BetCoin

# Final Project Report

*Smart Contract Development, Spring 2023*

Team Members

Apurva Harsulkar
Kevin Hou
Colby Meline
Joel Nail
Yang Wang

# Table of Contents

# Project Abstract

BetCoin is a blockchain-based platform that offers users the ability to create polls, vote on polls, and bet on the outcome of polls. The goal of BetCoin is to enable companies, individuals, and anyone in between to obtain the wisdom of the crowd at no cost. By allowing users to bet on the outcome of polls, BetCoin ensures a recurring stream of funds which is "taxed" and distributed to poll voters as an incentive for participation. Through this model, BetCoin prevents poll creators from having to spend money incentivizing users to vote on their polls. BetCoin is a Web 3.0 platform that utilizes blockchain technology to offer a betting experience that is fully integrated within the site through the use of cryptocurrency. At this point in time, BetCoin is merely a proof-of-concept, but with additional development, the platform has the potential to become a full-fledged offering that is available to all members of the public. In the following report, we explain the motivation behind the project, describe the technical details of the current BetCoin system, and outline the work that will need to be performed to produce a fully functional platform.

# Project Report

## Problem Statement

For years, companies have been seeking to obtain information straight from customers to help guide product development decisions. Design thinking, a lauded framework for developing solutions, emphasizes the importance of empathizing with end-users and understanding their needs and desires before jumping into development. Modern companies have embraced the importance of placing the user first when developing products which means the ability to communicate with these users is more vital than ever.

There are numerous methods that companies can employ to probe the minds of their users. Traditional surveying techniques have been used for years to obtain insights from customers, and focus groups are a popular technique for gathering rich information from a target group. As Web 2.0 emerged and came to dominate the 21st century, new and improved digital methods were created and aimed at harvesting data from more customers than ever before. Online channels such as social media and web-based surveys allow companies to reach an incredibly wide audience and do so at a fraction of the cost when compared with traditional methods; however, there is still a cost associated with these efforts. Some companies employ a sweepstakes approach where users are entered into a raffle for offering their feedback. Others might offer coupons or other promotions. The goal of these offerings is to provide users with an *incentive* to participate and provide their thoughts.

The fact that companies are willing to incur these costs for the sake of obtaining information from their user base demonstrates that customer feedback possesses immense value, and the challenge for organizations is balancing the cost of gathering feedback against the value it offers. This is a challenging problem for companies to navigate; it often seems as though a modern organization cannot succeed without consistently incorporating user feedback into their products, yet the costs associated with providing incentives required to obtain feedback can quickly grow out of hand.

## Project Novelty

Our project, BetCoin, intends to address the problem described above by providing companies of all sizes a platform where they can poll their customers on important questions and receive valuable insights without the company having to directly provide these users with an incentive for doing so. The BetCoin platform ensures that users are incentivized for their opinions at no cost to companies by offering three essential functionalities: creating polls, voting on polls, and betting on the outcome of polls. The betting aspect of the platform introduces a recurring stream of revenues that are essentially "taxed" in order to provide voters with an incentive for participating in the platform. As a result, all sides of the platform receive some sort of incentive for participation: poll creators receive insights from a diverse user base, voters receive a monetary reward for voting on a poll, and bettors have the potential to win money by placing successful bets.

Currently, there are a number of websites that companies can use to poll their customers and obtain answers to important questions. One example is Twitter Polls, a feature of the popular social media platform Twitter that allows users to ask questions and receive responses from other users on the platform. While Twitter Polls are easy to create and propagate, they do not offer any sort of incentive to the users who vote on the poll. It is therefore an unreliable method for obtaining consistent feedback from a company's target markets. In conducting research for this project, we were unable to find any website that allows companies to pose questions to their desired customers while also providing those customers with an incentive at no cost to the company. We believe that this aspect of our project is completely unique and currently unavailable on the market.

