

COGNIRA | retsci

End of studies projects 2026

Tunisia





Cognira is the leading promotion management solution provider for retailers and wholesalers, making it easy to collaborate across departments, analyze past performance and run smarter, more effective promotions.

Our single promotion solution – PromoAI – leverages data science and AI to effectively manage the entire end-to-end promotion lifecycle.



2020 2021 2022



2019 2020



2023 2024



2023

Trusted by the world's leading retailers and wholesalers





Back-end :

State Reachability Verification

Natural Language Query Interface for Logs

Distributed Rate Limiting Solution

Smart Monitoring & Incident Assistant

Front-end :

People management platform

DevConnect

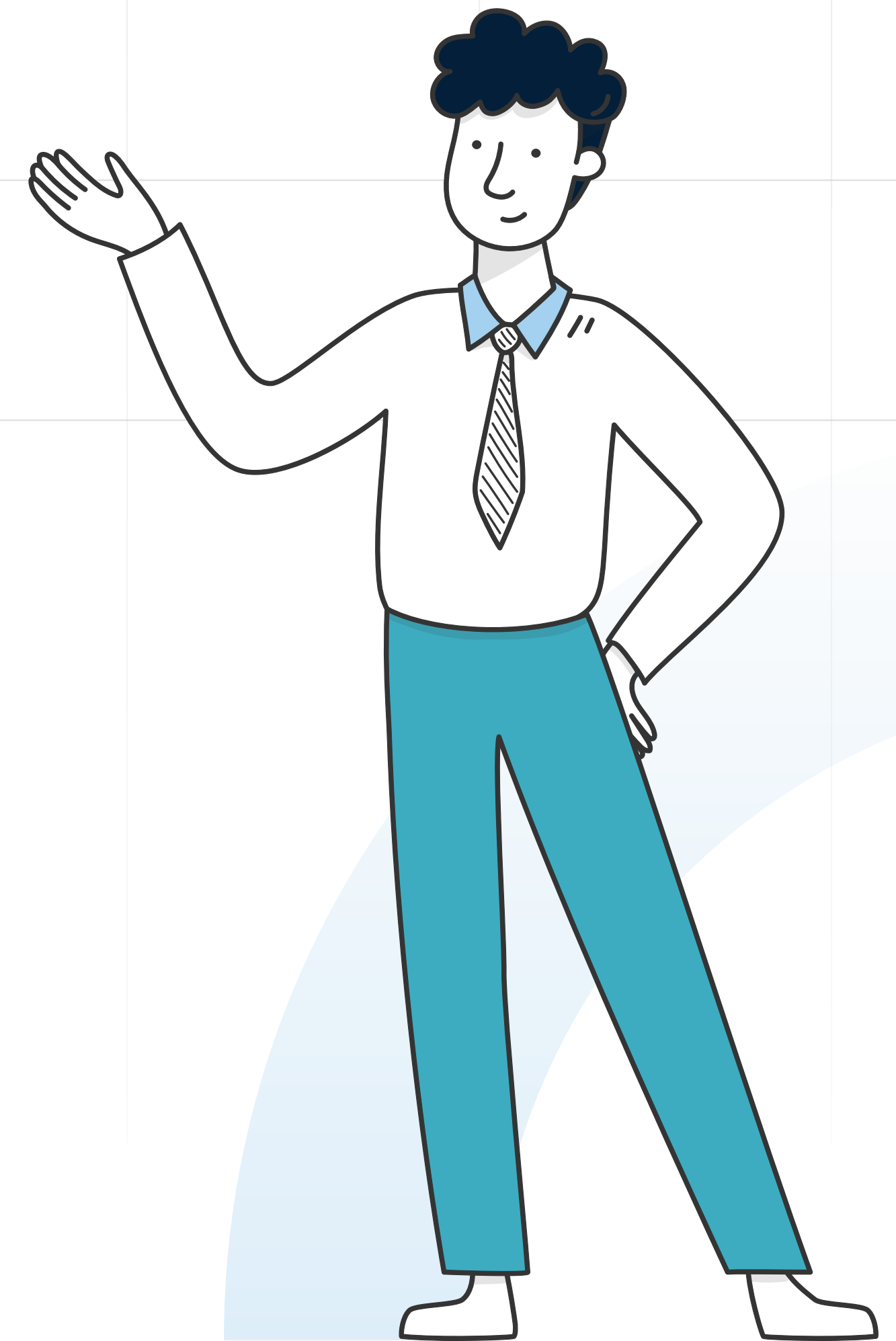
Configurations Studio

Codebase Dependency Analyzer

Quality Assurance :

Forecast Metrics Validation and Reliability Analysis Framework

Design of a Multi-Configurable, High-Performance UI Automation Framework for Cognira's SaaS Platform





Data Engineering :

Automated Data Pipeline Deployment and Optimization Framework using Databricks Bundles

Automated ETL Pipeline Deployment and Management Framework using Databricks REST API

