All the files required to follow these instructions are in the /Server directory, except for Node.js and PostgreSQL installations on the testing device.

Instructions for how to download and install these softwares are included. **All the images accompanying the instructions are from a computer running Windows 7 64-bit**

**Instructions:**

**SECTION 1: Setting up the PostgreSQL local server**

**You will need a local installation of PostgreSQL v9.6.1 or above.**

If you already have the appropriate version of PostgreSQL installed, skip to step 2.

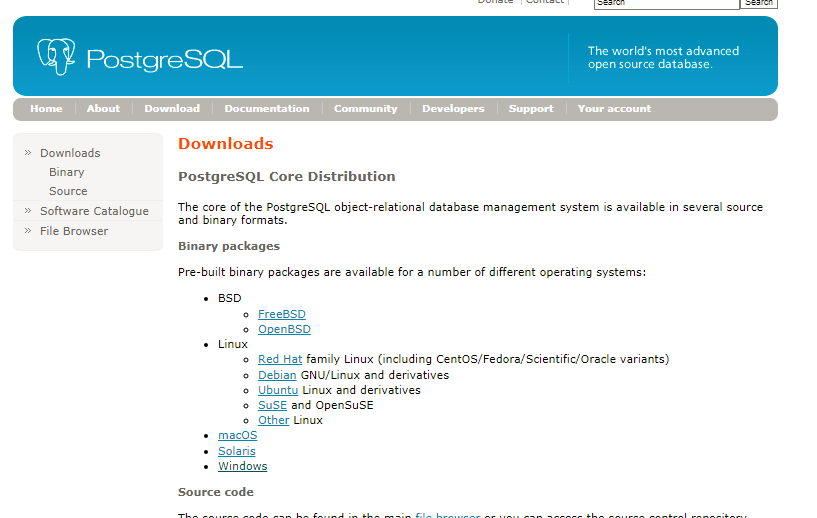
Otherwise, follow the instructions below to download and install PostgreSQL.

**1.1 Downloading PostgreSQL**

Go to <https://www.postgresql.org/download/>

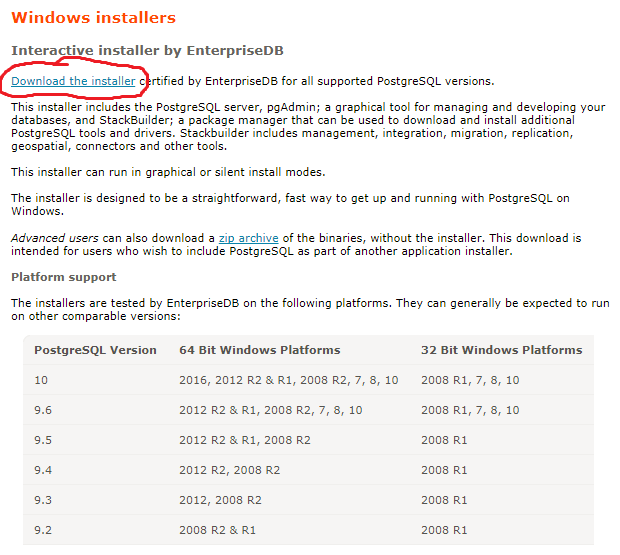
Select the appropriate link for your OS. The instructions below assume Windows.

If you are using a different OS, simply follow the instructions on the website.

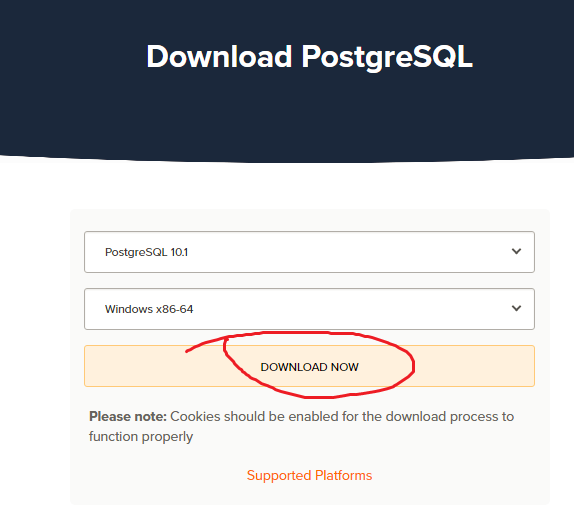


Download the EnterpriseDB Interactive installer. This will install two things that we want:  
The PostgreSQL server, which will hold our databases, and

The pgAdmin graphical tool, which makes it very easy to run queries on the database.



Select the PostgreSQL version you want (latest is 10.1, this will work with our app), and your OS version (here, we assume Windows 64 bit). Then click “DOWNLOAD NOW”



The download will start. Wait for it to complete, then run the executable file. (If any dialog boxes pop up asking you to confirm, click “Yes”)

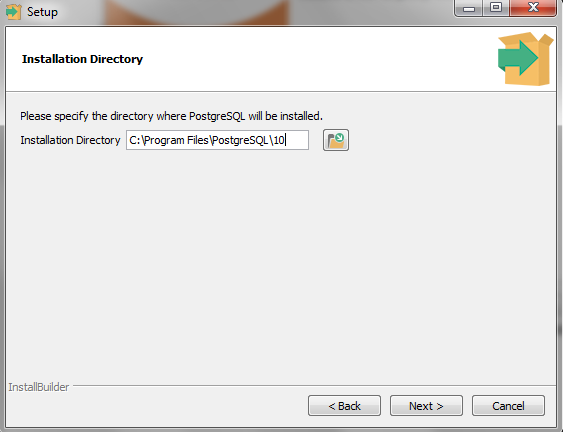
Downloading the installer is complete, now comes the installation phase.

