Columbia Fintech Project 3 Notes

Integration

- Responsible: Pravin

Blockchain

- Local (test) blockchain w/ Ganache + MetaMask
- 10 accounts
  - 1 account for owner of CBET contract owner/deployer
  - 1 account for CBET wallet (which holds all user/bettor deposited funds to use for betting)
  - Up to 8 user/bettor accounts

Odds Scraping

- Responsible: Will
- Platform: <placeholder>
- Will this be a dynamic scraping activity? Or a background task and save off the results in some format (json? csv?) that the application can read?
  - Will be live/dynamic.
- How will this be integrated into the main front-end application? (integrated w/ streamlit)

Back End / Contracts

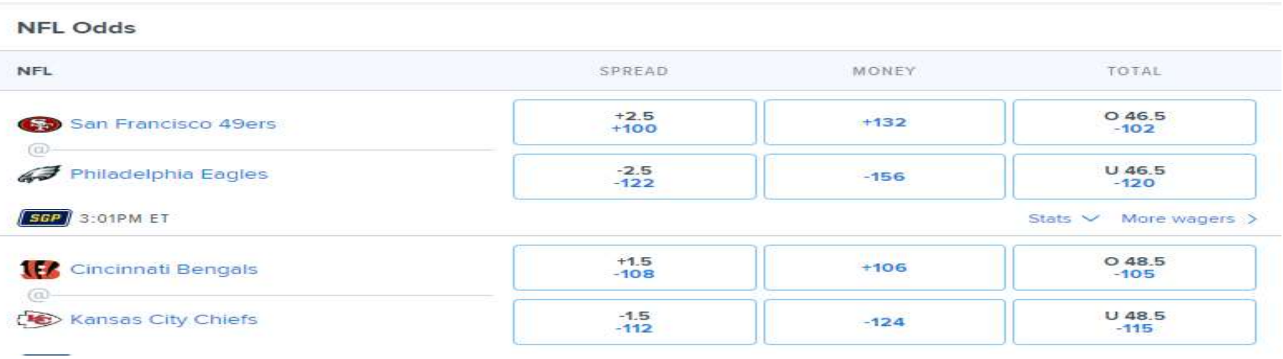
- Responsible: Stratis/Liset
- Platform: Solidity
- Should have a contract interface for above (5) items…
  - (1) Sport Accounts
  - (2) Team Accounts
  - (3) CBET Accounts
  - (4) Betting Games
  - (5) Betting Actions
- Details of these contracts added to the Front End section below

Front End

- Responsible: Esteban/Pravin/Will

- Platform: Streamlit? Other? Flask? React?
    - Streamlit and/or Flask.
- (1) Sports interface (owner applicable only) - creating Sport Accounts - e.g. Football, Hockey, Basketball, etc.
    - Create sports available to bet on:
        - *function createSport(string memory _sportName) {}*
    - Get the contract assigned sport id (to be used below when setting up game to bet on - this is the reference id):
        - *function getSportId(string memory _sportName) {}*
    - There is a reverse lookup as well if that would be useful:
        - *function getSportName(uint _sportId) {}*
- (2) Team interface (owner applicable only) - creating Team Accounts - e.g. NY Giants, NY Jets, NY Rangers, etc.
    - Create teams available to bet on:
        - *function createTeam(string memory _teamName) {}*
    - Get the contract assigned team id (to be used below when setting up game to bet on - this is the reference id):
        - *function getTeamName(uint _teamId) {}*
    - There is a reverse lookup as well if that would be useful:
        - *function getTeamId(string memory _teamName)*
- (3) User/Bettor Accounts: Creating User/Bettor Accounts
    - Will we require users to login (username, password)? Or just be able to open up the application?
        - If have more time, add username/password (nice to have)
    - Create the user/bettor account:
        - *function createBettorAccount(address payable _addr,*
          string memory _firstName, string memory _lastName,
          string memory _username, string memory _password) {}
    - Helper (getter) functions:
        - *function getBettorAccountName(address _addr) {}*
        - *function getBettorAccountUsername(address _addr) {}*
        - *function getBettorAccountPassword(address _addr) {}*
        - *function isBettorAccountActive(address _addr) {}*
    - Need the ability to activate or deactivate user accounts:
        - Note, by default, when the user account is created using "*createBettorAccount*", the account will be active.
        - *function setBetterAccountInactive(address payable _addr) {}*
        - *function setBetterAccountActive(address payable _addr) {}*
    - *Deposit funds into the Cbet blockchain application:*
        - Assign the Cbet application account/wallet address so users can deposit funds into:
            - *function setBACbetAccountWalletAddr(address payable _cbetAccountWalletAddr) {}*

        - Note, I may change this and just pass it in as a parameter to the deposit function
      - Note, the current application can only deposit and withdraw ether. Open/TBD to add custom CBET tokens.
      - *function depositBettorAccountEther() {}*
      - *function getBalanceBettorAccountEther() {}*
    - Get Balance:
      - *function getBalanceBettorAccountEther()*
- (4) User Betting Selection: Creating interface for users to select games to bet on
    - This is where the odds scraping results should be presented
    - Setting the games and the odds that can be bet on:
      - *function createGame(uint _homeTeamId, uint _awayTeamId,*
        int _homeTeamOddsMoneyline, int _awayTeamOddsMoneyline,
        int _homeTeamOddsSpread, int _awayTeamOddsSpread, bool _isHomeFavorite, uint _spread,
        int _homeTeamOddsOverUnder, int _awayTeamOddsOverUnder, uint _overUnder) {}
        - For reference: the following is what I used to setup the streamlist test app:



NFL Odds

NFL	SPREAD	MONEY	TOTAL
San Francisco 49ers	+2.5 +100	+132	O 46.5 -102
@ Philadelphia Eagles	-2.5 -122	-156	U 46.5 -120
SGP 3:01PM ET			Stats ∨ More wagers >
Cincinnati Bengals	+1.5 -108	+106	O 48.5 -105
@ Kansas City Chiefs	-1.5 -112	-124	U 48.5 -115

      - Helper (getter) functions:
        - function getGameTeamIds(uint _gameId)
        - function getGameMoneylineOdds(uint _gameId)
        - function getGameSpreadOdds(uint _gameId)
        - function getGameOverUnderOdds(uint _gameId)
      - Updating of the game odds, after the initial “createGame”:
        - function updateGameOddsMoneyline(uint _gameId, int _homeTeamOddsMoneyline, int _awayTeamOddsMoneyline)
        - function updateGameOddsSpread(uint _gameId, int _homeTeamOddsSpread, int _awayTeamOddsSpread, bool _isHomeFavorite, uint _spread)

        - function updateGameOddsOverUnder(uint _gameId, int _homeTeamOddsOverUnder, int _awayTeamOddsOverUnder, uint _overUnder)
    - Controlling the game time events:
        - (1) Pre game start - betting allowed,
        - (2) Game in progress - betting not allowed
        - (3) Game end - winnings/losses distributed
        - enum GameStatus { PRE_GAME_START, GAME_IN_PROGRESS, POST_GAME_END }
        - When the “CreateGame” function is called above, the default setting will be the “PRE_GAME_START” state
        - function setGameStart(uint _gameId)
        - function setGameOver(uint _gameId)
        - function isAcceptingBets(uint _gameId)
        - function isGameInProgress(uint _gameId)
        - function isGameOver(uint _gameId)

- (5) Betting Action:
    - Controlling the games that can be bet on.
    - Betting results:
        - If the user wins, distribute funds to user/bettor accounts
        - If the user loses, move the funds to the owner account? Or keep in betting application account?
    - (No functions developed yet)
- Presentation ??