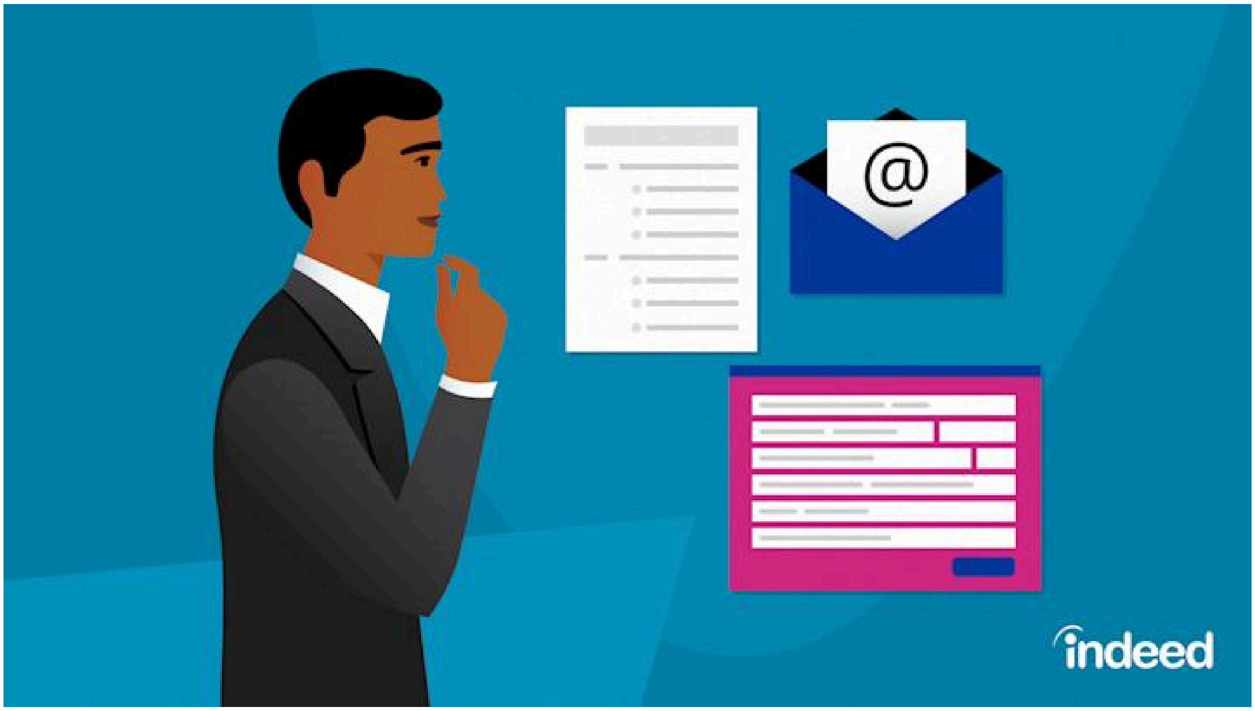
JOB APPLICATION PACKAGE

By: Robel Bruk



JOB DESCRIPTION:

Company: Affirm, Inc.

Role: Software Engineer I (Merchant Data Platform)

Affirm is reinventing credit to make it more honest and friendly, giving consumers the flexibility to buy now and pay later without any hidden fees or compounding interest.

The Merchant Data Platform is a backend, foundational service at Affirm. We are responsible for merchant data and identity across all merchant touchpoints. Our checkout-critical platform serves millions of requests per day, supports the onboarding of merchants across all onboarding touchpoints and maintains strict availability and latency SLAs given the criticality of the service. Further, the team maintains sophisticated pipelines that enrich our datasets for the purposes of merchant identification, risk analysis and fraud detection.

What You'll Do:

- Work collaboratively and proactively with your team and stakeholders.
- Strike the right balance of speed and quality in development.
- Contribute to the team's projects with mentorship and support.
- Engage in growth and development activities.

What We Look For:

- Experience in backend development using Python or Kotlin.
- Familiarity with distributed systems, AWS, MySQL, Kubernetes.
- Ability to design scalable systems and write well-tested code.
- Excellent communication and teamwork.
- Bachelor's degree or equivalent practical experience.

Compensation: USA base pay range (all other U.S. states): $115,000 - $155,000 annually, pro-rated for internship.

