**Ruiping Fang**

r6fang@ucsd.edu | (858) 399-1727 | https://github.com/ruiping-fang |

**EDUCATION**

**University of California San Diego**

**BS, Computer Science Expected graduation date: June 15th, 2025**

**GPA**: 3.56

**Course work:**

**Core**: Advanced Data Structures, Design and Analysis of Algorithms, Software Engineering. **Database**:Database System Principles

**System**: Computer Architecture, Parallel Computing, Components and Design Techniques for Digital Systems.

**SKILLS**

* Programming Languages: Python, Java, JavaScript, SQL, C++
* Frameworks: Flask, React

**INTERNSHIP**

**Qualcomm Institute - UC San Diego Division of Calit2 2023.1 - 2023.3**

* Implemented an Unreal Engine 5 virtual reality project on archaeological visualization using C++ based triggers.
* \*Engineered a data conversion pipeline to parse KML/ML files into JSON format\*

**RESEARCH**

**Computer Science and Engineering Early Research Scholar Program (CSE-ERSP)**

**University of California San Diego, CA**  **2023.9 - 2024.6**

**PROJECTS**

**Fanguage -** https://fangyan.fangruiping.com

An interactive web application that leverages the OpenAI services to provide independent access to GPT4

* Developed and deployed a Python-based interface which connects with OpenAI services, enabling independent access to LLMs
* Implemented a responsive frontend which featured **React.js** user input and frame-view components
* Hosted on personalized domain **fangyan.fangruiping.com**.

consider replacing with the calendar project.

State what you did and what others did:

“Implemented the calendar view of xx project within a group of 10 developers where others focused on yy”

“Instantiated zz react.js components which accomplished some functionality”

“Explain some design decisions on why it is the best for uses cases or some alternatives that you did not pick over the current design; frameworks I.E why MongoDB instead of CouchDB or reverse; why d3.js instead of react.js”

4-5 points.

Java-based OOP: abstraction, inheritance, polymorphism. Interface, service classes, contracts. (Back-end); JUnit - mock-based

Front-end React/Vue/D3.js/HTML/CSS , Stem/leaf components, Redux Store, React hooks; Jest/JSUnit - mock-based

Connecting server-side and client-side: RESTful API, RTKQueries

OS/DS:

Distributed Systems != Cloud Computing Applications

DS: Membership List, Message Passing, Replica Management, Paxos/Raft based protocols, Failure Detection

OS: Context-switching, Resource Optimization, Memory-mapping