StudyOwl's

Node JS Revise Document

Introduction

Node JS Asynchronous and Event Driven − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call

- Create a js file named `app.js` on your computer having the following code.

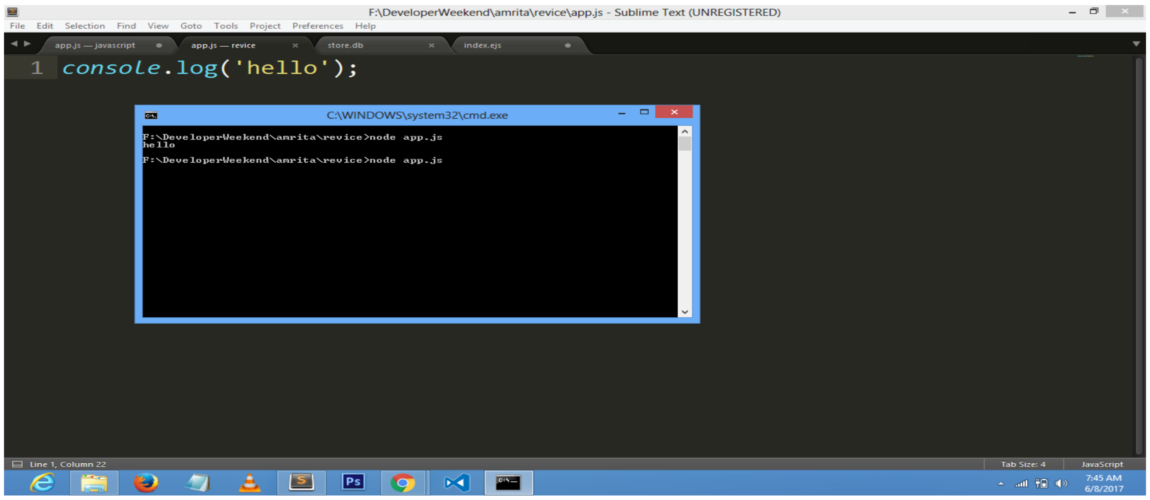
```
/* Hello, World! program in node.js */
console.log("Hello, World!")
```

Now execute main.js file using Node.js interpreter to see the result:

```
node app.js
```

If everything is fine with your installation, this should produce the following result in console:

```
Hello, World
```



```
console.log('hello');
```

```
F:\DeveloperWeekend\amrita\revice>node app.js
hello

F:\DeveloperWeekend\amrita\revice>node app.js
```

F:\DeveloperWeekend\amrita\revice\app.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.js — javascript
app.js — revice
store.db
index.ejs
C:\WINDOWS\system32\cmd.exe
Line 1, Column 22
Tab Size: 4
JavaScript
7:45 AM
6/8/2017

NPM

NPM is a package manager for Node.js packages, or modules if you like. www.npmjs.com hosts thousands of free packages to download and use. The NPM program is installed on your computer when you install Node.js

What is a Package?

A package in Node.js contains all the files you need for a module. Modules are JavaScript libraries you can include in your project.

Download a Package

Downloading a package is very easy.

Open the command line interface and tell NPM to download the package you want.

I want to download a package called **"express"**:

Download "express":

```
C:\Users\Your Name>npm install express
```

Now you have downloaded and installed you first package!

NPM creates a folder named "node_modules", where the package will be placed. All packages you install in the future will be placed in this folder.

your project now has a folder structure like this:
C:\Users\Your Project Folder\node_modules\express

Using a Package

Once the package is installed, it is ready to use.

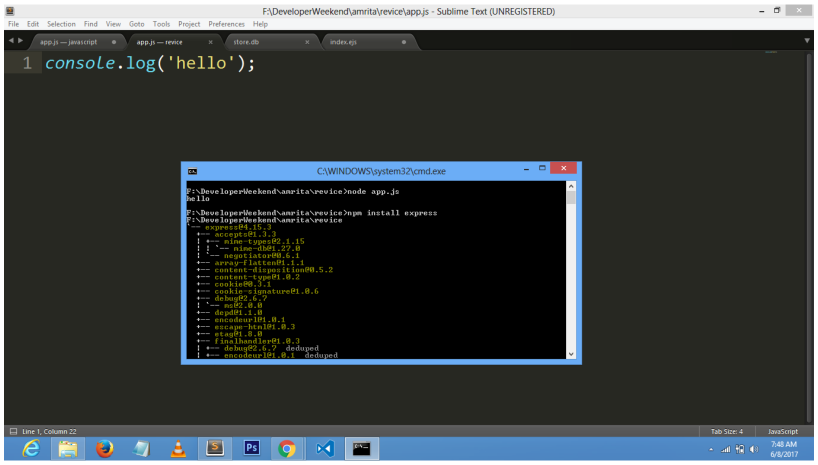
Using Express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework −

- Allows to set up middlewares to respond to HTTP Requests.
- Defines a routing table which is used to perform different actions based on HTTP Method and URL.
- Allows to dynamically render HTML Pages based on passing arguments to templates.

Firstly, install the Express framework globally using NPM so that it can be used to create a web application using node terminal.

```
$ npm install express
```


F:\DeveloperWeekend\amrita\revice\app.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.js — javascript
app.js — revice
store.db
index.ejs
C:\WINDOWS\system32\cmd.exe
Line 1, Column 22
Tab Size: 4
JavaScript
7:48 AM
6/8/2017

```
console.log('hello');
```

```
F:\DeveloperWeekend\amrita\revice>node app.js
hello

F:\DeveloperWeekend\amrita\revice>npm install express
F:\DeveloperWeekend\amrita\revice
`-- express@4.15.3
  +-- accepts@1.3.3
  ¦ +-- mime-types@2.1.15
  ¦ ¦ `-- mime-db@1.27.0
  ¦ `-- negotiator@0.6.1
  +-- array-flatten@1.1.1
  +-- content-disposition@0.5.2
  +-- content-type@1.0.2
  +-- cookie@0.3.1
  +-- cookie-signature@1.0.6
  +-- debug@2.6.7
  ¦ `-- ms@2.0.0
  +-- depd@1.1.0
  +-- encodeurl@1.0.1
  +-- escape-html@1.0.3
  +-- etag@1.8.0
  +-- finalhandler@1.0.3
  ¦ +-- debug@2.6.7  deduped
  ¦ +-- encodeurl@1.0.1  deduped
```

Hello world Example with Express

Following is a very basic Express app which starts a server and listens on port 3000 for connection. This app responds with **Hello World!** for requests to the homepage. For every other path, it will respond with a **404 Not Found.**

Once the package is installed, it is ready to use.
Include the "express" package the same way you include any other module:

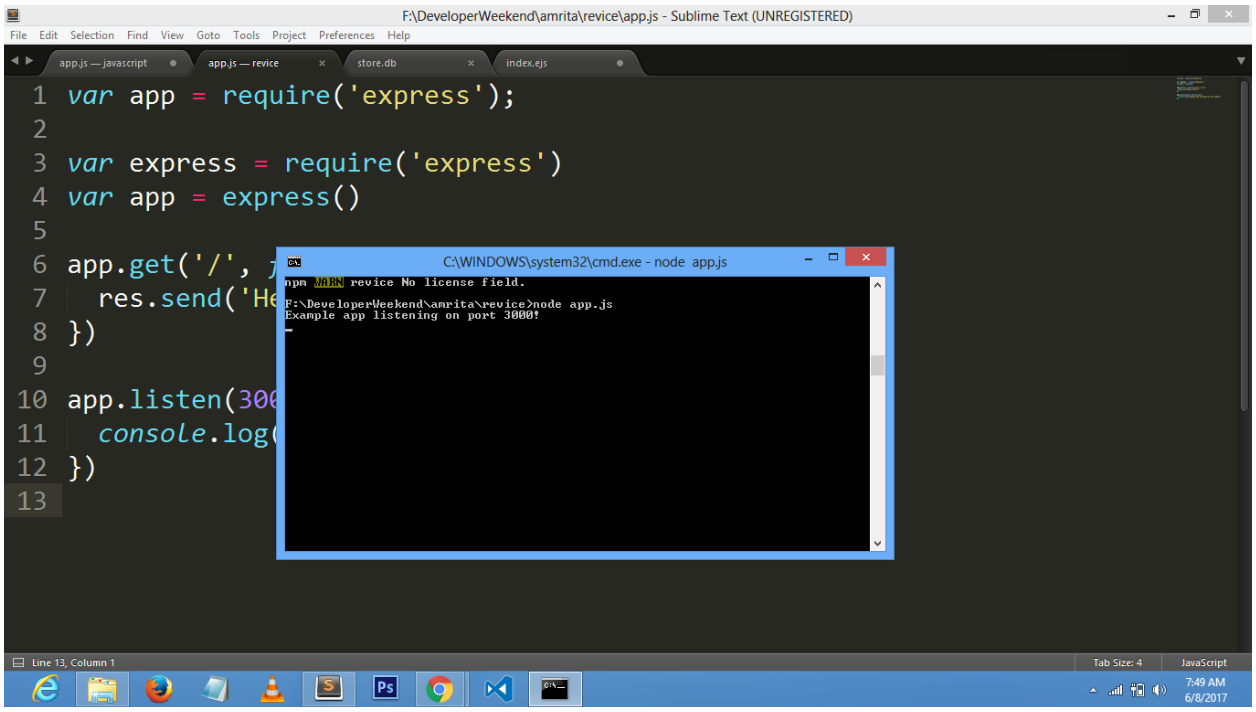
```
var app = require('express');

var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('Hello World!')
})

app.listen(3000, function () {
  console.log('Example app listening on port 3000!')
})
```

Above is a very basic Express app which starts a server and listens on port 3000 for connection. This app responds with **Hello World!** for requests to the homepage. For every other path, it will respond with a **404 Not Found.**

Run the app with the following command:

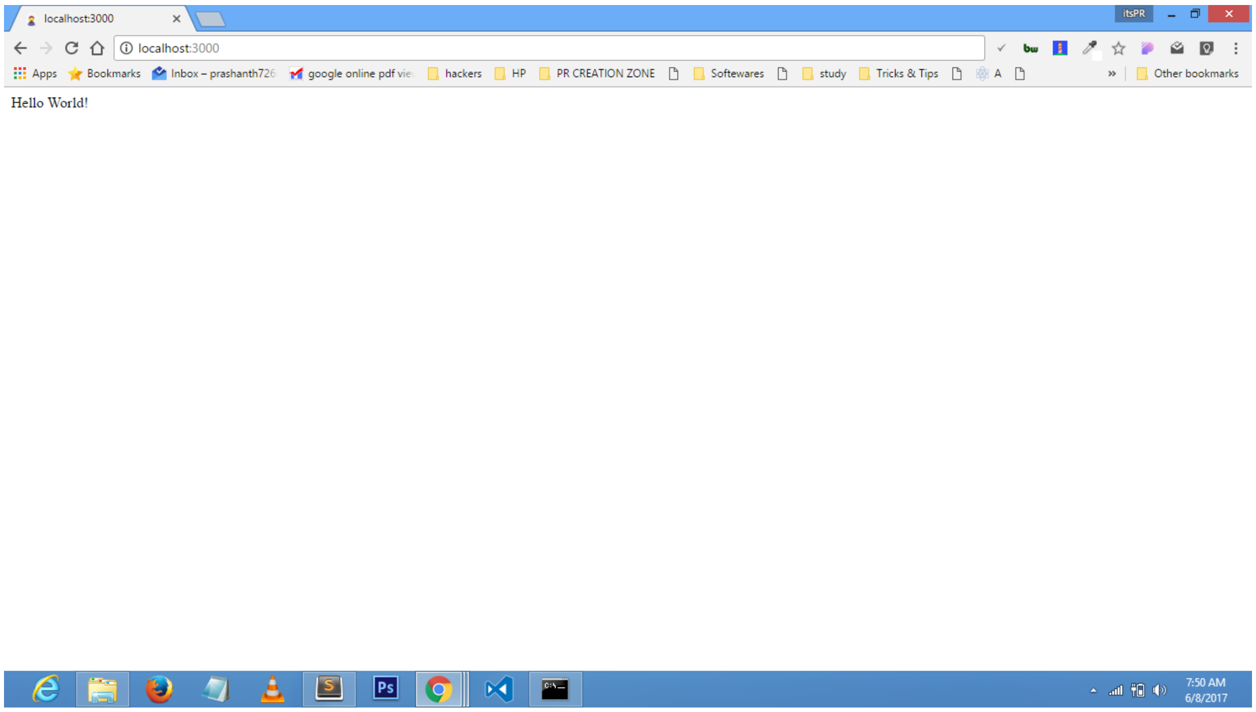
```
node main.js
```

Then, load `http://localhost:3000/` in a browser to see the output.


F:\DeveloperWeekend\amrita\revice\app.js - Sublime Text (UNREGISTERED)
C:\WINDOWS\system32\cmd.exe - node app.js

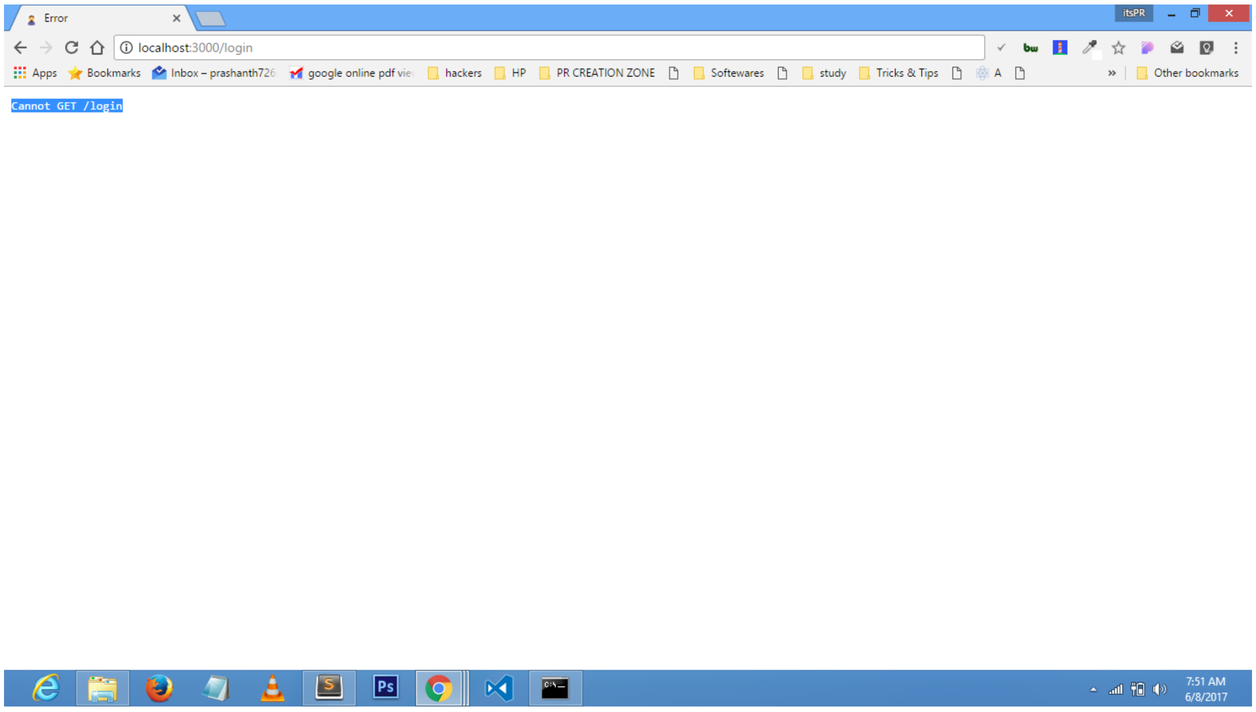
```
var app = require('express');

var express = require('express')
var app = express()

app.get('/', f
  res.send('He
})

app.listen(300
  console.log(
})
```

```
npm WARN revice No license field.

F:\DeveloperWeekend\amrita\revice>node app.js
Example app listening on port 3000!
_
```



```
Hello World!
```

Displaying Hello World in main route.



```
Cannot GET /login
```

Displaying Cannot GET /login Error. Since we haven't created the login route.