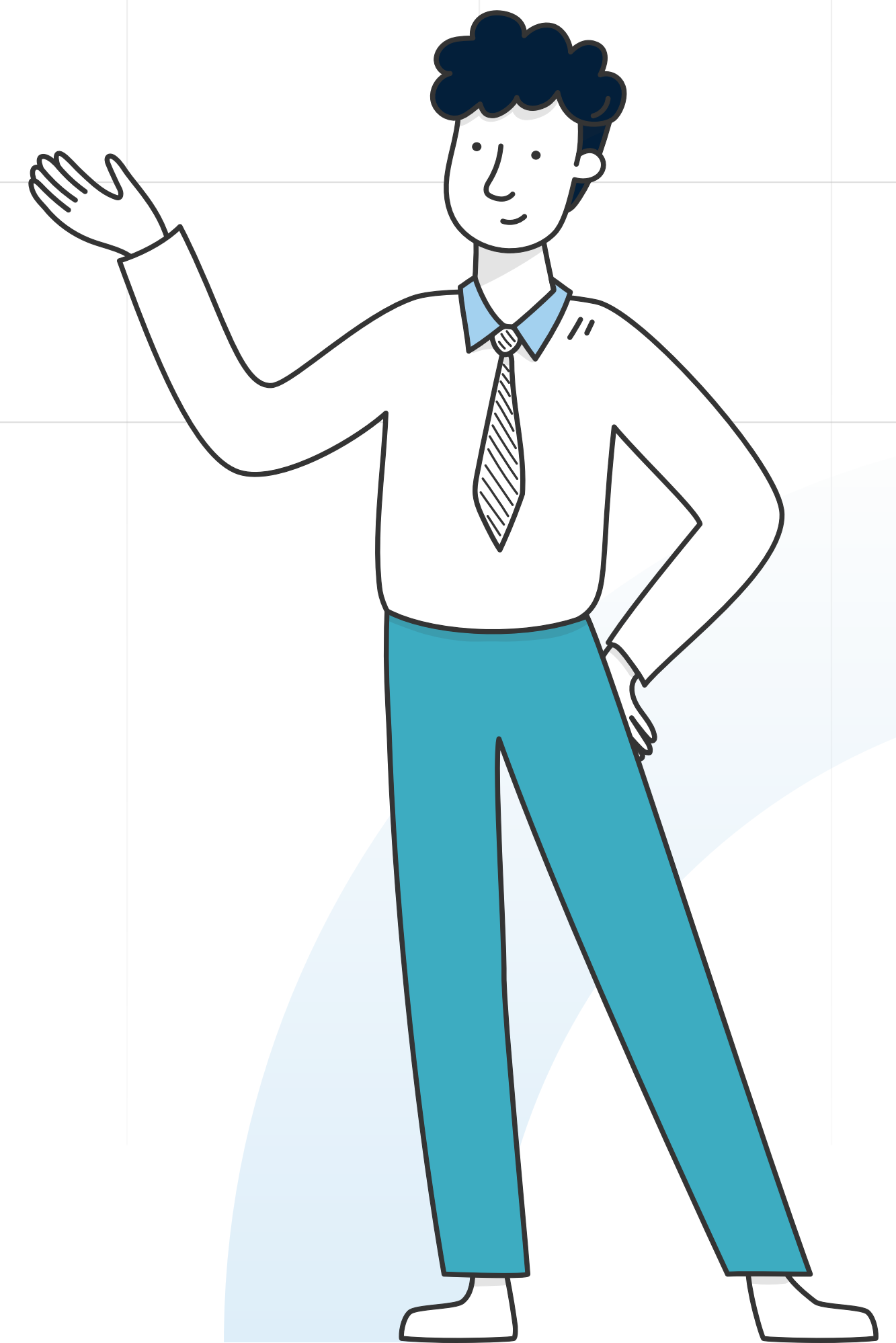
Data Collection API for Clients and File-Level Data Validation

Data Science :

