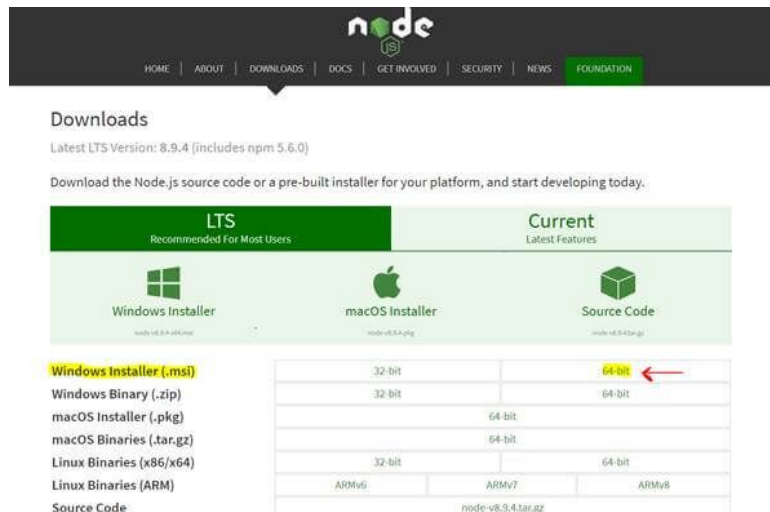


# Steps For installing Front-ENd

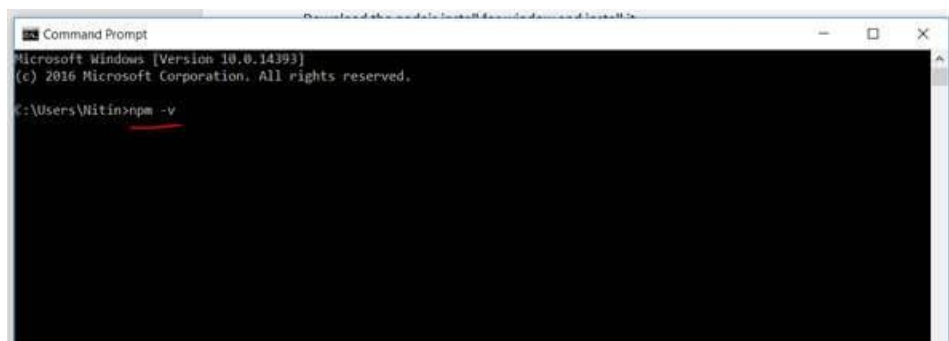
## Step 1 - Install NodeJS

Follow the link - <https://nodejs.org/en/download/>

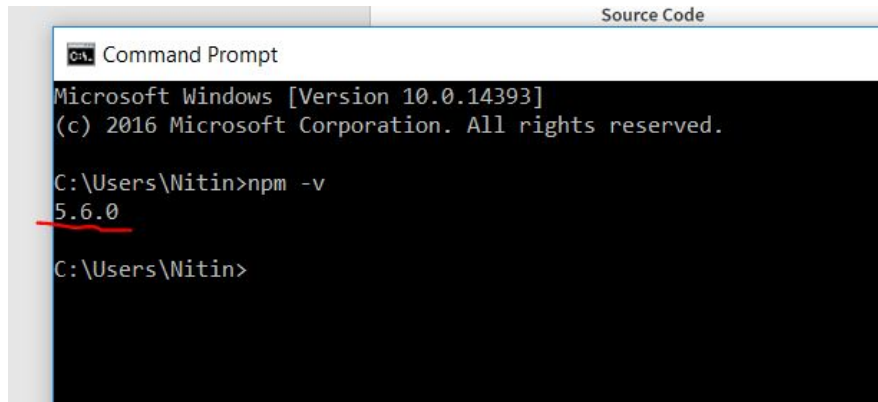
Download the node.js installer for Windows and install it.



To check the installed version of Node.js, open the command prompt.



Type the “*npm -v*” command to check the Node.js installation and version.

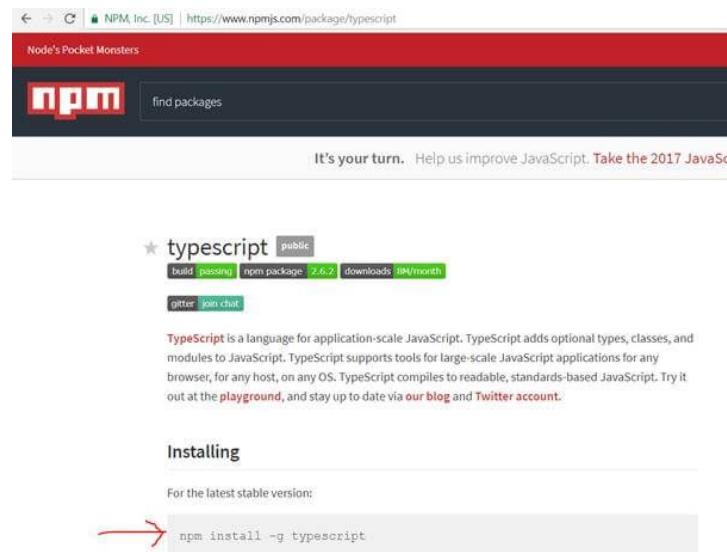


```
Source Code
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Nitin>npm -v
5.6.0
C:\Users\Nitin>
```

