LinkedIn Data

More than 1200+ connections as per 8th sep 2025 data

Name: Jay Patel

Headline
Software Engineer | Cloud & DevOps | AWS Certified Solutions Architect | Microservices & Automation | MACS ’25 Graduate

About
I’m a Software Engineer passionate about building scalable, reliable, and impactful systems. With a strong foundation in software engineering and cloud technologies, I focus on designing solutions that strike a balance between efficiency, security, and real-world usability.

I’ve had the chance to work with Java, TypeScript, C++, Go, and Python, along with tools and frameworks such as Node.js, Spring Boot, and React. Moving into the cloud space, I’ve worked with AWS and GCP, building automation pipelines, serverless platforms, and cloud-native architectures.

I value collaboration, continuous learning, and delivering results in fast-paced environments. Having recently completed my Master’s in Applied Computer Science at Dalhousie University, I’m eager to contribute to projects that drive innovation in software, cloud, and AI.

Experience
ApplyBoard
Software Engineer Co-op
May 2025 - August 2025 (4 months)
Kitchener, Ontario, Canada

TronTechh LLC
Full Stack Developer
October 2023 - May 2024 (8 months)
Remote

Galanto Innovations
Android Developer Intern
February 2023 - August 2023 (7 months)
Research Park, IIT Gandhinagar

Education
Dalhousie University
Master's degree, Computer Science · (January 2024 - August 2025)

Gujarat Technological University (GTU)
Bachelor of Engineering - BE, Information and Communication Technology · (May 2019 - May 2023)

Licenses & certifications
Amazon Web Services (AWS)
AWS Certified Solutions Architect
Issued Aug 2025 · Expires Aug 2028

Projects
RoomM8 (Associated with Dalhousie University)
RoomM8 – A Full-Stack Roommate Collaboration Platform (Java + Next.js + AWS)

⚙ Key Features:
• Manage shared expenses, tasks, announcements, and grocery lists in one collaborative platform.
• Room-based access: Users can create a room and invite friends via a unique

room code to collaborate within a private space.
• Secure authentication using JWT tokens.
• Custom Spring Interceptor to enforce room membership validation, ensuring users can only access APIs for rooms they belong to, with flexible include/exclude API control.

🔐 Quality & Testing:
• Achieved more than 90% test coverage through automated unit and integration tests using JUnit and Mockito, following the TDD (Test-Driven Development) approach.
• SonarQube is integrated for automated static code analysis to maintain high code quality.

🚀 Deployment & Automation:
• Built CI/CD pipelines using Docker and GitLab CI/CD for automated build, test, and deployment.
• Automated infrastructure provisioning with AWS CloudFormation (IaC) to ensure consistent deployments.
• Deployed architecture aligned with AWS Well-Architected Framework.

📈 Architecture & Design:
• Followed MVC architecture for clean separation of concerns and scalable design.
• Applied SOLID principles and standard design patterns (e.g., Factory, Builder, Singleton, where applicable) for maintainability.
• Load tested using Apache JMeter to validate scalability for high user concurrency.

📋 Development Methodology:
• Followed Agile practices using JIRA for sprint planning and task management.
• Comprehensive technical documentation was created, including architecture diagrams, sequence flows, API specs, security design, and future enhancement roadmap.

Skills: Java · Spring Boot · Next.js · MySQL · Docker · TypeScript · Gitlab · REST APIs · GitLab CI/CD · Amazon Web Services (AWS) · AWS Well-Architected Framework · Test-Driven Development · JWT Auth · Unit Testing · Continuous Integration and Continuous Delivery (CI/CD) · Integration Testing · Performance Testing · JUnit · JMeter · Mockito · SOLID Design Principles · Software Design Patterns · Technical Documentation · Infrastructure as code (IaC) · Agile Scrum · Code Review · Jira · Sonarqube · Amazon EC2 · AWS Auto Scaling · Amazon VPC · Amazon Relational Database Service (RDS) · AWS Secret Manager · Elastic Load Balancing · Amazon CloudWatch · AWS Lambda · AWS API Gateway · AWS CloudFormation

QuickData Processor (Associated with Dalhousie University)

QuickData Processor (QDP) - A Serverless Multi-Cloud Data Processing Platform (AWS + GCP)

⚙️Key Features:
- JSON to CSV Conversion using AWS Glue.
- Text Entity Extraction powered by Natural Language Processing (NLP).
- Word Cloud Generation using Google BigQuery and Looker Studio.