Request & Response

Express application uses a callback function whose parameters are **request** and **response** objects.

```
app.get('/', function (req, res) {

})
```

- Request Object – The request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on.
- Response Object – The response object represents the HTTP response that an Express app sends when it gets an HTTP request.

You can print **req** and **res** objects which provide a lot of information related to HTTP request and response including cookies, sessions, URL, etc.

Basic routing

Routing refers to determining how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and so on). Each route can have one or more handler functions, which are executed when the route is matched.

Route definition takes the following structure:

app.METHOD(PATH, HANDLER)

Where:

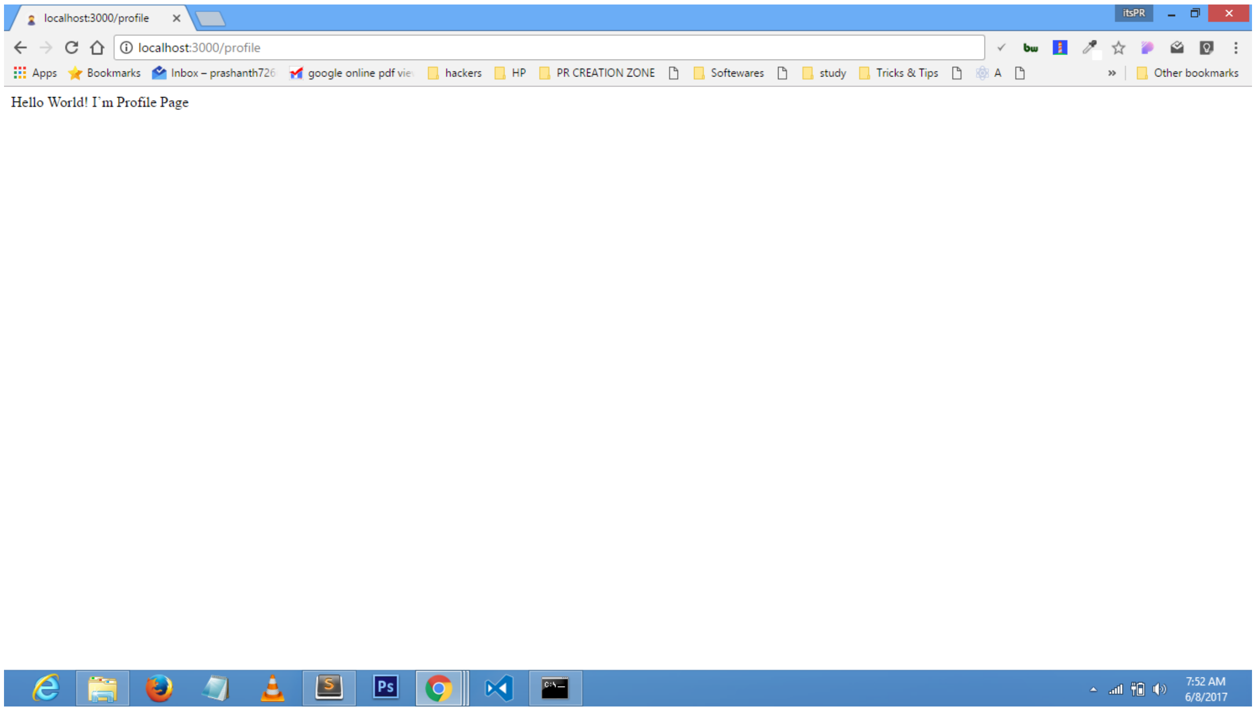
- app is an instance of express.
- METHOD is an HTTP request method, in lowercase.
- PATH is a path on the server.
- HANDLER is the function executed when the route is matched.

The following examples illustrate defining simple routes.
Respond with Hello World! on the homepage route:

```
app.get('/', function (req, res) {
  res.send('Hello World!')
})
```

Respond with Hello World I’m Profile Page! on the profile route:

```
app.get('/profile', function (req, res) {
  res.send('Hello World! I’m Profile Page')
})
```



Hello World! I'm Profile Page

Serving Static Files in Express

Express provides a built-in middleware **express.static** to serve static files, such as images, CSS, JavaScript, etc.

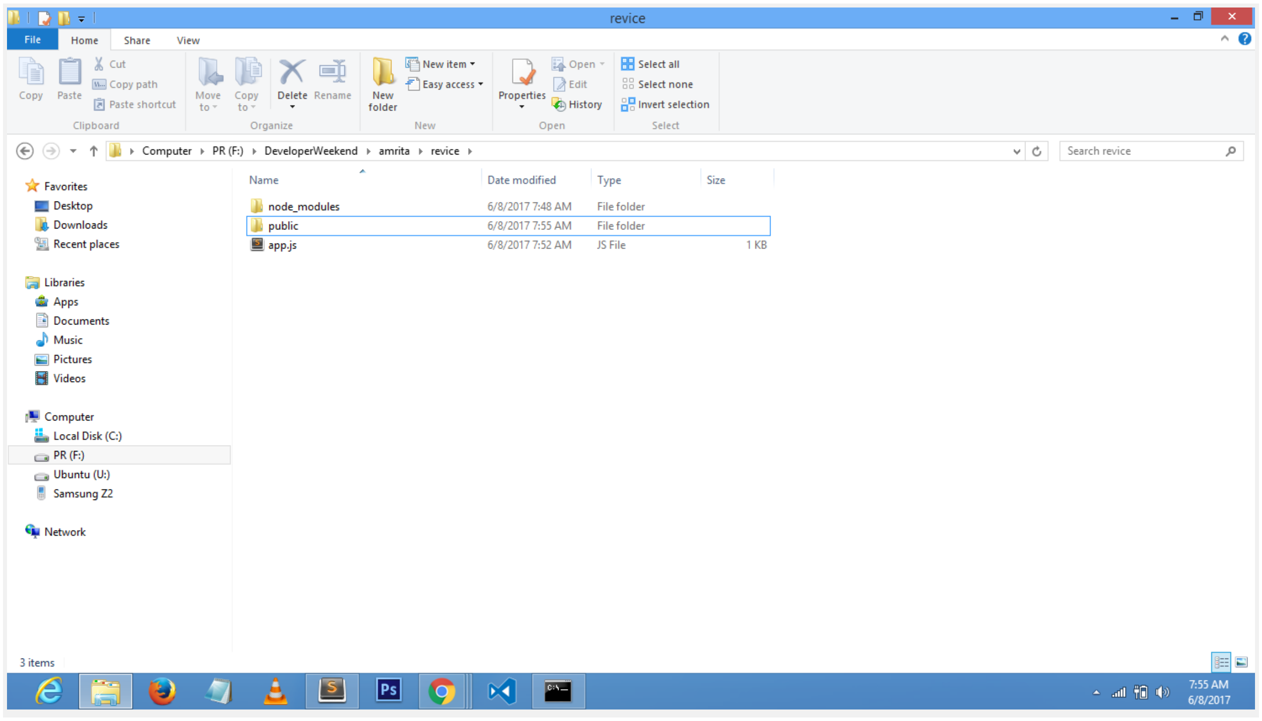
You simply need to pass the name of the directory where you keep your static assets, to the **express.static** middleware to start serving the files directly. For example, if you keep your images, CSS, and JavaScript files in a directory named public, you can do this –

```
app.use(express.static('public'));
```

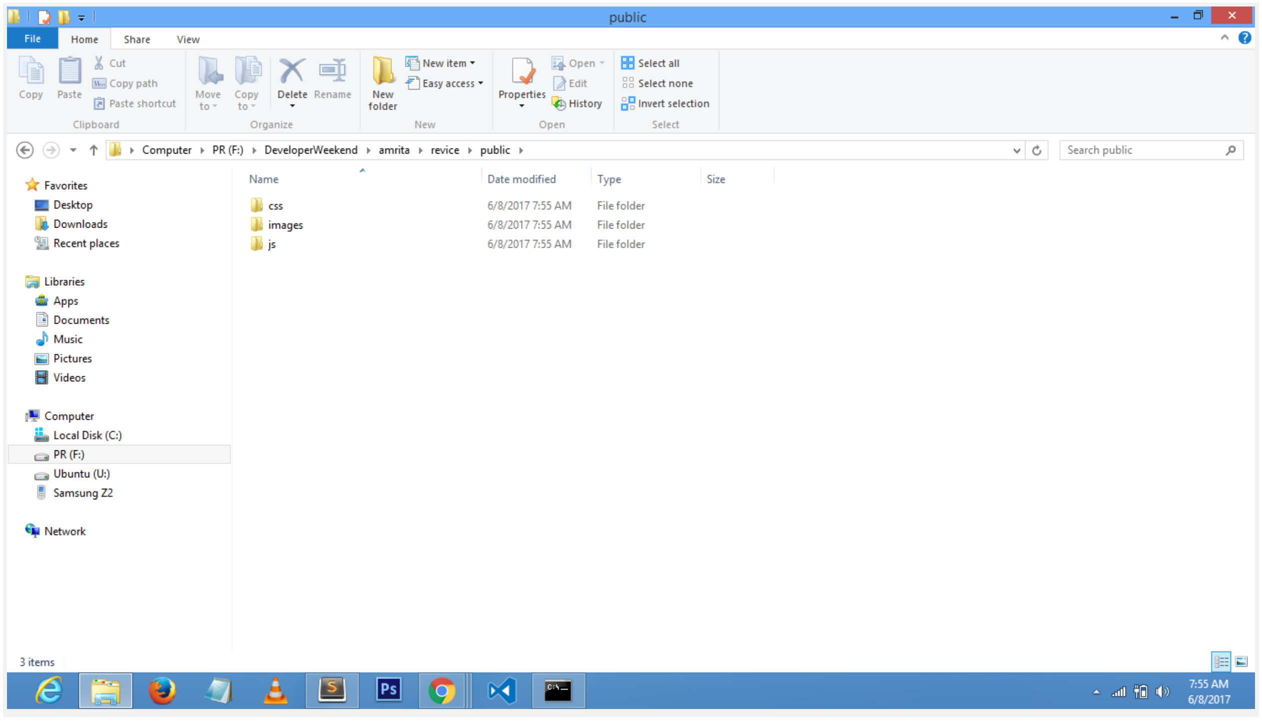
We will keep a few images in **public** sub-directory as follows –

```
node_modules
server.js
public/index.html
public/images
public/images/logo.png
```

```
File Structure to be maintained as follows
```



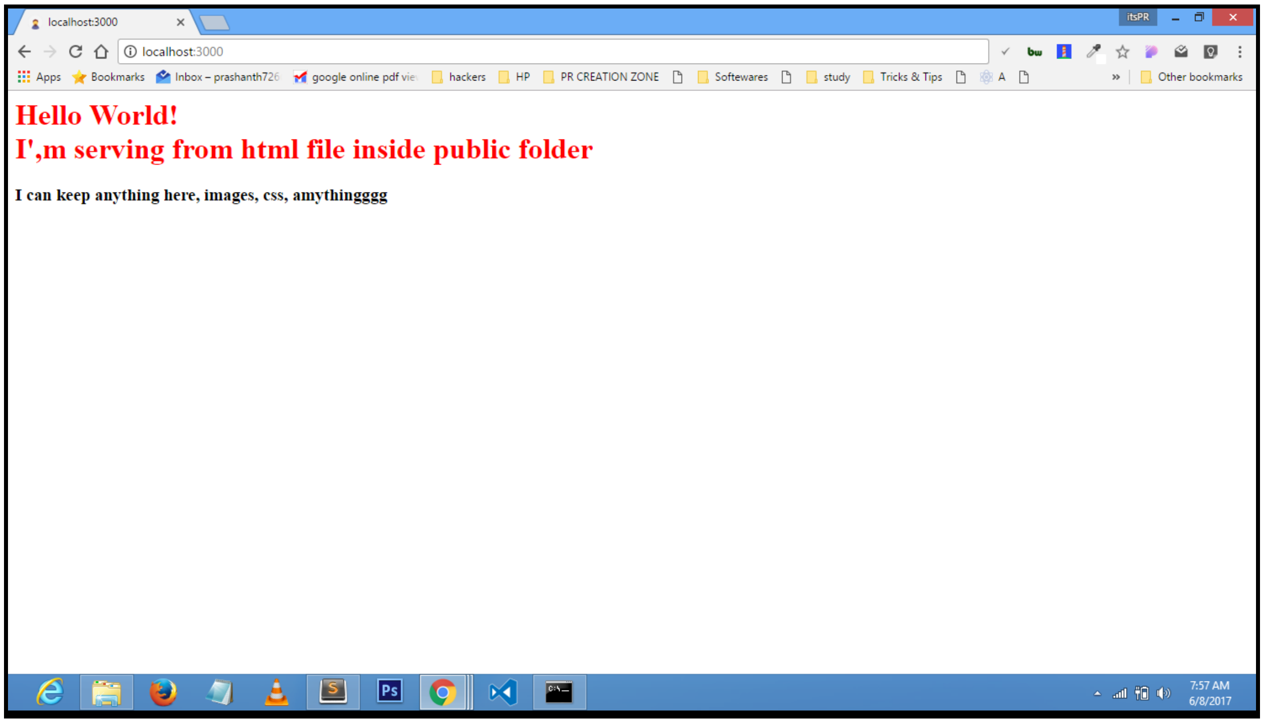
Name	Date modified	Type	Size
node_modules	6/8/2017 7:48 AM	File folder	
public	6/8/2017 7:55 AM	File folder	
app.js	6/8/2017 7:52 AM	JS File	1 KB



Name	Date modified	Type	Size
css	6/8/2017 7:55 AM	File folder	
images	6/8/2017 7:55 AM	File folder	
js	6/8/2017 7:55 AM	File folder	

Let's modify "Hello World" app to add the functionality to handle static files

```
var app = require('express');

var express = require('express')
var app = express()

app.use(express.static('public'));


app.get('/', function (req, res) {
  res.sendFile( __dirname + "/public/" + "index.html" );


})

app.listen(3000, function () {
  console.log('Example app listening on port 3000!')
})
```



Hello World!
I',m serving from html file inside public folder

I can keep anything here, images, css, amythingggg

ejs

Embedded JavaScript templates

EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.

