

Data Visualization Generator

Technical Installation Manual

Doofenshmirtz Evil Inc

COS 301 - 2020

Marco Lombaard u18026975

Elna Pistorius u18010319

Phillip Schulze u18171185

Byron Tomkinson u18042717

Gian Uys u18052569



DATA
VISUALIZATION GENERATOR

Contents

1	Introduction	2
2	Prerequisites	2
2.1	Requirements	2
2.2	Instructions on how to install prerequisites	3
2.2.1	UNIX System	3
2.2.2	Windows	4
3	Installation	7
3.1	UNIX System	7
3.2	Windows	8
4	Deployment/Running	15
4.1	UNIX systems	15
4.2	Windows	15

1 Introduction

The system consists of a front-end and a back-end. The front-end is a Web Application that is created using React.js (and various other node modules). This web application communicates with our back-end via HTTP-requests to specific REST endpoints. The back-end is a Node Web Server that is created using Express.js (and various other node modules). This server requests data from an external source, provided by the user, and then generates suggestions to visualise this data using an Interactive Genetic Algorithm.

2 Prerequisites

In this section we describe the necessary software and packages that are needed for our system to work. First we list the requirements and thereafter we specify instructions on how to install the prerequisites.

2.1 Requirements

1. Node Package Manager (npm) and Node.js

2. Dependencies

- axios 0.19.2
- bcryptjs 2.4.3
- connect-pg-simple 6.1.0
- cors 2.8.5
- dom-parser 0.1.6
- dotenv 8.2.0
- error 10.4.0
- events 3.1.0
- express 4.17.1
- express-session 1.17.1
- jest 26.2.2
- pg 8.2.1
- pg-pool 3.2.1
- rewire 5.0.0
- save 2.4.0
- url 0.11.0
- xmldom 0.3.0
- @material-ui/core latest
- @material-ui/icons latest
- @testing-library/jest-dom 4.2.4
- @testing-library/react 9.5.0
- @testing-library/user-event 7.2.1
- antd 4.4.0
- axios 0.19.2
- echarts 4.8.0
- echarts-for-react 2.0.16
- node-sass 4.14.1
- prop-types latest
- react 16.13.1
- react-addons-update 15.6.2
- react-dom 16.13.1
- react-grid-layout 0.18.3
- react-scripts 3.4.1
- react-test-renderer 16.13.1
- react-undo 1.2.1
- request 2.0.5
- styled-components 5.1.1
- styled-icons 10.6.0
- three 0.118.3
- three-orbitcontrols 2.110.3
- use-undo 1.0.3

3. Development Dependencies

- eslint-plugin-flowtype 5.2.0
- eslint-plugin-import 2.22.0
- eslint-plugin-jsx-a11y 6.3.1
- eslint-plugin-react 7.20.3
- eslint-plugin-react-hooks 4.0.8
- eslint-watch 7.0.0
- tailwindcss 1.1.3

2.2 Instructions on how to install prerequisites

All these packages named above works through **Node Package Manager (npm)**. The process works as follows:

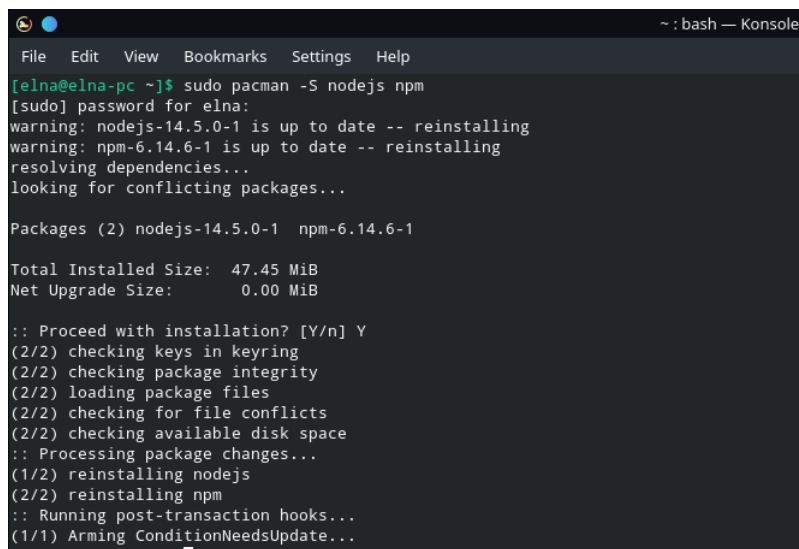
1. Install Node.js and npm packages
2. Install the **dependencies** and **development dependencies** using npm.

2.2.1 UNIX System

- To install Node.js and npm packages please use the system package manager or use the terminal. Beneath is the instructions to install via the terminal.

– For ArchLinux or Manjaro distribution

1. To install the Node.js and npm packages simultaneously please run **sudo pacman -S nodejs npm** in the terminal.



```
~ : bash — Konsole
File Edit View Bookmarks Settings Help
[elna@elna-pc ~]$ sudo pacman -S nodejs npm
[sudo] password for elna:
warning: nodejs-14.5.0-1 is up to date -- reinstalling
warning: npm-6.14.6-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

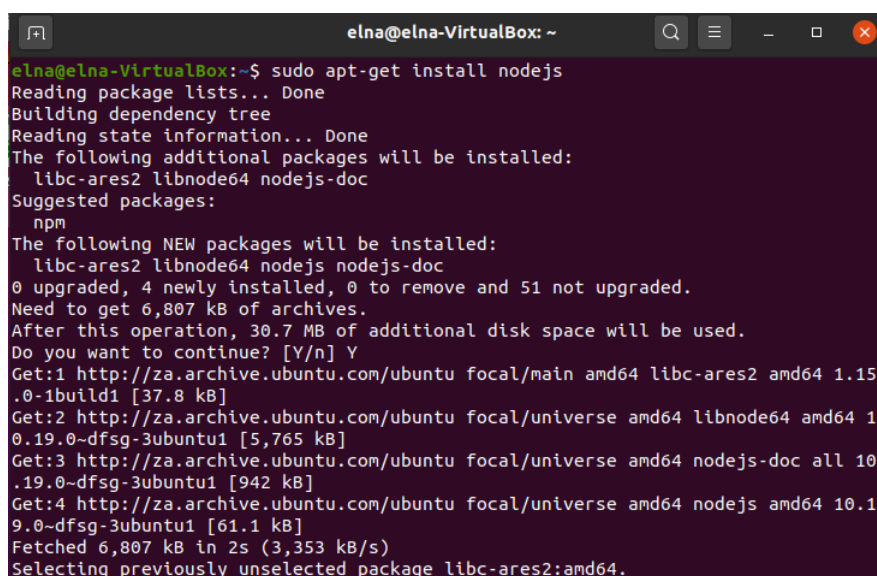
Packages (2) nodejs-14.5.0-1  npm-6.14.6-1

Total Installed Size:  47.45 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n] Y
(2/2) checking keys in keyring
(2/2) checking package integrity
(2/2) loading package files
(2/2) checking for file conflicts
(2/2) checking available disk space
:: Processing package changes...
(1/2) reinstalling nodejs
(2/2) reinstalling npm
:: Running post-transaction hooks...
(1/1) Arming ConditionNeedsUpdate...
```