There are also a number of gambling sites that allow users to place bets on a number of events e.g., sports, card games, elections, etc. That said, we were unable to uncover any platform that enables users to bet on the outcome of a poll that has been created by other users. There are also very few sites that offer betting supported by Blockchain technology which ensures that the privacy and security of bettors' funds are never in question. Overall, we believe that BetCoin is a novel and valuable solution that will enable all three sides of our platform to participate in a fun and rewarding ecosystem aimed at promoting the dissemination of knowledge for anyone who desires to join.

## System Architecture

There are three essential layers in the BetCoin system architecture: the frontend, the smart contract, and the blockchain. The frontend is supported by React JS and TypeScript with additional Web 3.0 functionality offered by Web3JS and Metamask. These Web 3.0 functionalities allow our frontend to connect with our smart contract which is written in Solidity. Our smart contract is deployed on Ganache where we have tested our functionality using a local blockchain. A diagram demonstrating the architecture of our system is provided in the appendix of this report. Additional details about the smart contract and frontend of our system are discussed in the following two sections.

## Smart Contract Design

All functionality for our platform is supported by a single smart contract which includes three primary functions: *createPoll* for poll creation, *voteOnPoll* for poll voting, and *betOnPoll* for poll betting. Although there are several other functions within the smart contract, they are primarily used to get information from the blockchain or are used to support one of the three primary functions. Each of the primary functions is associated with a data structure (*Poll*, *Voter*, and *Bet*) - a new instance of these structures is created each time a user triggers the corresponding primary functions.

The *Poll* structure is the most complex of the three structures. The Poll structure tracks the poll creator, the start and end time for the poll, the poll prompt and voting choices, the number of votes cast for each voting choice, a boolean variable denoting whether the poll is open or closed, and a variable to indicate which voting choice received the highest number of votes.

The *Voter* and *Bet* structures are much simpler than the Poll structure. The *Voter* structure simply includes the poll on which the vote was cast, the address of the voting user, and the voting choice that the voter selected. The *Bet* structure, similar to the *Voter* structure, tracks the poll and the voting choice on which the bet was placed. Unlike the *Voter* structure, however, the *Bet* structure includes a payable address and an amount which is the amount of money wagered by the bettor.

Each primary function is also associated with one or more events that are emitted whenever the function is triggered or whenever the created data structure undergoes a certain event. For example *PollCreated* is emitted upon the creation of a poll, and *PollClosed* is emitted once the end time for the poll has been reached. *VoteCasted* is emitted whenever a user votes on a poll and *BetPlaced* is emitted whenever a user places a bet on a poll. There is also an event named *RewardsDistributed* which lists all the winning bettors and their associated winnings whenever a poll closes. Although all of these events have been outlined, the contents of the events still need to be assessed, and code to actually emit the events must be implemented.

Although our smart contract is functional, the Solidity code that supports our smart contract is extremely rough due to time constraints and limited experience working in Solidity. For example, the code used to calculate the amount of winnings to be distributed to each winning bettor relies on matching array indexes rather than a more structured method of determining successful bets and their corresponding users. Before this project could be fully deployed, a full reassessment of the smart contract code would need to be performed to ensure that the smart contract is free from bugs and security concerns.

## Client & UX/UI

The frontend of our website was designed using ReactJS and TypeScript; however, we were unable to implement this code due to troubles during deployment and integration of our smart contract with the frontend. These issues stemmed from limited experience working with Web3JS and lacking a full understanding of how to implement the Metamask extension within our code. Due to these struggles, we were unable to connect our smart contract to a functional frontend which is why our final presentation only included a demo of our smart contract in Remix. The frontend code that was created but not implemented can be found in our GitHub repository which can be found in the Source Code section.

Although our lack of a functional frontend was a major shortcoming in our project development, we did produce mockups to demonstrate the UX/UI we had planned for our final implementation. We did not intend to implement every aspect of these mockups within our in-class demo, but we believe they showcase the idea that we had in mind for our final product. Mockups for both a web and mobile version of the BetCoin frontend can be found in the appendix.

