

## Project 3

### Prepare the backend Linux server

#### Update ubuntu

```
emmanuel@webserver:~$ sudo apt-get update
[sudo] password for emmanuel:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [263 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [205 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.2 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [56.5 kB]
Fetched 877 kB in 2s (475 kB/s)
```

#### Upgrade ubuntu

```
emmanuel@webserver:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  alsu-ucm-conf alsu-utils apport apport-gtk base-files bind9-dnsutils
  bind9-host bind9-libs bolt bsutils busybox-initramfs busybox-static cheese
  cheese-common command-not-found enchant-2 evolution-data-server
  evolution-data-server-common fdisk file-roller fonts-noto-mono
  fonts-opensymbol gdb gdbserver gir1.2-gnomebluetooth-1.0
  gir1.2-gnomedesktop-3.0 gir1.2-gweather-3.0 gir1.2-mutter-6 gir1.2-nm-1.0
  gir1.2-nma-1.0 gir1.2-rsvg-2.0 gir1.2-snapd-1 gnome-bluetooth
  gnome-control-center gnome-control-center-data gnome-control-center-faces
  gnome-desktop3-data gnome-initial-setup gnome-shell-extension-appindicator
  gnome-shell-extension-desktop-icons gnome-system-monitor im-config
  initramfs-tools initramfs-tools-bin initramfs-tools-core
  language-selector-common language-selector-gnome libasound2 libasound2-data
  libatopology2 libblkid1 libc-bin libc6 libc6-dbg libcamel-1.2-62
  libcheese-gtk25 libcheese8 libcryptsetup12 libdns-export1109
  libebbackend-1.2-10 libebook-1.2-20 libebook-contacts-1.2-3 libical-2.0-1
  libedata-book-1.2-26 libedata-cal-2.0-1 libedataserver-1.2-24
  libedataserverui-1.2-2 libefiboot1 libefivar1 libegl1 libenchant-2-2
  libevdev2 libfdisk1 libfprint-2-2 libfprint-2-tod1 libgl1 libgles2
  libglvnd0 libglx0 libgnome-bluetooth13 libgnome-desktop-3-19
  libgweather-3-16 libgweather-common libinput-bin libinput10
```

Get the location of nodejs software from ubuntu repositories over the internet.

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

```
emmanuel@webserver:~$ curl -sL https://deb.nodesource.com/setup_12.x | sudo -E
bash

Command 'curl' not found, but can be installed with:

sudo apt install curl

emmanuel@webserver:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 161 kB of archives.
After this operation, 411 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7
.68.0-1ubuntu2.4 [161 kB]
Fetched 161 kB in 0s (612 kB/s)
Selecting previously unselected package curl.
(Reading database ... 180245 files and directories currently installed.)
Preparing to unpack .../curl_7.68.0-1ubuntu2.4_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.4) ...
Setting up curl (7.68.0-1ubuntu2.4) ...
Processing triggers for man-db (2.9.1-1) ...
emmanuel@webserver:~$ *
```

I had to install curl using **sudo apt install curl** as shown above

```
emmanuel@webserver:~$ curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash

## Installing the NodeSource Node.js 12.x repo...

## Populating apt-get cache...

+ apt-get update
0% [Working]
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done

## Confirming "focal" is supported...

+ curl -sLF -o /dev/null 'https://deb.nodesource.com/node_12.x/dists/focal/Release'
## Adding the NodeSource signing key to your keyring...

+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | apt-key add -
OK

## Creating apt sources list file for the NodeSource Node.js 12.x repo...
```

## Install Node.js

```
sudo apt-get install -y nodejs
```

```
emmanuel@webserver:~$ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.4 MB of archives.
After this operation, 94.4 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_12.x focal/main amd64 nodejs amd64 12.20.0-deb-1nodesource1 [18.4 MB]
Fetched 18.4 MB in 2s (10.5 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 180252 files and directories currently installed.)
Preparing to unpack .../nodejs_12.20.0-0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (12.20.0-0-deb-1nodesource1) ...
Setting up nodejs (12.20.0-0-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
emmanuel@webserver:~$
```

## Verify the node installation

```
node -v
```

```
emmanuel@webserver:~$ node -v
v12.20.0
emmanuel@webserver:~$ npm -v
6.14.8
emmanuel@webserver:~$ █
```

## Application Code Setup

```
emmanuel@webserver:~$ mkdir todo
emmanuel@webserver:~$ ls -ltr
total 36
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Desktop
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Videos
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Templates
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Public
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Pictures
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Music
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Downloads
drwxr-xr-x 2 emmanuel emmanuel 4096 Dec 26 00:33 Documents
drwxrwxr-x 2 emmanuel emmanuel 4096 Dec 28 11:13 todo
emmanuel@webserver:~$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music     Public    todo
emmanuel@webserver:~$ cd todo
emmanuel@webserver:~/todo$ █
```

```
emmanuel@webserver:~/todo$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (todo)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/emmanuel/todo/package.json:

{
  "name": "todo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",

  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes
emmanuel@webserver:~/todo$ ls
package.json
emmanuel@webserver:~/todo$
```

## Install ExpressJS

```
emmanuel@webserver:~/todo$ npm install express
npm notice created a lockfile as package-lock.json. You should commit this file
.
npm WARN todo@1.0.0 No description
npm WARN todo@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 5.604s
found 0 vulnerabilities

emmanuel@webserver:~/todo$
```

```
emmanuel@webserver:~/todo$ npm install express
npm notice created a lockfile as package-lock.json. You should commit this file
.
npm WARN todo@1.0.0 No description
npm WARN todo@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 5.604s
found 0 vulnerabilities

emmanuel@webserver:~/todo$ touch index.js
emmanuel@webserver:~/todo$ ls
index.js  node_modules  package.json  package-lock.json
emmanuel@webserver:~/todo$ npm install dotenv
npm WARN todo@1.0.0 No description
npm WARN todo@1.0.0 No repository field.

+ dotenv@8.2.0
added 1 package and audited 51 packages in 1.96s
found 0 vulnerabilities

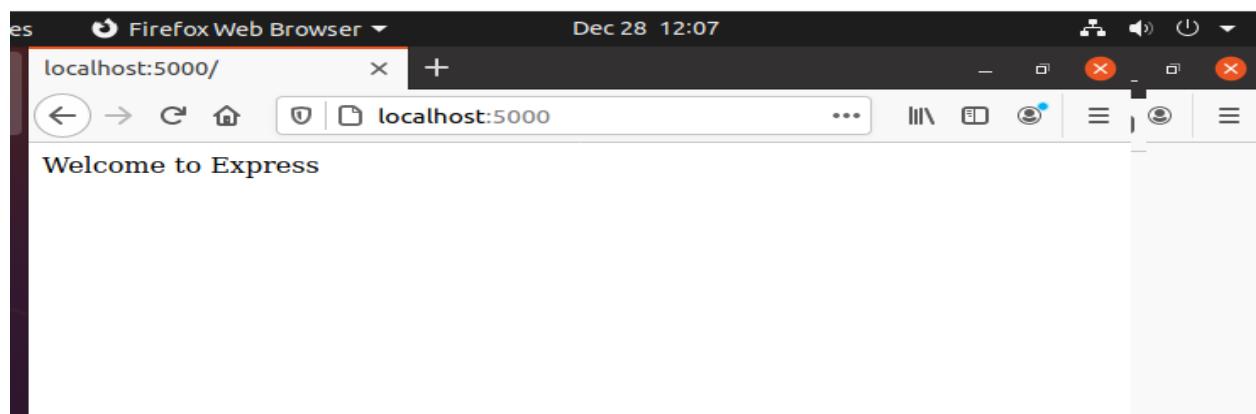
emmanuel@webserver:~/todo$
```

```
emmanuel@webserver:~/todo$ vim index.js
Command 'vim' not found, but can be installed with:
sudo apt install vim          # version 2:8.1.2269-1ubuntu5, or
sudo apt install vim-tiny      # version 2:8.1.2269-1ubuntu5
sudo apt install neovim        # version 0.4.3-3
sudo apt install vim-athena    # version 2:8.1.2269-1ubuntu5
sudo apt install vim-gtk3       # version 2:8.1.2269-1ubuntu5
sudo apt install vim-nox        # version 2:8.1.2269-1ubuntu5

emmanuel@webserver:~/todo$ sudo apt install vim
[sudo] password for emmanuel:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  vim-runtime
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,111 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
emmanuel@webserver:~/todo$ vim index.js
emmanuel@webserver:~/todo$
```

```
emmanuel@webserver:~/todo$ node index.js
Server running on port 5000
```



```
emmanuel@webserver:~/todo$ node index.js
Server running on port 5000
```

```
clear
exit
^C
```

```
emmanuel@webserver:~/todo$ █
```

## Routes

```
emmanuel@webserver:~/todo$ mkdir routes
emmanuel@webserver:~/todo$ cd routes
emmanuel@webserver:~/todo/routes$ touch api.js
emmanuel@webserver:~/todo/routes$ vim api.js
emmanuel@webserver:~/todo/routes$ █
```

## Models

```
emmanuel@webserver:~/todo/routes$ cd..
cd..: command not found
emmanuel@webserver:~/todo/routes$ cd ..
emmanuel@webserver:~/todo$ npm install mongoose
npm WARN todo@1.0.0 No description
npm WARN todo@1.0.0 No repository field.

+ mongoose@5.11.8
added 31 packages from 95 contributors and audited 82 packages in 6.727s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

emmanuel@webserver:~/todo$ mkdir models
emmanuel@webserver:~/todo$ cd models
emmanuel@webserver:~/todo/models$ touch todo.js
emmanuel@webserver:~/todo/models$ vim todo.js
emmanuel@webserver:~/todo/models$ █
```

```
emmanuel@webserver:~/todo/models$ cd ..
emmanuel@webserver:~/todo$ cd routes
emmanuel@webserver:~/todo/routes$ vim api.js
emmanuel@webserver:~/todo/routes$ █
```



```
emmanuel@webserver: ~/Todo/routes
```

```
router.get('/todos', (req, res, next) => {
  //this will return all the data, exposing only the id and action field to the client
  Todo.find({}, 'action')
    .then(data => res.json(data))
    .catch(next)
});

router.post('/todos', (req, res, next) => {
  if(req.body.action){
    Todo.create(req.body)
      .then(data => res.json(data))
      .catch(next)
  }else {
    res.json({
      error: "The input field is empty"
    })
  }
});

router.delete('/todos/:id', (req, res, next) => {
  Todo.findOneAndDelete({_id: req.params.id})
    .then(data => res.json(data))
    .catch(next)
})

module.exports = router;
:qa
```

## MongoDB Database

All Organizations

All Clusters Emmanuel

PREFERENCES

- Legacy 2FA
- Personalization
- Invitations
- Organizations**
- Public API Access

## Organizations



You don't belong to any Organizations

To get started, create an Organization. Within an Organization you can create projects, invite users, and setup a billing account.

