

# GRUNDY LLC

## Step-by-Step Integration Plan

- 1) First setup Paystack Account & Subaccounts (in Test and Live): This can be done programmatically or on the paystack dashboard after creating an account.
  - a) Create a Paystack main account for Grundy LLC.

```
annaoczor@Annas-MacBook-Pro ~ % curl https://api.paystack.co/subaccount \
-H "Authorization: Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe" \
-H "Content-Type: application/json" \
-d '{
    "business_name": "Grundy LLC",
    "settlement_bank": "Parallex Bank",
    "account_number": "1181393128",
    "percentage_charge": 10
}'

[{"status":true,"message":"Subaccount created","data":{"business_name":"Grundy LLC","account_number":"1181393128","percentage_charge":10,"settlement_bank":"Parallex Bank","currency":"NGN","bank":26,"integration":1595601,"domain":"test","account_name":"ANNA IDONGESIT OZOR","product":"collection","managed_by_integration":1595601,"subaccount_code":"ACCT_esglysqtcmv72gb","is_verified":false,"settlement_schedule":"AUTO","active":true,"migrate":false,"id":1485480,"createdAt":"2025-10-11T19:37:09.570Z","updatedAt":"2025-10-11T19:37:09.570Z"}]%
```

	Name	Bank Details	Split	Code	Status	Actions
<input type="checkbox"/>	Grundy LLC	Parallel Bank 1181393128	90%	ACCT_esglysqtcmv72gb	Verified	<button>Update</button> <button>Delete</button>

- a) Create subaccounts for each merchant (if splitting revenue): Example: GreenLeaf Farms 90% to merchant, 10% to Grundy LLC. Copy Subaccount Codes for later.

	Name	Bank Details	Split	Code	Status	Actions
<input type="checkbox"/>	Grundy LLC	Parallel Bank 1181393128	90%	ACCT_esglysqtcmv72gb	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	GreenLeaf Farms	Abbey Mortgage Bank 0006141068	10%	ACCT_pjvax63lv6q25yt	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	Fruitopia Distributors	Wema Bank 0413972743	10%	ACCT_reg0ko8dpjpj0432u	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	Root Harvest	Sterling Bank 0106986100	10%	ACCT_thazk7ft7ev8aeu	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	GrainHub Stores	Wema Bank 7650402903	10%	ACCT_2evfpmpqbcrg7gh5	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	Prime Proteins Ltd	Standard Chartered Bank 5006244261	10%	ACCT_bcutid6dtph9i7l	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	SpiceRoute Traders	Globus Bank 6791717421	10%	ACCT_qnjh11t72dqqjx	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	FoodieMart Distributions	Opay Digital Services Limited (Opay) 8139312776	10%	ACCT_d8b6443q3ik4xqj	Verified	<button>Update</button> <button>Delete</button>
<input type="checkbox"/>	HomeEase Essentials	Parallel Bank 1181393128	10%	ACCT_j2khgn1w93ck4yy	Verified	<button>Update</button> <button>Delete</button>

Grundy LLC		Subaccounts																																																																																						
1595601		<input type="checkbox"/> Test <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>																																																																																						
<input type="checkbox"/> Home <input type="checkbox"/> PAYMENTS <input type="checkbox"/> Transactions <input type="checkbox"/> Customers <input type="checkbox"/> Refunds <input type="checkbox"/> Payouts <input type="checkbox"/> Disputes <input type="checkbox"/> Transaction Splits <input type="checkbox"/> Subaccounts <input type="checkbox"/> Terminals <input type="checkbox"/> Direct Debit  <input type="checkbox"/> RECURRING <input type="checkbox"/> Subscribers <input type="checkbox"/> Plans <input type="checkbox"/> Subscriptions  <input type="checkbox"/> COMMERCE <input type="checkbox"/> Payment Pages <input type="checkbox"/> Products <input type="checkbox"/> Storefronts <input type="checkbox"/> Orders		Sort By: Date Created <input type="button" value="Search subaccounts"/> <span style="float: right;"> <input type="button" value="+ New Subaccount"/> <input type="button" value="Verify Subaccounts"/> </span>																																																																																						
		<table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>Name</th> <th>Bank Details</th> <th>Split</th> <th>Code</th> <th>Status</th> <th>Update</th> <th>Delete</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td><td>HomeEase Essentials</td><td>Parallel Bank 1181393128</td><td>90%</td><td>ACCT_qu4orm9azyfhb1f</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>FoodieMart Distributions</td><td>OPay Digital Services Limited (OPay) 8139312776</td><td>90%</td><td>ACCT_7bneke4zjamsn9</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>SpiceRoute Traders</td><td>Globus Bank 6791717421</td><td>90%</td><td>ACCT_a4s16tbnoq2jyx3</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>Prime Proteins Ltd</td><td>Standard Chartered Bank 5006244261</td><td>90%</td><td>ACCT_k0rum73h2lk8ax</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>GrainHub Stores</td><td>Wema Bank 7650402903</td><td>90%</td><td>ACCT_m1wjwqa4aszb8wo</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>Root Harvest</td><td>Sterling Bank 0106986100</td><td>90%</td><td>ACCT_q2yygwy46swma0l</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>Fruitopia Distributors</td><td>Wema Bank 0413972743</td><td>90%</td><td>ACCT_ox60ao3wmamko55</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>GreenLeaf Farms</td><td>Abbey Mortgage Bank 0006141068</td><td>90%</td><td>ACCT_bq9pkqqdutvtfy2</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> <tr> <td><input type="checkbox"/></td><td>Grundy LLC</td><td>Parallel Bank 1181393128</td><td>90%</td><td>ACCT_esglysqtcmv72gb</td><td><span>Verified</span></td><td><a href="#">Update</a></td><td><a href="#">Delete</a></td></tr> </tbody> </table>							<input type="checkbox"/>	Name	Bank Details	Split	Code	Status	Update	Delete	<input type="checkbox"/>	HomeEase Essentials	Parallel Bank 1181393128	90%	ACCT_qu4orm9azyfhb1f	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	FoodieMart Distributions	OPay Digital Services Limited (OPay) 8139312776	90%	ACCT_7bneke4zjamsn9	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	SpiceRoute Traders	Globus Bank 6791717421	90%	ACCT_a4s16tbnoq2jyx3	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	Prime Proteins Ltd	Standard Chartered Bank 5006244261	90%	ACCT_k0rum73h2lk8ax	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	GrainHub Stores	Wema Bank 7650402903	90%	ACCT_m1wjwqa4aszb8wo	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	Root Harvest	Sterling Bank 0106986100	90%	ACCT_q2yygwy46swma0l	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	Fruitopia Distributors	Wema Bank 0413972743	90%	ACCT_ox60ao3wmamko55	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	GreenLeaf Farms	Abbey Mortgage Bank 0006141068	90%	ACCT_bq9pkqqdutvtfy2	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>	<input type="checkbox"/>	Grundy LLC	Parallel Bank 1181393128	90%	ACCT_esglysqtcmv72gb	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>
<input type="checkbox"/>	Name	Bank Details	Split	Code	Status	Update	Delete																																																																																	
<input type="checkbox"/>	HomeEase Essentials	Parallel Bank 1181393128	90%	ACCT_qu4orm9azyfhb1f	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	FoodieMart Distributions	OPay Digital Services Limited (OPay) 8139312776	90%	ACCT_7bneke4zjamsn9	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	SpiceRoute Traders	Globus Bank 6791717421	90%	ACCT_a4s16tbnoq2jyx3	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	Prime Proteins Ltd	Standard Chartered Bank 5006244261	90%	ACCT_k0rum73h2lk8ax	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	GrainHub Stores	Wema Bank 7650402903	90%	ACCT_m1wjwqa4aszb8wo	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	Root Harvest	Sterling Bank 0106986100	90%	ACCT_q2yygwy46swma0l	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	Fruitopia Distributors	Wema Bank 0413972743	90%	ACCT_ox60ao3wmamko55	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	GreenLeaf Farms	Abbey Mortgage Bank 0006141068	90%	ACCT_bq9pkqqdutvtfy2	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	
<input type="checkbox"/>	Grundy LLC	Parallel Bank 1181393128	90%	ACCT_esglysqtcmv72gb	<span>Verified</span>	<a href="#">Update</a>	<a href="#">Delete</a>																																																																																	

