ECE496 Proposal

Project Title: Cryptocurrency Order Routing System

Project ID: 2025243

Supervisor: Michael Stumm

Administrator & Section: Meranda Salem, Section 8

Students: Viet Minh Nguyen, Somin Lee, Christopher Shih, Wen Ou

Date of Submission: October 18, 2025

Glossary

Blockchain: A shared, immutable digital ledger enabling the recording of transactions and tracking of assets within a network; it is typically distributed across nodes, uses cryptographic links between 'blocks', and provides a single source of truth that cannot be easily altered retroactively.

Contracts: In the blockchain context, self-executing programs ("smart contracts") that automate the actions required in a transaction such as transfers, conditional logic and event triggers.

Cryptocurrency: A digital or virtual form of currency that uses cryptography for security and operates on decentralized blockchain networks, independent of central banks or governments.

Decentralized: A system architecture in which control, decision-making, or regulation is distributed rather than residing in a single central authority; in blockchain or crypto contexts this means network participants validate and execute operations without relying on a central intermediary. This notion of decentralization is pervasive across disciplines and emphasizes multiple untrusted authorities rather than one trusted one.

Execution cost: The cost associated with completing a trade order in the market, defined as the difference between the ideal or benchmark execution price and the actual price obtained, including both market-impact and timing costs.

Exchange (or venue): A digital platform that facilitates the buying, selling, and trading of assets. Exchanges can be centralized (run by an intermediary) or decentralized (peer-to-peer). They provide services such as order matching, liquidity provision, and price discovery to enable efficient market transactions.

Fragmented liquidity: A condition in which available liquidity (trading volume and order-book depth) is spread across multiple venues, platforms, or chains - rather than being concentrated in a single pool. This can reduce the ability to transact large sizes or obtain best pricing at a single venue.

Latency: The delay between the initiation of an action and the moment it is completed. In trading systems, this commonly refers to the time difference between order submission and its execution or acknowledgement.

Limit order: An instruction to buy or sell a financial asset at a specific price or better. For a buy limit order, the trade will only execute at the limit price or lower; for a sell limit order, execution only occurs at the limit price or higher.

Liquidity: The degree to which an asset can be quickly bought or sold in the market at a stable price without significantly affecting its value. In the cryptocurrency market context, liquidity is also analysed via connectedness across assets and markets.

Liquidity pool: A collection of crypto funds locked in a smart contract to enable trading on decentralized exchanges, where users contribute assets and earn fees in return for providing liquidity.

Market order: An instruction to buy or sell a financial asset immediately at the best available current price in the market.

Open-source: Software released under a licence that allows users to view, use, modify, and distribute its source code freely. The open-source approach emphasises accessibility, transparency, and community engagement.

Order fill: An order fill refers to the execution of a trade order being completed - i.e., a buy or sell instruction is matched with another counterparty in the exchange, meaning the order is “filled”.

Routing (order routing): The process by which the destination of a trade order is selected and the order is forwarded to a trading venue (exchange, market-maker or dark pool) for execution; a key element in achieving best execution by minimising cost, delay or other adverse factors.

Sandbox environment: A testing environment that isolates untested code changes and experimentation from the production environment or repository, enabling development or security testing without risk of affecting live systems.

Throughput: The measure of how many transactions, operations, or data-units a system can process within a given period of time. In computing and networked systems, throughput is a key performance metric.

Trading: The buying and selling of financial instruments such as stocks, bonds, currencies, or derivatives on various markets to generate profits from price fluctuations.

Wallet: A digital tool or application that stores the cryptographic keys needed to access, send, and receive cryptocurrencies on the blockchain.

Executive Summary

Our project, the Cryptocurrency Order Routing System, proposes an open-source platform to optimize cryptocurrency trade execution across multiple exchanges. With over 560 million investors and roughly $600 billion traded daily, the crypto market faces major inefficiencies caused by fragmented liquidity, latency, and confusing execution algorithms. Existing platforms such as Robinhood and Wealthsimple have been criticized for hidden markups and suboptimal routing practices, creating a need for a more transparent and efficient solution.