The mockups include several details that we did not have time to refine; however, they include the three primary functionalities which were described in the Smart Contract Design section above. For our in-class demo, we had hoped to demonstrate the process of creating a poll, voting on that poll, placing a bet on that poll, and then distributing rewards to successful poll bettors. We were able to demonstrate this functionality by deploying our smart contract via the Remix virtual machine, but we did not get to showcase how a user would interact with the frontend of our site. Additional work must be performed to enable this functionality which will be discussed in more detail within the Future Work section below.

## Security & Deployment

Our existing smart contract includes numerous security concerns that would need to be addressed before a full-scale deployment of our project could be considered. The primary security concern for our site is the fact that our smart contract is not currently designed to encrypt the contents of transactions whenever a user votes on our poll. This functionality is essential to the existence of our platform since any knowledge of the number of votes for each poll voting choice would taint the integrity of any bets made on that poll. We described the necessity of vote encryption in our prior deliverables, but this functionality was deemed out of scope for the proof-of-concept system we developed this semester. Implementing vote encryption would be one of the first security concerns for our team to address before fully deploying our project.

Additional security concerns include several functions which have been made public for the purposes of testing but need to be made private within the final implementation of our system. If users were able to access these functions publicly, they would be able to learn the number of votes allocated to each voting choice for a poll which, like the issue described above, would compromise the integrity of polls on our platform. Additionally, we removed code to limit the number of votes and bets users can place on a poll to perform our in-class demonstration. This is another major security concern since users could manipulate polls by placing numerous votes and bets with the same account.

In addition to problems described above, we believe that there are numerous other potential security concerns that we are not currently aware of. We are also unable to ensure that our frontend is secure since we were unable to produce a functional frontend system during our initial development. We would therefore need to conduct extensive security testing after more development has been completed on our smart contract and frontend. Once we have verified the security of our system architecture, we can begin exploring full-scale deployment options.

## Future Work

The smart contract that was shown in our team’s in-class presentation is merely a proof-of-concept, and is nowhere near complete. There are numerous aspects of our project that require additional attention in order for our platform to be ready for deployment. In addition to addressing the security concerns described in the section above, we have identified several areas of future development for our team to pursue. The primary focus for future work on this project would be finalizing our frontend and ensuring that all aspects of our system architecture can successfully communicate with one another. Despite our inability to create a functional frontend prior to our in-class demonstration, we did develop the beginnings of a frontend system, and we believe that a prototype frontend for our system could be completed quite quickly.

Another aspect of our project that is essential to our platform’s success is the addition of protections against users who might try to manipulate our system. We discussed this concern extensively during our initial project pitch and developed several potential mitigations for this issue; however, we deemed these protections out of scope for our project’s initial phase of development. Implementing the ideas presented in our first project deliverable is a vital step in any future work done on our system.

In order for our system to provide value to users, poll voters must be rewarded for participating in the platform. Currently, we have not implemented the necessary code to provide poll voters with funds after they have voted on a poll which greatly limits the attractiveness of our platform. Implementing this aspect of our system will take a considerable amount of effort, but it is necessary in order for our platform to solve the problem statement described at the beginning of this report.

Finally, there are several additional functionalities that we considered and documented throughout the development of our system. For the purposes of creating a proof-of-concept, we decided not to pursue any of these advanced features, but we believe that could add a significant amount of value to our platform in its final form. Examples of advanced features include poll categories and the ability to filter based on these categories, different betting options other than betting on which poll voting choice will receive the highest number of votes, and providing rewards to poll creators to encourage participation in the platform. Although these ideas are not required for our system to be functional, they would provide additional value to users and make our website more desirable to all sides of the platform.

During this semester, our team barely scratched the surface of what BetCoin could become, but we are extremely proud of the work that we have completed. Going forward, we believe that creating a full-scale implementation of the BetCoin platform is attainable and could be an extremely popular website for a wide variety of users. We are so excited to have had the chance to work on this project and look forward to the future that BetCoin has in store.