Edit Subaccount

---

**Currency**

NGN

**Bank Name (For payouts)**

Abbey Mortgage Bank

**Account Number**

0006141068

**Account Name**

GreenLeaf Farms

**Account Alias (Optional)**

GreenLeaf Farms

**Transaction Split**

Your share of payment (%)      Subaccount gets (%)

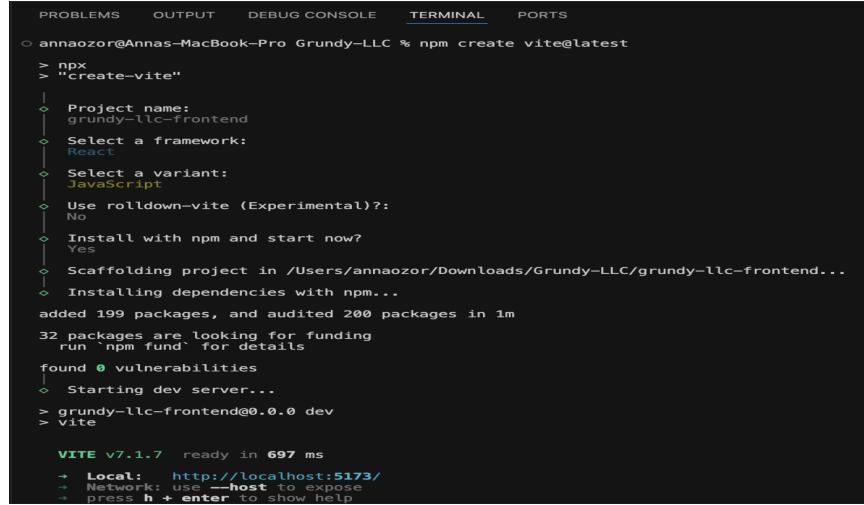
90      10

---

- 2) Developing the Grundy LLC Groceries Marketplace. For this, I chose to go with the MERN Stack. MERN - MongoDB, Express.js, React.js, and [Node.js](#). Since I am building a full-stack MVP, from experience working with JavaScript framework for building scalable, high-performance web applications provides a perfect balance of performance, simplicity, speed and efficiency as well as ease of use. Additionally, development with Javascript end-to-end simplifies development.
- a) Setup Backend Server with [Node.js](#) and Express.js. The combination is ideal for handling real-time payment webhooks, order processing and inventory updates. Express simplifies building RESTful APIs that connect the frontend to Paystack and MongoDB.
    - Install dependencies: `npm install express axios cors dotenv body-parser`
  - b) For the frontend, I used [React.js](#) which provides a component-based architecture for building reusable UI blocks, which is perfect for the pages I built like the merchant dashboard, order tracking, grocery listing, etc. Since I am building for a modern e-commerce user experience, [React.js](#) offers a fast, single-page application (SPA) experience. For the Frontend App, I usednReact Router for payment routes and react-paystack library for frontend Paystack integration
    - Install dependencies:

```
npm create vite@latest
npm install @paystack/paystack-sdk --save
```

SEE SCREENSHOTS BELOW



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
○ annaozor@Annas-MacBook-Pro Grundy-LLC % npm create vite@latest
> npx
> "create-vite"
|
| Project name: grundy-lhc-frontend
| Select a framework: React
| Select a variant: JavaScript
| Use rolldown-vite (Experimental)?: No
| Install with npm and start now? Yes
| Scaffolding project in /Users/annaozor/Downloads/Grundy-LLC/grundy-lhc-frontend...
| Installing dependencies with npm...
added 199 packages, and audited 200 packages in 1m
32 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
| Starting dev server...
> grundy-lhc-frontend@0.0.0 dev
> vite
VITE v7.1.7 ready in 697 ms
+ Local: http://localhost:5173/
+ Network: use -host to expose
+ press h + enter to show help
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
● annaozor@Annas-MacBook-Pro backend % npm install @paystack/paystack-sdk --save
up to date, audited 163 packages in 3s
25 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
○ annaozor@Annas-MacBook-Pro backend %
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
● annaozor@Annas-MacBook-Pro frontend % npm install react-paystack
added 13 packages, and audited 242 packages in 9s
39 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
○ annaozor@Annas-MacBook-Pro frontend %
```

Create [server.js](#)

```

GrundyLLC
http://localhost:4000
JS server.js

// server.js
1 // Import express from "express";
2 import express from "express";
3 import cors from "cors";
4 import { connectDB } from "./config/db.js";
5 import groceryRouter from "./routes/groceryRoute.js";
6 import userRouter from "./routes/userRoute.js";
7 import cartRouter from "./routes/cartRoute.js";
8 import orderRouter from "./routes/orderRoute.js";
9 import merchantRouter from "./routes/merchantRoute.js";
10 import paymentRoute from "./routes/paymentRoute.js";
11 import paystackRoute from "./routes/paystackRoute.js"; // webhook route
12 import path from "path";
13 import { fileURLToPath } from "url";
14 import dotenv from "dotenv/config";
15
16 const app = express();
17 const port = 4000;
18
19 // Connect to DB
20 connectDB();
21
22 // Middleware
23 app.use(cors());
24
25 // JSON parsing for regular routes
26 app.use("/api/grocery", express.json(), groceryRouter);
27 app.use("/api/user", express.json(), userRouter);
28 app.use("/api/cart", express.json(), cartRouter);
29 app.use("/api/order", express.json(), orderRouter);
30 app.use("/api/merchant", express.json(), merchantRouter);
31 app.use("/api/payment", express.json(), paymentRoute);
32
33 // Raw body parsing ONLY for Paystack webhook (required for signature verification)
34 app.use(
35   "/api/paystack",
36   express.raw({ type: "application/json" }),
37   paystackRoute
38 );
39
40 // Serve static uploads
41 const __filename = fileURLToPath(import.meta.url);
42 const __dirname = path.dirname(__filename);

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

found 0 vulnerabilities

annaaoz@Annas-MacBook-Pro: ~ frontend %

Ln 47, Col 27 Spaces: 4 UTF-8 LF (JavaScript) Completions quota reached

c) I used the MongoDB database: MongoDB is a NOSQL database, thus, it is well suited for flexible and dynamic schemas as this case study. The Merchants collection contains varying data structures such as the categories, split codes, subaccounts. Since this is an ecommerce (groceries) marketplace, the products can evolve quickly including future requests and accommodations for new fields, pricing, stocks, etc. Additionally, the document model aligns with the JSON objects coming from Paystack and the frontend.

SEE SCREENSHOT BELOW

To store:

- Orders
- Grocery list
- Merchants
- Payment references
- Transaction status
- Split fees

The screenshot shows the MongoDB Atlas Data Services interface. The top navigation bar includes 'Atlas', 'GrundyLLC', 'Access Manager', 'Billing', 'All Clusters', 'Get Help', and a user profile for 'Anna'. Below the navigation is a secondary navigation bar with 'Projects' (selected), 'Data Services' (selected), 'Charts', and icons for 'Clusters', 'Services', and 'Metrics'.