Tiny Time Mixers for Demand Forecasting

Advanced Modeling of Intermittent and Sparse Demand

LLM-Driven Feature Engineering for Knowledge-Augmented Retail Data





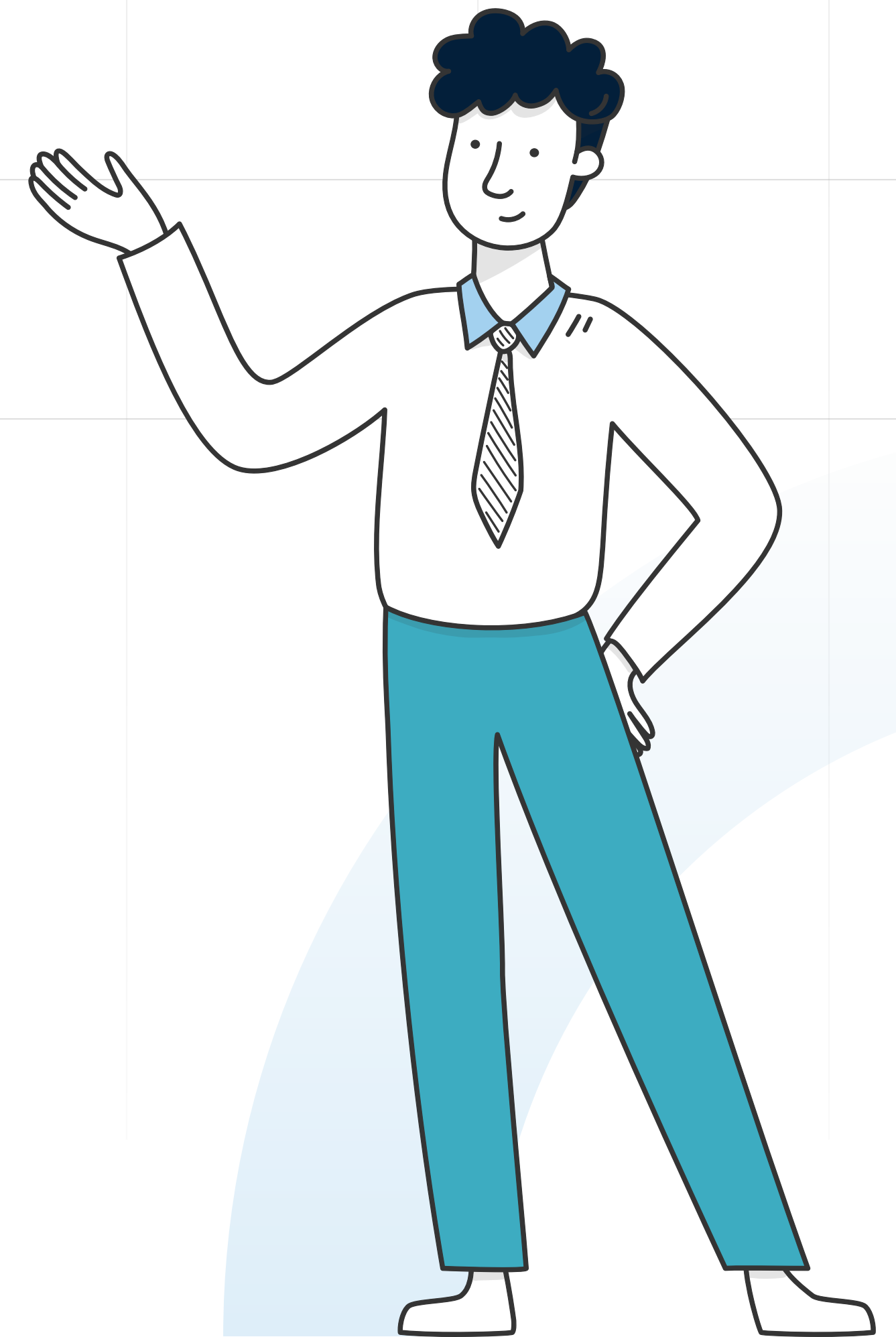
internships Topics :

Devops :

Azure Infrastructure Automated Disaster Recovery

Client Engagement and Delivery :

PromoAI Intelligent Assistant for Ticket Automation and Issue Analysis



State Reachability Verification

Project Description :

This project proposes the design and implementation of a State Verification system. The primary goal is to shift entity property testing from manual case-by-case definition to a rigorous, model-based approach. The system will take a defined set of entities, their initial states, and a catalog of possible actions as input. It will then automatically synthesize the sequence of actions, the "action path", required to reach a desired or hypothetical target state, thereby verifying its reachability. This is the question that this project answers "Given a target state (a complex "property" of the entity), can the system find a sequence of actions (a complex "input") that causes the entity to land in that state?"

Technologies:

Scala, JS, Docker, Kubernetes, UI (svelte / react)

Natural Language Query Interface for Logs

Project Description :

Developers and ops teams use complex query languages (like OpenSearch DSL) to filter logs. This project builds a backend interface that translates natural-language questions into structured queries, enabling conversational search.

Technologies:

Backend: FastAPI / Scala	AI: LangChain + GPT / Llama-3
Search: OpenSearch	Optional UI: Minimal web or CLI interface

Distributed Rate Limiting Solution

Project Description:

The goal of this project is to design and implement a distributed rate limiting service that can be integrated with Cognira’s PromoAI platform. The system should support both enforcement and “warning-only” modes, allowing flexible rule definitions based on clients, users, roles, or specific features to manage costly or high-frequency operations. The rate limiter should not become a bottleneck itself; it must be scalable and maintain low-latency performance. It should also provide detailed observability into rule evaluations, request rejections, and overall performance.

Technologies:

Scala, PostgreSQL, Redis, Kubernetes, Prometheus, Grafana

Smart Monitoring & Incident Assistant

Project Description:

This project creates a backend service that monitors logs and metrics, correlates them, and auto-summarizes incidents using AI.

Technologies:

Backend: Scala + Akka HTTP or FastAPI	Processing: PySpark, Pandas
Data Sources: OpenSearch + Prometheus	AI: GPT-based summarizer or fine-tuned model

People management platform

Project Description:

Implementing a people management tool, inspired by platforms like Lattice, designed to centralize and streamline HR functions such as performance and feedback management, growth and career tracking, employee engagement and recognition, and HR analytics and insights.

In addition to these features, the intern can explore the possibility of integrating other creative and AI-powered capabilities, such as:

- Automated Time Off Nudge: detects signs of burnout suggests appropriate rest periods.
- Sentiment Analytics: analyzes feedback and survey responses.
- Chat based HR Assistant: provides quick answers to HR related questions.
- Retention Risk Prediction: identifies employees potentially at risk of leaving.

Technologies:

Frontend: React.js, Material UI, Chart.js

Backend: Node.js, Express.js

Database: PostgreSQL

AI(optional): Fine tuned LLM / ML Models

DevConnect

Project Description:

DevConnect is a platform that combines a developer forum and a mentorship system within our company. It allows developers to share knowledge, ask questions, and collaborate through department- or subject-based forums, similar to an internal Stack Overflow. Each user has a profile displaying their skills and role, while mentors have special profiles showing their expertise and availability. Developers can browse mentors, send mentorship requests, and mentors can accept or decline them. Once connected, both can track sessions, goals, and progress.