# Source Code

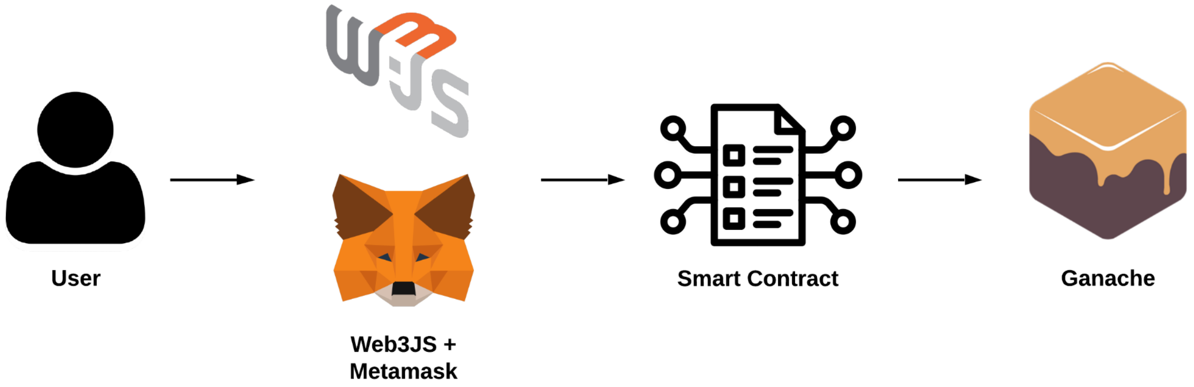
GitHub repository containing BetCoin smart contract and frontend source code