– For Ubuntu distribution

1. To install the Node.js please run **sudo apt-get install nodejs** in the terminal.



```
elna@elna-VirtualBox: ~
elna@elna-VirtualBox:~$ sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libnode64 nodejs-doc
Suggested packages:
  npm
The following NEW packages will be installed:
  libc-ares2 libnode64 nodejs nodejs-doc
0 upgraded, 4 newly installed, 0 to remove and 51 not upgraded.
Need to get 6,807 kB of archives.
After this operation, 30.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://za.archive.ubuntu.com/ubuntu focal/main amd64 libc-ares2 amd64 1.15
.0-1build1 [37.8 kB]
Get:2 http://za.archive.ubuntu.com/ubuntu focal/universe amd64 libnode64 amd64 1
0.19.0-dfsg-3ubuntu1 [5,765 kB]
Get:3 http://za.archive.ubuntu.com/ubuntu focal/universe amd64 nodejs-doc all 10
.19.0-dfsg-3ubuntu1 [942 kB]
Get:4 http://za.archive.ubuntu.com/ubuntu focal/universe amd64 nodejs amd64 10.1
9.0-dfsg-3ubuntu1 [61.1 kB]
Fetched 6,807 kB in 2s (3,353 kB/s)
Selecting previously unselected package libc-ares2:amd64.
```

2. To install the npm package please run **sudo apt-get install npm** in the terminal.

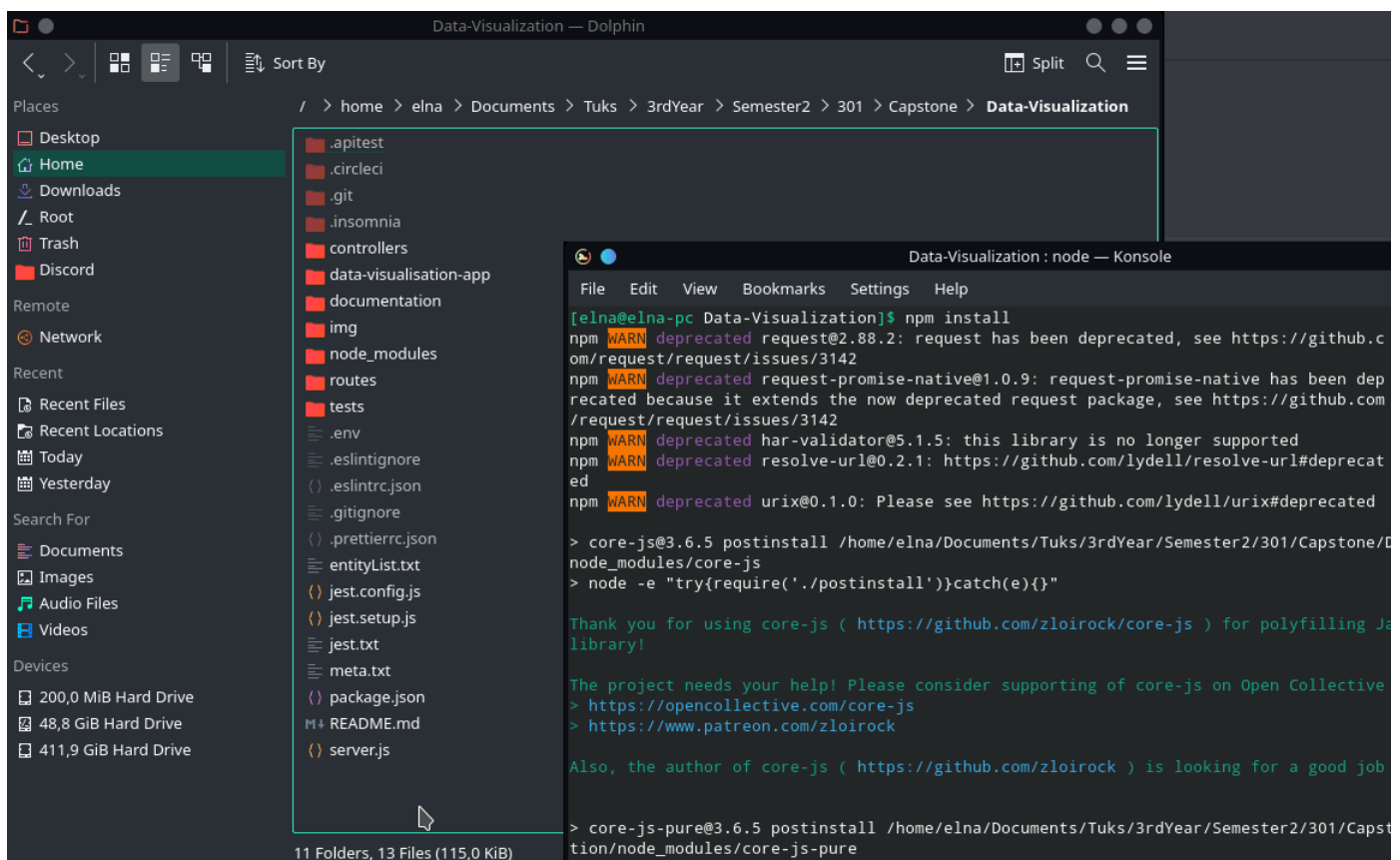
```

elna@elna-VirtualBox: ~
$ sudo apt-get install npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 gyp javascript-common libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libfakeroot libgcc-9-dev libitm1 libjs-inherits libjs-is-typedarray
  libjs-psl libjs-typedarray-to-buffer liblsan0 libnode-dev libpython2-stdlib
  libpython2.7-minimal libpython2.7-stdlib libquadmath0 libssl-dev
  libstdc++-9-dev libtsan0 libubsan1 libuv1-dev linux-libc-dev make
  manpages-dev node-abbrev node-ajv node-ansi node-ansi-align node-ansi-regex
  node-ansi-styles node-ansistyles node-aproba node-archy
  node-are-we-there-yet node-asap node-asn1 node-assert-plus node-async
  node-aws-sign2 node-aws4 node-balanced-match node-bcrypt-pbkdf node-bl
  node-bluebird node-boxen node-brace-expansion node-builtin-modules
  node-builtins node-cacache node-call-limit node-camelcase node-caseless
  node-chalk node-chownr node-ci-info node-cli-boxes node-clui node-clone
  node-co node-color-convert node-color-name node-colors node-columnify
  node-combined-stream node-concat-map node-concat-stream node-config-chain
  node-configstore node-console-control-strings node-copy-concurrently
  node-core-util-is node-cross-spawn node-crypto-random-string node-cyclist