This project seeks to develop a fully transparent and optimized routing system. This system should be able to simulate trade execution in real time and determine the optimal path across exchanges. It will operate through a web interface that allows users to submit buy or sell orders, which are then processed by a decision engine that dynamically routes the orders to the best exchange based on live market conditions, splitting up orders when necessary. Although the platform will not execute real trades, it will simulate order placements to evaluate latency, throughput, and fill performance.

We will design the system to minimize latency, maximize throughput, and guarantee complete simulated order fills for market orders. We will also employ REST and WebSocket APIs to gather live data from at least three exchanges in order to support both market and limit order simulations. Key risks include exchange API restrictions, algorithmic complexity, and performance challenges, which will be mitigated through parallel processing and heuristic optimization.

By offering a transparent order-routing system, this project directly addresses the lack of verifiability and fairness in the crypto trading scene. In the end, the result will be an equally transparent and efficient platform that improves execution quality, strengthens user confidence in a new, growing asset class, and can contribute to broader market stability. This demonstrates that open-source development promotes accountability and efficiency in financial technology, bridging the gap between innovation and ethical market design.

Table of Contents

1. Introduction

The cryptocurrency market has been expanding rapidly over the past few years, with over 69,000 new millionaires from this asset class in the past 12 months [1], more than 560 million eager investors [2] and up to $600B traded daily [3]. As the trading activity of this asset class expands, there is an increasing need for market transparency and efficiency. Many traders face challenges in trade execution, as directly connecting to exchanges leads to potential reductions in throughput and increases latency [4]. As a result, most trade activities have to go through platform providers, whose underlying order-routing algorithms and actual executed exchange price are hidden. Our project aims to address these challenges by designing an open-source system capable of real-time optimized order routing.

2. Background and Motivation

Cryptocurrencies (or cryptos) are digital assets built on blockchain protocols, enabling peer-to-peer transfers of value without traditional intermediaries [5]. Traders access crypto markets via centralized exchanges (CEXs), which use order books like other traditional asset classes, and decentralized exchanges (DEXs), which use smart contracts and liquidity pools so users can trade directly with each other from wallets [6].

However, interacting directly with multiple exchanges introduces significant friction to users. First, liquidity in each individual exchange is fragmented. Each exchange maintains its own order book, and traders sending an order to one venue risk encountering low volume activity, leading to higher probability of order incompletion. Empirical work shows that on Uniswap v3 (currently the biggest DEX), high-fee pools attract 58% of liquidity, but only execute 21% of the volume, illustrating that not all pools are equally active [4]. Persistent price discrepancies across exchanges have also been documented, especially during market stress [4], leading to traders potentially missing the optimal execution price for market orders. This effect is further exacerbated by trading latency, as trades on DEXs require on-chain confirmations, which can lag in times of congestion [7].

Moreover, despite the availability of order routing platforms like Robinhood or Wealthsimple, users have raised concerns about execution quality and hidden markups. Robinhood has been criticized for routing orders via “payment for order flow”, meaning it may prioritize venues that pay higher fees than those offering the best prices - an approach that has drawn regulatory scrutiny [8], [9]. Some users report that trades are executed at worse prices than prevailing market rates [10]. In Canada, Wealthsimple is flagged for charging crypto trading spreads of up to 2% [11] and providing suboptimal pricing that are hidden from customers [12].

The fragmentation of liquidity, heterogeneous execution costs and variable latency across crypto exchanges, and ambiguous activities of order routing providers highlight a critical need for an open-source, efficient order routing system. For traders, the ability to route orders effectively across exchanges is obviously important to maintain high execution quality and minimize price inefficiencies and losses. An efficient and fair routing system also attracts more market participants and promotes competitive pricing across exchanges, thereby reinforcing market stability and fostering confidence among both retail and institutional investors [13]. Most importantly, by exposing the routing algorithm to the public, we also offer transparency and verifiability, as users can inspect routing fairness and verify that no undisclosed price markups occur.

On the other hand, the feasibility of a crypto order routing system is already demonstrated by both industry practice and academic work. For instance, Robinhood's crypto product uses "smart exchange routing" to direct user orders across market makers or venues [14]. In the CEX domain, the Athena system merges order books from multiple centralized exchanges into a unified internal order book and splits orders to minimize implicit costs, showing that cross-venue routing in crypto is practical and beneficial [15].