RELATED COURSEWORK:

Fundamentals of Algorithms

This course introduces the theoretical and practical aspects of designing efficient algorithms to solve computational problems. It emphasizes the mathematical foundations necessary for evaluating algorithm efficiency, particularly through the use of Big-O notation. Core topics include sorting and searching techniques, divide and conquer strategies, greedy methods, and dynamic programming. The course provides a deep understanding of how algorithms behave under different inputs and constraints, preparing students to not only implement classical solutions but also design new algorithms tailored to specific real-world applications.

Applied Data Structures & Algorithms

Expanding on fundamental algorithm knowledge, this course focuses on the implementation and application of complex data structures in various computing scenarios. Students study how different data structures like linked lists, trees, graphs, and hash maps operate and how they can be used to develop robust solutions for real-time systems, search engines, and databases. Emphasis is placed on applying these tools to algorithmic problems that mirror challenges in software development, including route optimization, social network modeling, and recommendation systems. The course trains students to assess trade-offs in performance, memory usage, and implementation complexity, especially in the context of real-world systems.

Cloud Computing (AWS)

This course explores the principles and technologies behind modern cloud infrastructure, with a hands-on focus on Amazon Web Services (AWS). Students learn how to provision virtual machines, set up scalable storage, and manage databases in cloud environments using services like EC2, S3, RDS, and Redshift. Additionally, they study best practices for security, automation, and deployment using tools like IAM, load balancers, and CI/CD pipelines. The curriculum provides practical skills in architecting cloud-native applications, managing infrastructure as code, and building resilient systems that meet performance and cost-efficiency goals. By the end of the course, students are capable of deploying full-stack applications and services to the cloud with professional-grade configurations.

Database Systems

This course provides a comprehensive understanding of the design, implementation, and management of database systems. Students explore the relational model and master SQL for data manipulation and querying. Key topics include data normalization, indexing, query optimization, and transaction processing, all of which are essential for maintaining consistency and performance in real-world applications. Through the development of databases in systems like MySQL or PostgreSQL, students gain experience designing schemas, enforcing constraints, and handling concurrency. The course also introduces the differences between relational and non-relational databases, preparing students to make architectural decisions based on application needs.

Natural Language Processing (NLP)

Focusing on the interaction between computers and human language, this course introduces the fundamental concepts and techniques used to process and analyze large volumes of textual data. Students explore text preprocessing methods such as tokenization, stemming, and lemmatization, and build models for tasks like sentiment analysis, entity recognition, and document classification. The course includes practical use of Python libraries like NLTK and spaCy, and culminates in the application of advanced transformer-based models like BERT and GPT for language understanding. It equips students with the tools to develop applications such as intelligent chatbots, search engines, and sentiment trackers, all while gaining insight into the challenges of ambiguity and context in human language.

Computer Architecture II

This advanced course delves into the internal structure and operation of modern computer systems. It examines how hardware and software interact through the study of instruction set architectures, pipelining, and parallelism. Students learn about the design and optimization of CPU operations, memory hierarchies (including caching and virtual memory), and how modern processors achieve efficiency through speculative execution and out-of-order processing. Hands-on assignments involve writing low-level programs and analyzing performance trade-offs, giving students a deep understanding of how code is executed at the hardware level. The course builds the foundation for system performance tuning, embedded systems development, and hardware-aware software engineering.

Operating Systems

This course explores the principles and mechanisms behind the design and functionality of operating systems. Students learn how operating systems manage resources such as the CPU, memory, and I/O devices, and how they provide abstractions like processes, threads, and virtual memory. The curriculum covers process scheduling, inter-process communication, synchronization, file systems, and security policies. Practical assignments include implementing components of a simple operating system or simulating behaviors like round-robin scheduling or memory paging. By understanding how operating systems coordinate hardware and software, students become equipped to build more efficient, secure, and responsive applications.

Robel Bruk

robelbruk5@gmail.com | 240-495-0715 | linkedin.com/in/robel-bruk | github.com/robelbruk

EDUCATION

Howard University, Washington, D.C. Expected Graduation May 2026

Bachelor of Science, Computer Science

- **Coursework:** Fundamentals of Algorithms |Applied Data Structures & Algorithms | Cloud Computing (AWS) |Database Systems | Natural Language Processing (NLP) | Computer Architecture | Operating Systems
- **Honors:** AmazonNext Scholar, College of Engineering and Architecture (CEA) Dean's List
- **Organizations:** CodePath, Ethiopian Eritrean Student Association, National Society of Black Engineers, Google Developers Student Club, Howard University Fintech