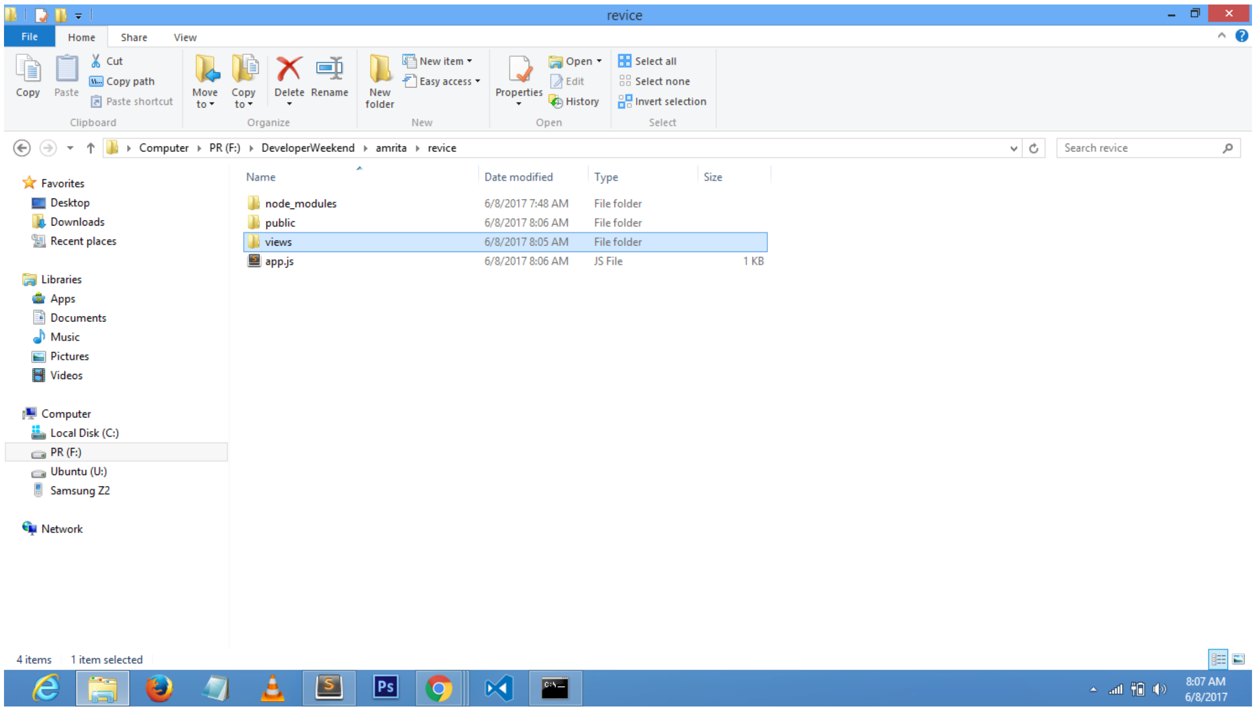
Installation

```
npm install ejs
```

And Create a New folder in route called Views in your project main route

Your file structure should look like this : (From Now on do not create any HTML files, create everything as ejs files)

```
node_modules
server.js
public/css/bootstrap.css
views/index.js
public/images
public/images/logo.png
```

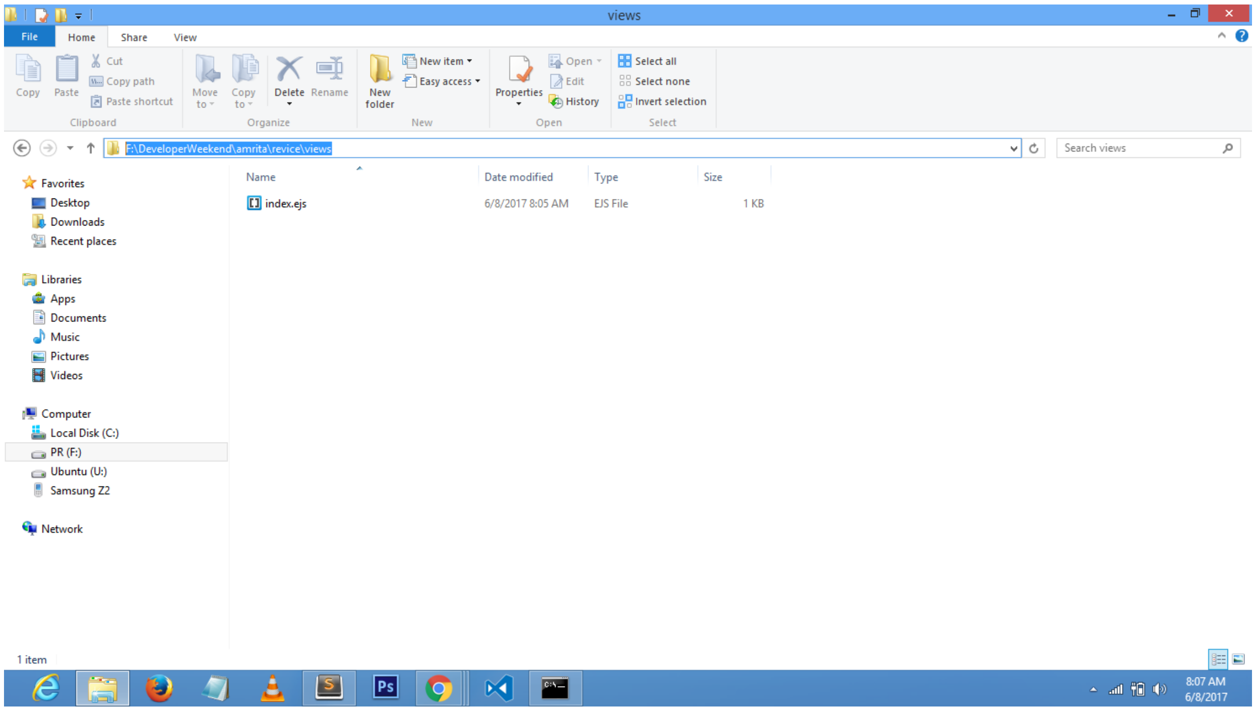


Name	Date modified	Type	Size
node_modules	6/8/2017 7:48 AM	File folder	
public	6/8/2017 8:06 AM	File folder	
views	6/8/2017 8:05 AM	File folder	
app.js	6/8/2017 8:06 AM	JS File	1 KB

Create views folder

Usage

Create a new ejs file with name “index” and save it in views folder.



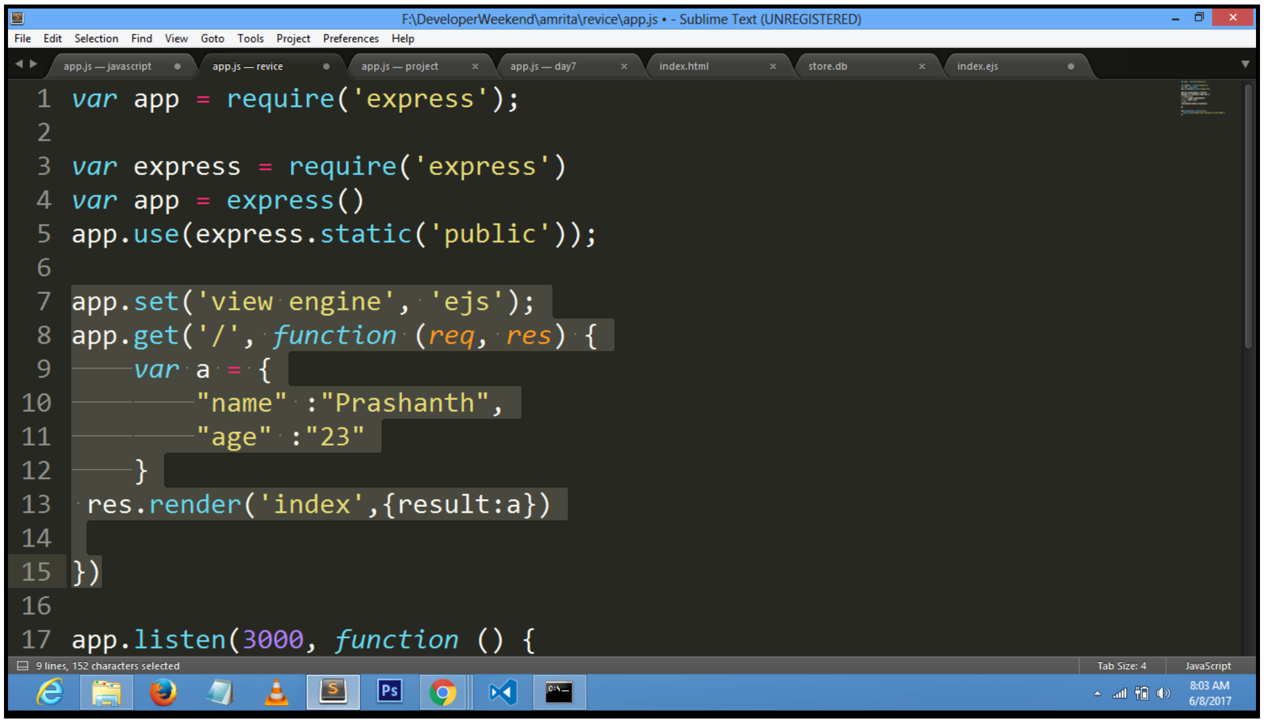
F:\DeveloperWeekend\amrita\revice\views

Name	Date modified	Type	Size
index.ejs	6/8/2017 8:05 AM	EJS File	1 KB

1 item

Code : app.js

```
var app = require('express');
var express = require('express')
var app = express()
app.use(express.static('public'));

app.set('view engine', 'ejs');
app.get('/', function (req, res) {
        var a = {
                "name" :"Prashanth",
                "age" :"23"
        }
 res.render('index',{result:a})
})
app.listen(3000, function () {
  console.log('Example app listening on port 3000!')
})
```

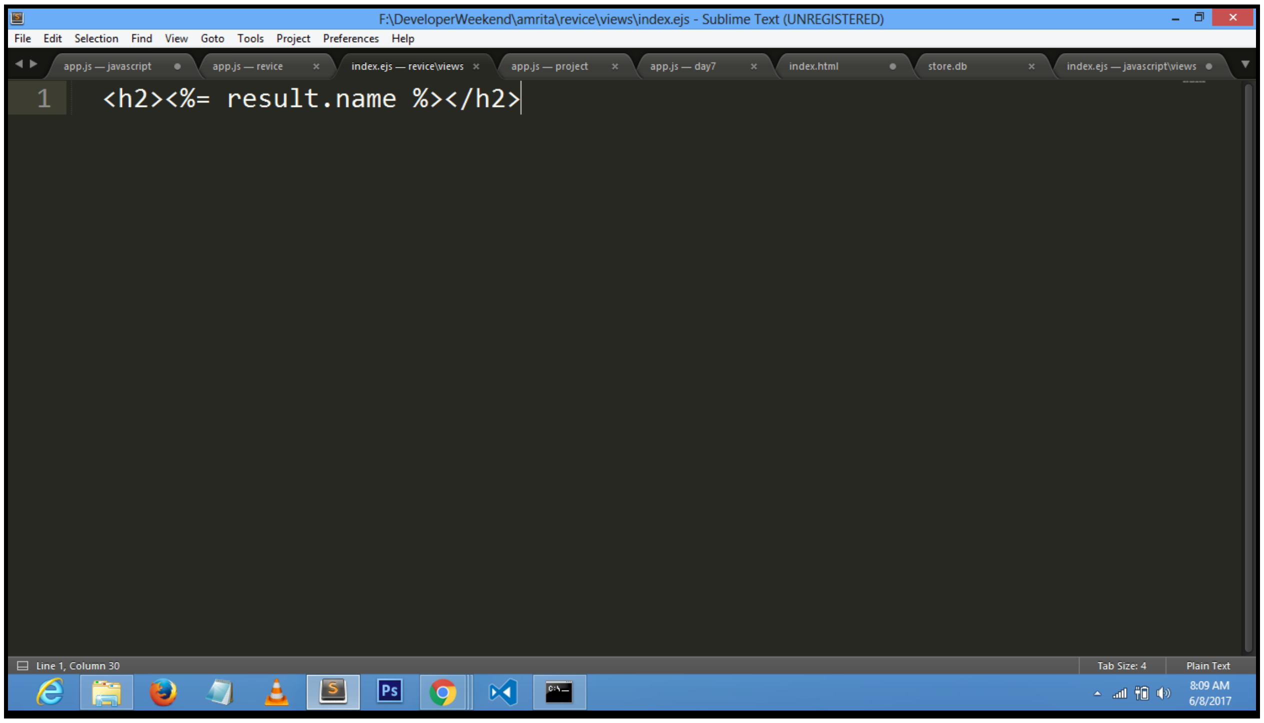

F:\DeveloperWeekend\amrita\revice\app.js • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.js — javascript
app.js — revice
app.js — project
app.js — day7
index.html
store.db
index.ejs
9 lines, 152 characters selected
Tab Size: 4
JavaScript

```
var app = require('express');

var express = require('express')
var app = express()
app.use(express.static('public'));

app.set('view engine', 'ejs');
app.get('/', function (req, res) {
    var a = {
        "name" :"Prashanth",
        "age" :"23"
    }
 res.render('index',{result:a})

})

app.listen(3000, function () {
```

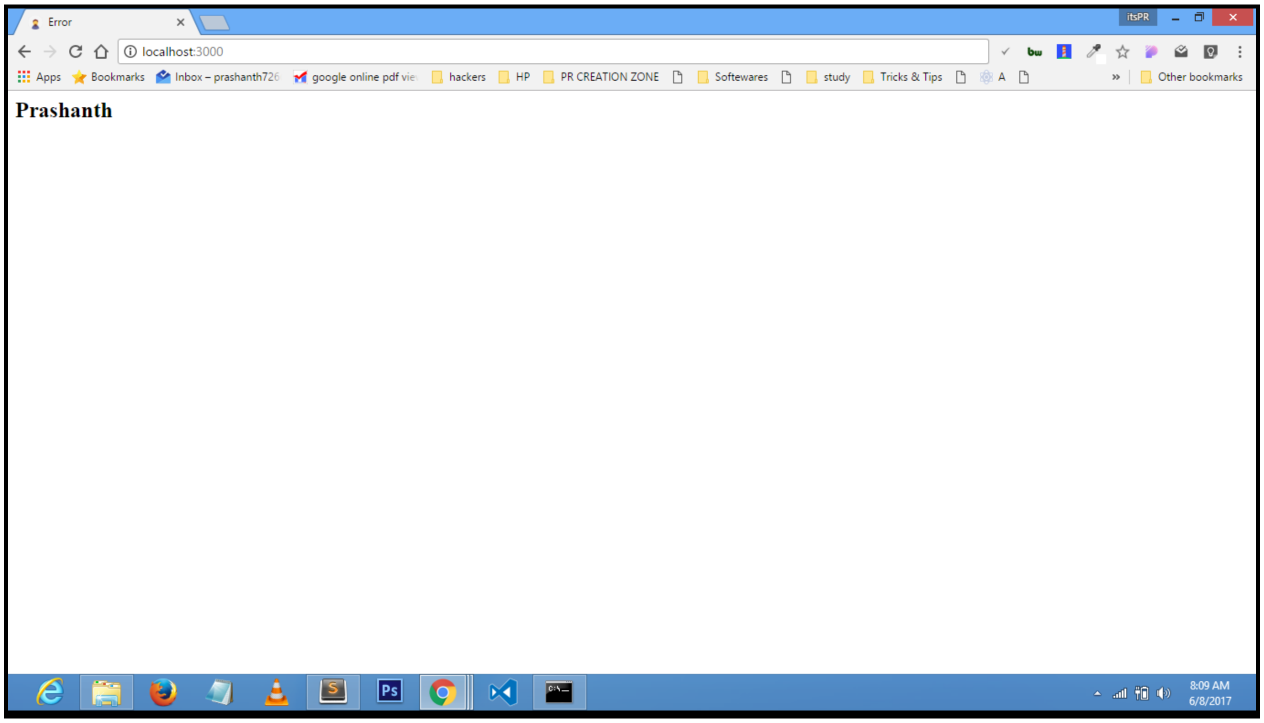
Accessing the passed variables in index ejs file:

