**Table of Content**

**1. Introduction**

**2. Frontend Flow**

**3. Backend Flow**

**4. Flow Diagrams**

**5. DB Structure**

**6. APIs**

**1. Introduction**

The HolyPennies project rounds up charity donations and sends them to different houses of worship based on the users' preferences. Basically, this is a clone of (roundupapp.com) except it is for donations to Houses of Worship (Temples, Churches, Synagogues, Mosques, etc.).

The idea for the project is that there will be different user accounts that do different things like House of Worship admins, regular donor users, and Super Donations can be made through both Stripe (https://stripe.com) and Plaid (https://plaid.com), and users can also see how these payments are tabulated so that they have a dashboard.dashboard. Each user can see the breakdown of their donations..

**2. Frontend Flow**

**Holy Pennies Frontend Flow**

**(1) Splash**

· Holy Pennies LOGO

· Holy Pennies "Do Good By Thoughtful Giving™"

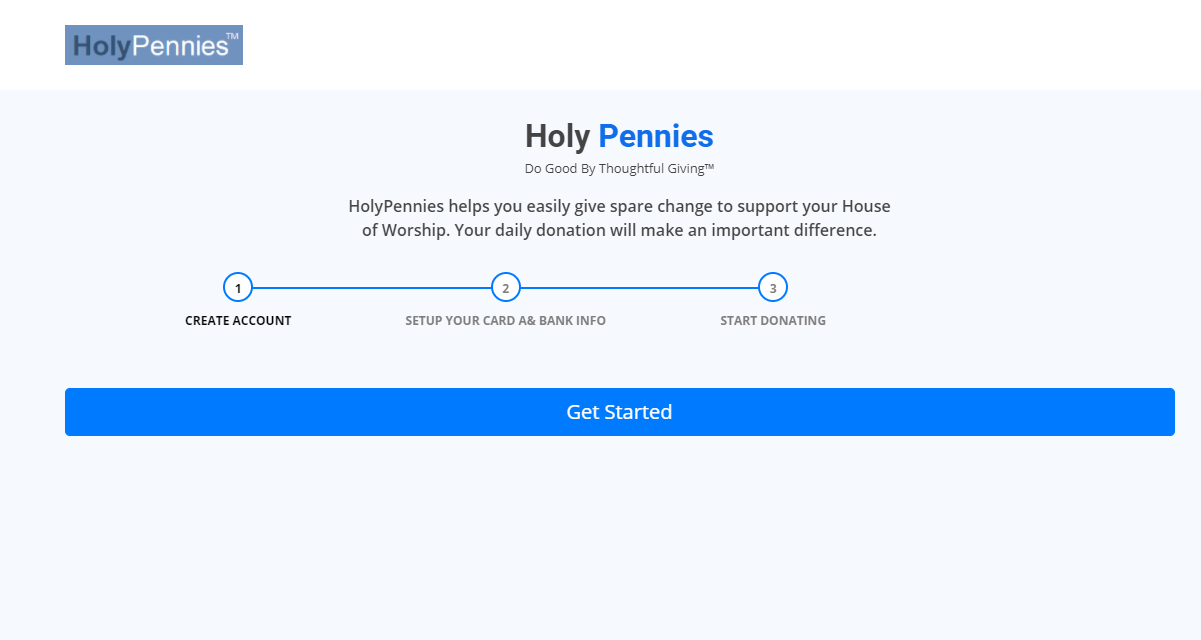


**(2) Welcome to Holy Pennies**

· Holy Pennies LOGO

· Holy Pennies helps you easily give spare change to support your House of Worship. Your daily donation will make an important difference.

Get Started (Button)



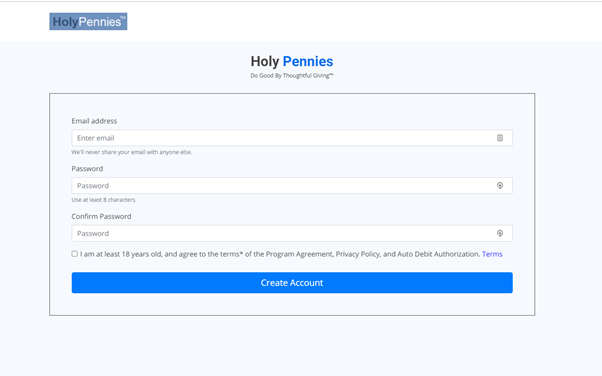
**3) Create an Account**

· Holy Pennies LOGO

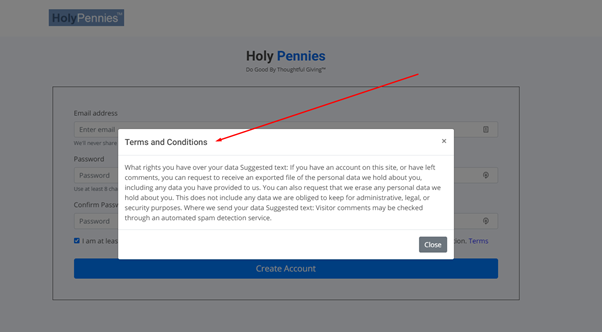
· Email (user ID)

· Password (have ability to show password)

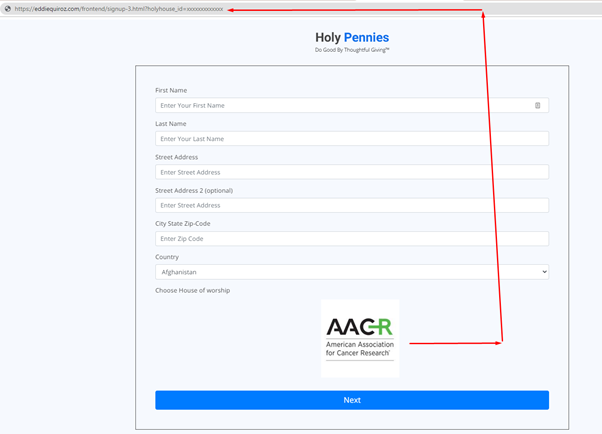
"I am at least 18 years old, and agree to the terms\* of the Program Agreement, Privacy Policy, and Auto Debit Authorization



\*each term will be hyperlinked and will pop up to be read, then top corner X to close and return back to the Create An Account screen



(4) Ask information from user and get holy house from using the id on URL (will be provided in email )



**(5)) LogIn (Persists) or step complete**

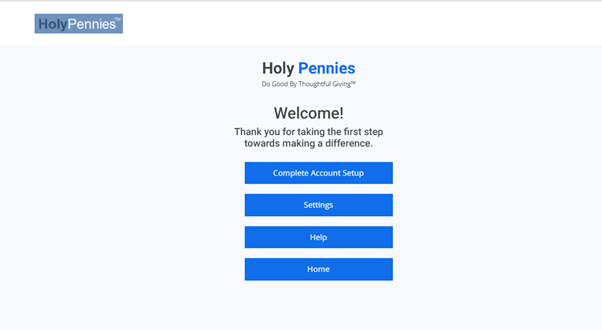
HolyPennies LOGO

Welcome! Thank you for taking the first step toward making a difference.