Technologies:

Frontend: React, TypeScript
Backend: Node.js, Express.js

Database: MongoDB (Mongoose), JWT Authentication, Socket.io (for real-time features), and optional integration with Google Calendar API.

Configurations Studio

Project Description:

Configurations Studio is an intelligent, user-friendly configuration management platform designed to revolutionize how organizations handle static configuration files. This project bridges the gap between technical complexity and business accessibility, empowering both developers and non-technical users to manage configurations efficiently and safely. [Expanded details](#)

Technologies:

Frontend: React, TypeScript
Backend: Node.js, Express.js

Database: MongoDB (Mongoose), JWT Authentication, Socket.io (for real-time features), and optional integration with Google Calendar API.

Codebase Dependency Analyzer

Project Description:

Designing a tool that enables UI engineers to visualize and analyze dependencies across application components, facilitating a clearer understanding of the interactions among different parts of the application and providing insights into areas of tight code coupling. It also identifies which APIs are used by each component and detects unused or misplaced functions. This will help the UI team to better understand the architecture, identify dead code or areas for refactoring and plan safer updates.

Technologies:

Frontend: React, TypeScript

Frontend: React.js, Material-UI

Data visualization libraries: 3D.js, Vis.js...

Backend: Node.js, Express.js

Database: NoSQL database (neo4j, MongoDB...)

AST parser: Babel, TypeScript Compiler API, Acorn...

Forecast Metrics Validation and Reliability Analysis Framework

Project Description:

Forecasting reliability plays a key role in evaluating the performance of retail promotions. This project focuses on building a user-friendly framework dedicated to validating and analyzing forecast metrics across various promotion hierarchies such as Products, Offers, Funds, and Events.

The framework empowers testers to ensure data accuracy and integrity by processing forecast data, recalculating metrics based on configuration rules, and identifying mismatches with their propagation across dependent metrics. It strengthens QA coverage by providing deeper visibility into the reliability of forecast outputs.

Key Objectives:

- Validate forecast metrics consistency and detect discrepancies early.
- Analyze mismatch propagation through metric hierarchies and dependencies.
- Provide testers with actionable insights into forecast reliability and data integrity.
- Visualize mismatches, reliability scores, and trends through interactive dashboards for better interpretation and reporting.

Technologies:

Backend: Node.js, Next.js

ORM & Database: Prisma, PostgreSQL

Data Visualization: Power BI

Versioning & Collaboration: Git/Bitbucket

Design of a Multi-Configurable, High-Performance UI Automation Framework for Cognira’s SaaS Platform

Project Description:

In Cognira’s PromoAI platform, one of the most vital pillars of product excellence is end-to-end automation testing. As the platform grows in complexity and scale, maintaining a reliable and performant UI automation ecosystem becomes increasingly challenging — especially across multiple tenants, configurations, and continuous delivery environments.

This internship focuses on the evolution of the existing automation framework into a next-generation, high-performance, and maintainable architecture. The goal is to enhance scalability, configurability, and stability within SaaS contexts, ensuring faster, more resilient, and smarter automation across CI/CD pipelines.

The project blends framework design, migration strategy, and performance optimization, aiming to strengthen the foundation of Cognira’s test automation stack and accelerate overall delivery speed and confidence in product releases.

Objectives:

- Assess and document the current automation framework, identifying performance and scalability gaps.
- Design a modular, multi-configurable framework architecture supporting:(Multiple tenants, configurations, and environments, Centralized configuration management and test data abstraction,Parallel execution and environment-specific customization)
- Migrate existing test assets to the new design while preserving functional coverage and improving execution performance.
- Integrate with CI/CD pipelines (e.g., Jenkins) to enable continuous test execution, monitoring, and automated reporting.
- Optimize execution speed and reliability through parallelization, resource reuse, and selective test execution strategies.

Technologies:

Programming language: Python	Containerization: Docker
Test automation tools: Selenium, Playwright, BDD	Reporting framework: Playwright Test
framework: Cucumber (Gherkin),	Reporter and Allure
CI/CD tools: Jenkins	
Version control: Git (Bitbucket)	

Automated Data Pipeline Deployment and Optimization Framework using Databricks Bundles

Project Description:

The goal of this internship project is to design and implement an automated ETL pipeline framework on Databricks that can be easily deployed, configured, and optimized using Databricks Bundles.

The student will work on building data ingestion and transformation pipelines, implementing Spark optimizations, and setting up continuous integration and deployment (CI/CD) to automatically deploy and monitor these pipelines across environments.

Design a modular ETL pipeline on Databricks to process sample retail data (e.g., sales, products, and promotions).

- Ingest data from multiple sources
- Apply transformations
- Load processed data into a Delta Lake

Integrate deployment using Databricks Bundles

- Define workflows and jobs using Databricks YAML configurations.
- Enable parameterized deployments for multiple environments.

Integrate CI/CD automation

Use Bitbucket Pipelines or Jenkins to:

- Deploy and update jobs
- Automatically promote configurations between environments/clients.