Smart order routing (SOR) systems are also well studied. These systems algorithmically direct portions of a trade across venues to optimize a mix of cost, speed, and likelihood of execution [16]. An example in crypto settings is the 0x protocol, which decomposes crypto swaps and route orders across different DEXs to maximize liquidity [16]. These works provide both methodology and inspiration for our project, as we can adapt and innovate on these various routing algorithms.

Consequently, our project seeks to design and implement a free and efficient cryptocurrency order routing system, which receives trade orders from users, dynamically decides the optimal execution path among available exchanges and liquidity pools based on changing market conditions, and eventually sends out trades.

3. Problem Statement

Cryptocurrency traders face execution inefficiencies as liquidity is fragmented across exchanges with differing fees and latencies. Existing platforms have been accused of executing orders at suboptimal prices via hidden markups, raising transparency concerns. For most users, there is no open, publicly available auditable crypto order routing system that directs orders in real time across venues to secure optimal execution. As a result, many traders may incur avoidable costs, missed order fills, or suboptimal pricing.

4. Project Goal

The goal is to develop an open-source, efficient cryptocurrency order routing system that maximizes throughput, minimizes latency, and ensures transparency and verifiability in trade execution between users and exchanges.

5. System Requirements Specification

To meet the stated project goals, we have designed a set of specifications for the system in Table 1, which include requirements, constraints, and objectives.

Requirements are mandatory features the team must complete by the end of the project. Since the platform acts as an intermediary between traders and cryptocurrency exchanges, it must satisfy the basic functionalities of an order routing platform [17]. Any routing platform must be accessible by users, in our case in the form of a published website (R1). It must have the ability to display live market conditions across multiple destinations (exchanges) (R2), and send the buy or sell order from the origin (traders) to the corresponding calculated destination (R3). Both limit orders and market orders shall be supported, with guaranteed completion for market orders as this is the nature of the order type (R4). Numerical values for requirements are roughly estimated by our team based on our capabilities and current order routing solutions.

For the scope of this project, we restrict usage to simulated orders to avoid causing any unintended monetary impacts (C1). The routing algorithm should be accessible to users, based on public data (C2), and implemented using standard network procedures like REST APIs and WebSocket (C3) for transparency and verifiability.

The system objectives will measure the success of the project, which aligns with the remaining project goals, including maximizing throughput (O1) and minimizing latency (O2). To add further user usability, we also aim to make the system accessible programmatically (O3), accept public development contributions (O4), and support trade analytics (O5).

Table 1. System Requirements Specification

ID	Specification	Description
R1	Deployed website with intuitive user-interface	**Requirement**: The platform shall be designed and published as a website for easy access
R2	Market data collection and display from ≥3 exchanges, updated every second	**Requirement**: System shall collect, aggregate, calculate, and display live order books across multiple exchanges.

R3	Calculate and execute routing algorithm dynamically with <2 seconds end-to-end delay	**Requirement**: Starting from user order placement in the interface, the system shall determine execution routes and connect to calculated exchanges to simulate placement confirmation, all within 2 seconds.
R4	100% market order fill rate, <1 second after placement at exchange	**Requirement**: Market order shall be filled (simulated) within 1 second after placement at the target exchanges.
C1	Simulated trades	**Constraint**: The system must not execute real trades. Testing is limited to placing simulated orders on real exchanges (if supported) or simulated exchanges (sandbox environment).
C2	Routing algorithm and data sources available for public access	**Constraint:** All source code underlying the platform shall be made public and have no access restrictions for verifiability.
C3	Exchange connectivity via APIs	**Constraint**: Must utilize standardized exchange APIs (REST/WebSocket).
O1	Maximize throughput	**Objective**: System shall fulfill as many order requests as possible, including multiple users and concurrent orders.
O2	Minimize latency	**Objective**: System shall minimize the calculation and execution time in R2, as well as time to complete order fills in R3.
O3	API access to system	**Objective**: The underlying execution system shall allow programmatic access to the public through APIs.
O4	Open for extension and public contributions	**Objective**: The code structure shall be built for scalability, and made public to accept open-source contributions.
O5	Display all decision-making data and calculations with analytics	**Objective**: The information in C3 shall be logged and displayed for users in an organized fashion, including options supporting data analytics.

