Documentation Dump

**SSL**

* RapidSSL howto: <https://knowledge.rapidssl.com/support/ssl-certificate-support/index?page=content&actp=CROSSLINK&id=SO6252>
* Public key, private key are generated during SSL purchase.
* CA bundle crt for rapidssl: <https://knowledge.rapidssl.com/support/ssl-certificate-support/index?page=content&actp=CROSSLINK&id=AR1548>
* Fixed: apache config file, ws\_server ssl settings, django settings, views functions
* Customize everytime: SSL keys, open port 443
* HTTP connections disabled by commenting out Listen 80 in /etc/apache2/ports.conf

**Web Sockets**

* Custom twistd bin: -y -n ws\_server and kill pid
* Port 1025 (ws:// and https://)
* ws used for javascript listener
* <https://~:1025> used to update to twistd server
* customize everytime: open port 1025
* twistd.log, twisd.pid

**Monaca:**

* Barcode scanner: <http://docs.monaca.mobi/cur/en/reference/phonegap_plugin/barcode_scanner/>

**Server stuff:**

* Apache /etc/apache2, config: /etc/apache2/sites-enabled/chai.conf (mirrored from chai.conf in django folder), project root: ~/chai\_cloud\_deploy/,

**Future improvements:**

* Separate files for remaining hardcoded CSS for clarity
* Reverse proxy to merge 1025 with 443 for network simplicity
* Use something cleaner than table for store app
* Merge some repeated code like complete button event
* Separate page for shopping cart, or scrollable dialog. Now only supports 10 orders
* App menu return from click, item is still clicked.
* Select multiple options per item
* Html ID tags global variables instead of hard coded

**App:**

* Places of static code
* Shopping cart json object
* Cart ID conventions: cart-<name>
* Options ID conventions: op-<name>
* User/pass only used for app account, no risk. Paypal handles transaction info and order confirmation.
* Data passed in hidden option div “options-content”

**Dev things**:

* All changes to django files require sudo service apache2 restart to take effect

**QR:**

* Javascript QR library: <http://davidshimjs.github.io/qrcodejs/>
* Api/views.py qr function, takes different input and outputs QR code accordingly