```

- To install all the **Dependencies and Development Dependencies**

1. Run **npm install** in the terminal of the root directory of the Data-Visualization project (please refer to the **Installation Section** for more information of where the command should be executed)



2.2.2 Windows

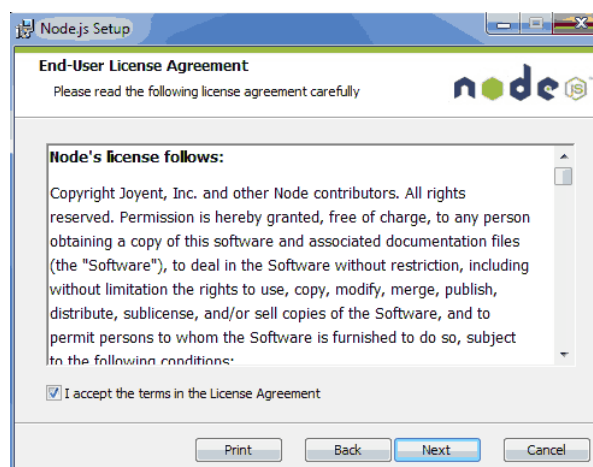
- To install Nodejs and npm
 1. **Go to this site** to download the Windows Installer (.msi).

		
Windows Installer node-v0.10.21-x86.msi	Macintosh Installer node-v0.10.21.pkg	Source Code node-v0.10.21.tar.gz
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)	Universal	
Mac OS X Binaries (.tar.gz)	32-bit	64-bit
Linux Binaries (.tar.gz)	32-bit	64-bit
SunOS Binaries (.tar.gz)	32-bit	64-bit
Source Code	node-v0.10.21.tar.gz	

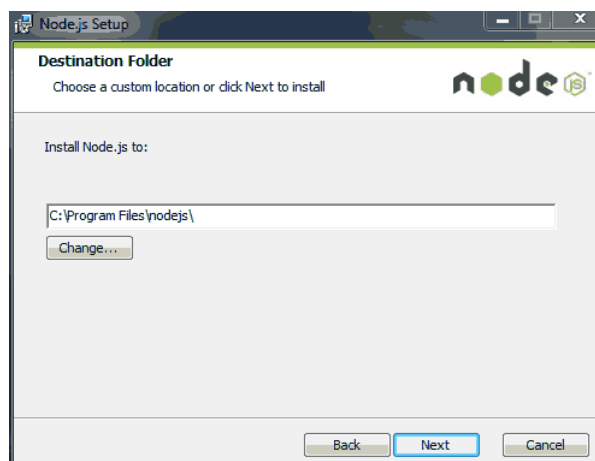
- Once it is installed, go to your downloads and double click on the downloaded folder to launch the executable. The system will ask if you want to run the software – click **Run**.
- The Node.js setup wizard will pop up - click **Next**.



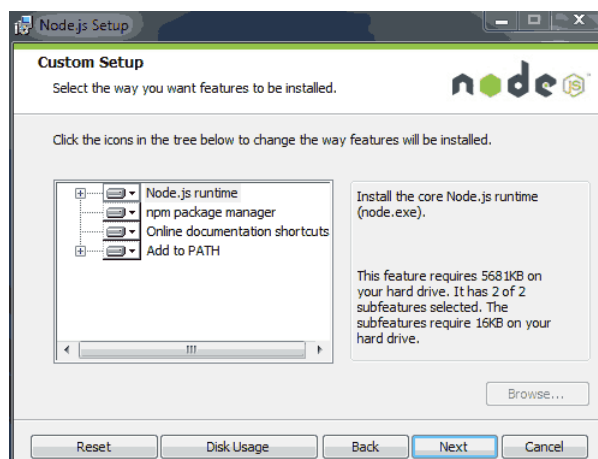
- If you accept the End-User License Agreement- check **the box** that says "I accept the terms in the License Agreement".



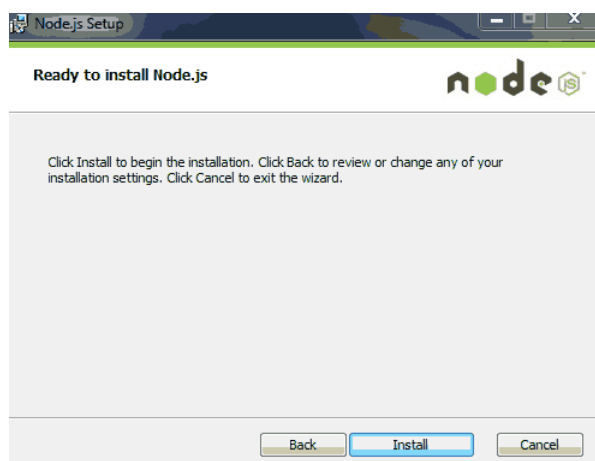
- The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – click **Next**.