## Step 2 - Install TypeScript

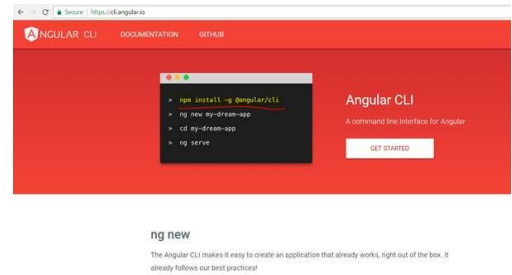
Open the link <https://www.npmjs.com/package/typescript>



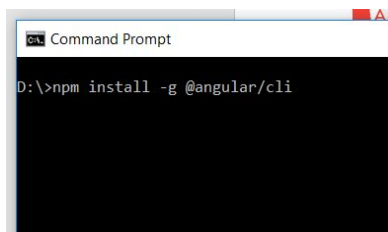
Copy the above command “*npm install -g typescript*” and run it on command prompt.

## Step 3 - Install Angular CLI

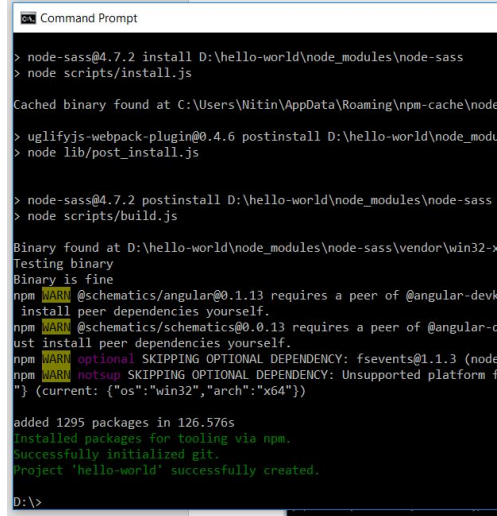
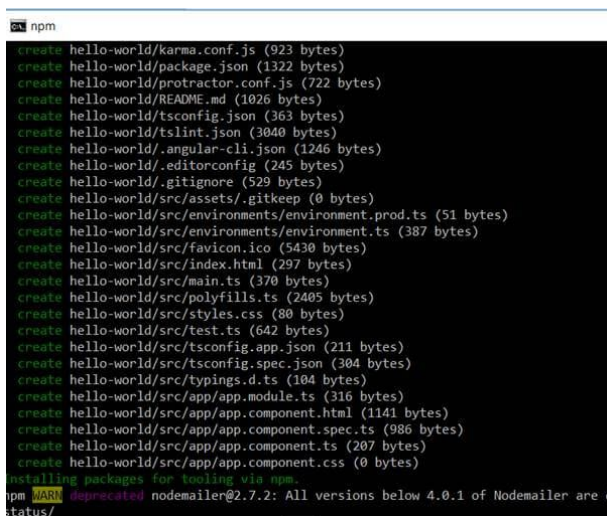
Open the link <https://cli.angular.io/> and follow the instructions to install Angular CLI



1. Type the command “npm install -g @angular/cli” on the command prompt and press enter to install Angular cli.



2. Unzip and navigate to “E-tendering-FrontEnd” folder in project and Type the command “npm install” on the command prompt and press enter to install all needed packages



3. Finally, the "MEET" Angular app is installed with all necessary packages; now type “ng serve”.

```
** NG Live Development Server is listening on localhost:4200, open your browser at http://localhost:4200/
Date: 2018-01-13T04:34:33.320Z
Hash: 9abcb875905c9d05482b
Time: 5325ms
chunk {inline} inline.bundle.js (inline) 5.79 kB [entry] [rendered]
chunk {main} main.bundle.js (main) 19.3 kB [initial] [rendered]
chunk {polyfills} polyfills.bundle.js (polyfills) 549 kB [initial] [rendered]
chunk {styles} styles.bundle.js (styles) 33.5 kB [initial] [rendered]
chunk {vendor} vendor.bundle.js (vendor) 7.4 MB [initial] [rendered]
webpack: Compiled successfully.
```

Now, open the browser and type *http://localhost:4200* in the address bar and hit enter to run the MEET Angular app in the browser.