Example :(index.ejs)

```
<h2><%= result.name %></h2>
```



```
<h2><%= result.name %></h2>
```



```
Prashanth
```

Result :)

DataBase :

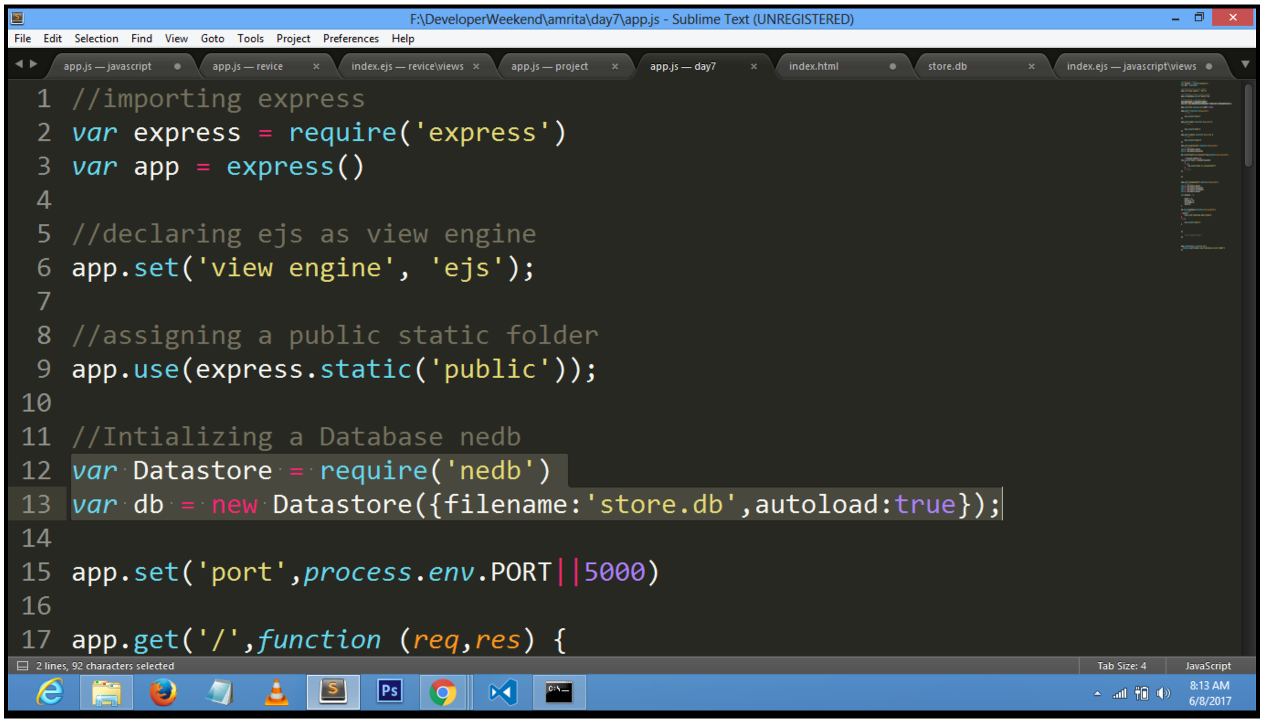
For Database we using a npm module called : **nedb**

Installation

```
npm install nedb
```

Initializing a Database nedb

```
var Datastore = require('nedb')
var db = new Datastore({filename:'store.db',autoload:true});
```

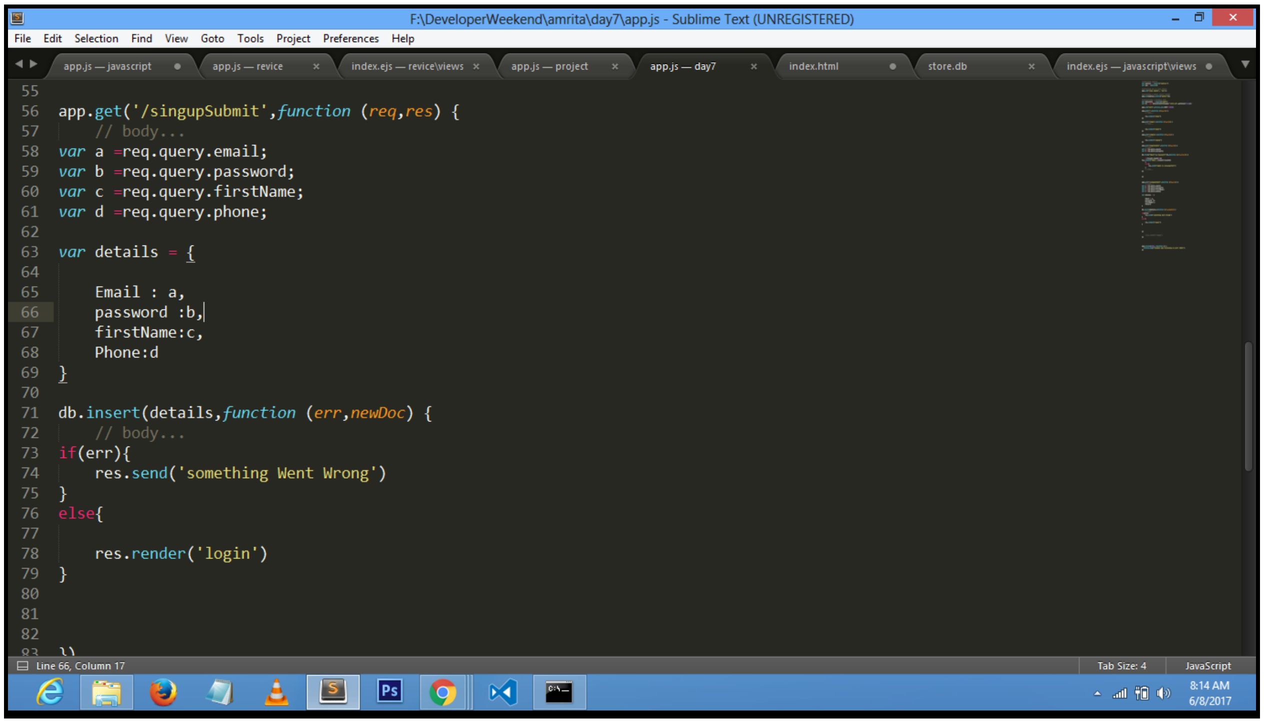


```
//importing express
var express = require('express')
var app = express()

//declaring ejs as view engine
app.set('view engine', 'ejs');

//assigning a public static folder
app.use(express.static('public'));

//Intializing a Database nedb
var Datastore = require('nedb')
var db = new Datastore({filename:'store.db',autoload:true});

app.set('port',process.env.PORT||5000)

app.get('/',function (req,res) {
```

F:\DeveloperWeekend\amrita\day7\app.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.js — javascript
app.js — revice
index.ejs — revice\views
app.js — project
app.js — day7
index.html
store.db
index.ejs — javascript\views
2 lines, 92 characters selected
Tab Size: 4
JavaScript
8:13 AM
6/8/2017

Insert Query :

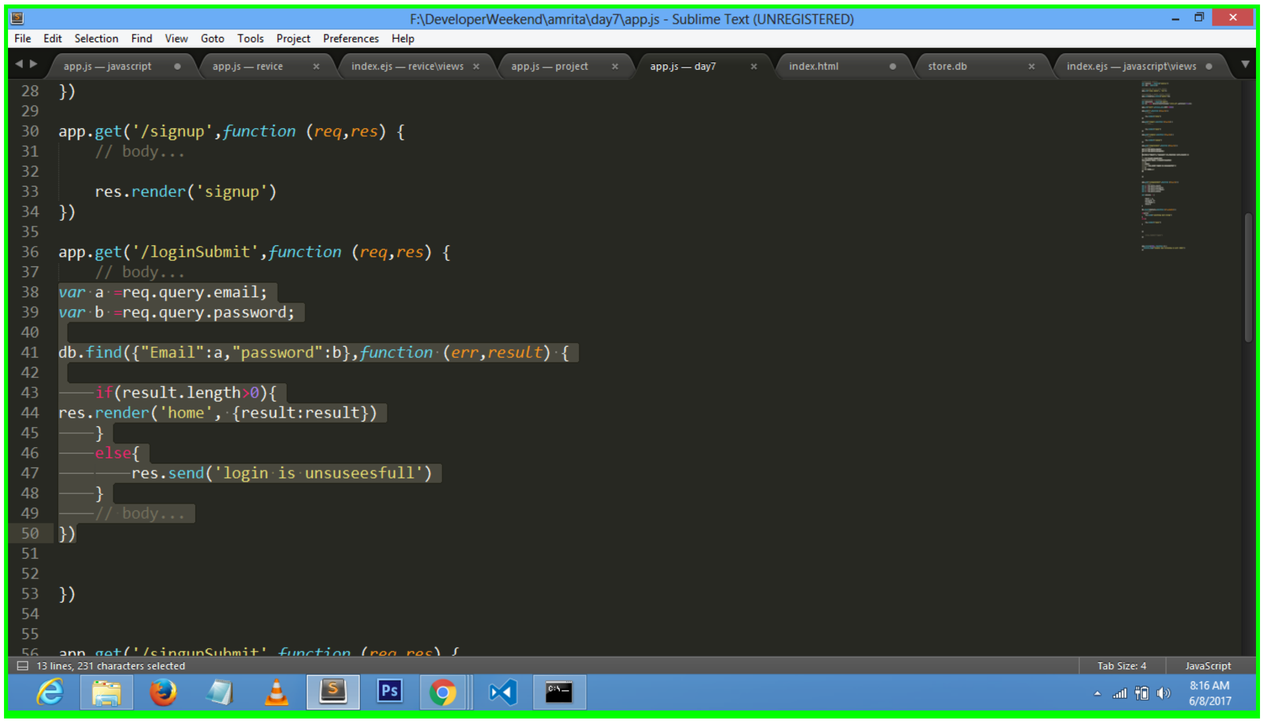
```
var a =req.query.email;
var b =req.query.password;
var c =req.query.firstName;
var d =req.query.phone;

var details = {

        Email : a,
        password :b,
        firstName:c,
        Phone:d
}

db.insert(details,function (err,newDoc) {
        // body...
if(err){
        res.send('something Went Wrong')
}
else{

        res.render('login')
}
})
```


F:\DeveloperWeekend\amrita\day7\app.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.js — javascript
app.js — revice
index.ejs — revice\views
app.js — project
app.js — day7
index.html
store.db
index.ejs — javascript\views
Line 66, Column 17
Tab Size: 4
JavaScript
8:14 AM
6/8/2017

```

app.get('/singupSubmit',function (req,res) {
    // body...
var a =req.query.email;
var b =req.query.password;
var c =req.query.firstName;
var d =req.query.phone;

var details = {

    Email : a,
    password :b,
    firstName:c,
    Phone:d
}

db.insert(details,function (err,newDoc) {
    // body...
if(err){
    res.send('something Went Wrong')
}
else{

    res.render('login')
}



})
```

Find Query :

```
var a =req.query.email;
var b =req.query.password;
db.find({"Email":a,"password":b},function (err,result) {

        if(result.length>0){
res.render('home', {result:result})
        }
        else{
                res.send('login is unsuseesfull')
        }
        // body...
})
```



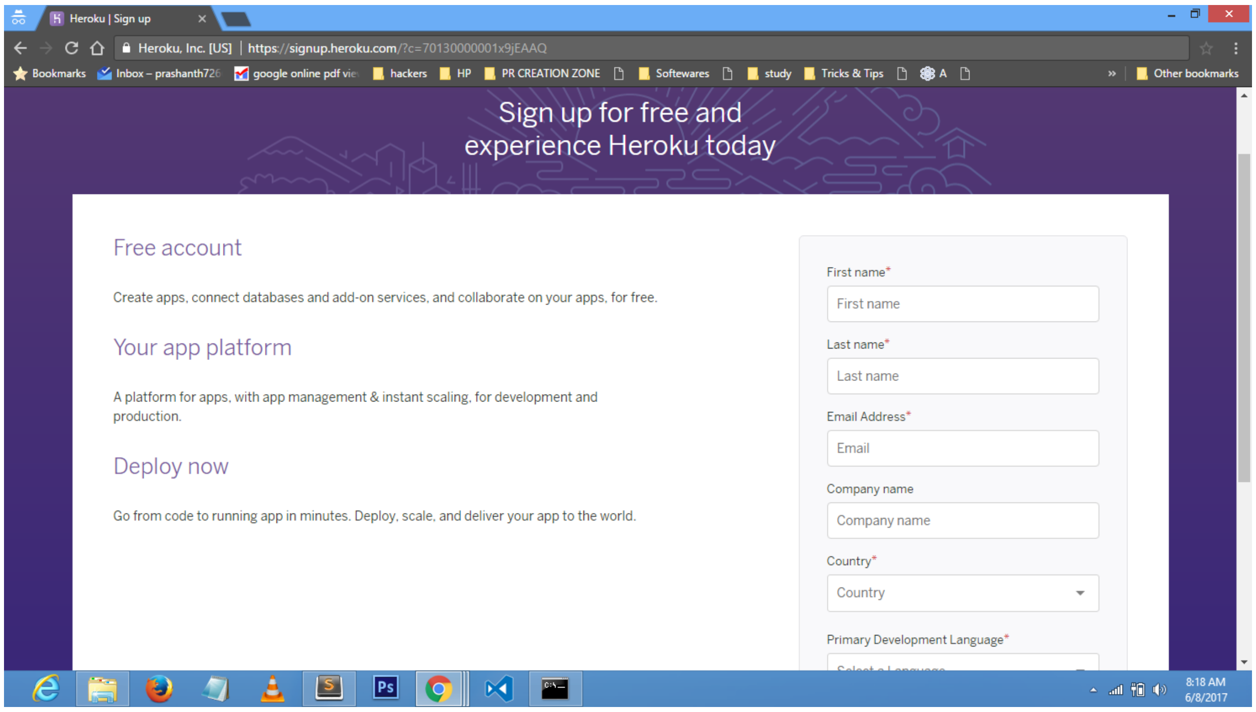
F:\DeveloperWeekend\amrita\day7\app.js - Sublime Text (UNREGISTERED)

```
})

app.get('/signup',function (req,res) {
    // body...

    res.render('signup')
})

app.get('/loginSubmit',function (req,res) {
    // body...
var a =req.query.email;
var b =req.query.password;

db.find({"Email":a,"password":b},function (err,result) {

    if(result.length>0){
res.render('home', {result:result})
    }
    else{
        res.send('login is unsuseesfull')
    }
    // body...
})


})


app.get('/signupSubmit',function (req,res) {
```

13 lines, 231 characters selected

Publishing to Heroku

Step 1 : Create a account in Heroku:



Sign up for free and experience Heroku today

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

Deploy now

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name*
First name

Last name*
Last name

Email Address*
Email

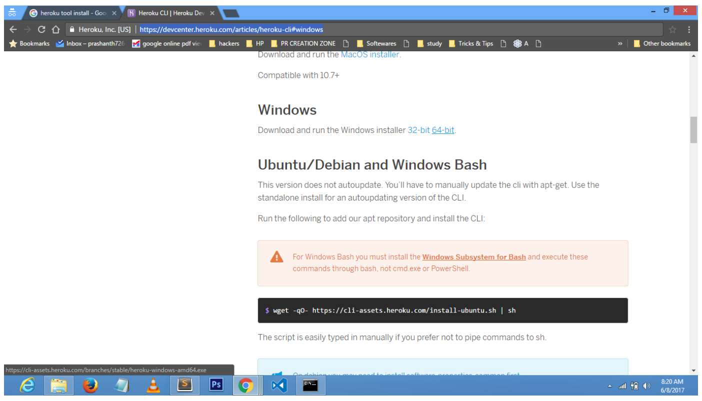
Company name
Company name

Country*
Country

Primary Development Language*

Step 2 : Next Install Heroku Tool in your laptop : by following this link

https://devcenter.heroku.com/articles/heroku-cli#windows



Download and run the MacOS installer.