Create an Organization

Learn more about what Organization can do in our docs.  
[Organizations and Projects](#)

Feature Requests

All Organizations

All Clusters Emmanuel

PREFERENCES

- Legacy 2FA
- Personalization
- Invitations
- Organizations**
- Public API Access

← Organizations

### Create Organization

Name and Service Add Members Next

Name Your Organization

Fuspec

Select Cloud Service

Features	<input checked="" type="radio"/> MongoDB Atlas	<input type="radio"/> Cloud Manager
Automated database configuration	✓	✓
Continuous backup and point-in-time recovery	✓	✓
Queryable backup snapshots	✓	✓
Fine grained database monitoring & customizable alerts	✓	✓

Feature Requests

Fuspec Access Manager Support Billing All Clusters Emmanuel

ORGANIZATION Projects Alerts 0 Activity Feed Settings Access Manager Billing Support

FUSPEC Projects Find a project... Project Name Clusters Users Teams Alerts Actions

System Status: All Good Last Login: 73.152.119.181 ©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales Feature Requests

Fuspec Access Manager Support Billing All Clusters Emmanuel

ORGANIZATION Projects Alerts 0 Activity Feed Settings Access Manager Billing Support

FUSPEC > PROJECTS Create a Project

Name Your Project Add Members Next

Name Your Project Project Name Cancel Next

System Status: All Good Last Login: 73.152.119.181 ©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales Feature Requests

Fuspec Access Manager Support Billing All Clusters Emmanuel

**ORGANIZATION**

- Projects**
- Alerts 1
- Activity Feed
- Settings
- Access Manager
- Billing
- Support

**FUSPEC > PROJECTS**

## Create a Project

Name Your Project Add Members Next

**Name Your Project**

Project names have to be unique within the organization (and other restrictions).

MERN STACK PRACTICE - P3

Cancel Next

System Status: All Good Last Login: 73.152.119.181 ©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

**Feature Requests**

**DATA STORAGE**

- Clusters**
- Triggers
- Data Lake

**SECURITY**

- Database Access
- Network Access
- Advanced

**Atlas** Realm Charts

**FUSPEC > MERN STACK PRACTICE - P3**

## Clusters

Find a cluster...

Create a cluster

Choose your cloud provider, region, and specs.

Build a Cluster

Once your cluster is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

Feature Requests

System Status: All Good

## Choose a plan. Adjust anytime.

Available as a fully managed service across 60+ regions on AWS, Azure, and Google Cloud

### Dedicated Multi-Cloud & Multi-Region Clusters

For teams developing world-class applications that require multi-region resiliency or ultra-low latency.

- ✓ Includes all features from Shared and Dedicated Clusters
- ✓ Replicate data across clouds and regions
- ✓ Globally distributed read and write operations
- ✓ Control data residency at the document level

[Create a cluster](#)

Starting at

**\$0.13/hr\***

\*estimated cost \$98.55/month

### Dedicated Clusters

For teams building applications that need advanced development and production-ready environments.

- ✓ Includes all features from Shared Clusters
- ✓ Auto-scaling
- ✓ Network isolation
- ✓ Realtime performance metrics

[Create a cluster](#)

Starting at

**\$0.08/hr\***

\*estimated cost \$56.94/month

### Shared Clusters

For teams learning MongoDB or developing small applications.

- ✓ Highly available auto-healing cluster
- ✓ End-to-end encryption
- ✓ Role-based access control

[Create a cluster](#)

Starting at

**FREE**

[CLUSTERS >](#) CREATE A STARTER CLUSTER

## Create a Starter Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Cloud Provider & Region

AWS, N. Virginia (us-east-1) >

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage) >  
Encrypted

Additional Settings

MongoDB 4.2, No Backup >

Cluster Name

Cluster0 >



[Privacy](#) · [Terms](#)

**FREE**

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

[Back](#)

[Create Cluster](#)



Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

### Cloud Provider & Region

AWS, N. Virginia (us-east-1) ▾

**aws** **Google Cloud** **Azure**

★ Recommended region ⓘ

**NORTH AMERICA** **EUROPE** **ASIA**

Oregon (us-west-2) ★	Frankfurt (eu-central-1) ★	Mumbai (ap-south-1)
N. Virginia (us-east-1) ★	Ireland (eu-west-1) ★	Singapore (ap-southeast-1) ★
<b>AUSTRALIA</b>		
Sydney (ap-southeast-2) ★		

Fuspec Access Manager Support Billing All Clusters Emmanuel

MERN STACK PRACTIC... : **Atlas** Realm Charts

**DATA STORAGE**  
**Clusters** We are deploying your changes  
FUSPEC > MERN STACK PRACTICE - P3

**SECURITY**  
Database Access **Create a New Cluster**  
Network Access  
Advanced

**Clusters**  
Find a cluster...

SANDBOX  
Cluster0 Version 4.2.11  
CONNECT METRICS COLLECTIONS ...

CLUSTER TIER M0 Sandbox (General)  
REGION AWS / N. Virginia (us-east-1)  
TYPE Replica Set - 3 nodes  
LINKED REALM APP None Linked

Your cluster is being created  
New clusters take between 1-3 minutes to provision.

Feature Requests 🔈

The screenshot shows the MongoDB Atlas interface. The top navigation bar includes 'Fuspec' (selected), 'Access Manager', 'Support', 'Billing', 'All Clusters', and 'Emmanuel'. Below the navigation is a header with 'MERN STACK PRACTIC...', 'Atlas' (selected), 'Realm', 'Charts', and search/filter icons. The main content area is titled 'Clusters' under 'DATA STORAGE'. A sidebar on the left lists 'Clusters', 'Triggers', 'Data Lake', 'Database Access', 'Network Access', and 'Advanced'. The 'Clusters' section displays 'Cluster0' (Version 4.2.11) in a 'SANDBOX' tier. Metrics for 'Operations' (R: 0 W: 0), 'Logical Size' (0.0 B max 512.0 MB), and 'Connections' (0 max 500) are shown over the last 6 hours. An 'Enhance Your Experience' callout suggests upgrading for dedicated throughput and security. A green 'Upgrade' button is visible. The bottom of the page shows 'Feature Requests' and system status 'All Good'.

This screenshot shows a modal window titled 'Add IP Access List Entry' overlaid on the main 'Add an IP address' configuration page. The modal contains instructions about IP access lists and two buttons: 'ADD CURRENT IP ADDRESS' and 'ALLOW ACCESS FROM ANYWHERE'. It also has fields for 'Access List Entry' (with an input field 'Enter IP Address or CIDR Notation') and 'Comment' (with an input field 'Optional comment describing this entry'). A toggle switch indicates the entry is temporary and will be deleted in 6 hours. A 'Confirm' button is at the bottom right. The background page shows the 'Add an IP address' heading and a sub-section 'Configure which IP addresses can access your cluster' with a 'Add IP Address' button. A 'Learn more' link is also present. The bottom of the page includes 'Feature Requests' and a green circular icon.

Add IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

**Access List Entry:**

**Comment:**

This entry is temporary and will be deleted in

Fuspec Access Manager Support Billing All Clusters Emmanuel

MERN STACK PRACTIC... Atlas Realm Charts

We are deploying your changes (current action: configuring MongoDB)

FUSPEC > MERN STACK PRACTICE - P3

## Network Access

IP Access List Peering Private Endpoint

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
0.0.0.0/0  (includes your current IP address)		Pending	

System Status: All Good

©2021 MongoDB, Inc. [Status](#) [Terms](#) [Privacy](#) [Atlas Blog](#) [Contact Sales](#)

Feature Requests

**Network Access**

**IP Access List**

IP Address	Comment	Status	Actions
0.0.0.0/0 <small>(1 WEEK) (includes your current IP address)</small>		Active	EDIT  DELETE

## Create a MongoDB database and collection inside mLab

**Clusters**

**Cluster0**

Overview    Real Time    Metrics    **Collections**    Search    Profiler    Performance Advisor

DATABASES: 0 COLLECTIONS: 0

**Explore Your Data**

- Find: run queries and interact with documents
- Indexes: build and manage indexes
- Aggregation: test aggregation pipelines
- Search: build search indexes

**Load a Sample Dataset** **Add My Own Data**

Learn more in Docs and Tutorials

## Creation of database name and collection name

The screenshot shows the MongoDB Atlas interface for the 'Cluster0' cluster. The left sidebar has sections for DATA STORAGE (Clusters, Triggers, Data Lake) and SECURITY (Database Access, Network Access, Advanced). A 'Get Started' button is visible. The main area shows the cluster details: EMMANUEL'S ORG - 2020-12-28 > PROJECT 3 TO DO APP > CLUSTERS. The cluster version is 4.2.11. The 'Collections' tab is selected, showing one database ('learningapps') and one collection ('todoapp1'). A search bar and a filter button are present. The query results section shows 0 results.

The screenshot shows the MongoDB Atlas interface for the 'Cluster0' cluster in the 'FUSPEC' organization. The left sidebar has sections for DATA STORAGE (Clusters, Triggers, Data Lake) and SECURITY (Database Access, Network Access, Advanced). The main area shows the cluster details: FUSPEC > MERN STACK PRACTICE - P3 > CLUSTERS. The cluster version is 4.2.11. The 'Collections' tab is selected, showing one database ('mydb1') and one collection ('mycollection'). A search bar and a filter button are present. The query results section shows 0 results.

**Edit IP Access List Entry**

This IP address is temporary and will be deleted in **01 WEEK**

Do not modify expiration ▾

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

**ADD CURRENT IP ADDRESS**

**Access List Entry:** 0.0.0.0/0

**Comment:** Optional comment describing this entry

**Cancel** **Confirm**

Emmanuel's Org - 20... Access Manager Support Billing All Clusters Emmanuel

Project 3 To do App : Atlas Realm Charts

**DATA STORAGE** Clusters Triggers Data Lake

**SECURITY** Database Access

**Network Access** Advanced

EMMANUEL'S ORG - 2020-12-28 > PROJECT 3 TO DO APP

## Network Access

- IP Access List **Peering** Private Endpoint

**+ ADD IP ADDRESS**

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
0.0.0.0/0 <b>01 WEEK</b> (includes your current IP address)		Active	<b>EDIT</b> <b>DELETE</b>

System Status: **All Good**  
 ©2021 MongoDB, Inc. [Status](#) [Terms](#) [Privacy](#) [Atlas Blog](#) [Contact Sales](#)

Feature Requests 



```
emmanuel@webserver:~/Todo/routes$ cd ..
emmanuel@webserver:~/Todo$ touch .env
emmanuel@webserver:~/Todo$
```

The screenshot shows the MongoDB Atlas interface for connecting to a cluster. On the left, a sidebar for the 'MERN STACK PRJ' project is visible, showing sections for DATA STORAGE (Clusters, Triggers, Data Lake), SECURITY (Database Access, Network Access, Advanced), and Feature Requests. The main area is titled 'Connect to Cluster0' and has three tabs: 'Setup connection security' (selected), 'Choose a connection method', and 'Connect'. A note says: 'You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now.' Below this, a box contains the message: 'You can't connect yet. Set up your user security permission below.' Step 1, 'Add a connection IP address', has a green checkmark indicating an IP address is whitelisted. Step 2, 'Create a Database User', shows a form with 'Username' set to 'fuspec', 'Password' masked as '.....', and a 'Create Database User' button. To the right, a sidebar shows cluster details: 3 KB, 512.0 MB max, 0.0 B, and a 'Create a New Cluster' button. A 'Your Experience' section encourages upgrading for better throughput and security.

## Connect to Cluster0

Setup connection security > Choose a connection method > Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

You're ready to connect. Choose how you want to connect in the next step.

### 1 Add a connection IP address

✓ An IP address has been whitelisted. [Add another whitelist entry in the IP Whitelist tab.](#)

### 2 Create a Database User

✓ A MongoDB user has been added to this project. [Not yours? Create one in the MongoDB Users tab.](#)

You'll need your MongoDB user's credentials in the next step.