🔐 QDP ensures robust user protection through a three-factor authentication system implemented with:
- AWS Cognito for user management.
- AWS Lambda for serverless verification logic.
- DynamoDB for storing credentials, security questions, and math-based verification tokens.

📩 The platform provides real-time email notifications triggered on user actions such as login, signup, or data processing events, using AWS SNS/SQS for asynchronous message delivery.

💬 Dialogflow-based Virtual Assistant for handling user queries, using Google Pub/Sub for event-driven messaging, and Firestore for real-time data synchronization.

📊 QDP offers an interactive dashboard built with Looker Studio, enabling admins to:
- Monitor all users' activity.
- Processing Sentiment Analysis.

🚀 Deployment & Infrastructure:
- Frontend and backend Docker containers are deployed to Google Artifact Registry via Cloud Build.
- The entire infrastructure is provisioned using Terraform, supporting a cross-cloud deployment model (AWS + GCP) to automate and standardize the provisioning.

Skills: Amazon Web Services (AWS) · Google Cloud Platform (GCP) · Serverless Architecture · Cross-Cloud Integration · Amazon Cognito · AWS Lambda · Amazon Dynamodb · AWS Glue · Amazon SQS · Amazon Simple Notification Service (SNS) · AWS Secret Manager · GCP Cloud Build · GCP Cloud Run · GCP Pub/Sub · Google Natural Language API (Sentiment Analysis) · GCP Dialogflow (Virtual Assistance) · Looker Studio · Amazon S3 · Firebase Database

TinyDB (Associated with Dalhousie University)

TinyDB – Lightweight Relational Database System in Java

• Built a MySQL-like relational database from scratch using Java, featuring a command-line interface for executing both DDL and DML SQL queries.
• Implemented transactional support with commit and rollback, adhering to the ACID properties to ensure atomicity, consistency, isolation, and durability of data operations.
• Integrated two-factor authentication (2FA) using username/password and security questions, enhancing user account security.
• Enforced essential database constraints such as primary keys, foreign keys, UNIQUE, and NOT NULL, ensuring data consistency and referential integrity.
• Developed functionality for automatic Entity Relationship Diagrams (ERD) generation and exporting SQL files, enhancing schema visualization, data portability, and ease of database migration.

Skills: Core Java

Slackify (Associated with Dalhousie University)

Slackify – A Cloud-Native GitHub to Slack Integration Platform (AWS + Java + Spring Boot)

⚙️ Project Overview
• Slackify is a SaaS integration platform designed to boost developer productivity by seamlessly connecting GitHub repository events with Slack channels, automating real-time collaboration and issue tracking.
• Automatically creates Slack channels for new GitHub issues, posts issue updates and comments, and keeps teams instantly informed — eliminating manual monitoring.

🚀 Architecture & Implementation
• API Gateway acts as the entry point for user requests and webhook integration.
• AWS Lambda functions handle input validation, webhook creation, and SNS email

triggers.
• Spring Boot app (deployed on EC2) processes GitHub events and manages Slack communications.
• Backend data stored in Amazon RDS (MySQL); sensitive tokens secured with AWS Secrets Manager.
• Full infrastructure provisioned using AWS CloudFormation (IaC).
• Docker-based deployments for rapid versioning and updates.

🤗 Security & Reliability
• API Gateway protects the Lambda backend with fine-grained access control.
• Secrets Manager safeguards sensitive credentials.

💡 Impact
• Enabled instant visibility of GitHub activity across distributed teams.
• Reduced response time on issues and boosted team agility.

Skills: AWS CloudFormation · Amazon EC2 · Amazon Relational Database Service (RDS) · AWS Lambda · AWS API Gateway · Amazon Simple Notification Service (SNS) · AWS Secret Manager · GitHub WebHook · Slack API · Docker · Java · Spring Boot · Infrastructure as code (IaC) · GitHub

KubeMicro (Associated with Dalhousie University)

☁ KubeMicro – Scalable Microservice Deployment on Kubernetes with Terraform

• Developed two independent yet interacting microservices using Java and Spring Boot, following a modular architecture.
• Containerized both services with Docker and established a CI/CD pipeline to deploy the images to Google Artifact Registry via GCP Cloud Build.
• Deployed the services on Google Kubernetes Engine (GKE), ensuring scalable and reliable orchestration.
• Provisioned and managed Kubernetes infrastructure using Terraform, enabling reproducible and automated cluster setup.
• Configured Persistent Volumes in GKE to allow seamless file sharing and retrieval between the services.

