Bijon Setyawan Raya

bijonsetyawan@gmail.com GitHub · https://bsraya.com

Experience

Intermediate Fullstack Developer October 2022 – March 2024
Faria Education Group *Taipei, Taiwan*

- Optimized file export processes, achieving a significant reduction in export time by 98.33%.
- Sped up user feed search processes using ElasticSearch and efficient SQL queries.
- Enhanced search precision and user navigation within user feeds by expanding options in the feed search navigation modal, significantly improving user experience and content discovery.
- Consistently managed and ensured data integrity across all tables within the MySQL database.
- Refactored the jQuery codebase and converted some into Stimulus.js and Vue.js.

Graduate Research Scientist February 2021 – January 2023
National Tsing Hua University *Hsinchu, Taiwan*

Teaching Assistant Sep. 2020 – January 2023
National Tsing Hua University *Hsinchu, Taiwan*

Technical Skills

Languages: C/C++, Python, JavaScript/TypeScript.
Web Development: Django, FastAPI, Next.js.
Deep Learning & Machine Learning: Horovod, Keras, PySpark, PyTorch, TensorFlow, Numpy, Pandas, Scipy, Scikit Learn.
Database: DuckDB, ElasticSearch, MySQL, SQLite, Weaviate.
Utilities: Docker, Kubernetes, Git, Linux/Unix.

Education

National Tsing Hua University Hsinchu, Taiwan
Master of Science in Computer Science *January 2023*

National Tsing Hua University Hsinchu, Taiwan
Bachelor of Science in Computer Science *January 2021*

Projects

Retrieval-Augmented Generation System for Books | *Gradio, Llama 3.1, Llama Index, Python, and Weaviate.*

- **Book Search Engine and Summarizer:** Created a RAG system that assists readers in discovering new books, while also providing users with factual information about books, along with concise summaries.
- **Improved Retrieval Efficiency:** Optimized retrieval processes speed by leveraging Weaviate's vector embedding database, allowing for faster and more accurate search results.
- **Enhanced Search Results:** Improved search results with enhanced metadata-driven retrieval capabilities, enabling users to find relevant books based on specific criteria.
- **User-Centric Interface:** Crafted an intuitive UI that enables users to input custom prompts and manage their responses and document access with precision.

Image Search Engine | *DuckDB, FastAPI, Jinja, Python, PyTorch, and SQLite.*

- **Visual Search Capabilities:** Developed a robust image search engine that utilizes ResNet to identify and retrieve similar images based on their visual features, streamlining the image upload process.
- **User-Centric Interface:** Crafted an intuitive UI for uploading images and showcasing similar content in a visually appealing way, enhancing the user experience.
- **Database Integration and Optimizations:** Integrated DuckDB and reduced vector dimensions using Principal Component Analysis (PCA), resulting in improved search speed, reduced memory usage, and optimized performance.

Music Recommendation System | *Numpy, Pandas, Python, and scikit-learn.*

- **Personalized Recommendations:** Leveraged Spotify's vast music library to extract and process audio features from over 2,000 songs across various genres. Applied Non-negative Matrix Factorization (NMF) to analyze user preferences and provide tailored song recommendations, enhancing the music discovery experience.

AI Coding Assistant | *Llama 3, Ollama, and Python WASM.*

- **Collaborative Coding Experience:** Developed an innovative coding assistant that enables users to write and run Python code directly in their browser. Integrated a Large Language Model (LLM) to provide real-time assistance with coding tasks, debugging, and more, enhancing productivity and accuracy.

Schedulearn | *Docker, FastAPI, Horovod, Python, and SQLite.*

- **Scalable Deep Learning Scheduling:** Designed a highly scalable deep learning scheduling system that utilizes microservices architecture to optimize resource allocation and dynamically scale compute resources in response to changing workload demands.
- **Improved Efficiency:** Leveraged the system to reduce overall makespan and turnaround time by 10%-50% across various frameworks (PyTorch, TensorFlow, MXNet), leading to improved efficiency and productivity.
- **Optimized System Performance:** Transformed the codebase by cutting its size by 50%, improving requirement utilization by 70%, and resulting in significant performance gains and simplified maintenance.