Programmer Manual

IUS Nursing Skills Checklist Website

**1. Vision statement**

The IUS Nursing Skills Checklist website is a page that students and faculty/staff use to track student’s task along the time in the program. Previously, students were required to carry a 10 page document with 160+ tasks that they must complete over either 4 semesters or as long as it takes. This means, they have to keep track of one physical document for multiple years. Upon successful return of the document, the Nursing staff then enters each students data into a excel document. This is highly inefficient and cumbersome for both administrators and students alike.

.

**2. Introduction**  
The IUS Nursing Skills Checklist website is a lightweight and user friendly page to help facilitate a someone cumbersome process throughout a students time in the program. When tracking something over multiple years of time, it is important to have a solid foundation and database for work that has been accomplished. This is where the website excels versus the existing process of tracking a physical document. Progress is always stored and retrievable indefinitely.

**3. Component Overview (one for each)**

**a) User Interface (Students)**

The user interface is self-explanatory with a “Usage” tab depicting the websites appropriate usage as intended by the administrator sponsors as well as the developer.

**b) User Interface (Administrators)**

Administrators have access to a separate view than the Students. This allows them to access the data for every student while never actually being able to alter the data.

**c) Data Access Layer**

Only developers can alter the data access layer. The current administrators do not   
 need access to the data access layer and could potentially cause harm to the state of   
 website if they did. There is a process to follow to gain access to the data access layer  
 contained in the repository documentation.

**4. Tool overview (one for each)**

**a) NodeJS** Node is used to install packages needed by the website. It installs all of the required   
 dependencies, such as socket, port, and hosting modules.

**b) Vue CLI** Vue is a CLI used to setup the initial repository. This is not necessary anymore since   
 the repository is established. However, there may be some reason to use it in the   
 future for this application.  
  
 **c) Visual Studio Code** VS Code, as its often referred to, is a lightweight text editor. Despite VS code being  
called a text editor, is has nearly infinite scalability. Since it is open source, any   
 contributor can add extensions to the marketplace (of course following specific coding   
 guidelines). Many extensions can be used in the project but none are necessary.   
  
 **d) Firebase CLI**

The firebase cli is a node module that can be downloaded via the npm command in a   
 terminal or directly from their website. The cli streamlines the process of setting up  
 the firebase database withing the Vue Javascript application.  
  **5. Project Repository**

**The project repository is being stored on GitHub. All source code is available by   
 “cloning” the project and following the guidelines listed in the “Installation for new   
 install” section of this document (step 6).**

**6. Installation for new install -AND- 7. Installation for new platform**

1. Navigate to the URL:<https://github.com/Trauma-Troopers/Nursing-App>
2. On the right-hand side of the page click the drop down “Clone or Download.”
3. In this step either do one of the following:

* Click “Download ZIP” and unzip the folder in your desired directory.  
  -OR-
* Perform a “git clone” in a terminal with the link in the text box to your desired directory. Please research “git clone” to utilize this method as it goes beyond the scope of this document. In the end, it does nothing different then the first method

1. Download NodeJS (<https://nodejs.org/en/download/>) with the appropriate installer for your machine (eg. 32bit vs 64bit, windows vs mac, etc.).   
   \*Tip: Make sure this is installed in your Environmental Variables PATH
2. Navigate to the folder path ‘Nursing-App\clientform.’ This is where your package.json is stored which tells NodeJS which packages/dependencies to download.
3. In a terminal, type ‘npm install’. Node will begin downloading all the necessities for the project to run locally.
4. In a terminal, type ‘npm run dev’. The terminal will then say the project is running on ‘localhost:xxxx’ where ‘xxxx’ is the port number that is it running locally on your machine.
5. Type that ‘localhost:xxxx’ in a web browser (Chrome preferred for its amazing dev tools) and you will see the project running locally.

**8. Further development statement (if I had another year to do this I would …)**

There are many optimizations to make to the current IUS Nursing Skills Checklist website.

If I had another year I would do the following (by category):

Signup/Login User Interface:

* Reset password functionality and appropriate UI verbiage

Student User Interface:

* A “Toggle Edit” button/checkbox when a students check items are populated. The page defaults to editable. So this would shut off editing for the current session. This may be useful if the user wishes to leave the page open but does not want data altered. (The button for the Admin user interface below is quite different).
* A single page of all checklist items (currently it only filters by category)
* An export to excel spreadsheet conversion button (there is open source software that converts Firebase JSON to xlxs format)

Admin User Interface:

* A “Toggle Edit” button/checkbox when a students check items are populated. A robust and detailed warning message in bold border, bold letters, and red font should be displayed beside this button because it is dangerous! The page defaults to having the checkboxes greyed out and not editable; this functionality was purposeful and was so admins didn’t unintentionally edit student data because data can be manipulated from the admin user interface.
* Fix a defect where categories are populated each time student data is populated asynchronously.
  + To test this:
    - Enter a students name in the admin page
    - Enter another students username in the admin page
    - The categories populate twice but function for the current user (scroll right on the browser)
    - The categories should not populate more than the first time
* Provide more in depth queries of student data for Admins. Currently the database only changes a Boolean value when a student checks an item. It would be ideal for the database to also provide a “getdate()” (sorry I still think in SQL) for the current time of the click. Then the admins would be able to query what date an item for a specific user was checked. This is a larger task than it seems because it is multi-dimensional changes to the database as well as the controllers/UI.

Firebase database:

* As stated in the ‘Admin User Interface’ section above, we need a way to provide more details for querying. There may need to be more things added to the collections in the database.