6. System Block Diagram

The system block diagram in Figure 1 illustrates the architecture and data flow of our order routing system. The platform enables users to submit orders through a web-hosted trading interface, which are then processed by our routing system that optimally splits and routes orders across multiple exchanges to achieve the best fill price and minimal latency. This diagram highlights the core system components, including the API gateway, routing algorithm, and Blockchain verification layer. It also lists performance metrics that will be measured at each step of the process, such as throughput, latency, and fill accuracy (see Glossary).

Modules:

1. **User/Frontend:** Displays trade performance and provides user interface to make an order.
2. **Order Routing System:** Contains routing logic
    a. **Decision Engine:** Optimizes routing using algorithms
    b. **Execution Module:** Places orders across multiple exchanges
3. **Metrics:** Used to calculate trade and system performance. These metrics are measured at different points of the user flow.
4. **Exchanges:** Crypto exchanges which we will attempt to connect to in order to send the order.
5. **Blockchain:** The ledger used to record and verify trading activity across the connected exchanges. Note that this is just shown for visualization purposes. We will not actually complete an order and record it in the blockchain, we just want to design the routing system.

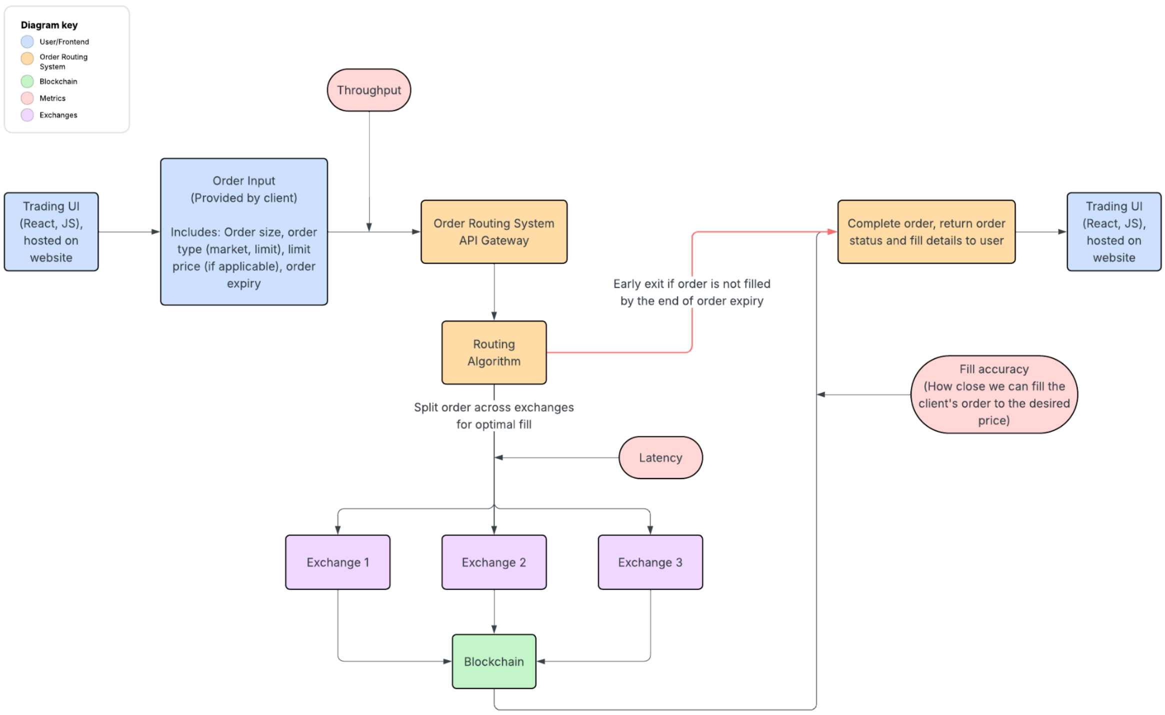


Figure 1. System Block Diagram for Cryptocurrency Order Routing System

The block diagram in Figure 2 describes how a limit order passes through our routing system. Supporting limit orders for our routing system is one of the main milestones we have decided on. This diagram can also be considered an introduction to how limit orders are executed in the marketplace. The user first inputs the order in the user interface (e.g. buy 0.2 BTC at the price of $105,000) as described in the previous diagram. This order is then recorded in the user's "order book", which keeps track of all orders. The order will wait in our system until we find someone selling at the right price on one of the connected exchanges, during which the order will be routed and either be partially or completely filled. If no matched orders at the desired price are found, the order will eventually expire.