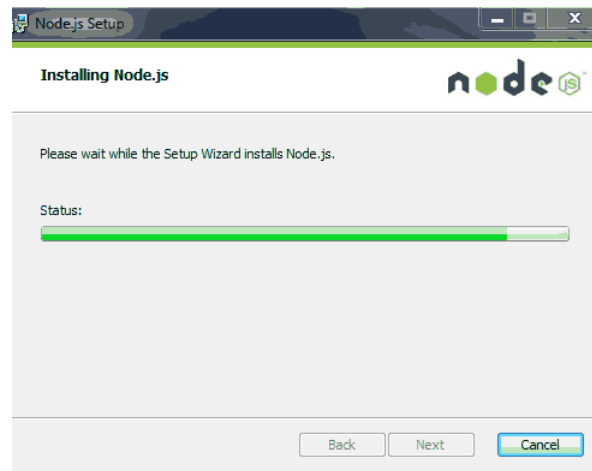
6. You are then prompted to select components to include or remove from the installation. Unless you have a specific need, to accept the defaults - click **Next**.



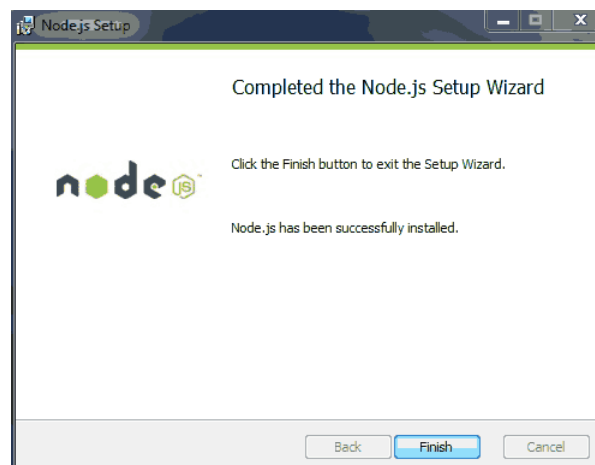
7. Finally, to run the installer - click **Install**.



8. You will see the installation progresses as shown below.



9. Node.js has been successfully installed to exit the Node.js Setup Wizard - click **Finish**.



10. Restart your computer. (This is required)

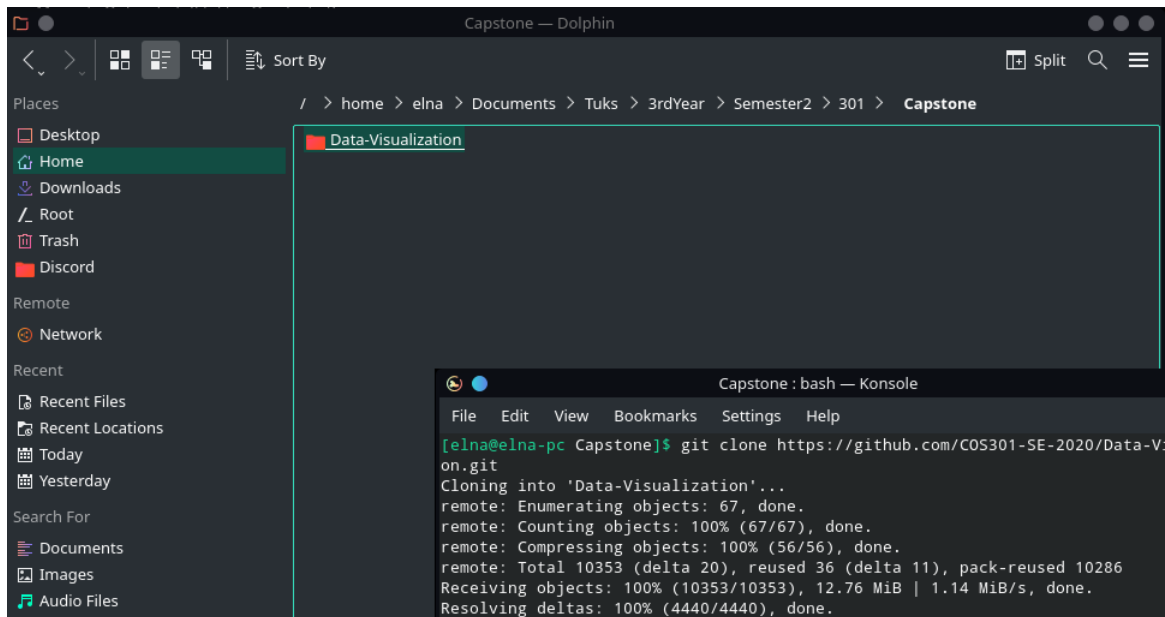
- To install all the **Dependencies and Development Dependencies** run **npm install** in the terminal of the root directory of the Data-Visualization project (please refer to the **Installation Section** for more information of where the command should be executed)

