Feasibility Study Report

Project: Crypto Trading Bot Integrated with MetaTrader 5 (MT5)

# 1. Executive Summary

This feasibility study evaluates the development of a crypto trading bot integrated with MetaTrader 5 (MT5), allowing users to log in, start or stop the bot, view real-time market data, and access cryptocurrency-related news.  
  
Key Findings:  
Technical: MT5 provides APIs suitable for automated trade execution. The proposed cloud-based architecture is scalable and technically viable.  
Economic: Development and infrastructure costs are modest compared to the potential recurring subscription-based revenue.  
Operational: The system is easy to use and requires minimal training and maintenance.  
Legal: The project is legally feasible with compliance to KYC/AML, broker agreements, and data privacy regulations.  
  
Recommendation: Proceed with the project and begin the design and prototyping phase with a focus on risk mitigation, particularly for regulatory compliance, market volatility, and system security.

# 2. Introduction

## 2.1 Background

The cryptocurrency market is rapidly growing and highly volatile, creating opportunities for automated trading solutions that can outperform manual trading. This project proposes building a crypto trading bot integrated with MT5 to automate trading strategies and provide users with real-time market insights.

## 2.2 Objectives

* Enable users to start/stop a crypto trading bot seamlessly.
* Provide real-time market data and news updates.
* Execute trades with pre-defined stop-loss and take-profit strategies.
* Reduce human error and emotional bias in trading decisions.

## 2.3 Scope

This feasibility study assesses:

* Technical viability (architecture, integration, infrastructure).
* Economic viability (costs, revenue models, ROI potential).
* Operational viability (ease of use, training, support requirements).
* Legal/ethical considerations (broker compliance, regulatory constraints).
* Risk factors (market, security, and regulatory risks).

# 3. Problem Definition

## 3.1 Current Situation

Crypto traders often rely on manual execution, which is prone to delays and emotional decisions that can lead to financial losses.

## 3.2 Challenges

* Manual trading is slow compared to market fluctuations.
* Traders face decision fatigue and emotional bias.
* Lack of centralized tools combining data, news, and execution in one place.

## 3.3 Business Need

There is a need for an automated trading solution that executes strategies in real-time, provides market transparency, and simplifies decision-making for traders.

# 4. Proposed Solution

## 4.1 High-Level Description

Develop a cloud-hosted trading platform where users can:  
- Log in securely and manage their bot.  
- View market data and news feeds.  
- Configure or select pre-defined trading strategies.  
- Execute trades through MT5 automatically.

## 4.2 Alternative Solutions Considered

- Manual Trading: Too slow and inefficient for volatile markets.  
- Third-Party Bots: Lack of customization and control over strategies.  
  
The proposed in-house solution provides better control, scalability, and integration with user requirements.

# 5. Feasibility Analysis

## 5.1 Technical Feasibility

System Architecture:  
- Frontend: Web or mobile application for user interaction.  
- Backend: Python-based trading bot integrated with MT5 via its Python API.  
- Data Services: Market data from MT5 brokers or third-party APIs (Binance, CoinGecko), news aggregation via RSS feeds.  
- Database: PostgreSQL/MySQL/Firebase for authentication and storage.  
- Deployment: Hosted on AWS/GCP/Azure for scalability.  
  
Technical Challenges & Solutions:  
- Real-Time Data Latency: Use WebSockets or MT5 socket streaming.  
- Order Execution Reliability: Implement retry mechanisms.  
- Scalability: Use load balancers and microservices.  
- Security: SSL/TLS encryption, API key vaulting, RBAC for users.  
  
Conclusion: The project is technically feasible given the available APIs, cloud infrastructure, and team expertise.

## 5.2 Economic Feasibility

Estimated Development Costs:  
- 1–2 Python developers (backend & trading logic)  
- 1 frontend developer (React/Flutter)  
- 1 DevOps/Cloud engineer  
- Part-time QA/tester  
  
Timeline: 4–6 months (prototype), 9–11 months (full release)  
  
Infrastructure Costs:  
- Cloud hosting: $50–200/month  
- Database & storage: $20–50/month  
- Premium APIs: $100–300/month  
  
Revenue Models:  
- Subscription-based model (monthly fee)  
- Freemium (basic features free, premium strategies paid)  
- Revenue-sharing (percentage of user profits)  
  
Conclusion: The project is economically viable with moderate upfront costs and scalable recurring revenue potential.

## 5.3 Operational Feasibility

- Ease of Use: Simple interface with start/stop bot functionality.  
- Automation: Reduces manual workload and emotional bias.  
- Support Requirements: Regular monitoring of servers and algorithm updates.  
- Training: Minimal training required; users only need guidance on risk settings.  
  
Conclusion: Operationally feasible with manageable maintenance and user adoption effort.

## 5.4 Legal & Regulatory Feasibility

- Ensure MT5 broker compliance for automated trading.  
- Adhere to KYC/AML regulations based on target markets.  
- GDPR/CCPA compliance for user data handling.  
- Provide disclaimers and risk disclosures to users.  
  
Conclusion: Legally feasible if regulatory compliance and broker agreements are ensured.

## 5.5 Schedule Feasibility

Prototype: 4–6 months  
Beta Release: 8–9 months  
Full Release: 9–11 months  
  
This timeline is realistic given available resources and agile development methodology.

# 6. Risk Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Likelihood | Impact | Mitigation |
| API Downtime (MT5/Data APIs) | Medium | High | Fallback sources, retry logic |
| Market Volatility | High | High | Strict stop-loss and risk management |
| Security Breaches | Medium | High | Encrypted communication, penetration testing |
| Regulatory Changes | Medium | High | Flexible architecture, periodic legal consultation |
| User Misuse (Over-leveraging) | High | Medium | Enforce leverage limits, display warnings |

# 7. Stakeholder Analysis

|  |  |
| --- | --- |
| Stakeholder | Role/Interest |
| End Users (Traders) | Use the bot to automate trading and increase profitability |
| Development Team | Build and maintain the platform |
| Management/Investors | Fund the project and seek ROI |
| Brokers/Exchanges | Provide trade execution services |
| Regulators | Ensure compliance with trading and data privacy laws |

# 8. Implementation Approach

- Methodology: Agile (Scrum) for iterative delivery and quick feedback loops.  
- Phases: Requirements → Design → Development → Testing → Deployment.  
- Resource Plan: Small cross-functional team with backend, frontend, DevOps, and QA capabilities.

# 9. Conclusion and Recommendation

The feasibility study demonstrates that the proposed MT5 crypto trading bot is technically, economically, operationally, and legally viable.  
  
Recommendation:  
- Proceed to the design and prototyping phase.  
- Focus on risk mitigation for security and regulatory compliance.  
- Ensure scalability and low-latency architecture.  
- Educate users on risk management.  
  
This project has strong potential to capture market demand for automated crypto trading solutions.