[Close](#)

[Choose a connection method](#)

To obtain the connection string

Connect to Cluster0

✓ Setup connection security ✓ Choose a connection method Connect

1 Select your driver and version

DRIVER	VERSION
Node.js	3.6 or later

2 Add your connection string into your application code

Include full driver code example

```
mongodb+srv://fuspec:<password>@cluster0.qbr37.mongodb.net/<dbname>?retryWrites=true
```



Replace <password> with the password for the **fuspec** user. Replace <dbname> 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](#)

Go Back Close

```
emmanuel@webserver:~/Todo$ vim .env
```

Edited .env created earlier with the connection string below as retrieved from the previous snapshot

DB =  
mongodb+srv://fuspec:password@cluster0.qbr37.mongodb.net/mydb1?retryWrites=true&w=majority

To update the **index.js** to reflect the use of **.env** so that nodejs can connect to the database. I opened the file with **vim index.js**

```
const express = require('express');
require('dotenv').config();

const app = express();

const port = process.env.PORT || 5000;

app.use((req, res, next) => {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
  next();
});

app.use((req, res, next) => {
  res.send('Welcome to Express');
});

app.listen(port, () => {
  console.log(`Server running on port ${port}`)
});
```

~  
~  
~  
~  
~  
~  
~

"index.js" 20L, 457C 20,3 All



Pressed **esc** and typed :%d and enter

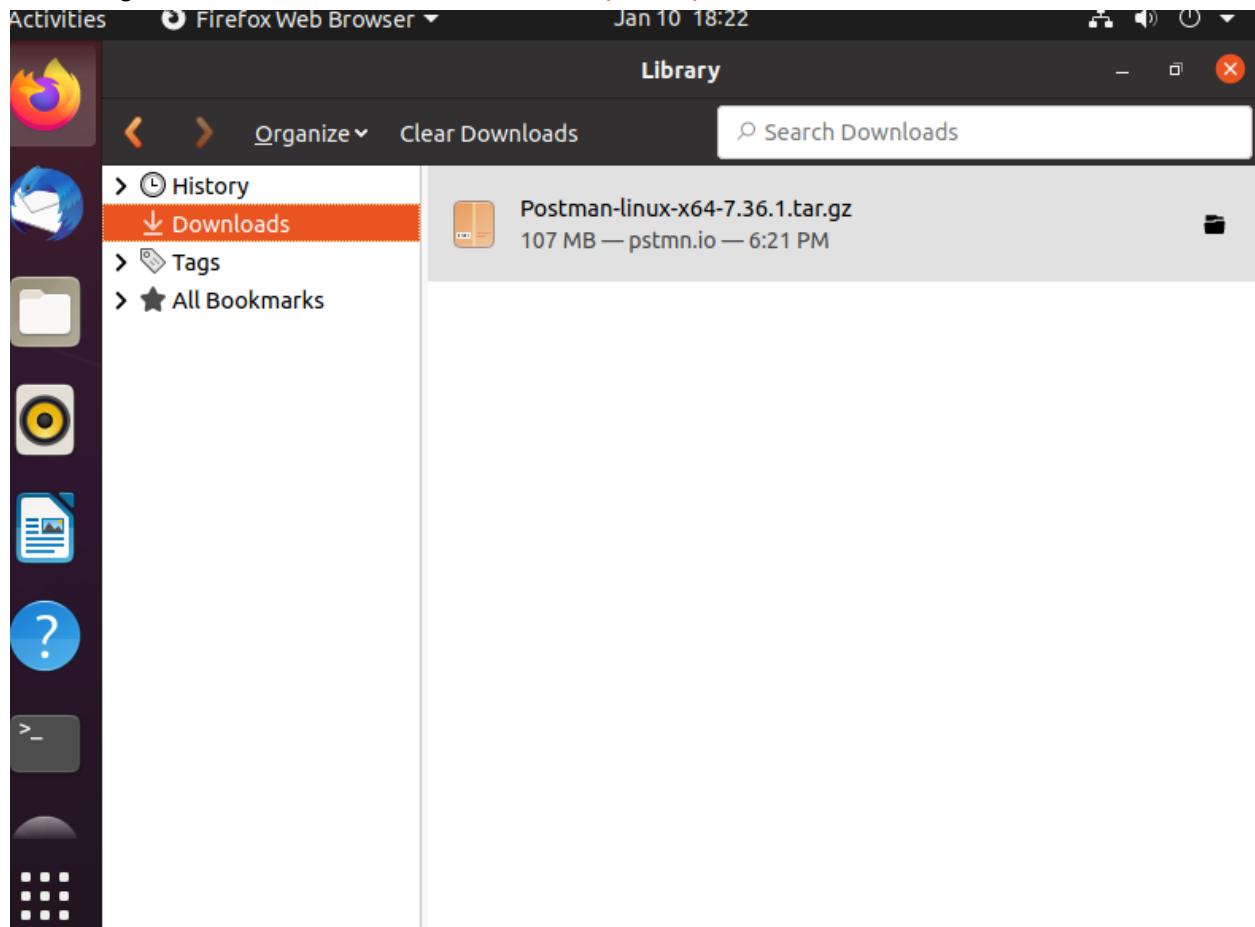
Next I entered the insert mode by pressing **i** on the keyboard and then copied the provided code into index.js.

Next was to run the node index.js command as below

```
emmanuel@webserver:~/todo$ node index.js
Server running on port 5000
(node:35067) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in
a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the
MongoClient constructor.
Database connected successfully
```

## Testing The Code Without A Frontend User Interface

First thing I did was to install Postman from <https://dl.pstmn.io/download/latest/linux>.



### Install Postman App (source: <https://speedysense.com/install-postman-on-ubuntu-20-04/>)

After that I went to my terminal and ran the following commands as shown below to extract the Postman App.

```
tar -xzf Postman-linux-x64-7.36.1.tar.gz
```

```
emmanuel@webserver:~/Todo$ cd
emmanuel@webserver:~$ cd Downloads
emmanuel@webserver:~/Downloads$ ls
Postman-linux-x64-7.36.1.tar.gz

emmanuel@webserver:~/Downloads$ tar -xzf Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$
```

Also used the following command to Move Postman directory to opt/ directory

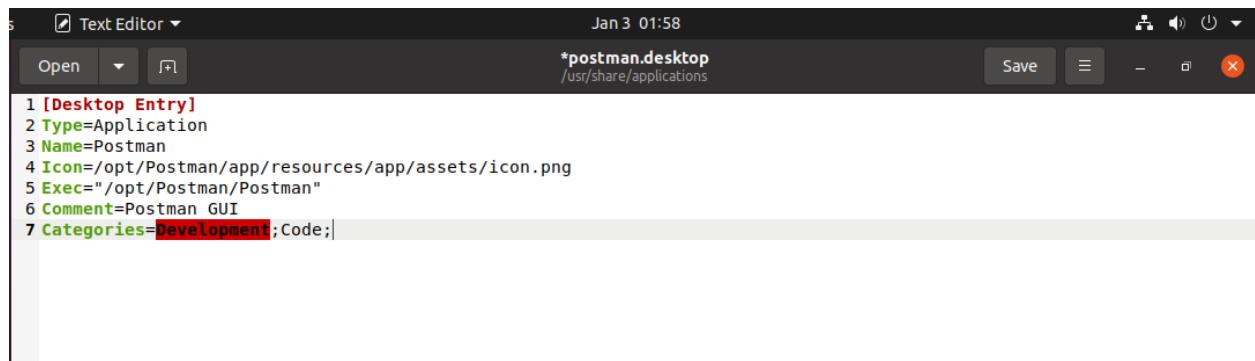
```
sudo mv Postman /opt
```

## Creating a Symbolic Links

```
sudo ln -s /opt/Postman/Postman /usr/local/bin/postman
```

```
emmanuel@webserver:~/Downloads$ tar -xzf Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$ ls -ltr
total 109380
drwxr-xr-x 3 emmanuel emmanuel      4096 Dec 17 06:28 Postman
-r----- 1 emmanuel emmanuel 111994751 Jan  1 19:56 Postman-linux-x64-7.36.1.t
ar.gz
emmanuel@webserver:~/Downloads$ sudo mv Postman /opt
[sudo] password for emmanuel:
emmanuel@webserver:~/Downloads$ ls
Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$ cd /opt
emmanuel@webserver:/opt$ ls
Postman VBoxGuestAdditions-6.1.16
emmanuel@webserver:/opt$ cd ..
emmanuel@webserver:/$ sudo ln -s /opt/Postman/Postman /usr/local/bin/postman
emmanuel@webserver:/$
```

Next thing was to create a desktop file for Postman App to enable easy search of Postman in App Dash. I ran the command `sudo gedit /usr/share/applications/postman.desktop` and added the following lines then saved.



```
emmanuel@webserver: /
```

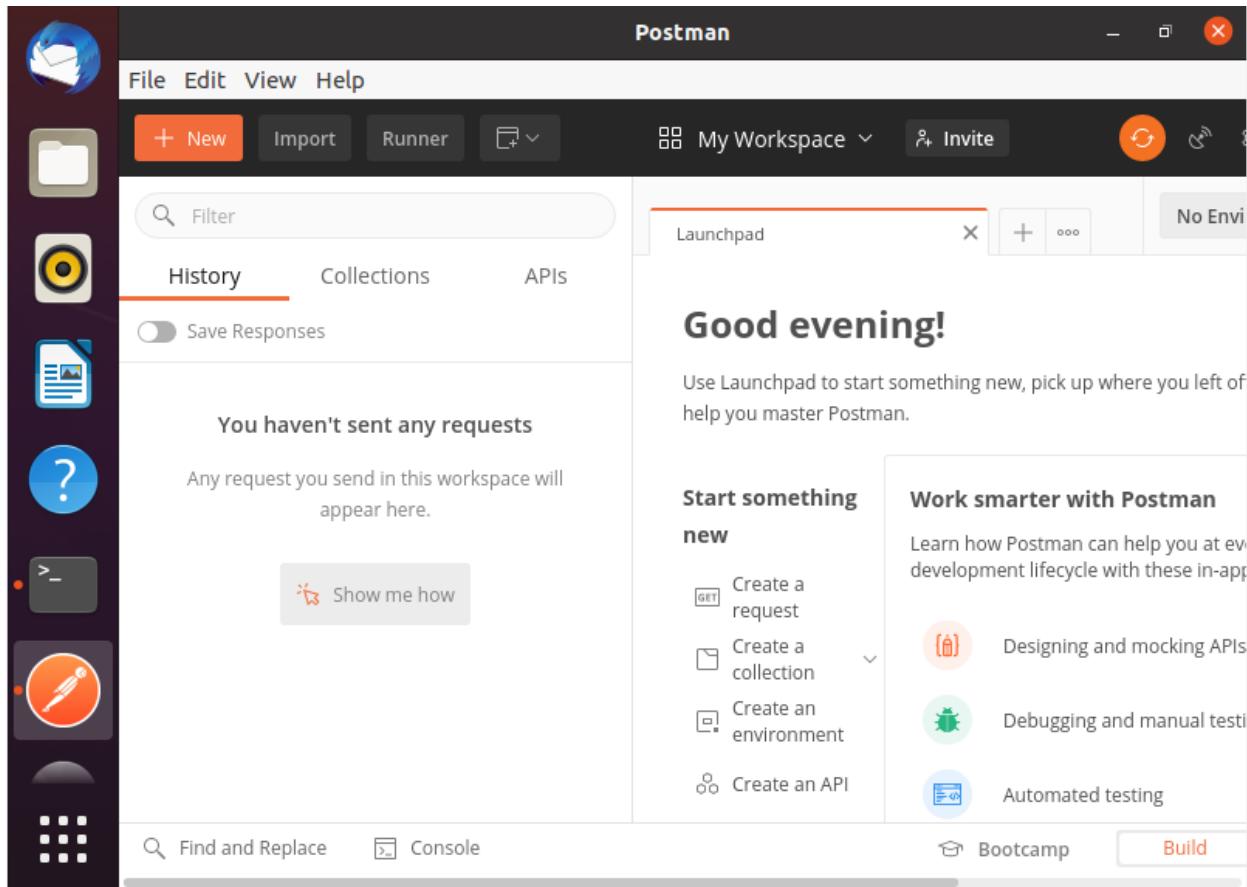
```
-r----- 1 emmanuel emmanuel 111994751 Jan  1 19:56 Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$ tar -xzf Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$ ls -ltr
total 109380
drwxr-xr-x 3 emmanuel emmanuel      4096 Dec 17 06:28 Postman
-r----- 1 emmanuel emmanuel 111994751 Jan  1 19:56 Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$ sudo mv Postman /opt
[sudo] password for emmanuel:
emmanuel@webserver:~/Downloads$ ls
Postman-linux-x64-7.36.1.tar.gz
emmanuel@webserver:~/Downloads$ cd /opt
emmanuel@webserver:/opt$ ls
Postman  VBoxGuestAdditions-6.1.16
emmanuel@webserver:/opt$ cd ..
emmanuel@webserver:$ sudo ln -s /opt/Postman/Postman /usr/local/bin/postman
emmanuel@webserver:$ sudo gedit /usr/share/applications/postman.desktop

(gedit:2290): Tepl-WARNING **: 01:59:56.473: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metadata.
emmanuel@webserver:$
```