**1.2 Installing PostgreSQL**

This will start the EnterpriseDB installer. Click “Next”

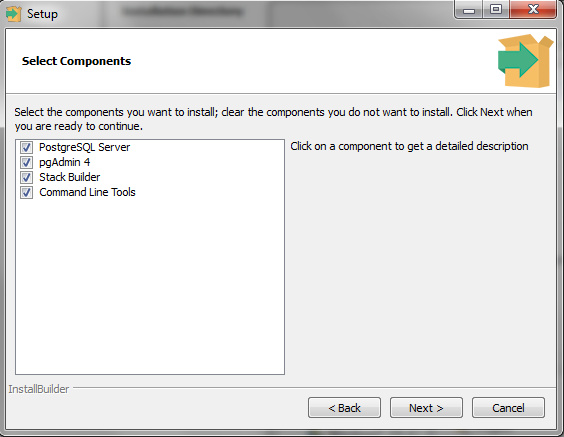


Choose the installation directory for PostgreSQL, then click next.

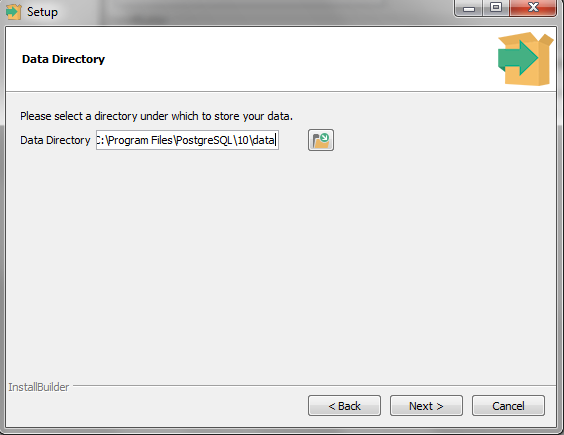


Now you must select which components to install. We will only need the first two: **PostgreSQL Server** and **pgAdmin 4**, but you can install Stack Builder and the Command Line Tools as well, if you wish.

After you select what to install, click next.



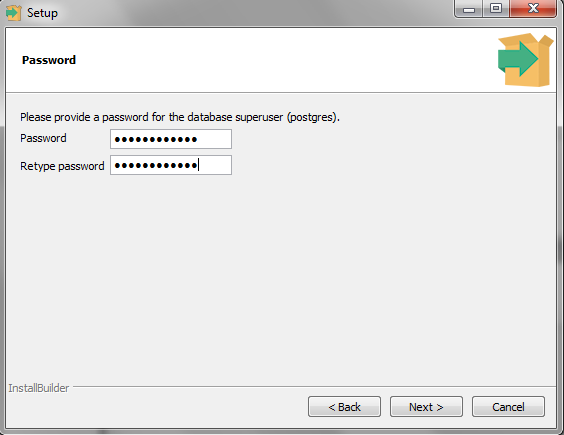
Now choose the data directory. This is the directory where your databases, schemas, tables, etc. are stored. By default, this will be <installation directory>/data. There’s no real reason to change it from the default, so keep it, and click next.



Now you must create a password for the superuser, which is called **postgres**.

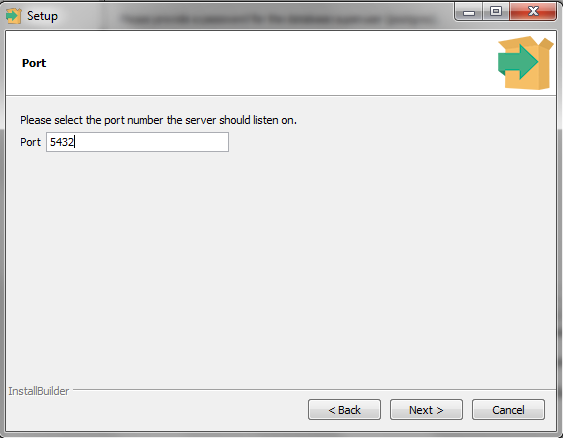
**YOU MUST REMEMBER THIS PASSWORD**.

The app code connects to the database using the superuser username and password.



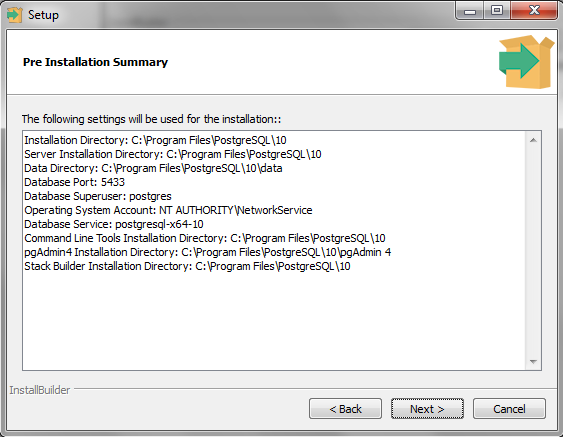
Now, you must select which port to host the server on. By default, the port number is 5432.

