector database. This layer acts as a controller that orchestrates data flow between the frontend and backend components.

The AI and NLP layer utilizes Large Language Models (LLMs) accessed through OpenAI/OpenRouter APIs to process user queries and generate intelligent, context-aware responses. User queries and portfolio data are converted into vector embeddings, enabling semantic understanding of the content.

The data storage layer uses Qdrant, a vector database, to store portfolio information as embeddings and perform semantic similarity searches. This enables accurate retrieval of relevant information even when user queries vary in wording.

Overall, the implementation integrates modern web technologies with AI-driven NLP techniques, resulting in an intelligent and interactive portfolio platform. The modular design supports future enhancements such as multi-user support, voice interaction, and advanced analytics while maintaining system efficiency and reliability.

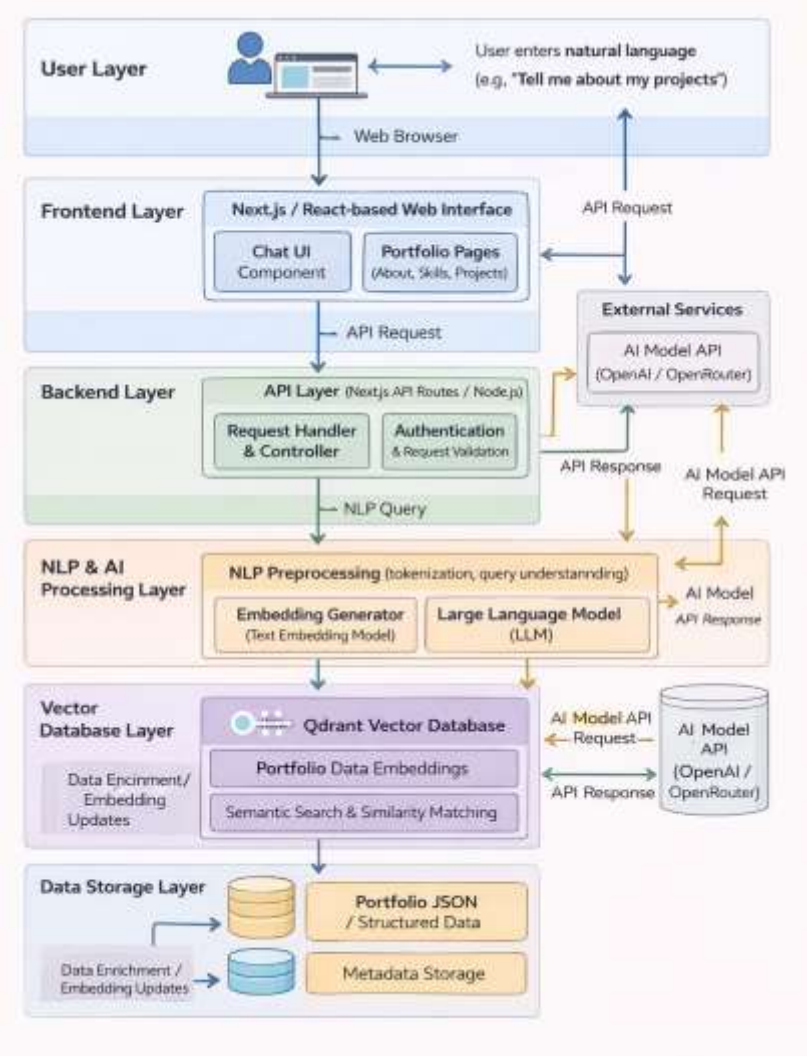
System WorkFlow

1. The user accesses the AI-enhanced personal portfolio website through a web browser and interacts with the system using the chat interface.
2. The user enters a natural language query related to portfolio information such as skills, projects, education, or experience.
3. The frontend interface, developed using Next.js and React, sends the user query to the backend API for processing.
4. The backend API receives the query, performs validation, and forwards it to the AI processing module.
5. The query is converted into a vector embedding using the text embedding model provided by the AI service.
6. A semantic similarity search is performed in the Qdrant vector database to retrieve relevant portfolio information based on the query embedding.

7. The retrieved portfolio data is provided as contextual input to the Large Language Model (LLM).
8. The LLM generates a context-aware and human-like response using the retrieved information.
9. The backend API sends the generated response back to the frontend.
10. The frontend displays the AI-generated response to the user in the chat interface.

