1 Preparing the local environment

The web application is well-suited to be deployed and run on a server. However, there is always a way to run it locally on any device to use it on a small scale for testing and development.

1.1 Setting up the required software

There is a set of software that are required to run the application locally on any machine

1.1.1 Node.js

The web application is based entirely based on the Node.js framework. So, we need to install the LTS (life-time support) version of node.js: https://nodejs.org/en/

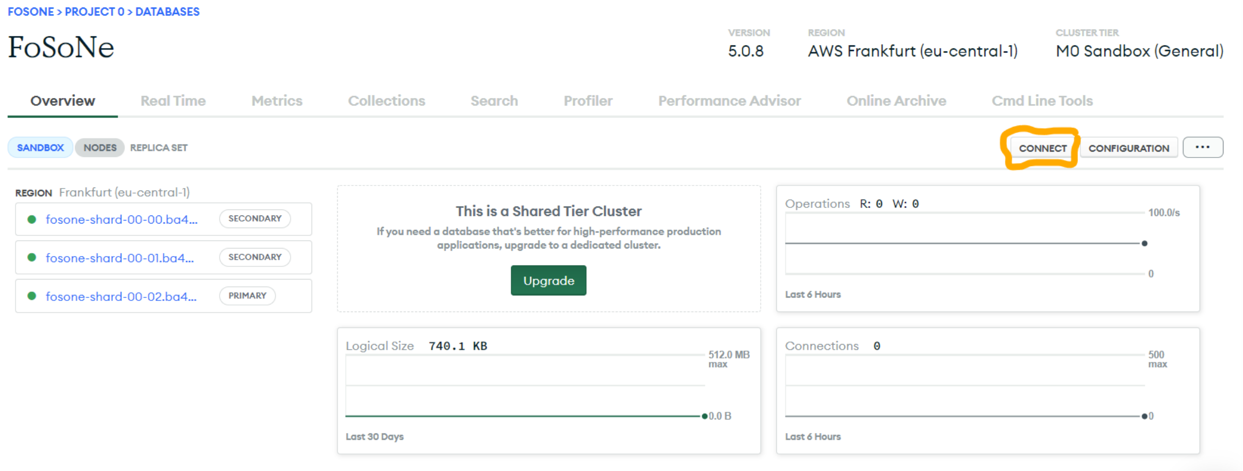
1.1.2 MongoDB

RedPlus uses a noSQL database called MongoDB which stores the information in documents of json format. You can use MongoDB either on an online hosting or locally on your machine.

1.1.2.1 Online choice

MongoDB can be used online by creating an account on any MongoDB online hosting service and creating a free cluster. Also create a database user and save its credentials for later use.

After creating a cluster, choose connect (this example is on MongoDB atlas hosting)



FOSONE > PROJECT 0 > DATABASES

FoSoNe

VERSION 5.0.8

REGION AWS Frankfurt (eu-central-1)

CLUSTER TIER M0 Sandbox (General)

REGION Frankfurt (eu-central-1)

● fosone-shard-00-00.ba4...	SECONDARY
● fosone-shard-00-01.ba4...	SECONDARY
● fosone-shard-00-02.ba4...	PRIMARY

This is a Shared Tier Cluster

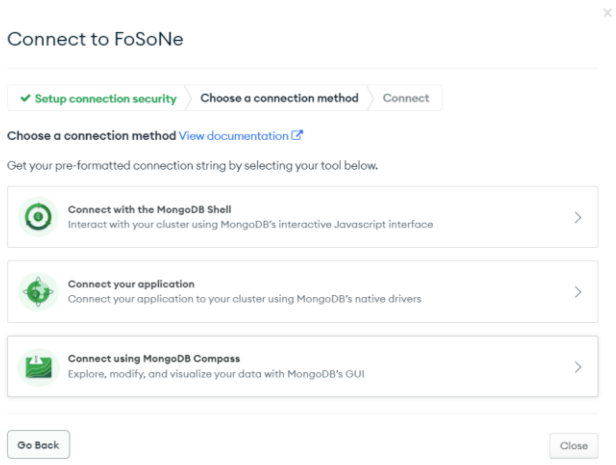
If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.