Compatible with 10.7+

Windows

Download and run the Windows installer 32-bit 64-bit.

Ubuntu/Debian and Windows Bash

This version does not autoupdate. You'll have to manually update the cli with apt-get. Use the standalone install for an autoupdating version of the CLI.

Run the following to add our apt repository and install the CLI:

For Windows Bash you must install the Windows Subsystem for Bash and execute these commands through bash, not cmd.exe or PowerShell.

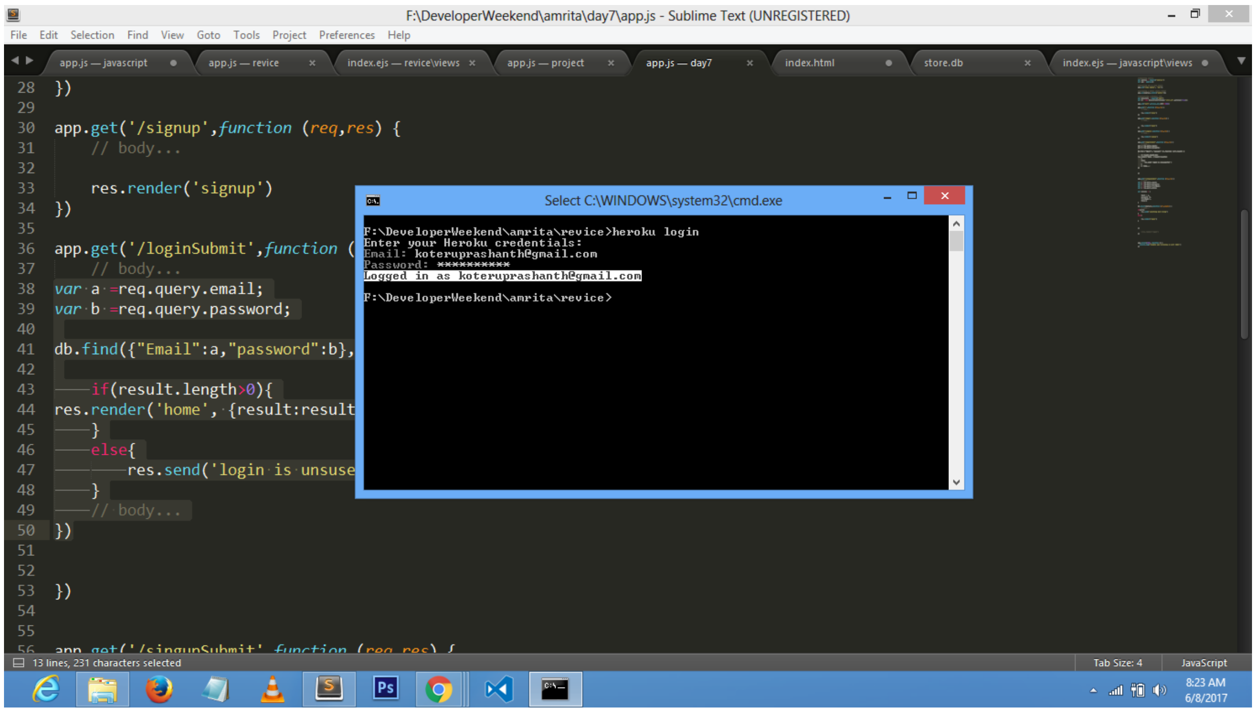
```
$ wget -qO- https://cli-assets.heroku.com/install-ubuntu.sh | sh
```

The script is easily typed in manually if you prefer not to pipe commands to sh.

https://cli-assets.heroku.com/branches/stable/heroku-windows-amd64.exe
8:20 AM
6/8/2017

Step 3 :

Login to heroku in your command prompt : by typing `heroku login`

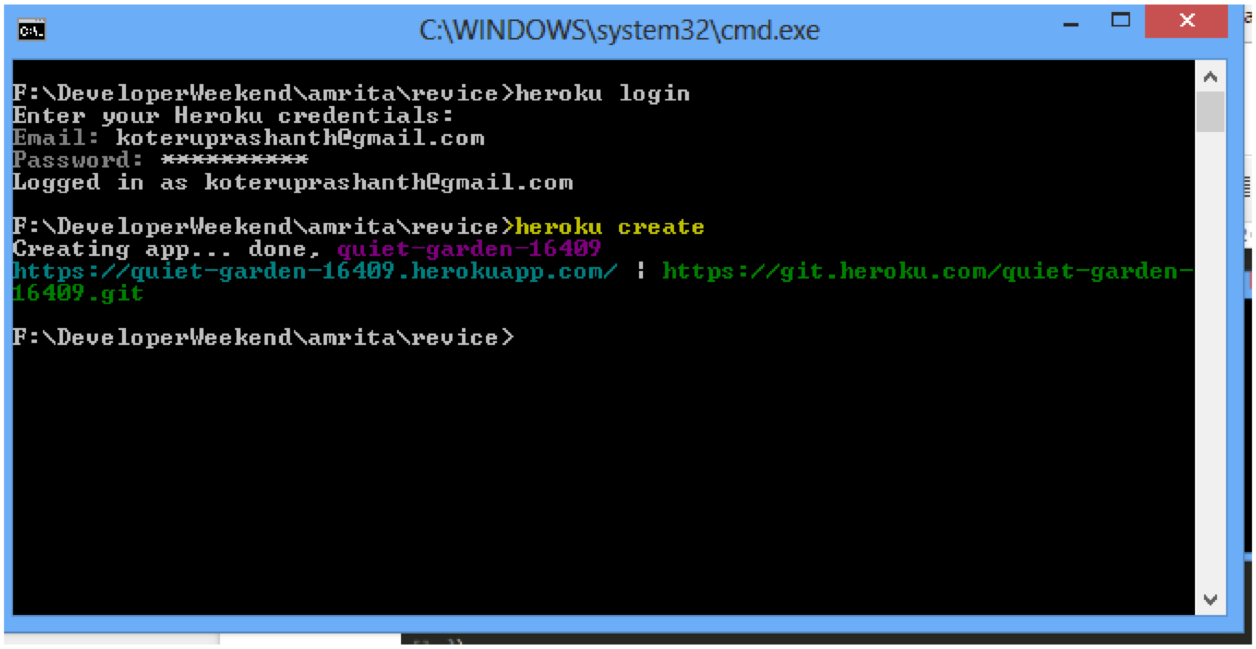

F:\DeveloperWeekend\amrita\day7\app.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
app.js — javascript
app.js — revice
index.ejs — revice\views
app.js — project
app.js — day7
index.html
store.db
index.ejs — javascript\views
Select C:\WINDOWS\system32\cmd.exe
13 lines, 231 characters selected
Tab Size: 4
JavaScript
8:23 AM
6/8/2017

```
})

app.get('/signup',function (req,res) {
    // body...

    res.render('signup')
})

app.get('/loginSubmit',function (
    // body...
var a =req.query.email;
var b =req.query.password;

db.find({"Email":a,"password":b},

    if(result.length>0){
res.render('home', {result:result
    }
    else{
        res.send('login is unsuse
    }
    // body...
})


})
```

```
F:\DeveloperWeekend\amrita\revice>heroku login
Enter your Heroku credentials:
Email: koteruprashanth@gmail.com
Password: ***********
Logged in as koteruprashanth@gmail.com

F:\DeveloperWeekend\amrita\revice>
```

Step 4 :

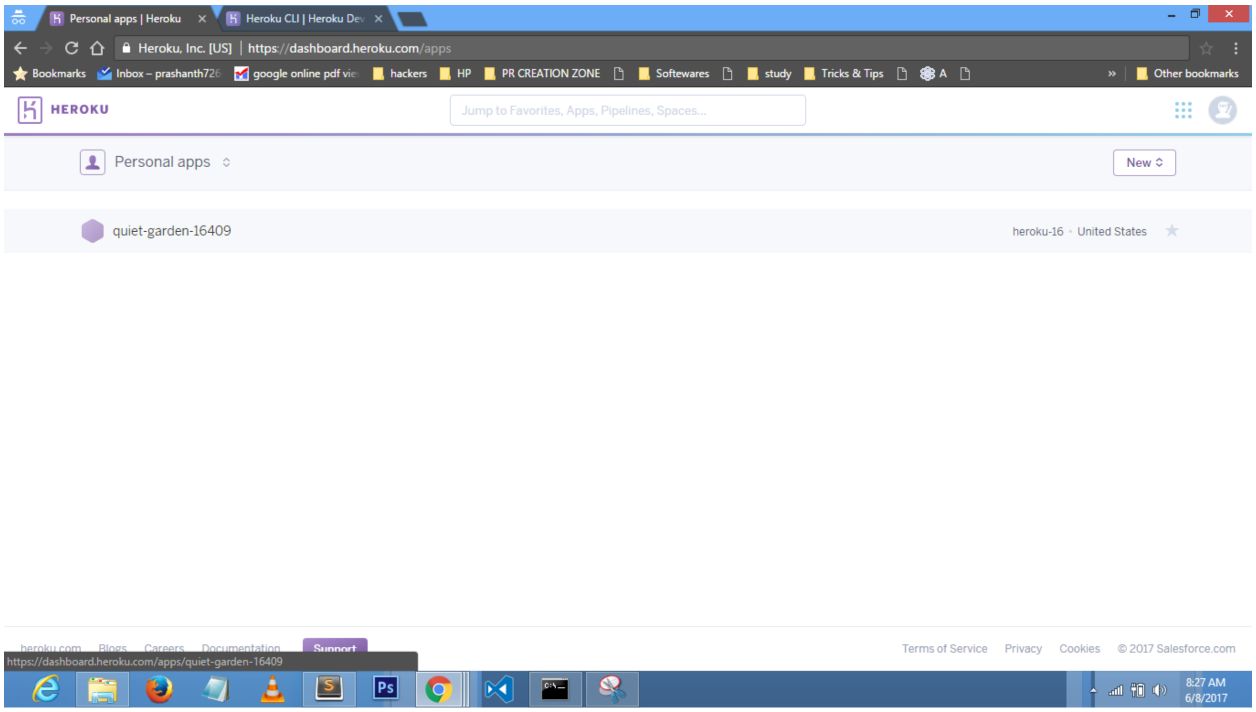
Type `heroku create` in command prompt to create a new application



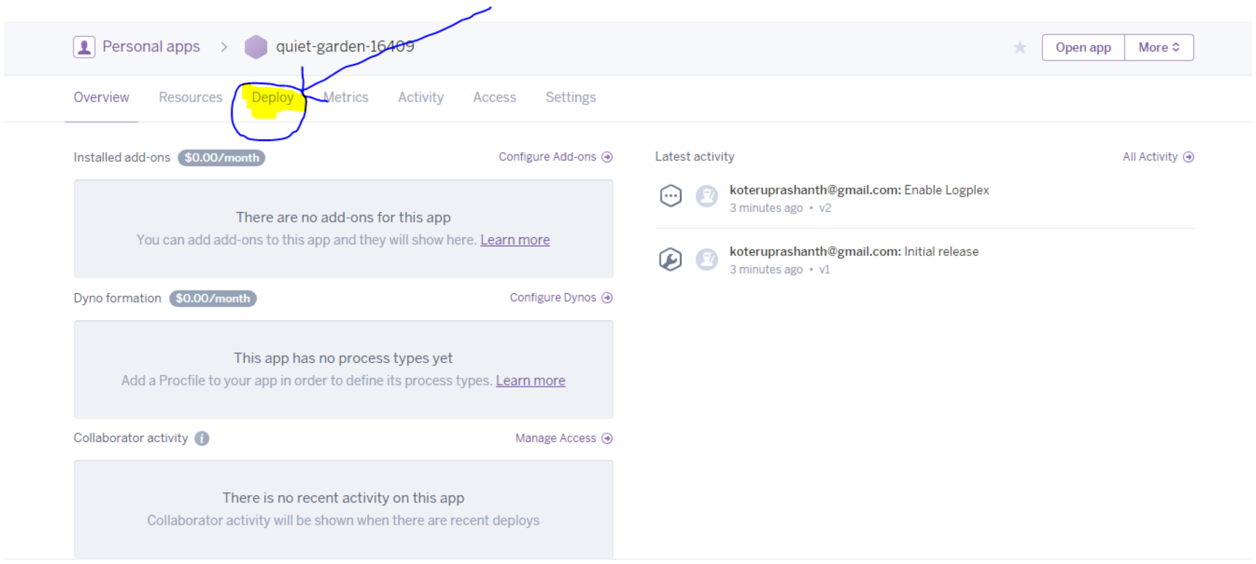
```
F:\DeveloperWeekend\amrita\revice>heroku login
Enter your Heroku credentials:
Email: koteruprashanth@gmail.com
Password: **********
Logged in as koteruprashanth@gmail.com

F:\DeveloperWeekend\amrita\revice>heroku create
Creating app... done, quiet-garden-16409
https://quiet-garden-16409.herokuapp.com/ | https://git.heroku.com/quiet-garden-
16409.git

F:\DeveloperWeekend\amrita\revice>
```

Step 5 :

Login into your heroku dashboard in web, then you can see something similar to this


Personal apps | Heroku
Heroku CLI | Heroku Dev
Heroku, Inc. [US] | https://dashboard.heroku.com/apps
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal apps
New
quiet-garden-16409
heroku-16 · United States
Terms of Service
Privacy
Cookies
© 2017 Salesforce.com
https://dashboard.heroku.com/apps/quiet-garden-16409
8:27 AM
6/8/2017