We will keep the default value. Only change it if you have something else listening on that port number.

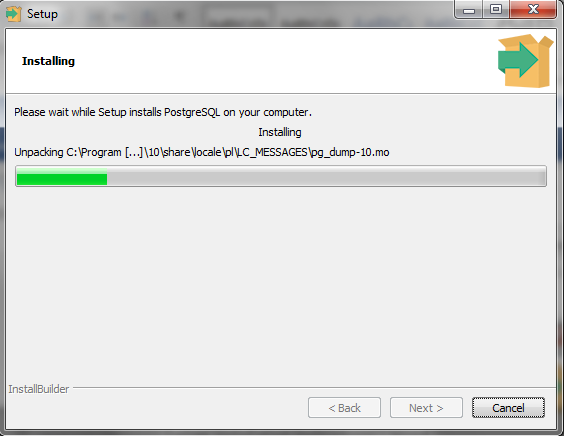


Next, it will ask you to select the Locale. Leave this as the default value, and click **next**.

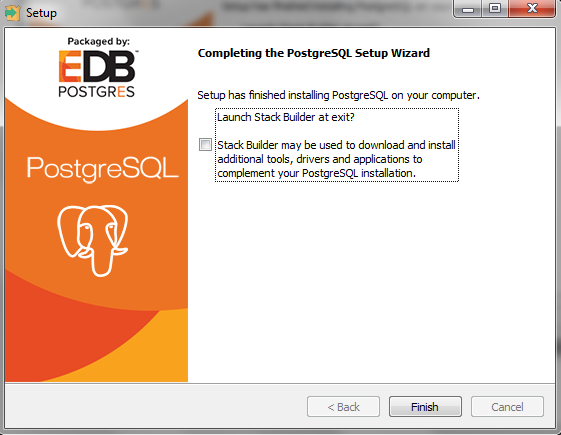
The next page shows a summary of the installation plan. Review it and, if satisfied, click **next**. (**NOTE** that in this picture, my port number is 5433, not the default 5432. That’s because I have PostgreSQL 9.6.1 installed, which is already using port 5432)



Click **next** again, and the installation will begin. Wait for it to finish.



Once the installation finishes, if you had selected Stack Builder in the installation options, it will prompt if you want to open Stack Builder. We don’t need to use Stack Builder right now, so uncheck the box, and click finish.



Now we are done installing PostgreSQL.

**1.3** **Open pgAdmin 4**

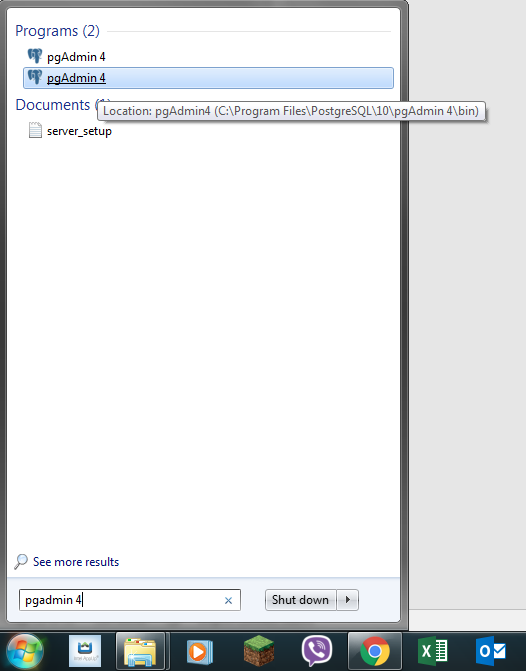
In order to execute the SQL instructions in the ./Server/DB\_Scripts folder, we have to open pgAdmin 4.

(Optionally, you could simply use the command line tool **psql** to execute all the statements, but the GUI of pg Admin 4 is MUCH easier to use)

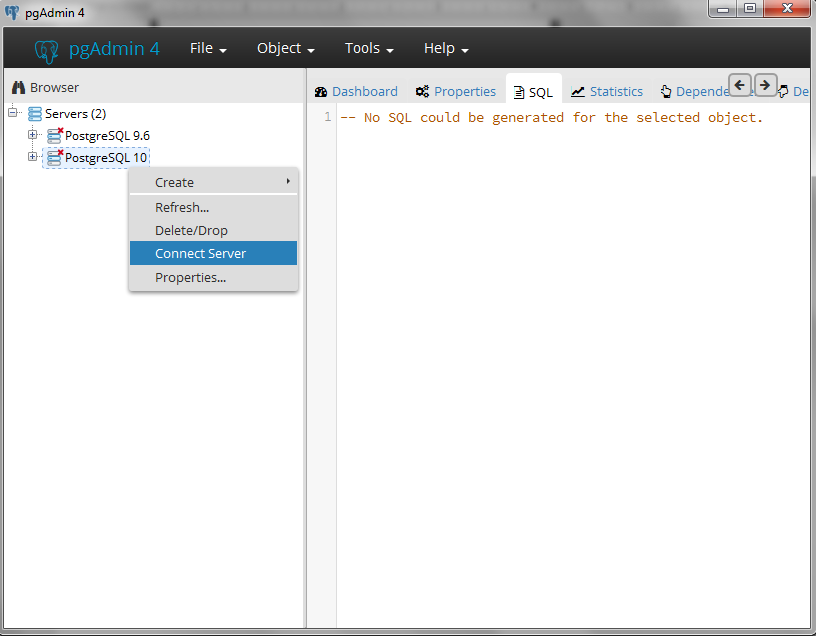
Search for “pgAdmin 4” in the start menu.

