GenAI Proficiency Test - Data Tech Lead

Test Overview

Test Objectives and Deliverables

▼ Purpose, Duration, and Expected Outputs

Test Objective

- **Evaluate technical leadership proficiency** in **designing and planning complex data infrastructure projects using genAI tools (Cursor, Windsurf, Claude, etc.)**
- **Assess technical architecture skills** - system design, infrastructure planning, deployment strategies
- **Measure project management integration** - combining technical delivery with team coordination and timeline management
- **Test technical communication ability** - translating complex infrastructure into clear stakeholder documentation
- **Evaluate GenAI usage for technical planning** - leveraging AI tools for infrastructure design and team coordination

Duration and Deliverables

- **Expected time commitment**:
  - This exercise is designed for you (the candidate) to complete **in OUT-OF-OFFICE time in 5 days**
  - We understand you have senior responsibilities and designed this to showcase your technical architecture and leadership abilities
- **Primary deliverables**: `report_<task>.md` files for each selected task **(AT LEAST ONE MAIN FILE FOR EACH TASK)**
  - If you have multiple main files, name them: `report_<task>_part01_<part_name>.md`
  - You should save your main/long prompts for technical workflow illustration
    - Prompt file naming: `report_<task>_prompt.md` or `report_<task>_part01_prompt.md`
- **Technical architecture focus**: Infrastructure design, deployment planning, system integration **with** leadership coordination
- **Report specifications**: Expected 1200-2000 lines per report file, 2-8 files per task
  - **Choose your approach** based on technical complexity and stakeholder communication needs

Documentation Viewing (Optional but Recommended; Skip if you have issues)

- **Install npm first**:
  - **Windows**: Download and install Node.js using the MSI installer from https://nodejs.org/en/download/
  - **Ubuntu**: `sudo apt update && sudo apt install nodejs npm`
  - **Mac**: `brew install node npm`

- **Use Docusaurus for better viewing**: Download `docusaurus_stub.zip` from this GoogleDrive link, unzip, add your markdown files to `/docs` folder
- **Setup**: Run `npm install` and `npm run start` at root project, fix any errors if needed
  - npm server should run at `http://localhost:3000/` after `npm run start`
- **Docusaurus view is superior to IDE view or git view** for reading and reviewing technical leadership reports

Requirements and Assessment Criteria

Submission Requirement

▼ Deliverable Format and Submission Standards

- **Primary deliverable format**: `report_<task>.md` files with technical architecture focus AND leadership coordination
- **File naming convention**:
  - Main files: `report_<task>.md` (AT LEAST ONE MAIN FILE FOR EACH TASK)
  - Multiple parts: `report_<task>_part01_<part_name>.md`
  - Prompt files: `report_<task>_prompt.md` or `report_<task>_part01_prompt.md`
- **Report specifications**: Expected 1200-2000 lines per report file, 2-8 files per task
- **Technical leadership prompts**: Include `report_<task>_prompt.md` showing strategic genAI usage for technical planning
- **Style compliance**: Must follow `ctx_doc_style.md` formatting exactly
- **Multi-audience accessibility**: Technical content understandable by both engineering teams and business stakeholders
- **Task selection**: Minimum one from List A (mandatory), additional from List B (optional)
- **Supplementary materials**: Technical project coordination materials, infrastructure plans, technical team coordination strategies **if applicable**
- **GenAI UTILIZATION**: DEMONSTRATE EFFECTIVE USE OF AI TOOLS to meet all technical, architectural and leadership requirements through documented examples and workflows

Infrastructure Requirement

▼ Technical Architecture and System Design Standards

Infrastructure Architecture Requirements

- **System architecture documentation** - detailed component interactions, dependencies, and technical integration patterns
- **Infrastructure components** - VMs, services, modules, component interactions, access control systems
- **Network and storage design**:
  - Network architecture with security groups and load balancing
  - Storage architecture for shared access and data persistence
  - Access control systems and permission structures
- **Infrastructure as Code** - Terraform configurations and Ansible automation strategy
- **Performance considerations**:
  - Scaling approach and resource allocation
  - Performance specifications and optimization
  - Load testing and monitoring requirements

