# Microapps front-end challenge

Avisek Chakraboty

7th April, 2016

|  |  |
| --- | --- |
| **Environment**  **Set up** | To run the files, copy full codebase and run 'index.html' from Localhost.  Payon API calls are using PHP code, that needs Apache & PHP. CURL must be enabled. (Xampp for windows) |
| **Technology**  **Used** | * Front end framework: Materialize CSS * Scripting Language: ES6 * CSS Preprocessor: SCSS * Server Side: PHP   (couldn't get a chance to work with Node.JS, will learn soon) |
| **Technical**  **Requirements** | * Entirely Single-page application * Used Pay-on API test server * Store data in localstorage * Used AJAX calls |
| **Screens** | **i. Donate**    **ii. Payment**    **iii. Process Payment**    **iv. Status** |
| **Development**  **process** | 1. For first time users, opening donation form to take inputs of donation-amount & preferred currency. 2. Once user submits, disabling Submit-button to block multiple submission & validating as per business requirements. 3. Firing AJAX call to server-side PHP page with form data to Prepare-checkout Payon API call. 4. Loading Payment form using proper Payon Script 5. Once payment is done and redirected back, detecting it, showing proper screen and showing response to the user. Checking status-code & on success- keeping payment details in Locastorage.   Also removing parameters from URL-string (without page refresh). So that accidental page-refresh will not bring the same scenario.   1. Showing Thank-You screen to user after a few seconds, along with last-successful-payment details.   Also showing the remaining-time to be able to donate next time- which refresh automatically & once it reaches 1 hour, enabling the button so, user can donate again. |
| **Improvements** | There are few improvement points, which I wish to apply in next version -   1. There are lot of status codes- we need to do a vast checking using Regex, to perfectly detect a successful transaction. 2. Once Payment is done & redirected back, am detecting that and retrieving payment status. It's better to detect accidental page-refresh or exit. 3. Validate the localstorage data, before manipulation in every step. SO that, we can detect a cache-clear in the middle and act accordingly. 4. In server-side code (PHP), set up a config file to properly keep Test/Live data, User id etc. |