(**NOTE:** in the picture, two search results show up; this is because one of them came with my prior PostgreSQL 9.6 installation. If you have multiple installations, simply hover over the icon and check that the program’s directory is the latest installation directory. You most likely won’t face this issue, as you will likely only have a single installation of PostgreSQL)

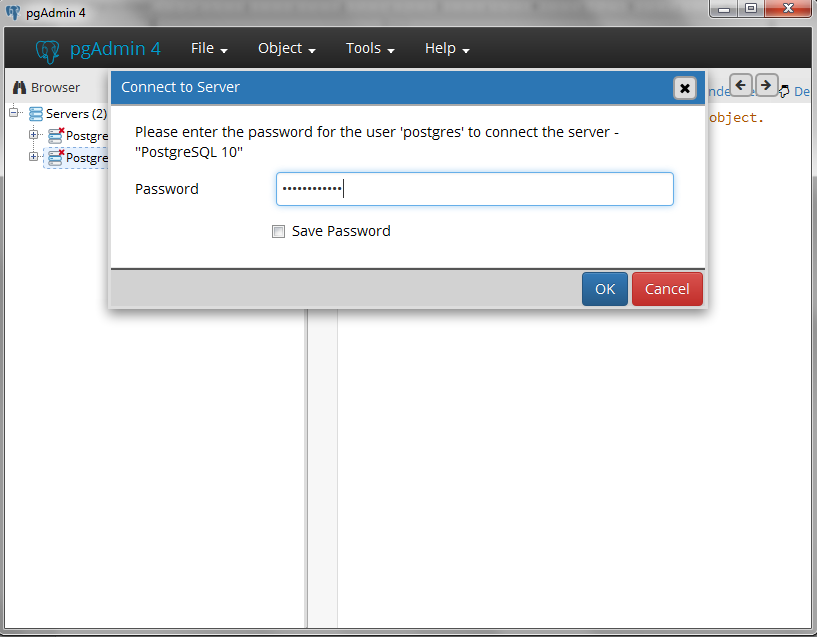
**Click** on the program to run pgAdmin 4.



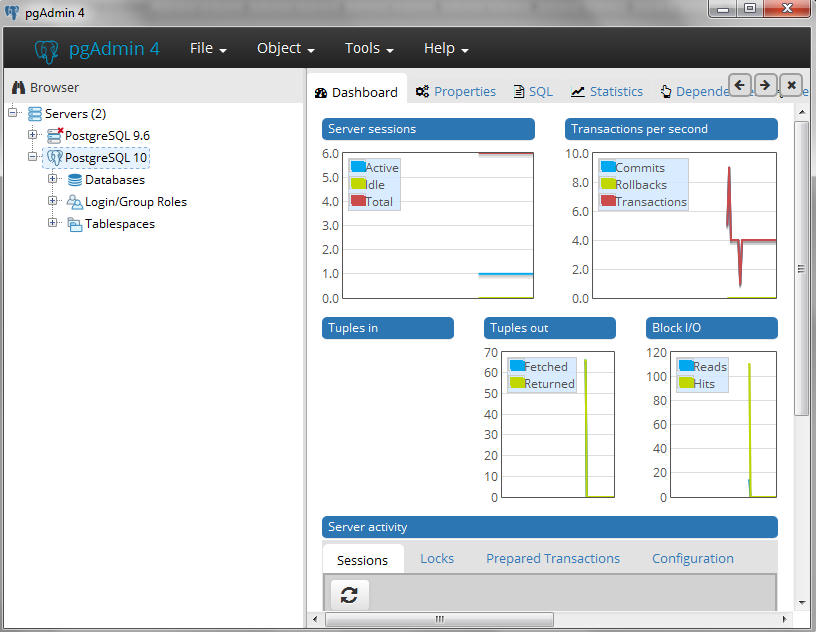
The pgAdmin 4 GUI will show up. On the left side of the screen, click the “+” icon next to “Servers” to view the available servers. Right click on the server you want to connect to, and choose **Connect Server**.



It will ask you for the password for user ‘postgres’. This is the password you set during the installation process. Enter it, and click **OK**.



If successful, pgAdmin 4 will connect to the server, and we are ready to run queries.



Now, our database is fully set up, we just need to have our app connect to it, and of course, it needs to have the appropriate tables and functions.