3 Installation

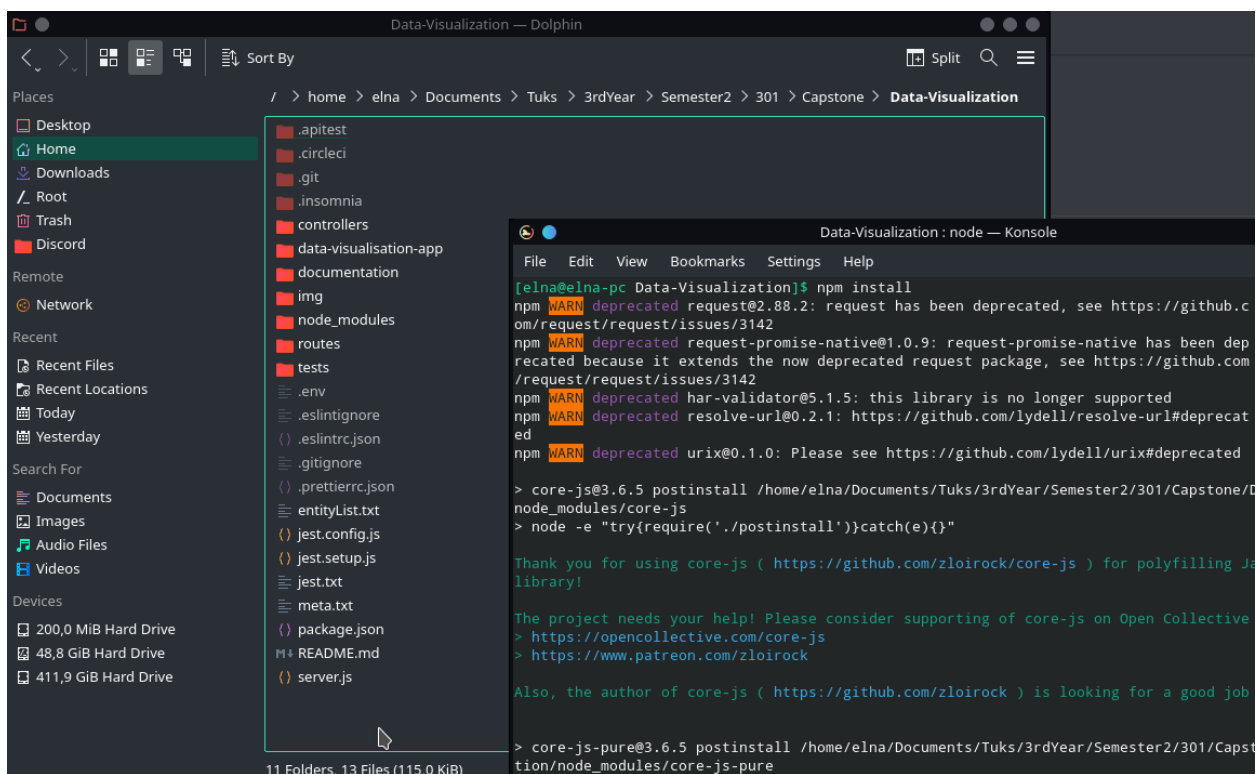
This section describes the steps to follow to install the Data-Visualization system on the different operating system distributions.

3.1 UNIX System

1. Clone the repo by creating a new folder calling it any name preferred, in this folder open the terminal and type **git clone <https://github.com/COS301-SE-2020/Data-Visualization.git>**



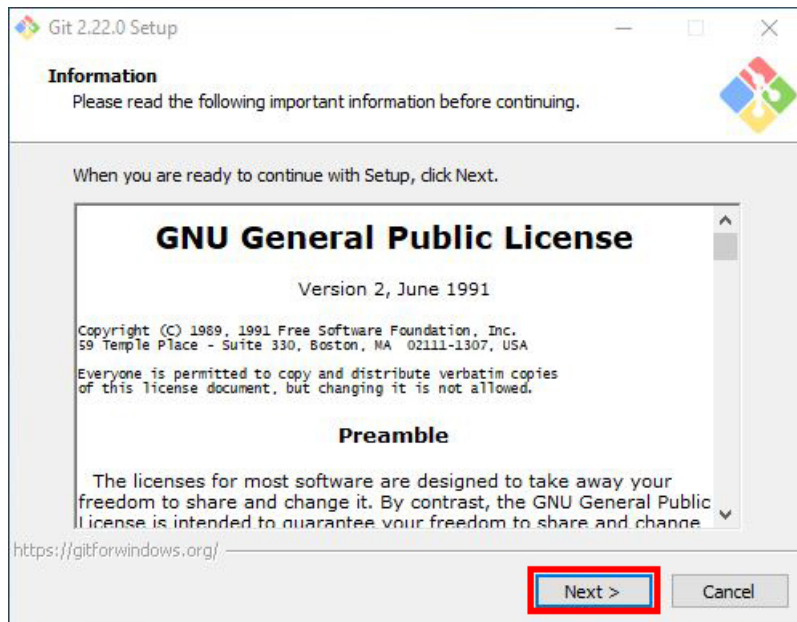
2. Inside the root directory of Data Visualization type in the terminal **npm install** (This will install all the Dependencies and Development Dependencies packages as stated in the Prerequisites Section.