The main content area shows the 'GRUNDYLLC > PROJECTS > DATABASES' section. It displays the 'GrundyLLC' database with 'OVERVIEW', 'REAL TIME', 'METRICS', and 'COLLECTIONS' tabs. The 'COLLECTIONS' tab is selected, showing 'DATABASES: 1' and 'COLLECTIONS: 4'. The collections listed are 'groceries', 'merchants', 'orders', and 'users'. A table provides detailed statistics for each collection:

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
groceries	32	6.63KB	213B	36KB	1	36KB	36KB
merchants	10	2.83KB	290B	36KB	2	72KB	36KB
orders	0	0B	0B	4KB	1	4KB	4KB
users	5	1.19KB	244B	36KB	2	72KB	36KB

Buttons for 'PREVIEW', 'New Data Explorer', 'VISUALIZE YOUR DATA', and 'REFRESH' are also present.

3) I used Ngrok for testing: This was necessary to expose local backend endpoints to Paystack for testing the webhooks and callback. These URLs were put under my API Configuration section on the paystack dashboard.

SEE SCREENSHOTS BELOW

The screenshot shows a terminal window with the following content:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
ngrok
Using ngrok for OSS? Request a community license: https://ngrok.com/r/oss
Session Status          online
Account                 annai.ozor@gmail.com (Plan: Free)
Version                3.30.0
Region                 Europe (eu)
Latency                122ms
Web Interface          http://127.0.0.1:4040
Forwarding             https://marco-crescive-vowely.ngrok-free.dev -> http://localhost:4000
Connections            ttl     opn      r1      r5      p50      p90
                        3       0       0.00    0.01    6.01    6.02
HTTP Requests
03:06:17.695 WAT POST /api/paystack/webhook/    200 OK
03:05:47.600 WAT POST /api/paystack/webhook /  404 Not Found
03:05:40.122 WAT POST /api/paystack/webhook    200 OK
(Ctrl+C to quit)

```

API Configuration - Test Mode

⚠ These keys are for testing only. Please DO NOT use them in production.

<b>Test Secret Key</b>	<input type="text" value="sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe"/> <span style="font-size: small;">[Copy]</span> <span style="font-size: small;">[QR]</span>
<small>Generate new secret key <span style="color: blue;">(i)</span></small>	
<b>IP Whitelist</b> <span style="color: blue;">(i)</span>	<input type="button" value="+ Add IP addresses"/>
<b>Test Public Key</b>	<input type="text" value="pk_test_28652f8f09044af0310d759f840e3136ea089fea"/> <span style="font-size: small;">[Copy]</span>
<b>Test Callback URL</b>	<input type="text" value="https://marco-crescive-vowely.ngrok-free.dev/payment/success"/>
<b>Test Webhook URL</b>	<input type="text" value="https://marco-crescive-vowely.ngrok-free.dev/api/paystack/web"/>

Save changes

安娜奥佐尔 — zsh — 80x24

```

"event": "charge.success",
"data": {
    "metadata": {
        "order_id": "123457"
    }
}
OK
annaozor@Annas-MacBook-Pro ~ % curl https://api.paystack.co/transaction/initialize \
-H "Authorization: Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe" \
-H "Content-Type: application/json" \
-d '{
    "email": "customer@example.com",
    "amount": 50000,
    "subaccount": "ACCT_esglysqtcmv72gb",
    "payment_method": "card"
}'
[{"status":true,"message":"Authorization URL created","data":{"authorization_url":"https://checkout.paystack.com/pug5s3sim0hr6tu","access_code":"pug5s3sim0hr6tu","reference":"g50fzxebw6"}]}%
annaozor@Annas-MacBook-Pro ~ %

```

```
[{"Fresh_Produce"}, {"split_code": "SPL_RB_PctTteoG", "dateJoined": "2025-10-11T08:31:39.627Z", "...": 0}], annaozor@Annas-MacBook-Pro ~ % curl https://marco-crescive-vowely.ngrok-free.dev/payment/success
<!DOCTYPE html>
<html lang="en">
<head>
<script type="module">import { injectIntoGlobalHook } from "@/react-refresh";
injectIntoGlobalHook(window);
window.$RefreshReq$ = () => {};
window.$RefreshSig$ = () => (type) => type;</script>
<script type="module" src="@vite/client"></script>
<meta charset="UTF-8" />
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
<!-- <link rel="icon" type="image/svg+xml" href="/vite.svg" /-->
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Grundy LLC</title>
</head>
<body>
<div id="root"></div>
<script type="module" src="/src/main.jsx?t=1760214153246"></script>
</body>
annaozor@Annas-MacBook-Pro ~ %
```

```
^C
annaozor@Annas-MacBook-Pro backend % npm install -g ngrok
added 44 packages in 16s
9 packages are looking for funding
  run `npm fund` for details
annaozor@Annas-MacBook-Pro backend % ngrok http 4000
```

Code:

```
```bash
curl -X POST https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook \
-H "Content-Type: application/json" \
-H "x-paystack-signature: faketest" \
-d '{
  "event": "charge.success",
  "data": {
    "metadata": {
      "order_id": "123456"
    }
  }
}'```
-
```

```
- annaozor@Annas-MacBook-Pro ~ % curl -X POST https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook \
-H "Content-Type: application/json" \
-H "x-paystack-signature: faketest" \
-d '{
  "event": "charge.success",
  "data": {
    "metadata": {
      "order_id": "123457"
    }
  }
}'OK
annaozor@Annas-MacBook-Pro ~ % curl
```

The screenshot shows the Postman interface with a collection named "Paystack - GrundyLLC". A new POST request is being made to the URL <https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook>. In the Headers section, two fields are present: "Content-Type" with the value "application/json" and "x-paystack-signature" with the value "<your test signature>". The response status is 200 OK with a time of 869 ms and a size of 248 B.

4) Set up .env / Environment Variables for Security, automation and flexibility during development, testing and production. I stored my PAYSTACK\_SECRET\_KEY and PAYSTACK\_PUBLIC\_KEY in my .env file.

SEE SCREENSHOT BELOW

```

GrundyLLC

.env
  X README.md
  PaymentPage.jsx
  PaystackButton.jsx
  vite.config.js
  JS paystackRoute.js
  JS p

backend > .env
Import to Postman
1  JWT_SECRET="random#secret"
2
3  PAYSTACK_SECRET_KEY="sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe"
4
5  PAYSTACK_PUBLIC_KEY="pk_test_28652f8f09044af0310d759f840e3136ea089fea"
6
7  PLATFORM_COMMISSION_PERCENT=10
8
9  GRUNDY_SUBACCOUNT_CODE=ACCT_esglysqtcmv72gb
10
11 PAYSTACK_CALLBACK_URL="https://marco-crescive-vowely.ngrok-free.dev/payment/success"
12
13 PAYSTACK_WEBHOOK_URL = "https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook"
14
15 NODE_ENV=development
16 PORT=4000
17

```

5) Create [server.js](#)

Load environment variables

```

import 'dotenv/config';
const PAYSTACK_SECRET_KEY = process.env.PAYSTACK_SECRET_KEY;

```

SEE SCREENSHOT BELOW

```

backend > JS server.js > ...
1 // server.js
2 import express from "express";
3 import cors from "cors";
4 import { connectDB } from "./config/db.js";
5 import groceryRouter from "./routes/groceryRoute.js";
6 import userRouter from "./routes/userRoute.js";
7 import cartRouter from "./routes/cartRoute.js";
8 import orderRouter from "./routes/orderRoute.js";
9 import merchantRouter from "./routes/merchantRoute.js";
10 import paymentRoute from "./routes/paymentRoute.js";
11 import webhookRoute from "./routes/paystackRoute.js"; // webhook route
12 import path from "path";
13 import { fileURLToPath } from "url";
14 import { dotenv/config };
15
16 const app = express();
17 const port = 4000;
18 |
19 // Connect to DB
20 connectDB();
21
22 // Middleware
23 app.use(cors());
24
25 // JSON parsing for regular routes
26 app.use("/api/grocery", express.json(), groceryRouter);
27 app.use("/api/user", express.json(), userRouter);
28 app.use("/api/cart", express.json(), cartRouter);
29 app.use("/api/order", express.json(), orderRouter);
30 app.use("/api/merchant", express.json(), merchantRouter);
31 app.use("/api/payment", express.json(), paymentRoute);
32
33 // Raw body parsing ONLY for Paystack webhook (required for signature verification)
34 app.use(
35   "/api/paystack",
36   express.raw({ type: "application/json" }),
37   paystackRoute
38 );
39
40 // Serve static uploads
41 const _filename = fileURLToPath(import.meta.url);
42 const _dirname = path.dirname(_filename);
43 app.use("/uploads", express.static(path.join(_dirname, "uploads")));
44
45 // Test route
46 app.get("/", (req, res) => {
47   res.send("Hello, World!");
48 });