Postman is now installed and running on my Ubuntu machine.

Next was to create and account and sign in

I tried to run post request on Postman but was getting error message below



The screenshot shows the Postman application interface. On the left, there's a sidebar with tabs for History, Collections (which is selected), APIs, and Trash. A collection named 'FirstAPI' is listed under Collections, containing 1 request. Below the sidebar, a main panel shows a POST request titled 'Post to API'. The URL is set to 'http://localhost:5000/api/todos'. The 'Body' tab is active, showing a JSON payload: { "action": "Finish Project 8 and 9" }. Other tabs like Params, Authorization, Headers, and Tests are also visible. At the bottom, the 'Response' section displays an error message: 'Error: connect ECONNREFUSED 127.0.0.1:5000'. The status bar at the bottom indicates '1 ERROR'.

The error was because I did not connect to my database from the terminal using **node index.js**

```
emmanuel@webserver:~$ cd todo
emmanuel@webserver:~/todo$ node index.js
Server running on port 5000
(node:3253) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Database connected successfully
SyntaxError: Unexpected token T in JSON at position 6
    at JSON.parse (<anonymous>)
    at parse (/home/emmanuel/todo/node_modules/body-parser/lib/types/json.js:89:19)
    at /home/emmanuel/todo/node_modules/body-parser/lib/read.js:121:18
    at invokeCallback (/home/emmanuel/todo/node_modules/raw-body/index.js:224:16)
    at done (/home/emmanuel/todo/node_modules/raw-body/index.js:213:7)
    at IncomingMessage.onEnd (/home/emmanuel/todo/node_modules/raw-body/index.js:273:7)
    at IncomingMessage.emit (events.js:314:20)
    at endReadableNT (_stream_readable.js:1241:12)
    at processTicksAndRejections (internal/process/task_queues.js:84:21) {
  expose: true,
  statusCode: 400,
  status: 400,
  body: '{\n    This is what programming feels like\n}\n',
  type: 'entity.parse.failed'
}
SyntaxError: Unexpected token
in JSON at position 42
    at JSON.parse (<anonymous>)
    at parse (/home/emmanuel/todo/node_modules/body-parser/lib/types/json.js:89:19)
    at /home/emmanuel/todo/node_modules/body-parser/lib/read.js:121:18
    at invokeCallback (/home/emmanuel/todo/node_modules/raw-body/index.js:224:16)
    at done (/home/emmanuel/todo/node_modules/raw-body/index.js:213:7)
    at IncomingMessage.onEnd (/home/emmanuel/todo/node_modules/raw-body/index.js:273:7)
    at IncomingMessage.emit (events.js:314:20)
    at endReadableNT (_stream_readable.js:1241:12)
    at processTicksAndRejections (internal/process/task_queues.js:84:21) {
  expose: true,
  statusCode: 400,
  status: 400,
  body: '{\n    "This is what programming feels like\n}\n',
  type: 'entity.parse.failed'
}
```

The result after connecting is shown below

The screenshot shows the Postman application interface. In the top navigation bar, 'File', 'Edit', 'View', and 'Help' are visible. Below the bar, there are buttons for '+ New', 'Import', 'Runner', and a workspace dropdown 'My Workspace'. To the right of the workspace dropdown are 'Invite', 'Upgrade', and other icons. The main workspace area has tabs for 'Launchpad' and 'Post to API'. Under 'Post to API', the method is set to 'POST' and the URL is 'http://localhost:5000/api/todos'. The 'Body' tab is selected, showing a JSON payload:

```
1 [  
2   {  
3     "_id": "5ff178294f86c60cb510ee5f",  
4     "action": "Finish project 3 today",  
5     "__v": 0  
6   }  
7 ]
```

Below the body, the response status is '200 OK' with a response time of '280 ms' and a size of '421 B'. There are buttons for 'Save Response' and 'Beautify'. On the left sidebar, under 'FirstAPI', there is a collection named 'FirstAPI' containing 1 request. At the bottom, there are buttons for 'POST' (Post to API) and 'GET' (http://localhost:5000/api/todos).

Create get request to my API on <http://localhost:5000/api/todos>

The screenshot shows the Postman application interface. The top navigation bar and workspace area are identical to the previous screenshot. In the main workspace, the method is set to 'GET' and the URL is 'http://localhost:5000/api/todos'. The 'Body' tab is selected, showing the response data:

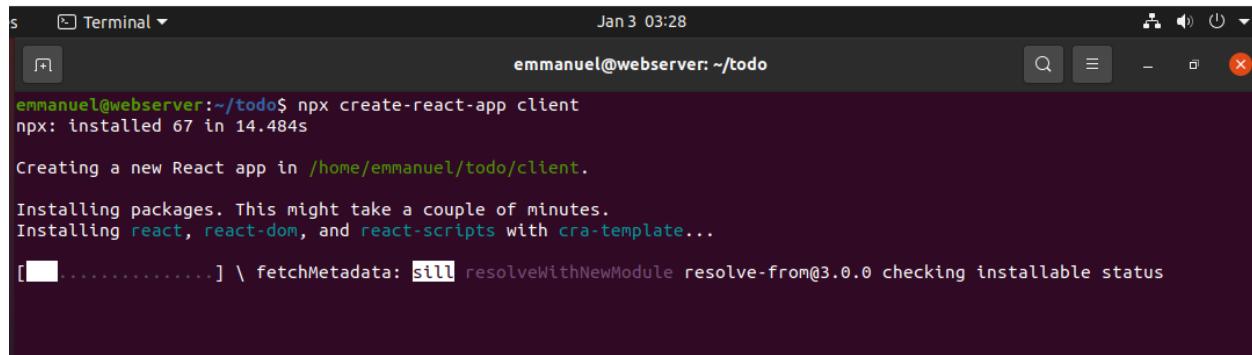
```
1 [  
2   {  
3     "_id": "5ff178294f86c60cb510ee5f",  
4     "action": "Finish project 3 today",  
5     "__v": 0  
6   },  
7   {  
8     "_id": "5ff178724f86c60cb510ee60",  
9     "action": "Finish project 3 today"  
10  }  
11 ]
```

The response status is '200 OK' with a response time of '44 ms' and a size of '485 B'. There are buttons for 'Save Response' and 'Beautify'. On the left sidebar, under 'FirstAPI', there are 2 requests. At the bottom, there are buttons for 'POST' (Post to API) and 'GET' (http://localhost:5000/api/todos).

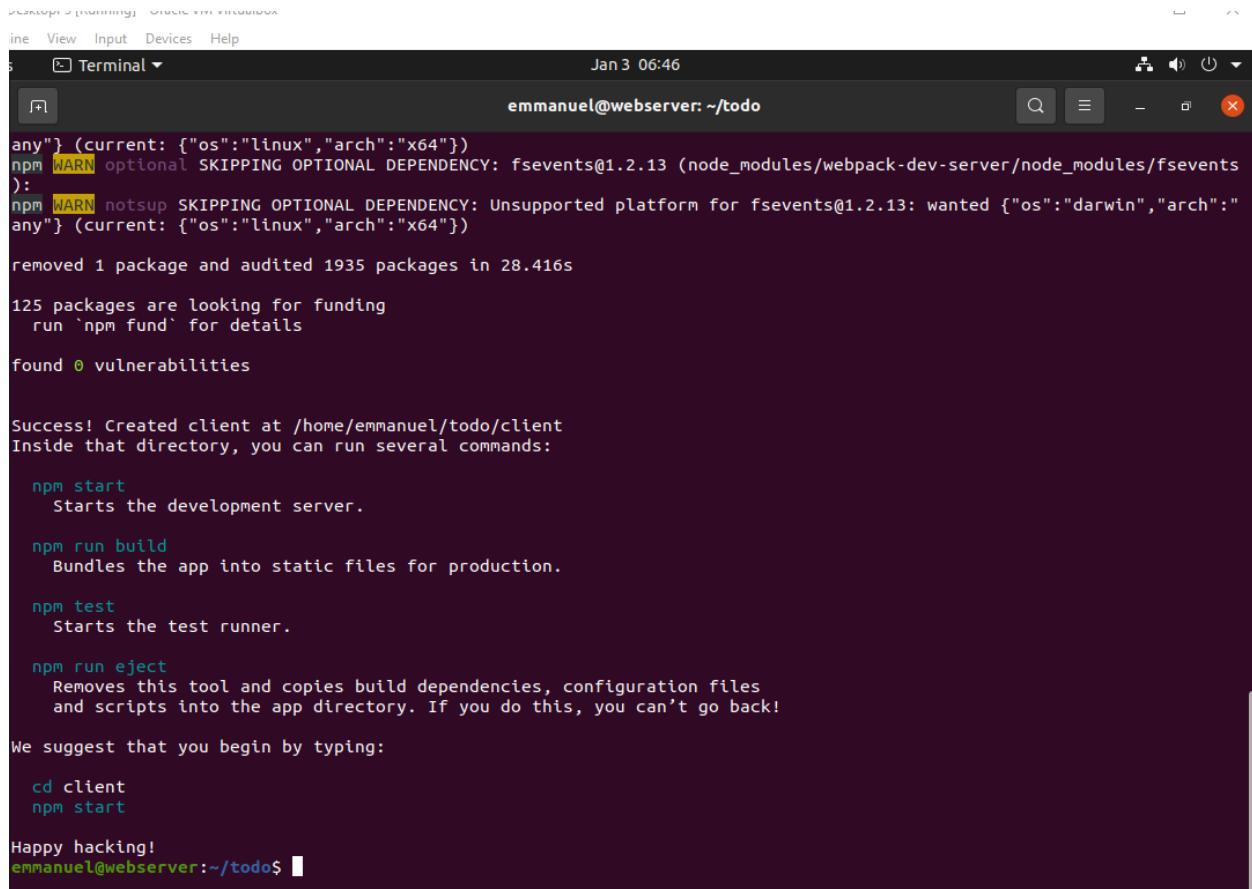
## Creating the Frontend

In the same root directory as my backend code, which is the Todo directory, I ran:

```
$ npx create-react-app client
```



```
Terminal Jan 3 03:28
emmanuel@webserver:~/todo$ npx create-react-app client
npx: installed 67 in 14.484s
Creating a new React app in /home/emmanuel/todo/client.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
[██████████] \ fetchMetadata: sill resolveWithNewModule resolve-from@3.0.0 checking installable status
```



```
Terminal Jan 3 06:46
emmanuel@webserver:~/todo
any} (current: {"os": "linux", "arch": "x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/webpack-dev-server/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os": "darwin", "arch": "any"} (current: {"os": "linux", "arch": "x64"})
removed 1 package and audited 1935 packages in 28.416s
125 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

Success! Created client at /home/emmanuel/todo/client
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd client
  npm start

Happy hacking!
emmanuel@webserver:~/todo$
```

This created a new folder in my **Todo** directory called **client**, where the react code was installed.