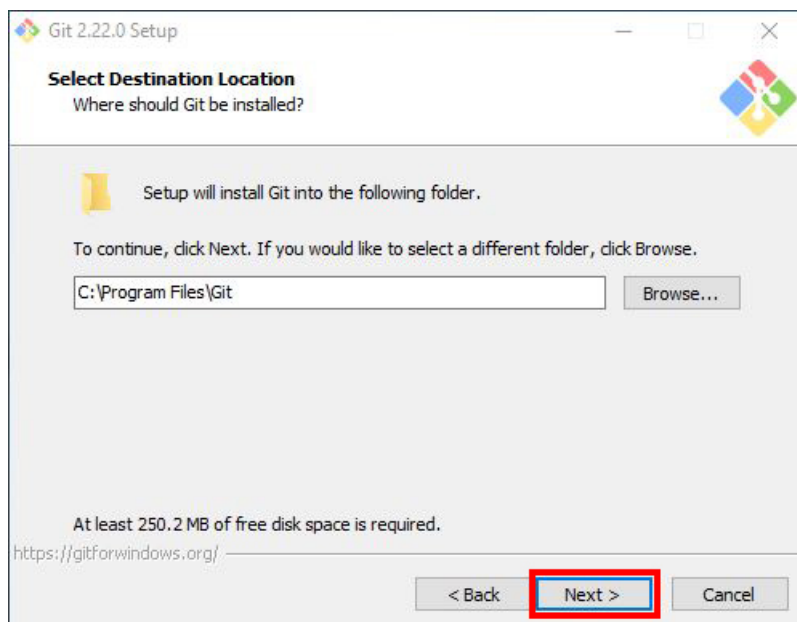
3.2 Windows

- **Click Here To Install Git for Windows.**

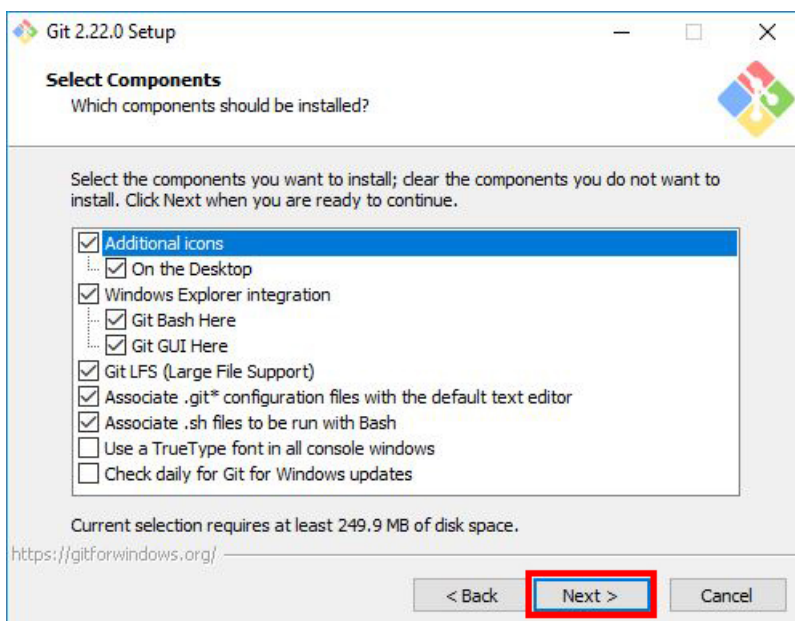
1. Once it is installed, go to your downloads and double click on the downloaded folder to launch the executable. The system will ask if you want to run the software – click **Run**.
2. After you have read the license - click **Next**



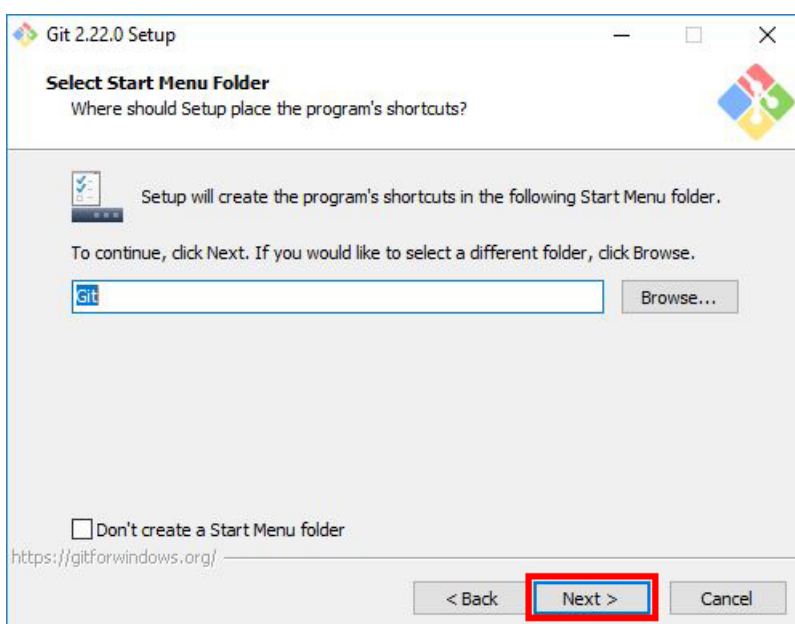
3. The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – click **Next**



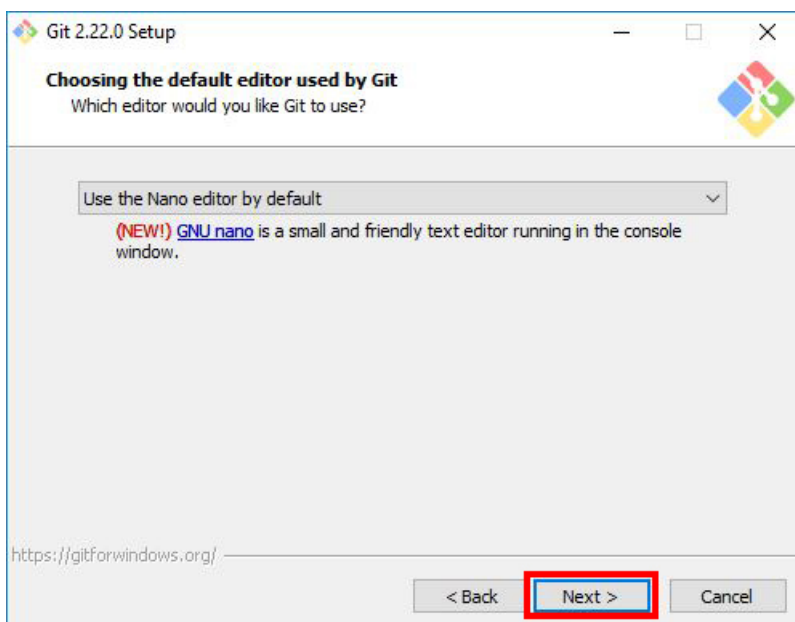
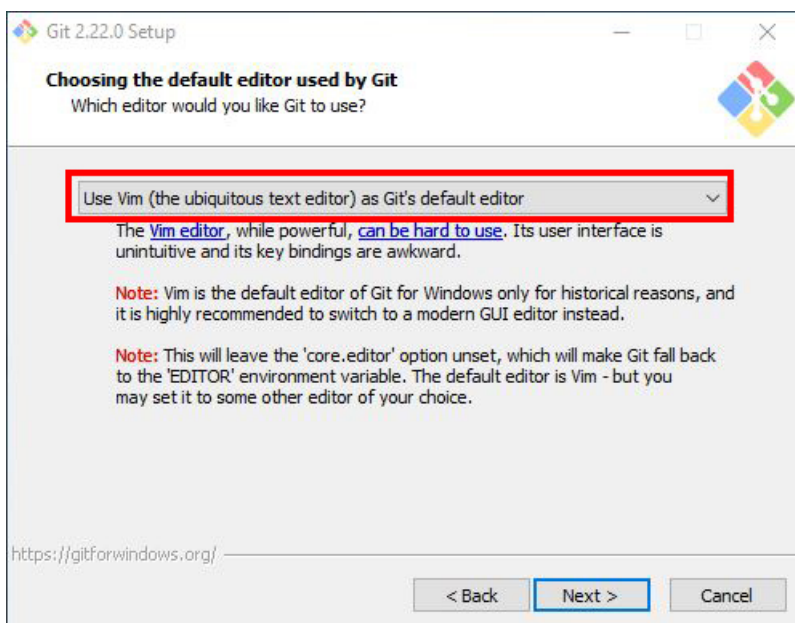
4. Select the components you want to install, or proceed with the default options- click **Next**