· Account Setup (link)

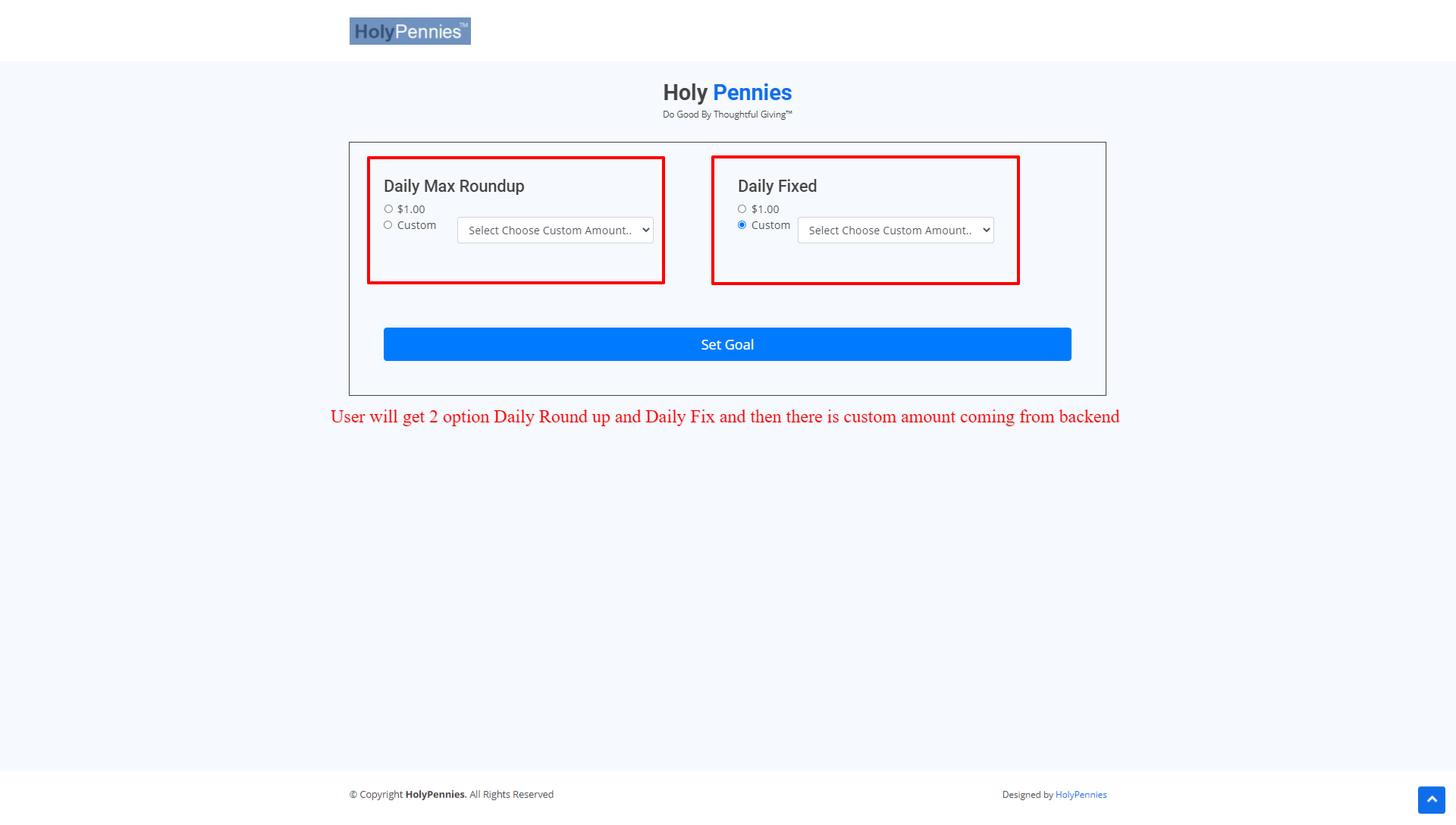
· Settings (link)

· Help (link)