Operations R: 0 W: 0

Logical Size 740.1 KB

Connections 0

Connect your application



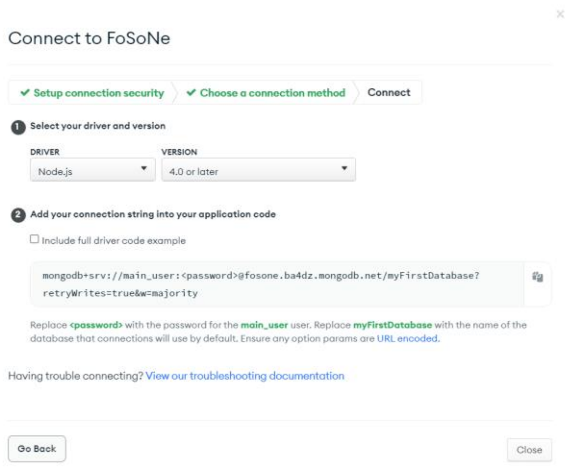
Connect to FoSoNe

✓ Setup connection security | Choose a connection method | Connect

Choose a connection method View documentation

Get your pre-formatted connection string by selecting your tool below.

- **Connect with the MongoDB Shell** Interact with your cluster using MongoDB's interactive Javascript interface
- **Connect your application** Connect your application to your cluster using MongoDB's native drivers
- **Connect using MongoDB Compass** Explore, modify, and visualize your data with MongoDB's GUI



