Feiran (Philip) Huang

philiphuang439@gmail.com - philipfhuang.github.io - linkedin.com/in/feiran-huang - github.com/philipfhuang

EDUCATION

University of Toronto **Toronto, Canada**

Honors Bachelor of Science - Computer Science Specialist *September 2020 - June 2024*

- **CGPA:** 3.46
- **Relevant courses:** Algorithm Design, Data Structures, Scalable Computing, Software Engineering, Databases, Web Programming, Operating Systems, Systems Programming, Computer Security, Artificial Intelligence, Machine Learning, Deep Learning, Neural Network, Machine Vision, Advanced LLM and Distributed Systems for Personalized AI, Comparative and Empirical Analysis of LLMs versus SLMs, Robotics

TECHNICAL SKILLS

Programming Languages: C, C++, Python, Java, PHP, SQL, HTML, CSS, JavaSript, TypeScript, R, Assembly, RISC-V
Frameworks and Libraries: React, Next.js, Node.js, JQuery, Bootstrap, Tailwind CSS, Flask, Django, requests, Numpy, Pandas, PyTorch
Tools: AWS, Git, Docker, Nginx, Node.js, Redis, Cassandra, Shell, Gazebo, Bootstrap, Figma

WORK EXPERIENCE

Teaching Assistant

University of Toronto Mississauga, Mississauga, Canada *January 2024 - April 2024*

- Planned and conducted weekly laboratory sessions following course curriculum and objectives.
- Collaborated with course instructors to develop laboratory materials.
- Graded assignments, exams, and projects in a timely and fair manner, providing constructive feedback to students to aid in their learning process.
- Managed discussion board to address students' questions, clarify doubts, and provide additional assistance outside of class time.
- Contributed to creating weekly posts of current events that related to databases, fostering a deeper understanding of real-world applications and implications of database technologies among students.

PUBLICATIONS

- Suqing Liu, Zezhu Yu, Feiran Huang, Yousef Bulbulia, Andreas Bergen, and Michael Liut. 2024. **Can Small Language Models With Retrieval-Augmented Generation Replace Large Language Models When Learning Computer Science?** In Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), ACM, New York, NY, USA, 7 https://doi.org/10.1145/3649217.3653554

RESEARCH

Advanced LLM and Distributed Systems **University of Toronto**

Researcher *September 2023 - January 2024*

- Develop and utilize small language models in conjunction with Retrieve and Generate (RAG) architecture to construct personalized AI systems.
- Explored innovative approaches to leverage small language models within the context of the Retrieve and Generate (RAG) architecture, aiming to enhance the effectiveness and personalization of AI systems.

Comparative and Empirical Analysis of LLMs versus SLMs **University of Toronto**

Researcher *January 2024 - April 2024*

- Conduct comparative and empirical analysis of large language models (LLMs) and small language models (SLMs) to evaluate their performance, efficiency, and suitability for various natural language processing tasks.
- Leverage conversations from 300-level and 400-level courses, where students were provided with AI Assistants power by different models,and evaluated the performance, efficiency.
- Published research findings in ITiCSE 2024.

PROJECTS

AI Assistant ai-assistant.utm.utoronto.ca
Developer *September 2023 - Present*

- An AI Chatbot utilizing the Retrieval-Augmented Generation (RAG) method, designed to function as a virtual assistant. It is tailored to address users' individual needs and queries. Specifically implemented within 300-level and 400-level Computer Science courses at the University of Toronto, it serves as an adjunct teaching assistant, providing support and guidance to students.