Implement Spark performance optimizations

- Benchmark different optimization techniques

Technologies:

Databricks
Spark
Delta Lake
Python

Automated ETL Pipeline Deployment and Management Framework using Databricks REST API

Project Description:

The goal of this internship project is to design and implement an automated ETL pipeline framework on Databricks that can be programmatically deployed, configured, and monitored using the Databricks REST API.

Design a modular ETL pipeline on Databricks

- Process sample retail data (e.g., sales, products, and promotions).
- Ingest data from multiple sources
- Transform data
- Load processed results into a Delta Lake

Automate job and environment management using Databricks REST API

- Use Databricks REST API to create and configure clusters, jobs, and notebooks dynamically.
- Automate pipeline deployment, triggering, and cleanup without manual interaction.
- Parameterize environments via JSON configurations.
- Store and manage credentials securely

Implement Spark performance optimizations

- Benchmark different optimization techniques

Integrate CI/CD automation

Use Bitbucket Pipelines or Jenkins to:

- Deploy and update jobs via REST API calls.
- Automatically promote configurations between environments/clients.
- Run integration tests to validate successful deployments.

Technologies:

Databricks
Spark
Delta Lake
Python

Data Collection API for Clients and File-Level Data Validation

Project Description:

This project is to architect and build a scalable, secure, and automated ingestion gateway. This API serves as the primary, governed entry point for all file-based data, allowing external clients to programmatically push data directly into the data ecosystem.

Develop a secure File Ingestion API:

- Perform configurable validation checks on uploaded files, such as ensuring files are not empty or corrupted (additional checks can be added as needed).
- Automatically copy validated files to Azure Blob Storage.
- Support auto-renaming of files based on configurable naming conventions for each file type

Generate a manifest file capturing metadata for all uploaded files, including:

- File name and type
- Upload timestamp
- File size
- Validation results and status

Provide full traceability and auditing for all client file uploads.

Build a configurable, maintainable, and production-ready backend service

Technologies:

Python
Docker
Azure Data Lake

Tiny Time Mixers for Demand Forecasting

Project Description:

Context:

Large pre-trained models have been successful at a variety of language and vision tasks and similar (although not equally large) models have been pre-trained from a diverse corpus of data for doing multivariate timeseries forecasting. These might be employed for retail demand forecasting either immediately (as zero-shot forecasting) or after fine-tuning.

Objectives:

Build a data pipeline to take structured input data containing sales and promotional activity and convert it to a multivariate input appropriate for timeseries models.

Build a demand forecasting model based on the IBM TSMixer or Google TSMixer backbone.

Evaluate model performance versus traditional regression approaches and with zero-shot, few-shot, and full-shot forecasting.

Technologies:

Python, Pandas, Scikit-learn, Tensorflow, Pyspark

Advanced Modeling of Intermittent and Sparse Demand

Project Description:

Problem Overview: Optimizing Forecasts for Lumpy Demand

A significant challenge in inventory management is accurately forecasting demand for products with intermittent or sparse sales patterns (i.e., long periods of zero demand interspersed with lumpy, non-zero orders). Traditional forecasting methods (like standard ARIMA or simple smoothing) fail spectacularly in these scenarios, leading to excessive inventory holding costs or critical stockouts.

This project aims to develop and rigorously test advanced sequential modeling techniques specifically designed for these sparse datasets. The core goal is to build a robust model that accurately predicts both the timing of the next demand event and the magnitude of that demand, while explicitly modeling the "state" of the product (e.g., active, dormant, or trending).

Key Research Objectives:

The intern will design and implement a comparative analysis and develop a Deep State-Space prototype.

Benchmarking and Selection: Conduct a comparative analysis of established intermittent demand models, including Croston's Method variations (e.g., TSB, ADIDA) and Zero-Inflated models (e.g., Zero-Inflated Poisson or Negative Binomial), to establish a performance baseline on real-world sparse demand data.

Sequential State Modeling (The "Fuzzy States" Extension): Design and implement a sequential model that formally captures the underlying "fuzzy state" of a product. This will involve exploring and applying:

Hidden Markov Models (HMMs) or Switching State-Space Models to model the probability of a product transitioning between different demand regimes (e.g., "dormant" to "active-trending").

Deep State-Space Models (DSSMs), leveraging Recurrent Neural Networks (RNNs) like LSTMs or GRUs to learn the complex, non-linear dynamics of the demand process.

Two-Part Forecasting: Focus on a two-part forecasting mechanism:

Predicting the demand interval/timing (when the next order will occur).

Predicting the non-zero demand size (how large the order will be).

Metric Validation: Establish appropriate performance metrics for sparse data, focusing on accuracy at both the inventory level (e.g., Mean Squared Error on the total forecast) and the specific event level (e.g., Intermittency Error or a custom cost-based metric).

Technologies:

Python, Pandas, Scikit-learn, Tensorflow, Pyspark

LLM-Driven Feature Engineering for Knowledge-Augmented Retail Data

Project Description:

Problem Overview: Generating Contextual Features from Unstructured Data

Traditional retail forecasting relies heavily on structured historical data (sales, prices, promotions) and often fails because it ignores critical external factors—such as competitor campaigns, emerging trends, shifting sentiment, and macroeconomic events—which are contained in unstructured text and contextual sources.

