GAURANG JOTWANI

217-648-8657 jotwani.g@northeastern.edu San Jose, CA (Open to Relocation)
Linkedin Github Medium Blogs Portfolio

EDUCATION

Northeastern University, Khoury College of Computer Science Expected Graduation: **May 2024**
Master of Science, **Computer Science**, **GPA: 4.00 / 4.00**
Key Courses: Operating Systems, Database Systems, Distributed Systems, Algorithms and Data Structures, Object Oriented Design, Web Development, Augmented Reality

University of Illinois Urbana-Champaign Aug 2015 - May 2019
Bachelor of Science in Chemical Engineering (Bio-molecular Eng Specialization), GPA: 3.83/4.0

[AWS Certified Developer]

[AWS Certified Solutions Architect]

TECHNICAL SKILLS

- ✅ **Programming Languages:** C++, C, C#, Python, Java, JavaScript, R
- ✅ **Frameworks:** Django Rest Framework, NodeJS, React, Unity
- ✅ **AWS Cloud Technologies:** DynamoDB, Lambda, API Gateway, SQL, S3, Kinesis, SNS, KMS, RDS, Code Pipeline
- ✅ **Other Technologies:** HTML, CSS, Jenkins, Kubernetes, jQuery, Git, CI / CD, Docker, Bootstrap, Selenium
- ✅ **DBMS:** MySQL, Postgres, MongoDB (NoSQL), DynamoDB (NoSQL)

WORK EXPERIENCE

Northeastern University Jan 2023 - Present
GRADUATE TEACHING ASSISTANT

- ➢ Graduate TA for the Machine Learning, Computer Systems and Algorithms Graduate Courses
- ➢ Checked assignments, proctored tests and provided grades according to university standards for over 100 students

IALM Feb 2021 - Aug 2022
SOFTWARE DEVELOPER INTERN (BACK-END)

- ➢ Led backend development for IALM, managing a daily user base of 15,000+, utilizing AWS services (DynamoDB, S3, CloudFront) to ensure seamless scalability and 99.9% uptime
- ➢ Collaborated on Spring Boot microservices architecture, boosting app performance by 40%, enabling efficient cross-team collaboration, and timely feature delivery
- ➢ Developed and maintained Java-based backend code adhering to coding standards, supporting 15,000+ daily users with rapid response times
- ➢ Managed MySQL backend database, optimizing data processes for rapid access, achieving <2s average page load time for thousands of concurrent users
- ➢ Engineered APIs integrating frontend interfaces, contributing to 23% increase in user engagement and session duration.
- ➢ Skillfully administered AWS services, configuring DynamoDB for peak performance, managing S3 media assets, serving a user base of 15,000+
- ➢ Collaborated with cross-functional teams, leading to a 95% customer satisfaction rate, seamlessly integrating features.
- ➢ Key contributor in agile methodologies, delivering high-quality code for a 17% increase in daily active users
- ➢ Proactively resolved bottlenecks, optimizing backend for enhanced responsiveness, reducing bounce rate by 12%
- ➢ Documented architecture, APIs, and processes, fostering knowledge sharing, fueling 30% growth in user registrations

Pattern Jobs Jan 2020 - May 2020
SOFTWARE DEVELOPER INTERN (FRONT-END)

- ➢ Successfully delivered 10 new app features, meeting 95% of pressing deadlines in a fast-paced startup environment
- ➢ Led 15 member team discussions with Engineers and Product Managers, resulting in the design and implementation of 8 high-performance user experiences and the creation of robust, quality, and scalable React components for new products
- ➢ Refactored 30 classical components into improved functional components, reducing code complexity by 40% and improving maintainability
- ➢ Proactively created Python scripts to automate the migration of a Django database to a production-ready MySQL Database, reducing migration time by 50% and minimizing the risk of human error
- ➢ Enhanced app security by improving the Login/Sign-up feature, resulting in a 30% reduction in security vulnerabilities and integrating Phone authentication, increasing user data protection
- ➢ Developed and maintained responsive user interfaces using ReactJS, HTML, and CSS, achieving a 25% faster page load time and improving the overall user experience