SKILLS

Front-End Development: React, React Native, JavaScript, TypeScript, Angular, Remix.js, GraphQL, HTML, CSS
Back-End Development: Python, Java, Swift, Node.js, Flask, Django, PostgreSQL, MySQL, Spring, MongoDB
Other: Amazon Web Services (EC2, S3, RDS, Redshift), Microsoft Azure, Git, Vercel, Linux, Nix, Jenkins, Agile/Scrum

EXPERIENCE

Cambridge Mobile Telematics June 2024 - December 2024

Software Engineering Intern, Cloud Applications

- Developed and implemented a Python ETL tool at scale to streamline data transfer and transformation between Redshift and RDS, improving production-to-release candidate processing time.
- Automated deployment pipelines and enhanced CI/CD efficiency using Jenkins.
- Created technical solutions that resolved key system bottlenecks, increasing overall performance and reliability.
- Partnered with teams in Cloud Infrastructure, Data Science, and Quality Assurance to deliver scalable solutions.

Cambridge Mobile Telematics May 2023 - August 2023

Engineering Intern

- Developed a proposal for Airbnb's explore feature based on industry research, presenting innovative insights.
- Optimized complex queries in Amazon Redshift using PostgreSQL, improving data retrieval efficiency by **10%**.
- Enhanced team productivity by building data-driven tools to address technical challenges.
- Coordinated with Security Engineering, Product Management, and Data Analysis teams to support cross-functional initiatives.

Debre Tsion Mariam & Gabriel Church, 501(c) June 2023 - July 2023

Back-End Software Engineer, *Contractor*

- Led the development of a client website using Angular, Firebase, and Google Cloud Platform APIs, reducing operational expenses by **60%**.
- Built TypeScript modules for seamless HTTP requests to Firebase, ensuring a robust backend infrastructure.
- Developed a production-ready website on schedule. **Production**: https://debretsion.org/en

RELEVANT CERTIFICATIONS

- AmazonNext x CodePath Intro to Software Engineering Course -- *Certification*

COVER LETTER

Recruiting Team
Affirm, Inc.

650 California St, 12th floor,

San Francisco, CA 94108

April 16, 2025

Dear Affirm Team,

I am writing to express my strong interest in the Backend Engineering Intern position on the Merchant Data Platform at Affirm. As a junior pursuing a Bachelor of Science in Computer Science at Howard University, I bring a blend of backend development experience, cloud computing exposure, and a passion for building scalable and impactful systems.

During my time at Cambridge Mobile Telematics, I designed and implemented an ETL pipeline in Python that improved data processing efficiency between Redshift and RDS. Additionally, I automated deployment workflows with Jenkins and collaborated with infrastructure and data science teams to ensure system reliability. My hands-on exposure to AWS services, CI/CD, and distributed data systems has prepared me well to contribute meaningfully to Affirm's high-volume, latency-sensitive backend systems.

I am especially drawn to Affirm’s mission of transparent financial services. My background in optimizing cloud applications and passion for backend system design align perfectly with the work done by the Merchant Data Platform team. I am confident I would contribute proactively to team goals, bring clear communication, and help enhance system performance.

My salary expectations are flexible based on company standards for interns. I am available for interviews at your convenience and can be reached at robelbruk5@gmail.com or 240-495-0715.

Thank you for considering my application. I look forward to the opportunity to contribute to Affirm’s continued innovation.

Sincerely,
Robel Bruk

REFLECTIVE MEMO

To: Prof. John Harris
From: Robel Bruk
Subject: Job Application Package Reflective Memo
Date: April 16, 2025

The following memo outlines the reasoning behind my writing and organizational choices for this job application package.

Job Selection & Purpose

I chose Affirm's Backend Engineering Intern position because it aligns with my current technical skills and future aspirations. Affirm's commitment to transparent financial services resonated with my values, and the backend focus complements my ETL and cloud experience.

Résumé Design

My résumé emphasizes technical experience and quantifiable achievements. I structured it to lead with education, followed by skills, experience, and projects. The design choice was clean and functional, reflecting standard industry expectations while showcasing high-impact bullet points.

Cover Letter Strategy

In my cover letter, I focused on communicating enthusiasm for the company's mission and drawing clear parallels between the job requirements and my past experience. I deliberately highlighted one main internship (CMT, 2024) for depth, while mentioning others to establish consistency and reliability. My conclusion includes availability and pay expectations while inviting further conversation.

Overall Structure

The package begins with the job description to ground the reader. This is followed by coursework to contextualize my qualifications, then the résumé, and finally the cover letter and this memo. This logical flow mirrors professional job application materials and demonstrates an understanding of real-world technical communication expectations.

Sincerely,
Robel Bruk
B.S. Candidate, Computer Science
Howard University