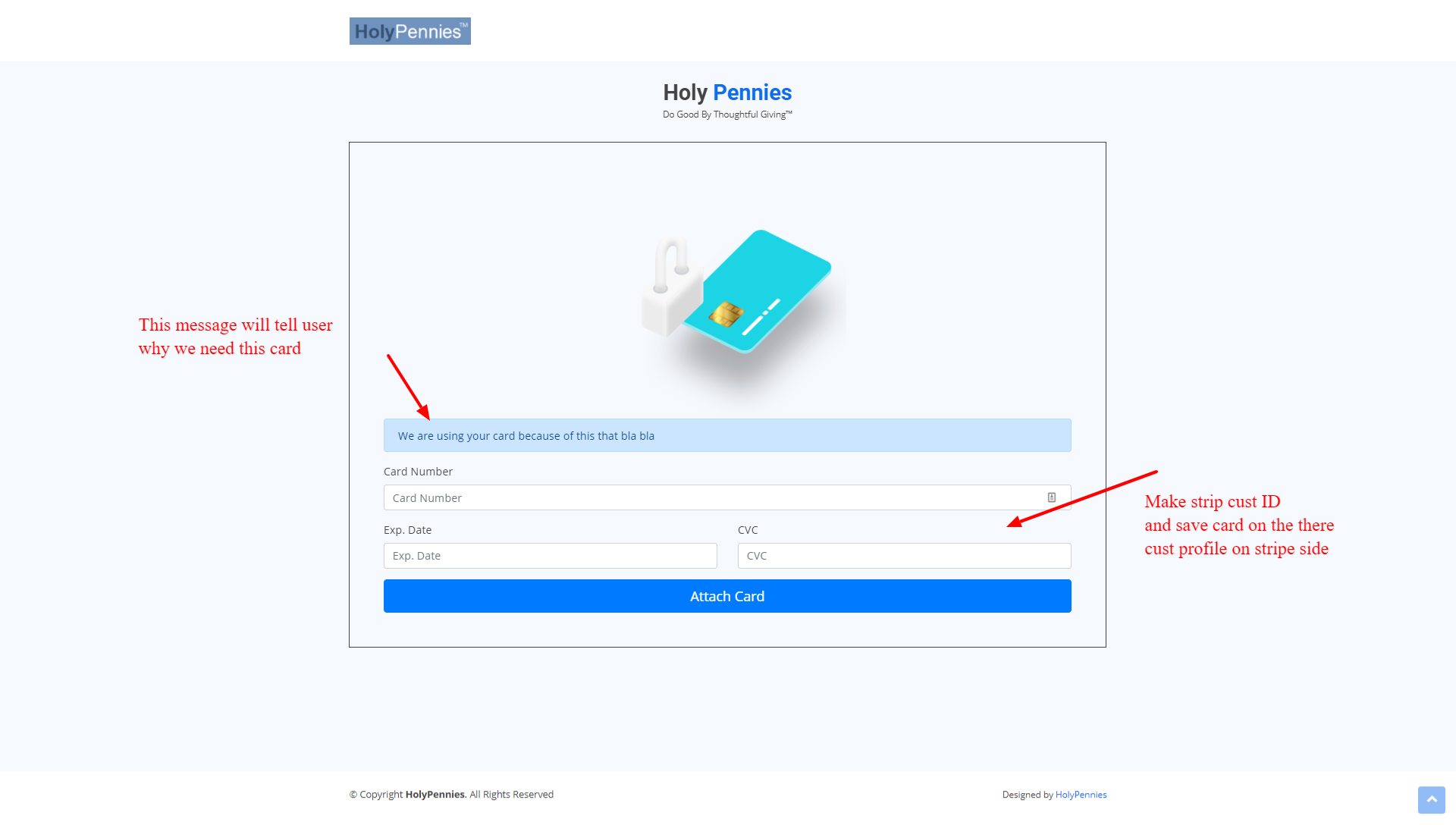


**Screenshot for Card Setup and other stuff**

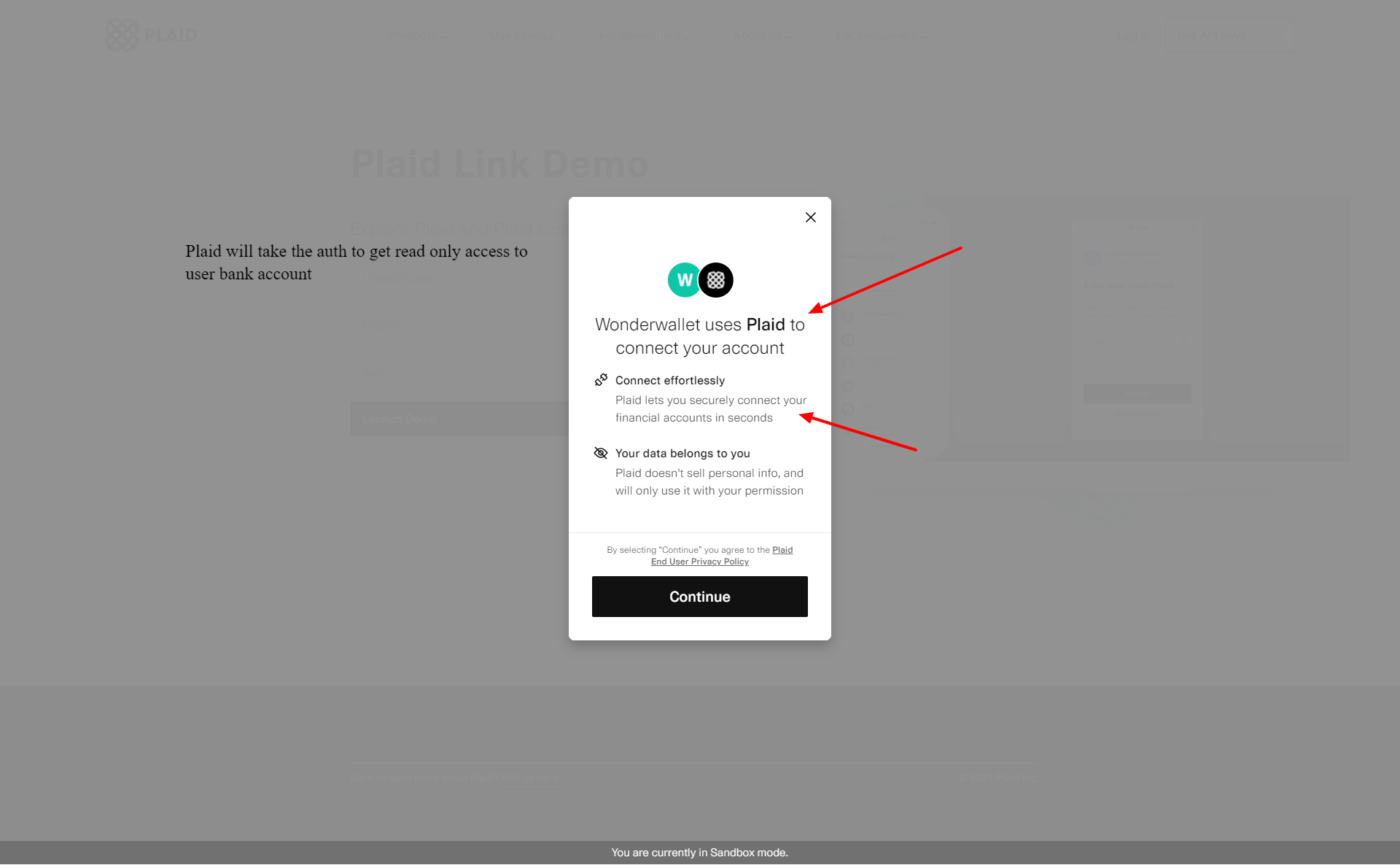
Set Goal ([**https://tinyurl.com/yey4h7bc**](https://tinyurl.com/yey4h7bc))



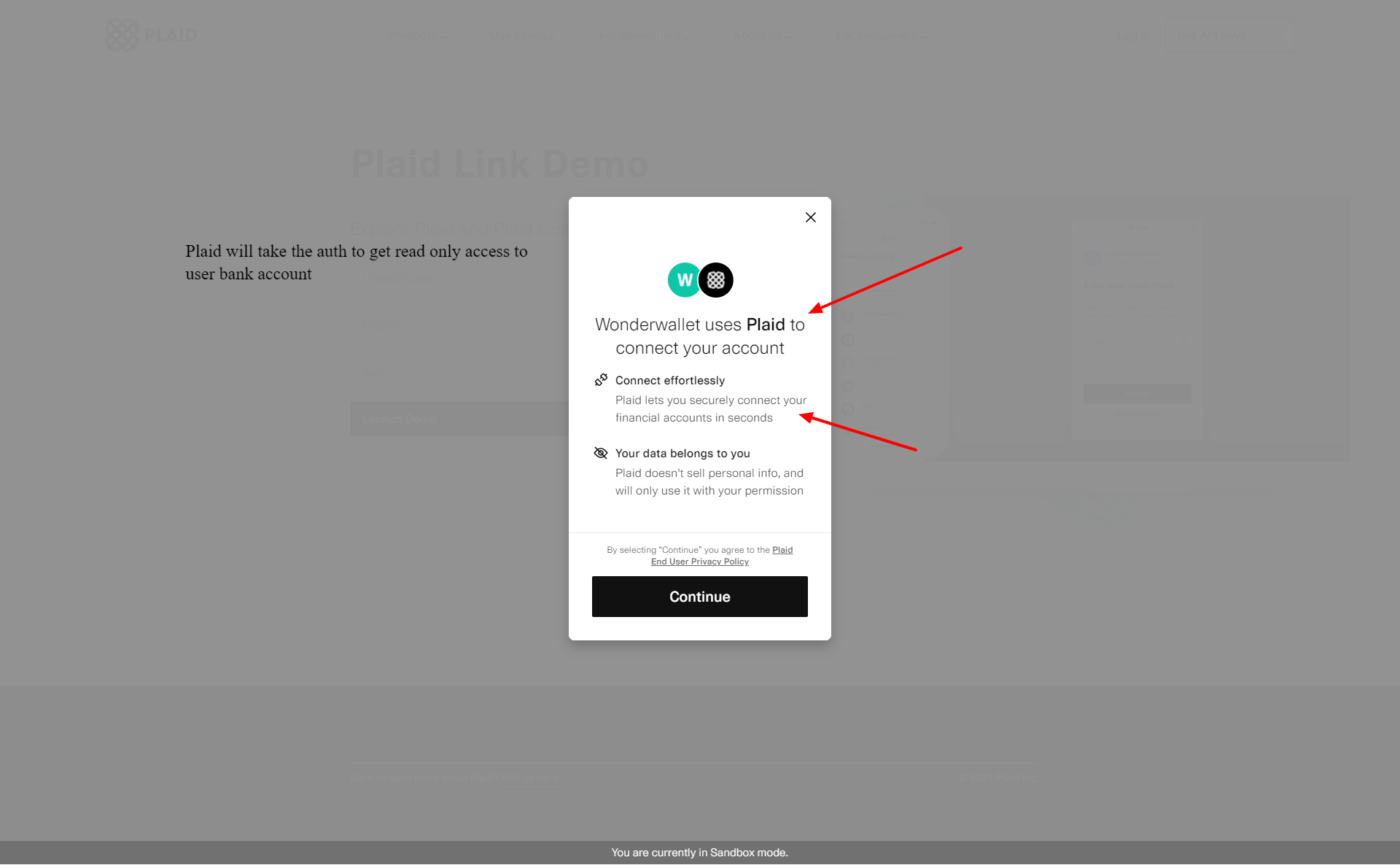
Card Attach [(https://tinyurl.com/yetex2jk)](https://tinyurl.com/yetex2jk))