Connect to FoSoNe

✓ Setup connection security | ✓ Choose a connection method | Connect

1 **Select your driver and version**

DRIVER: Node.js

VERSION: 4.0 or later

2 **Add your connection string into your application code**

□ Include full driver code example

```
mongodb+srv://main_user:<password>@fosone.ba4dz.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **main_user** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Copy the link and change ***main_user*** to the user name of your account, change ***<password>*** to your password, and change ***myFirstDatabase*** to the name of the database that you wish to use in RedPlus

Paste the final URI in **ATLAS_URI** variable in the **.env.example** file, and change the file name to **.env**

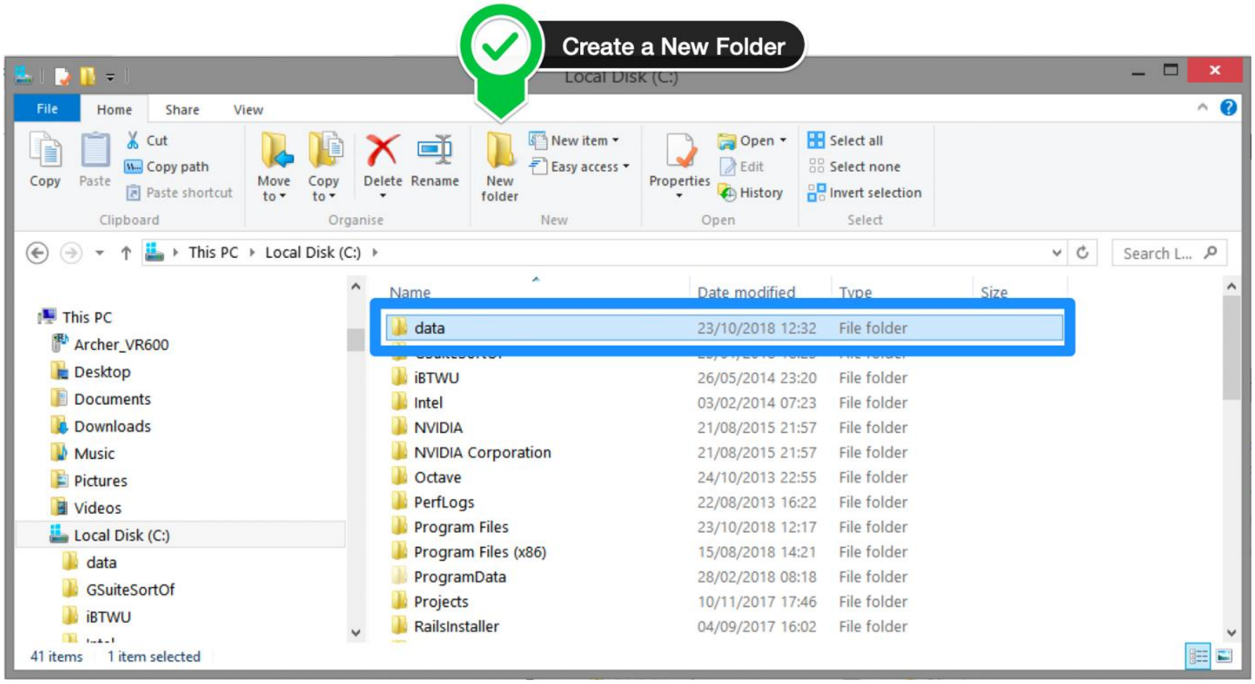
1.1.2.2 Local choice

Local MongoDB Installation can be found here: https://www.mongodb.com/try/download/community

Git bash is also required to correctly follow the following steps. You can install it by installing the GIT (version controlling system) software: https://git-scm.com/downloads

The next page embeds a guide copied from How to Download & Install MongoDB on Windows | by London App Brewery | Medium that we followed during our local database installation.

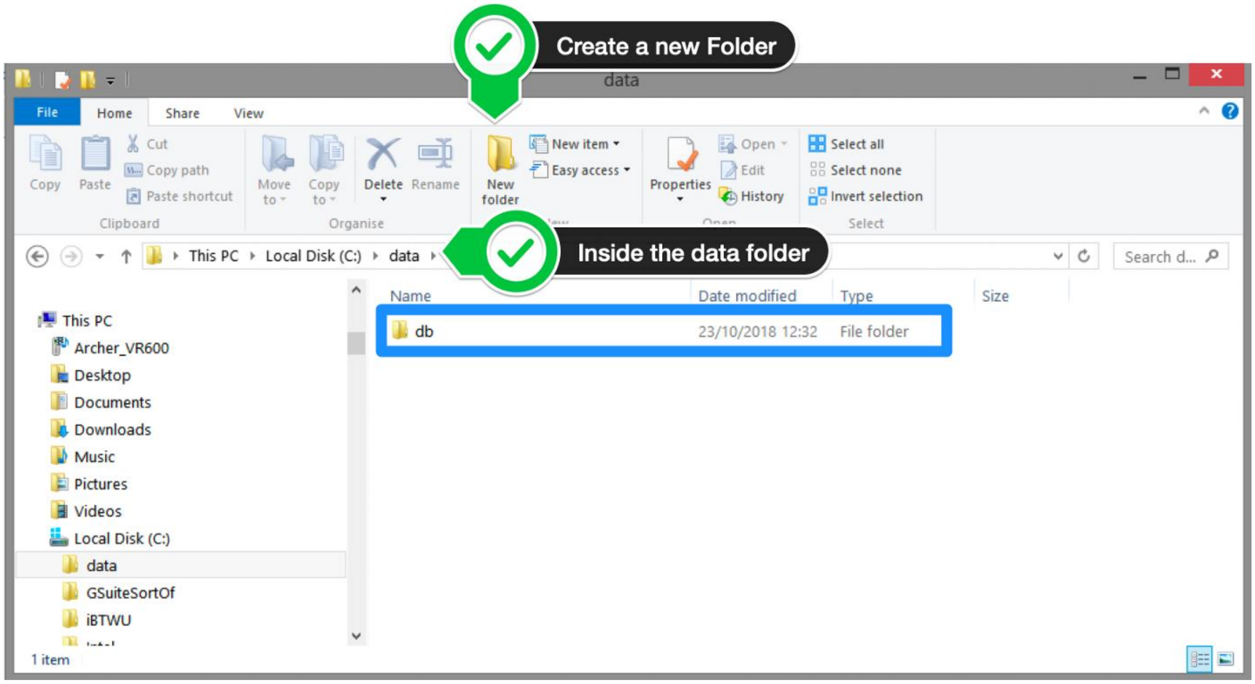
A. Navigate to the **C Drive** on your computer using Explorer and create a new folder called **data** here.



Create a New Folder

Name	Date modified	Type	Size
data	23/10/2018 12:32	File folder	
iBTWU	26/05/2014 23:20	File folder	
Intel	03/02/2014 07:23	File folder	
NVIDIA	21/08/2015 21:57	File folder	
NVIDIA Corporation	21/08/2015 21:57	File folder	
Octave	24/10/2013 22:55	File folder	
PerfLogs	22/08/2013 16:22	File folder	
Program Files	23/10/2018 12:17	File folder	
Program Files (x86)	15/08/2018 14:21	File folder	
ProgramData	28/02/2018 08:18	File folder	
Projects	10/11/2017 17:46	File folder	
RailsInstaller	04/09/2017 16:02	File folder	

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



Create a new Folder

Inside the data folder

Name	Date modified	Type	Size
db	23/10/2018 12:32	File folder	

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 ~
```