This project focuses on the foundational, high-value task of integrating these external and contextual factors into the structured historical data. The core challenge is to transform noisy, text-based knowledge into concrete, quantifiable exogenous feature vectors that are compatible with existing forecasting systems. The deliverable is a robust data augmentation pipeline that significantly enriches the historical time-series data, making it "forecasting-ready."

Key Research Objectives:

The intern will design and implement a Knowledge-Augmented Feature Pipeline (KAFF) prototype focused on feature generation and data integration.

Contextual Data Sourcing and Integration: Define the architecture to ingest and integrate diverse unstructured data sources (e.g., news feeds, social media summaries, competitor announcements, weather reports) relevant to specific product categories and time periods.

LLM-Powered Retrieval and Analysis: Implement a Retrieval-Augmented Generation (RAG) pipeline to efficiently query and extract relevant knowledge, specifically focusing on information pertaining to the historical context of past sales or the target period for future forecasts.

Exogenous Feature Generation (Core Objective): Develop and implement robust Large Language Model (LLM)-based prompt engineering and reasoning chains to generate concrete, quantifiable, time-stamped features from the retrieved knowledge. The focus is on creating measurable predictors that can be joined to the structured historical data:

- "Negative Sentiment Score" for Product Category X (e.g., daily/weekly score).
- "Emerging Trend Volume" for Style Y (e.g., an index of relative increase in mentions).
- "Competitive Activity Index" (a quantified measure of detected competitor campaigns).

Data Harmonization and Output: Design the final output schema to seamlessly join the generated exogenous features with the existing structured historical data (sales, price, promo type) at the appropriate time granularity (e.g., daily or weekly), resulting in a unified, "knowledge-augmented" dataset ready for model training.

Technologies:

Python, Pandas, Scikit-learn, Tensorflow, Pyspark

Azure Infrastructure Automated Disaster Recovery

Project Description:

This project focuses on designing and implementing an automated disaster recovery (DR) solution for Azure Kubernetes Service (AKS) environments. The intern will build workflows to orchestrate failover and restore processes for clusters and workloads, both within the same region and across regions. Key goals include minimizing downtime, ensuring consistency for stateless and stateful workloads, and integrating monitoring and alerting for DR readiness.

As an optional extension, the project can be expanded to include Azure Databricks workspace recovery, ensuring analytics and data workloads can also be restored automatically in disaster scenarios.

Technologies:

Azure, Kubernetes, Argo CD, Git

PromoAI Intelligent Assistant for Ticket Automation and Issue Analysis

Project Description:

Within the Client Engagement & Delivery department, a significant amount of time is spent manually documenting internal tickets, refining user stories, and analyzing client-reported issues to identify reproduction steps. These repetitive manual tasks often lead to inconsistent documentation quality and longer resolution times.

To address this challenge, the PromoAI Smart Assistant project aims to design and implement an intelligent business assistant made up of multiple agents powered by Large Language Models (LLMs). To support these assistants, we need structured, clean data related to PromoAI, the company's internal solution, such as functionality documentation, Jira tickets, and configuration sheets, into a structured, standardized Markdown document aligned with the company's internal documentation format (for Confluence or Jira).

These datasets will then be used to train and support specialized AI agents capable of performing tasks such as:

- Ticket Matching: pairing new help desk tickets with known issues to improve Jira integrity and reduce duplicates.
- Configuration Identification: Detect whether client requests are already supported by current configurations.
- Request Clarity and Completeness: review tool functionality and identify questions and clarifications to flesh out a request.
- Feature Support: Evaluate tool behaviors to confirm expected results, find workarounds, or propose reproduction steps.
- Tone Consistency: Suggest client communication phrasing aligned with the Cognira brand and tone.
- Priority Suggestions: Propose issue priorities based on the business impact and user experience.

This project combines AI-driven process automation, business analytics, and knowledge management to create a solution that improves documentation quality, accelerates problem-solving, and enhances cross-team efficiency across the PromoAI ecosystem.

Project Objectives :

Enhance Operational Efficiency:

Reduce time spent on manual ticket documentation through AI-assisted automation.

Accelerate Root-Cause Analysis:

Use structured data and LLMs to quickly identify and suggest likely issue reproduction scenarios.

Ensure Consistency and Quality:

Implement standardized documentation practices that ensure clarity, traceability, and compliance across internal systems.

Leverage Data for Continuous Improvement:

Create a reusable framework that can be integrated into Jira or Confluence to capture learnings and improve future AI models.

Technology

Analytical Thinking: Ability to break down tickets/user stories and identify key components (context, issue, expected behavior).

Prompt Engineering: Design and refine prompts for structured LLM outputs (Markdown, scenarios).

Python Programming: Basic scripting, API usage (OpenAI API or LangChain), JSON handling.

Text Processing: Clean and structure unformatted text into organized Markdown sections.

Evaluation & Documentation: Assess output quality, write clear reports, and document findings.

Tools:

Python (Jupyter / VS Code)

OpenAI API or LangChain

Pandas, JSON, and Markdown libraries

Git/GitHub for version control

Nice-to-Have:

Knowledge of SaaS workflows or customer support processes

Basic experience with Jira or Confluence APIs

Interest in AI automation and explainability



retsci was founded as Cognira in 2015 by experienced data scientists and retail technology experts.

Since its inception, retsci has focused its attention in two areas: AI and ML based micro solutions and analytical solution implementations. The experience of our team has worked with many of today's Fortune 50 retailers.



