Satoshi-Box Using Blockonomics

Feature Requirements

A. Seller

a. Upload file to sell (max size - 5 mb)
b. Description for the file (optional)
c. Price of the file (choice)
   i. BTC
   ii. BCH
d. Gets a token
   i. Token can be used for
      1. Watching all purchases of the file
      2. Withdraw the net amount to your wallet
e. Get a public URL to share

B. Buyer

a. Uses URL to land on interface
b. Pays using normal checkout interface
c. Gets the file upon confirmation

Technical Design

Schema Design

1. Products Table

a. token - secret token used for withdraw
   i. DataType - **UUID**
b. uid - Public identifier
   i. DataType - **UUID**
   ii. **PRIMARY_KEY**
c. product_name - Mini public description of product
   i. DataType - **string**
d. product_description - Detailed public description of product
   i. DataType - **string**
e. secret_description - description visible only after purchase
   i. DataType - **string**
f. price in fiat
   i. DataType - **float**

- g. currency (USD/EUR/INR etc..)
    - i. DataType - **string**

2. Files Table

- a. id
    - i. DataType - **auto_increment integer**
    - ii. **PRIMARY_KEY**
- b. uid
    - i. DataType - **UUID**
- c. file_data
    - i. DataType - **File**

3. Payments Table

- a. uid
    - i. DataType - **UUID**
- b. address (will be used by buyer to track it's payment status and the files)
    - i. DataType - **string**
    - ii. **PRIMARY_KEY**
- c. crypto (BCH/BTC)
    - i. DataType - **string**
- d. status_of_transaction
    - i. DataType - **Integer**
- e. expected_value - Expected amount in satoshi of BCH/BTC
    - i. DataType - **Integer**
- f. received_value - Received amount in satoshi of BCH/BTC
    - i. DataType - **Integer**
- g. txid
    - i. DataType - **string**
- h. timestamp
    - i. DataType - **DateTime**
- i. order_id (used by buyer to get repeated access to the files)
    - i. DataType - **string**
    - ii. Reference - Link

System API's

1. POST /api/create_record (meant for Seller)

- a. Request Params
    - i. Files
    - ii. BTC/BCH
    - iii. Price
    - iv. Currency
    - v. File Name (visible to everyone)
    - vi. File Description (visible to everyone)
    - vii. Secret Description (visible to paid users)

- b. Response Params

        i. JSON(status_code, uid, token)

2. GET /api/track_record (meant for Seller)
    a. Request Params
        i. uid
        ii. token

    b. Response Params
        i. JSON(status_code, payment_history, net_withdrawable_amount, total_confirmed_purchases)

3. POST /api/withdraw (meant for Seller)
    a. Request Params
        i. token
        ii. uid
        iii. address
    b. Response Params
        i. JSON(status_code, message)

4. GET /api/payment_history (meant for Seller)
    a. **NEEDS TO BE DESIGNED**

5. GET /api/display_record (meant for buyer)
    a. Request Params
        i. uid
    b. Response Params
        i. JSON (status_code, product_name , product_description , price in crypto, type of crypto)
6. GET /api/receive_payment (meant for buyer)
    a. Request Params
        i. uid
    b. Response Params
        i. JSON(status_code, uid, btc_address, order_id, converted_to_crypto_price, start_timestamp, expiry_timestamp, duration)
7. GET /api/track_order (meant for buyer)
    a. Request Params
        i. Order_id
    b. Response Params
        i. JSON(status_code, uid, status, file_name, file_description) + (if payment successful) JSON(secret_description, files)

User Story

1. Buyer
    a. Lands on the website => chooses to buy files
    b. Enters the *UID* to purchase the files
    c. Lands on buying page of that *UID*
    d. Makes the Payment => Gets an *order_id* to track payment status and continuous access to files.
    e. Opens Buyer Tracking Center => enters *order_id* and gets transaction status and if transaction is successful => access to files
2. Seller
    a. Lands on the website => chooses to sell files
    b. Lands on selling page
    c. Seller Uploads Files, chooses payment type (BTC/BCH/both?), choose the amount and currency (USD/INR/EUR...etc) and upload content.
    d. Receives => token (used for withdrawal), uid (that should be shared with others)
    e. Opens Seller Tracking Center => enters *uid* and *token*, a list of purchases appear, net withdrawal amount displayed.
    f. Clicks on withdrawal => enters the address and initiates payment request.