Before handing off the application we will package our electron application using the npm tool electron-packager. There are multiple potential tools we can use as outlined on the electron website including electron-packager, electron-forge, and electron-builder. This tool will package the application into a .app or .exe for a specific OS, in our client's case Ubuntu, and bundle the necessary node modules along with it.

How to make the executables

For our case, with the electron-package manager the following steps can be taken to create an executable for OSX, Windows, and Linux x86 architectures (electron-packager):

Install the electron-packager dependency:

```
        `npm install electron-packager -g`
```

Navigate to the directory containing the electron app code and run:

```
        `electron-packager <sourcedir> <appname> [options...]`
```

This will create an executable for the current host. To specify the creation of an executable for all platforms and architectures simply add an `--all` option

Packaging with this tool will allow our client to have a file that they can run as a normal executable instead of having to run command line commands to compile the code. In order for the client to get the GUI code to edit and update, they just need to simply pull the code from our github and edit the code from there. The client will then need to run "npm install -g electron" to get the current version of electron. Run `npm install` to get the necessary node modules installed. From there run "npm start" to transpile and compile the necessary files and run the application. We will meet with the client to further discuss the hand off plan.

https://www.npmjs.com/package/electron-packager
https://github.com/benl1/objectDetectionGUI