

HumanDefiHæck

ækiti æmbassy

Overview

The ækiti æmbassy team for the HumanDefiHæck is made up of six(6) techies, with HumanDefiHæck we want to go one step further and reinvent financial interactions together.

Team Members

- Amaka Ubaka - amakarivas15@gmail.com
- Emmanuel Joseph(JET) - emmanueljet774@gmail.com
- Moyinoluwa Fesobi - moyinoluwafesobi7@gmail.com
- Oluwatosin Esther Ayeni - ayenitosin03@gmail.com
- Norniola Samuel - norniola@gmail.com
- Yinka Enoch Adedokun - adedokunyinkaenoch@gmail.com

Project Description

After the team deliberating on ten(10) different ideas we voted and concluded to build a Decentralized Finance(Defi) chat æpp on the Superhero platform. This is a chat æpp unlike the regular chat apps, it runs on æternity blockchain with the following aims;

- Secure chatting with friends and families,
- Making Sending of funds as easy as sending messages to a friend,
- Group chat functionalities that include sending funds to the group members, and most importantly,
- Having the chat æpp as a subcategory on the Superhero platform.

Inspiration

We were inspired by the idea of building a secure chat platform, on the blockchain where tokens can be sent with minimal loss and optimum security in the form of a message.

What it does

Superchat sends AE tokens from one user to another easily as they chat with friends and family thereby reducing the loss of funds in AE tokens transfer. Superchat runs with three smart contracts that help it perform and build a solution for our inspiration. The contracts are SuperChatProfile, SuperChatFriend, and SuperChatMessage.



The SuperChatProfile contract starts the authentication flow by registering a user and getting the user details. The SuperChatFriend contract connects remotely with the ProfileContract to send a friend request, accept or reject friend requests, get friend requests, and finally get each user friend list.

Lastly, the SuperchatMessage contract performs the work of sending messages and even AE tokens to friends as a message

How we built it

Superchat is built on æternity blockchain and it uses ReactJS for its Front-End implementation. It's an open-source platform on GitHub at <https://github.com/aekiti/superchat>.

The repository contains source codes that sum up superchat. It has its smart contracts written and tested with the AEproject tool. It has its wallet created using the AECLI tool.

The react Front-End implementation uses the AEproject React shape that follows the theme and layout of the voting section of the Superhero.com platform.

Challenges we ran into

1. Saving Objects in a ReactJS state
2. Using the appropriate testnet SDK
3. æpp working only on a desktop with the Superhero.com wallet extension

Accomplishments that we're proud of

A perfectly working unit test script that tests our contract functions and entrypoints, using Google Firebase and GitHub Action to seamlessly deploy our æpp to the live server at <https://superchat.aekiti.com>, and users being able to perform the following activities on the æpp:

1. Register profile using the Superhero.com Backend API
2. Getting User Profile Details
3. Sending Friend Request Functionality
4. Rejecting or Accepting a friend request from users
5. Friends Functionality
6. Chatting with Friends functionality that includes sending of funds

What we learned

1. Making use of AECLI tool to create a new wallet
2. Using the AEproject tool to test our contracts
3. Using the AEproject React shape as our æpp boilerplate.
4. Adding a testnet network to the Superhero Desktop extension.

5. Registering a .chain name programmatically(superchat.chain)
6. Using GitHub Action to communicate with Firebase thereby making the deployment of our chat æpp to the live server smooth.
7. Using the æternity JS SDK to convert AE domination from one to another
8. ReactJS Framework
9. Teamwork amplified the smooth development of our idea. Some of the results can be seen in the [Idea Box Sheet](#) and the [Team Research Document](#)

What's next for Superchat

1. Implementation of the group functionalities, to enable a user to send tokens and messages to members of a created group.
2. Implementing a group saving feature to allow users to save AE token for some time.

Resources

- Frontend design of the chat æpp using the [governance æpp](#) theme and layout.
- Smart contract development using the [AEproject tool](#).
- æpp development using the [AEproject react shape tool](#).
- Testnet account creation([superchat.chain](#)) using the [AECLI tool](#).