2020 2021 2022



2019 2020



2023 2024



2023

Trusted by the world's leading retailers and wholesalers

HIBBETT

EST'D 1960
RURAL KING
AMERICA'S FARM & HOME STORE

See's
CANDIES

Bath & Body Works®



Data Science Consulting :

Natural Language Retail Assistant

Proactive Lost Sales Analytics

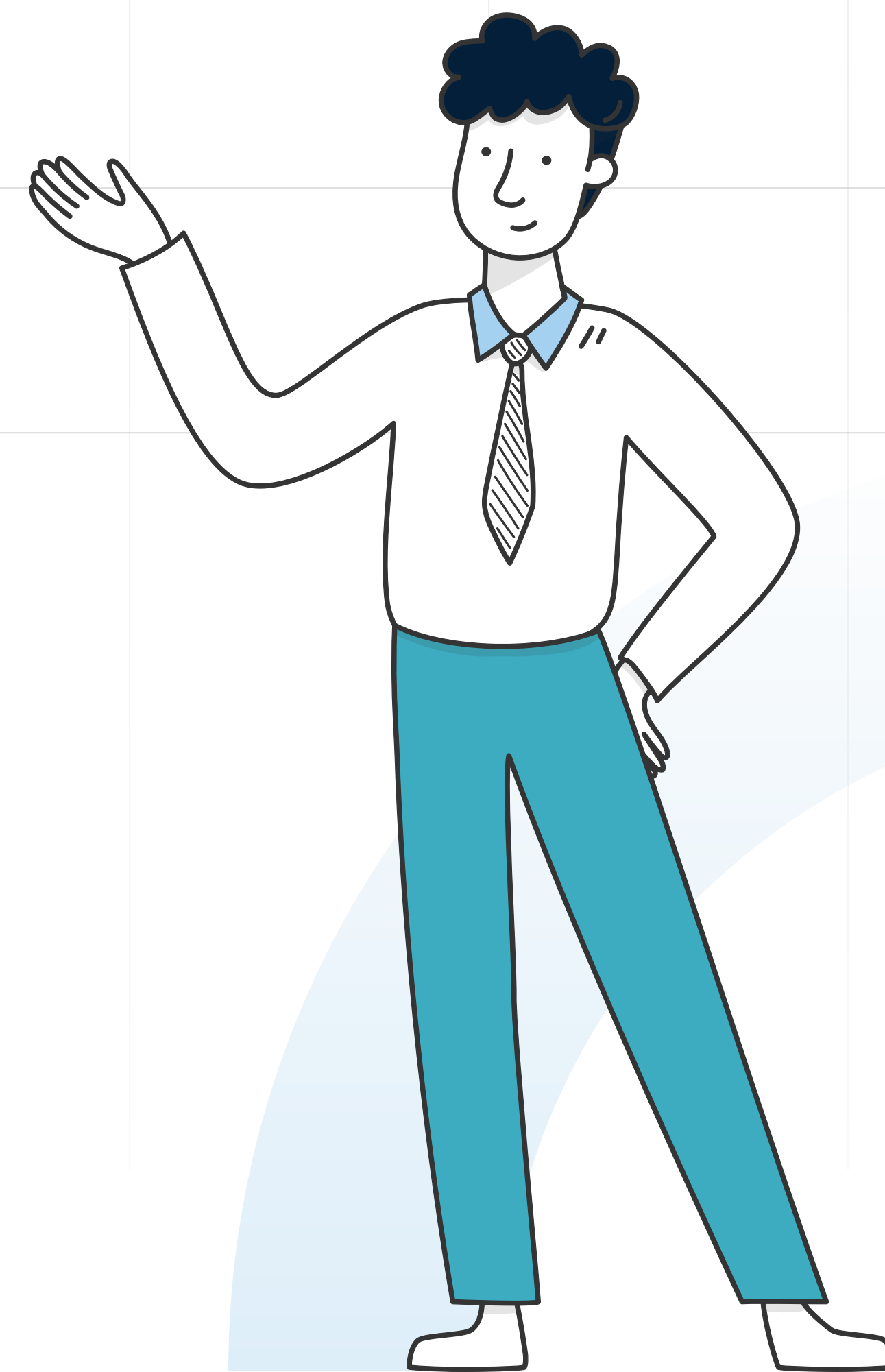
Intelligent Agent for Product Attribute Extraction

Consulting Team :

Developing an Assortment Planning Template on Oracle RPAS

Forecast Bias Diagnostics Tool for Retailers - Forecasting

Implementation of Constrained Replenishment Process for DC to Store Ordering



Natural Language Retail Assistant

Project Description:

A comprehensive virtual assistant designed for retail data analytics, composed of three core layers:

- 1. Text-to-Query:** Allows users to ask retail data questions in natural language, which are translated into structured queries across multiple retail tables (e.g., products, sales, inventory, promotions, store locations). Automatically handles joins, filters, and aggregations, enabling fast access to insights without SQL or technical knowledge.
Example question: “Which product sold the most in Q1 2025?” or “Which store had the highest inventory turnover last month?”
- 2. What-If Simulator:** Uses precomputed ML coefficients to estimate the impact of hypothetical business changes. For example, a user could simulate the effect of increasing the price of a product or changing promotion strategy, with the system updating expected metric outcomes based on modified feature values.
- 3. Recommendation Generator:** Produces natural language suggestions by mapping data features to actions. These recommendations help guide business decisions such as stock optimization or promo planning.