# Project Presentation

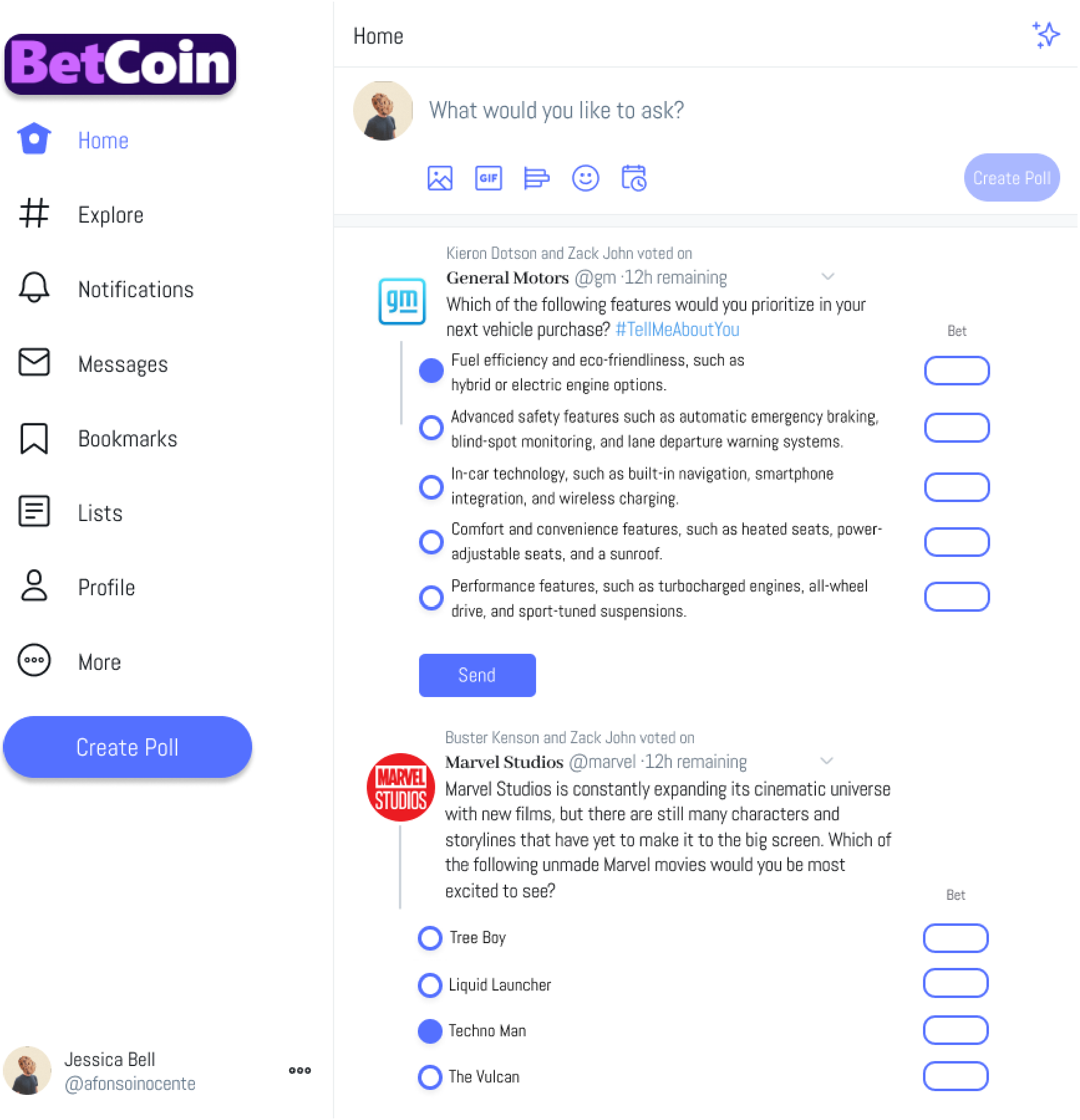
Link to project presentation on Canva

# Appendix

## System Architecture Diagram



# Frontend Mockup - Web



Kieron Dotson and Zack John voted on

**General Motors** @gm ·12h remaining

Which of the following features would you prioritize in your next vehicle purchase? #TellMeAboutYou

● Fuel efficiency and eco-friendliness, such as hybrid or electric engine options.

○ Advanced safety features such as automatic emergency braking, blind-spot monitoring, and lane departure warning systems.

○ In-car technology, such as built-in navigation, smartphone integration, and wireless charging.

○ Comfort and convenience features, such as heated seats, power-adjustable seats, and a sunroof.

○ Performance features, such as turbocharged engines, all-wheel drive, and sport-tuned suspensions.

Buster Kenson and Zack John voted on

**Marvel Studios** @marvel ·12h remaining

Marvel Studios is constantly expanding its cinematic universe with new films, but there are still many characters and storylines that have yet to make it to the big screen. Which of the following unmade Marvel movies would you be most excited to see?