## Steps For installing Back-End

**Step 1-** Unzip and navigate to “e-tenderingBackEnd” folder in project and Type the command “npm install” on the command prompt inside project and press enter to install all needed packages

**Step 2-** Type the command “npm start” on the command prompt and press enter to start the back-end.

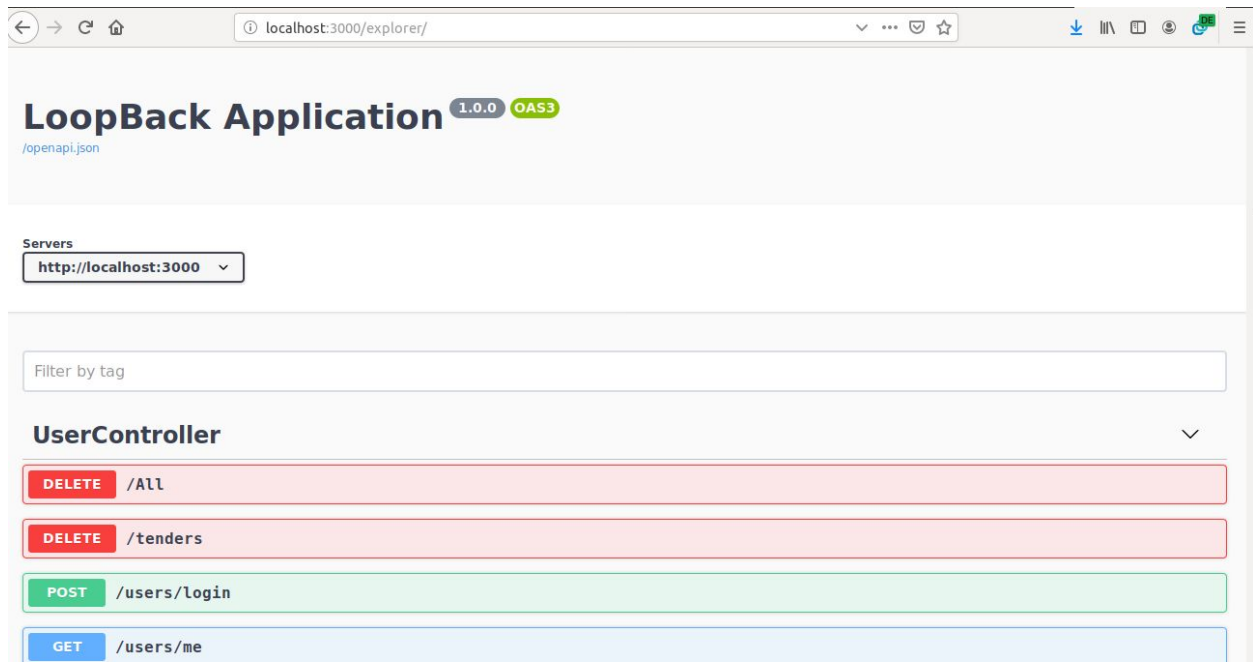
```
habiba@Beebz:~/E-tendering/e-tenderingBackEnd$ npm start
> e-tendering@1.0.0 prestart /home/habiba/E-tendering/e-tenderingBackEnd
> npm run build

> e-tendering@1.0.0 build /home/habiba/E-tendering/e-tenderingBackEnd
> lb-tsc

> e-tendering@1.0.0 start /home/habiba/E-tendering/e-tenderingBackEnd
> node -r source-map-support/register .

Server is running at http://[::1]:3000
```

The BackEnd is ready to launch so now, open the browser and type `http://localhost:3000/explorer/` in the address bar and hit enter to run the MEET Loopback APIs in the browser.



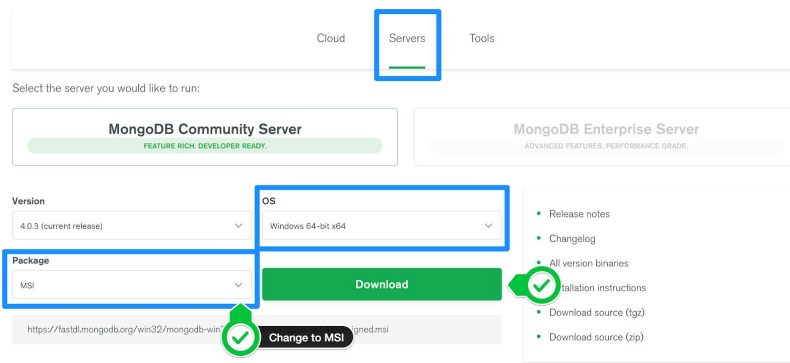
### Step 3 - Install MongoDB for your Database