Technical Implementation Standards

- **Deployment procedures**:
  - Detailed installation steps and configuration
  - Chronological deployment timeline
  - Technical maintenance workflows
- **Integration specifications**:
  - API design and database schema
  - Real-time processing architecture
  - Component communication patterns
- **Technical accountability**:
  - Comprehensive documentation for engineering teams
  - Infrastructure delivery timelines
  - Technical snippets for architecture clarity
  - System monitoring and troubleshooting procedures

Leadership Requirement

▼ Demonstrating Technical Leadership Excellence and Strategic Project Management

Technical Team Leadership & Development

- **Engineering team guidance** - coordinate and mentor engineers through complex infrastructure implementations
- **Cross-functional coordination** - manage technical integration across departments and infrastructure teams
- **Technical adoption & training** - develop training plans and mentor team members on new technologies
- **Technical leadership approach** - demonstrate ability to guide teams through complex delivery while fostering growth
- **Continuous improvement** - show adaptability and learning in infrastructure project approaches

Project & Resource Management

- **Technical planning & timelines** - create comprehensive infrastructure plans from requirements to deployment
- **Resource allocation** - manage technical resources, timelines and risks realistically
- **Milestone coordination** - establish clear implementation schedules and team assignments
- **Cross-team integration** - coordinate dependencies between multiple teams effectively
- **Project management competency** - demonstrate clear timeline, resource and risk management capabilities

Stakeholder Communication & Alignment

- **Business communication** - translate technical concepts for business audiences and provide progress updates
- **Executive engagement** - present architecture decisions and capabilities to business leaders
- **Strategic alignment** - ensure technical solutions support long-term business objectives
- **Proactive transparency** - identify issues early and maintain clear communication about infrastructure status
- **Value demonstration** - articulate technical capabilities in terms of business value

Documentation Requirement

▼ Documentation Standards and Technical Communication Requirements

Documentation and Standards

- **Technical-first documentation** - comprehensive infrastructure design accessible to both engineering teams AND business stakeholders
- **Multi-audience technical documentation** - infrastructure details accessible to both engineering teams and business leaders
- **Terminology standardization** - create consistent technical terminology section and use throughout report
- **Documentation frameworks** - establish standards for team infrastructure documentation and knowledge sharing
- **Project management integration** - timeline, resource, and technical risk considerations
- **Technical cross-functional communication** - translates infrastructure complexity for business stakeholders
- **Clear accountability** - demonstrates understanding of delivery commitments and timeline management
- **Stakeholder alignment** - shows approach for keeping all parties informed about progress

Technical Visualization and Implementation

- **Architecture diagrams** - detailed system integration, component relationships, infrastructure topology
- **Timeline visualization** - Gantt charts with milestones, dependencies, deployment sequences
- **Team coordination charts** - role assignments, communication flows, responsibility matrices
- **Stakeholder materials** - infrastructure overviews and progress updates for business/executive audiences
- **Mermaid charts** - for architecture, flows, and system integration diagrams
- **Leadership focus** - diagrams facilitate team coordination and stakeholder understanding
- **Integration approach** - include visualizations directly in reports or as supplementary materials

List A - Mandatory Tasks - Choose AT LEAST one task

▼ Technical Leadership and Project Management Core Tasks (Choose Minimum One)

A01 - AWS Data Platform Foundation - Technical Architecture & Leadership

What You Need to Deliver - Technical Architecture FIRST

Design the complete technical architecture and deployment plan for an AWS Data Platform that serves as the foundation for a data engineering team. You must create implementable technical specifications.