In order to set up the server code, see **Section 2.**

In order to populate the database with the app’s tables and functions, see **Section 4**

**NOTE:** don’t close pgAdmin 4, we will use it in **Section 4**.

**SECTION 2: Setting up the Node.js backend server**

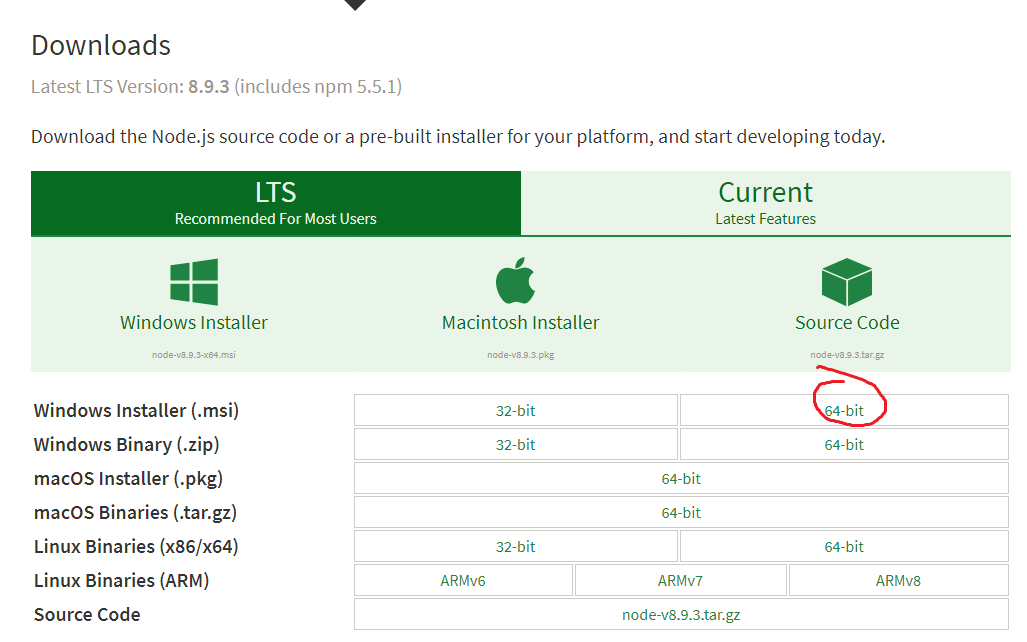
**You will need to download and install Node.js and npm, the node package manager.**

**2.1 Downloading Node.js v6.9.4 or above and npm v3.10.10 or above.**

**If you already have the appropriate versions installed, you can skip to Section 3**

Go to <https://nodejs.org/en/download/>

We’re downloading the Windows 64 bit installer, choose whichever installer is appropriate for your OS. **NOTE:** this installer includes the latest npm version, so you don’t need to get that separately.

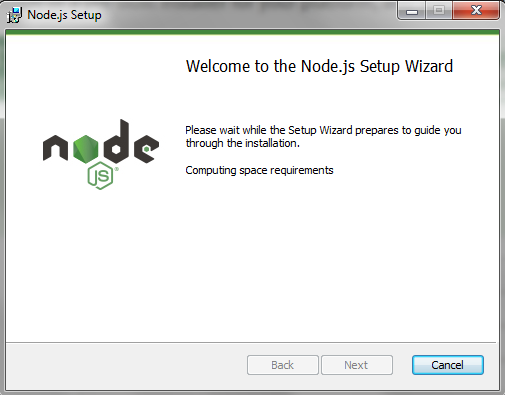


The download is now complete.

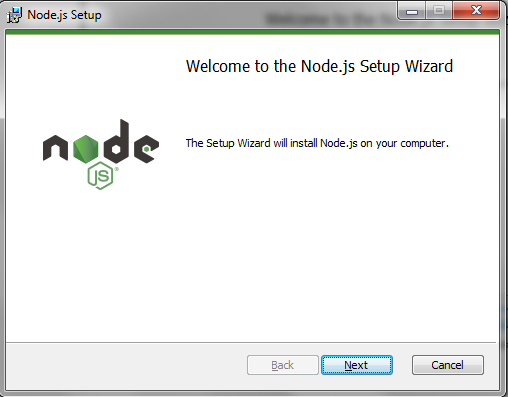
**2.2 Installing Node.js and npm**

Open the installer.

Wait for the wizard to finish computing space requirements,



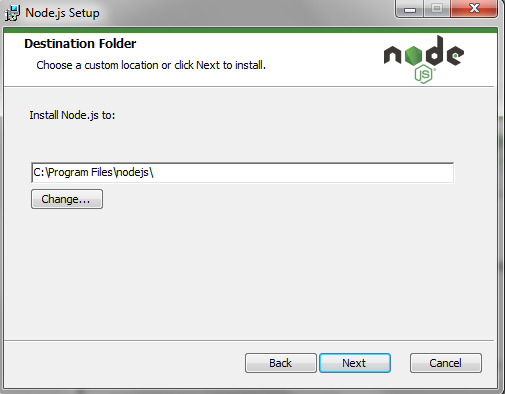
Then click **next**



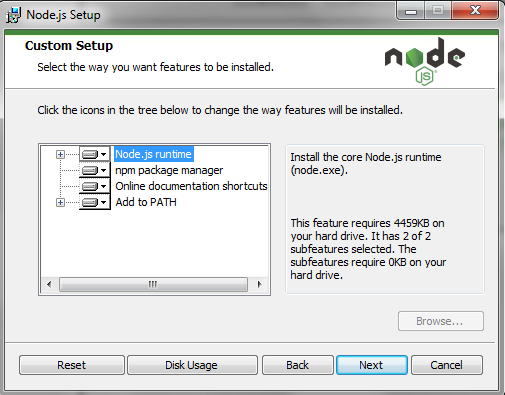
Accept the terms in the License Agreement, and click next