For widows use follow these steps here or for further clarification use this link :

<https://medium.com/@LondonAppBrewery/how-to-download-install-mongodb-on-windows-4ee4b3493514>

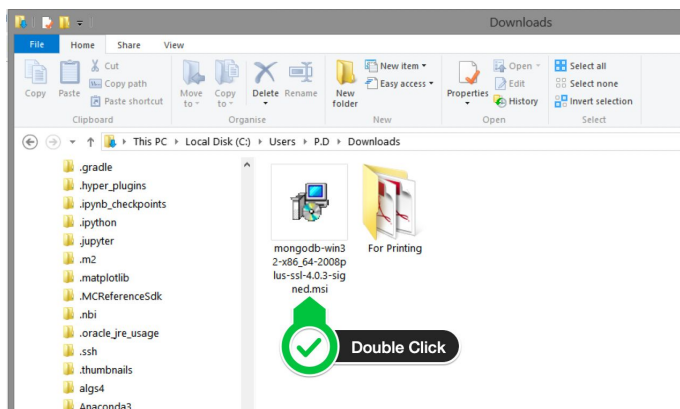
#### Step 3a — Download the MongoDB MSI Installer Package

Head over [here](#) and download the current version of MongoDB. Make sure you select **MSI** as the package you want to download.



## Step 3b — Install MongoDB with the Installation Wizard

A. Make sure you are logged in as a user with Admin privileges. Then navigate to your downloads folder and double click on the .msi package you just downloaded. This will launch the installation wizard.

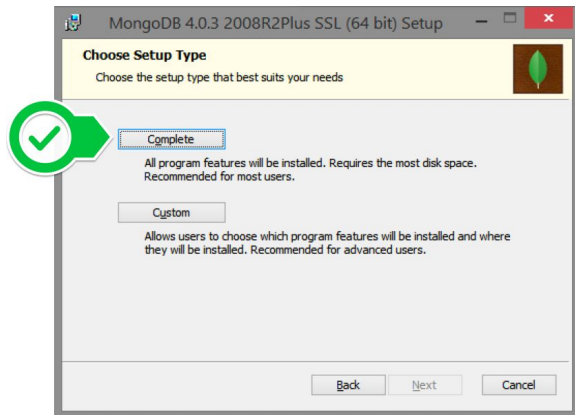


B. Click Next to start installation.

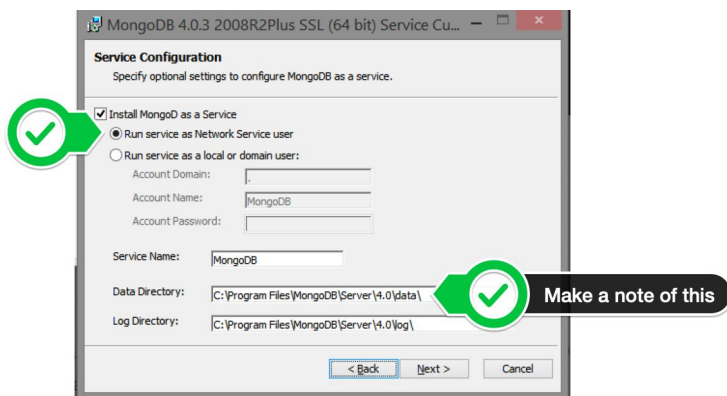


C. Accept the licence agreement then click Next.

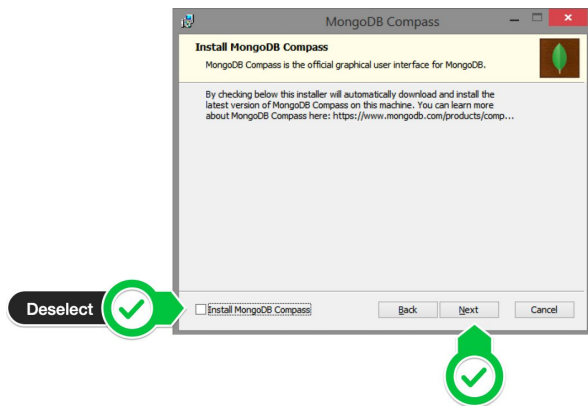
D. Select the Complete setup.



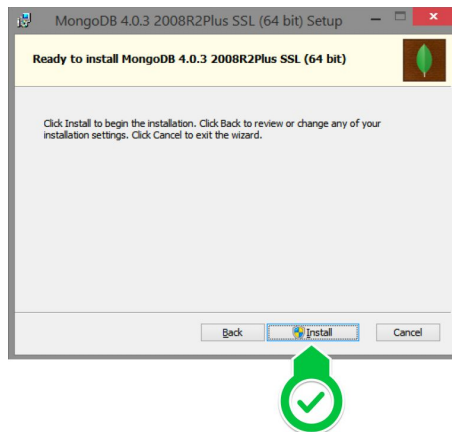
E. Select “Run service as Network Service user” and make a note of the data directory, we’ll need this later.