## Running the React App

1. Install **concurrently**. It is used to run more than one command simultaneously from the same terminal window. `$ npm install concurrently --save-dev`

```
emmanuel@webserver:~/todo$ npm install concurrently --save-dev
npm WARN todo@1.0.0 No description
npm WARN todo@1.0.0 No repository field.

+ concurrently@5.3.0
added 56 packages from 50 contributors and audited 138 packages in 16.933s

6 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

emmanuel@webserver:~/todo$ █
```

2. Install **nodemon**. It is used to run and monitor the server. If there is any change in the server code, nodemon will restart it automatically and load the new changes.

`$ npm install nodemon --save-dev`

```
emmanuel@webserver:~/todo$ npm install nodemon --save-dev
npm WARN deprecated fsevents@2.1.3: Please update to v 2.2.x

> nodemon@2.0.6 postinstall /home/emmanuel/todo/node_modules/nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN todo@1.0.0 No description
npm WARN todo@1.0.0 No repository field.

+ nodemon@2.0.6
added 120 packages from 53 contributors and audited 259 packages in 16.764s

17 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

emmanuel@webserver:~/todo$ █
```

3. In the **Todo** folder and open the **package.json** file. Change the highlighted part of the below screenshot and replace with the code below.

```
"scripts": {  
  "start": "node index.js",  
  "start-watch": "nodemon index.js",  
  "dev": "concurrently \"npm run start-watch\" \"cd client && npm start\""  
},
```

```
emmanuel@webserver:~/todo$ ls -ltr  
total 112  
drwxrwxr-x  2 emmanuel emmanuel  4096 Dec 28 12:32 models  
drwxrwxr-x  2 emmanuel emmanuel  4096 Dec 28 12:38 routes  
-rw-rw-r--  1 emmanuel emmanuel   953 Dec 28 15:27 index.js  
drwxrwxr-x  5 emmanuel emmanuel  4096 Jan  3 03:33 client  
drwxrwxr-x 224 emmanuel emmanuel 12288 Jan  3 06:54 node_modules  
-rw-rw-r--  1 emmanuel emmanuel   383 Jan  3 06:54 package.json  
-rw-rw-r--  1 emmanuel emmanuel 80687 Jan  3 06:54 package-lock.json  
emmanuel@webserver:~/todo$ vim package.json  
emmanuel@webserver:~/todo$
```

The screenshot shows a terminal window titled 'Terminal' with the command 'emmanuel@webserver: ~/todo'. The window displays the contents of the package.json file. The file is a JSON object with several properties: name, version, description, main, scripts, start, start-watch, dev, author, license, dependencies, and devDependencies. The 'scripts' section is highlighted in red, indicating it needs to be modified. The 'scripts' section contains three entries: 'start', 'start-watch', and 'dev'. The 'start' entry is set to 'node index.js', 'start-watch' to 'nodemon index.js', and 'dev' to 'concurrently \"npm run start-watch\" \"cd client && npm start\"'. The rest of the file contains standard Node.js dependency definitions for dotenv, express, and mongoose.

```
{  
  "name": "todo",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "start": "node index.js",  
    "start-watch": "nodemon index.js",  
    "dev": "concurrently \"npm run start-watch\" \"cd client && npm start\""  
  },  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "dotenv": "^8.2.0",  
    "express": "^4.17.1",  
    "mongoose": "^5.11.8"  
  },  
  "devDependencies": {  
    "concurrently": "^5.3.0",  
    "nodemon": "^2.0.6"  
  }  
}
```

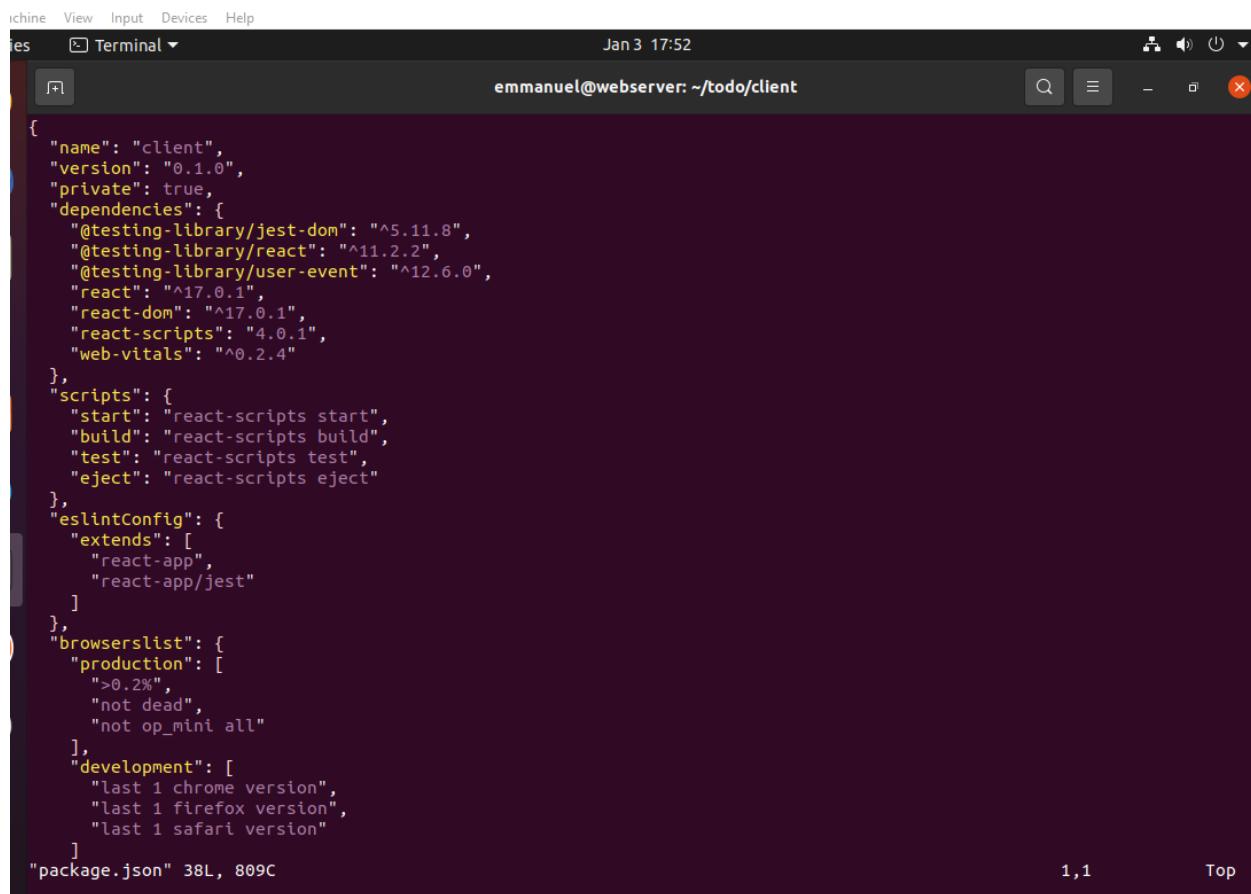
## Configure Proxy in `package.json`

1. Enter into the client folder from Todo directory

```
emmanuel@webserver:~/todo$ cd client  
emmanuel@webserver:~/todo/client$
```

2. Open the `package.json` file

vim `package.json`



A screenshot of a terminal window titled "Terminal" showing the command "emmanuel@webserver: ~/todo/client". The window displays the contents of the `package.json` file in JSON format. The file includes dependencies like jest-dom, react, user-event, react, react-dom, react-scripts, and web-vitals, along with scripts for start, build, test, and eject, and configurations for eslint and browserslist. The terminal status bar shows "Jan 3 17:52" and file statistics "package.json" 38L, 809C.

```
{  
  "name": "client",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    "@testing-library/jest-dom": "^5.11.8",  
    "@testing-library/react": "^11.2.2",  
    "@testing-library/user-event": "^12.6.0",  
    "react": "^17.0.1",  
    "react-dom": "^17.0.1",  
    "react-scripts": "4.0.1",  
    "web-vitals": "^0.2.4"  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test",  
    "eject": "react-scripts eject"  
  },  
  "eslintConfig": {  
    "extends": [  
      "react-app",  
      "react-app/jest"  
    ]  
  },  
  "browserslist": {  
    "production": [  
      ">0.2%",  
      "not dead",  
      "not op_mini all"  
    ],  
    "development": [  
      "last 1 chrome version",  
      "last 1 firefox version",  
      "last 1 safari version"  
    ]  
  }  
}  
"package.json" 38L, 809C
```

3. Add the key value pair in the `package.json` file "`proxy": "http://localhost:5000"` to make it possible to access the application directly from the browser by simply calling the server url like `http://localhost:5000` rather than always including the entire path like `http://localhost:5000//api/todos`

```
{  
  "name": "client",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    "@testing-library/jest-dom": "^5.11.8",  
    "@testing-library/react": "^11.2.2",  
    "@testing-library/user-event": "^12.6.0",  
    "react": "^17.0.1",  
    "react-dom": "^17.0.1",  
    "react-scripts": "4.0.1",  
    "web-vitals": "^0.2.4"  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test",  
    "eject": "react-scripts eject"  
  },  
  "proxy": "http://localhost:5000",  
  "eslintConfig": {  
    "extends": [  
      "react-app",  
      "react-app/jest"  
    ]  
  }  
}
```

