HI-ER Readme

Getting the Application

The application consists of many folders and files. In order to allow multiple people to work on the HI-ER content, and for continued application development, the HI-ER is managed through a `git` repository. `git` is a platform that allows for very powerful version control - it tracks changes and allows you to "commit" them and "push" your changes to a central repository so others can always see the latest work.

Git needs to be installed on your machine. You can download it here

We will not be using many features of `git`, and since the number of people editing the HI-ER content will somewhat limited, it will be rare for you to need any advanced knowledge of how git works.

The initial step is to create a Github account - this website is the central repository for the HI-ER, and it is a private repository - you will need to be granted access before moving forward. Create your account here, and contact me (sfrees@intelliquip.com) for access.

Once you have access, open the Command Prompt and use the `cd` command to navigate to the directory you want to put the HI-ER application in. For example, create a directory called `C:\projects\`, and navigate the command prompt there by entering `cd C:\projects` at the command prompt.

Next, clone the repository with the following command:

```
git clone https://github.com/edlpumps/hi-er.git
```

This will create a directory called `hi-er` under your `C:\projects` directory.

Later, as you begin to make changes to the HI-ER, you will need to use `git` to commit your changes. The details of this is covered below in the "Version control with `git`" section.

Text Editor - Visual Studio Code

While any text editor is suitable for creating the HI-ER content, you should use something geared towards programming, to avoid character encoding problems. You may download and install this here.

Once installed, you can open the `hi-er` directory.

Visual Studio Code allows you to open a folder - `hi-er` - which is the most efficient way of working. This will give you a side panel on the left side of the screen that you can use to navigate and open any file in the directory structure.

Running code Locally

Initialize Node JS and NPM in your Development environment

Follow instructions at: `https://code.visualstudio.com/docs/nodejs/nodejs-tutorial`

Additional information can be found at: `https://developer.ibm.com/tutorials/learn-nodejs-installing-node-nvm-and-vscode/`

Confirm npm is installed

```
$> npm --help
```

Confirm node js is installed

```
$> node --version
```

Just one time, install the modules for your development environment:

In your VSC app and in a `powershell` terminal window:

```
$> npm init
$> npm install
$> npm audit fix <== if needed
```

Local Database

You must be connected to a local Mongoose database install MongoDB install MongoDB Compass create .env with:

- MONGO_CONNECTION_DATA = mongodb://127.0.0.1:27017/er
- ADMIN_PASSWORD_OVERRIDE=test You will need to have your user information in the database. $>npm init $>npm install $>npm -g install supervisor $>supervisor index.js

Local Preview - Debugging

The easiest way to do your development is to execute the following command from the root project directory in a Terminal window:

```
$>  node index.js
```

OR to set breakpoints and debug:

```
Open index.js file in VSC
Click on the Debug icon on the left
Select Launch Program
Step through code
Use the debug toolbar to step into/over code
Use the debug toolbar to restart the code and stop execution
```

Connect to the localhost on port:

```
http://127.0.0.1:3003
```

Deploying a GitHub branch to Heroku

First, push the branch to GitHub and create a Pull Request.

Login to Heroku using the `higladetech@gmail.com` account on a browser:

```
https://heroku.com
Select the "intelliquip-hi/hi-er-beta" project (make sure that it is NOT in
Maintenance mode in the Settings)
Select Deploy
Select Deply a GitHub branch and enter the branch name
```

Deploying GitHub Master branch to Heroku

Make sure all branches are merged to `master` on GitHub.

Follow the steps above to Deploy a GitHub branch to Heroku, but instead, select the `intelliquip-hi/hi-er` project and deploy the `master` branch.