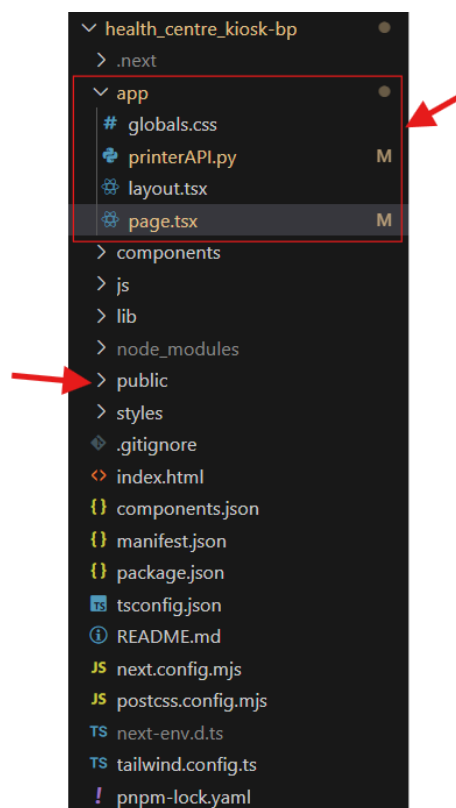


Technical Documentation

Files and Folders:

The source files for the kiosk application can be found in the GitHub repository at the link https://github.com/KingFlameThe1/health_centre_kiosk-bp. You will need to download the repository to run the application. Specific details regarding hosting the application will be given in the “testing / hosting” section. In further development of the application, you will mostly be interacting with the files in the “app” and “public” folders. These folders are shown below.



The app’s content and functionality are handled mostly by page.tsx, while globals.css is responsible for the design / visuals. The layout.tsx files ties both together and holds some meta data normally found at the top of an HTML file. The printerAPI.py file is a flask script meant to handle interaction between the kiosk app and the thermal printer.

The “public” folder houses the images used in app and the “manifest.json” and “sw.js” (service worker) files. The manifest.json and service worker are necessary for caching so the app can run offline. (**NOTE**: as of writing the app does not have offline printing capabilities).

Note: do not be confused by the other manifest.json file, the kiosk.js file or the index.html file. Those were from earlier versions of the kiosk app.

Testing / Hosting:

Note: Set IP values at the top of the page.tsx and printerAPI.py files before attempting to host them

The kiosk

The kiosk application was hosted locally via Node.js during the testing phase. To run the program, simply navigate to the main directory (the one housing the “app” folder and other directories) and use the command

```
npm run dev
```

or the commands

```
npm run build
```

then

```
npm start
```

If it your first time hosting the application, you may need to install a few dependencies for the project. Do this using the command

```
Npm install next react react-dom --legacy-peer-deps
```

After starting the application, two URLs should appear on your terminal. You may select either localhost or the network address. Selecting either will compile the application so it can be accessed. The system should run normally from here.

The Printer API

This is a python flask script meant to act as an intermediary between the kiosk and the printer. During testing, both the kiosk and the API were both hosted on the same device (more details in limitations sub-section). This on can be executed like a regular python file.

Limitations / Constraints

1. Printing tickets from the kiosk was only ever tested when the kiosk and printerAPI.py were hosted on the same device. This was because the printer, the hosts for the kiosk and printerAPI.py all need to be on the same subnet.
2. Printing was not tested when the kiosk system was accessed via tablet but hosted on another machine (the same machine that would have hosted printerAPI.py) because of similar network constraints.

Notes for Future development:

As of writing, the code responsible for displaying the webpages starts at line 170. The first few lines house information that remain constant across all pages. Each page is separated into sections identified by the variable “currentStep”. This variable is updated to change the contents displayed in the browser. All functions called when a user performs an action are defined above line 170.

Some sections of code are no longer in use due to evolving requests from the client (UWI Health Centre). This includes the ``currentStep == “affiliation”` and ``currentStep == “information”` sections and related functions, variables and constants. The functions and constant not in use should be labelled as such.