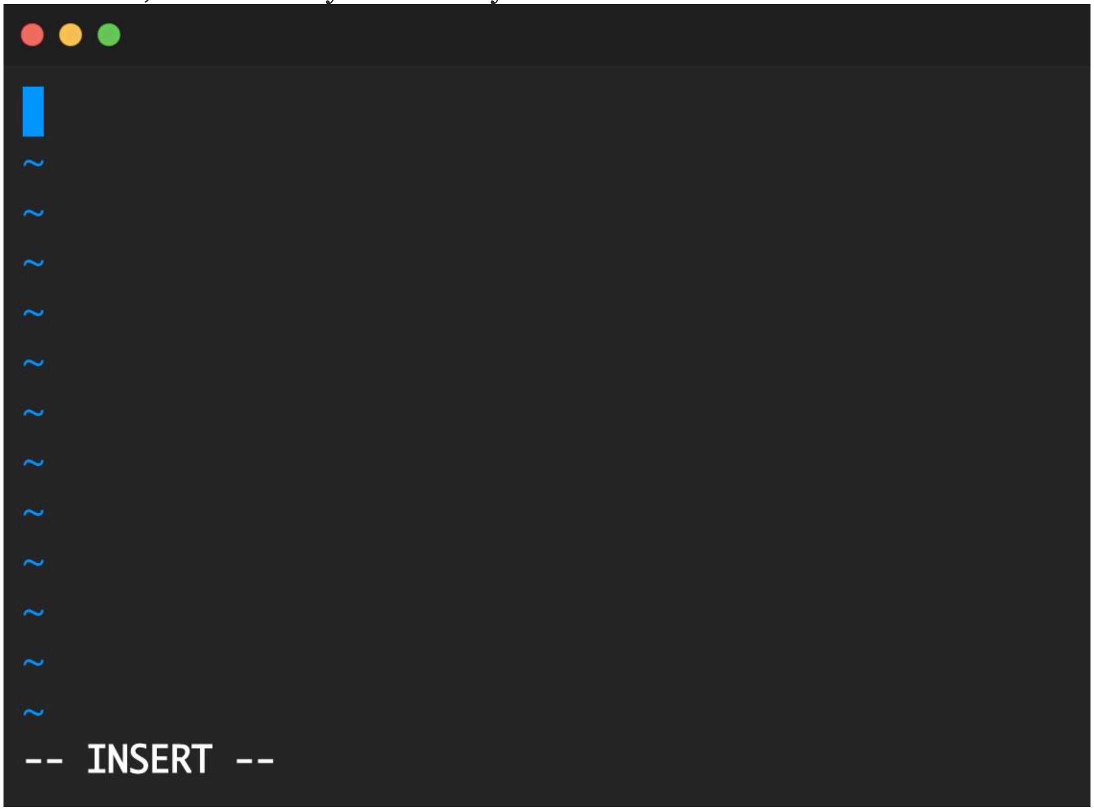
C. Here, we’re going to create a file called .bash_profile using the following command:

```
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.

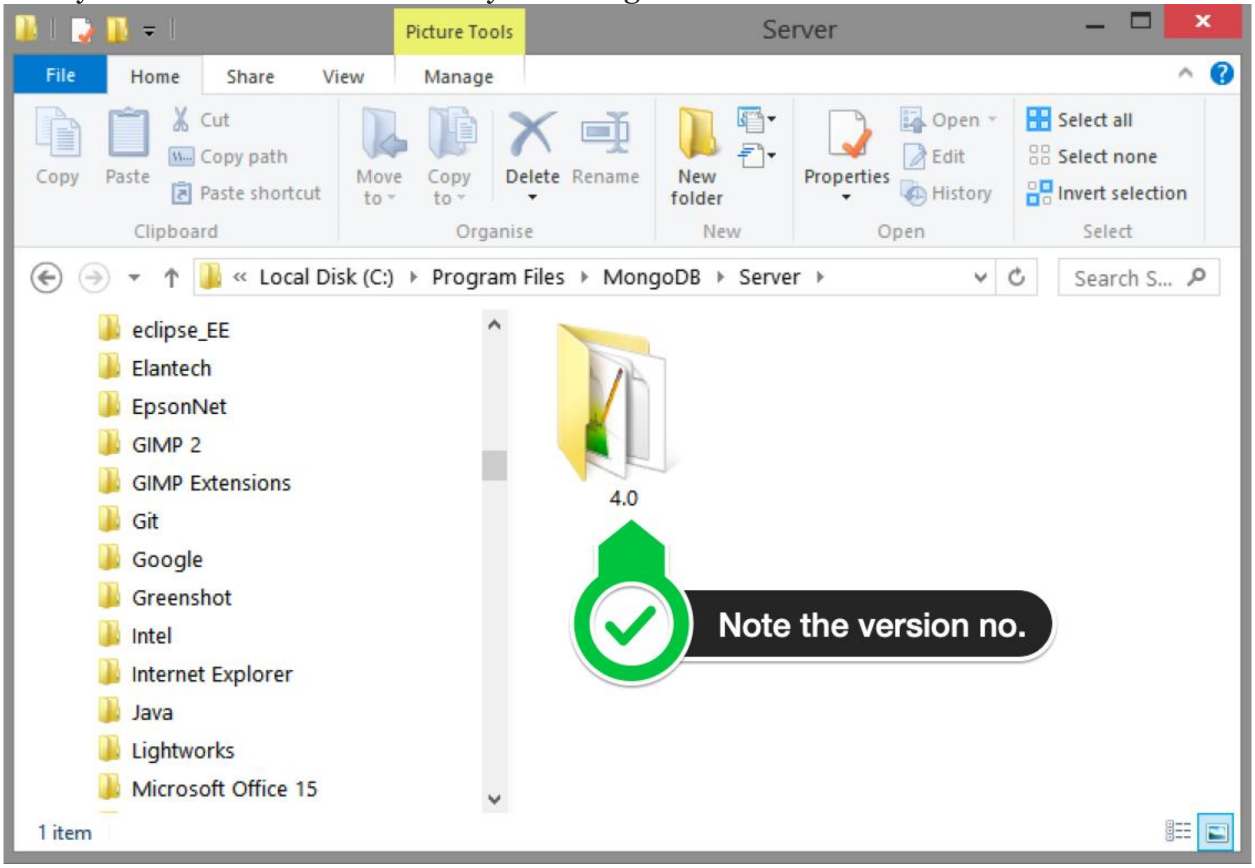


