Post-purchase Messaging

This is an App for Shopify and allows the merchant to manage the post-purchase message in their Shopify store. It works in two ways, the first is in the store's admin for the merchant, and the second is in the client's side, specifically in the store's checkout process.

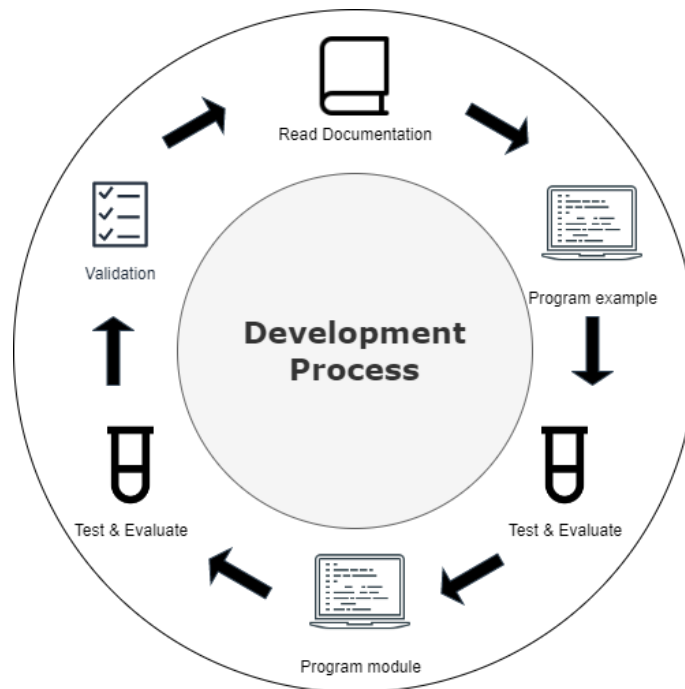**Table of Contents**

Introduction

This document aims to explain not only the structure of the application and its operation but also its development process, in order to understand the reason for the technologies used, logic, and result. The next section will explain the development process, decisions, and challenges.

Development process

Because there was no experience in this type of development, it was decided to use JavaScript as the base language, since there was some knowledge of this language; In addition to JavaScript, technologies provided by Shopify such as Polaris, Checkout Extension, Post-purchase Extension and more were used. Below is a list of these technologies:

Area	Technologies
Base language	JavaScript
Shopify store's admin	Polaris, App-Bridge-React, App-Bridge-React, React
API layer	React, React-Query, Express, COORS, JSON web token
Backend	SQLite3, Node JS
IDE	Visual Studio Code

The development process cycle was as follows:

The stages of development:

Critical decisions and issues

- First, I decided to develop an app instead of a StoreFront, because at the time I found more information in that direction.
- As database use SQLite by recommendation of the requirement.
- To display the message the following options were evaluated:
  - Script Tags, this option was the first one, but at that moment, I didn't understand much about it and read something about this option will be obsolete in 2024, For that reason, I thought it was not the best option to invest time in it.
  - Metafields, this is similar to script tags and for that reason, it was discarded.
  - Checkout extension, this option was the most logical because, according to the documentation, Shopify is directing all the customization of the checkout process towards this technology, but it was discarded because a beta version was used and its limitation is the customization of the order status page.
  - Post-Purchase order, this option was chosen, and although it is not exactly what is requested in the requirement, it is quite close.

Installation

The first thing to clarify is that the instructions will be based on the fact that the solution is in development mode and not deployed, therefore the instructions will be detailed, with several steps and clarifications.

This are the commands to use:

- GitHub: https://github.com/ademirnyk33/st-messages-manager
- Create App: npm install
- AppServer: npm install cors jsonwebtoken uuid
- To set your store: npm run dev -- --reset
- To run the server: npm run dev

To test the project, follow the instructions in the next video:

- https://www.youtube.com/watch?v=oigk5s3mlt0

Usage

To know how to use the App, you can see the video:

- https://www.youtube.com/watch?v=WVeRK2wWW3A