Τεχνολογικό Υπόβαθρο

- Blockchain:
- Ethereum:
- Smart Contract:
- Server:
- JWT:
- SQLite:
- SHA224:
- API:
- Hash:

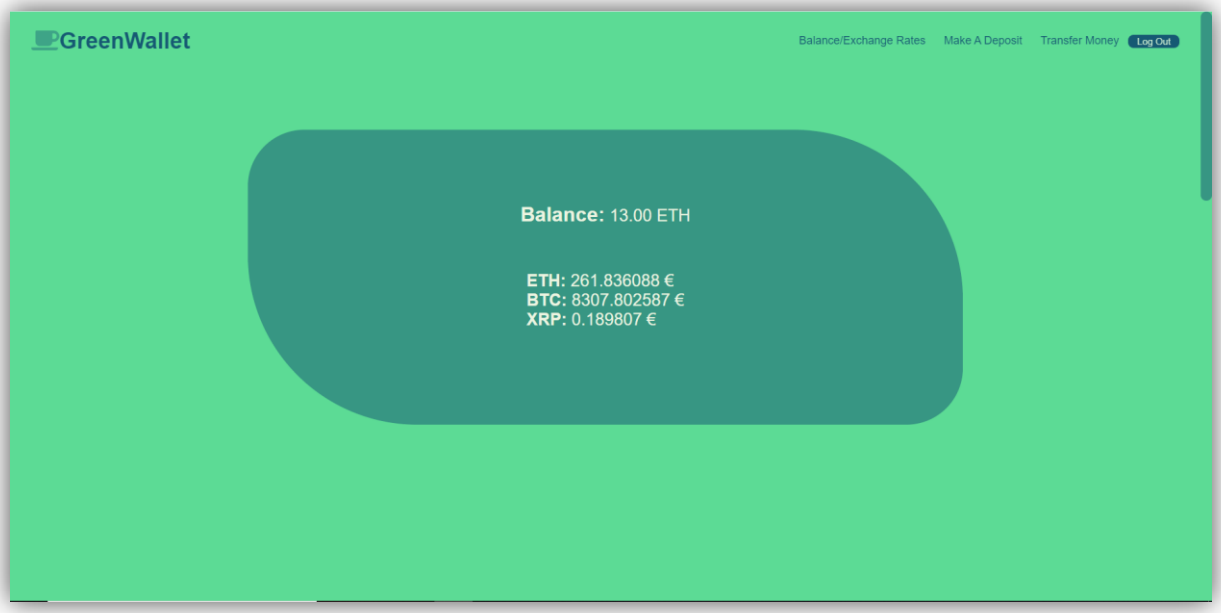
Συντομογραφίες

- F/E - f/e: Front end, το κομμάτι της εφαρμογής με το οποίο διαδρά ο χρήστης
- B/E - b/e: Back end, το κομμάτι της εφαρμογής που υλοποιεί βασικές λειτουργίες οι οποίες σχετίζονται με την βάση δεδομένων
- CRUD: Create / Read / Update / Delete
- JWT: JSON Web Tokens
- ETH: Ethereum
- BTC: Bitcoin
- XRP: Ripple
- ORM: Object Relational Mapper
- API: Application Programming Interface