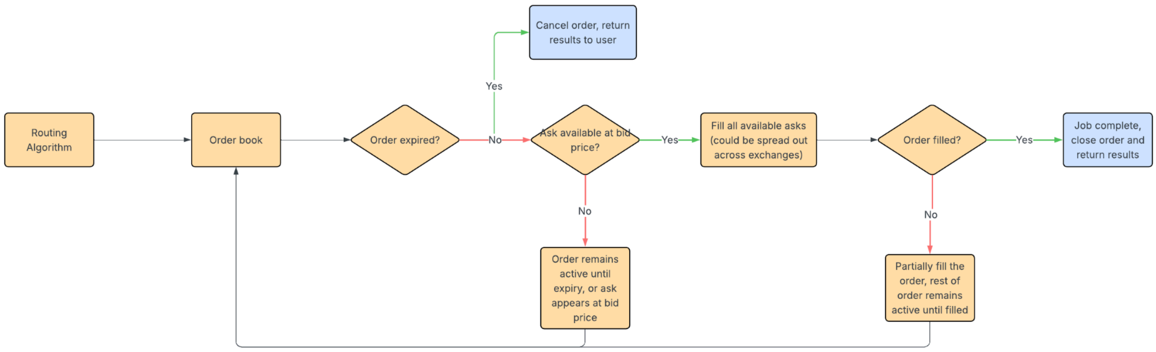


Figure 2. Execution and routing flow for limit order

7. Risk Assessment

Effective system development requires organized risk management to mitigate potential threats that could affect performance, security, or project delivery [18], [19]. In software development, risk assessment serves as a critical process to anticipate technical and operational uncertainties, ensuring system reliability and timely implementation [20]. Table 2 therefore summarizes main risks identified for the proposed system as well as their level of likelihood, impact, and planned mitigation strategies.

The risks address potential issues with exchange API access, system latency, regulatory constraints, and algorithmic complexity, with mitigation plans designed to maintain adaptability, efficiency, and compliance throughout development.

Table 2. System Risk Assessment

Risk	Likelihood	Impact	Mitigation
Exchange APIs restrict access or change	Medium	High	Use simulated exchange APIs, or modular API wrappers.
High processing and execution latency.	High	High	Implement efficient data structures, parallel processing, and use high-frequency programming languages (C++).
Latency bottleneck due to algorithm complexity	High	High	Begin with heuristic algorithms; add ML and complex extensions if time permits.
Legal, ethical, and budgeting restrictions on live trading	Medium	High	Limit the system to execute only simulated orders, or interact with simulated exchanges only.

8. Timeline

Our timeline moving forward is to first complete the project implementation plan before November 8. The following two weeks will be allocated towards our implementation of an interim demo. The first subtask is to set up a connection to a crypto exchange and display live data, which is a prerequisite for any subsequent trading operation. Next, we will develop support for CAD to BTC functionality as a demonstration of the project's trading capability. A clean UI and metrics/logging will also be implemented concurrently for full operational transparency. Afterward, time is allocated towards verification as well as preparing for the interim presentation. See Appendix A for a detailed Gantt chart timeline of our task breakdown as well as individual assignments. A task breakdown for the winter semester is also present, including support for multiple exchanges, limit orders, market orders, and optimizing for efficient routing to reduce latency while increasing throughput.

9. Conclusion

This project aims to design and implement an open-source, efficient cryptocurrency trading system that dynamically optimizes order routing across multiple exchanges, thereby reducing latency, maximizing throughput, while maintaining the transparency and verifiability for public users. This directly addresses market fragmentation issues in multi-exchange market environments and mitigates exposure to opaque or potentially unfair fees imposed by institutional order-routing solutions.

