

# eHealth Dashboard - Brandan King 18007325

## Installation

eHealth dashboard requires [Node.js](#) v4+ to run its dependencies.

### Install the dependencies

Running these npm functions will allow you to generate new JS and CSS files.

```
$ npm install
$ npm run start
```

## Tech

eHealth dashboard was build using a bunch of different frameworks:

- [node.js](#) - used to manage node modules
- [VueJS](#) - Front end framework
- [tailwind](#) - CSS Library
- [chartjs](#) - used to generate JS charts
- [axios](#) - Used to create http requests via JS
- [codeigniter](#) - PHP MVC Framework
- [Gulp](#) - the streaming build system
- [webpack](#) - Used to compile JS

## Application Overview

The ehealth dashboard has 2 main purposes. Allow users to login/register and to create/update/view their questionnaire.

You have 2 different types of users, the admin and a standard user, which both have access to different functionality within the site. Each user has the ability to login to the dashboard and see the status of their questionnaire and be able to update/submit the questionnaire. The admin has the ability to see all users questionnaires, add a user, delete a user, approve/reject a users questionnaire as well as having access to different charts that show stats about the data that has been submitted.

## Setup

Create folder called coursework within xampp / htdocs. This is required or sites content will not work due to config. Then in PHP My Admin create a database called ehealth and import the ehealth.sql database that is on the root of this application

## Login

The application has a Auth controller and Model that is used to control the login page. A user can both login / register from this controller.

When a user tries to register there is validation ran to check to see if the username and email already exist within the Database to prevent duplicate users. Once a user is registered they are then logged into the system.

When a user tries to login the validation checks to see if the username and password match a user within the database.

### **Questionnaire**

The questionnaire is managed by the Users controller and model. It has multiple different functions build within it from getting the users data to show within the table before the questionnaire, allowing a user to be deleted and added, allow a questionnaire to be created and updated by the user as well as approving and rejecting a users form.

### **Chart**

The charts are managed by the chartdata controller and model. It returns the data for each chart back to the http call that was made from the javascript function

### **VueJs**

Vue is used to manage all of the front end functionality that the user gets as well as making the http calls to the models within the function. The user will never be redirected from the dashboard when loading their form/updating their form etc. This has been used to create a single page application which takes advantage of the Axios library.