Theme and Plugin Folder Structure

The Juicebox WordPress Base folder structure is similar to putting WordPress in its own subdirectory but with some improvements.

- To avoid exposing sensitive files in the webroot, what's required has been moved into a `www/` directory including the `wp/` source and the `wp-content/` source.
- The `wp-content/` folder has been named `app` to better reflect its contents and contains application code and not just "static content". This also matches up with other frameworks such as Symfony and Rails.
- The `wp-config.php` remains in the `www/` because it's required by WordPress, but it only acts as a loader. The actual configuration files have been moved to `config/` for better separation.
- The `vendor/` folder is where the Composer-managed dependencies are installed to.
- The `wp/` folder is where WordPress core lives. It's also managed by Composer but can't be put under vendor due to WordPress limitations.

File Folder Structure in Detail

```
|- config/ ________________________________# WordPress configuration files
|  |- environments/ ______________________ # Environment specific configs
|     |- development.php _________________ # Development config
|     |- production.php __________________ # Production config
|     |- staging.php _____________________ # Staging config
|  |- application.php _____________________ # Primary WP config file (wp-
config.php equivalent)
|- node_modules/ _________________________ # npm modules
|- src/ __________________________________ # Installation scripts
|- www/ __________________________________ # Web root (document root on your
webserver)
|  |- app/ _______________________________ # wp-content equivalent
|     |- languages/ ______________________ # Translations
|     |- mu-plugins/ _____________________ # Must use plugins
|     |- plugins/ ________________________ # Plugins
|     |- themes/ _________________________ # Themes
|     |- uploads/ ________________________ # Uploads
|  |- wp/ ________________________________ # WordPress Core
|  |- index.php __________________________ # WordPress view bootstrapper
|  |- wp-config.php ______________________ # Required by WP (never edit)
|- vendor/ _______________________________ # Composer dependencies
|- .deployignore _________________________ # Files to ignore during deployment
|- .editorconfig _________________________ # Code Editor config settings
|- .env.live _____________________________ # Environment file for production
|- .env.staging __________________________ # Environment file for staging
|- .env.template _________________________ # Environment file for template
|- .eslintrc _____________________________ #
|- .gitattributes ________________________ #
|- .gitignore ____________________________ # Ignore committing files to
repository
|- .prettierignore _______________________ #
|- .prettierrc ___________________________ #
|- browserslist __________________________ #
|- composer.json _________________________ # Manage versions of WordPress,
plugins & dependencies
|- package.json __________________________ # npm package file
|- phpcs.xml _____________________________ #
|- webpack.mix.js ________________________ # Webpack config settings
|- wp-cli.yml ____________________________ #
```

Theme Structure

The default theme folder structure is described separately below to improve readability:

```
|- acf-json/ _____________________________ # This stores ACF field configurations automatically
|- dist/ _________________________________ # Compiled CSS/JS - don't edit file in here
|- fonts/ ________________________________ # Custom/hosted fonts
|- img/ __________________________________ # Theme images
|- js/ ___________________________________ # Source JS files
|  |- plugins/ ___________________________ # External JS plugins that are not installed via NPM
|  |- src/ _______________________________ # Custom JS scripts
|- scss/ _________________________________ # Source Sass files
|  |- abstracts/ _________________________ # Functions, mixins, variables
|  |- base/ ______________________________ # CSS at the top of the cascade
|  |- pages/ _____________________________ # Page specific overrides
|  |- vendor/ ____________________________ # 3rd party library CSS
|  |- vendor-overrides/ __________________ # Overrides of 3rd party libraries CSS
|- src/
|  |- JuiceBox/
|     |- Components/ _____________________ # Each folder contains a specific Component that can be used or re-used around the site. It contains all the required html, scss, js to function
      |- Config/ _________________________ # Configuration based hooks - no need to touch
      |- Core/ ___________________________ # See below
         |- ACFJson.php __________________ # Handles automatically putting module ACF fields into fields.json within the module folder.
         |- Admin.php ____________________ # Admin based hooks
         |- GravityForms.php _____________ # Gravity forms customisations
         |- Helpers.php __________________ # Helper functions
         |- Image.php ____________________ # Custom Image class, extends Timber
         |- Menu.php _____________________ # Custom Menu class, extends Timber
         |- MenuItem.php _________________ # Custom MenuItem class, extends Timber
         |- Module.php ___________________ # Base Module class, all Modules inherit
         |- Post.php _____________________ # Base Post class, extends Timber
         |- Site.php _____________________ # Site class, custom actions/hooks
         |- Term.php _____________________ # Term class, extends Timber
       |- Modules/ _______________________ # Each folder contains a specific Module that is re-used around the site. It contains all the required html, scss, js to function
       |- PostType/ ______________________ # Add custom post types here, they are registered automatically
            └── (add custom taxonomies here, they are registred automatically)
```

If you're working on older Juicebox websites created before 2020, you may notice a parent-child theme structure, with a parent theme named `juicy` located in the themes folder; the Juicy parent theme was converted to a plugin in 2020.

Although the Juicy theme includes a few hooks and functions, it doesn't have any template or style code. It's important to note that making any changes to the Juicy theme won't be committed to the site repository, so you should work from the child theme instead.