Click on the app name and go to Deploy tab



There are no add-ons for this app

You can add add-ons to this app and they will show here. Learn more

This app has no process types yet

Add a Procfile to your app in order to define its process types. Learn more

There is no recent activity on this app

Collaborator activity will be shown when there are recent deploys

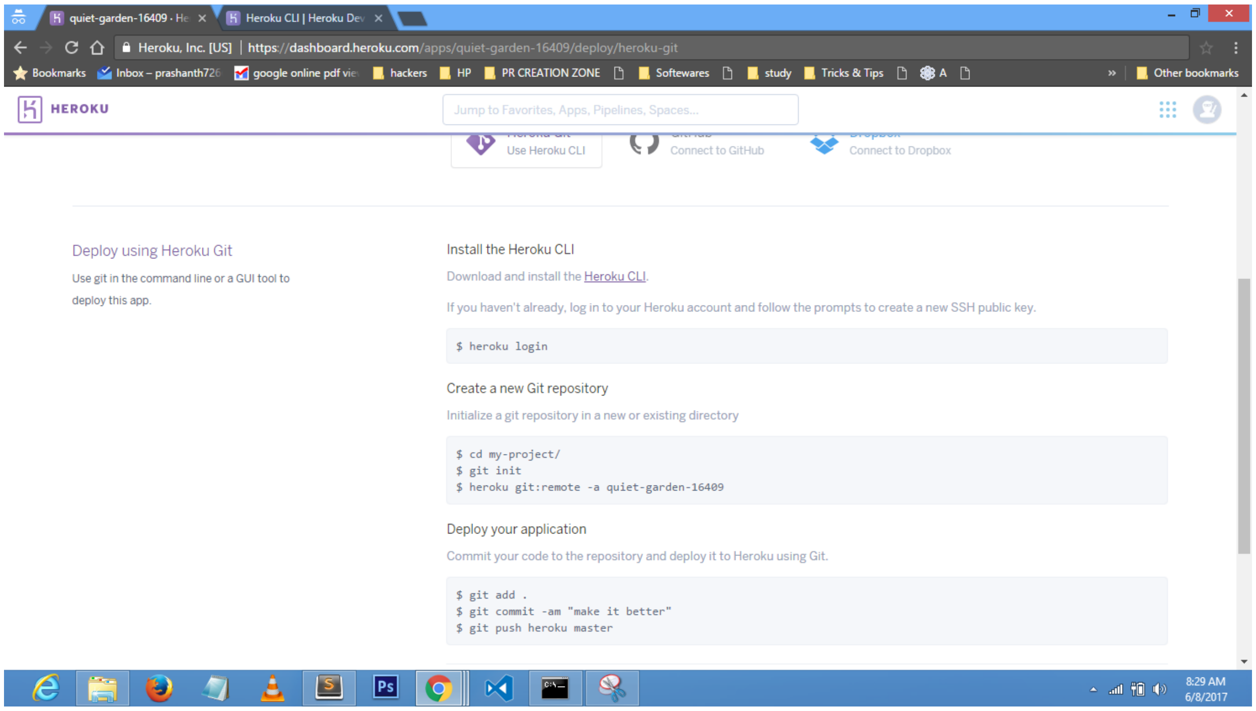
koteruprashanth@gmail.com: Enable Logplex

3 minutes ago • v2

koteruprashanth@gmail.com: Initial release

3 minutes ago • v1

Then you will see this:


Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the Heroku CLI.

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

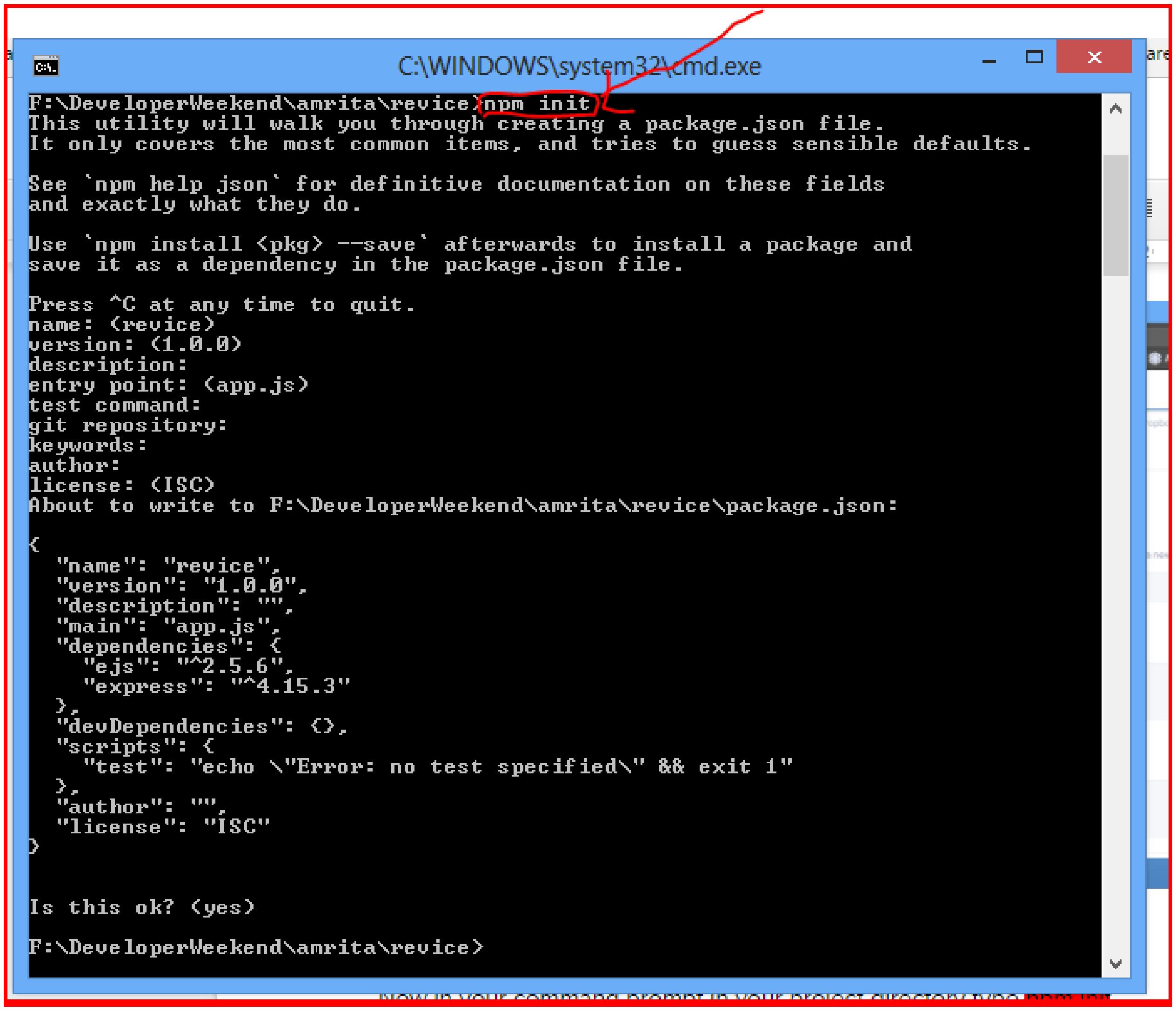
```
$ cd my-project/
$ git init
$ heroku git:remote -a quiet-garden-16409
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

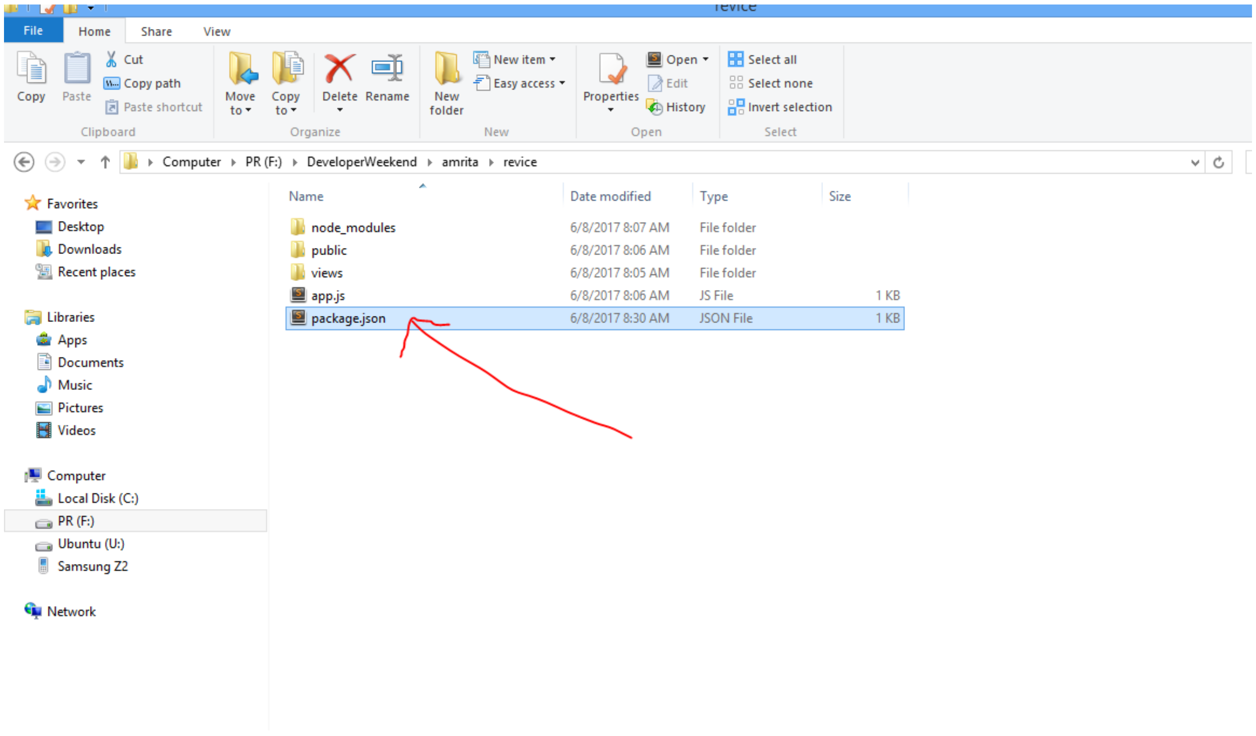
Now in your command prompt in your project directory type npm init



```
F:\DeveloperWeekend\amrita\revice>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 json` for definitive documentation on these fields
and exactly what they do.

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

Press ^C at any time to quit.
name: (revice)
version: (1.0.0)
description:
entry point: (app.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to F:\DeveloperWeekend\amrita\revice\package.json:

{
  "name": "revice",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "ejs": "^2.5.6",
    "express": "^4.15.3"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this ok? (yes)

F:\DeveloperWeekend\amrita\revice>
```

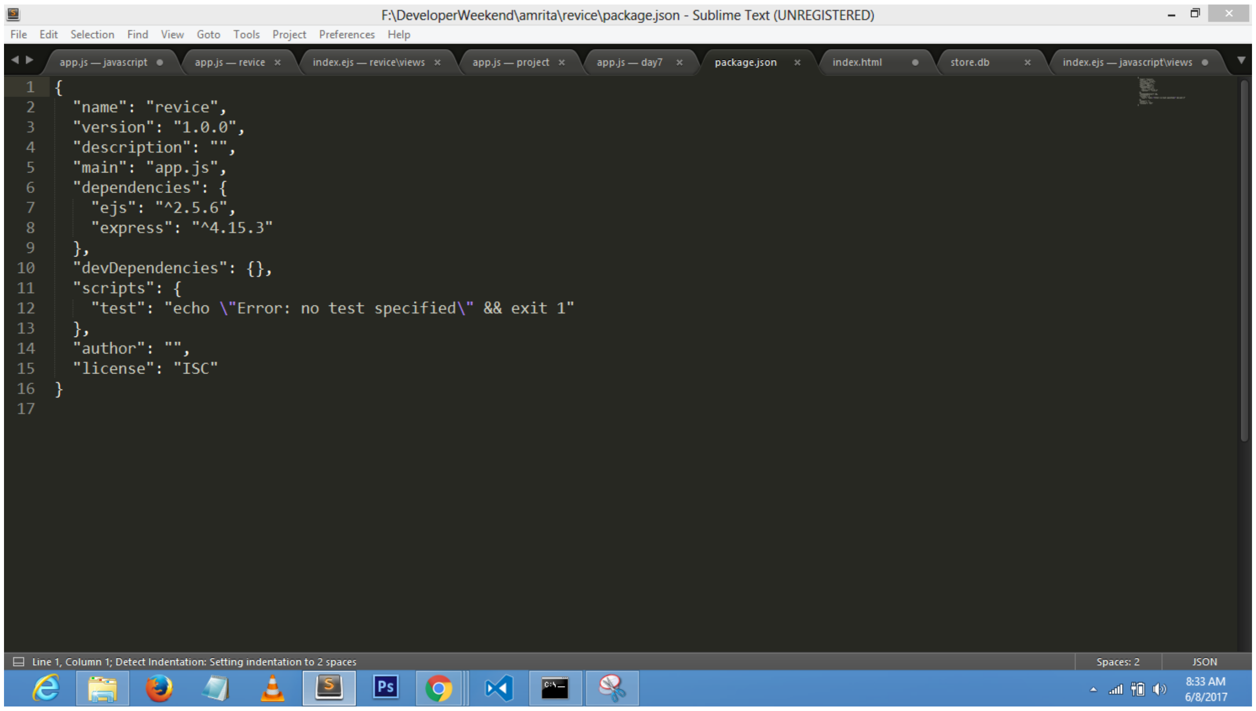
Just press enter till you reach this.

Then you can see a file called package.json in your project directory, open that



Name	Date modified	Type	Size
node_modules	6/8/2017 8:07 AM	File folder	
public	6/8/2017 8:06 AM	File folder	
views	6/8/2017 8:05 AM	File folder	
app.js	6/8/2017 8:06 AM	JS File	1 KB
package.json	6/8/2017 8:30 AM	JSON File	1 KB

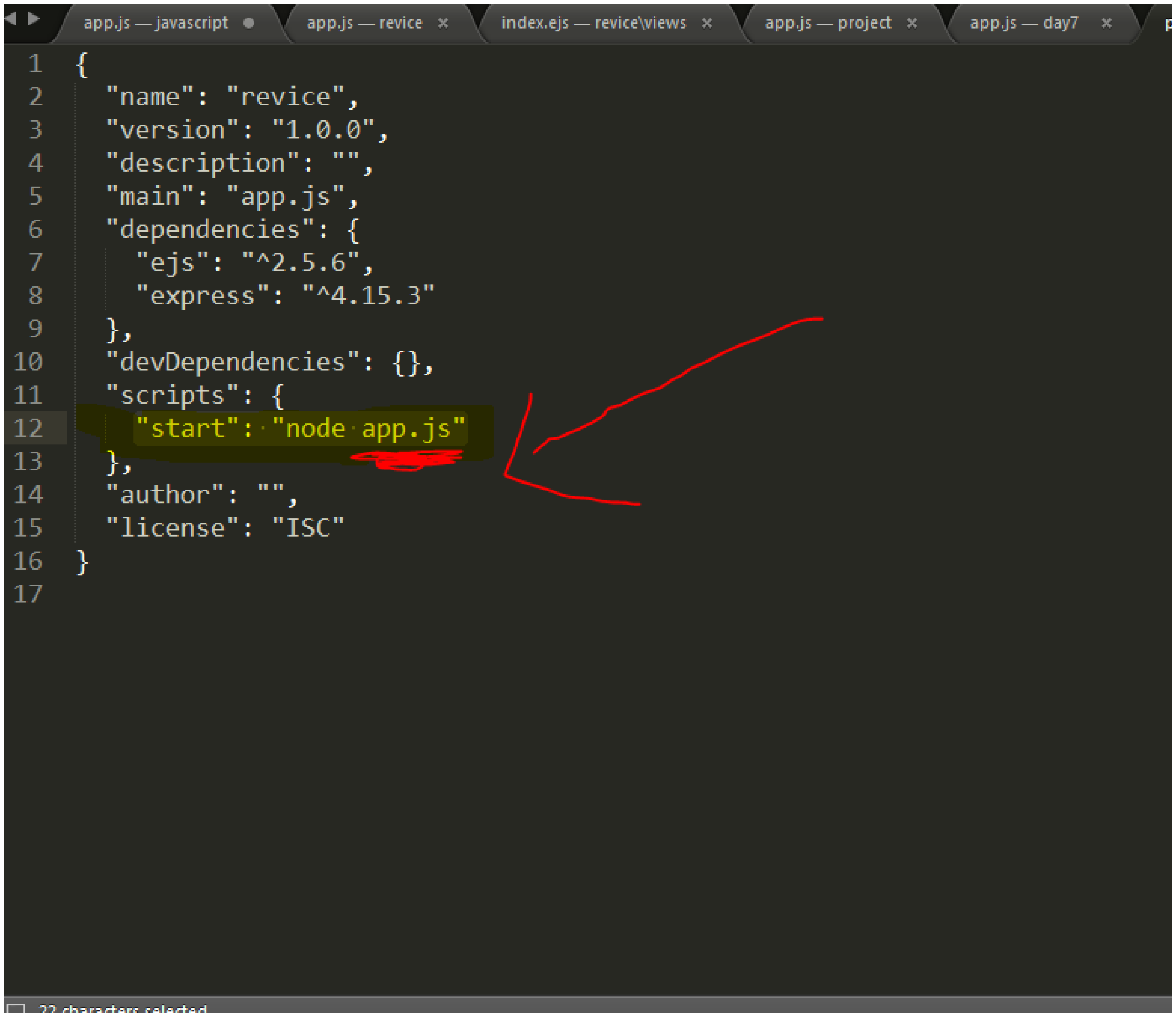
Then you will see this



F:\DeveloperWeekend\amrita\revice\package.json - Sublime Text (UNREGISTERED)
package.json