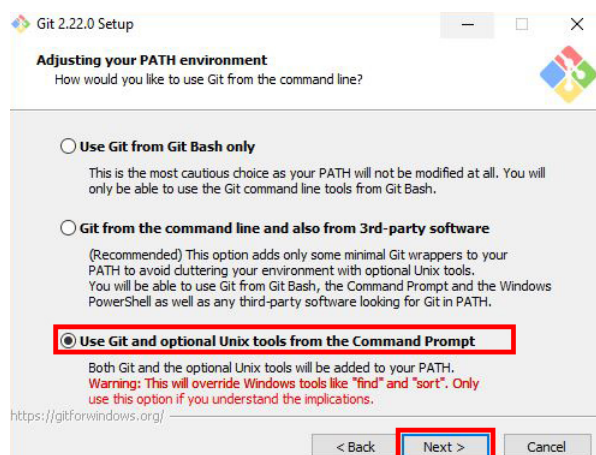
5. You can change the name of start menu folder here if you want, or just leave the default name and - click Next.



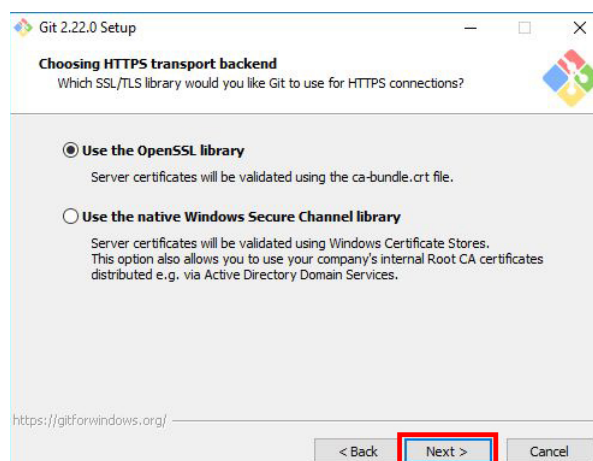
6. Next, select the default editor for Git to use. Choose the one you like and - click **Next**



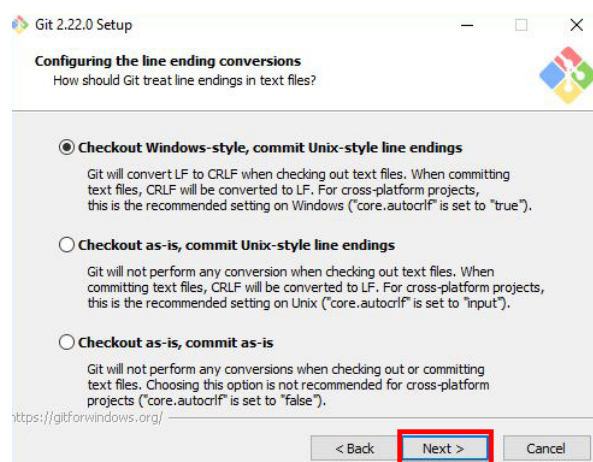
7. To adjust the PATH environment as preferred and - click **Next**



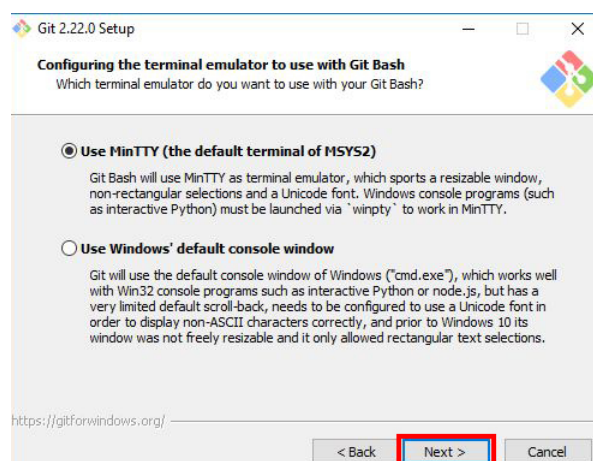
8. Choose the preferred HTTPS Transport Backend and - click **Next**



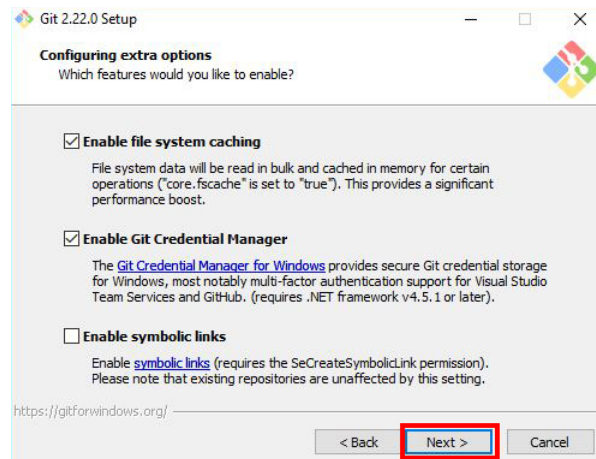
9. Configure the Line Ending Conversions to your preference and - click **Next**



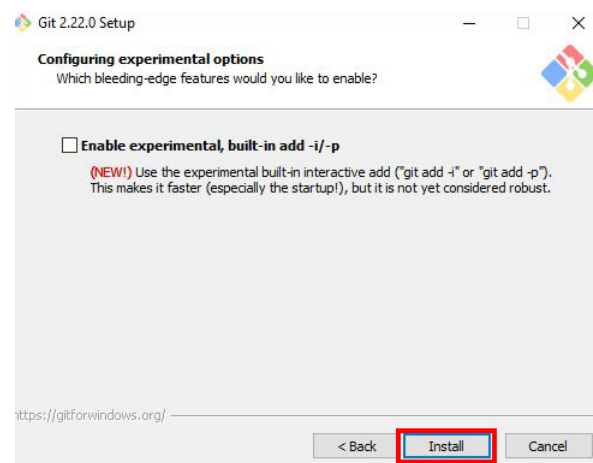
10. Configure the Terminal Emulator to use with Git Bash as preferred and - click **Next**