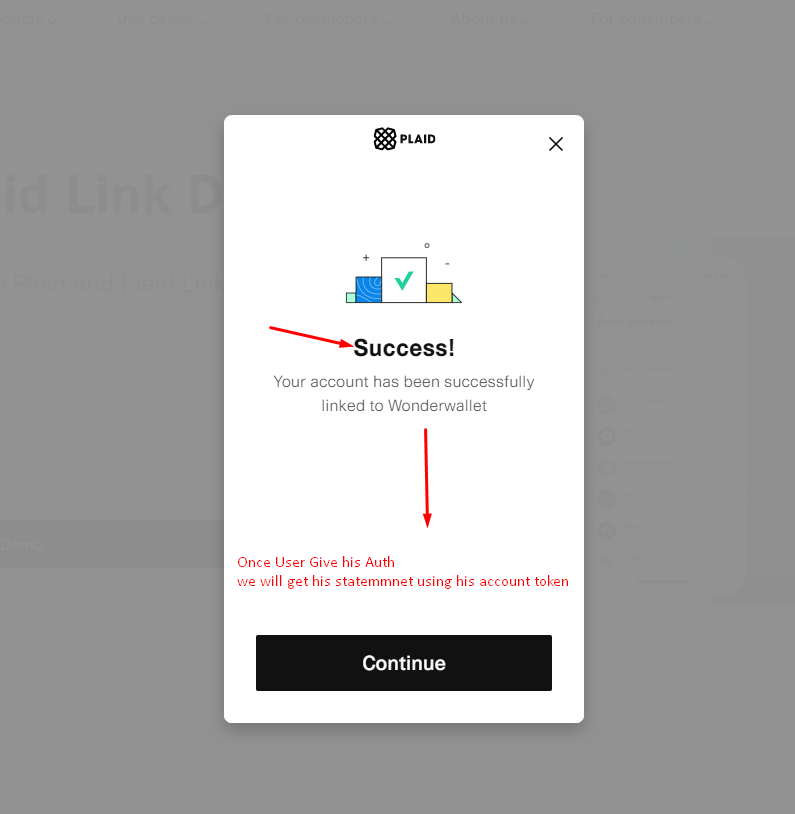
**Plaid Setup (**[**https://tinyurl.com/yg3ehmec**](https://tinyurl.com/yg3ehmec)**)  
\* Full working Clip (**[**https://screenrec.com/share/EvC86wNBQg**](https://screenrec.com/share/EvC86wNBQg)**)**



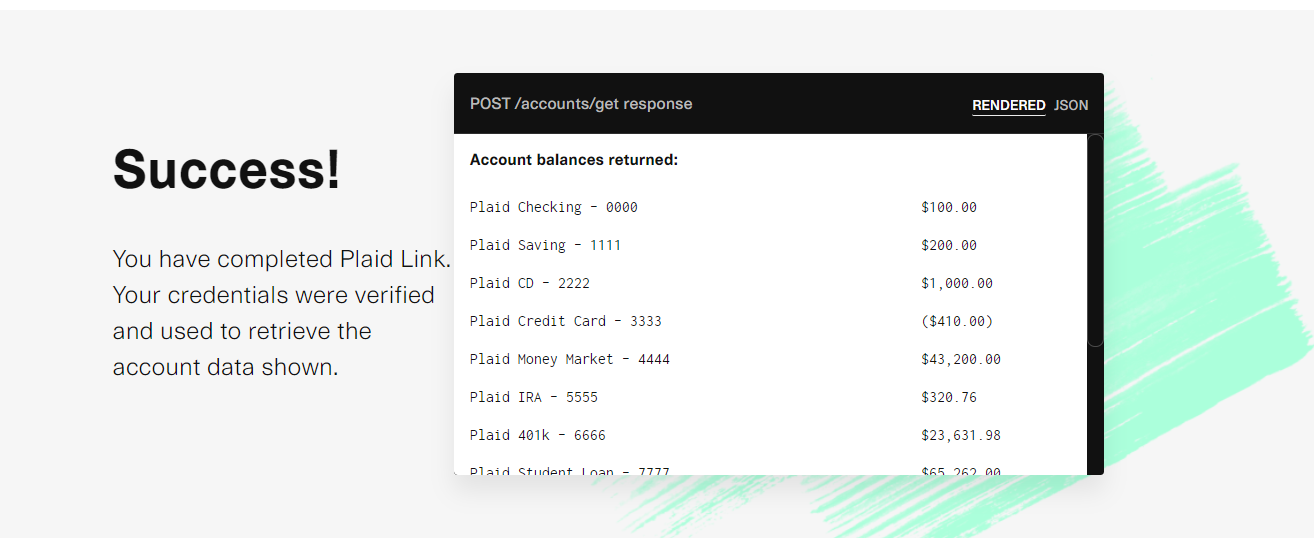
**Choose bank Account (**[**https://tinyurl.com/yjhefw7q**](https://tinyurl.com/yjhefw7q)**)**

