$MNM Token■Gated Comic on Sui — Replit Build Guide

This guide ships a token■gated, page■flip comic dapp on **Sui** using React + Sui dApp Kit. It's tailored for the **Little Man ($MNM)** project.

Project Constants (Provided by You)

Sui Coin Type	0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ecc12
blast.fun Token Page	https://blast.fun/token/0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ec

Environment / Secrets (.env)

```
VITE_MNM_COIN_TYPE=0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ecc12bb91022b::mnm::M
VITE_MNM_MIN=100000000    # Example: 0.1 MNM if decimals=9
```

1) Coin Module (Move) — optional if $MNM already deployed

If $MNM is already live (as above), you can skip publishing a new coin. For completeness, here's a minimal coin module pattern using Sui's coin + TreasuryCap.

```
module mnm_coin::mnm {
    use sui::tx_context::{TxContext};
    use sui::coin;
    use sui::transfer;
    struct MNM has drop {}

    public entry fun init(ctx: &mut TxContext) {
        let (treasury, metadata) = coin::create_currency<MNM>(
            b"MNM",
            b"Little Man Token",
            b"$MNM for the Little Man universe",
            9,
            ctx
        );
        transfer::public_transfer(treasury, tx_context::sender(ctx));
        transfer::public_transfer(metadata, tx_context::sender(ctx));
    }

    public entry fun mint_and_send(
        cap: &mut coin::TreasuryCap<MNM>,
        amount: u64,
        recipient: address,
        ctx: &mut TxContext
    ) {
        coin::mint_and_transfer<MNM>(cap, amount, recipient, ctx);
    }
}

# Switch to desired network
```

```
sui client switch --env testnet

# Publish & initialize (only if creating a new coin)
sui client publish --gas-budget 20000000
sui client call --package <PACKAGE_ID> --module mnm --function init --gas-budget 20000000

# Your existing coin type (use this in the dapp, no publish required):
# 0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ecc12bb91022b::mnm::MNM
```

2) Replit Frontend (React + Sui dApp Kit + Flipbook)

Build a React dapp that connects a Sui wallet, checks the user's $MNM balance for the provided coin type, and unlocks a flipbook if they hold the threshold.

```
# Create & enter a React app
npx create-vite@latest mnm-comic --template react
cd mnm-comic

# Dependencies
npm i @mysten/dapp-kit @mysten/sui.js react-pageflip
npm i -D tailwindcss postcss autoprefixer

# Tailwind init
npx tailwindcss init -p

# In Replit, add Secrets (.env):
VITE_MNM_COIN_TYPE=0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ecc12bb91022b::mnm::M
VITE_MNM_MIN=100000000
```

```
// src/main.tsx
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import { SuiClientProvider, createNetworkConfig, WalletProvider } from "@mysten/dapp-kit";
import "@mysten/dapp-kit/dist/index.css";

const { networkConfig } = createNetworkConfig({
  testnet: { url: "https://fullnode.testnet.sui.io:443" },
  mainnet: { url: "https://fullnode.mainnet.sui.io:443" },
});

ReactDOM.createRoot(document.getElementById("root")!).render(
  <React.StrictMode>
    <SuiClientProvider networks={networkConfig} defaultNetwork="testnet">
      <WalletProvider autoConnect>
        <App />
      </WalletProvider>
    </SuiClientProvider>
  </React.StrictMode>
);
```

```
// src/App.tsx
import { ConnectButton, useCurrentAccount, useSuiClientQuery } from "@mysten/dapp-kit";
import HTMLFlipBook from "react-pageflip";
```

```
import { useMemo } from "react";

const COIN_TYPE = import.meta.env.VITE_MNM_COIN_TYPE as string;
const GATE_MIN = Number(import.meta.env.VITE_MNM_MIN ?? "100000000");

export default function App() {
  const account = useCurrentAccount();
  const { data: balance } = useSuiClientQuery(
    "getBalance",
    account ? { owner: account.address, coinType: COIN_TYPE } : undefined,
    { enabled: !!account }
  );

  const unlocked = useMemo(() => {
    if (!balance) return false;
    return BigInt(balance.totalBalance ?? 0) >= BigInt(GATE_MIN);
  }, [balance]);

  return (
    <div className="min-h-screen bg-black text-white flex flex-col items-center p-6">
      <div className="w-full max-w-5xl flex justify-between items-center">
        <h1 className="text-2xl font-semibold">$MNM Comic</h1>
        <ConnectButton />
      </div>

      {!account && <p className="mt-12 opacity-80">Connect a Sui wallet to continue.</p>}

      {account && !unlocked && (
        <div className="mt-10 text-center space-y-4">
          <p>You need at least {(GATE_MIN/1e9).toFixed(2)} $MNM to unlock the comic.</p>
          <p className="opacity-70 text-sm">
            Buy or acquire $MNM here:
            https://blast.fun/token/0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ecc1
          </p>
        </div>
      )}

      {account && unlocked && (
        <div className="mt-8 w-full max-w-4xl">
          <HTMLFlipBook width={800} height={600} className="mx-auto rounded-2xl overflow-hidd
            <img src="/pages/01.jpg" alt="Page 1" />
            <img src="/pages/02.jpg" alt="Page 2" />
            <img src="/pages/03.jpg" alt="Page 3" />
            <img src="/pages/04.jpg" alt="Page 4" />
          </HTMLFlipBook>
        </div>
      )}
    </div>
  );
}
```

Assets & Performance

- Place comic pages under /public/pages (01.jpg, 02.jpg, …).

- Add: <link rel="preload" as="image" href="/pages/01.jpg"> in index.html.
- Start: npm run dev → test in Replit preview. Switch defaultNetwork to "mainnet" when read

3) Gating Logic (How It Works)

1. User connects a Sui wallet via dApp Kit.
2. The dapp queries getBalance(owner=wallet, coinType=$MNM).
3. If balance ≥ VITE_MNM_MIN (base units), the flipbook renders.
4. Otherwise, show CTA to acquire at blast.fun:
   https://blast.fun/token/0xefde5ddb743bd93e68a75e410e985980457b5e8837c7f4afa36ecc12bb91022b

4) Fast Follows (Optional Enhancements)

- Admin mint panel (owner-only) calling mint_and_send.
- CDN (Cloudflare R2) for faster image delivery.
- LocalStorage: remember last-read page; add “Continue”.
- zkLogin / gas sponsorship to cut friction.

Prepared for DA / Full Digital — $MNM (Little Man Token)