Chen Wei

47 Nicholas Brown Yards, Providence, RI, 02904

☎ 617-888-1495 ✉ chen_wei@alumni.brown.edu linkedin.com/in/chen-wei-57b225198/ github portfolio

Education

Brown University **Sep. 2022 – May 2024**
Master of Science in Computer Science *GPA: 4.0 / 4.0*

Brandeis University **Sep. 2019 – May 2022**
Bachelor of Science in Computer Science and Applied Mathematics (Magna cum Laude) *GPA: 3.789 / 4.0*

Skills

Languages: Python, Java, C, C++, SQL, JavaScript, HTML, CSS, Shell, TypeScript, R, MATLAB, Go, Latex

Frameworks & Tools: Docker, Express, MongoDB, AWS, React, Git, GraphQL, Node.js, Websocket, FastAPI, RAG, LLM, Heroku, Wireshark, Linux, Bootstrap, Next.js, Spring, Nginx, Numpy, Scikit-learn, PyTorch, Tensorflow, Spark, BeautifulSoup, NLP, DL, Django, REST API, Hbase, Redis, Cursor, etc.

Work Experience

AMA Career **Apr. 2025 - Present**
Full Stack Engineer *Remote*

- Developed a real-time job application automation platform using **FastAPI**, **React**, and **TypeScript**, supporting 3 core services: job referral, auto-apply, and interview preparation, improving job submission efficiency by 60% for early users.
- Designed and implemented **RESTful APIs** using **FastAPI** to handle CRUD operations across user, task, and job data stored in Supabase; optimized performance with pagination, indexing, and response caching, reducing average API latency by 40%.
- Architected **Supabase** integration for real-time data storage of task states, and job updates; optimized read/write throughput with row-level indexing and selective fetching; implemented an **event-driven** Supabase Edge Function triggered on **PostgreSQL** insert to decouple onboarding from backend load.
- Built a scalable **WebSocket** framework with **FastAPI** to enable real-time syncing of task states, and profile edits across modules; optimized with connection pooling and auto-reconnect logic, reducing redundant network traffic by 40%.
- Implemented **async** batch sending for **Twilio SMS** to deliver task updates, frontend **lazy loading**, and (de)serialization layers to streamline cross-module communication, reduce UI rendering latency by 35%, and ensure scalable notification handling.

Blyss **Jan. 2025 - Apr. 2025**
Full Stack Engineer *Remote*

- Designed a social media platform, enabling real-time chat, image/video sharing, and personalized feeds; built with **Node.js microservices**, and implemented **Hystrix**-style circuit breakers to isolate failures, improving system resilience by 35% under simulated peak load.
- Developed **RESTful APIs** for user sessions, media, and post management, and implemented **gRPC** with **Protocol Buffers** for high-throughput chat, notifications, and event sync between microservices, supporting 3K+ concurrent messages with sub-100ms delivery latency.
- Architected **PostgreSQL** hosted on **AWS RDS** with **replicas**; normalized schemas for users, posts, events, and messages; optimized queries with B-tree indexing, implemented read/write separation with master-slave replication.
- Utilized **RabbitMQ** for async task processing with prioritized queues, DLX, and prefetch tuning to optimize media handling, notifications, and bulk messaging with high throughput and reliable retries.
- Utilized **Redis** for session caching, pub/sub chat delivery, and notification buffering; optimized with **TTL**-based cache invalidation, **Lua scripting** for atomic operations, and Redis **Cluster** for horizontal scalability.
- Built **CI/CD** pipelines using **Docker**, **GitHub Actions**, and **Kubernetes** for automated testing and deployment of microservices; Configured **Nginx** as a gateway and **load balancer** to route REST and gRPC traffic across services.

GreenPlatter **Jun. 2024 - Mar. 2025**
Full Stack Engineer *Remote*

- Built a serverless AI-powered recipe generation platform using **AWS Amplify** and **Amazon Bedrock** with Claude 3 Sonnet, delivering personalized meal plans for diabetic and health-conscious users; hosted the frontend on a global **CDN** for low-latency access.
- Implemented **GraphQL APIs** with custom input types , optimized resolvers with query batching to efficiently retrieve 10K curated recipes stored in **DynamoDB**.
- Designed a scalable data model in **DynamoDB** using sparse indexes and enabled **Auto Scaling**; integrated **AWS Lambda** functions to process ingredient inputs and post-signup flows.
- Secured user access with **JWT**-based email OTP authentication using **Amazon Cognito** and 2FA logic.
- Applied **prompt engineering** with few-shot learning in Claude 3 Sonnet to dynamically retrieve and generate recipes aligned to user health profiles, dietary restrictions, and culinary skill level.

Goldman Sachs **Jun. 2023 - Aug. 2023**
Summer Analyst Intern *New York City, New York*

- Optimized complex **SQL** analytics pipelines and implemented multithreaded processing to accelerate risk calculations in **SecDB**, reducing end-to-end query latency by 60% for daily options trading reports.
- Developed risk management models using **XGBoost** and **logistic regression** to assess stress exposure under varying market conditions; conducted sensitivity analysis and achieved AUC-ROC of 0.91 and precision-recall of 0.87.

Research Experience & Projects

Youtube Teller (Video - Captioning) | *Transformer, Teacher forcing* | [github] & [devpost] **Nov. 2022**

- Built a **video captioning multimodal** pipeline for YouTube clips, including data preprocessing (frame extraction, sequential embedding), model training and data visualization using **NumPy**, **Pytorch** and **Seaborn** with a **Transformer** Mapper + GPT-2 architecture.
- Applied **prompt**-based prefix tuning and few-shot learning with **CLIP** visual embeddings and GPT-2 language modeling, achieving BLEU-4 score of 0.52 and METEOR score of 0.30 on held-out video caption tasks.

Brandeis University Computer Vision Research Group | *PyTorch, Tensorflow, GAN* **Jul. 2021 - Aug. 2022**

- Developed a **multi-modal** cGAN for image-sound transformation using TensorFlow and PyTorch; pretrained **encoder-decoder** architecture to extract image features.
- **Multithreaded web scraped** over 5,000 YouTube videos and Google images utilizing **Requests** and **Selenium**.
- Performed **feature engineering** and visualized training metrics using **NumPy**, **Pandas**, **Matplotlib**, and **Seaborn**.