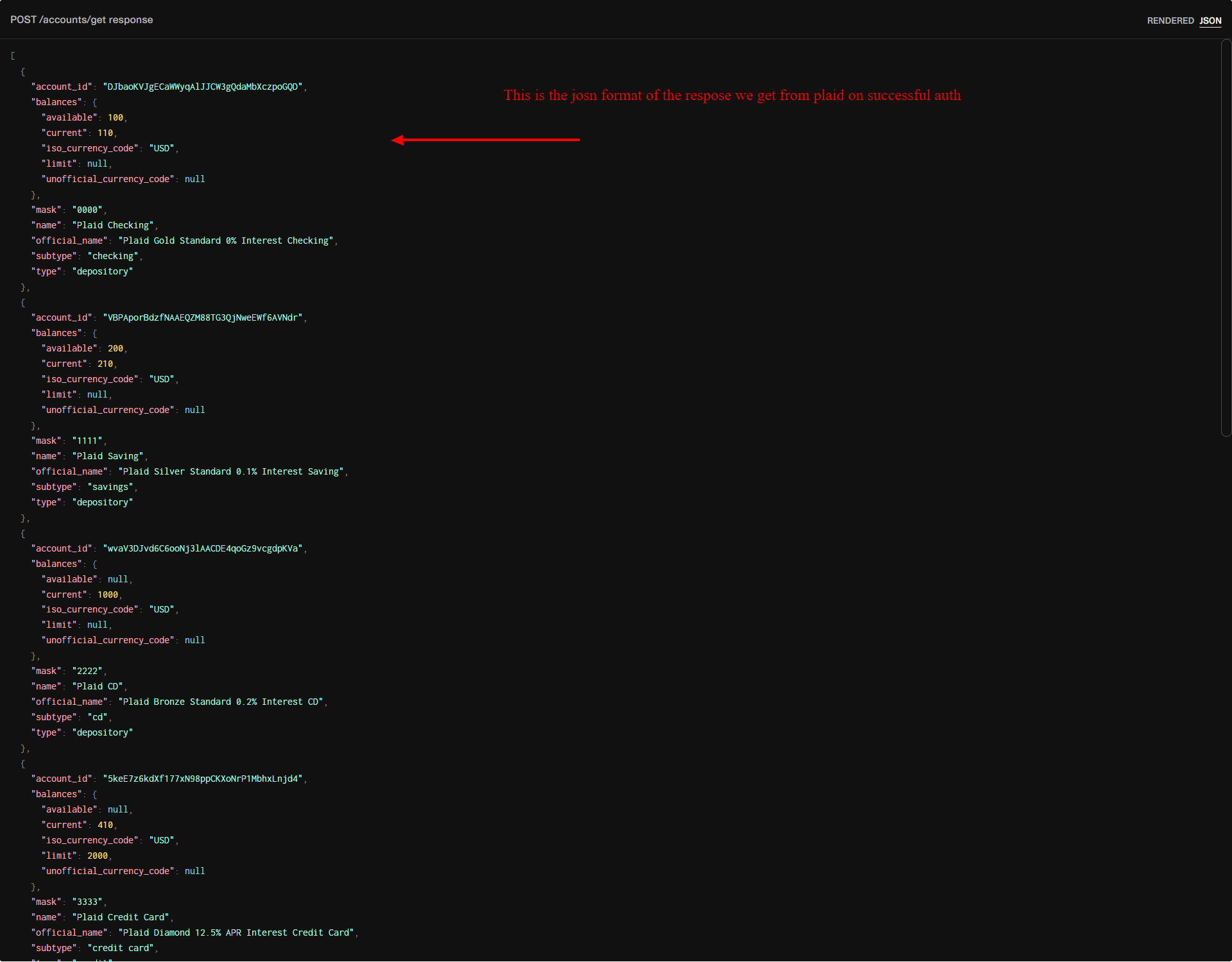


**Success and get account statement (**[**https://prnt.sc/1n4urbt**](https://prnt.sc/1n4urbt)**)**



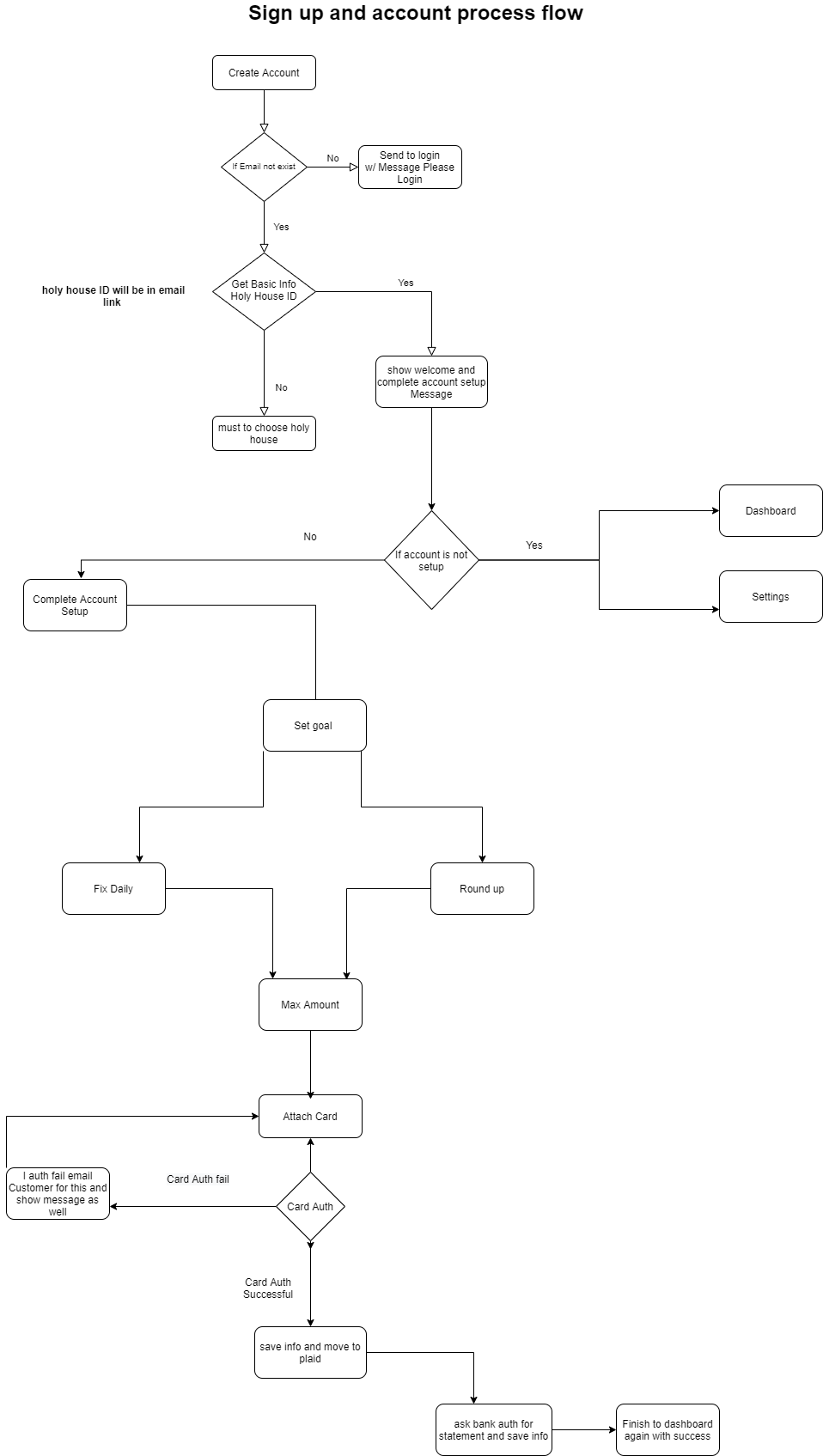
**User Data will be start sync (**[**https://prnt.sc/1n4war9**](https://prnt.sc/1n4war9)**)**