Choose the Node.js installation directory. By default, it goes in C:\Program Files\nodejs\

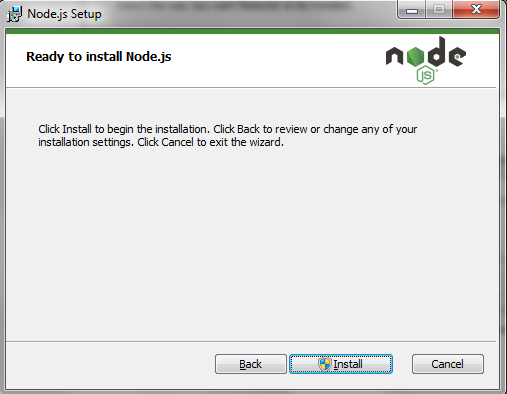


Now choose what will be installed. **NOTE:** everything but **online documentation shortcuts** is **REQUIRED**.

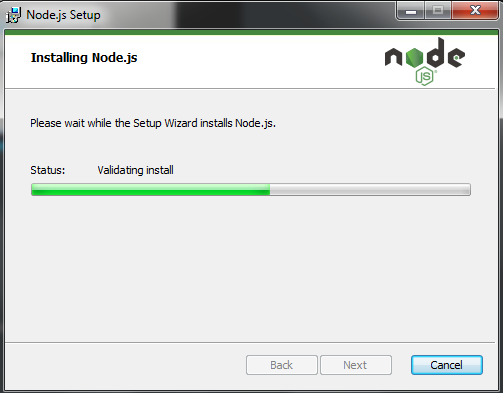


Click **install**.

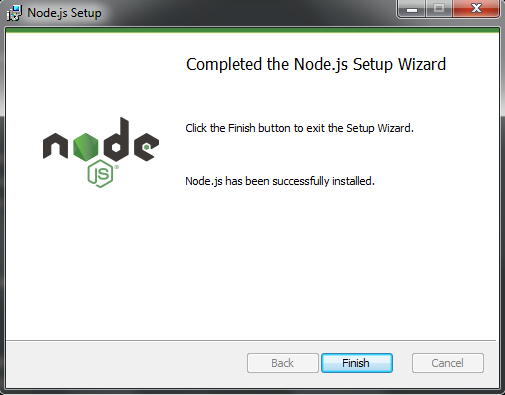
If Windows pops up a dialog box asking permission, **allow** it.



Wait for installation to finish



After installation finishes, click **finish** to close the installer.



Now we have successfully installed Node.js and npm.

If you haven’t installed PostgreSQL yet, go to **Section 1**.

If you have completed both **Section 1 (PostgreSQL)** and **Section 2 (Node and npm)**, Go to **Section 3**.

**SECTION 3: Setting up the server**

This section contains a few things we have to take care of before the server code will work properly.

**3.1 Setting up system variables**

**NOTE:** these instructions assume the tester is familiar with creating system environment variables. If not, there are simple instructions that can be searched on Google.

For Windows, system variables are created by going to **Control Panel -> System and Security -> System**, click on **advanced system settings**, click on **Environment Variables**, and under System Variables, click “New”, then put in the appropriate variable name and value.

**The app requires two system variables to be set up:**

**PORT:** 3000

Right now the client side of the app is hard coded to use <http://localhost:3000> for testing purposes. **If you choose a port number other than 3000**, you will have to change the BASE\_URL string in the RequestString.java class.

**DATABASE\_URL:** A string of the format: postgres://postgres:<your\_password>@localhost:<pg\_port>/postgres

<your\_password> is the password you set for user ‘postgres’ during installation, and <pg\_port> is the port number you set during PostgreSQL installation (default is 5432).