```json
{
  "name": "revice",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "ejs": "^2.5.6",
    "express": "^4.15.3"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

Now do the following modification: add node <your main executable filename> (eg:node app.js)

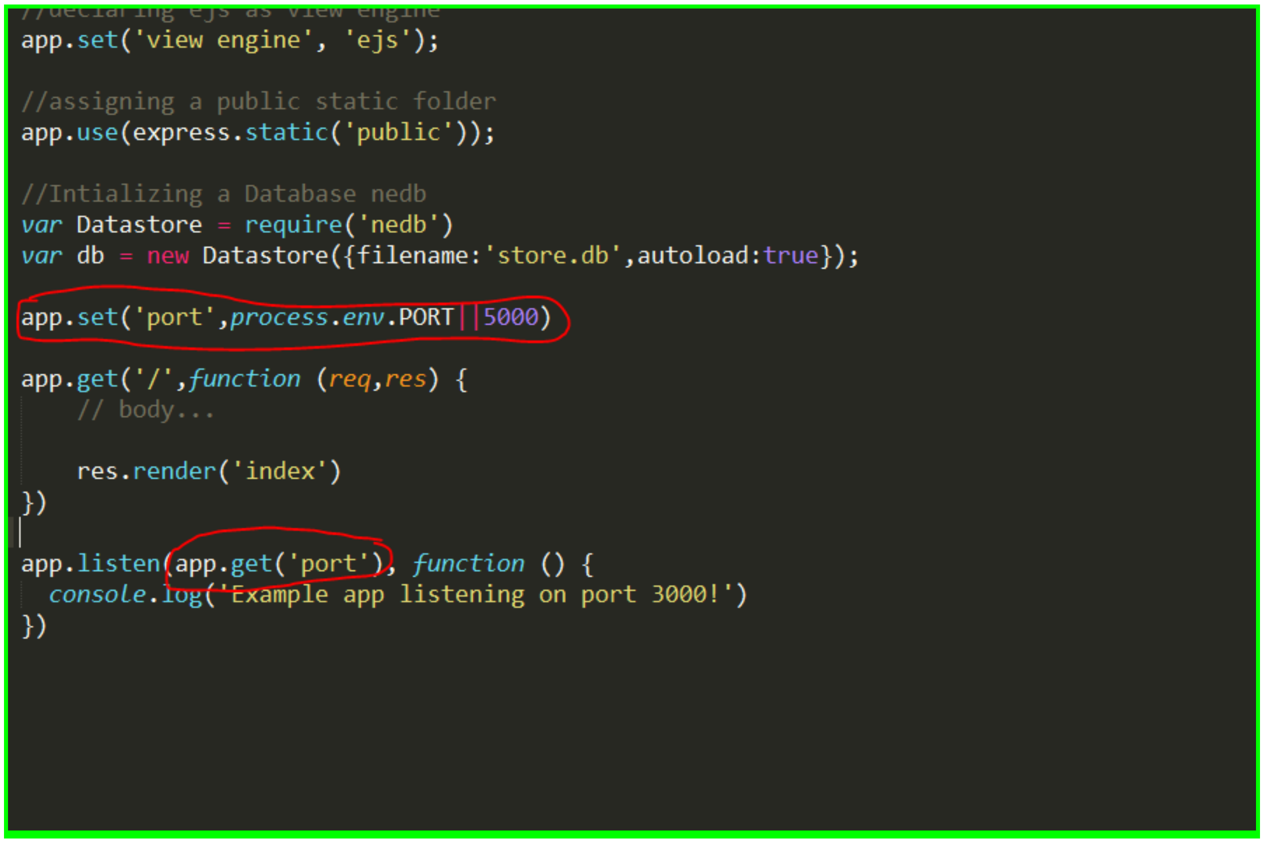


```
{
  "name": "revice",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "ejs": "^2.5.6",
    "express": "^4.15.3"
  },
  "devDependencies": {},
  "scripts": {
    "start": "node app.js"
  },
  "author": "",
  "license": "ISC"
}
```

Now in your app.js file do the following modifications:

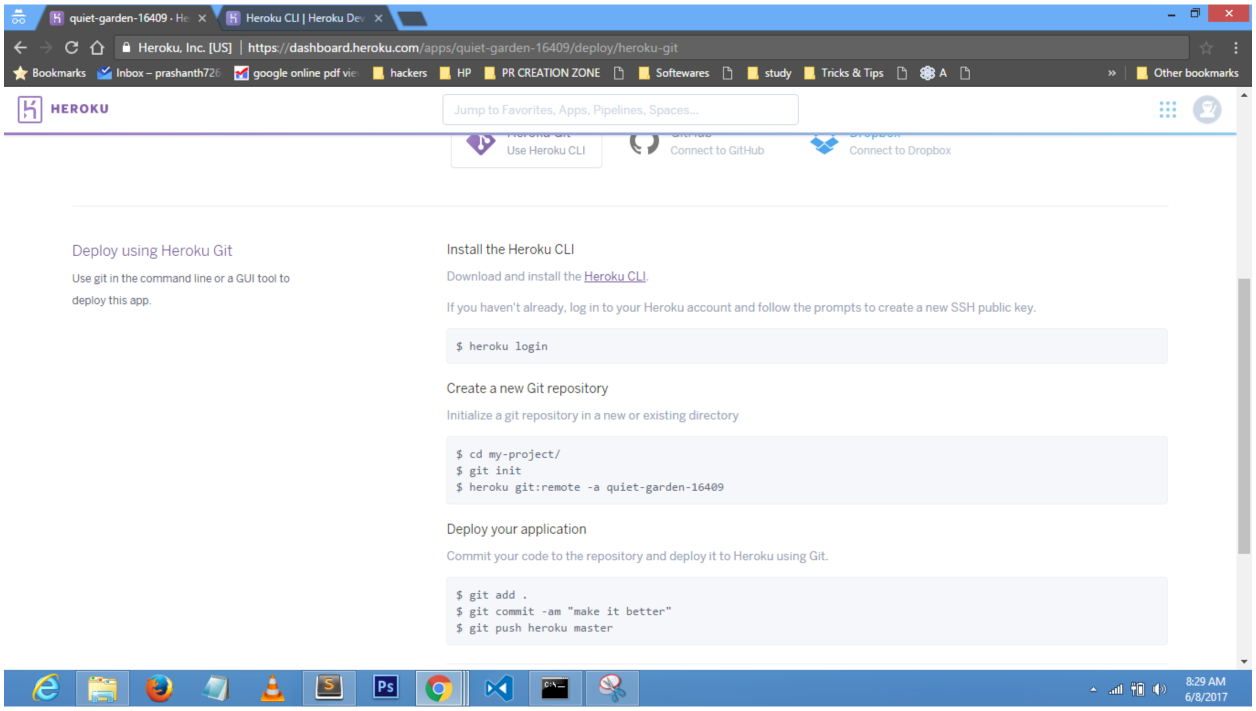
Add this line

app.set('port',process.env.PORT||5000)



```
//declaring ejs as view engine
app.set('view engine', 'ejs');

//assigning a public static folder
app.use(express.static('public'));

//Intializing a Database nedb
var Datastore = require('nedb')
var db = new Datastore({filename:'store.db',autoload:true});

app.set('port',process.env.PORT||5000)

app.get('/',function (req,res) {
    // body...

    res.render('index')
})

app.listen(app.get('port'), function () {
  console.log('Example app listening on port 3000!')
})
```

Now in go back to your app heroku dashboard


Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the Heroku CLI.

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/
$ git init
$ heroku git:remote -a quiet-garden-16409
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

This page

And now type these 5 commands in the command prompt

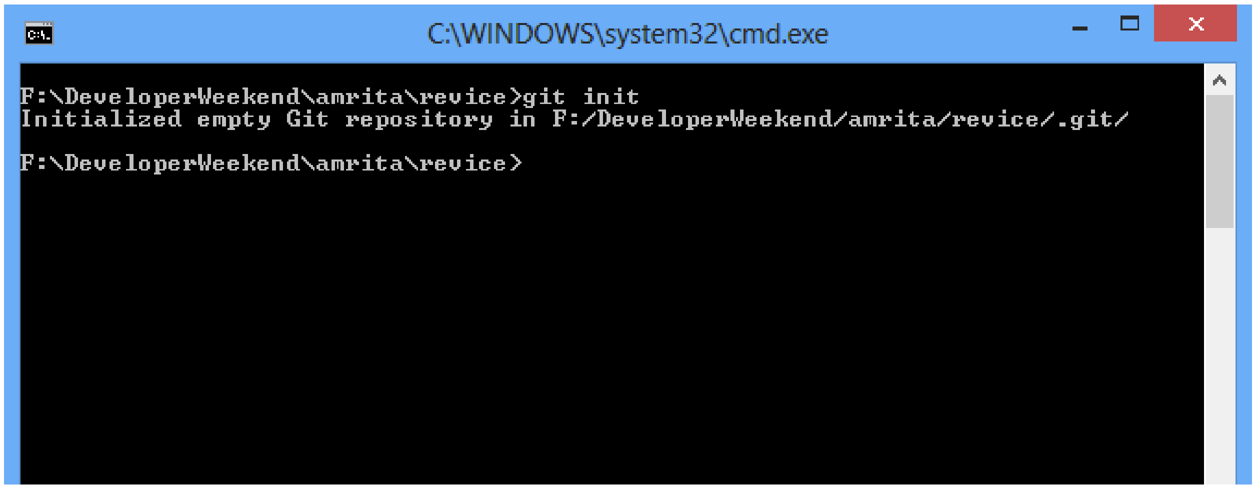
```
git init
heroku git:remote -a quiet-garden-16409
```

(this line will change for you it is heroku git:remote -a <your app name>)

```
git add .
git commit -am "make it better"
git push heroku master
```

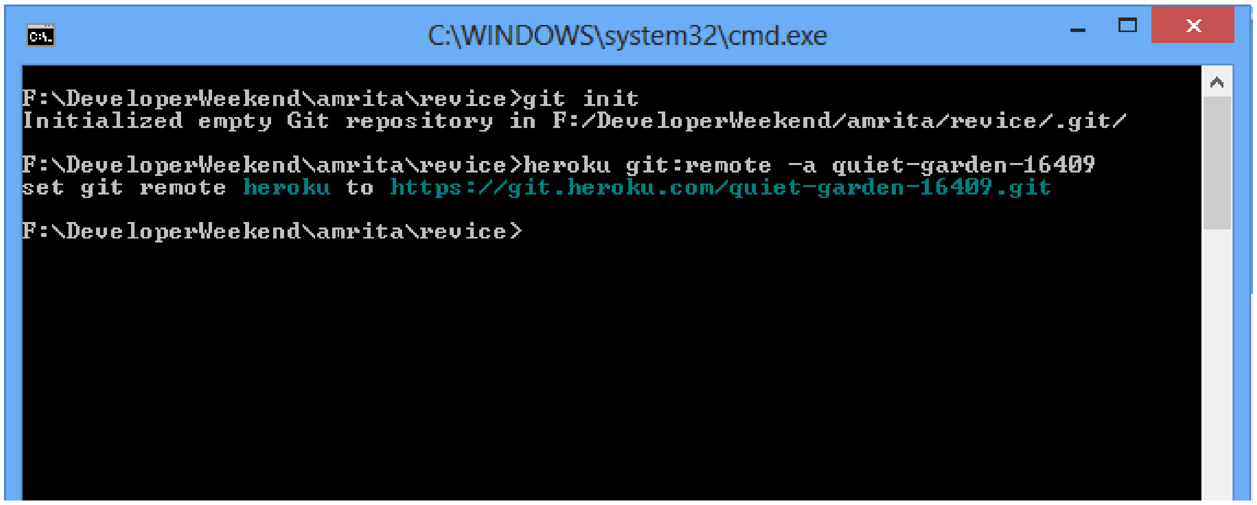
`Observe the below screenshots`

First Command git init



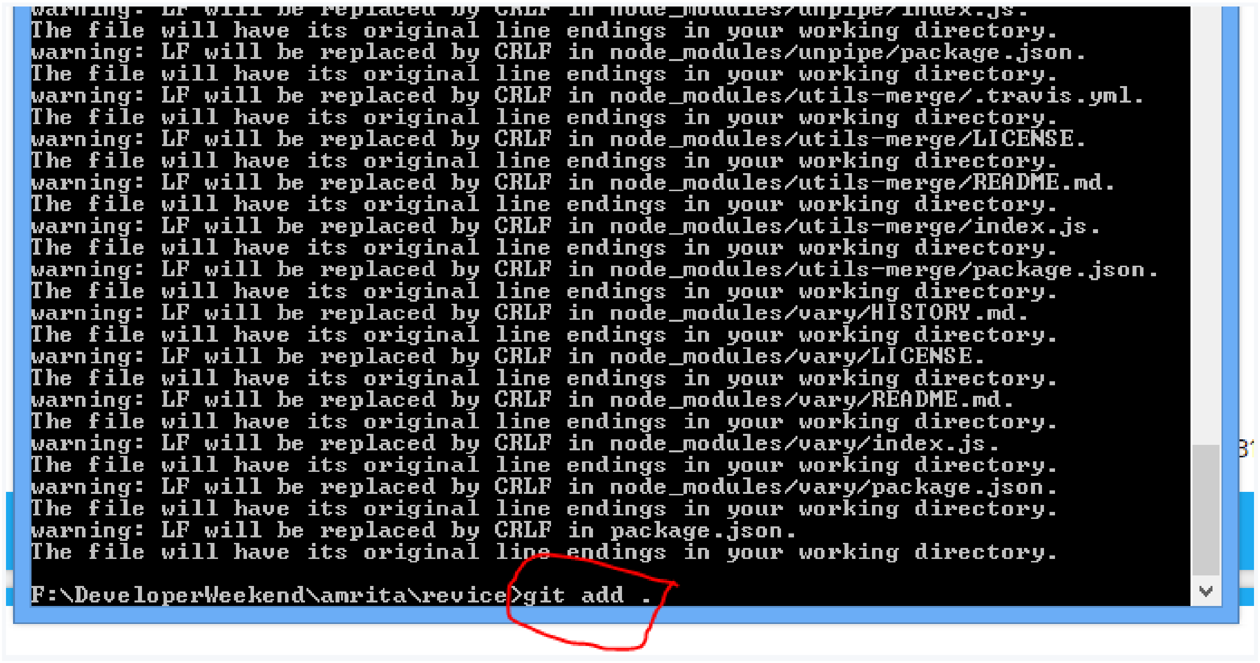
```
F:\DeveloperWeekend\amrita\revice>git init
Initialized empty Git repository in F:/DeveloperWeekend/amrita/revice/.git/

F:\DeveloperWeekend\amrita\revice>
```