Required Technical Deliverables

- **Complete infrastructure architecture** - detailed system design showing all AWS components and their interactions
- **Deployment chronology** - step-by-step technical implementation timeline in exact chronological order
- **Infrastructure as Code specifications** - Terraform configuration approach and Ansible automation strategy
- **Access control architecture** - IAM roles, policies, security groups, authentication flow design
- **System integration documentation** - how components connect, data flows, network architecture
- **Operational procedures** - monitoring setup, backup procedures, maintenance workflows

Required Platform Components (Technical)

- **User Linux systems** - EC2 instance architecture, sizing, configuration specifications
- **User AWS access management** - IAM role design, policy structures, access control implementation
- **NFS storage architecture** - shared file system design, capacity planning, performance considerations
- **FreeIPA integration design** - authentication system architecture, directory service integration
- **Infrastructure automation** - complete IaC approach using Terraform and Ansible (no manual AWS console work)

Leadership Coordination (Added Layer)

- **Team implementation guidance** - how you would coordinate 3-4 engineers through this technical implementation
- **Project timeline management** - 8-week delivery schedule with technical milestones and team assignments
- **Technical stakeholder communication** - how to explain infrastructure progress to business leaders
- **Technical risk management** - infrastructure risks and mitigation strategies
- **Team technical development** - how to upskill engineers during platform implementation

What Success Looks Like

Your deliverable should answer: "If I give this technical plan to a DevOps team, can they implement this AWS platform step-by-step? And how would you lead them through it?"

A02 - Dask Cluster - Technical Integration & Team Leadership

What You Need to Deliver - Technical Architecture FIRST

Design the complete Dask distributed computing cluster architecture that integrates with the AWS Data Platform (A01). You must create detailed technical specifications for a production-ready cluster.

Required Technical Deliverables

- **Dask cluster architecture** - scheduler design, worker node configuration, networking topology
- **Integration specifications** - how Dask connects with AWS platform, FreeIPA, and existing infrastructure
- **Resource management design** - memory allocation, CPU scheduling, concurrent user handling for 20-30 users
- **Deployment automation** - Terraform and Ansible configurations for cluster provisioning
- **Performance optimization** - cluster tuning, scaling policies, resource monitoring setup
- **User access architecture** - how engineers connect to cluster, job submission workflows, authentication

Technical Requirements (Specifications)

- **Dask scheduler architecture** - high availability setup, job coordination, metadata management

- **Worker node design** - auto-scaling configuration, resource allocation, fault tolerance
- **Networking architecture** - internal cluster communication, security groups, load balancing
- **Storage integration** - shared storage access, temporary data handling, result persistence
- **Monitoring and alerting** - cluster health metrics, performance tracking, user activity monitoring

Leadership Coordination (Added Layer)

- **Cross-team integration** - coordinate with platform team (A01), manage dependencies
- **Technical team management** - guide 2-3 engineers through complex distributed systems implementation
- **User adoption strategy** - technical training plan for 20-30 data scientists on cluster usage
- **Integration timeline** - 6-week technical implementation with milestone coordination
- **Performance accountability** - ensure cluster meets concurrent workload requirements

What Success Looks Like

Your deliverable should answer: "Can engineers implement this Dask cluster using your technical specifications? And how would you coordinate the integration with existing infrastructure?"

A03 - Metaflow ML Pipeline Platform - Technical Implementation & Strategic Leadership

What You Need to Deliver - Technical Architecture FIRST

Design the complete Metaflow ML Pipeline Platform architecture that enables Machine Learning (ML) workflow orchestration at scale. You must create comprehensive technical implementation specifications.

Required Technical Deliverables

- **Metaflow architecture design** - service components, metadata store, workflow orchestration system
- **AWS service integration** - S3 artifacts, EC2/Batch compute, Step Functions, parameter store integration
- **User workspace architecture** - development environment setup, pipeline creation workflows, execution management
- **Infrastructure automation** - complete Terraform and Ansible deployment specifications
- **ML pipeline templates** - standardized workflow architectures for common ML patterns

- **Operational monitoring** - pipeline execution tracking, performance metrics, failure handling

Technical Components (Specifications)

- **Metaflow service architecture** - metadata service, workflow scheduler, execution engine design
- **Compute scaling architecture** - local execution vs cloud scaling, resource allocation, batch job management
- **Storage architecture** - artifact management, version control, experiment tracking, model storage
- **Integration specifications** - connection with existing AWS platform, authentication, networking
- **Development workflow** - how engineers create, test, and deploy ML pipelines using the platform