```

## 6) Endpoints to create:

### a) Initialize Payment

```http

POST /api/payment/initialize

```

POST /api/payment/initialize - <https://api.paystack.co/transaction/initialize>

Request body:

```
{
  "email": "customer@example.com",
  "amount": 500000,
  "subaccount": "ACCT_xxxxxxxxxx",
  "payment_method": "card"
}
```

POST | https://api.paystack.co/transaction/initialize

Params Authorization Headers (12) Body Scripts Settings

Body (raw JSON)

```

1 {
2   "email": "customer@example.com",
3   "amount": 500000,
4   "subaccount": "ACCT_esglysqtcmv72gb",
5   "payment_method": "card"
6 }
7
  
```

Body Cookies (2) Headers (20) Test Results

Pretty Raw Preview JSON

Status: 200 OK Time: 711 ms Size: 1.03 KB

```

1 {"status":true,"message":"Authorization URL created","data":{"authorization_url":"https://checkout.paystack.com/xp3eqthstc2tjrk","access_code":"xp3eqthstc2tjrk","reference":"evvy6m7szj"}}
  
```

POST | https://api.paystack.co/transaction/initialize

Params Authorization Headers (12) Body Scripts Settings

Headers (10 hidden)

Key	Value
Authorization	Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe
Content-Type	application/json
Key	Value

Body Cookies (2) Headers (20) Test Results

Pretty Raw Preview JSON

Status: 200 OK Time: 711 ms Size: 1.03 KB

```

1 {"status":true,"message":"Authorization URL created","data":{"authorization_url":"https://checkout.paystack.com/xp3eqthstc2tjrk","access_code":"xp3eqthstc2tjrk","reference":"evvy6m7szj"}}
  
```

annaozor -- zsh -- 80x24

```

"event": "charge.success",
"data": {
  "metadata": {
    "order_id": "123457"
  }
}
OK
annaozor@Annas-MacBook-Pro ~ % curl https://api.paystack.co/transaction/initialize \
-H "Authorization: Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe" \
-H "Content-Type: application/json" \
-d '{
  "email": "customer@example.com",
  "amount": 500000,
  "subaccount": "ACCT_esglysqtcmv72gb",
  "payment_method": "card"
}'
{"status":true,"message":"Authorization URL created","data":{"authorization_url":"https://checkout.paystack.com/pug5s3sim0hr6tu","access_code":"pug5s3sim0hr6tu","reference":"g50fzxebw6"}}
annaozor@Annas-MacBook-Pro ~ %
  
```

```

curl https://api.paystack.co/transaction/initialize \
-H "Authorization: Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe" \
-H "Content-Type: application/json" \
-d '{
  "email": "customer@example.com",
  "amount": 500000,
  "subaccount": "ACCT_esglysqtcmv72gb",
  "payment_method": "card"
}
  
```

}

- Calls Paystack /transaction/initialize API
- Returns authorization\_url to frontend

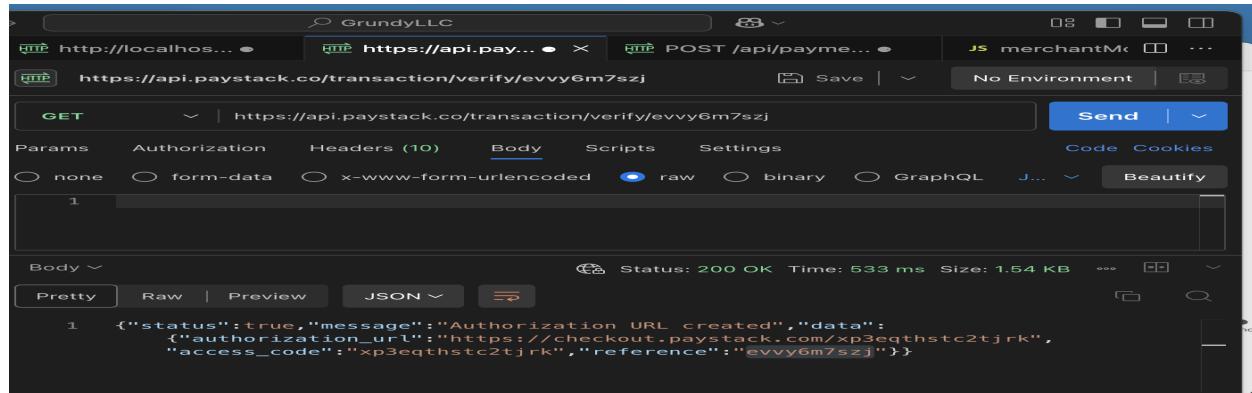
b) Verify Payment

```GET <https://api.paystack.co/transaction/verify/reference>

```

``` GET <https://api.paystack.co/transaction/verify/evvy6m7szj>

```



- Backend calls Paystack /transaction/verify/:reference
- Updates order status in DB

c) Webhook Handler

``` POST /api/paystack/webhook

```

- Receives events like charge.success for all payment methods
- Updates order/payment records in DB

HTTP <https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook>

POST <https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook>

Params Authorization Headers (10) Body Scripts Settings

Headers [8 hidden](#)

Key	Value
<input checked="" type="checkbox"/> Content-Type	application/json
<input checked="" type="checkbox"/> x-paystack-signature	<your test signature>
Key	Value

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 624 ms Size: 248 B

Pretty Raw Preview Text

1 OK

```
annaozor@Annas-MacBook-Pro ~ % curl -X POST https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook \
-H "Content-Type: application/json" \
-H "x-paystack-signature: faketest" \
-d '{
  "event": "charge.success",
  "data": {
    "metadata": {
      "order_id": "123457"
    }
  }
}'
[{"event": "charge.success", "data": {"metadata": {"order_id": "123457"}}, "id": 1}
annaozor@Annas-MacBook-Pro ~ % curl -X POST https://marco-crescive-vowely.ngrok-free.dev/api/paystack/webhook \
-H "Content-Type: application/json" \
-H "x-paystack-signature: faketest" \
-d '{
  "event": "charge.success",
  "data": {
    "metadata": {
      "order_id": "123457"
    }
  }
}'
[{"event": "charge.success", "data": {"metadata": {"order_id": "123457"}}, "id": 2}
OK
annaozor@Annas-MacBook-Pro ~ %
```

## 7) Setup Frontend Payment Flow:

- PaymentPage.jsx
  - User selects payment method: Card, Bank, USSD
  - Clicks Pay Now → Calls /api/payment/initialize

```

PaymentPage.jsx ✘ PaystackButton.jsx ✘ vite.config.js ✘ paystackRoute.js
frontend > src > pages > PaymentPage > PaymentPage.jsx > ...
1 // src/pages/PaymentPage.jsx
2 import React from "react";
3 import { useNavigate } from "react-router-dom";
4 import PayButton from "../../components/PaystackButton/PaystackButton";
5
6 const PaymentPage = () => {
7   const navigate = useNavigate();
8
9   const handleSuccess = (reference) => {
10     // Redirect to /payment/success with the transaction reference
11     navigate(`/payment/success?reference=${reference.reference}`);
12   };
13
14   const handleClose = () => {
15     alert("Payment was closed without completing.");
16   };
17
18   return (
19     <div style={{ textAlign: "center", padding: "2rem" }}>
20       <h1>Paystack Test Payment</h1>
21       <PayButton
22         email="customer@example.com"
23         amount={500000} // ₦5,000 in kobo
24         onSuccess={handleSuccess}
25         onClose={handleClose}
26       />
27     </div>
28   );
29 };
30
31 export default PaymentPage;
32