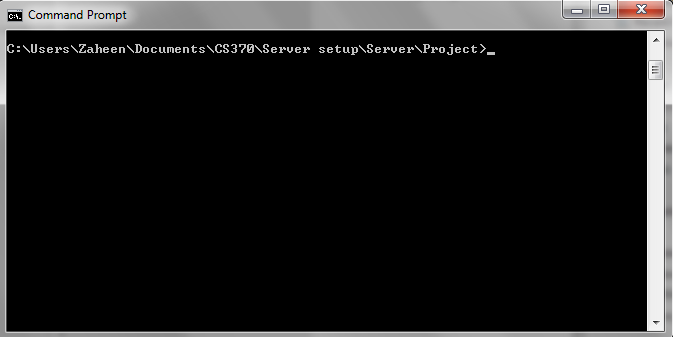
For example:

postgres://postgres:pass1234@localhost:5432/postgres

**3.2 Fetching Node dependencies**

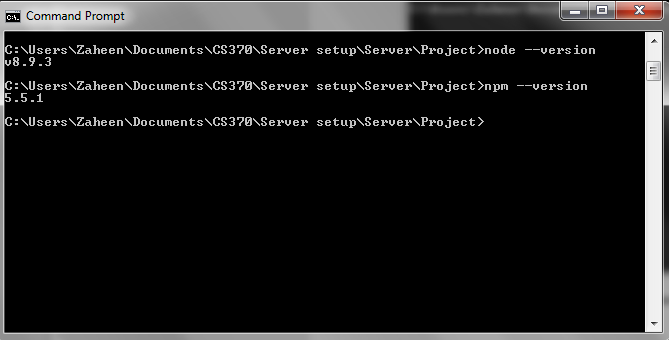
Open up a command line and cd into the ./Server/Project directory.

(**NOTE:** the “C:\Users\Zaheen\Documents\CS370” part is specific to my machine, you will have the Server\_Setup folder wherever you decided to unzip it)

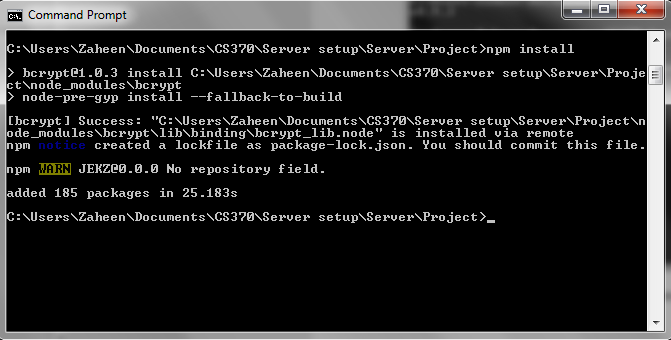


Here, you can test your Node and npm installations by running the following commands:

**node --version** and **npm --version**



Now we will fetch the Node dependencies, which are listed in the Package.json file. Run the command **npm install**



Now, the dependencies required for the server to work have been imported.

If you haven’t created the database tables and functions yet, go to **Section 4**.

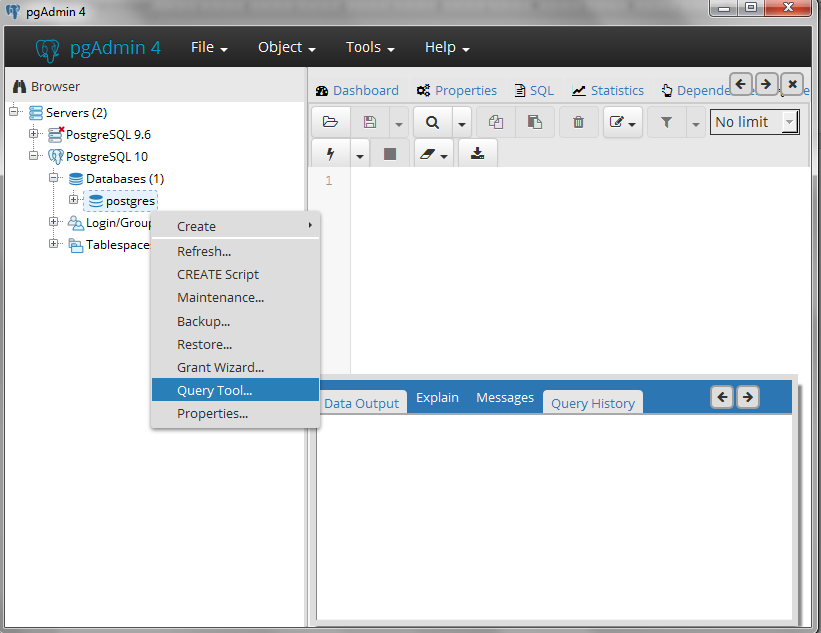
If you’ve finished Sections 1-4, go to **Section 5** to start the server. (Don’t close this command terminal, we will use it in **Section 5**).

**SECTION 4: Setting up the database tables and functions**

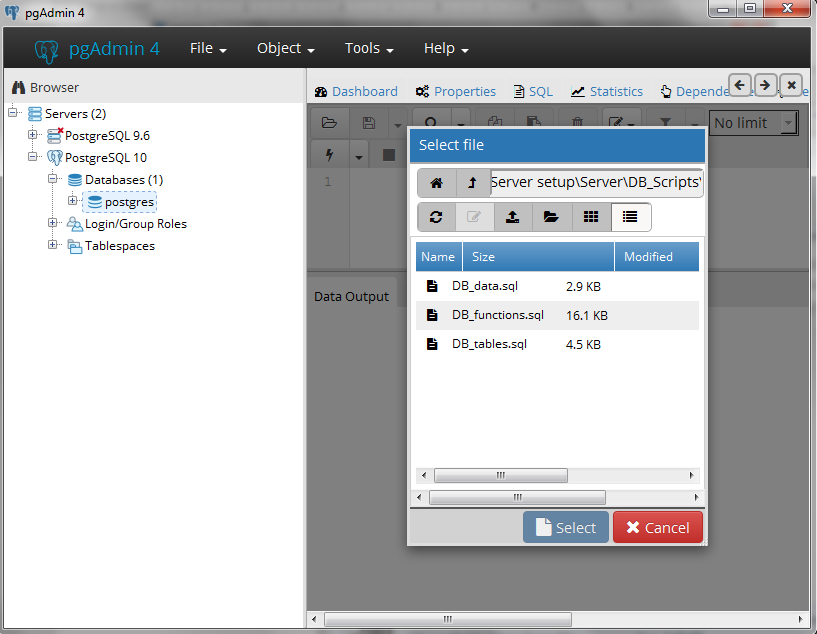
This section will walk you through running the queries in the ./Server/DB\_Scripts folder, which will create the required tables and functions for the app, and populate the tables with some test data.