Second `heroku git:remote -a quiet-garden-16409` *(this line will change for you it is heroku git:remote -a <your app name>, check the app dashboard for your app name)*



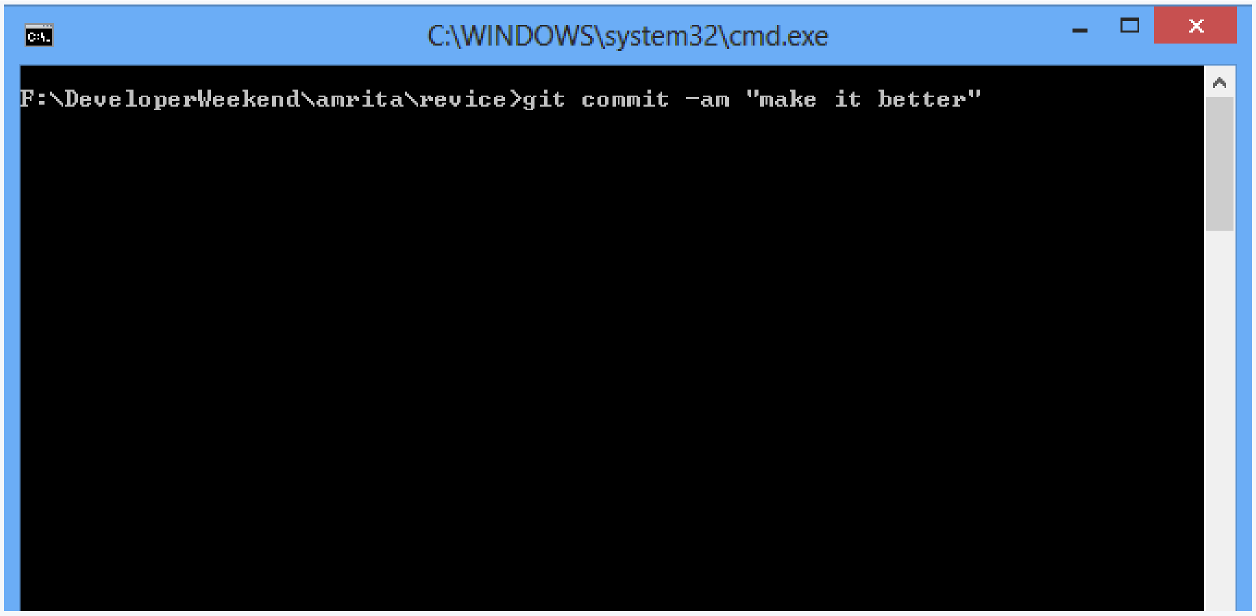
```
F:\DeveloperWeekend\amrita\revice>git init
Initialized empty Git repository in F:/DeveloperWeekend/amrita/revice/.git/

F:\DeveloperWeekend\amrita\revice>heroku git:remote -a quiet-garden-16409
set git remote heroku to https://git.heroku.com/quiet-garden-16409.git

F:\DeveloperWeekend\amrita\revice>
```

Third Command : `git add .`



```
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/unpipe/package.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/utils-merge/.travis.yml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/utils-merge/LICENSE.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/utils-merge/README.md.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/utils-merge/index.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/utils-merge/package.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/vary/HISTORY.md.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/vary/LICENSE.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/vary/README.md.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/vary/index.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in node_modules/vary/package.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory.

F:\DeveloperWeekend\amrita\revice>git add .
```

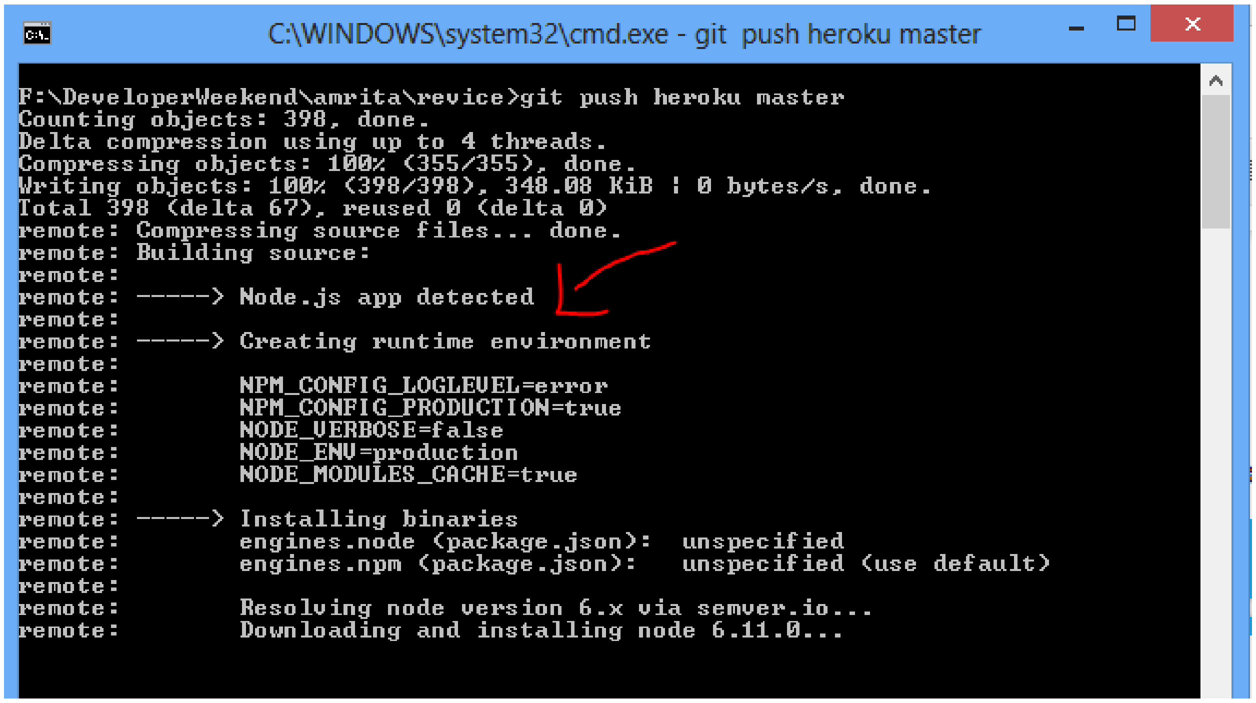
`Fourth command git commit -am "make it better"`



C:\WINDOWS\system32\cmd.exe

```
F:\DeveloperWeekend\amrita\revice>git commit -am "make it better"
```

`Fifth and final one git push heroku master`

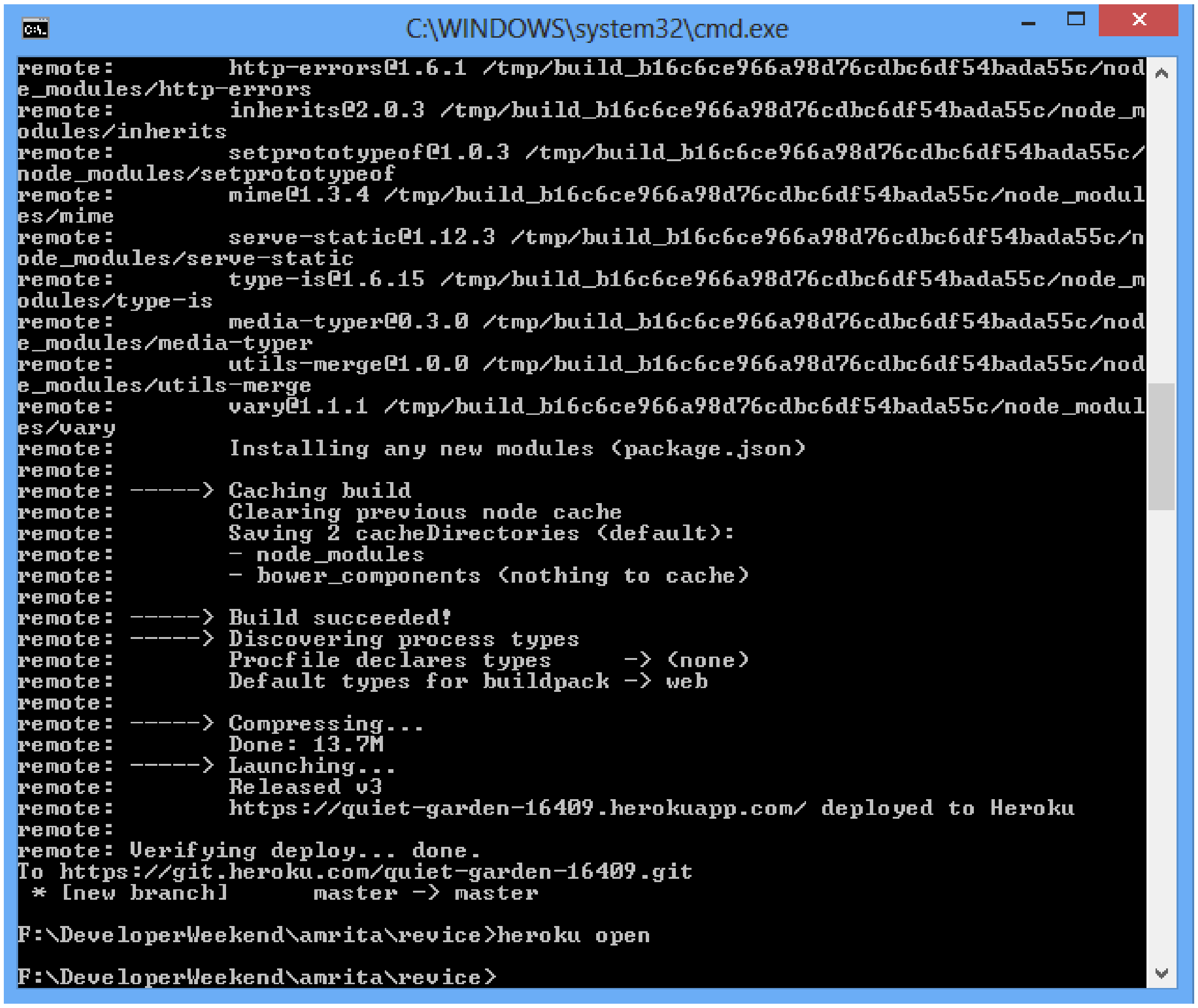


```
F:\DeveloperWeekend\amrita\revice>git push heroku master
Counting objects: 398, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (355/355), done.
Writing objects: 100% (398/398), 348.08 KiB | 0 bytes/s, done.
Total 398 (delta 67), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:        NPM_CONFIG_LOGLEVEL=error
remote:        NPM_CONFIG_PRODUCTION=true
remote:        NODE_VERBOSE=false
remote:        NODE_ENV=production
remote:        NODE_MODULES_CACHE=true
remote:
remote: -----> Installing binaries
remote:        engines.node (package.json):  unspecified
remote:        engines.npm (package.json):   unspecified (use default)
remote:
remote:        Resolving node version 6.x via semver.io...
remote:        Downloading and installing node 6.11.0...
```

Then type

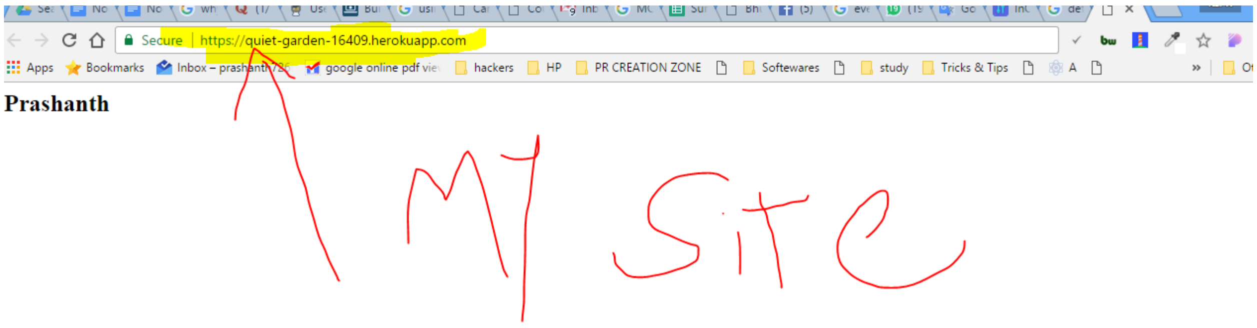
```
Heroku open
```

Boom Boom.................



C:\WINDOWS\system32\cmd.exe

```
remote:        http-errors@1.6.1 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/nod
e_modules/http-errors
remote:        inherits@2.0.3 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/node_m
odules/inherits
remote:        setprototypeof@1.0.3 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/
node_modules/setprototypeof
remote:        mime@1.3.4 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/node_modul
es/mime
remote:        serve-static@1.12.3 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/n
ode_modules/serve-static
remote:        type-is@1.6.15 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/node_m
odules/type-is
remote:        media-typer@0.3.0 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/nod
e_modules/media-typer
remote:        utils-merge@1.0.0 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/nod
e_modules/utils-merge
remote:        vary@1.1.1 /tmp/build_b16c6ce966a98d76cdbc6df54bada55c/node_modul
es/vary
remote:        Installing any new modules (package.json)
remote:
remote: -----> Caching build
remote:        Clearing previous node cache
remote:        Saving 2 cacheDirectories (default):
remote:        - node_modules
remote:        - bower_components (nothing to cache)
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:        Procfile declares types     -> (none)
remote:        Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:        Done: 13.7M
remote: -----> Launching...
remote:        Released v3
remote:        https://quiet-garden-16409.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/quiet-garden-16409.git
 * [new branch]      master -> master

F:\DeveloperWeekend\amrita\revice>heroku open

F:\DeveloperWeekend\amrita\revice>
```



https://quiet-garden-16409.herokuapp.com

Prashanth

Got Doubts???

Search in Google

Got one more doubt???

Search in Google

One more doubt????

Search in Google

Still doubts ....

....

....

....

...

Then Search in Google

Or

whatsapp me..

+919738074729

Thank you

Prashanth

StudyOwl