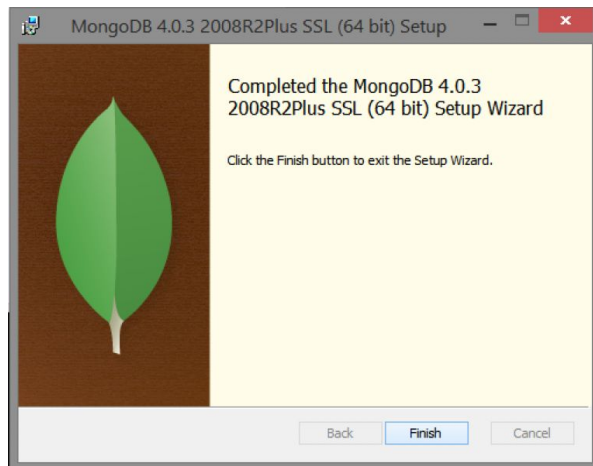
F. We won’t need Mongo Compass, so deselect it and click Next.



G. Click Install to begin installation.



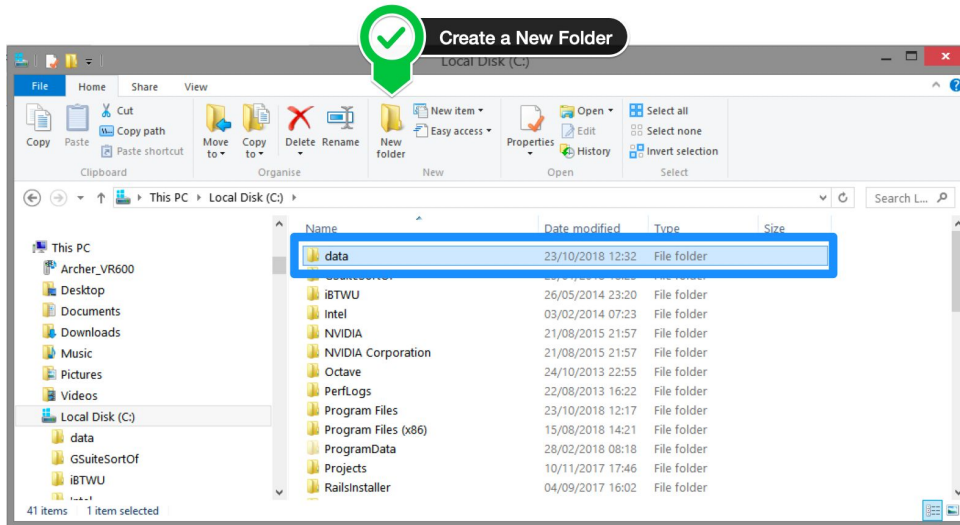
F. Hit Finish to complete installation.



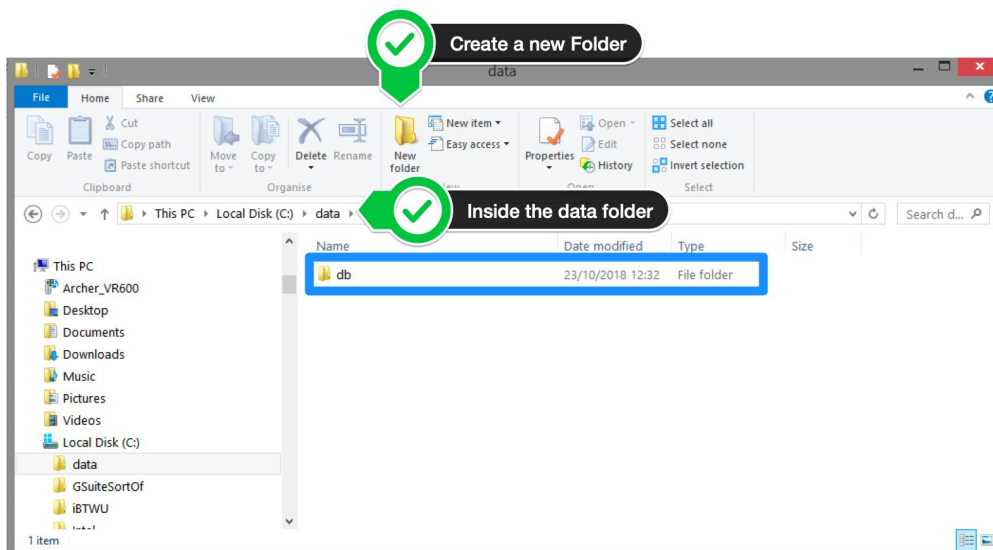
### Step 3— Create the Data Folders to Store our Databases

A. Navigate to the **C Drive** on your computer using Explorer and create a new folder called **eTenderingDatabase(not data)** here.