System Architecture Diagram

Figure 1 System Architecture of AI Enhanced Personal Portfolio Website



4.2. Technologies Used

Technologies included in AI Enhanced Personal Portfolio Website is as follows:-

Table 1 Technology Stack for AI Enhanced Personal Portfolio Website

Category	Technology
Frontend Framework	Next.js (React)
Programming Language	JavaScript
Type Safety	TypeScript
Styling Framework	Tailwind CSS
Backend Runtime	Node.js
Backend Framework	Next.js API Routes

AI/NLP Engine	OpenAI / OpenRouter
Vector Database	Qdrant
Embedding Model	Text Embedding Models
Data Storage	JSON / Structured Files
Development Environment	Visual Studio Code
Version Control	Git and GitHub
Package Manager	npm

4.3. User Interface(UI)

The user interface of the AI-enhanced personal portfolio website is designed to be simple, responsive, and user-friendly. It provides a clean layout with well-structured sections for personal details, skills, projects, and contact information. A key feature of the interface is the interactive AI chatbot, which allows users to ask questions about the portfolio using natural language.

4.3.1. User Interface Overview

The user interface of the AI-enhanced personal portfolio website is designed to provide an intuitive, interactive, and visually appealing experience for users. The interface presents portfolio information such as personal profile, skills, projects, and achievements in a structured and easy-to-navigate layout. A prominent feature of the UI is the AI-powered chat interface, which allows visitors to interact with the portfolio using natural language queries. Built with Next.js and React, the interface ensures fast rendering and seamless navigation, while Tailwind CSS provides a modern, responsive design. The UI adapts efficiently to different screen sizes, ensuring accessibility across desktops, tablets, and mobile devices.

4.3.2. UI Screenshots

The following figures illustrates the key UI screens of AI Enhanced Personal Portfolio Website:

Figure 2 UI of AI Enhanced Personal Portfolio Website (ChatBot)

Figure 3 User asking question to the ChatBot (AI Enhanced Personal Portfolio Website).

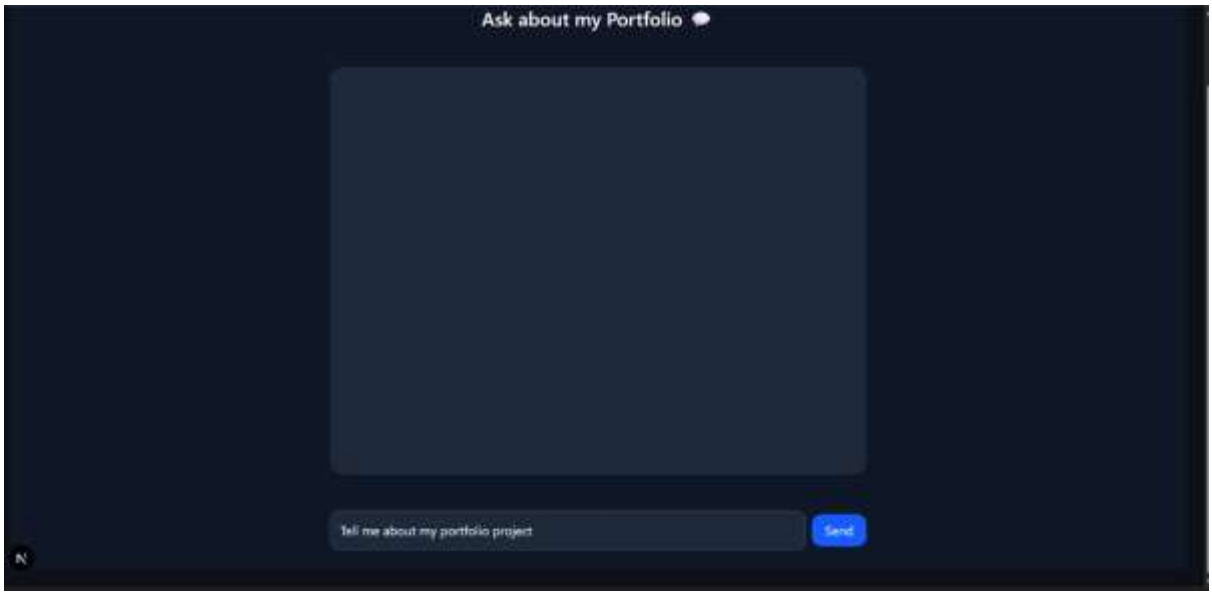


Figure 4 ChatBot collecting data from the Qdrant vector database and gives an AI generated reply / answer .