Technologies:

Python, Pandas, NumPy, SQL, NLP, LLMs
Optional: Azure Databricks, PySpark

Proactive Lost Sales Analytics

Project Description:

The Lost Sales Analytics project aims to detect, estimate, and prevent sales lost due to stockouts across retail operations. It provides both retrospective insights and proactive guidance for inventory decision-making.

The solution is composed of three layers:

Stockout Detection & Retrospective Analysis: Analyzes historical sales, demand, and inventory signals to identify when and where products were out of stock.

Lost Sales Estimation & Data Correction: Uses predictive models to estimate the sales that would have occurred without stockouts, based on demand history, product attributes, and seasonality. These estimates are applied to correct demand signals and data gaps, ensuring accurate analytics and planning.

Proactive Lost Sales Prevention: Extends into future planning by helping retailers determine optimal order quantities based on forecasts. The system evaluates whether to order equal to, more than, or less than the forecasted demand. This decision is informed by the risk of stockouts and the cost of excess inventory (e.g., spoilage or storage costs), enabling precision ordering that balances missed sales versus stale stock.

The final deliverable includes analytical interfaces and data pipelines for stockout detection, lost sales estimation, historical correction, and proactive ordering intelligence.

Technologies:

Python, Pandas, NumPy, SQL
Optional: Azure Databricks, PySpark

Intelligent Agent for Product Attribute Extraction

Project Description:

The Smart Attributes Scraper project focuses on automatically discovering and extracting structured product attributes from client messy data or websites across diverse retail categories such as fashion, hunting, firearms, and grocery. Unlike traditional scrapers that rely on predefined URLs or HTML structures, this system leverages an AI-driven agent to dynamically navigate sites, identify relevant product pages, and detect key attributes without manual configuration.

For example, the agent may extract:

- Firearms: caliber, action type, barrel length, magazine capacity, material, weight
- Hunting Gear: type (e.g., binoculars, tents, knives), size, material, intended use, brand
- Grocery Products: brand, flavor, weight, ingredients, nutritional info, packaging type

The extracted data is cleaned, standardized, and structured for downstream integration into recommendation systems, analytics models, or inventory enrichment pipelines.

Technologies:

Python, Pandas, NumPy, SQL, LLMs APIs, web scraping/navigation
Optional: Azure Databricks, PySpark

Developing an Assortment Planning Template on Oracle RPAS

Project Description:

Assortment Planning (AP) is the critical pre-season process of deciding which products a retailer will sell, in which stores, and in what quantities. It is the strategic blueprint that balances financial targets

This project places you at the center of this strategic function, giving you the responsibility to build the technological tool that planners use to make these multi-million dollar decisions.

You will be responsible for designing and building a foundational Assortment Planning template from the ground up, focusing on the critical configuration elements and scenario-based planning capabilities that drive modern retail.

Project Goals: Your Key Learning Objectives

- **Understand Assortment Planning Fundamentals:** Gain a deep understanding of the pre-season assortment planning process, including hind-sighting, assortment strategy, shopping list creation, and buy plan development.
- **Master RPAS Configuration:** Learn to configure the Oracle RPAS platform, including hierarchies, measures, rules, and administrative workbooks.
- **Develop a Functional AP Template:** Create a working AP template that addresses core business requirements for assortment planning.
- **Implement Scenario-Based Planning:** Incorporate "what-if" scenario capabilities to enable flexible analysis and decision-making.

Technologies:

Oracle RPAS

Forecast Bias Diagnostics Tool for Retailers - Forecasting

Project Description:

Accurate demand forecasting is vital for supply chain efficiency. However, forecasts often suffer from systematic bias consistent over- or underestimation of demand which impacts replenishment, inventory, and planning decisions.

This project aims to develop a Forecast Bias Diagnostics Tool using the RELEX Solutions Forecasting Module/Platform. The tool will detect, visualize, and explain bias patterns over time, helping demand planners and the Project Implementation team continuously improve model performance and decision-making.

Objectives:

- Detect and quantify recurring forecast bias (over/under-forecasting).
- Analyze potential root causes (seasonality, promotions, planner overrides).
- Visualize forecast bias and accuracy KPIs over time within RELEX.
- Support continuous model improvement within RELEX.

Technologies:

RELEX Platform

Implementation of Constrained Replenishment Process for DC to Store Ordering

Project Description:

Retailers usually have a network of stores that are supplied by warehouses (Distribution centers - DC). RELEX Replenishment system suggests daily orders from DC to store and from Vendor to DC for every product/location combination. In a perfect scenario the DCs will always have inventory to support the stores but with unexpected supply chain disruptions, irregularities in demand and defect products, the DCs will not be able to always support the orders of the stores on a given day.

In this case the retailers need to have a replenishment system that calculates orders based on real life scenarios and suggest constrained orders to supply instead of unconstrained orders. RELEX has a solution for this process but it is fairly new and limited.

This project will consist of an end to end implementation of constrained replenishment processes. This implies setting up store and DC replenishment workflows. As a second step the intern will be conducting a testing strategy to validate results of constrained projections and creating an analysis of the limitations of this functionality.

Technologies:

RELEX Platform

Apply at: 

<https://cognira.com/pfe-book-end-of-study-projects/>

