Chqbook Payment API

Python language with Django framework has been primarily used for building api's and integration with razorpay. MySQL has been used as the database.

Steps to integrate in the project:
1)Git clone the project pc_rcpb from github
2)Create a python3 virtualenv by using the command *virtualenv python3 -v env_name*
3)activate the virtual environment
4)Install the necessary requirements for the project by using command pip3 install -r requirements.txt
5)Create a mysql database with name chqbook and configuration details as provided in the settings file
6)Run the command *python manage.py migrate* to create the required tables.

Database Modelling
According to the requirements of the project, mainly 4 tables have been created (Order, OrderNoteMapping, Payment, Wallet)

- **Order Table**
  This table is primarily used for saving the orders created associated with any payment. It contains *order_id* (a unique key used for storing the order id for the associated payment), *amount, amount_paid, amount_due, currency* (used for transactions), *receipt, offer_id,*
  status (**any payment has 5 associated status which are created, authorized, captured, refunded and failed which have represented by integer keys),** attempts, active (is order active or not) and created and updated columns to store the order.

- **OrderNoteMapping Table**
  Table used to save all the notes associated with the order.

- **Payment Table**
  Model used to save all payment data associated with an Order
  Contains columns such as payment_id, order_id (order associated with payment), invoice_id, is_international_payment, method of payment, is_amount_refunded, refund_status (can have 4 possible values ie initiated, processing, refunded, cancelled), card_id, vpa and fee and tax related details. Besides it also contains the wallet_id (pk of Wallet Table) if mode of payment is wallet.

- **Wallet Table**
  Model containing details of all the different wallet partners including merchant_id, merchant_key and industry_type.

API

- **CREATE ORDER**
  endpoint : /chqbook/api/v1/create-order/
  method : **POST**
  payload --> {
  "amount" : 1000,
  "currency" : "INR",
  “notes” : {
  “note1” : “sample data 123”, “note2” : “SOme data”}
  }

  Here the amount and currency are necessary parameters. This api basically creates the order for which payment needs to be created.

- **CAPTURE PAYMENT**
  endpoint : /chqbook/api/v1/capture-payment/
  method : POST
  payload :{

  ```
     "payment_id" : "pay_8912oUBi66xm2d",
     "amount" : "1000",
     "currency" : "INR"
  }
  ```

  This api is basically used to verify if the amount deducted from a user's bank account is the same as the amount he paid via razorpay. It basically takes payment id,amount and currency as request data and firstly checks if corresponding payment has been initiated , if not will raise an error and based on response from razorpay api, it gives us the the status if captured or failed.