Project Tree v1.0
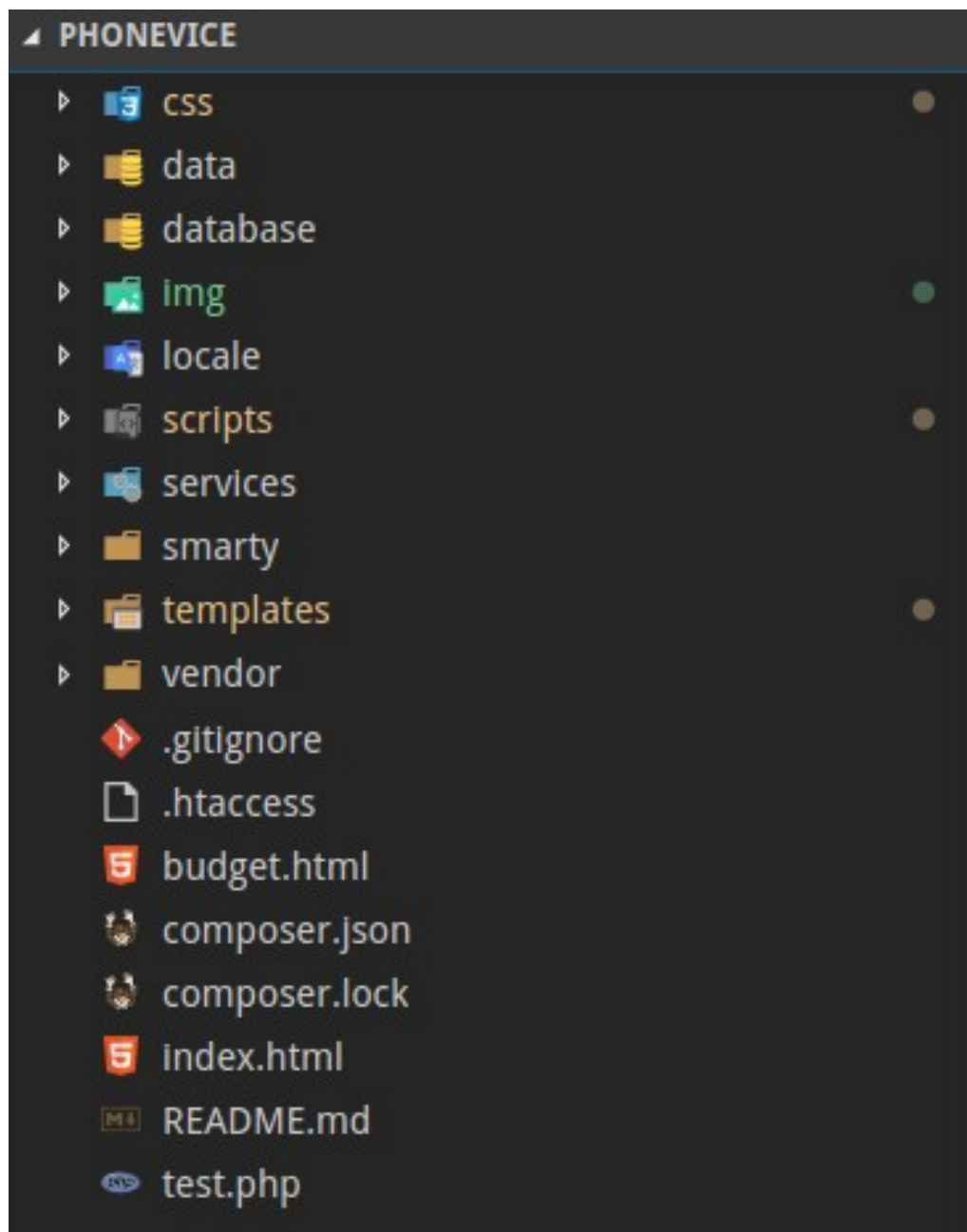

This is the main project tree. There may be some minor changes in the future.

Folder explanations

css/   Css folder, nothing to say about it. It will contain boostrap.min.css , style.css, and any other css libraries that are essential (eg fontawesome) but in **minified** form.
style.css will contain all custom css used for the project, which we want to keep at a minimum since bootstrap already provides many ready-to-play classes. For example if we want a div to have its

content center-aligned, we will assign the class **text-center** to it
instead of writing **text-align: center;** in the style.css

data/   Any data/documentation/scripts/images you want to store for
your own use, use this folder. This folder **will not be uploaded** to
the production server.

database/   Contains the smartphone database in a json file.

img/   Contains all the images what will be used in the web
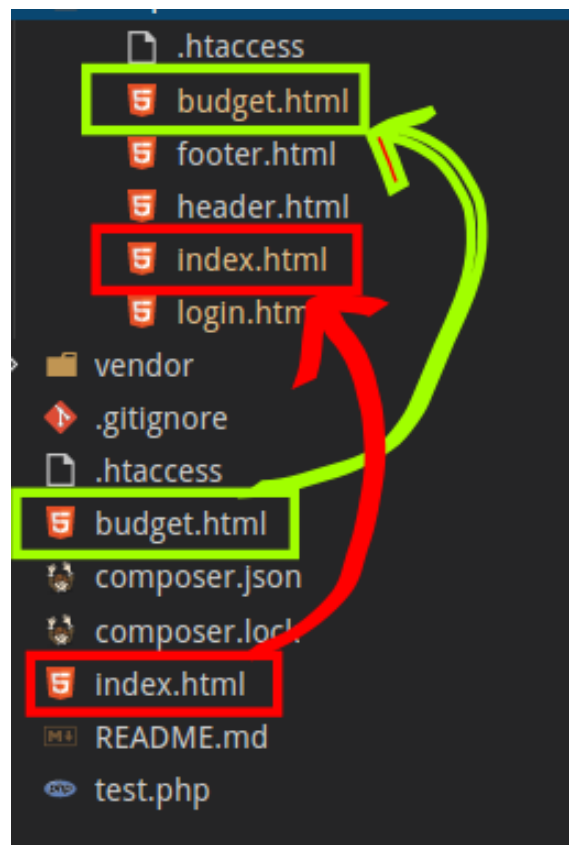application.

locale/   Smarty folder, used for translating content in
different languages

scripts/   To be renamed into **js/** , the folder that contains all
the javascript files.
custom.js will be for custom javascript/jquery, and other that that
there will be jquery.min.js and bootstrap.bundle.min.js
Any other js libraries you need feel free to add it to this folder,
in minified form.

services/   Folder that contains the backend code of the
application

smarty/   Folder that contains all configuration for smarty, a
template language for php that connects backend with frontend. That
way we do not have to write php code in html files.

templates/   Important, this is where the frontend (HTML) code
lies.

On the above example, the bottom budget.html and index.html files are actually php files with html extension for security (we do not want to easily show that we use php for backend). If you open those files you will see on the end $smarty - > display...
For example, bottom index.html displays templates/index.html file etc. So we can use the same procedure to display any page on the frontend.

Vendor/ Folder that contains all mongodb/composer configuration files

composer.json and composer.lock are composer files used for the configuration of mongodb dependency.