```

- b) PaystackButton (or redirect to authorization\_url)
- Opens Paystack modal for Card / USSD / Bank

```

PaymentPage.jsx ✘ PaystackButton.jsx ✘ vite.config.js ✘ paystackRoute.js
frontend > src > components > PaystackButton > PaystackButton.jsx > ...
1 // src/components/PaystackButton.jsx
2 import React from "react";
3 import { PaystackButton } from "react-paystack";
4
5 const PayButton = ({ email, amount, onSuccess, onClose }) => {
6   const config = {
7     email,
8     amount, // in kobo (₦5,000 = 500000)
9     publicKey: "pk_test_28652f8f09044af0310d759f840e3136ea089fea",
10    channels: ["card", "bank", "ussd"], // optional
11  };
12
13  return (
14    <PaystackButton
15      {...config}
16      text="Pay Now"
17      onSuccess={onSuccess}
18      onClose={onClose}
19    />
20  );
21};
22
23 export default PayButton;
24

```

- c) Callback URL

- /payment/success?reference=xxx
- React component calls /api/paystack/verify/:reference
- Shows success/failure message

```
:true,"migrate":false,"id":1485480,"createdAt":"2025-10-11T19:37:09.570Z","updatedAt":"2025-10-11T19:37:09.570Z"}]
annaozor@Annas-MacBook-Pro ~ % curl https://marco-crescive-vowely.ngrok-free.de

<!DOCTYPE html>
<html lang="en">

<head>
  <script type="module">import { injectIntoGlobalHook } from "/react-refresh";
  injectIntoGlobalHook(window);
  window.$refreshReq$ = () => {};
  window.$refreshSig$ = () => (type) => type;</script>

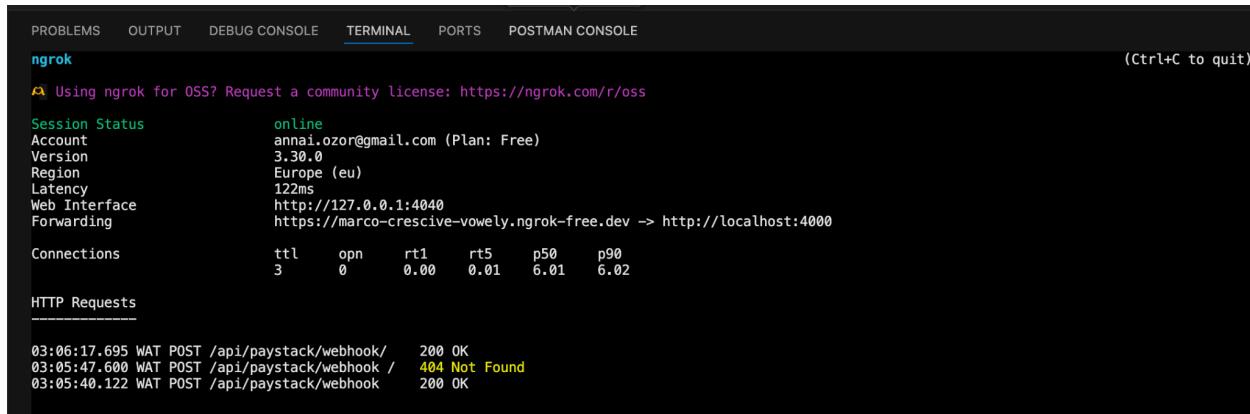
  <script type="module" src="/vite/client"></script>

  <meta charset="UTF-8" />
  <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
  <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
  <link rel="manifest" href="/site.webmanifest">
  <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
  <!-- <link rel="icon" type="image/svg+xml" href="/vite.svg" /-->
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>GrundyLLC</title>
</head>

<body>
  <div id="root"></div>
  <script type="module" src="/src/main.jsx"></script>
</body>
```

## 8) Test Flow with Ngrok

- Run backend locally: node server.js
- Expose with ngrok: ngrok http 4000
- Update Test Webhook URL in Paystack:



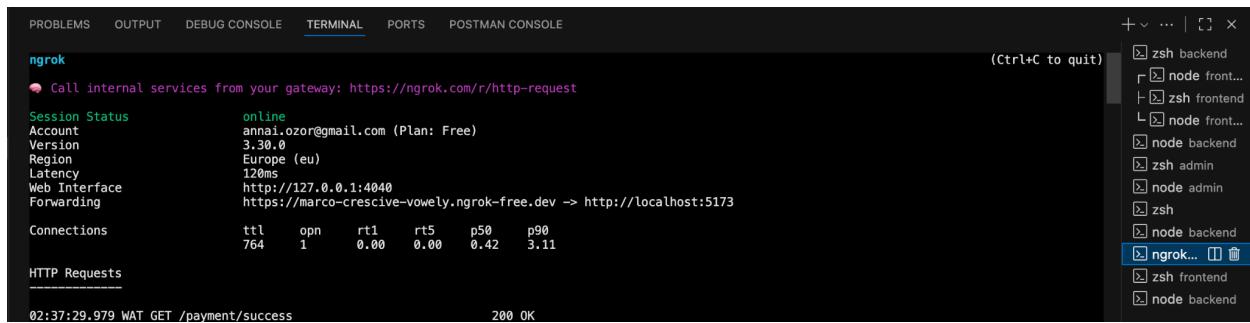
```
(Ctrl+C to quit)

ngrok
⚠️ Using ngrok for OSS? Request a community license: https://ngrok.com/r/oss

Session Status      online
Account            annai.ozor@gmail.com (Plan: Free)
Version             3.30.0
Region              Europe (eu)
Latency             122ms
Web Interface      http://127.0.0.1:4040
Forwarding          https://marco-crescive-vowely.ngrok-free.dev -> http://localhost:4000

Connections        ttl     opn      rt1     rt5     p50     p90
                  3       0       0.00    0.01   6.01    6.02

HTTP Requests
-----
03:06:17.695 WAT POST /api/paystack/webhook/    200 OK
03:05:47.600 WAT POST /api/paystack/webhook /  404 Not Found
03:05:40.122 WAT POST /api/paystack/webhook     200 OK
```



```
(Ctrl+C to quit)

ngrok
⚠️ Call internal services from your gateway: https://ngrok.com/r/http-request

Session Status      online
Account            annai.ozor@gmail.com (Plan: Free)
Version             3.30.0
Region              Europe (eu)
Latency             120ms
Web Interface      http://127.0.0.1:4040
Forwarding          https://marco-crescive-vowely.ngrok-free.dev -> http://localhost:5173

Connections        ttl     opn      rt1     rt5     p50     p90
                  764     1       0.00    0.00   0.42    3.11

HTTP Requests
-----
02:37:29.979 WAT GET /payment/success           200 OK
```

```

^C
annaozor@Annas-MacBook-Pro backend % npm install -g ngrok

added 44 packages in 16s

9 packages are looking for funding
  run `npm fund` for details
annaozor@Annas-MacBook-Pro backend % ngrok http 4000

```

## 9) Update Test Webhook URL in Paystack:

```

```perl
https://<ngrok-url>/api/paystack/webhook
```

```

## 10) Update Test Callback URL:

```

```perl
https://<ngrok-url>/payment/success
```

```

```

[{"Fresh_Produce"}, {"split_code": "SPL_RBpctTeoG", "dateJoined": "2025-10-11T08:31:39.627Z", "__v": 0}]

annaozor@Annas-MacBook-Pro ~ % curl https://marco-crescive-vowely.ngrok-free.dev/payment/success

<!DOCTYPE html>
<html lang="en">

<head>
  <script type="module">import { injectIntoGlobalHook } from '/@react-refresh';
  injectIntoGlobalHook(window);
  window.$RrefreshSig$ = () => (type) => type;</script>
  <script type="module" src="/vite/client"></script>
  <meta charset="UTF-8" />
  <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
  <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
  <link rel="manifest" href="/site.webmanifest">
  <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
  <link rel="icon" type="image/svg+xml" href="/vite.svg" /> -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>GrundyLIC</title>
</head>

<body>
  <div id="root"></div>
  <script type="module" src="/src/main.js?t=1760214153246"></script>
</body>