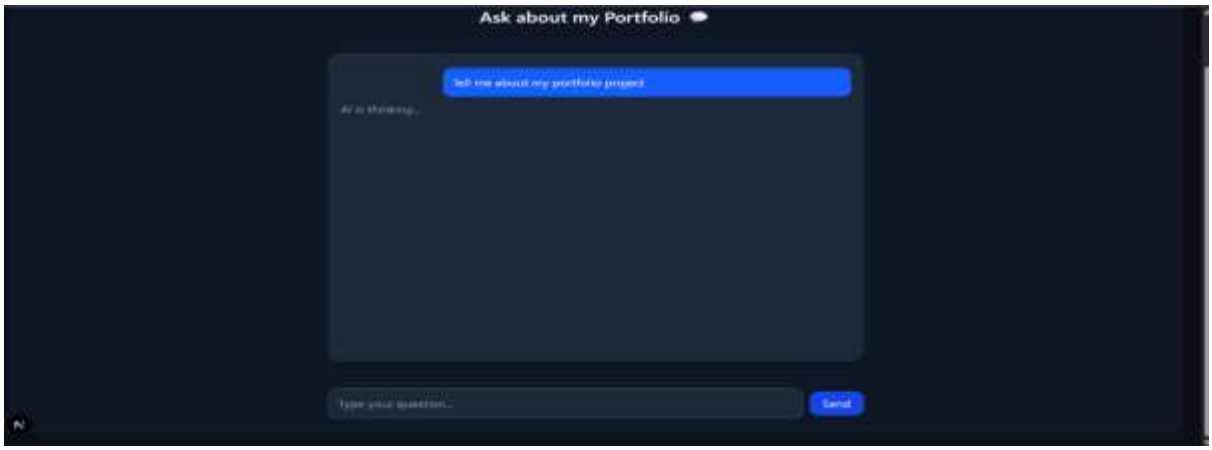
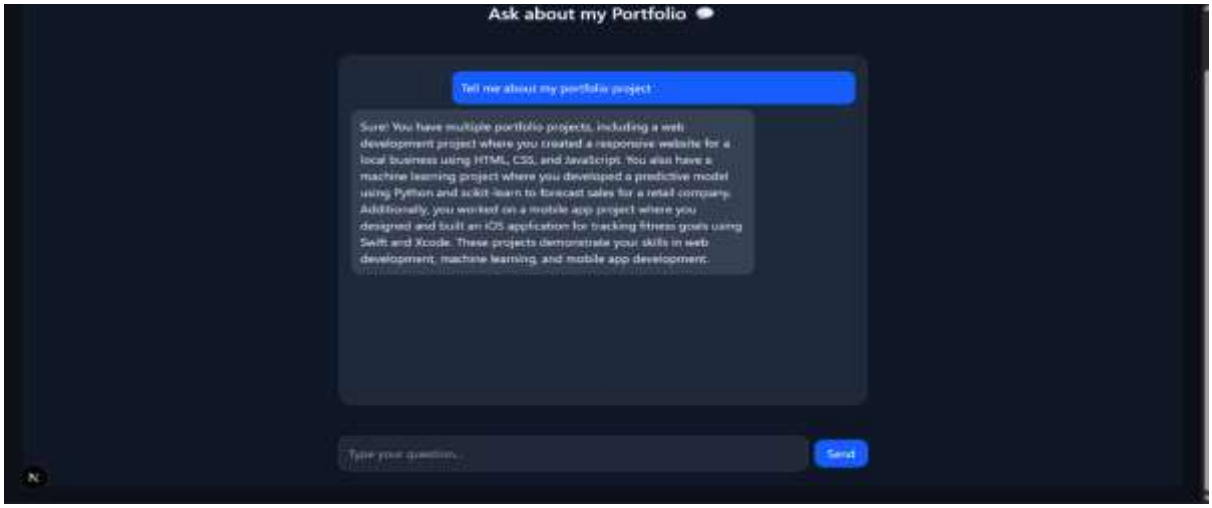


Figure 5 ChatBot (AI Enhanced Personal Portfolio Website) gives answer to the user's question .



5. Discussion

5.1. Strengths of the System

- **Interactive User Experience** : The system transforms a traditional static portfolio into an interactive platform by allowing users to ask questions and receive instant responses through an AI-powered chat interface.
- **Natural Language Understanding** : By utilizing Natural Language Processing (NLP), the system can understand user queries in everyday language, eliminating the need for keyword-based searching.
- **Context-Aware Responses** : The integration of Large Language Models (LLMs) enables the system to generate meaningful, accurate, and context-aware responses based on portfolio data.
- **Semantic Search Capability** : The use of a vector database (Qdrant) allows semantic similarity search, ensuring relevant information retrieval even when queries are phrased differently.
- **Modern Web Technologies** : Built using Next.js, React, and Tailwind CSS, the system offers fast performance, responsiveness, and a clean user interface.
- **Scalability and Extensibility** : The architecture allows easy addition of new features such as more portfolio content, multiple users, or advanced AI capabilities.
- **Improved Accessibility** : Users can easily access portfolio information without navigating multiple pages, making the system user-friendly for both technical and non-technical users.
- **Real-Time Interaction :** The chatbot provides instant responses, enhancing engagement and providing a smooth conversational experience.
- **Practical Application of AI :** The project demonstrates real-world implementation of AI, NLP, and vector search techniques in a web-based system.
- **Personalized Information Retrieval :** The system delivers responses specifically tailored to the portfolio owner's data, improving accuracy and relevance.