The project is feasible within our timeline through phased development (see Gantt chart in Appendix A), sandbox environment testing, and robust risk mitigation, including modular API design and algorithmic fallbacks. Regulatory risks may be overcome by limiting the prototype to simulated orders and exchanges, thereby ensuring compliance. This approach ensures deliverable outcomes while managing technical and regulatory challenges. The society will benefit from transparency in execution optimization previously unavailable to individual traders.

10. References

[1] A. Santoreneos, "Crypto boom creates 69,000 new millionaires in the past 12 months," *Forbes,* https://www.forbes.com.au/news/investing/crypto-millionaires-surge-40-in-12-months (accessed Oct. 1, 2025).

[2] D. Kemmerer, "How many people own crypto in the world? - june 2025 data," *CoinLedger*, https://coinledger.io/research/how-many-people-own-crypto-in-the-world (accessed Oct. 1, 2025).

[3] "Crypto market cap charts | coingecko," *Global Cryptocurrency Market Cap Charts*, https://www.coingecko.com/en/charts (accessed Oct. 1, 2025).

[4] A. Lehar, C. A. Parlour, and M. Zoican, "Fragmentation and optimal liquidity supply on decentralized exchanges," *SSRN Electronic Journal*, Nov. 2022. doi:10.2139/ssrn.4267429

[5] C. Murry, "Cryptocurrency explained with pros and cons for investment," *Investopedia*, https://www.investopedia.com/terms/c/cryptocurrency.asp (accessed Oct. 1, 2025).

[6] N. Reiff, "What are centralized cryptocurrency exchanges?," *Investopedia*, https://www.investopedia.com/tech/what-are-centralized-cryptocurrency-exchanges (accessed Oct. 1, 2025).

[7] U. W. Chohan, "The limits to blockchain? scaling vs. decentralization," *SSRN Electronic Journal*, Mar. 2019. doi:10.2139/ssrn.3338560

[8] "Florida AG probes robinhood crypto over claims of low-cost trading," *Reuters*, https://www.reuters.com/legal/government/florida-ag-probes-robinhood-crypto-over-claims-low-cost-trading-2025-07-10/ (accessed Oct. 1, 2025).

[9] "Robinhood investing review: 4 reasons to avoid this broker," *Wantfi*, https://wantfi.com/robinhood-stock-trading-app-review.html (accessed Oct. 1, 2025).

[10] "R/bitcoin on reddit: Warning: BTC on Robinhood," *Reddit*, https://www.reddit.com/r/Bitcoin/comments/1kqdm1p/warning_btc_on_robinhood/ (accessed Oct. 1, 2025).

[11] "Understand why your cryptocurrency order filled at a different price," *Wealthsimple*, https://help.wealthsimple.com/hc/en-ca/articles/360060805574-Understand-why-your-cryptocurrency-order-filled-at-a-different-price (accessed Oct. 1, 2025).

[12] "Wealthsimple Crypto Review: Pros, cons and how it compares (October 2025)," *RankFi*, https://rankfi.com/ca/wealthsimple-crypto-review/ (accessed Oct. 1, 2025).

[13] "Why is liquidity important for exchanges?," *ShiftMarkets*, https://www.shiftmarkets.com/blog/why-is-liquidity-important-for-exchanges (accessed Oct. 1, 2025).

[14] "Crypto Order Routing," *Robinhood*, https://robinhood.com/us/en/support/articles/crypto-order-routing/ (accessed Oct. 1, 2025).

[15] R. Henker *et al.*, “Athena: Smart order routing on centralized crypto exchanges using a Unified Order Book,” *International Journal of Network Management*, vol. 34, no. 4, Mar. 2024. doi:10.1002/nem.2266

[16] “0X Smart Order Routing,” *0x*, https://0x.org/post/0x-smart-order-routing (accessed Oct. 1, 2025).

[17] “Order Routing System (ORS),” *QuestDB Glossary*, https://questdb.com/glossary/order-routing-system-%28ors%29/ . (accessed: Oct. 18, 2025)

[18] International Organization for Standardization, ISO 31000:2018 Risk Management – Guidelines, Geneva, Switzerland, 2018.

[19] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 7th ed., Newtown Square, PA, USA: PMI, 2021.