4,1-8

Top

CD into **Todo** directory,

```
emmanuel@webserver:~/todo/client$ cd ..  
emmanuel@webserver:~/todo$
```

```
npm run dev
```

```
emmanuel@webserver:~/Todo$ npm run dev

> todo@1.0.0 dev /home/emmanuel/Todo
> concurrently "npm run start-watch" "cd client && npm start"

[0]
[0] > todo@1.0.0 start-watch /home/emmanuel/Todo
[0] > nodemon index.js
[0]
[1]
[1] > client@0.1.0 start /home/emmanuel/Todo/client
[1] > react-scripts start
[1]
[0] [nodemon] 2.0.7
[0] [nodemon] to restart at any time, enter `rs`
[0] [nodemon] watching path(s): ***!
[0] [nodemon] watching extensions: js,mjs,json
[0] [nodemon] starting `node index.js`
[0] Server running on port 5000
[0] (node:44309) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[0] Database connected successfully
```

```
[0] Database connected successfully
[1] i 「wds」: Project is running at http://10.0.2.15/
[1] i 「wds」: webpack output is served from
[1] i 「wds」: Content not from webpack is served from /home/emmanuel/Todo/client
/public
[1] i 「wds」: 404s will fallback to /
[1] Starting the development server...
[1]
[1] Compiled successfully!
[1]
[1] You can now view client in the browser.
[1]
[1] Local:          http://localhost:3000
[1] On Your Network: http://10.0.2.15:3000
[1]
[1] Note that the development build is not optimized.
[1] To create a production build, use npm run build.
[1]
```

```
emmanuel@webserver:~/todo$ npm run dev
> todo@1.0.0 dev /home/emmanuel/todo
> concurrently "npm run start-watch" "cd client && npm start"

[0]
[0] > todo@1.0.0 start-watch /home/emmanuel/todo
[0] > nodemon index.js
[0]
[1] npm ERR! code EJSONPARSE
[1] npm ERR! file /home/emmanuel/todo/client/package.json
[1] npm ERR! JSON.parse Failed to parse json
[1] npm ERR! JSON.parse Unexpected string in JSON at position 37 while parsing '{'
[1] npm ERR! JSON.parse "proxy": "http://localhost:5000"
[1] npm ERR! JSON.parse "na"
[1] npm ERR! JSON.parse Failed to parse package.json data.
[1] npm ERR! JSON.parse package.json must be actual JSON, not just JavaScript.
[1]
[1] npm ERR! A complete log of this run can be found in:
[1] npm ERR!     /home/emmanuel/.npm/_logs/2021-01-03T23_19_40_719Z-debug.log
[1] cd client && npm start exited with code 1
[0] [nodemon] 2.0.6
[0] [nodemon] to restart at any time, enter `rs`
[0] [nodemon] watching path(s): ***!
[0] [nodemon] watching extensions: js,mjs,json
[0] [nodemon] starting `node index.js`
[0] Server running on port 5000
[0] (node:5492) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed
in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to
the MongoClient constructor.
[0] Database connected successfully
```

## Creating React Components

cd client

```
^C[0] npm run start-watch exited with code SIGINT
emmanuel@webserver:~/todo$ cd client
emmanuel@webserver:~/todo/client$
```

move to the src directory

cd src

```
emmanuel@webserver:~/todo/client/src$
```

Inside **src** folder create another folder called **components**

mkdir components

```
emmanuel@webserver:~/todo/client/src$ mkdir components  
emmanuel@webserver:~/todo/client/src$
```

Move into the components directory with

```
cd components
```

```
emmanuel@webserver:~/todo/client/src$ cd components  
emmanuel@webserver:~/todo/client/src/components$
```

inside the components directory create three files [Input.js](#), [ListTodo.js](#) and [Todo.js](#).

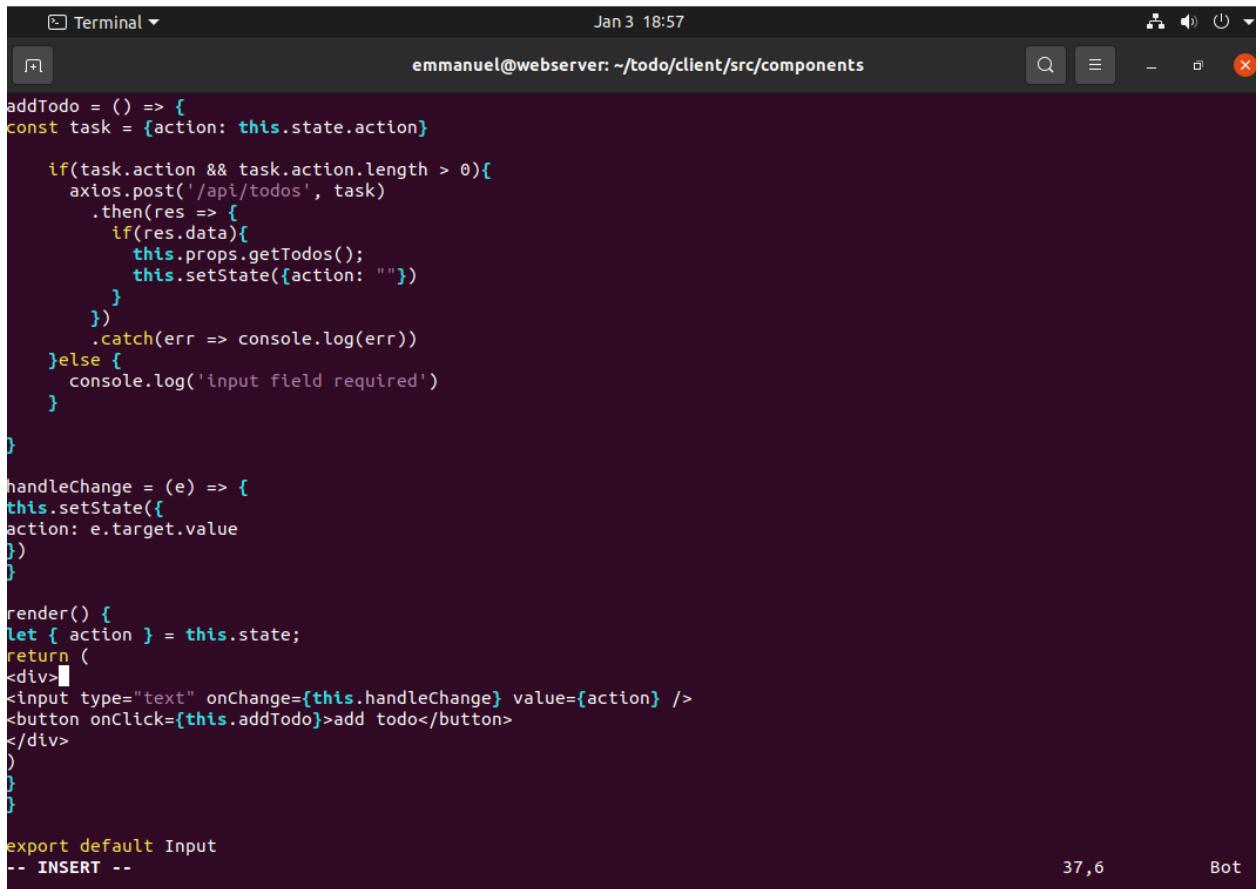
```
touch Input.js ListTodo.js Todo.js
```

```
emmanuel@webserver:~/todo/client/src/components$ touch Input.js ListTodo.js Todo.js  
emmanuel@webserver:~/todo/client/src/components$ ls -l  
total 0  
-rw-rw-r-- 1 emmanuel emmanuel 0 Jan  3 18:52 Input.js  
-rw-rw-r-- 1 emmanuel emmanuel 0 Jan  3 18:52 ListTodo.js  
-rw-rw-r-- 1 emmanuel emmanuel 0 Jan  3 18:52 Todo.js  
emmanuel@webserver:~/todo/client/src/components$
```

Open [Input.js](#) file and edit

```
vim Input.js
```

Copy and paste the required code



```
Terminal ▾ Jan 3 18:57
emmanuel@webserver: ~/todo/client/src/components
addTodo = () => {
  const task = {action: this.state.action}

  if(task.action && task.action.length > 0){
    axios.post('/api/todos', task)
      .then(res => {
        if(res.data){
          this.props.getTodos();
          this.setState({action: ""})
        }
      })
      .catch(err => console.log(err))
  }else {
    console.log('input field required')
  }
}

handleChange = (e) => {
  this.setState({
    action: e.target.value
  })
}

render() {
  let { action } = this.state;
  return (
    <div>
      <input type="text" onChange={this.handleChange} value={action} />
      <button onClick={this.addTodo}>add todo</button>
    </div>
  )
}
}

export default Input
-- INSERT --
37,6           Bot
```

Save and exit using escape :w and :qa enter

To make use of axios, which is a Promise based HTTP client for the browser and node.js, we need to cd into the client directory from the terminal and run yarn add axios or npm install axios.

Move to the src folder

cd ..

```
emmanuel@webserver:~/todo/client/src/components$ cd ..
emmanuel@webserver:~/todo/client/src$
```

Move to clients folder

```
cd ..
```

```
emmanuel@webserver:~/todo/client/src$ cd ..  
emmanuel@webserver:~/todo/client$
```

## Install Axios

```
$ npm install axios
```

```
emmanuel@webserver:~/Todo/client$ npm install axios  
npm WARN tsutils@3.19.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.1 (node_modules/fs-events):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.1: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/watchpack-chokidar2/node_modules/fsevents):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/webpack-dev-server/node_modules/fsevents):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})  
+ axios@0.21.1  
added 1 package from 1 contributor and audited 1936 packages in 79.037s  
  
126 packages are looking for funding  
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
emmanuel@webserver:~/Todo/client$
```

Go to components directory

```
cd src/components
```

```
emmanuel@webserver:~/todo/client$ cd src/components  
emmanuel@webserver:~/todo/client/src/components$
```

Open [ListTodo.js](#)

vim ListTodo.js

```
emmanuel@webserver: ~/todo/client          emmanuel@webserver: ~/todo/client/src/components
import React from 'react';
const ListTodo = ({ todos, deleteTodo }) => {
  return (
    <ul>
      {
        todos && todos.length > 0 ?
        (
          todos.map(todo => {
            return (
              <li key={todo._id} onClick={() => deleteTodo(todo._id)}>{todo.action}</li>
            )
          })
        )
        :
        (
          <li>No todo(s) left</li>
        )
      }
    </ul>
  )
}

export default ListTodo
~  
~  
~  
~  
~  
~  
~  
:w
```