**IN JSON FORMAT (**[**https://tinyurl.com/yhj2x4xo**](https://tinyurl.com/yhj2x4xo)**)**

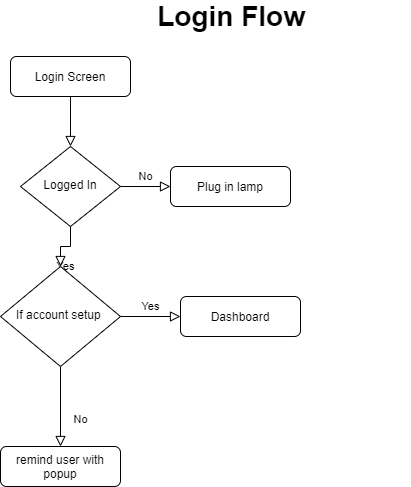
**Signup Flow Diagram**

**(**[**https://prnt.sc/1lunrlx**](https://prnt.sc/1lunrlx)**)**



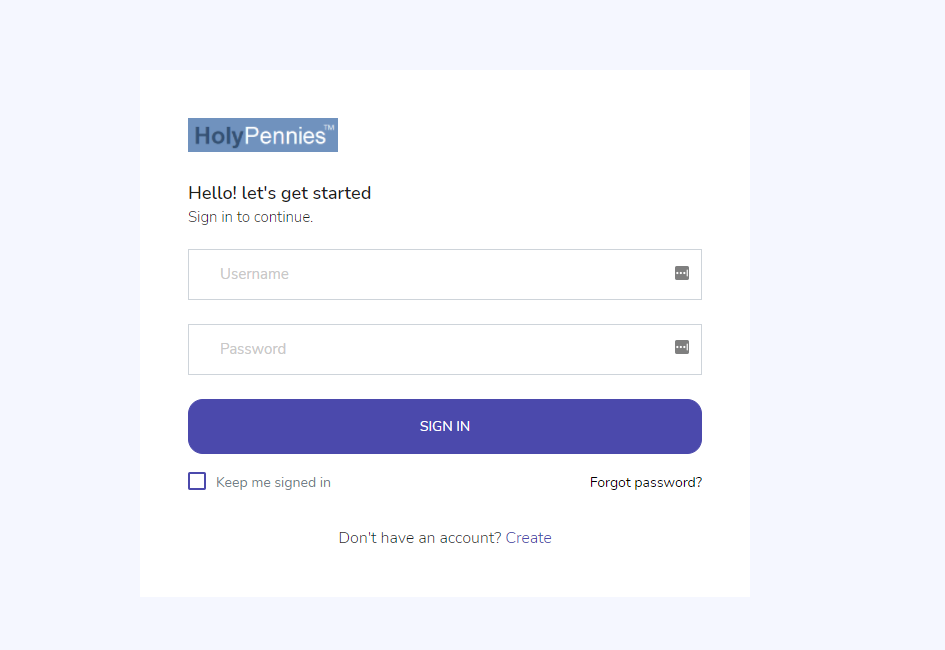
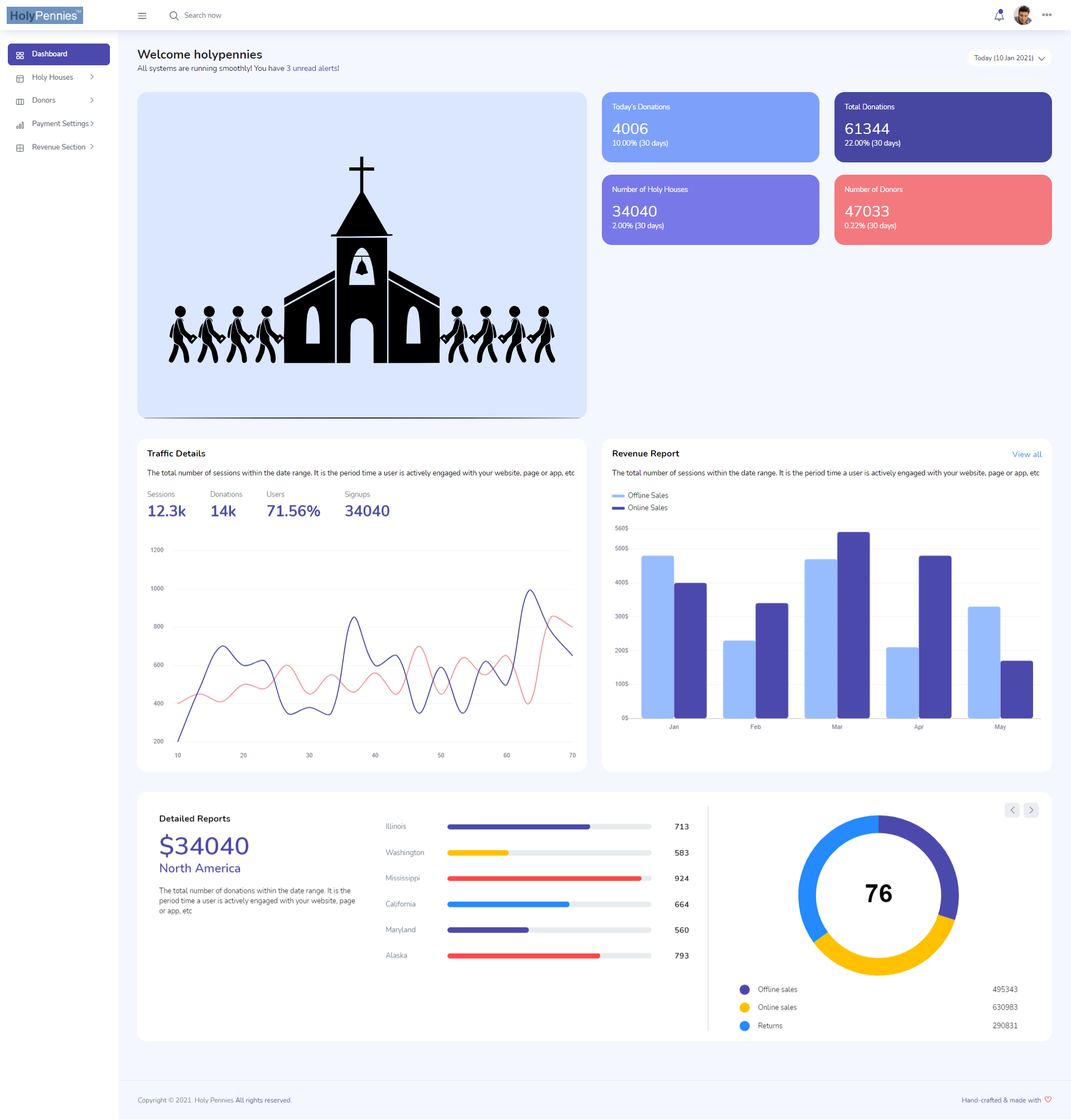
**Login Flow Diagram**

**(**[**https://prnt.sc/1lunrlx**](https://prnt.sc/1lunrlx)**)**

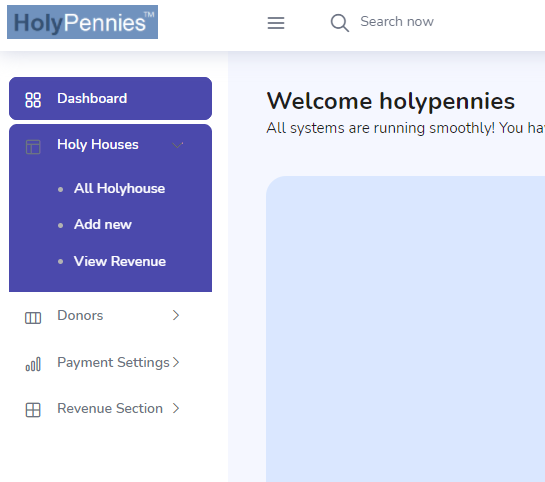
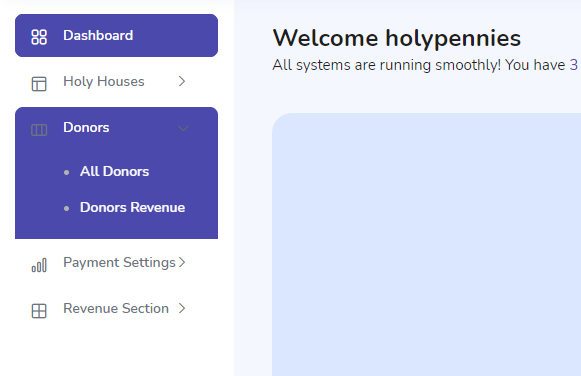
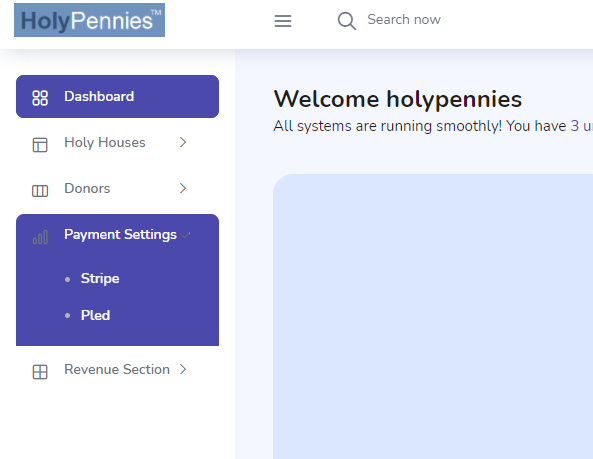


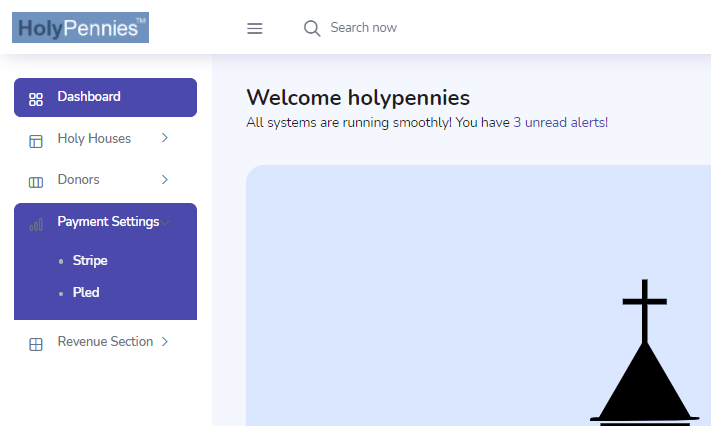
**3. Backend Flow**[**https://prnt.sc/1p8w6p8**](https://prnt.sc/1p8w6p8)