```

~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

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

Now you should see the version of your MongoDB.



Note the version no.

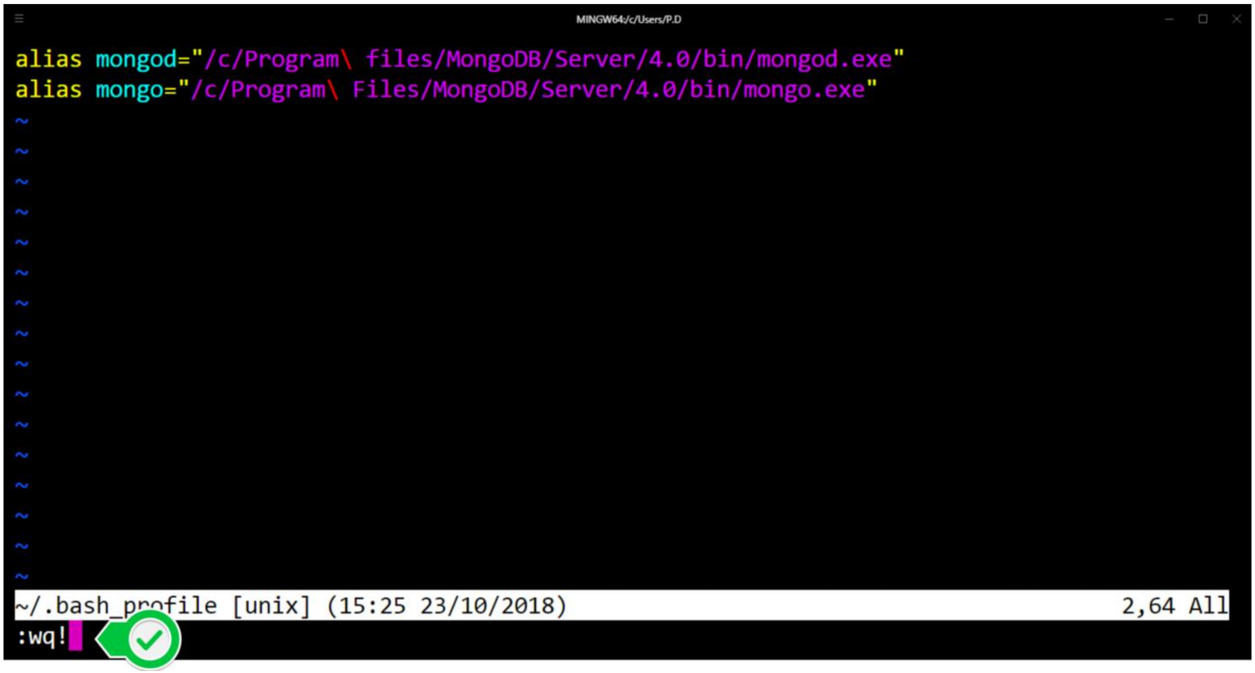
G. Paste in the following code into vim, make sure your replace the 4.0 with your version that you see in explorer

```
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. (screenshot next page)



