



## ***Automated Price Prediction Trading System***



**Planning Phase**



Public Works and  
Government Services  
Canada

Travaux publics et  
Services gouvernementaux  
Canada

**Canada**

## Table of Contents

Section 1.	Charter Introduction .....	2
1.1	Document Change Control .....	2
1.2	Executive Summary .....	3
1.3	Authorization.....	4
Section 2.	Project Overview .....	5
2.1	Project Summary .....	5
2.1.1	Project Goals, Business Outcomes and Objectives .....	5
2.1.2	Project Scope.....	6
2.1.3	Scope Definition.....	6
2.1.4	Boundaries .....	6
2.2	Milestones .....	7
2.3	Deliverables.....	7
2.4	Project Cost Estimate and Source of Funding.....	<b>Error! Bookmark not defined.</b>
2.4.1	Project Cost Estimate .....	<b>Error! Bookmark not defined.</b>
2.4.2	Source of Funding .....	<b>Error! Bookmark not defined.</b>
2.5	Dependencies.....	<b>Error! Bookmark not defined.</b>
2.6	Project Risks, Assumptions, and Constraints .....	8
2.6.1	Risks.....	8
2.6.2	Assumptions.....	9
2.6.3	Constraints .....	9
Section 3.	Project Organization.....	9
3.1	Project Governance .....	<b>Error! Bookmark not defined.</b>
3.2	Project Team Structure.....	<b>Error! Bookmark not defined.</b>
3.3	Roles and Responsibilities.....	<b>Error! Bookmark not defined.</b>
3.4	Project Facilities and Resources.....	9
Section 4.	Project References.....	<b>Error! Bookmark not defined.</b>
Section 5.	Glossary and Acronyms .....	10

## Section 1. Charter Introduction

### 1.1 Document Change Control

Revision Number	Date of Issue	Author(s)	Brief Description of Change
1.0	2026-01-24	Mbachu Emmanuel	Charter Creation

## 1.2 Executive Summary

**Project Overview** the Automated Price Prediction System is an advanced data analytics initiative designed to automate the complex process of financial market analysis and decision-making for individual traders. By leveraging supervised machine learning, the system acts as a digital mirror to a trader's existing strategies—predicting price movements for a variety of financial instruments, including currencies, assets, and securities. The solution provides a unified platform where a custom UI connects to a robust database, fetching data from multiple real-time sources to deliver actionable predictions. A centralized dashboard enables users to monitor account metrics, track trade formations across time zones, and validate strategy performance through automated win-rate comparisons (Passed vs. Failed trades).

**Initiation and Business Need** The project was initiated to bridge the gap between manual market analysis and high-speed execution. Traditional trading often suffers from human emotional bias and the inability to monitor multiple markets simultaneously. This project aims to improve speed, precision, and consistency by replacing manual oversight with a data-driven framework that adheres to predefined logical rules 24/7.

### 1.3 Authorization

This project charter formally authorizes the existence of the project, *<Project Name>*, and provides the project manager with the authority to apply organizational resources to project activities described herein. If there is a change in the project scope, the project charter will be updated and submitted for re-approval.

George Campanis

*Full name*

Date

Project Sponsor

*Position, Client Organization*

Mbachu Emmanuel

2026-01-24

*Full name*

Date

Project Manager

*Chief Data Officer, \_\_\_\_\_*

## Section 2. Project Overview

### 2.1 Project Summary

#### Background and Purpose:

The Automated Price Prediction System was initiated to solve the business problem of manual trading inefficiencies, such as emotional bias and limited market monitoring. By utilizing supervised machine learning to blend technical, sentimental, and fundamental data, the project transforms a manual process into a consistent, data-driven workflow. This project serves both as a school assignment to demonstrate mastery of the data analytics lifecycle and as a functional tool for real-world application.

#### Key Stakeholders and Benefits

**Project Manager (Self):** Benefits from a rigorous testing framework and an automated tool for personal account management.

**Project Sponsor (Professor):** Benefits from the successful demonstration of applied data analytics and documentation requirements.

**Target Organization:** The project outcomes will directly impact the user's trading account efficiency and school performance.

#### Core Project Highlights

**Goal:** Achieve a high-accuracy prediction engine (>65% win rate) using classified signals.

**Deliverables:** A Python-based UI, an MS Power BI performance dashboard, and full technical documentation (ERD/Data Dictionary).