Leadership Coordination (Added Layer)

- **Business stakeholder alignment** - translate ML platform technical capabilities into business value propositions
- **Technical team leadership** - coordinate 2-3 engineers through ML infrastructure implementation
- **ML team integration** - technical collaboration with 15-20 ML engineers for platform adoption
- **Strategic timeline** - 10-week technical implementation with quarterly business milestone alignment
- **Technical adoption metrics** - measurable productivity improvements for ML workflow efficiency

What Success Looks Like

Your deliverable should answer: "Can engineers build this Metaflow platform using your technical architecture? And how do you ensure it delivers business value for ML teams?"

A04 - Web/App Tracking Analysis - Technical Architecture & Strategic Assessment

What You Need to Deliver - Technical Analysis FIRST

Create comprehensive technical analysis of web and mobile app tracking systems, comparing custom-built technical architecture versus AppsFlyer integration. You must provide implementable technical specifications.

Task A04a - Custom Tracking Service Technical Architecture

Design complete technical architecture for custom tracking system

Required Technical Deliverables

- **Tracking system architecture** - data collection, attribution logic, event processing pipeline
- **Technical implementation specifications** - API design, database schema, real-time processing architecture
- **Integration architecture** - web SDK, mobile SDK, server-to-server tracking implementation
- **Data flow design** - event collection, processing pipeline, attribution calculation, reporting system
- **Infrastructure requirements** - scaling considerations, database design, API performance specifications
- **Technical deployment plan** - implementation timeline, system dependencies, testing procedures

Task A04b - AppsFlyer Integration Technical Assessment

Complete technical evaluation of AppsFlyer integration architecture

Required Technical Deliverables

- **AppsFlyer integration specifications** - API integration, SDK implementation, data flow architecture
- **Technical comparison analysis** - feature mapping between custom vs AppsFlyer capabilities
- **Implementation architecture** - how AppsFlyer integrates with existing systems, data pipeline design
- **Cost-benefit technical analysis** - development resources, maintenance overhead, technical capabilities comparison
- **Migration considerations** - technical transition plan, data consistency, system integration

Leadership Coordination (Added Layer)

- **Cross-functional technical coordination** - work with marketing, product, and engineering teams on tracking requirements
- **Executive technical communication** - present technical architecture decisions to business leaders
- **Technical team guidance** - coordinate engineers through complex tracking system implementation

- **Business impact analysis** - connect technical tracking capabilities to marketing attribution strategy
- **Technical timeline accountability** - 4-week analysis with executive presentation milestone

What Success Looks Like

Your deliverable should answer: "Can engineers implement either tracking approach using your technical specifications? And how do you communicate technical decisions to business stakeholders?"

A05 - Real-Time Streaming Data Pipeline - Technical Architecture & Enterprise Leadership

What You Need to Deliver - Technical Architecture FIRST

Design complete technical architecture for enterprise-grade streaming data pipeline that processes AppsFlyer data in real-time. You must create comprehensive technical implementation specifications.

Required Technical Deliverables

- **End-to-end pipeline architecture** - data ingestion, stream processing, aggregation layers, dashboard integration
- **Streaming infrastructure design** - Kafka/Kinesis architecture, partitioning strategy, scaling configuration
- **Processing engine specifications** - Apache Flink/Spark Streaming implementation, windowing, state management
- **Storage architecture** - real-time data store, aggregation tables, dashboard query optimization
- **Data transformation logic** - event processing, aggregation calculations, business metric derivations
- **Monitoring and alerting architecture** - pipeline health, data quality, performance metrics, SLA tracking

Technical Components (Specifications)

- **Data ingestion architecture** - AppsFlyer API integration, rate limiting, error handling, backpressure management

- **Stream processing design** - event-time processing, late data handling, exactly-once semantics, fault tolerance
- **Aggregation layer architecture** - hourly/daily aggregation logic, incremental processing, data consistency
- **Dashboard integration specifications** - real-time query engine, API design, caching strategy
- **Infrastructure automation** - complete deployment pipeline, scaling policies, configuration management