```

annaozor@Annas-MacBook-Pro ~ %

API Configuration - Test Mode

⚠ These keys are for testing only. Please DO NOT use them in production.

|                                         |                                                               |  |  |  |
|-----------------------------------------|---------------------------------------------------------------|--|--|--|
| <b>Test Secret Key</b>                  | sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe              |  |  |  |
| <a href="#">Generate new secret key</a> |                                                               |  |  |  |
| <b>IP Whitelist</b>                     | <a href="#">+ Add IP addresses</a>                            |  |  |  |
| <b>Test Public Key</b>                  | pk_test_28652f8f09044af0310d759f840e3136ea089fea              |  |  |  |
| <b>Test Callback URL</b>                | https://marco-crescive-vowely.ngrok-free.dev/payment/success  |  |  |  |
| <b>Test Webhook URL</b>                 | https://marco-crescive-vowely.ngrok-free.dev/api/paystack/web |  |  |  |

[Save changes](#)

## 11) Revenue Splitting (if multiple merchants)

When initializing payment, include subaccount and percentage split:

```
{
  "subaccount": "ACCT_merchantsubaccount",
  "transaction_charge": 500,
  "split": 10
}
```

Paystack automatically transfers 10% to Grundy LLC's main account.

...

**POST /subaccount** Create merchant subaccounts (for split payments)

curl https://api.paystack.co/subaccount \

```
-H "Authorization: Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe" \
-H "Content-Type: application/json" \
-d '{
```

```
  "business_name": "Grundy LLC",
  "settlement_bank": "ParalleX Bank",
  "account_number": "1181393128",
  "percentage_charge": 10
}'
```

...

```
annaizar@Annas-MacBook-Pro ~ % curl https://api.paystack.co/subaccount \
-H "Authorization: Bearer sk_test_4d02b3f0db15c5f89a4f759dc90b4794a1929abe" \
-H "Content-Type: application/json" \
-d '{
  "business_name": "Grundy LLC",
  "settlement_bank": "ParalleX Bank",
  "account_number": "1181393128",
  "percentage_charge": 10
}'

{"status":true,"message":"Subaccount created","data":{"business_name":"Grundy LLC","account_number":"1181393128","percentage_charge":10,"settlement_bank":"ParalleX Bank","currency":"NGN","bank":26,"integration":1595681,"domain":"test","account_name":"ANNA IDONGESIT OZOR","product":"collection","managed_by_integration":true,"subaccount_code":"ACCT_e8glsyqtcm72gb","is_verified":false,"settlement_schedule":"AUTO","active":true,"migrate":false,"id":1485480,"createdAt":"2025-10-11T19:37:09.570Z","updatedAt":"2025-10-11T19:37:09.570Z"}]
```

## Other APIs

| Merchant Name            | Email                       | Contact Phone     | Categories                   | Split Code               |
|--------------------------|-----------------------------|-------------------|------------------------------|--------------------------|
| Greenleaf Farms          | info@greenleaffarms.com     | +234 803 1110000  | Fruits                       | SPL_F3PKEv4f             |
| Fruitopia Distributors   | fruitopia@grundyllc.com     | +234 816 932 4421 | Fruits                       | SPL_F3PKEv4f             |
| Grainhub Stores          | grainhub@grundyllc.com      | +234 801 662 4412 | Grains and Staples           | SPL_zq9Q90W5VG           |
| Prime Proteins Ltd       | primeproteins@grundyllc.com | +234 895 208 4412 | Protein                      | SPL_b1HxS1tF             |
| Foodie Mart Distribution | foodiemart@grundyllc.com    | +234 801 662 4412 | Packaged Foods and Beverages | SPL_a019c921ea4f45cf02f2 |
| HomeEasy Essentials      | homeeasy@grundyllc.com      | +234 880 993 7851 | Household Essentials         | SPL_dedGAb0c1f           |

GrundyLLC

http://localhost:4000/api/merchant/68ea15ebc921eaf4e5fc0f02

Send | ↴

Params Authorization Headers (9) Body Scripts Settings

Query Params

| Key | Value |
|-----|-------|
| Key | Value |

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ↴

```

1   "success": true,
2   "merchant": {
3     "isActive": true,
4     "percentage_charge": 10,
5     "_id": "68ea15ebc921eaf4e5fc0f02",
6     "name": "GreenLeaf Farms",
7     "email": "greenleaffarms@grundyllc.com",
8     "contact": "+1 4234 810 555 2489",
9     "categories": [
10       "Fresh Produce"
11     ],
12     "split_code": "SPL_RBPctTteoG",
13     "dateJoined": "2025-10-11T08:31:39.627Z",
14     "__v": 0
15   }
16

```

Status: 200 OK Time: 122 ms Size: 579 B

http://localhost:4000/api/grocery/add

Save | ↴ No Environment | ↳

Send | ↴

POST http://localhost:4000/api/grocery/add

Params Authorization Headers (10) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

| Key                                             | Value                                                              |
|-------------------------------------------------|--------------------------------------------------------------------|
| <input checked="" type="checkbox"/> name        | Tomatoes (Basket)                                                  |
| <input checked="" type="checkbox"/> price       | 15000                                                              |
| <input checked="" type="checkbox"/> category    | Fresh Produce                                                      |
| <input checked="" type="checkbox"/> description | Freshly harvested red tomatoes                                     |
| <input type="checkbox"/> merchantId             | 64f8c0ab1234567890abcdef                                           |
| <input checked="" type="checkbox"/> image       | /Users/annaazor/Downloads/GrundyLLC/frontend/src/assets/item_1.png |
| Key                                             | Value                                                              |

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON ↴

```

1   {"success":true,"message":"Grocery Added","data":{"name":"Tomatoes (Basket)","price":15000,"image":"1760362450260-item_1.png","category":"Fresh Produce","description":"Freshly harvested red tomatoes","merchantId":"68ecce86c2555f892e64b1f4","_id":"68ecffd2c76823f5593ca8c5","__v":0}}

```

Status: 200 OK Time: 253 ms Size: 551 B

http://localhost:4000/api/grocery/add

Save | ↴ No Environment | ↳

Send | ↴

POST http://localhost:4000/api/grocery/add

Params Authorization Headers (10) Body Scripts Settings

Headers @ 9 hidden

| Key                                               | Value                                                                                 |
|---------------------------------------------------|---------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> Authorization | Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJpZCI6IjY4ZWNjZTg2YzI1NTVmODkyZTY0YFm... |
| Key                                               | Value                                                                                 |

HTTP http://localhost:4000/api/grocery/add

POST http://localhost:4000/api/grocery/add

Params Authorization Headers (10) Body Scripts Settings

Headers  Authorization

| Key           | Value                                                                                  |
|---------------|----------------------------------------------------------------------------------------|
| Authorization | Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJpZCI6IjY4ZWNjZTg2YzI1NTVmODkyZTY0YjFm... |
| Key           | Value                                                                                  |

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```
1 {"success":true,"message":"Grocery Added","data":{"name":"Onions (Bag)","price":20000,"image":"1760363654785-item_2.png","category":"Fresh Produce","description":"Full bag of onions sourced from local markets.","merchantId":"68ecce86c2555f892e64b1f4","_id":"68ed0486c7b823f5593ca8fd","_v":0}}
```

Status: 200 OK Time: 283 ms Size: 562 B

GrundyLLC

localhost... ● http://localhost... ● http://localhost:4000/api/user/login ● http://localhost... ● http://localhost... ● http://localhost... ● http://localhost... ● http://localhost... ● http://localhost... ●

HTTP http://localhost:4000/api/user/login

POST http://localhost:4000/api/user/login

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON

```
1 {
2   ...
3     "email": "clarab@gmail.com",
4     "password": "clarab123"
}
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```
1 {"success":true,"message":"Login successful","token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4ZWNjZTg2YzI1NTVmODkyZTY0YjFm...","user":{"id":"68ecf24e9a1504dd1bdb8a5b","name":"Clara Brown","email":"clarab@gmail.com"}}
```

Status: 200 OK Time: 266 ms Size: 585 B

<http://localhost:4000/api/order/place> for online

HTTP <http://localhost:4000/api/order/place>

POST <http://localhost:4000/api/order/place>

Params Authorization Headers (10) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL [JSON](#) [Beautify](#)

```

1
2 "userId": "68ecf24e9a1504dd1bdb8a5b",
3 "merchantId": "68ecce86c2555f892e64b1f4",
4 "items": [
5     {
6         "productId": "68e6708f9a5f921fcc08d89a",
7         "name": "Tomatoes (Basket)",
8         "quantity": 2,
9         "price": 15000
10    },
11    {
12        "productId": "68ed0476c7b823f5593ca8fa",
13        "name": "Onions (Bag)",
14        "quantity": 1,
15        "price": 20000
16    },
17 ],
18 "address": "12 Banana Street, Lagos, Nigeria",
19 "email": "clarab@gmail.com",
20 "paymentMethod": "online"
21
22

```

Body Cookies Headers (8) Test Results

[Pretty](#) [Raw](#) [Preview](#) [JSON](#) [Beautify](#)

Status: 201 Created Time: 1431 ms Size: 493 B

```

1 {"success":true,"message":"Payment link generated successfully","payment_url":"https://checkout.paystack.com/q0g66us3sm1auth",
"reference": "ORD-68ed0c5da5d4461e210a5e0e-1760365661556","orderId": "68ed0c5da5d4461e210a5e0e"}

```

HTTP <http://localhost:4000/api/order/place>

POST <http://localhost:4000/api/order/place>

Params Authorization Headers (10) Body Scripts Settings

[Headers](#) [9 hidden](#)

| Key                                               | Value                                                                                 |
|---------------------------------------------------|---------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> Authorization | Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJpZCI6IjY4ZWNNmMjRlOWExNTA0ZGQxYmRiOG... |
| Key                                               | Value                                                                                 |

Body Cookies Headers (8) Test Results

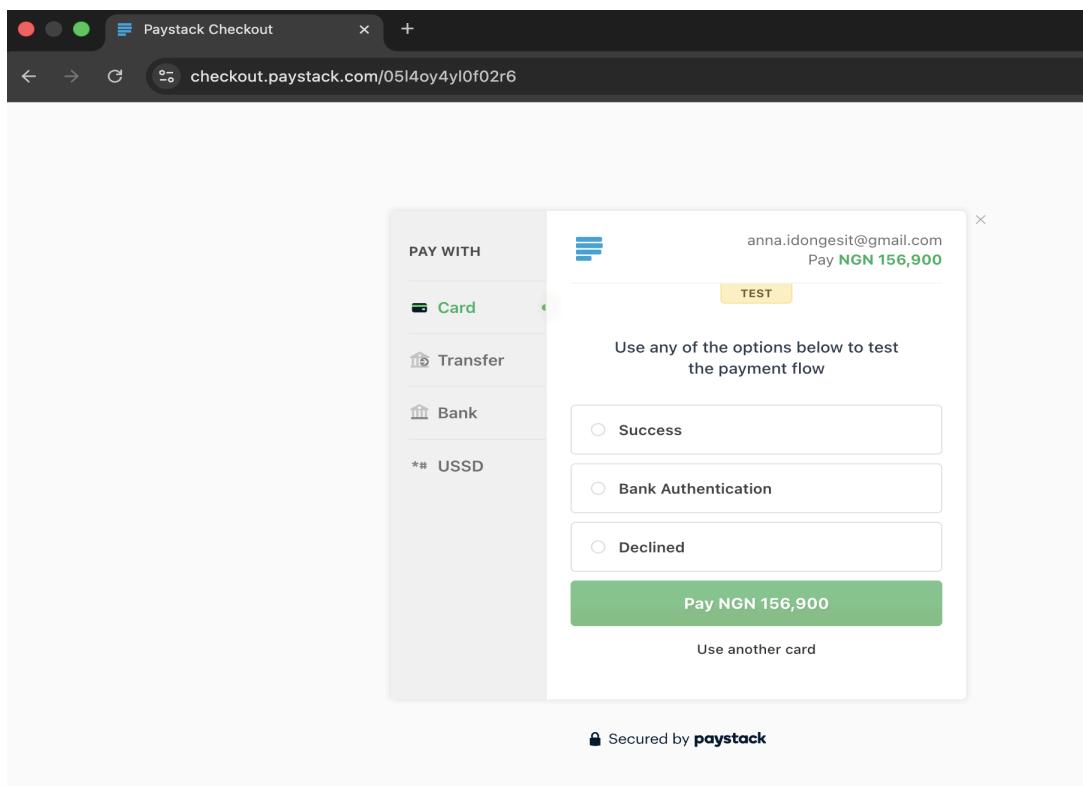
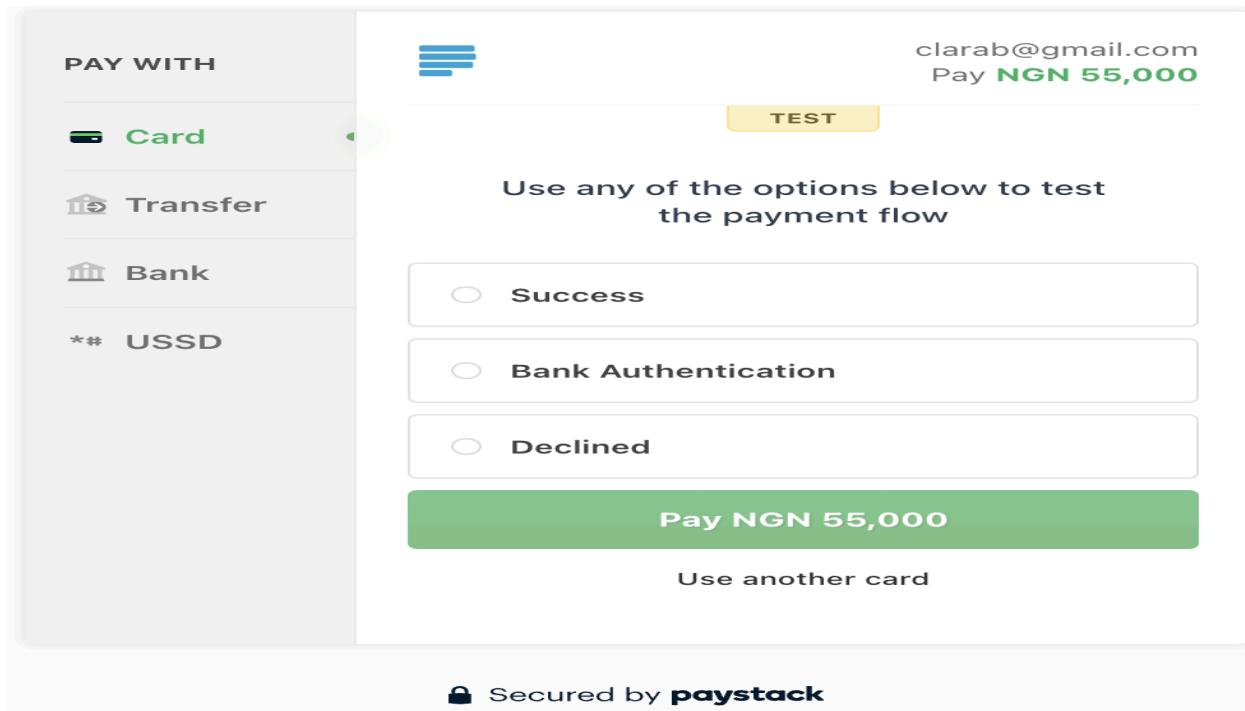
[Pretty](#) [Raw](#) [Preview](#) [JSON](#) [Beautify](#)

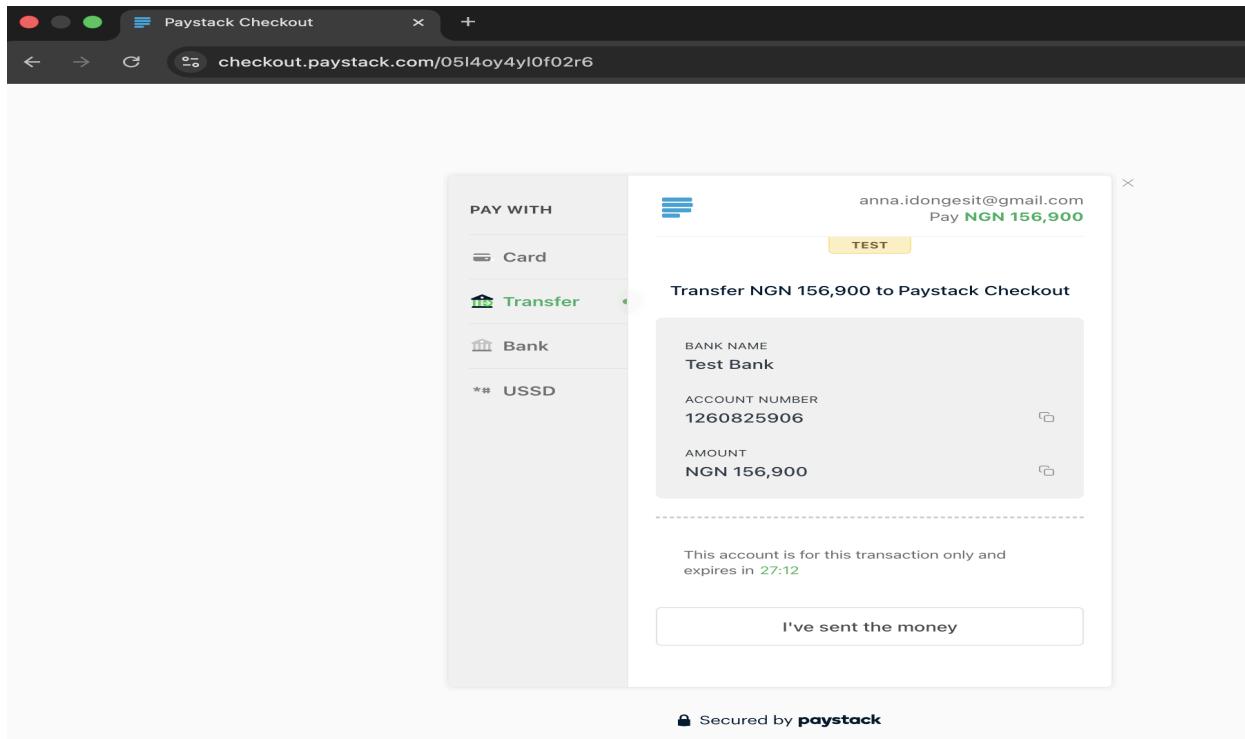
Status: 201 Created Time: 1431 ms Size: 493 B