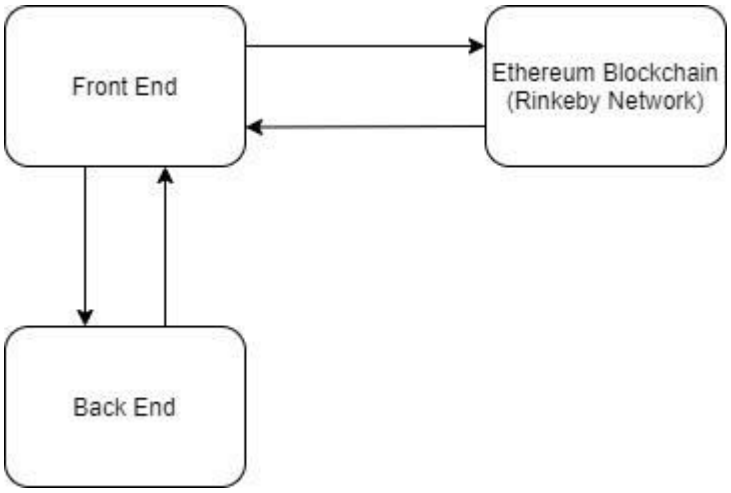
Εφαρμογή GreenWallet

Το GreenWallet είναι μια εφαρμογή η οποία αποτελεί ένα ψηφιακό πορτοφόλι το οποίο δίνει την δυνατότητα στους χρήστες του να εκτελούν συναλλαγές με καταστήματα τα οποία επιτρέπουν την πληρωμή μέσω του Ethereum Blockchain. Οι συναλλαγές πραγματοποιούνται με την μεταφορά χρημάτων από το πορτοφόλι του χρήστη στο smart contact του αντίστοιχου μαγαζιού. Επιπλέον υποστηρίζονται συναλλαγές μεταξύ χρηστών, από πορτοφόλι σε πορτοφόλι χωρίς την χρήση smart contact.

// Η φωτογραφία θα αλλάξει με την τελευταία έκδοση



Αρχιτεκτονική Υψηλού Επιπέδου



Για την λειτουργία της εφαρμογής, το κομμάτι με το οποίο διαδρά ο χρήστης πρέπει να επικοινωνεί με το b/e και το Ethereum blockchain. Το b/e αναλαμβάνει κυρίως το βάρος της ταυτοποίησης του χρήστη και της αποθήκευσης βασικών δεδομένων για την ομαλή λειτουργεία της εφαρμογής. Με την επικοινωνία με το Ethereum blockchain γίνεται δυνατή η μεταφορά χρημάτων σε κάποιο smart contract ή σε άλλο πορτοφόλι χρήστη του GreenWallet. Η μεταφορά χρημάτων σε δημόσια διεύθυνση που δεν ανήκει στο GreenWallet είναι επίσης δυνατή. Πιο συγκεκριμένα:

Λειτουργίες που εκτελεί το b/e είναι:

- Αποθήκευση βασικών πληροφοριών για τα καταστήματα / smart contracts που έχει αποθηκεύσει ο χρήστης
- CRUD στα δεδομένα των καταστήματα / smart contracts
- Αποθήκευση αναγκαίων πληροφοριών για την ταυτοποίηση του χρήστη
- Ταυτοποίηση του χρήστη
- Δημιουργία “session” για την ομαλότερη λειτουργία της εφαρμογής και την καλύτερη εμπειρία περιήγησης του χρήστη (JWT)
- Ενημέρωση της εφαρμογής με τις πιο πρόσφατες συναλλαγματικές αξίες των τριών πιο χρησιμοποιημένων κρυπτονομισμάτων ETH, του BTC και του XRP
- Παραγωγή mnemonic για την δημιουργία καινούργιων πορτοφολιών

Λειτουργίες που εκτελούνται στο Blockchain:

- Μεταφορά χρημάτων από πορτοφόλι σε πορτοφόλι μεταξύ χρηστών χωρίς την μεσολάβηση smart contract
- Πληρωμές σε καταστήματα που υποστηρίζουν την πληρωμή μέσω του GreenWallet
- Ενημέρωση για το υπόλοιπο του λογαριασμού του χρήστη

Λειτουργίες του f/e:

- Σύνδεση και προβολή όλων των λειτουργιών που προσφέρουν το b/e και το blockchain
- Δημιουργία μιας ευχάριστης εμπειρίας για τον χρήστη

Ανάλυση υλοποίησης των λειτουργιών της εφαρμογής GreenWallet

Αποθήκευση αναγκαίων πληροφοριών για την ταυτοποίηση του χρήστη:

Η διαδικασία της αποδοχής των στοιχείων, τα οποία εισάγει ο χρήστης, και η εισαγωγή τους στην βάση γίνεται στο b/e κομμάτι της εφαρμογής. Η εφαρμογή χρησιμοποιεί την SQLite για την διαχείριση της σχεσιακής βάσης δεδομένων στην οποία αποθηκεύονται όλα τα δεδομένα της εφαρμογής. Για την ευκολότερη ενσωμάτωση και χρήση της βάσης δεδομένων, χρησιμοποιείται ένα ORM το SQLAlchemy.