Skills: Google Kubernetes Engine (GKE) · Terraform · Docker · Continuous Integration and Continuous Delivery (CI/CD) · Java · Spring Boot · Kubernetes · GCP Cloud Build · Artifact Registry · Microservices · Google Cloud Platform (GCP) · GitHub

Smart Towing Assistance (Associated with Gujarat Technological University (GTU))

- This Mobile Application is useful for government towing departments to do safe and secure towing vehicles in public areas.
- Functionalities: Scanning the number plate, calling or sending a message to the owner of the vehicle with just a click, and also having a facility to pay fine through digital Mode.
- Modules: OCR Dependency, Firebase Database, Razor pay.

Quick and Share Logistics (Associated with Gujarat Technological University (GTU))

- An End-to-End transport facility Mobile Application used to share the transportation of logistics mainly by a small-scale industry.
- Through this fuel consumption, air pollution, and cost of transportation can be decreased via sharing of transportation between two or more companies.
- Modules: Material Components, Search Bar, Firebase CRUD Operations

Posts (Most Recent to oldest)

1)
📘 What started with small AWS assignments soon grew into full projects and later gave me the chance to put those skills into practice during my co-op on a real, scalable system. Each step made me more curious and pushed me to go deeper.

🎉 That journey has led me here today, and I'm excited to share that I've earned the AWS Certified Solutions Architect – Associate! (SAA-C03)

🚀 Looking forward to applying these skills to bigger challenges ahead.

Check it out: https://www.credly.com/badges/5d2af1b8-d81d-4919-84c5-d982ae224321/public_url

2)
Excited to begin a new chapter this summer at ApplyBoard!

I'm happy to share that I've joined ApplyBoard as a Software Engineer Co-op, where I'll be contributing to a platform that empowers students globally to access the best educational opportunities.

ApplyBoard is on a mission to educate the world by simplifying the study abroad process, and I look forward to building impactful, scalable solutions that support this vision and improve the student experience.

Special thanks to Conrad Chan and Xeniya Shirinova for the opportunity, as well as Isabelle Côté and Thivya Sivananthan for making the onboarding experience smooth and welcoming.

Looking forward to an inspiring and meaningful term ahead! 🚀

3)
🚀 My Amazon SDE Intern Interview Experience.

Recently, I had the opportunity to interview for the Software Development Engineer (SDE) Intern position at Amazon. While I didn't receive the offer, I gained invaluable insights and a renewed determination to keep pushing forward.

📌 The Interview Process:

🔷 Online Assessment (Divided into two parts)
1️⃣ Two coding problems (Each with 15 test cases)
2️⃣ Work style assessment – A multiple-choice questionnaire evaluating alignment with Amazon's Leadership Principles.

🔷 Final Interview Round (1 Hour)
💬 Behavioral (30 min) – Focused on Amazon's Leadership Principles, requiring structured responses with real-world experiences.
💻 Technical (30 min) – Typically involves LeetCode Medium to Hard DSA problems, but I was given a low-level design question instead.

💡 Key Takeaways:

✅ Master the Leadership Principles – Amazon places significant emphasis on these. Structuring answers using the STAR method helps articulate experiences effectively.
✅ Clarity in Thought Process is Crucial – It's not just about getting the right solution but also about how you communicate your approach.
✅ Stay Calm & Adapt – Sometimes, things don't go as planned. Staying calm and adapting can make a big difference.
✅ Every Experience is a Stepping Stone – This wasn't a failure, but a chance to

improve and come back stronger.

While I had hoped for a different outcome, this experience has strengthened my DSA skills as well as my communication skills. I’m grateful for the learning, the challenge, and the motivation it has given me to refine my skills even further in a short span of time.

For anyone preparing for interviews—keep coding, keep learning, and keep growing. This is just a single step in the journey, and I’m excited for what’s next 🚀

4)
Starting February with a very exciting Internship! So happy to announce that I will be joining Galanto Innovations as Android Developer Intern.

I look forward to improving my skills and gaining invaluable working experience by joining a wonderful team who work with the very special product in Healthcare Sector. I will surely carry that knowledge with me into the next phase of my career.

Special thanks to Chandan Kumar Jha, PhD, and Rupsha Mukherjee for giving me this opportunity.