PAYSTACK SUBSCRIPTION AND PAYMENT MANAGEMENT SYSTEM

This project is a simple web-based system that connects with Paystack to help you manage payments, subscriptions, and plans. You can use it to test how Paystack subscriptions work and to view all related transactions easily.

📘 WHAT THE SYSTEM DOES

1. PLANS

Plans are like payment packages that users can subscribe to.

For example:

- **Basic Plan** – ₦1000 per month
- **Premium Plan** – ₦3000 per month
- **VIP Plan** – ₦5000 per month

You can create a plan and list all the existing ones on Paystack.

Each plan has a unique code (called `plan_code`) which Paystack uses to identify it.

2. SUBSCRIPTIONS

A subscription links a customer to a specific plan. Once subscribed, Paystack automatically charges the customer based on the plan's interval (e.g. monthly).

For example:

- If "John" subscribes to the "Premium Plan", Paystack will charge John ₦3000 monthly.

You can:

- Subscribe a customer to a plan.
- Disable (cancel) a customer's subscription.
- List all subscriptions in a simple table.

3. TRANSACTIONS

Transactions are all the payments that have been made by customers.

You can see:

- Who made the payment

- The amount

- The date

- The status (successful or failed)

The list supports pagination, meaning you can view them page by page.

Example:

- Page 1 → first 10 transactions

- Page 2 → next 10 transactions

4. WEBHOOKS

Paystack sends automatic notifications (called webhooks) to your server whenever

something happens — for example, when a user pays successfully or when a subscription fails.

This means you don't have to manually check; your app is automatically informed.

Example of webhook events:

- `**charge.success**` → when a payment succeeds.

- `**invoice.payment_failed**` → when a payment fails.

- `**subscription.create**` → when a user subscribes to a plan.

- `**subscription.disable**` → when a subscription is cancelled.

These events are received automatically by your server and logged in your console.

Later, they can be saved in a database if needed.

⚙️ HOW TO USE THE SYSTEM

All the demo pages are found in the public/ folder.

You can open them in your browser after starting the app (default: http://localhost:5001).

Here's what each page does:

File Name	Description
index.html	The home page
create-plan.html	Create a new plan on Paystack
list-plans.html	View all created plans
create-subscription.html	Subscribe a customer to a plan
delete-subscription.html	Cancel or disable a customer's subscription
list-subscriptions.html	View all active subscriptions
list-transactions.html	View all transactions (with pagination)
test-payment.html	Make a test payment
verify-payment.html	Verify a payment reference
success.html	Shown when payment succeeds
failed.html	Shown when payment fails
webhook-test.html	Used to test webhook events manually

Example usage:

1. Open `create-plan.html` → Fill in plan name, amount, and interval → Click "Create Plan"

2. Copy the generated Plan Code

3. Open `create-subscription.html` → Enter customer email and Plan Code → Click "Subscribe"

4. To cancel, go to `delete-subscription.html` and enter the same Plan Code

🧩 HOW WEBHOOKS WORK (IN SIMPLE TERMS)

Imagine Paystack as a messenger.

Whenever something happens (like a customer paying for a plan),

Paystack sends a "message" to your app to notify it.

Your app listens to these messages through the `/api/webhooks/paystack` route.

To make sure the message is really from Paystack and not fake,

your app checks a secret “signature” before trusting the data.

If the signature matches, it logs the event in the console like this:

✅ Webhook received: charge.success

If it doesn’t match, it prints:

❌ Invalid Paystack signature

This helps keep your payment system safe and accurate.

✅ EXAMPLE FLOW (REALISTIC SCENARIO)

1. You create a plan: “Gold Plan – ₦2000 monthly”
2. A customer subscribes to that plan.
3. Paystack charges them and sends a “charge.success” webhook.
4. Your app logs: “Charge successful”.
5. If the customer cancels, Paystack sends a “subscription.disable” webhook.
6. Your app logs: “Subscription disabled”.
7. You can open `list-subscriptions.html` to confirm the change.