High Level requirements

# Abstract

This document will cover all high-level requirements in a table. The purpose of this table is to be a reference to the developer. It is meant to be used together with the use cases and use case diagram.

|  |  |
| --- | --- |
| Requirement | Description |
| Setup MongoDB database | In order to prove that mongoDB is a suitable database for this project, it is the first part to be developed. |
| Implements node.js template “Node Login” | The Node Login template will form the infrastructure for the website. It is therefore of utmost importance to clean the code, and strip all unnecessary or deprecated code from the code base.  The result should be to have a clear understanding of the mechanisms behind the code. |
| Find suitable HTML 5 template to integrate with the node.js backend | It was decided to opt for a template instead of writing the front-end from scratch.  This is due to the project being largely a proof-of-concept, and process should be made quickly. |
| Bitcoin Blockchain API integration | A suitable node.js implementation of the bitcoin protocol layer must be found.  Most likely, this will be the Blockchain “bitcore” library on GitHub. |
| Bitcoin Price API integration | The backend must fetch the price of 1 BTC every few seconds to ensure that users can’t abuse price fluctuations. |
| Indicate guest/logged-in user with a top bar | Permanently top fixed bar to greet visitors and show a link to the log-in form.  It must be responsive and have different content depending on:   * Is the user logged in? * Is the user an admin?   The website should indicate these things. |
| Securely store and transfer sensitive information | Security guide lines must be followed to ensure the application does not have obvious security leaks.  When transferring information between the backend and database, the password must be hashed or in other ways obfuscated.  Sensitive information may not be displayed or accessed on or from the front-end. |
| Display and modify orders placed by users | The website shall allow users to place orders on a dedicated page. The form must be able to store orders to the account logged in. If the user is not logged in, placing an order must not be allowed.  Admins must have the functionality to accept orders that were paid for. Using checkboxes or radioboxes and a submit button should suffice. |
| Design/use attractive website design | The website developed is a consumer product and must look and feel professional to gain credibility from users.  Dark grey and brown colours should be favoured, with an appropriate background, relating Bitcoin and linking the website to London. |
| Simplistic UI with social media indicators | Navigation must be intuitive.  For the prototype a 3-page site will perform all required tasks without problems.  To use social media icons a third-party library must be used. |
| HTML5 Login/Sign-up form with email retrieval | Login and signup forms will ideally be developed using HTML 5 forms. This means that error checking is built-in and compatible with many platforms and browsers. |
| Google Maps API integration | A google map must be included on the purchase page of the website.  It must highlight the vendors on the map. Preferably, it would display the distance to the user’s locations, but for the prototype the following features are necessary:   * Responsive Map. * Search box for user to locate themselves using Post Code or Address name. * Icons placed at vendors locations * Show information when an icon is clicked. |
| Host website publicly | The final step is to host all the source files on a public server. A free server may be used for demonstration purposes. |