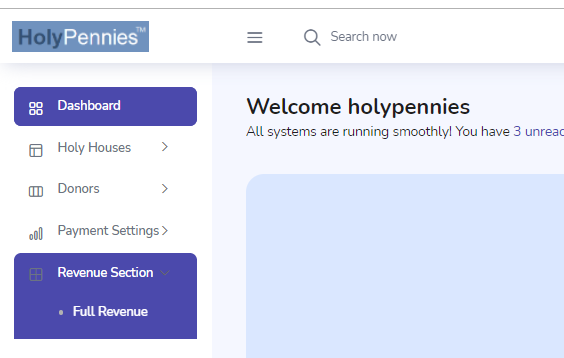
Sign Screen for admin will be like this

**Dashboard Screen:(**[**https://tinyurl.com/yk5to2lt**](https://tinyurl.com/yk5to2lt)**)**

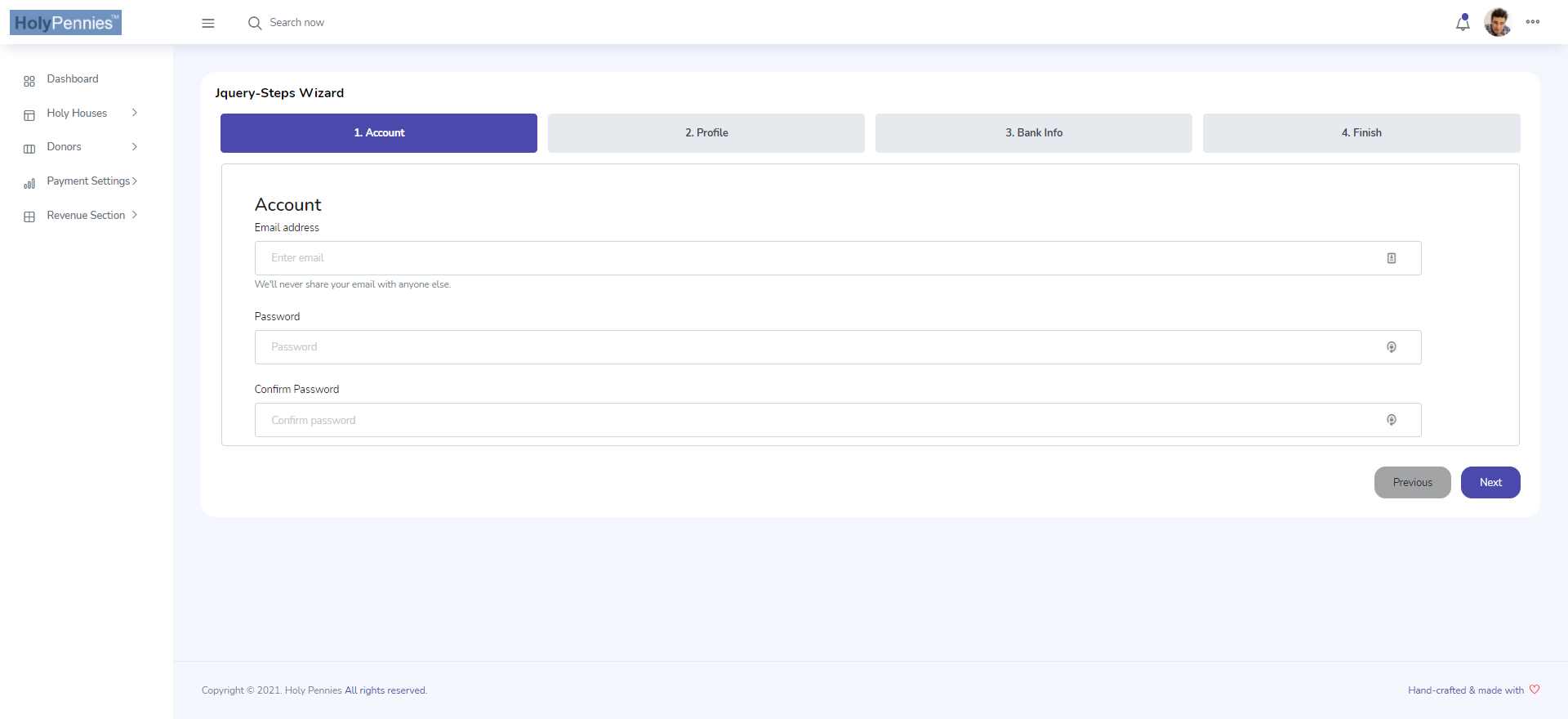
**Side bard menu on admin side**

****

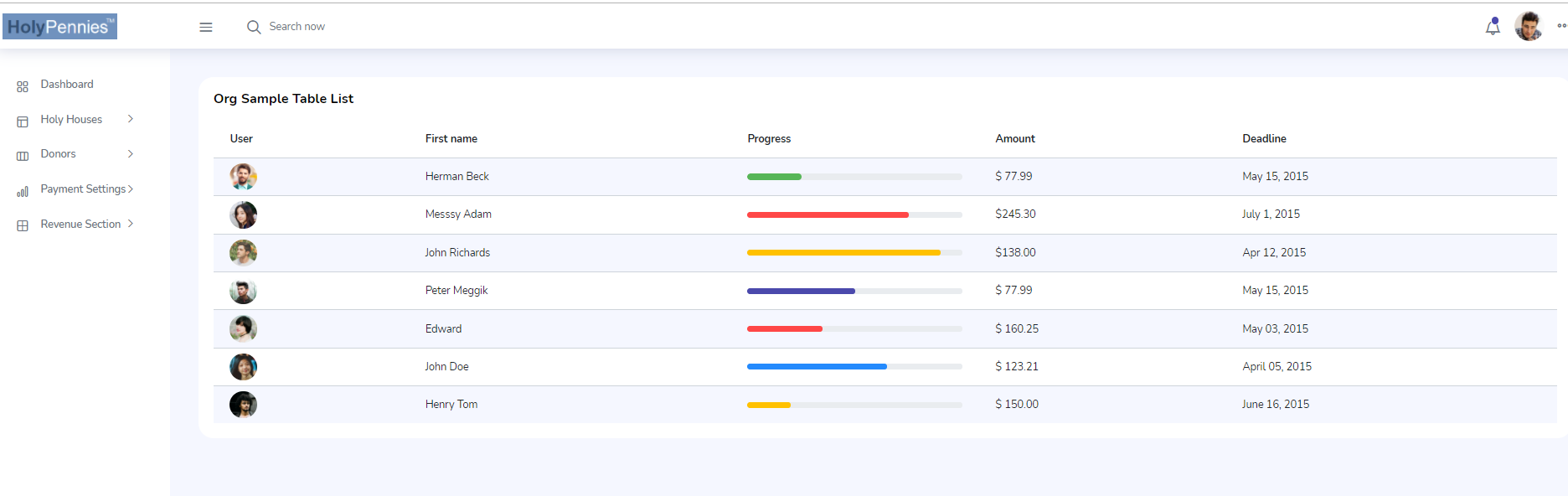
****

****

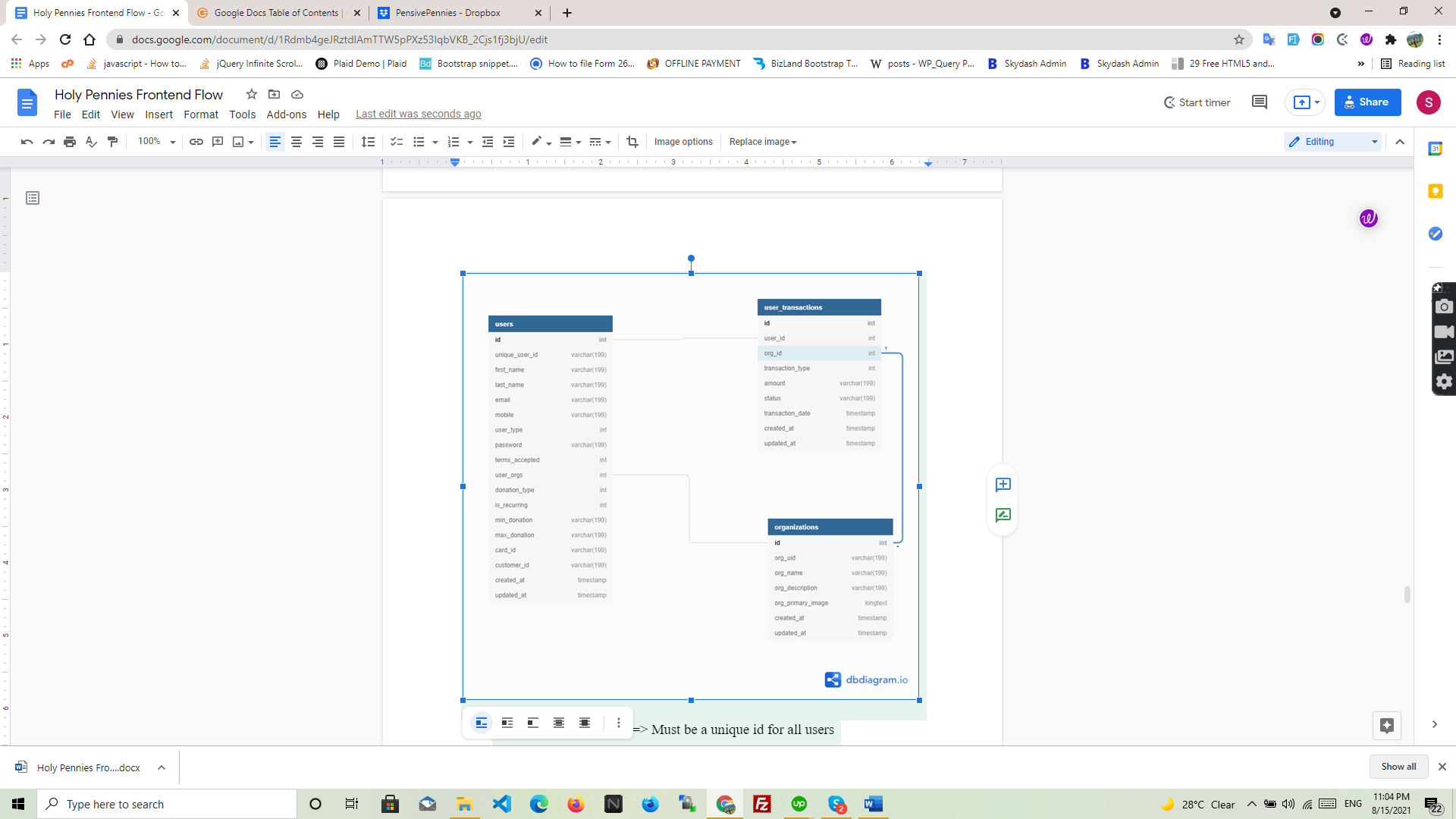
**Add Org Options are available on admin side too**

****

**List all Org can be done on admin side**

****

**\*note All CRUD option will be there for admin and as well as user and other settings for admin**

**4. Basic Database Schema** 

● Unique\_user\_id ===> Must be a unique id for all users

● First\_name ====> will save first name of user

● Last\_name ===> will save last name of user

● Email ===> must be unique (so need to check before signup)

● Mobile ⇒ must be unique( so need to chcek before signup)

● Password ⇒ must be Bcrypt

● Terms\_accepted ==>is mandatory

● User\_org ===> will be selected from org list on signup page

● Is\_recurring ===> set 1 for if its recurring else set 0 if not

● Min\_donation ===> as per org min donation

● Max\_donation -===> as per user limit how much he want to donate at max

● Card\_id ====> when user give card info we have to save card on stripe end acdn save card id here

● Customer\_id====> this is stripe customer id

● Created\_at ===>server timestamp for when record is created

● Updated\_at ===>server timestamp for when record is updated

**====================USER TYPES=================**

● 1 for Donors

● 2 for Organisations

● 3 for admins

**====================Donation TYPES=================**

● 1 for Round ups

● 2 for fixed amount

**Transactions**

(please note this table will be changed as per our 3rd party api interaction this is basic structure which will be there additional column can be added to table)

● User\_id ( id from user table its a forign key )

● Org\_id ( id from org table its a forign key )

● Transaction type( its type of transaction round up or fixed)

● Amount ( how much amount is cur from user account)

● Status ( status of transaction)

● Transaction date ( date of transaction)

● Created\_at ===>server timestamp for when record is created

● Updated\_at ===>server timestamp for when record is updated

**Organizations**

(please note this table will be changed as per our later changes this is basic structure which will be there additional column can be added to table)

● Org\_id ( Must be a unique id for all holyhouses)