- Actively participated in Agile (SCRUM) SDLC methodologies, contributing to 12 successful feature implementations and a 15% improvement in existing ones. Utilized ReactJS components and libraries, leading to a 20% increase in codebase efficiency and maintainability
- Conducted comprehensive testing of user interface components and features using tools such as Jest and Enzyme, resulting in a 90% reduction in post-release bug reports and ensuring high code quality
- Identified and resolved 20 UI performance bottlenecks, implemented lazy loading and code splitting techniques, and integrated best practices for optimizing ReactJS applications, resulting in a 35% increase in application speed
- Contributed to design discussions and decisions, providing input on user interface layout, color schemes, and visual design elements. Achieved a cohesive and consistent user experience across the application, resulting in positive user feedback
- Demonstrated excellent problem-solving skills by resolving 15 complex issues related to user interface behavior, performance, and compatibility across multiple browsers and devices
- Kept up to date with the latest developments in front-end technologies and trends, dedicating 10 hours per week to research, experimentation, and self-study to expand knowledge and skillset

TECHNICAL PROJECTS

Implemented Virtual Memory in C
- Spearheaded the creation of a virtual memory manager employing paging and the LRU page replacement algorithm, facilitating memory allocation, deallocation, and data manipulation
- Implemented mutex locks to safeguard against race conditions, guaranteeing synchronized memory access for multiple concurrent threads and bolstering system stability
- Developed a robust testing suite, including diverse test functions, to validate the memory manager's functionality and performance, covering allocation, deallocation, data operations, and multi-threaded support

Profile API with Django Rest Framework
- Created a fully functioning REST API that can handle creating and updating user profiles, login and authentication, posting status updates and viewing status update feeds
- Built a REST API from scratch using Django, Django Rest Framework, Vagrant, VirtualBox, Atom and deployed the API to AWS dev server

Recipe App API with Django Rest Framework
- Built a back-end REST API using Python, Django, Django REST Framework, Docker, GitHub, Postgres and Test Driven Development
- Created a fully functioning Rest API that can handle User Authentication, Creating Recipe Objects, Filtering and Sorting Recipes and Uploading and Viewing Images

Google Maps Store Locator with React JS, Google Maps API and Yelp API
- Built a visually stunning Store Locator Web App using React by integrating Yelp API and Google Maps API
- Added stylized Input box, Store Containers and a beautiful transition on the hover of the individual store

Budget Management App with React JS
- Built a responsive, visually appealing Budget Management App where users can perform CRUD actions such as adding / updating / deleting / search / filter an expense
- Performed authentication with Google Account and Integrated Firebase Database

Augmented Reality Spacebook in Unity & C#
- Designed and developed the Augmented Reality Space Book using Unity and C#, incorporating Vuforia for image recognition
- Enhanced user accessibility by implementing an audio feature, enabling users to listen to content
- Created three interactive AR experiences, including an Alien Attack Game, Mars Rover Simulation, and Solar System Portal, enriching the educational value of the project

Distributed Key-Value Store with Paxos Consensus Algorithm in Java
- Implemented Robust Distributed Key-Value Store: Orchestrated the development of a distributed key-value store employing the Paxos consensus algorithm, featuring 9 replicas, and efficiently managing over 10,000 commands while maintaining a 5% simulated node failure rate
- Managed Scalable System Load: Controlled system stability by limiting concurrent proposers to 3, effectively averting potential deadlocks and ensuring the smooth operation of the Paxos algorithm
- Demonstrated Fault Tolerance and Recovery: Illustrated resilience through the simulation of failures in acceptor and learner nodes, showcasing how the system persists in reaching consensus even in the presence of node failures and temporary inconsistencies