Software Requirements Specification (SRS) Document

Project name: Pure Earth Foundation
Team's name: Team 8 (as per Moodle teams list)
Team members: Amey Kunte, Adhiraj Deshmukh, Yash Agrawal, Hariharan Kalimuthu

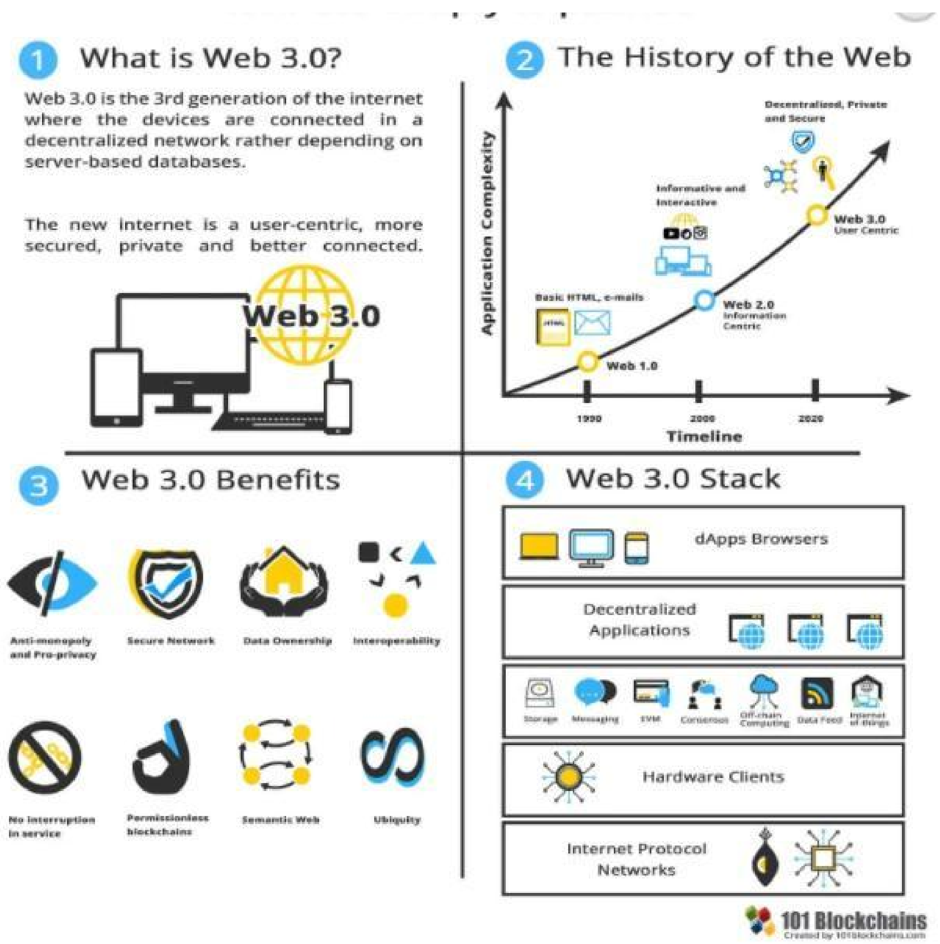
Brief problem statement

Nowadays, the farmers and other low-level workers like small craftsmen, fishermen, etc are getting familiar with the technology but getting small loans is very big of an issue for these individuals.

To address this, we plan to develop a community-based, decentralized cryptocurrency that can be used to give small-scale, short-term loans. The concept is that the amount lent would be too tiny to be worth defaulting on (and suffering the social implications), but large enough to help them advance their work easily.

System Requirements

1. Make a web 3.0 application


1 What is Web 3.0?
Web 3.0 is the 3rd generation of the internet where the devices are connected in a decentralized network rather depending on server-based databases.
The new internet is a user-centric, more secured, private and better connected.
Web 3.0
2 The History of the Web
Application Complexity
Timeline
1990
2000
2020
Web 1.0
Basic HTML, e-mails
HTML
Web 2.0
Information Centric
Informative and Interactive
Web 3.0
User Centric
Decentralized, Private and Secure
3 Web 3.0 Benefits
Anti-monopoly and Pro-privacy
Secure Network
Data Ownership
Interoperability
No interruption in service
Permissionless blockchains
Semantic Web
Ubiquity
4 Web 3.0 Stack
dApps Browsers
Decentralized Applications
Storage
Messaging
EVM
Consensus
Off-chain Computing
Data Feed
Internet of things
Hardware Clients
Internet Protocol Networks
101 Blockchains
Created by 101blockchains.com

2. A general run-through of the application components are

   a. **Javascript**

   b. **Ethereum Provider API + Metamask** (detect-provider, onboarding libraries) for wallet and account/blockchain interaction

   c. **Solidity** to write our contracts that connect our Metamask wallet with our application

   d. **Truffle,** which is a personal blockchain for ethereum development that we will use to deploy contracts. We use **Ganache** that helps us to quickly fire up a personal blockchain with 10 accounts, each with a balance of 100 ETHs for testing our contract

e. An **ERC-20** contract (written in **Solidity**) compiled and deployed beforehand using Truffle

f. **Web3.js** to interact with the Ethereum blockchain and my smart contract

Users profile

Farmers:

Most poor farmers require small-scale loans to facilitate their work, however, most of them can't access liquidity through traditional banks. Such farmers will be the primary beneficiaries of this project since traditional banks are quite difficult to secure loans from for these poor farmers especially due to the lack of assets to be put as collateral and considering the current economy, and taking loans from private individuals where they don't have any goodwill to help these individuals is always a risk.

This maintains them in a state of permanent poverty, prohibiting them from working. There must be a mechanism to deliver financing to these farmers while simultaneously assuring the lenders' security and returns. This is where our platform can help them greatly in securing small scale loans as a decentralized cryptocurrency with some collateral.

Lenders:

The lenders are the people who will provide the capital to the Farmers who post requests on our application and want to acquire funds for their ventures that will aid in their development.

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Rendering the collateral from IPFS	The unique code from IPFS is put as a collateral by the borrower which links to a specific document in the IPFS network. To make it easier for the lender to read the document directly from the app, instead of getting it from IPFS, the document will be rendered for the lender to go through before approving the loan for the borrower.	R2
2.	Status of the transactions	As cryptocurrency is being used as the main mode of transactions, anonymity between the users is one of the most important features. Moreover, as it is on a blockchain, even the third party (our app) is unaware of the identity of the parties involved in the transactions. So a “status bar” is needed to make both the lender and receiver know the current status of the transaction.	R2
3	Connecting our app to metamask wallet	Metamask is one of the most popular wallets used for crypto-currency transactions. Our app will automatically connect to the metamask wallet via their extension as soon as the person logs into the site.	R2
4	Secure Transactions	Make transitions secure from any other users on this platform not involved in the transaction.	R2
5.	Create borrowing / lending platform	Establishing the base protocol on which the borrowing/lending will be based in the communities on our platform.	R2
6.	A simple way to access / lend capital through our cryptocurrency	Creating a very simple way for a person to borrow capital in the form of crypto-currency.	R2