5.2. Limitations

- **Dependency on Internet Connectivity :** The system requires a stable internet connection to access AI services and the vector database, limiting offline usage.
- **Reliance on External AI APIs :** The performance and availability of the system depend on third-party AI services (such as OpenAI/OpenRouter), which may introduce latency or usage limits.
- **Limited Personalization Scope :** The system currently focuses on a single user's portfolio and does not support multiple user profiles or role-based access.
- **Initial Data Preparation Required :** Portfolio data must be preprocessed and converted into embeddings before effective semantic search can be performed**.**
- **AI Response Variability :** AI-generated responses may vary in phrasing and occasionally require manual validation to ensure accuracy.
- **Security and Privacy Concerns :** Since the system processes user queries and portfolio data through cloud-based services, data privacy and security need careful handling.

5.3. Future Enhancements

- **Multi-User Support :** Extend the system to support multiple portfolio owners with user authentication and personalized dashboards.
- **Voice-Based Interaction :** Integrate speech-to-text and text-to-speech features for voice-enabled interaction with the portfolio**.**

- **Advanced Analytics Dashboard :** Add analytics to track visitor interactions, frequently asked questions, and engagement metrics.
- **Offline Mode with Local Models :** Implement lightweight local AI models to enable limited offline functionality.
- **Enhanced Context Memory :** Improve conversational memory to maintain long-term context across multiple user interactions.
- **Improved UI and Visualization :** Add interactive charts, timelines, and project visualizations to enhance presentation quality.
- **Integration with Professional Platforms :** Connect the portfolio with LinkedIn, GitHub, and other professional platforms for real-time data updates.
- **Security Enhancements :** Implement advanced authentication, encryption, and access control mechanisms.

6. Conclusion

This project successfully demonstrates the design and implementation of an AI-Enhanced Personal Portfolio Website that integrates Natural Language Processing (NLP) and Artificial Intelligence (AI) to improve user interaction and information accessibility. Unlike traditional static portfolio websites, the proposed system provides an interactive chatbot interface that allows users to query portfolio information using natural language and receive accurate, context-aware responses.

By utilizing Large Language Models (LLMs) along with a vector database (Qdrant), the system effectively performs semantic search and intelligent retrieval of portfolio data. The use of modern web technologies such as Next.js, React, and Tailwind CSS ensures a responsive and user-friendly interface, while the backend architecture enables efficient communication between the frontend, AI services, and data storage layers.

The project highlights the practical application of AI and NLP techniques in real-world web development and demonstrates how intelligent systems can enhance personal branding and digital self-presentation. Overall, the system offers an innovative approach to portfolio design, improving accessibility, engagement, and usability, and serves as a strong foundation for future enhancements such as multi-user support, voice interaction, and advanced analytics.

7. References

1. D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson Education, 2023.
2. T. B. Brown et al., “Language models are few-shot learners,” Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.
3. A. Vaswani et al., “Attention is all you need,” in Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.
4. P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” Advances in Neural Information Processing Systems, vol. 33, 2020.
5. OpenAI, “OpenAI API Documentation,” 2024. [Online]. Available: https://platform.openai.com/docs
6. OpenRouter, “OpenRouter API Documentation,” 2024. [Online]. Available: https://openrouter.ai/docs
7. Qdrant Team, “Qdrant: Vector Database for AI Applications,” 2024. [Online]. Available: https://qdrant.tech/documentation/
8. Next.js Team, “Next.js Documentation,” 2024. [Online]. Available: https://nextjs.org/docs

9. React Team, “React: A JavaScript Library for Building User Interfaces,” 2024. [Online]. Available: https://react.dev/
10. Tailwind Labs, “Tailwind CSS Documentation,” 2024. [Online]. Available: https://tailwindcss.com/docs
11. GitHub, “GitHub Documentation,” 2024. [Online]. Available: https://docs.github.com
12. Vercel, “Vercel Platform Documentation,” 2024. [Online]. Available: https://vercel.com/docs