**Project Handover:**

* **Project Name:** InstantbillsPay/AMAPGS
* **Client/Organization:** UBA/OJAPAY
* **Current Frontend Developer:** David Ogbaudu
* **Handover to:** Mr. Kennedy Egwuda (CTO)
* **Date of Handover:** 01 March 2024

**1. Introduction**

This document gives an overview of the current state of the InstantbillsPay app, its dependencies, file directory, and other important information.

**2. Project Overview**

This project is a modern, user-friendly web application developed using the React JavaScript library. The application is designed to be highly responsive and scalable by breaking each feature into stand-alone components that are reusable. The goal is to redesign the current InstantbillsPay web app with all the advanced and effective web technology available.

**3. Environment Setup**

Setting up the development environment is crucial to ensure a smooth and efficient development process. Below is a brief guide on how to set up the environment for the Ojapay IBP React Web Application.

**Prerequisites:**

Before starting the setup process, make sure you have the following installed on your machine:

* Node.js: The JavaScript runtime.
* npm: The Node.js package manager.
* Git: Version control system.

**Steps:**

1. **Clone the Repository:**

git clone [repository-url]-url]

1. **Navigate to Project Directory:**

cd [project-directory]-directory]

1. **Install Dependencies:**

npm installnpm install

1. **Start Development Server:**

npm start

This command will start the development server, and you can view the application by navigating to http://localhost:3000 in your web browser.

1. **Build for Production:**

npm run build

**4. Dependencies**

**"@testing-library/jest-dom": "^5.17.0",**

**"@testing-library/react": "^13.4.0",**

**"@testing-library/user-event": "^13.5.0",**

**"@types/jest": "^27.5.2",**

**"@types/node": "^16.18.61",**

**"@types/react": "^18.2.37",**

**"@types/react-dom": "^18.2.15",**

**"axios": "^1.6.7",**

**"react": "^18.2.0",**

**"react-dom": "^18.2.0",**

**"react-google-charts": "^4.0.1",**

**"react-icons": "^4.12.0",**

**"react-router-dom": "^6.19.0",**

**"react-scripts": "5.0.1",**

**"typescript": "^4.9.5",**

**"web-vitals": "^2.1.4"**

**5. Folder Structure**

**Src:** Contain all the important files running on the web app

* **Admin:** contains all the files and folders for the Admin
* Assets: Contains images, svgs, pngs, and other assets used in building the admin
* Builders: contain reusable components for building rows and data content

- BuildMerchants

- BuildTransactionLog

* Components: Holds each functionality tab

- Dashboard: The welcome dashboard users see on sign-in after activation

- Login: The login portal where users can sign in with either email or phone and a password

- Merchant: contains information particular to a merchant

- Transactions: All transactions history is viewable here

* Styles: contains styling
* **Merchant:** contains all the files and folders for the merchant and its associated builders, functionalities and components.
* **Assets:** Contains images, svgs, pngs, and other assets used in building the admin
* **Builders:** contain reusable components for building rows and data content

- **AddSplitMembers :** a form modal for allocating a split percentage

- **AddStaff:** a form modal for adding a new staff privilege

- **BuildAddCustomer:** a form modal for adding a new customer

- **BuildCustomer2Subscription:** a component displaying all customers' subscription information

- **BuildCustomer2Transaction:** a component for displaying a subscription plan’s information

- **BuildCustomers:** displays all the customers for a merchant

- **BuildDisputes:** display all pending and resolved disputes

- **BuildEditCustomer:** a modal for editing customer information

- **BuildLineGraph:** builds the transaction graph visible on the dashboard

- **BuildProduct:** builds each store’s product

- **BuildShop:** builds each merchant shop for display

- **BuildTransaction:** builds the transaction history with data

- **CreateNewPlanModal:** a modal for creating a new payment plan

- **CreatePaymentRequestModal:** a modal for requesting payment

- **CreatePaymentSplit:** a modal for creating/scheduling a payment split between partners

- **CreateRole:** a modal for creating user-defined roles

- **CreateShop:** A modal for creating a new shop

- **CustomersInfo:** displays all the info of a customer

- **EditRole:** a modal for editing the privileges or information of an existing role

- **EditSplitMember:** a modal for editing the details of split partners

**- EditSubscriptionPlan:** a modal for editing the details of a subscription plan

**- RequestPayoutModal:** a modal for requesting payment

**- SuccessModal:** a modal that displays success messages or prompt

**- UnvarifiedWelcomeOverlay:** an overlay that stop users from accessing the contents of the dashboard

**- WarningModal:** a modal for displaying a warning or error message

* **Components:**

**- AccessControl**

**- ApiWebHook**

**- AuditLog**

**- BusinessVerification2**

**- CreateProduct**

**- Customers**

**- Customers2**

**- Dashboard**

**- Disputes**

**- HandleSubmit**

**- MyShop**

**- PaymentRequest**

**- PaymentRequestDetails**

**- PaymentSplit**

**- PaymentSplit2**

**- Payouts**

**- Products**

**- Profile**

**- Subscribers**

- SubscriptionDetails

- Subscriptions

- Transactions

* Styles: contains stylingDocument the build process and deployment procedures. Include any scripts or commands necessary for these tasks.

**CONCLUSION**

All documents, files, and folders relevant to the project has been committed to the Ojapay remote repository. Should there be need for any additional information, kindly reach out to me.

**CONTACT**

**Email:** [Ogbaudu.dc@gmail.com](mailto:Ogbaudu.dc@gmail.com)

**Tel:** 08052714449