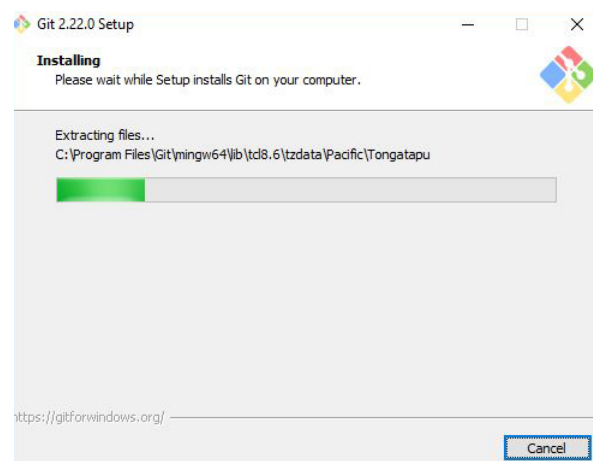
11. Configuring Extra Options as preferred and - click **Next**



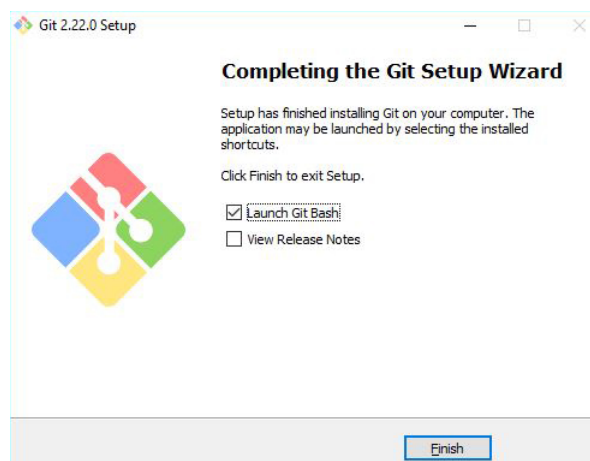
12. Configuring Extra Options as preferred and - click **Next**



13. Wait for Installation to happen.



14. To complete the Git Setup Wizard - click **Next**



- Clone the repo by creating a new folder calling it any name preferred, in this folder open the terminal and type **git clone <https://github.com/COS301-SE-2020/Data-Visualization.git>**

```
Windows PowerShell
PS C:\temp> git clone https://github.com/COS301-SE-2020/Data-Visualization.git
Cloning into 'Data-Visualization' ...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 10372 (delta 33), reused 58 (delta 24), pack-reused 10286
Receiving objects: 100% (10372/10372), 12.76 MiB | 209.00 KiB/s, done.
Resolving deltas: 100% (4453/4453), done.
PS C:\temp>
```

- Inside the root directory type in the terminal **npm install** and then **npm run install:front** (These will install all the Dependencies and Development Dependencies packages as stated in the Prerequisites Section.)

4 Deployment/Running

4.1 UNIX systems

- (a) First the project must be built. It lets you perform any necessary building/prep tasks for your project. Open the terminal in the root directory of the cloned repository and type: **npm run build**

```
phillipstemmlar:~/dev/capstone/Data-Visualization$ npm run build
> data-visualisation@1.0.0 build /home/phillipstemmlar/dev/capstone/Data-Visualization
> npm run build --prefix ./data-visualisation-app

> data-visualisation-app@0.1.0 build /home/phillipstemmlar/dev/capstone/Data-Visualization/data-visualisation-app
> react-scripts build

Creating an optimized production build...
File sizes after gzip:

 718.79 KB  build/static/js/2.37e70644.chunk.js
149.24 KB  build/static/js/main.9b04ebb3.chunk.js
 63.99 KB  build/static/css/2.08883718.chunk.css
  3.53 KB  build/static/css/main.3fc7d3b5.chunk.css
   786 B   build/static/js/runtime-main.eb0ed037.js

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  serve -s build

Find out more about deployment here:

  bit.ly/CRA-deploy

phillipstemmlar:~/dev/capstone/Data-Visualization$
```

- (b) After npm build has run, and type **npm start** to start the server, the web-application url can now be visited by going to <http://127.0.0.1:8000> in the web-browser

```
phillipstemmlar:~/dev/capstone/Data-Visualization$ npm start

> data-visualisation@1.0.0 start /home/phillipstemmlar/dev/capstone/Data-Visualization
> node server.js

Server started at http://127.0.0.1:8000
```

- (c) **Click Here** to view our user manual to see how our system works.

4.2 Windows

1. First the project must be built. It lets you perform any necessary building/prep tasks for your project. Open the command prompt in the root directory of the cloned repository and type: **npm run build**


```
phillipstemmlar:~/dev/capstone/Data-Visualization$ npm run build

> data-visualisation@1.0.0 build /home/phillipstemmlar/dev/capstone/Data-Visualization
> npm run build --prefix ./data-visualisation-app

> data-visualisation-app@0.1.0 build /home/phillipstemmlar/dev/capstone/Data-Visualization/data-visualisation-app
> react-scripts build

Creating an optimized production build...
File sizes after gzip:

 718.79 KB   build/static/js/2.37e70644.chunk.js
 149.24 KB   build/static/js/main.9b04ebb3.chunk.js
 63.99 KB    build/static/css/2.08883718.chunk.css
 3.53 KB     build/static/css/main.3fc7d3b5.chunk.css
 786 B       build/static/js/runtime-main.eb0ed037.js

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  serve -s build

Find out more about deployment here:

  bit.ly/CRA-deploy

phillipstemmlar:~/dev/capstone/Data-Visualization$
```

2. After npm build has run, and type **npm start** to start the server, the web-application url can now be visisted by going to <http://127.0.0.1:8000> in the web-browser

```
phillipstemmlar:~/dev/capstone/Data-Visualization$ npm start

> data-visualisation@1.0.0 start /home/phillipstemmlar/dev/capstone/Data-Visualization
> node server.js

Server started at http://127.0.0.1:8000
```

3. **Click Here** to view our user manual to see how our system works.