Edan Bash

edanbash@berkeley.edu | (323)-806-3872 | Berkeley, CA | linkedin.com/in/edan-bash | github.com/edanbash

EDUCATION

University of California, Berkeley *May 2023*
Bachelor of Science in Electrical Engineering and Computer Science **GPA: 3.9**

SKILLS

Strong: AWS | Java | Python | Kotlin | Typescript | C | Git | CI/CD | OOP | DevOps
Proficient: REST APIs | ReactJS | MySQL | NoSQL | Golang | Docker | Kubernetes
Non-Technical: Product Design | Documentation Writing | Customer Obsession | Data Analysis

AWARDS

Regents' & Chancellor's Scholar • Amazon Future Engineer Scholar • Cal Leadership Award

PROFESSIONAL EXPERIENCE

Lime Fintech - Backend Engineer • Jan 2023 - May 2023 • *Berkeley, CA*

- Maintained a robust financial web application using the open-source Laravel framework, ensuring efficient data processing, secure user authentication, reliable task-processing, and seamless frontend-backend integration
- Orchestrated the deployment to production on AWS Elastic Beanstalk through a streamlined CI/CD pipeline
- Designed automation tool to pull user files from S3, parse them in Lambda, and reliably store data in RDS
- Used Docker Compose to manage MySQL database and PHP server for local development and testing
- Produced comprehensive documentation for the codebase and cloud architecture, ensuring seamless knowledge transfer

Amazon - Software Dev Intern
May 2022 - August 2022 - *New York, New York*

- Deployed a distributed microservice on AWS to automate the transcoding of thousands of advertiser videos daily
- Designed and implemented innovative API endpoints using the OpenAPI specification
- Spearheaded design reviews with senior engineers and drove codebase refactoring across different teams
- Successfully rectified 15% of failed submissions, resulting in increased monthly ad campaigns

May 2021 - August 2021 - *Seattle, Washington*

- Created web application using AWS and ReactJS to increase data access and automate customer list generation for PMs
- Utilized StepFunctions to manage workflows, using Lambdas as compute, and storing metadata in DynamoDB
- Conducted in-depth stakeholder meetings, translating feedback into actionable design concepts
- Established CloudWatch logs for application performance monitoring, complemented by alarms for failure detection
- Co-authored comprehensive design document detailing UI, system architecture, database design, and maintenance plans
- Reduced customer list generation time from 2 weeks to 2 minutes, saving data managers 5 hours weekly

May 2020 - August 2020 - *Virtual*

- Created a data pipeline for Alexa dialog data that would allow for in-depth analysis by linguists and engineers
- Utilized SQL queries to extract data from the Alexa database, processed it with Python scripts, and stored the results in S3
- Conducted an experiment on Alexa data, employing various G2P models to enhance contact recognition
- Showed improvement in contact recognition by 30%, saved engineers multiple hours when initiating experiments

PROJECTS

Pintos - Implemented key components of an Operating System in C, including support for user programs & a file system

- Built thread scheduler using priority-based queue; implemented aging to prevent thread starvation, priority donations to promote resource accessibility, round-robin to ensure fair share of CPU, and timer interrupts to manage running processes
- Designed data structures for efficient file metadata management and precise tracking of disk block allocation
- Employed linked allocation to prevent fragmentation and created system calls that enable user interactions with file system

Gitlet - Developed a feature-rich, object-oriented mini version control system in Java, incorporating Git's core functionalities

- Implemented multiple branch merges, using an efficient algorithm to find the LCA of the branches and resolve conflicts
- Introduced a "global-log" command to display a comprehensive version history and leveraged Java's serialization library to efficiently store and manage commit data

Mini Dropbox - Built an end-to-end file sharing system in Golang, including efficient appends for large files

- Used RSA encryption library to securely store files and HMACs to make them tamper-proof
- Used signing and verifying keys to validate the authenticity of file invitations