● Org name ( full legal name of organisation)

● org\_desription( small org description )

● Organisation primary image( can be a logo or image )

● Created\_at ===>server timestamp for when record is created

● Updated\_at ===>server timestamp for when record is updated

\*additional columns like bank information etc will be added to the system later

**5. HolyPennies Frontend Donor Platform API**

POST

<https://eddiequiroz.com/hpn/login>

Parameter:

Email: email of the user

Password : password of user

**\*Using this Api, users will be able to login to the donor platform on the front-end API must be protected with passport authentication or jwt token auth**

POST

<https://eddiequiroz.com/hpn/signup>

Parameter:

Email: email of the user

Password : password of user

**\*In this Api, users will be able to sign up for the donor platform and be given a passport authentication token or a JWT token authentication token, which can be used in all other API calls**.

POST

<https://eddiequiroz.com/hpn/get_holyhouse_info>  
Parameter:holyhouse id

**\*In this Api, we will show holy house logo and description to user on signup page**

POST

<https://eddiequiroz.com/hpn/set_goal>  
Parameter:oAuth Token

**\*In this Api, user prefer goal settings will save to database**

POST

<https://eddiequiroz.com/hpn/attach_card>

oAuth Token

\***In this Api,create stripe customer ID and save card to that customer id and save stripe id on database**

POST

<https://eddiequiroz.com/hpn/auth_plaid>

oAuth Token

\***In this Api,save required param from plaid to DB and create a job in job tabel for sync statement data**

GET

<https://eddiequiroz.com/hpn/get_profile>

Parameter:

oAuth Token

\***In this Api, user will auth by the token and backend will provide the profile data to frontend**

POST

<https://eddiequiroz.com/hpn/update_profile>

Parameter:

oAuth Token

Profile image

Name ….etc etc

\***In this Api, user will auth by the token and profile data updated as per fields on frontend including the image**

POST

<https://eddiequiroz.com/hpn/donate_now>

oAuth Token

\***In this Api, user will do instant donation and transaction will saved to database**

POST

<https://eddiequiroz.com/hpn/download_transaction_data>

oAuth Token

\***In this Api, user will able to download historic transaction data**

GET

<https://eddiequiroz.com/hpn/get_transaction>

oAuth Token

\***In this Api, user’s transaction data will display using this**