B. Inside the **data** folder you just created, create another folder called **db**.



## Step 4 — Setup Alias Shortcuts for Mongo and Mongod

Once installation is complete, we'll need to set up MongoDB on the local system.

A. Open up your Hyper terminal running Git Bash.

B. Change directory to your home directory with the following command:

```
cd ~
```

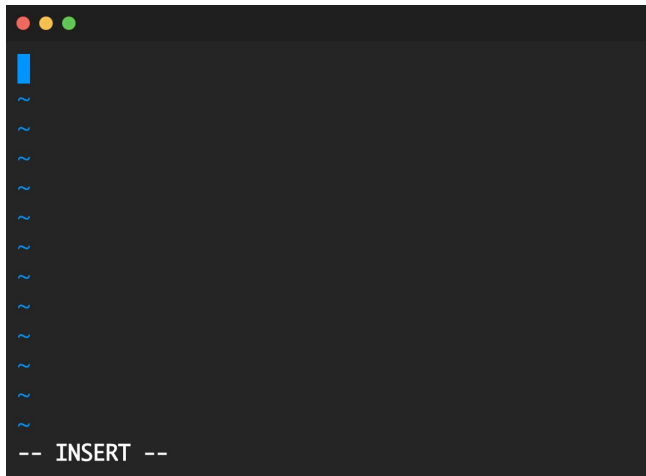
C. Here, we're going to create a file called `.bash_profile` using the following command:

```
touch .bash_profile
```

D. Open the newly created `.bash_profile` with vim using the following command:

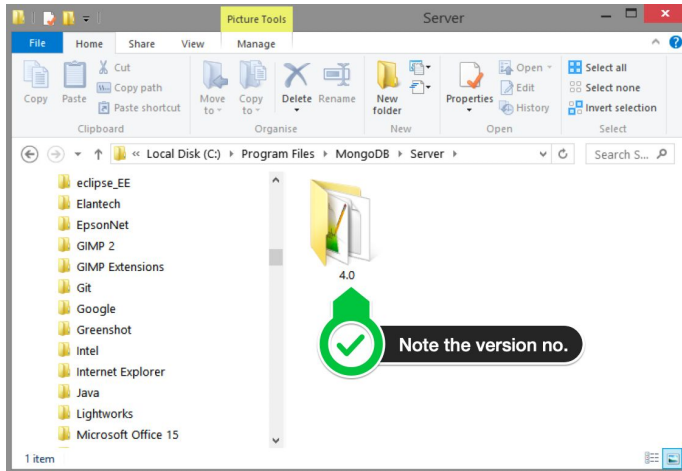
```
vim .bash_profile
```

E. In vim, hit the **I** key on the keyboard to enter insert mode.



F. In your explorer go to C → Program Files → MongoDB → Server

Now you should see the version of your MongoDB.



G. Paste in the following code into vim, make sure your replace the 4.0 with your version that you see in explorer

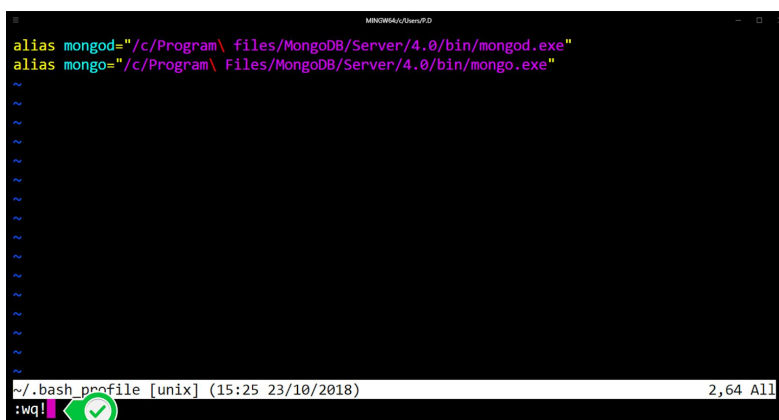
```
alias mongod="/c/Program\ files/MongoDB/Server/4.0/bin/mongod.exe"
```

```
alias mongo="/c/Program\ Files/MongoDB/Server/4.0/bin/mongo.exe"
```

F. Hit the Escape key on your keyboard to exit the insert mode. Then type

```
:wq!
```

to save and exit Vim.