Open pgAdmin 4 and connect to the PostgreSQL server, following the instructions in **Section 1**.

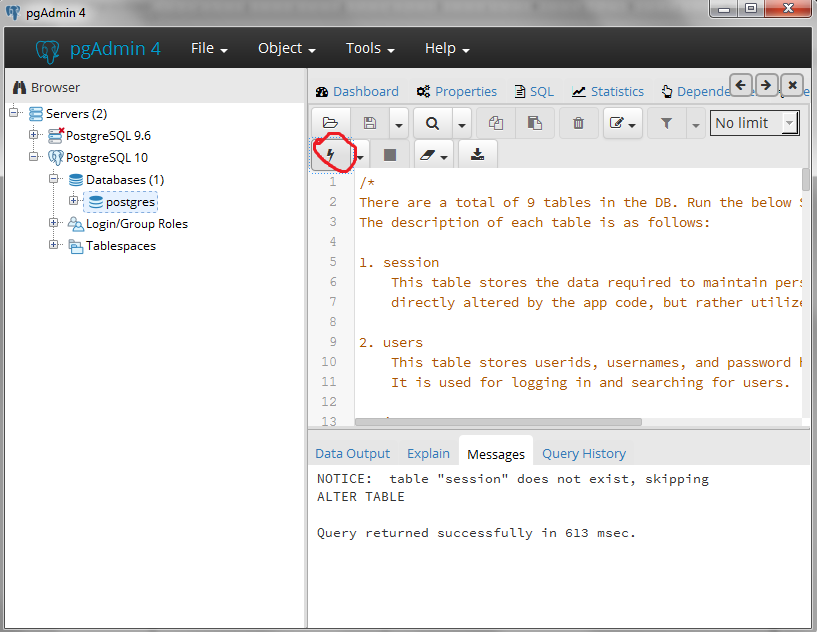
Go to **Servers -> PostgreSQL 10 -> Databases -> postgres**, right click on postgres, and select **Query Tool**.



In the query tool window on the right, click on the **Open File** icon in the top left, and navigate to the DB\_Scripts folder.



Open the **DB\_tables.sql** file, and click the **execute** button, or press F5.



If successful, you should be able to see the tables under **postgres -> Schemas -> public -> Tables**

Then, **open** and **execute** the other two files, **DB\_functions.sql** and **DB\_data.sql**, **IN THAT ORDER**.

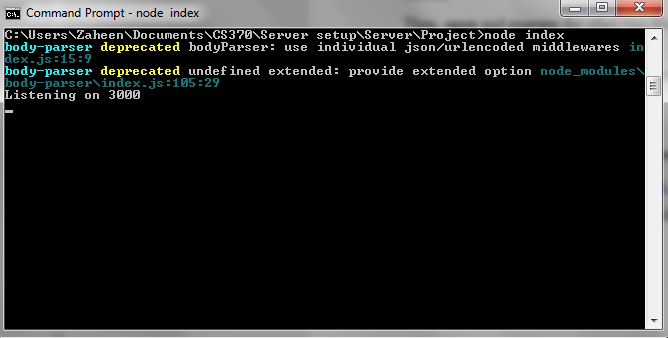
Now, the database is fully ready to accept queries from the server, and has a bit of test data to work with.

Move on to **Section 5** to start the server.

**SECTION 5: Starting the local Node server**

Using the command terminal, cd into the ./Server/Project folder, just as you did in **Section 3**.

Run the command **node index**, which will start the server at localhost:3000



**And that’s it!** Now you have the server running on your local machine, and you can test it from Android Studio running the client app.

In order to set up the **client app**, see the **Client\_Setup** folder, in the same directory you found the **Server\_Setup** folder (this one).

Once you have the client side set up, you can play around with the test data in the DB, outlined in **Section 6**.

**SECTION 6: Database Test Data**

This is just an outline, if you want to see exactly what test data is generated for the DB, take a look at the DB\_data.sql file.

**Users:**

The DB\_data scripts automatically generate 3 test users, with usernames: **user1**, **user2**, and **user3**

All have password: **test**

Each user has varying user\_data, owned\_accessories, and friends. These can be checked by using the appropriate functionality on the client app.

**session\_data:**

Each of the test users is given 2000 steps per day for the past 7 days, including the current day.