PlayPal github.com/philipfhuang/PlayPal-Frontend
Team Leader, Backend Manager, Backend Developer *January 2024 - April 2024*

- A web application which facilitates individuals finding people to play sports together, simplifies the process of finding sports partners and organizing sports activities and fostering a community of sports enthusiasts.
- Frontend: Employed Next.js + Material UI to craft interactive and responsive user interfaces.
- Backend: Leveraged Django REST framework for robust backend development, ensuring efficient RESTful API creation and management. The Django ORM was used to seamlessly interact with the database. Employed Django ORM to model and manage database entities, ensuring efficient data manipulation and storage.
- Authentication: Implemented JWT for secure user authentication.
- Deployment: Utilized Continuous Integration/Continuous Deployment (CI/CD) practices to automate the deployment process, ensuring seamless updates and maintenance.
- Integration: Integrated PayPal API to facilitate secure and convenient payment transactions within the platform, enhancing user satisfaction and trust.

ThreeMusketeers github.com/philipfhuang/ThreeMusketeers
Team Leader, Developer *September 2021 - December 2021*

- Implemented a Java-based version of the classic board game ThreeMusketeers, incorporating modern design principles and advanced features to enhance gameplay experience. Collaborated with a team of developers to design, develop, and deploy the project.
- Utilized object-oriented programming principles to design a scalable and modular codebase, ensuring flexibility and ease of maintenance.
- Added new features including player vs player (locally/remotely), player vs greedy agent mode, timer functionality, custom board loading, and difficulty adjustment, improving overall user engagement and satisfaction.

Easy Chef github.com/philipfhuang/EasyChef
Team Leader, Full Stack Developer *January 2023 - April 2023*

- Developed Easy Chef, a dynamic social media platform enabling users to share and discover recipes effortlessly. Leveraged React.js for the frontend and Django Rest Framework for the backend to ensure seamless user experience and robust functionality.
- Frontend: Employed Next.js + Semi Design & Ant Design to craft interactive and responsive user interfaces.
- Backend: Leveraged Django REST framework for robust backend development, ensuring efficient RESTful API creation and management. The Django ORM was used to interact with the database. Employed Django ORM to model and manage database entities, ensuring efficient data manipulation and storage.
- Authentication: Implemented JWT for secure user authentication.

r/place github.com/philipfhuang/rplace-AWS
Developer *November 2023 - December 2023*

- A replica of Reddit's "r/place" using resources provided by Amazon Web Services (AWS).
- Orchestrated infrastructure as code (IaC) using CloudFormation, streamlining the deployment and management of AWS resources, and fostering automation.
- Employed Redis clusters and CDN (Content Delivery Network) for efficient state caching, coupled with WebSockets to broadcast real-time updates to clients, ensuring a seamless user experience.
- Achieved high scalability and concurrency by abstracting components into serverless functions using AWS Lambda, optimizing resource utilization and performance.
- Employed DynamoDB, a NoSQL database service, for efficient data storage and retrieval. Utilized its flexible schema and seamless scalability to handle dynamic data requirements and high throughput, ensuring optimal performance under varying workloads.

Distributed URL Shortener System github.com/philipfhuang/Distributed-URL-Shortener-System
Team Lead, System Architect, Developer *Sept 2023 - Oct 2023*

- Led development of a distributed URL shortener system, employing Docker containers for easy deployment and Docker Swarm for scalability.
- Implemented disaster recovery mechanism for real-time system resilience, ensuring minimal downtime.
- Engineered scalable architecture using Docker Swarm for automatic scaling.
- Designed fault-tolerant architecture, ensuring continuous availability and data integrity.

mysh github.com/philipfhuang/mysh

Developer *January 2022 - April 2022*

- A robust Unix shell that offers a seamless command-line interface for Unix-like operating systems.

Simon - RISC-V github.com/philipfhuang/Simon-RISC-V

Developer *March 2022 - April 2022*

- RISC-V Assembly Programming: Utilized RISC-V Assembly for writing the game code, ensuring efficient execution on a RISC-V architecture.
- Hardware Simulation: Configured and used Ripes, a graphical RISC-V computer simulator, to run and test the game.
- Troubleshooting: Debugged and resolved issues related to game execution and user inputs.