Leadership Coordination (Added Layer)

- **Enterprise stakeholder management** - coordinate with marketing, product, analytics, and executive teams
- **Technical team leadership** - manage complex architecture implementation across multiple engineers
- **Business-critical system accountability** - ensure zero data loss, maintain system reliability for business decisions
- **Cross-functional technical coordination** - integrate with data analysts, marketing ops, and BI teams
- **Executive technical visibility** - regular technical progress updates for business leaders

What Success Looks Like

Your deliverable should answer: "Can engineers build this streaming pipeline using your technical architecture? And how do you ensure it meets enterprise reliability and business requirements?"

A06 - Rapid Analytics Solution - Technical Architecture & Strategic Implementation

What You Need to Deliver - Technical Architecture FIRST

Design complete technical architecture for rapid-deployment analytics solution that provides immediate business value while proper infrastructure is being built. You must balance speed with technical quality.

- This is different from the proper long term planning for A01+A02+A03 (Batch data) and A05 (Streaming Data)

Task Clarity

- **Speed over perfection** - deploy in days/weeks, not months
- **Temporary nature** - designed to be replaced by a full pipeline later (like A01+A02+A03, or A05)
- **Business priority** - satisfy immediate analytics requests while buying time for proper infrastructure
- **Flexibility focus** - easily accommodate various business team requests
- **Your role** - design a pragmatic solution that balances speed with functionality

Required Technical Deliverables

- **Rapid deployment architecture** - lightweight but functional system design optimized for quick implementation
- **Multi-source integration specifications** - flexible data ingestion from streaming and batch sources
- **Processing pipeline design** - simplified but effective data transformation and aggregation logic
- **Dashboard architecture** - business-friendly analytics interface with self-service capabilities
- **Technical deployment plan** - implementation timeline optimized for speed, configuration procedures
- **Migration architecture** - clear technical transition plan to full enterprise pipeline later

Technical Components (Specifications)

- **Data integration architecture** - APIs, connectors, data format handling, error management
- **Processing simplification** - effective analytics without over-engineering, maintainable code structure
- **Storage optimization** - database design for rapid queries, appropriate indexing, reasonable scalability
- **User interface design** - intuitive dashboards, self-service analytics, business-friendly visualizations
- **Monitoring essentials** - basic system health, data quality checks, performance tracking

Leadership Coordination (Added Layer)

- **Business stakeholder expectation management** - balance rapid delivery with quality considerations
- **Technical resource optimization** - efficiently allocate engineering time between rapid solution and strategic work
- **Strategic planning integration** - ensure rapid solution doesn't compromise long-term technical architecture

- **Business value demonstration** - prove analytics value to justify enterprise infrastructure investment
- **Technical timeline accountability** - deliver immediate business value while maintaining stakeholder confidence

What Success Looks Like

Your deliverable should answer: "Can engineers implement this rapid analytics solution quickly using your technical specifications? And how do you maintain business confidence while building toward the proper enterprise solution?"

List B - Optional Enhancement Tasks

▼ Leadership Development and Strategic Analysis Tasks (Additional Credit)

B01 - Vector Database Tutorial - Team Training Leadership

Task Description

- **Team training material creation** for vector database technology using GenAI tools
- **Leadership focus** - develop training materials you would use to upskill your data engineering team
- **Knowledge transfer planning** - structured approach for team learning and capability building
- **Business context integration** - connect technical concepts to business use cases

Deliverable Requirements

- **Team training curriculum** - structured learning path for data engineers
- **Business use case analysis** - vector database applications for business problems
- **Technical tool evaluation** - comparison guide for team technology decisions
- **Implementation mentoring guide** - how you would guide team through first vector database project
- **Knowledge retention strategy** - approaches for ensuring team retains and applies learning

B02 - LiteLLM and LangGraph Analysis - Technology Leadership

Task Description

- **Strategic technology evaluation** for LiteLLM and LangGraph as a Tech Lead
- **Team capability planning** - assess tools for team productivity and project applications
- **Business value analysis** - connect technology capabilities to business objectives
- **Implementation leadership planning** - how you would lead team adoption of these tools

