



<https://github.com/TheJollyLaMa/BabelBet>

Justin LaManna, aka The Jolly LaMa



Babel Bet

An Exploration

of

Duolingo

and

Web3

**Babel Bet**

Babel Bet is a Decentralized Application (Dapp) leveraging Web3 and a smart contract in solidity to allow students of the Duolingo app to challenge each other to friendly wagers that make language training exciting by raising the stakes and involving friends in your journey!

Frontend

Shameless AngelToken Plug

Decorated About Page with links to the vitals

Shows Connected Account

InitiateChallenge

Frontend form to start a challenge with a fellow Duolingo student

Collects:

Email & Password, Email to Challenge, Eth Account,

Duolingo Username & Password, Challenge Proposal

Sends to backend route to fill email parameters and send offer via email

CounterOffer

Frontend form to offer different terms if the initial terms do not satisfy

Sends to backend to fill email parameters and send counteroffer via email

Accept

Frontend form to accept terms and stake funds

Sends to backend to initiate escrow and send confirmation emails

ChallengeView

Frontend view of the challenge’s vital statistics and status

Connects to Eth account, enter token ID to view

Terms, Stakes, Escrow Earnings, and Challenge History

Maybe use the Duo Avatars if you can get ahold of them

(in terms negotiations emails too)

ChallengeBank

Historical view of challenges

Searchable by Id

Backend

InitiateChallenge

Sends email to your chosen language buddy with terms to a challenge

CounterOffer

Sends email back to the initiator to suggest different terms

AcceptChallenge

GetDuoInfo to get base stats going into the challenge

Call to ChallengeToken tokenGenesis to mint starting stats and challenge details

Call to update\_Token\_Player\_Email\_Map() to map emails 🡪 ethAddr 🡪 token id

Call to update\_Token\_Player\_Duo\_Map() to map DuoInfo 🡪 ethAddr 🡪 token id

transfer funds (Matic) from both parties and place in Escrow with Web3

Send email to both parties to confirm agreement with escrow details

For multiplayer: countdown begins for everyone else to accept within 24hrs

WatchChallenge

Set up twice daily Cron Schedule

Fetch challenges

If challenge active

Fetch\_Duo\_Info For Each Player

Check Students progress against challenge terms

If (student.Streak < days\_into\_challenge){

Email notifying of forfeiture

}else{

If (end\_date == 0 ) {

Today is the end date

Add player to payoutList

}else{

Wager is still active, Player is meeting challenge

do nothing, check next player

}

ExecutePayout

Iterate through payoutList

Payout goes to winner’s Eth account via web3 call to release escrow

Emails are sent confirming challenge completion and reporting results

Challenge is forever stored on chain

Payouts can be dynamic or all or nothing (80% completion = 20% loss in stake)

SearchById

Look up past challenges by ID to fetch the historical record from the chain

Solidity

Challenge Library

Each deposit entry has an ID, depositing addresses, and depositing terms

Funds can be held frozen or released to earn yield by ‘status’

ChallengeToken

Generate Challenge Tokens from challenge library

Makes a unique id, Mints an 1155, transfers them to the players

getChallenges

public call to view all challenges

getChallengeTokenLength()

gets the total number of challenge tokens ever minted on this contract

update\_Token\_Player\_Email\_Map()

maps a token id to one of its players by eth address to show their email

update\_Token\_Player\_Duo\_Map()

maps a token id to one of its players by eth address to show their Duo Info

ChallengeTokenExecution

changeStatus: changes status of token to signify ‘paid’ status

Escrow

Deposit to Escrow – two or more accounts deposit funds in contract account

Escrow account sweeps funds over to whatever dapp has the best schema

Different schemas could be selected according to risk appetite.

Maybe only sweep forfeited funds to high yield account as an option

Tests

Frontend backend and solidity

Notes:

Plenty of other challenges to dream up –maybe even a custom challenge option

Portions of winnings go to charity

Losers feed Losers – if you lose, all your money goes to your chosen charity.

Tip the scales – payout toppers given by the babelbet contract to incentivize certain activites

Link to codeacademy or some other code language platform to do the same with computer languages

Also link to other skillz type apps – like workout apps or apple watch or diet apps

Gonna need a library of apis that link – like the quicken bank connect app

Oooorrrr …

Link to social networks for friends to challenge:

<https://docs.cyberconnect.me/>

https://cyberconnect.me/

Cyberconnect – found on gitcoin grants rnd12

<https://www.theblockcrypto.com/post/124510/web3-social-infrastructure->startup-cyberconnect-raises-10-million-in-seed-funding

CyberChat – messaging service from cyberconnect!

Possible replay on Angel Tokens – each Challenge being an 1155. The simplest form being between two players but could include groups and handle group payouts. Switch out the angeltoken product for a set of terms to the challenge, the coefficient controls the payout amounts, and cost is the amount the players wish to stake.

So there is a challenge (contract1 (alms)), a ChallengeToken (contract2 AngelToken(genesis and buy)) and an execution contract (contract3(AT\_X)). Execution contract is the on chain watcher.

Polygon Ecosystem Round Gitcoin Grants

04DEC2021

The polygon team wants to help support with development and marketing!

Check in at discord server – The team seems very supportive!