[20] IEEE, IEEE Standard 1540-2001: Standard for Software Life Cycle Processes—Risk Management, Piscataway, NJ, USA: IEEE, 2001.

11. Appendices

Appendix A. Gantt Chart

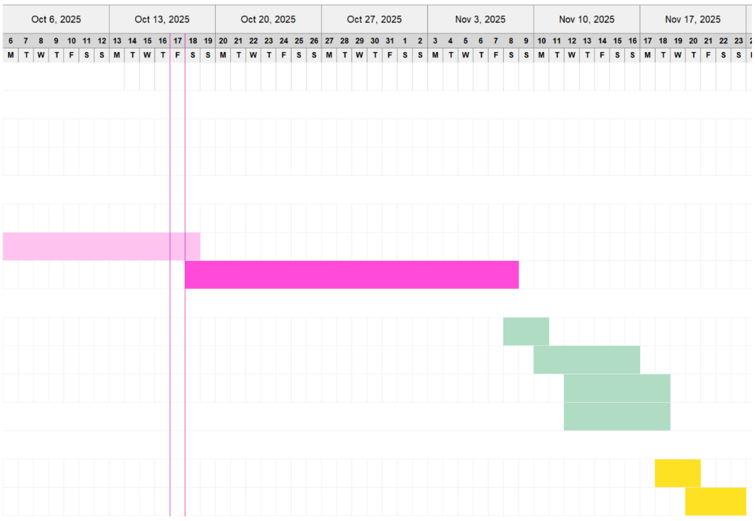
Order Routing System

ECE496

Project start: **Tue, 9/2/2025**

Display week: **6**

TASK	MEMBER	PROGRESS	START	END
Initiation				
Project Snapshot		100%	9/2/25	9/9/25
Preliminary Proposal		100%	9/9/25	9/18/25
Proposal and Implementation Plan				
Detailed Proposal		100%	9/18/25	10/1/25
Updated Proposal		100%	10/1/25	10/18/25
Implementation Plan	All	0%	10/18/25	11/8/25
Partial Preliminary Implementation				
Exchange and live data	Michael	0%	11/8/25	11/10/25
CAD <-> BTC only	Chris, Wen	0%	11/10/25	11/16/25
Clean functionality and UI	Michael	0%	11/12/25	11/18/25
Metrics and Logging	Mindy	0%	11/12/25	11/18/25
Validate Implementation & Interim Demo				
Validate implementation	All	0%	11/18/25	11/20/25
Prepare for demo	All	0%	11/20/25	11/23/25



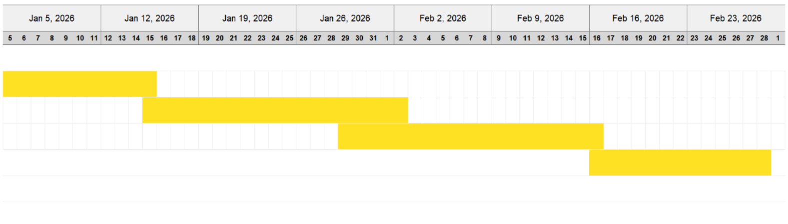
Order Routing System

ECE496

Project start: **Tue, 9/2/2025**

Display week: **19**

TASK	MEMBER	PROGRESS	START	END
Winter				
Multiple exchanges	Wen	0%	1/5/26	1/15/26
Limit orders	Michael, Wen	0%	1/15/26	2/2/26
Market orders	Chris, Mindy	0%	1/29/26	2/16/26
Routing optimization	Michael, Wen	0%	2/16/26	3/18/26
Performance Testing	Chris, Mindy	0%	3/18/26	3/23/26
Presentation	All	0%	3/21/26	3/31/26



Order Routing System

ECE496

Project start: **Tue, 9/2/2025**

Display week: **25**

TASK	MEMBER	PROGRESS	START	END
Winter				
Multiple exchanges	Wen	0%	1/5/26	1/15/26
Limit orders	Michael, Wen	0%	1/15/26	2/2/26
Market orders	Chris, Mindy	0%	1/29/26	2/16/26
Routing optimization	Michael, Wen	0%	2/16/26	3/18/26
Performance Testing	Chris, Mindy	0%	3/18/26	3/23/26
Presentation	All	0%	3/21/26	3/31/26