**Schedule:** A phased approach from data architecture to demo testing, as detailed in the Gantt Chart.

#### 2.1.1 Project Goals, Business Outcomes and Objectives

No.	Goals	Objectives	Business Outcomes
1	<i>Automate Market Analysis</i>	<ul style="list-style-type: none"><li>• Develop a Python-based data pipeline to ingest technical, fundamental and sentimental data from at least two APIs</li></ul>	<ul style="list-style-type: none"><li>• Reduced manual screening time by 90%, allowing for 24/7 market monitoring without human fatigue.</li></ul>
2	<i>Optimize Predictive Accuracy</i>	<ul style="list-style-type: none"><li>• Implement a supervised classification model to generate trading signals with a validated back testing win rate of &gt;65% across, 1,000 historical setup</li></ul>	<ul style="list-style-type: none"><li>• Increased confidence in trading signals and a measurable reduction in "false positive trades"</li></ul>
3	<i>Enhance Performance Visualization</i>	<ul style="list-style-type: none"><li>• Create an MS Power BI dashboard to track real-time metrics, system "Passed/ailed" trade ration and account growth.</li></ul>	<ul style="list-style-type: none"><li>• Improved transparency of system performance for better decision-making and school assignment validation.</li></ul>

## 2.1.2 Project Scope

### 2.1.3 Scope Definition

The project scope involves the design, development, and validation of a modular Automated Price Prediction System. The final product will be a functional software pipeline characterized by the following features and functions:

- **Data Ingestion Layer:** Automated scripts to fetch historical and real-time market data (Price, Volume), sentiment data (News/Social), and fundamental indicators via REST APIs.
- **Research & Strategy Selection:** Conduct a formal review of established trading methodologies (e.g., Mean Reversion, Trend Following, Scalping) to identify confirmed setups suitable for automation.
- **Feature Engineering Engine:** Functions to calculate technical indicators (e.g., RSI, Moving Averages) and convert raw text into numerical sentiment scores.
- **Predictive Modeling Core:** A supervised machine learning model (Classification) designed to categorize market "setups" into Buy, Sell, or Hold signals.
- **Automated Validation Framework:** A backtesting module that evaluates the model's logic against a minimum of 1,000 historical data points to calculate win rates and risk metrics.
- **Interactive UI & Reporting:** A user interface for system configuration and an MS Power BI dashboard to visualize trade performance, system accuracy, and account metrics.

### 2.1.4 Boundaries

Activities In Scope	Activities Out of Scope
1. Data ingestion Layer	1. Web Scraping Sentimental data from sites (forex groups, news media etc.),
2. Research & Strategy Selection	2. Live Capital Execution: The system will not execute trades automatically its only limited to simulation environment only.  in the future: there is chance to be connected to a broker for a demo then a live trading execution.
3. Feature Engineering Engine	3. <b>Reinforcement Learning (RL):</b> The project is strictly limited to <b>Supervised Learning</b> (Classification/Regression). RL agents are out of scope.
4. Predictive Modeling Core	4. <b>High-Frequency Trading (HFT):</b> Low-latency infrastructure (nanosecond execution) is not required; the system focuses on minute-to-hour intervals.
5. Automated Validation Framework	5. <b>Mobile Application Development:</b> No native mobile app (iOS/Android) will be built; the UI is desktop/web-based only.
6. Interactive UI & Reporting: Python Interactivity with the database, Power BI Dashboard (interactivity with the database).	6.

## 2.2 Milestones

Project Milestone	Description	Expected Date
1. Strategy & Research Approval	Completion of the “Strategy Discovery” phase, identifying at least two confirmed trading strategies and “Ground Truth” rules for the model	Week 2
2. Data Architecture Finalization	Successful connection to APIs, creation of the SQL Database schema, ERD, Data dictionary, data model of the system	Week 4
3. Model Training and Validation	Completion of the Machine learning training phase (Classification Model) and execution of the 1,000-setup back test.	Week 8
4. System Integration & Dashboard	Final Integration of the Python/SQL backend with the Power Bi-Dashboard; system is fully operational for demo	Week 10
5. Final Project Presentation	Submission of the final report and presentation of the working system to the project Sponsor	Week 12

