Payment Backend Logic and Steps

1. Setup Project

- Create new projects **payment_prj**
- Create new app (core)
- Install Apps in Settings.py
- Configure Templates
- Configure Static and Media Files in settings and Urls.py
- Create a new view, urls and template then runserver.
- Configure template inheritance and partials

2. Configure Admin Page, Superuser and Jazzmin

- Install **Jazzmin** (pip install django-jazzmin)
- Add jazzmin in **INSTALLED_APPS**
- Add Jazzmin Config Code in Settings.py
- Create Superuser
- Login To Admin Section

3. Custom User Model

- Create custom `class User(AbstractUser):` in **models.py**
- Add **AUTH_USER_MODEL = 'userauths.User'** in settings.py
- Login to admin with **email and password**

4. User Register System

- Create new form `class UserRegisterForm(UserCreationForm):` in forms.py
- Write view to **register** `def RegisterView(request):` user
- Configure template to show form
- Login to website from Frontend

5. User Login System

- Write view to **login** `def LoginView(request):` user
- Configure template to grab input field

- Login to website from Frontend

6. User Logout System

- Write view `def LogoutView(request):` to Logout a user
- Configure URL

7. Account Model

- Create new `class Account(models.Model)`**:** in account models.py
- Configure model (account) in admin.py
- View Account Model In Admin Section

8. Django Signal to Create Account

- Create new function models.py to create and save user profile using post_save
- Test In Admin Section by Creating a New User

9. KYC Model and Form

- Create new **app** Account
- Create new **class KYC(…):** in models.py
- Configure model (KYC) in admin.py
- Create KYC Forms in Forms.py
- Test Model in Admin Section

10. KYC Views

- Create new view to display and process the KYC Form
- Configure View in URLS
- Create new kyc.html Template
- Display Form in KYC template

11. Alerts In Django

- Grab Alert Snippet from Bootstrap (version 4)

- Copy and Paste CDN
- Write alert conditional statements

12. View to Display All Account Information

- Create new view to get account details (`account_view`)
- Get all the details of a user from the KYC Form and Display in template

13. Transfer - Get users by Account number

14. Transfer - Choose and Validate Amount to be sent

- Choose amount to be sent
- Validate the Amount to be sent
- Store in current amount in model

15. Transfer - Process Amount to be Sent

- Create new function, get account and other objects
- Store the current **amount** to be sent and **details** in model.
- Create URL for the New function
- Add URL in form

16. Transfer - Confirm Transfer and Process the Transaction and Complete Transfer Process

- Create New function in payment request to confirm the transfer
- Create another function to Process the transfer to be sent
- Create last function to Shows successfully sent transfer.

17. Create Transaction View and Function

- Filter transaction for sender and receiver differently
- Create transaction detail page

18. Django Context Processor

- Create and configure django context processor to pass data to all the pages all at once.

19. Request Payment - Get/Search users by Account number

- Create new function to search for users in DB
- Write search user query
- Create URLS.py
- Configure Templates

20. Choose and Validate Amount to be Received

- Choose amount to be received
- Validate the Amount to be received
- Store in current amount in model

21. Process Amount to be Received

- Create new function, get account and other objects
- Store the current **amount** to be recieved and **details** in `PaymentRequest` model.
- Create URL for the New function
- Add URL in form

22. Confirm Request

- Create New function to confirm the payment request.

23. Process the Request

- Create another function to Process the request to be sent

24. Complete Complete request

- Create the last function to Show successfully sent requests.

25. Request Settlement - Get all Sent and Received Request Transactions

- Create a new function `AllPaymentRequest` in payment_request.py and write queries to get all payment requests.
- Create URLs.py
- Configure the Template

26. Request Settlement - Confirm Payment Settlement

- Create a new function `SettlePaymentsConfirmation` in payment_request.py and write queries to get all payment details for confirmation.
- Create URLs.py
- Configure the Template

27. Request Settlement - Processing Payment Settlement

- Create a new function `SettlePaymentsConfirmation` in payment_request.py and write code to process the payment.
- Create URLs.py
- Add the URL to the Pay button

28. Request Settlement - Complete Payment Settlement

- Create a new function `SettlePaymentsCompleted` in payment_request.py and write code to get just completed payment.
- Create URLs.py
- Configure the Template and Conclude Tutorial.

29. Adding Credit Card to System

- Create a new model to save credit card to system
- Register ne model in admin.py
- Makemigrations and migrate then vie the ne data

30. Add Credit Card Form

- Create a form for credit card form using django forms

31. Display All User Credit card on dashboard

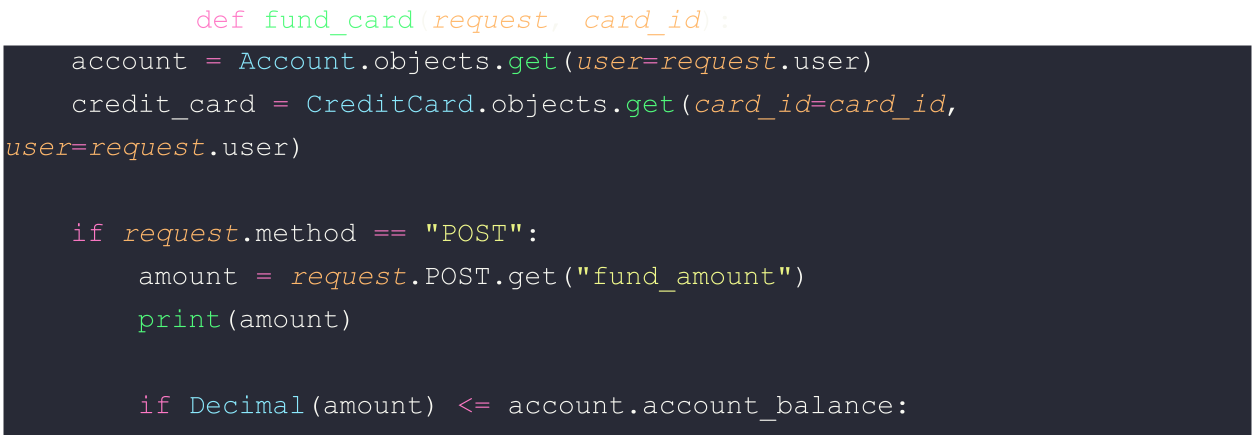
- Create a query to filter selected users card and display in dashboard card section

32. Credit Card Detail Page

- Create a query to get single card detail and update the card page

33. Fund Credit Card from Paylo Account

- Create a function to process and fund selected credit card



```
def fund_card(request, card_id):
    account = Account.objects.get(user=request.user)
    credit_card = CreditCard.objects.get(card_id=card_id, 
user=request.user)

    if request.method == "POST":
        amount = request.POST.get("fund_amount")
        print(amount)

        if Decimal(amount) <= account.account_balance:
```

```
        account.account_balance -= Decimal(amount)
        account.save()

        credit_card.amount += Decimal(amount)
        credit_card.save()

        messages.success(request, "Funding Successfull")
        return redirect("core:card-detail", credit_card.card_id)
    else:
        messages.warning(request, "Insufficient Funds")
        return redirect("core:card-detail", credit_card.card_id)
```

34. Withdraw Cash from Credit Card Available Balance

- Create a function to process and withdraw from selected credit card