

Cory Kane

Software Developer

CONTACT

- 📞 1-760-596-2078
- @ ckane30691@gmail.com
- 📍 Tempe, AZ
- in linkedin.com/in/cory-kane
- github.com/ckane30691
- 🌐 corykane.dev

SKILLS

- Distributed Systems
- DynamoDB
- AWS Lambda/API Gateway
- CloudWatch
- CloudFormation
- Kafka
- Typescript/Javascript
- Node/Express
- MongoDB
- React/Redux
- Python/Flask
- Ruby on Rails
- PostgreSQL
- Heroku/Render/Netlify

EDUCATION

No Degree
Victor Valley College
Victorville, CA
2010

High School Diploma
Sultana High
Victorville, CA
2009

WORK EXPERIENCE

SOFTWARE DEVELOPER/SR. TECHNICAL INSTRUCTOR

App Academy – Remote - 2017 - Present

A premier programming bootcamp that teaches **React on Rails, React with Flask, AWS, and the MERN stack.**

- Automated data scraping from internal web apps using **Puppeteer**, improving efficiency in tracking student performance.
- Streamlined Salesforce workflows with **JSForce**, automating note-taking and doubling instructor capacity for student caseloads.
- Built a script with **Google APIs** to automate appointment scheduling through Google Calendar, enabling instructors to manage 10% more students.
- Developed and delivered lectures, apps, and instructional videos on the **MERN stack**, contributing to students' technical skill development.
- Created curriculum for **computer science, data structures, and algorithms, & scalability** reducing average student placement time from 5.5 to 3.5 months.
- Instructed students in **JavaScript, Python, Flask, React/Redux, Express/Node**, as well as **HTML & CSS** best practices and led teams through project cycles.

PROJECTS

WROTE-IT

[LIVE](#) | [GH](#)

Professional Project 2024

A Reddit-style web app built, using the **DLRN (DynamoDB Lambda React Node)** stack, for teaching at App Academy.

- Designed and implemented a serverless architecture using **AWS Lambda, DynamoDB, and API Gateway**, focusing on scalability and high availability.
- Utilized **DynamoDB Streams** for background data processing and updates regarding upvotes and reply counts, enhancing the responsiveness of the application.
- Employed best practices for cloud architecture, including serverless deployment through **CloudFormation**, caching strategies with **Redis**, and proper database sharding with **DynamoDB** to avoid scatter-gather & optimize performance.
- Incorporated AWS services such as **CloudWatch** and **IAM** for monitoring and security, demonstrating comprehensive knowledge of the AWS ecosystem.