```

1 {"success":true,"message":"Payment link generated successfully","payment_url":"https://checkout.paystack.com/q0g66us3sm1auth",
"reference": "ORD-68ed0c5da5d4461e210a5e0e-1760365661556","orderId": "68ed0c5da5d4461e210a5e0e"}

```





This is a test payment. No real money was sent to you.

clarab@gmail.com just paid  
Grundy LLC

**NGN 55,000.00**

#### Transaction Details

Business

Grundy LLC

Reference

**ORD-68ed0c5da5d4461e210a5e0e-1760365661556**

Date

**13th Oct, 2025**

Card

**VISA Ending with 4081**

[View on dashboard](#)

## GrundyLLC.orders

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 510B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

[Find](#) [Indexes](#) [Schema Anti-Patterns \(0\)](#) [Aggregation](#) [Search Indexes](#)

Generate queries from natural language in Compass ↗

[INSERT DOCUMENT](#)

Filter ↗

Type a query: { field: 'value' }

[Reset](#)

[Apply](#)

[Options ▾](#)

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('68ed0c5da5d4461e210a5e0e')
userId: ObjectId('68ecf24e9a1504dd1bdb8a5b')
merchantId: ObjectId('68ecce86c2555f892e64bf4')
items: Array (2)
amount: 55000
deliveryFee: 5000
address: "12 Banana Street, Lagos, Nigeria"
status: "Processing Groceries"
payment_method: "online"
payment_verified: false
createdAt: 2025-10-13T14:27:41.297+00:00
updatedAt: 2025-10-13T14:27:41.297+00:00
__v: 0
```

POST <http://localhost:4000/api/order/place> for bank transfer demo

POST <http://localhost:4000/api/order/place>

```

1 {
2     "success": false,
3     "message": "Internal server error"
4 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

```

Assigning virtual account for: clarab@gmail.com
Paystack customer created: CUS_rm4v1b9objaa87
✖ Paystack Virtual Account Error:
status: false,
meta: {
  message: 'Dedicated NUBAN is not available for your business',
  nextStep: 'You can send us an email at support@paystack.com to make a request for the service'
},
type: 'api_error',
code: 'feature_unavailable'
✖ Error placing order: Dedicated NUBAN is not available for your business

```

POST <http://localhost:4000/api/order/place> for terminal demo

POST <http://localhost:4000/api/order/place>

```

1 {
2     "success": false,
3     "message": "Internal server error"
4 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

```

meta: {
  nextStep: 'You can send us an email at support@paystack.com to make a request for the service'
},
type: 'api_error',
code: 'feature_unavailable'
✖ Error placing order: Dedicated NUBAN is not available for your business
✖ Paystack Terminal Payment Error: undefined
✖ Paystack Terminal Simulation Error: Request failed with status code 404
✖ Error placing order: Failed to simulate terminal payment
✖ Simulation terminal payment for terminal undefined
✖ Paystack Terminal Simulation Error: Request failed with status code 404
✖ Error placing order: Failed to simulate terminal payment

```

## PAYSTACK TERMINAL DEMO

The image shows a wireframe interface for the Paystack Terminal Demo, likely created in a tool like Figma. It consists of three main panels:

- Merchant Dashboard:** Shows an account summary for "Grundy In-Person Pa..." with a linked split code "SPL\_NPPIctfew0". A callout bubble says "Ready to expand?" with the text "Register a new payment terminal to grow your business and streamline transactions across all your sales points." and a "Register New Terminal" button.
- Terminal Details:** Displays terminal information for "MKTTELL001" (Order ID: MKTTELL0002024, Merchant: GreenLeaf Farms, Location: Open Air Market - Stall 5, Status: Active). It includes a "Process Payment" section with a payment amount of "N 100" and a "Confirm Payment" button. A "Back to Dashboard" button is at the bottom.
- Customer Terminal View:** Shows a payment interface with a total of "N100,000.00". It instructs the user to "Please insert your card into the terminal or scan the QR code to complete the payment." A "Payment Successful" message is shown with a checkmark icon.

The top of the interface has navigation tabs: DESIGN, PROTOTYPE (highlighted), and INSPECT. On the left, there's a sidebar with icons for Add, Wireframe, Diagram, Draw, Saved, and Theme. On the right, there are sharing and collaboration icons.