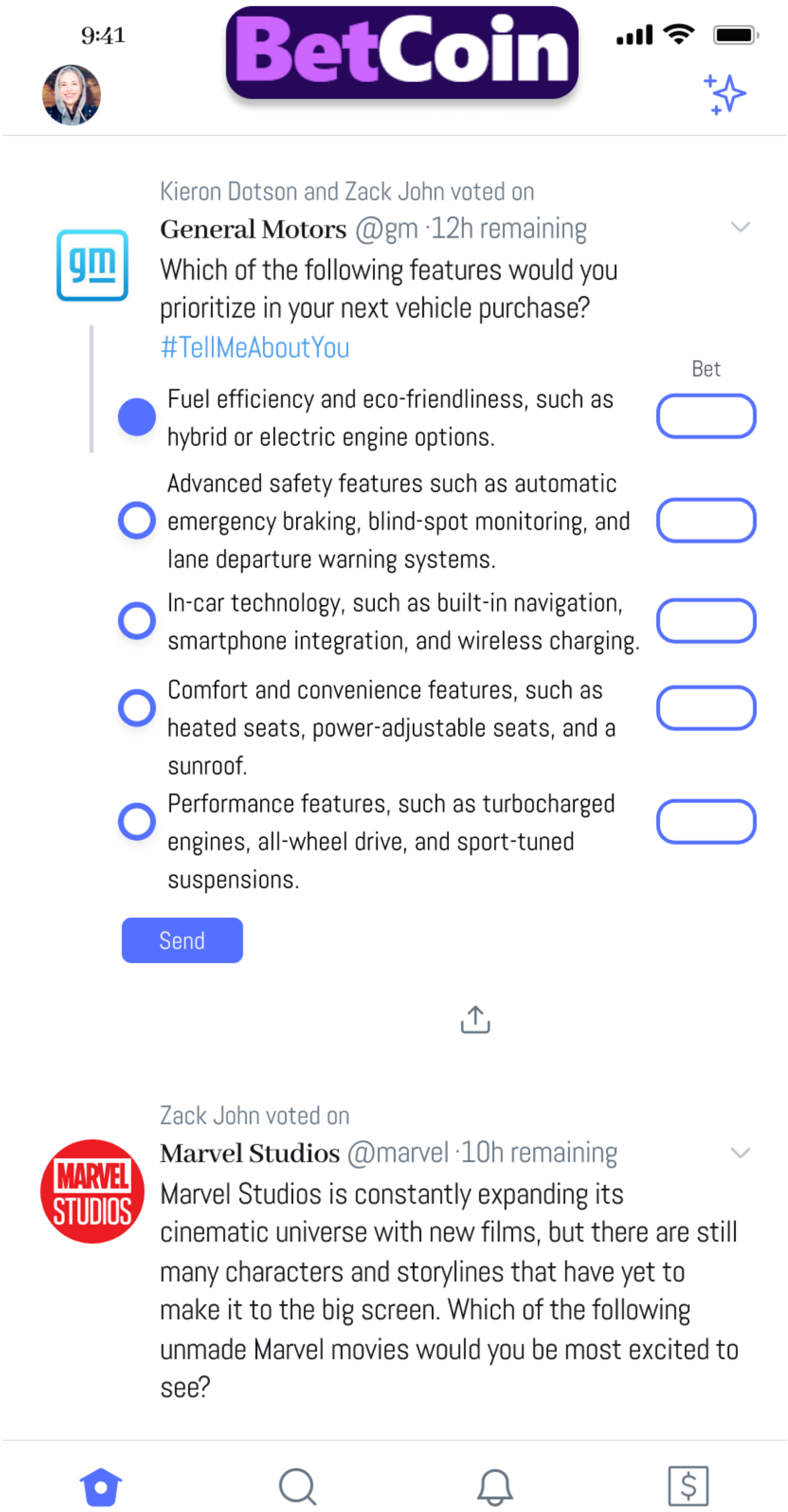
○ Tree Boy

○ Liquid Launcher

● Techno Man

○ The Vulcan

# Frontend Mockup - Mobile



9:41

BetCoin

Kieron Dotson and Zack John voted on

**General Motors** @gm ·12h remaining

Which of the following features would you prioritize in your next vehicle purchase? #TellMeAboutYou

Bet

● Fuel efficiency and eco-friendliness, such as hybrid or electric engine options.

○ Advanced safety features such as automatic emergency braking, blind-spot monitoring, and lane departure warning systems.

○ In-car technology, such as built-in navigation, smartphone integration, and wireless charging.

○ Comfort and convenience features, such as heated seats, power-adjustable seats, and a sunroof.

○ Performance features, such as turbocharged engines, all-wheel drive, and sport-tuned suspensions.

Send

Zack John voted on

**Marvel Studios** @marvel ·10h remaining

Marvel Studios is constantly expanding its cinematic universe with new films, but there are still many characters and storylines that have yet to make it to the big screen. Which of the following unmade Marvel movies would you be most excited to see?