Στην βάση δεδομένων ένας χρήστης αναπαρίσταται από τα παρακάτω στοιχεία:

- **Id**: Ένας μοναδικός ακέραιος αριθμός ο οποίος ξεχωρίζει τον χρήστη από όλους τους άλλους χρήστες. Ο αριθμός αυτός συμπληρώνεται απευθείας από την βάση δεδομένων κατά την δημιουργία του χρήστη.
- **Username**: Το όνομα με το οποίο αναπαρίσταται ο χρήστης μέσα στην εφαρμογή. Προσοχή, το Username ενός χρήστη δεν αποτελεί στοιχείο με βάση το οποίο μπορούμε να τον ξεχωρίσουμε από τους άλλους χρήστες. Δύο χρήστες επιτρέπεται να έχουν το ίδιο Username.
- **Password**: Το συνθηματικό του χρήστη, το οποίο είναι αναγκαίο για την ταυτοποίηση του στην εφαρμογή. Ο κάθε χρήστης μπορεί να επιλέξει το συνθηματικό του κατά την δημιουργία του λογαριασμού του. Για την μεγαλύτερη ασφάλεια των δεδομένων των χρηστών, κανένα συνθηματικό δεν αποθηκεύεται στην βάση δεδομένων. Στην βάση δεδομένων αποθηκεύεται το αποτέλεσμα που προκύπτει από την εφαρμογή της συνάρτησης κατακερματισμού SHA224 στον κωδικό που παρέχει ο χρήστης.
- **Mnemonic**: Μια σειρά 12 τυχαίων λέξεων, στην αγγλική γλώσσα, οι οποίες χρησιμοποιούνται για την παραγωγή του ιδιωτικού κλειδιού του πορτοφολιού του χρήστη. Παραδείγματος χάριν: ‘almost spice garment loop slight blanket copper sun empty cement work sail’. Το Mnemonic παράγεται αυτόματα από την εφαρμογή, όταν ο χρήστης δημιουργεί τον λογαριασμό του.

```
class User(base):
    __tablename__ = 'Users'

    Id = Column(Integer, primary_key=True, autoincrement=True)
    Username = Column(String)
    Password = Column(String)
    Mnemonic = Column(String)
```

// Κλάση η οποία περιγράφει την δομή του πίνακα του χρήστη στο ORM

Οι βασικές λειτουργίες που υλοποιούνται πάνω στα δεδομένα του χρήστη είναι:

- **Η δημιουργία νέου χρήστη:**

  Η δημιουργία ενός νέου χρήστη γίνεται με την υποβολή της αντίστοιχης φόρμας στο f/e κομμάτι της εφαρμογής. Κατά την υποβολή γίνεται χρήση του εκτεθειμένου API υπεύθυνου για την δημιουργία νέων χρηστών, το οποίο βρίσκεται (τοπικά) στο `'http://127.0.0.1:5000/createUser/'`. Ο κώδικας που τρέχει με την κλήση του API είναι υπεύθυνος για την υποδοχή των δεδομένων, την παραγωγή ενός μνημονικού, την δημιουργία του hash που θα αποθηκευτεί στην θέση του συνθηματικού του χρήστη και τελικά την αποθήκευση των δεδομένων στην βάση. Για την συγγραφή πιο κατανοητού αλλά και επεκτάσιμου κώδικα, όλες αυτές οι λειτουργίες δεν εκτελούνται μέσα σε μια μέθοδο αλλά σε πολλές μεθόδους οι οποίες εκτελούν μια συγκεκριμένη λειτουργία.