## 2.3 Deliverables

Project Deliverable 1: Project Charter & Schedule	
<b>Stakeholder:</b>	Project Sponsor
<b>Description:</b>	A formal document defining the project scope, objectives, and risks, accompanied by a detailed Gantt Chart created in MS Visio.
<b>Acceptance Criteria:</b>	Document is fully signed off; Gantt Chart clearly shows critical path and dependencies; all sections (1.1 - 3.0) are complete.
<b>Due Date:</b>	By Jan 2
Project Deliverable 2 : Technical Documentation (ERD & Data Dictionary)	
<b>Stakeholder:</b>	Project Sponsor
<b>Description:</b>	A comprehensive technical guide including the Entity Relationship Diagram (ERD) for the trade database and a Data Dictionary defining all model features (e.g., RSI, Sentiment Score)
<b>Acceptance Criteria:</b>	ERD must be normalized (3NF); Data Dictionary must define data types and sources for every variable used in the ML model.
<b>Due Date:</b>	By Week 4
Project Deliverable 3 Automated Trading System (Python Codebase)	
<b>Stakeholder:</b>	Project Manager (Self)
<b>Description:</b>	The functional Python scripts responsible for data ingestion, feature engineering, and generating "Buy/Sell" signals via the classification model.
<b>Acceptance Criteria:</b>	System must successfully fetch fresh data via API, process it, and output a prediction without manual error.
<b>Due Date</b>	By Week 10



Project Deliverable 4	Performance Dashboard (MS Power BI Report)
Stakeholder:	Project Manager and Project Sponsor
Description:	An interactive visual report connected to the database, displaying Key Performance Indicators (KPIs) like Win Rate, Total Trades, and Profit Factor.
Acceptance Criteria:	Dashboard must be interactive (drill-down capable); must clearly visualize the "System Win Rate" against the >65% target.
Due Date	By Week 12

## 2.4 Project Risks, Assumptions, and Constraints

### 2.4.1 Risks

No.	Risk Description	Probability (H/M/L)	Impact (H/M/L)	Risk Management Plan	OPI
1	API Rate Limiting: The free-tier API limit requests potentially blocking the system from fetching sufficient data for training	High	High	Implement a "Failover Mechanism" that automatically switches from Alpha Vantage to Yahoo Finance or Finnhub if a 429 (Rate Limit) error is detected.	Project Manager
2	<b>Model Overfitting:</b> The ML model may "memorize" historical data (100% backtest accuracy) but fail on new data, leading to a system that doesn't work in real-time.	Medium	High	<b>Mitigate:</b> Use "Cross-Validation" and separate data strictly into Training (70%) and Testing (30%) sets.	Project Manager

No.	Risk Description	Probability (H/M/L)	Impact (H/M/L)	Risk Management Plan	OPI
3	<b>Execution Latency:</b> The Python script may take too long to process sentiment data, causing the "Buy" signal to arrive after the optimal price point has passed.	High	Medium	<b>Eliminate:</b> Strictly adhere to the "MoSCoW" prioritization; drop the complex UI features if time runs short and focus only on the Model.	Project Manager

### 2.4.2 Assumptions

The following table lists the items that cannot be proven or demonstrated when this project charter was prepared, but they are taken into account to stabilize the project approach or planning.

No.	Assumptions
1	<b>Data Continuity:</b> It is assumed that the chosen APIs (e.g., Alpha Vantage or Polygon.io) will remain available and provide consistent data formats throughout the 12-week semester
2	<b>Predictive Correlation:</b> It is assumed that historical technical indicators and sentiment scores contain repeatable patterns that can be classified by a supervised learning model to predict future price direction.
3	<b>Software Interoperability:</b> It is assumed that Python-generated trade logs will seamlessly integrate with MS Power BI for real-time dashboard updates as required for the project deliverables.

### 2.4.3 Constraints

The following table lists the conditional factors within which the project must operate or fit.

No.	Category	Constraints
1	Time	The project must be completed, including the backtesting of 1,000 setups and the Power BI dashboard, within the current 12-week academic semester.
2	Scope	The logic is strictly limited to Supervised Learning (Classification or Regression); Deep Learning and Reinforcement Learning are excluded to ensure semester-long feasibility.
3	Budget	The project must be developed with zero financial cost, utilizing only free-tier APIs (e.g., Alpha Vantage, Yahoo Finance) and open-source software
	Technical	The data used for the predictive model must be updatable at least monthly via online downloads, webpage scraping, or official APIs.

## Section 3. Project Organization

## Glossary and Acronyms

*<Define all terms and acronyms required to interpret the project charter properly.>*

Term	Definition
Backtesting	The process of testing a trading strategy on historical data to see how it would have performed in the past.
Feature Engineering	The process of using domain knowledge to create variables (features) that help machine learning algorithms work better.
Acronym	Name in Full
API	Application Programming Interface