```
alias mongod="/c/Program\ files/MongoDB/Server/4.0/bin/mongod.exe"
alias mongo="/c/Program\ Files/MongoDB/Server/4.0/bin/mongo.exe"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~/.bash_profile [unix] (15:25 23/10/2018)                   2,64 All
:wq!
```

Verify That Setup was Successful

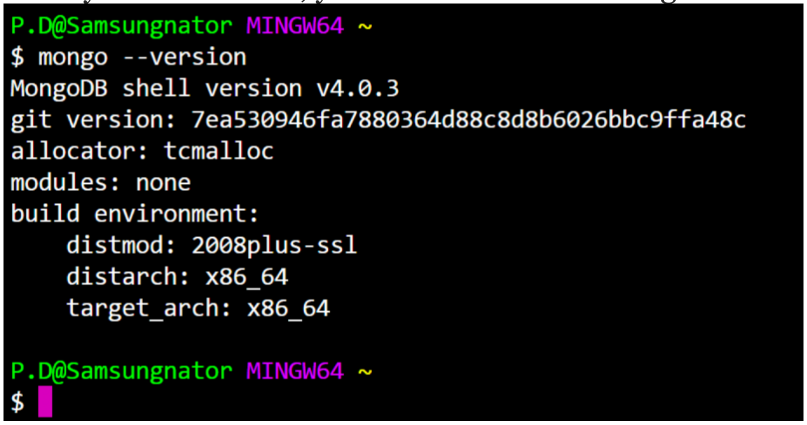
A. Close down the current Hyper terminal and quit the application.

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!

If you see something that looks like bash mongo command not found, then make sure you check back at all the steps above and follow it step-by-step making sure there are no typos and you haven't missed any of the steps.

1.2 Setting up the app with dependencies

After unzipping the files open the terminal in the folder where you can find package.json file (the main folder)

1.2.1 Setup the required dependencies

RedPlus is build using some main and development dependencies. All of them are defined with their required versions in the packages.json file. You can install them by running

```
npm install && cd frontend && npm install && cd ..
```

in the terminal.

1.2.2 Change environment variables

The first step in changing the environment variables was in adding the database URI. This second step involves changing the **JWT_SECRET** key which is the signature for encrypting Json Web Tokens that are required in the authentication process. You can change it or keep it as it is.

After making sure the environment variables are set correctly. Change the file name to **.env** if you have not already changed it.

2 Running the application

2.1 The whole application

There is a command for running the front-end (in dev mode) and back-end (in dev mode) concurrently.

You can start the application by running (in the terminal)

```
npm run dev
```

2.2 Front-end

You can start the front-end only (dev mode) by running (in the terminal)

```
npm run client
```

You can build the front-end (although you will not need this in a local environment) by running:

```
cd frontend && npm run build && cd ..
```

2.3 Back-end

You can run the back-end only by running (in the terminal) for watch mode:

```
npm run server
```

Or, for production mode

```
npm run start
```

The application will be run by default on http://localhost:3000 (if you have not changed the port)