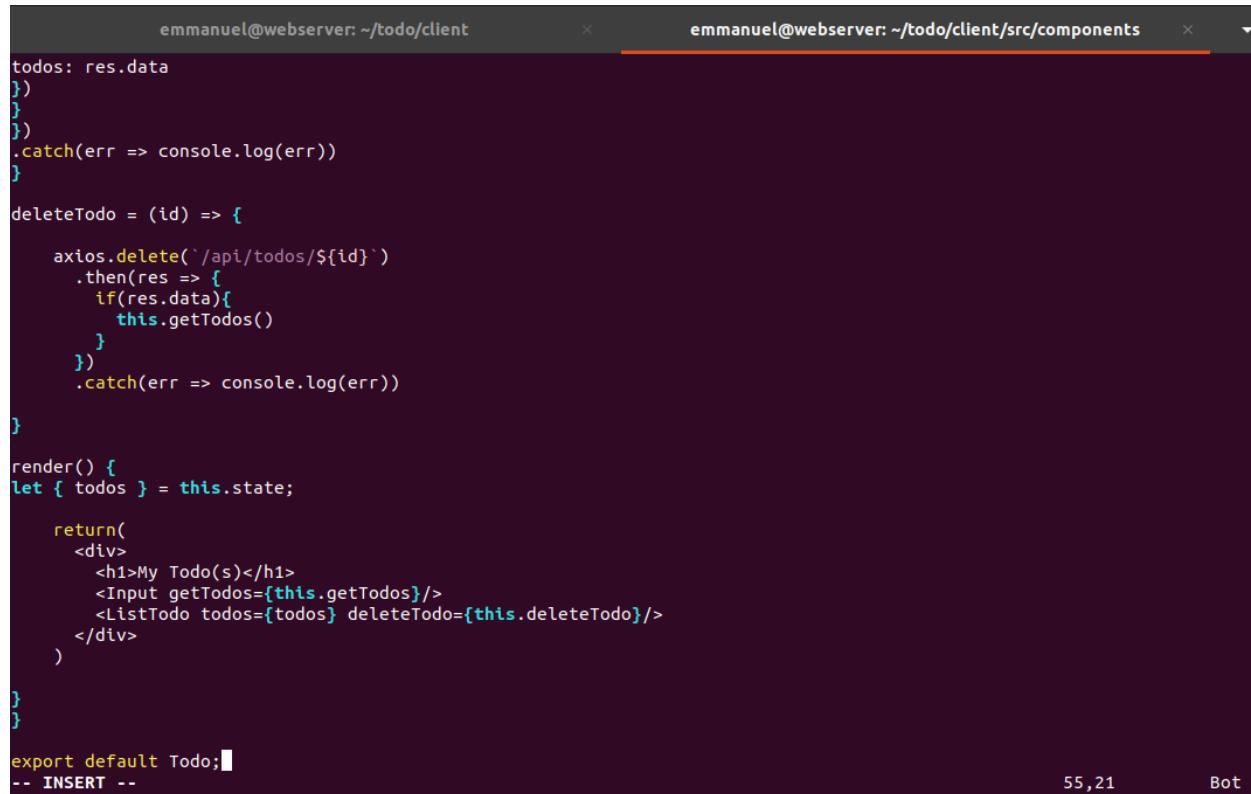
In the [ListTodo.js](#) copy and paste the necessary code

Press escape.

Type :w to write and :qa to quit and exit

Open [Todo.js](#) using vim Todo.js

Then in [Todo.js](#) file you write the necessary code



```
emmanuel@webserver: ~/todo/client          emmanuel@webserver: ~/todo/client/src/components
todos: res.data
})
}
}
})
.catch(err => console.log(err))
}

deleteTodo = (id) => {
  axios.delete(`/api/todos/${id}`)
    .then(res => {
      if(res.data){
        this.getTodos()
      }
    })
    .catch(err => console.log(err))
}

render() {
let { todos } = this.state;
  return(
    <div>
      <h1>My Todo(s)</h1>
      <Input getTodos={this.getTodos}/>
      <ListTodo todos={todos} deleteTodo={this.deleteTodo}/>
    </div>
  )
}
}

export default Todo;■
-- INSERT --
```

55,21      Bot

Save and exit.

We need to make little adjustment to our react code. Delete the logo and adjust our [App.js](#) to look like this.

Move to the src folder

```
cd ..
```

```
emmanuel@webserver:~/todo/client/src/components$ cd ..
emmanuel@webserver:~/todo/client/src$
```

Run vim App.js from the src folder

Copy and paste the code below into it.

```
import React from 'react';
```

```
import Todo from './components/Todo';
```

```
import './App.css';
```

```
const App = () => {
```

```
return (
  <div className="App">
    <Todo />
  </div>
);
}

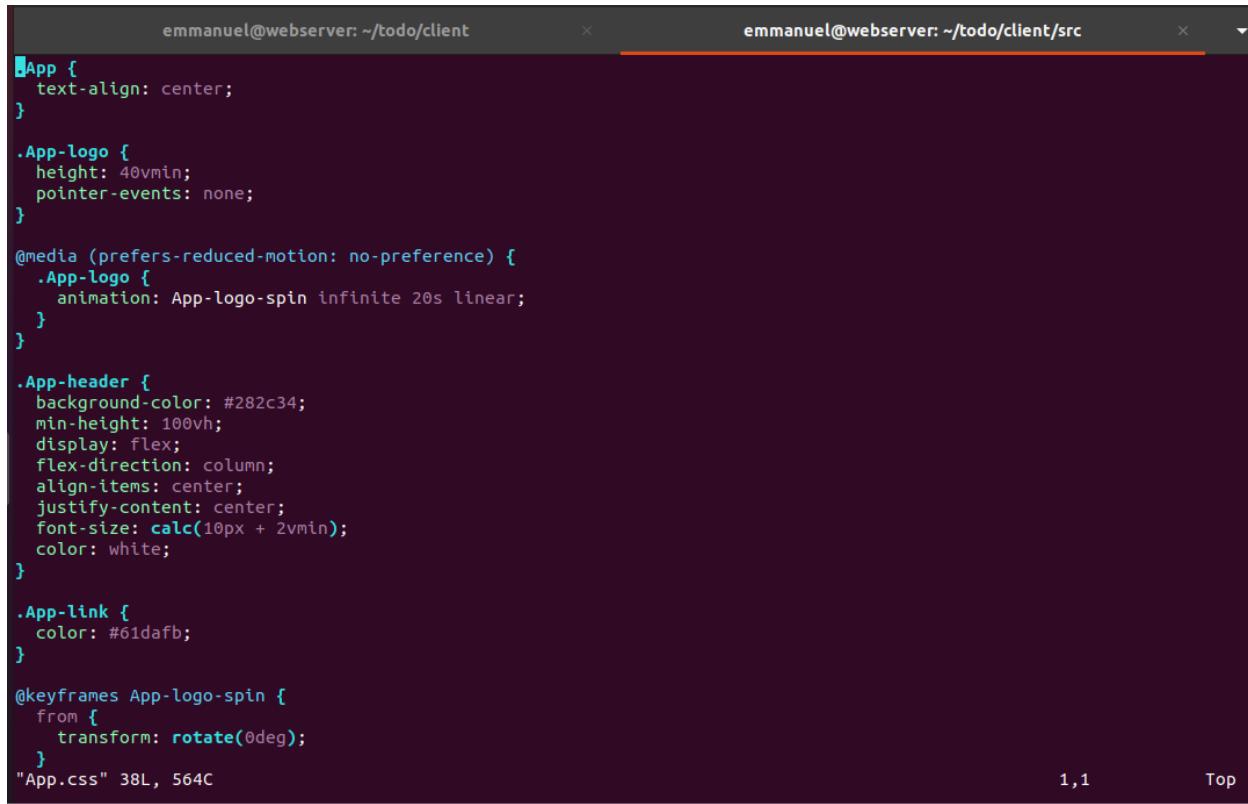
export default App;
```

14,19 All

After pasting, exit the editor.

In the src directory open the [App.css](#)

vim App.css



```
emmanuel@webserver: ~/todo/client          emmanuel@webserver: ~/todo/client/src
/App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
}
"App.css" 38L, 564C
1,1           Top
```

Then paste the required code into [App.css](#):

```
[+]
emmanuel@webserver: ~/todo/client/src
emmanuel@webserver: ~/todo/client
emmanuel@webserver: ~/todo/client/src

cursor: pointer;
}

@media only screen and (min-width: 300px) {
.App {
width: 80%;
}

input {
width: 100%
}

button {
width: 100%;
margin-top: 15px;
margin-left: 0;
}
}

@media only screen and (min-width: 640px) {
.App {
width: 60%;
}

input {
width: 50%;
}

button {
width: 30%;
margin-left: 10px;
margin-top: 0;
}
}

"App.css" 87L, 1023C written
87,1          Bot
```

Exit

In the src directory open the **index.css**

vim index.css

```
emmanuel@webserver: ~/todo/client/src
emmanuel@webserver: ~/todo/client
emmanuel@webserver: ~/todo/client/src

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
  monospace;
}
:;%d
```

Use :%d to delete existing file

Copy and paste the required code:

**Save and exit.**

Go to the Todo directory

```
cd ../../
```

```
emmanuel@webserver:~/todo/client/src$ cd ../../  
emmanuel@webserver:~/todo$
```

Run `npm run dev` from the Todo directory run:

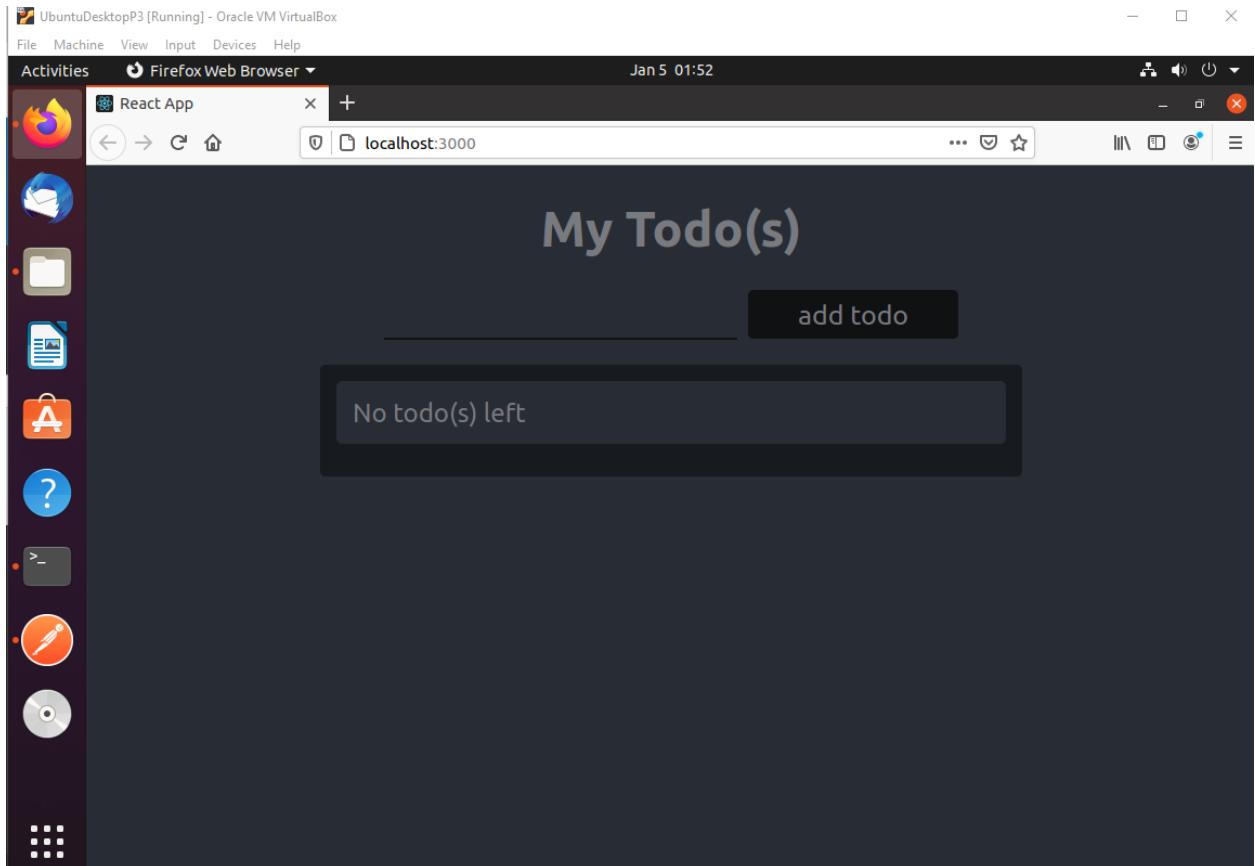
Got the result below

```
emmanuel@webserver:~/todo$ npm run dev
> todo@1.0.0 dev /home/emmanuel/todo
> concurrently "npm run start-watch" "cd client && npm start"

[0]
[0] > todo@1.0.0 start-watch /home/emmanuel/todo
[0] > nodemon index.js
[0]
[1]
[1] > client@0.1.0 start /home/emmanuel/todo/client
[1] > react-scripts start
[1]
[0] [nodemon] 2.0.6
[0] [nodemon] to restart at any time, enter `rs`
[0] [nodemon] watching path(s): ***!
[0] [nodemon] watching extensions: js,mjs,json
[0] [nodemon] starting `node index.js`
events.js:291
      throw er; // Unhandled 'error' event
      ^
[0]

[0] Error: listen EADDRINUSE: address already in use ::::5000
    at Server.setupListenHandle [as _listen2] (net.js:1316:16)
    at listenInCluster (net.js:1364:12)
    at Server.listen (net.js:1450:7)
    at Function.listen (/home/emmanuel/todo/node_modules/express/lib/application.js:618:24)
    at Object.<anonymous> (/home/emmanuel/todo/index.js:35:5)
    at Module._compile (internal/modules/cjs/loader.js:999:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
    at Module.load (internal/modules/cjs/loader.js:863:32)
    at Function.Module._load (internal/modules/cjs/loader.js:708:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12)
    at internal/main/run_main_module.js:17:47
[0] Emitted 'error' event on Server instance at:
    at emitErrorNT (net.js:1343:8)
```

```
[0] Emitted 'error' event on Server instance at:
[0]   at emitErrorNT (net.js:1343:8)
[0]   at processTicksAndRejections (internal/process/task_queues.js:84:21) {
[0]     code: 'EADDRINUSE',
[0]     errno: 'EADDRINUSE',
[0]     syscall: 'listen',
[0]     address: '::',
[0]     port: 5000
[0]   }
[0] [nodemon] app crashed - waiting for file changes before starting...
[1] i [wds]: Project is running at http://192.168.0.11/
[1] i [wds]: webpack output is served from
[1] i [wds]: Content not from webpack is served from /home/emmanuel/todo/client/public
[1] i [wds]: 404s will fallback to /
[1] Starting the development server...
[1]
[1] Compiled successfully!
[1]
[1] You can now view client in the browser.
[1]
[1] Local:          http://localhost:3000
[1] On Your Network: http://192.168.0.11:3000
[1]
[1] Note that the development build is not optimized.
[1] To create a production build, use npm run build.
```



