# Bo Peng Frontend Engineer

## Experience

**Amazon AWS | Frontend Engineer II**

➔ Developed New Reader Experience Topics in QuickSight for a better Q&A experience for QuickSight users
➔ Utilized React and Typescript to build QuickSight Scenarios for thousands of current QuickSight Pro users
➔ Wrote end to end tests using PlayWright to ensure full coverage and intended user experience for Scenarios
➔ Implemented node-based editor canvases on the UI using React Flow for Scenarios, giving customers a reliable and streamlined experience when viewing their data
➔ Integrated new GenAI features with frontend, allowing users to leverage large language models with Scenarios

**Fanbyte | Junior Software Engineer**

➔ Developed microservices as npm modules to help aid the ad service platform for a better ad experience
➔ Utilized mocha testing framework to integrate meaningful tests to ensure microservices function properly
➔ Polished vue ad service front-end to integrate with the new modular npm services to better gather web traffic
➔ Constructed new api endpoints to retrieve publisher revenue data for our renovated vue dashboard service
➔ Created new wordpress theme with vue and wordpress components for complete redesign of Fanbyte website

**ManageGo | Software Developer**

➔ Implemented new website designs in React for software applications, improving user experiences for thousands
➔ Created new custom React hooks using React history module to manipulate query strings when handling API calls and state management with Redux and Redux Saga for the utmost dynamic front-end performance
➔ Designed and finalized an API schema on a new commercial application feature/software for concierge users
➔ Utilized styled-components to design multiple React components to ensure optimization, stability, and consistent universal styling throughout the entire application structure and user interface
➔ Wrote polished tests on redux actions, reducers, and sagas to ensure proper functionality in application state

**Nova | Software Engineer**

➔ Combined React's dynamic rendering and D3 to generate a continuously animated interactive data-driven SVG representation of GraphQL schemas that are capable of persisting throughout the application life cycle
➔ Architected custom React Hooks to eliminate the need for lifecycle methods and shifting the Class paradigm into a more organized functional architecture by isolating functionality into multiple reusable methods
➔ Implemented Redis cache to handle high volumes of incoming GraphQL endpoint fetch requests which minimizes the number of asynchronous calls resulting in reduced latency and a streamlined UX
➔ Deployed Node.js application as a multi-container docker application, isolating Node.js environment and Redis images, resulting in a modularized and standardized service infrastructure across all platforms
➔ Implemented React Router to enable client-side dynamic routing which reduced server calls and produced seamless view transitions between multiple documentation pages across the application infrastructure
➔ Developed a breadth-first search algorithm to produce a specific color pattern based on the schema's correlation with one another to display a clear understanding and overall visual structure of the nodes

## Education

**New York University |** BA in **Computer Science** with a minor in **Economics** and **Web Design**

**Relevant Coursework**

Data Structures, Basic Algorithms, Computer System Organization, Operating Systems, Applied Internet Technology Discrete Mathematics, Web Development and Programming, iOS Development, Natural Language Processing

## Technical Skills

**Experienced**: React, React hooks, Typescript, Jest, PlayWright, React Query, Redux/Saga, CSS, Git

**Worked with:** GraphQL, Node, Java, Python