Deliverable Requirements

- **Technology leadership evaluation** - strategic analysis of both tools for team productivity
- **Team adoption roadmap** - plan for introducing tools to engineering team
- **Business application analysis** - specific use cases that deliver business value
- **Training and mentoring strategy** - how to develop team expertise in these technologies
- **Integration planning** - how tools fit into existing development workflow and architecture

B03 - LLM Fine-tuning Guide - Advanced Technical Leadership

Task Description

- **Advanced technical leadership tutorial** for LLM fine-tuning projects
- **Team technical development** - guide for developing team's ML/AI capabilities
- **Business case evaluation** - when fine-tuning provides business value vs alternatives
- **Project management integration** - planning and managing fine-tuning projects

Deliverable Requirements

- **Technical leadership guide** - comprehensive fine-tuning strategies for team projects
- **Business case framework** - decision-making guide for fine-tuning vs alternatives
- **Team development roadmap** - building team ML/AI capabilities through fine-tuning projects
- **Project management integration** - timeline, resource, and risk management for fine-tuning projects
- **Quality assurance framework** - testing and validation procedures for fine-tuned models

B04 - Product-UIUX-Designer Team Analysis for Data Tech Lead Collaboration

Task Description

- **Comprehensive analysis of Product-UIUX-Designer team structures from DTL perspective** - understand how product teams operate to enable effective data collaboration
- **Data-Product integration strategies** - how DTL coordinates with product teams to deliver data-driven features and analytics
- **Cross-functional workflow optimization** - DTL's role in product development lifecycle and design processes
- **Data requirements gathering** - how DTL interfaces with product teams to understand data needs and deliver technical solutions

Deliverable Requirements

- **Product team structure analysis** - Product Manager, UI Designer, UX Designer, UX Researcher roles from DTL collaboration perspective
- **Data-Product workflow integration** - how DTL coordinates with product teams during feature development, A/B testing, and analytics implementation
- **Cross-functional communication frameworks** - DTL strategies for effective collaboration with Product, Design, and Engineering teams
- **Data requirements translation** - how DTL translates product team needs into technical data architecture and infrastructure requirements
- **Project lifecycle coordination** - DTL involvement in product roadmap planning, sprint cycles, and feature delivery from data infrastructure perspective
- **Success measurement frameworks** - how DTL establishes metrics and KPIs that align data delivery with product success metrics
- **Stakeholder management strategies** - DTL approaches for managing expectations and communication with product stakeholders
- **Tool and process alignment** - how DTL coordinates data tools and processes with product team workflows and design systems

B05 - Product Management Office (PMO) Framework for Data Tech Lead Operations

Task Description

- **Comprehensive PMO analysis for DTL project management** - understanding PMO methodologies to improve data project delivery
- **Data project management framework** - adapting PMO best practices for data infrastructure and analytics project management
- **Cross-departmental coordination strategies** - how DTL leverages PMO principles to coordinate with engineering, product, business teams
- **Resource and timeline management** - PMO approaches that DTL can adapt for data team and infrastructure project management

Deliverable Requirements

- **PMO methodology adaptation** - how DTL adapts PMO frameworks (portfolio management, resource planning, process standardization) for data projects
- **Data project governance** - PMO-inspired governance frameworks for data infrastructure projects, analytics initiatives, and team coordination
- **Cross-functional project coordination** - DTL strategies for coordinating data projects with Engineering, Product, Marketing, Finance using PMO principles
- **Resource allocation optimization** - PMO-based approaches for DTL to manage data team resources, infrastructure budget, and technology investments
- **Performance measurement systems** - adapting PMO KPI tracking and project success metrics for data infrastructure and team productivity
- **Stakeholder communication protocols** - PMO communication frameworks adapted for DTL to manage technical and business stakeholder expectations
- **Risk management and project recovery** - PMO risk management strategies applied to data infrastructure projects and technical crisis management
- **Process standardization for data teams** - how DTL implements PMO-style process standardization for data engineering workflows and project delivery