I was unable to add items to my todo app. The app seemed not to be working. I checked the terminal and got the following messages for each time i tried to add items to my todo list

```
Compiled successfully!

You can now view client in the browser.

Local:          http://localhost:3000
On Your Network:  http://192.168.0.11:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

Proxy error: Could not proxy request /api/todos from localhost:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from localhost:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).

Proxy error: Could not proxy request /api/todos from 192.168.0.11:3000 to http://localhost:5000.
See https://nodejs.org/api/errors.html#errors_common_system_errors for more information (ECONNRESET).
```

I tried to troubleshoot by going back to the app.js file. I ran vim app.js and realized that the file was empty so I copied and pasted the required code again.

Ran `npm run dev` again with the following result

```
1] Compiled successfully!
1]
1] You can now view client in the browser.
1]
1] Local:          http://localhost:3000
1] On Your Network: http://192.168.0.11:3000
1]
1] Note that the development build is not optimized.
1] To create a production build, use npm run build.
1]
0] MongooseError: Operation `todos.find()` buffering timed out after 10000ms
0]     at Timeout.<anonymous> (/home/emmanuel/todo/node_modules/mongoose/lib/drivers/node-mongodb-native/collection.j
::184:20)
0]         at listOnTimeout (internal/timers.js:554:17)
0]         at processTimers (internal/timers.js:497:7)
0] MongooseError: Operation `todos.insertOne()` buffering timed out after 10000ms
0]     at Timeout.<anonymous> (/home/emmanuel/todo/node_modules/mongoose/lib/drivers/node-mongodb-native/collection.j
::184:20)
0]         at listOnTimeout (internal/timers.js:554:17)
0]         at processTimers (internal/timers.js:497:7)
0] MongooseError: Operation `todos.insertOne()` buffering timed out after 10000ms
0]     at Timeout.<anonymous> (/home/emmanuel/todo/node_modules/mongoose/lib/drivers/node-mongodb-native/collection.j
::184:20)
0]         at listOnTimeout (internal/timers.js:554:17)
0]         at processTimers (internal/timers.js:497:7)
0] MongooseError: Operation `todos.insertOne()` buffering timed out after 10000ms
0]     at Timeout.<anonymous> (/home/emmanuel/todo/node_modules/mongoose/lib/drivers/node-mongodb-native/collection.j
::184:20)
0]         at listOnTimeout (internal/timers.js:554:17)
0]         at processTimers (internal/timers.js:497:7)
0] MongooseError: Operation `todos.insertOne()` buffering timed out after 10000ms
0]     at Timeout.<anonymous> (/home/emmanuel/todo/node_modules/mongoose/lib/drivers/node-mongodb-native/collection.j
::184:20)
0]         at listOnTimeout (internal/timers.js:554:17)
0]         at processTimers (internal/timers.js:497:7)
```

I still am not able to add items on the todo app on my browser.

I ran **npm run build** as below

```
emmanuel@webserver:~$ cd todo
emmanuel@webserver:~/todo$ npm run build
```

Result

```
emmanuel@webserver:~/todo$ npm run build
npm ERR! missing script: build

npm ERR! A complete log of this run can be found in:
npm ERR!     /home/emmanuel/.npm/_logs/2021-01-07T00_42_21_176Z-debug.log
emmanuel@webserver:~/todo$
```

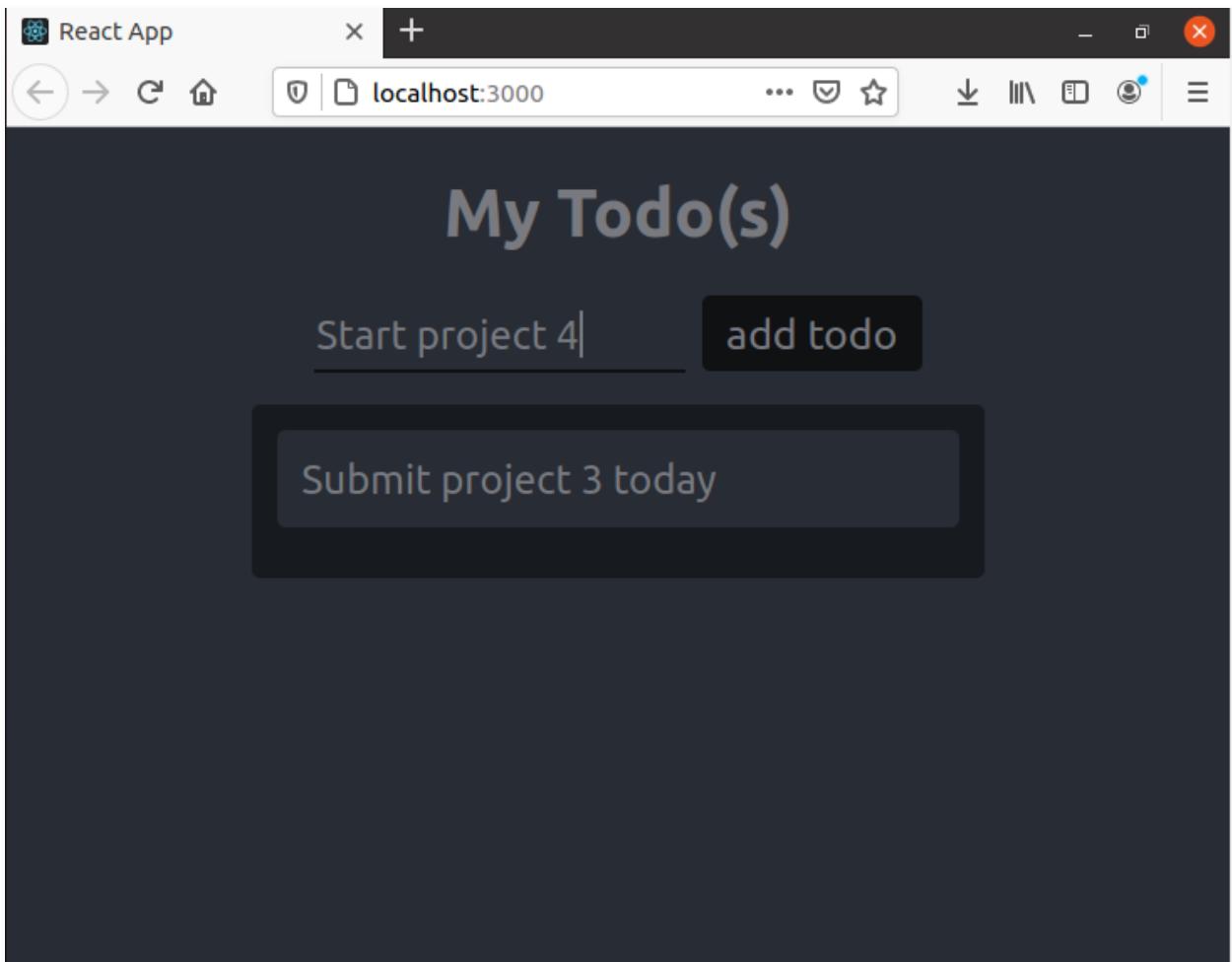
At this point I have exhausted troubleshooting list and had to reinstall the virtual machine and start project 3 again.

I realized while redoing this project that I missed a few lines while editing my package.json file in the client folder in an attempt to add the key value pair

```
emmanuel@webserver:~/Todo$ npm run dev

> todo@1.0.0 dev /home/emmanuel/Todo
> concurrently "npm run start-watch" "cd client && npm start"

[0]
[0] > todo@1.0.0 start-watch /home/emmanuel/Todo
[0] > nodemon index.js
[0]
[1]
[1] > client@0.1.0 start /home/emmanuel/Todo/client
[1] > react-scripts start
[1]
[0] [nodemon] 2.0.7
[0] [nodemon] to restart at any time, enter `rs`
[0] [nodemon] watching path(s): ***!
[0] [nodemon] watching extensions: js,mjs,json
[0] [nodemon] starting `node index.js`
[0] Server running on port 5000
[0] (node:46339) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[0] Database connected successfully
[1] i 「wds」: Project is running at http://10.0.2.15/
[1] i 「wds」: webpack output is served from
[1] i 「wds」: Content not from webpack is served from /home/emmanuel/Todo/client/public
[1] i 「wds」: 404s will fallback to /
[1] Starting the development server...
[1]
[1] Compiled successfully!
[1]
[1] You can now view client in the browser.
[1]
[1] Local:          http://localhost:3000
[1] On Your Network: http://10.0.2.15:3000
[1]
[1] Note that the development build is not optimized.
[1] To create a production build, use npm run build.
[1]
```



In this tutorial, I gained insight into how to create a todo app using the MERN stack. This was done using a frontend application (React) that communicates with a backend application written using expressjs. I also learnt how to create a Mongodb backend for storing tasks in a database.