## Step 5 — Verify That Setup was Successful

- A. Close down the current Hyper terminal and quit the application.
- B. Re-launch Hyper.
- C. Type the following commands into the Hyper terminal:

```
mongo --version
```

Once you've hit enter, you should see something like this:

```
P.D@Samsungnator MINGW64 ~  
$ mongo --version  
MongoDB shell version v4.0.3  
git version: 7ea530946fa7880364d88c8d8b6026bbc9ffa48c  
allocator: tcmalloc  
modules: none  
build environment:  
  distmod: 2008plus-ssl  
  distarch: x86_64  
  target_arch: x86_64  
  
P.D@Samsungnator MINGW64 ~  
$
```

This means that you have successfully installed and setup MongoDB on your local system!

**For Ubuntu use follow these steps [here](https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-18-04) or for further clarification use this link :**

<https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-18-04>

Ubuntu's official package repositories include an up-to-date version of MongoDB, which means we can install the necessary packages using apt.

First, update the packages list to have the most recent version of the repository listings:

```
sudo apt update
```

```
$ sudo apt update
```

Now install the MongoDB package itself:

```
sudo apt install -y mongodb
```

```
$ sudo apt install -y mongodb
```

This command installs several packages containing the latest stable version of MongoDB, along with helpful management tools for the MongoDB server. The database server is automatically started after installation.

Next, let's verify that the server is running and works correctly.

## Step 2 — Checking the Service and Database

The installation process started MongoDB automatically, but let's verify that the service is started and that the database is working.

First, check the service's status:

```
sudo systemctl status mongodb
```

```
$ sudo systemctl status mongodb
```

You'll see this output:

```
Output
● mongodb.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2018-05-26 07:48:04 UTC; 2min 17s ago
     Docs: man:mongod(1)
  Main PID: 2312 (mongod)
    Tasks: 23 (limit: 1153)
   CGroup: /system.slice/mongodb.service
           └─2312 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/mongodb.conf
```

According to systemd, the MongoDB server is up and running.

## Step 4 - Add necessary Data

Go to <http://localhost:3000/explorer/>

Step 1 - Post a company User :

Go to CompanyController:

CompanyController	
POST	/company-reject
POST	/company-submit
GET	/company-users/count
PATCH	/company-users/{id}
DELETE	/company-users/{id}
GET	/company-users/{userId}
POST	/company-users
GET	/company-users
GET	/company-users-names

Clicking on Post /company-users :

POST /company-users

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
{
  "id": "string",
  "name": "string",
  "email": "string",
  "password": "string",
  "specificTenderingProcessesEntered": [
    "string"
  ],
  "TenderingProcessesEntered": [
    "string"
  ],
  "specificTenderingProcessesAccepted": [
    "string"
  ],
  "TenderingProcessesAccepted": [
    "string"
  ]
}
```

Then click on “Try it out button”

POST /company-users

Parameters

No parameters

Request body

application/json

```
{
  "id": "string",
  "name": "string",
  "email": "string",
  "password": "string",
  "specificTenderingProcessesEntered": [
    "string"
  ],
  "TenderingProcessesEntered": [
    "string"
  ],
  "specificTenderingProcessesAccepted": [
    "string"
  ],
  "TenderingProcessesAccepted": [
    "string"
  ]
}
```

Execute

Add a company user in body : For Example:

```
{
  "name": "GE",
  "email": "GE@gmail.com",
  "password": "GECompany"
```

```
}
```

```
{  
  "name": "GE",  
  "email": "GE@gmail.com",  
  "password": "GECompany"  
}
```

Execute

And Click “Execute”.

Now a *new company* user has been added to your database. Let’s now add a new hospital user.

Step 1 - Post a hospital User

Go to HospitalUserController:

HospitalUserController

PATCH /hospital-users/{id}

DELETE /hospital-users/{id}

GET /hospital-users/{userId}

POST /hospital-users

GET /hospital-users

Clicking on Post /hospital-users :

POST /hospital-users

Parameters

Try it out

No parameters

Request body

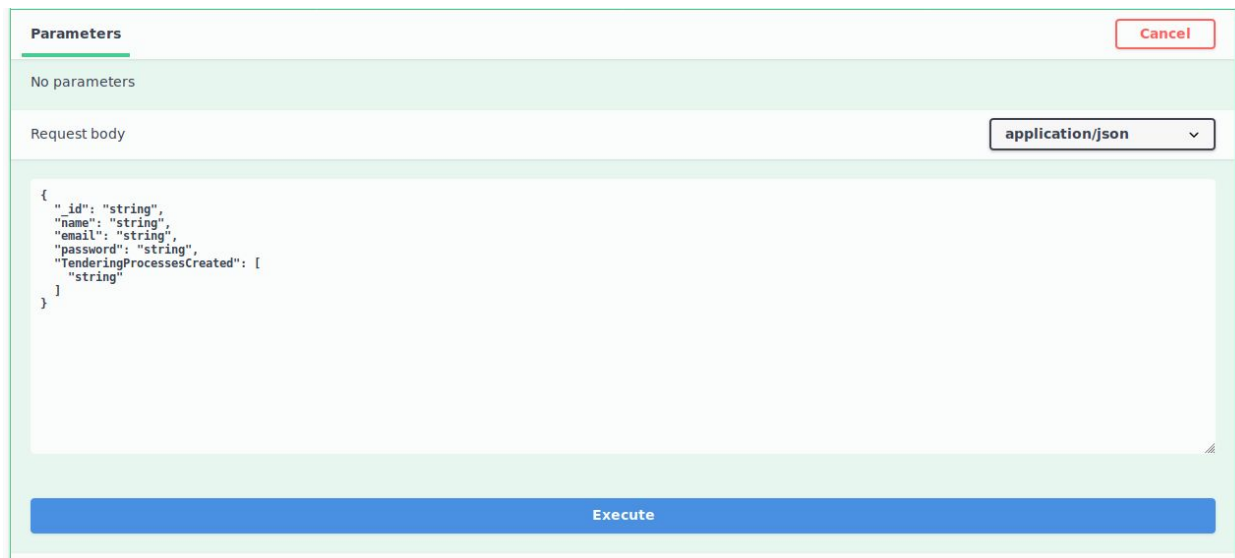
application/json

Example Value | Schema

```
{  
  "id": "string",  
  "name": "string",  
  "email": "string",  
  "password": "string",  
  "TenderingProcessesCreated": [  
    "string"  
  ]  
}
```



Then click on “Try it out button”



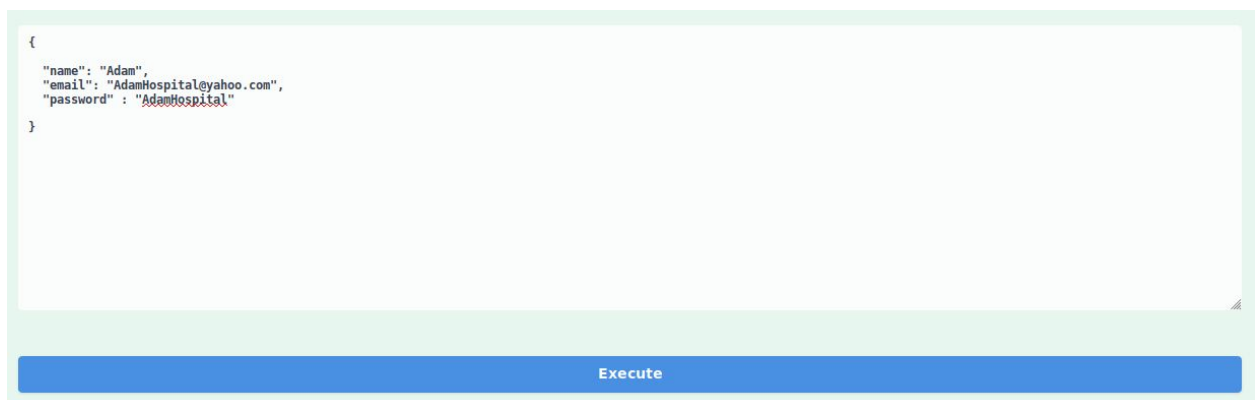
The screenshot shows a REST client interface. At the top, there is a 'Parameters' tab with a 'Cancel' button. Below it, a green bar indicates 'No parameters'. The 'Request body' section has a dropdown menu set to 'application/json'. The main area contains a JSON object: 

```
{  "id": "string",  "name": "string",  "email": "string",  "password": "string",  "TenderingProcessesCreated": [    "string"  ]}
```

 At the bottom, there is a large blue 'Execute' button.

Add a company user in body, For Example:

```
{  "_id": "5d83ecbe91fed14ee103b3ab",  "name": "Adam",  "email": "AdamHospital@yahoo.com",  "password": "AdamHospital"}
```



The screenshot shows the same REST client interface as before, but with the JSON body updated to the example provided: 

```
{  "name": "Adam",  "email": "AdamHospital@yahoo.com",  "password": "AdamHospital"}
```

 The 'Execute' button at the bottom is still visible.

And Click “Execute”. Now a new hospital user has been added to your database.

**Now you are good to go.**

Open <http://localhost:4200/login> and start navigating our project.

**P.S:**

You can add more hospital and company users as you like for better enjoying the project.