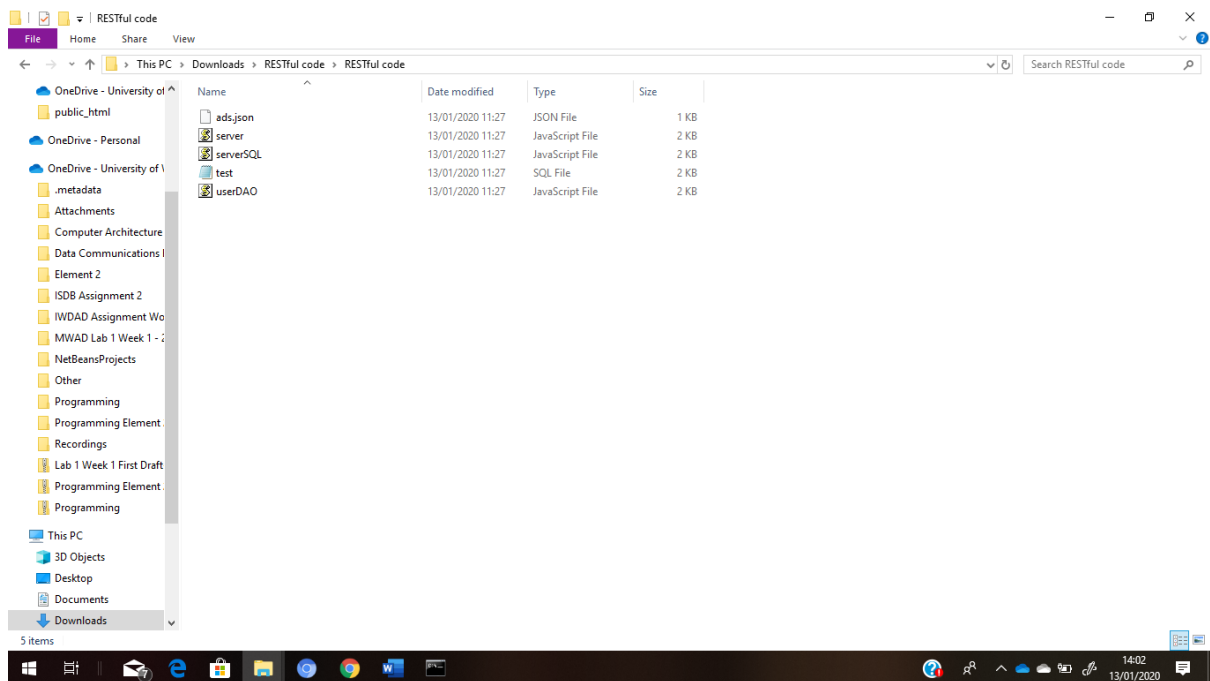
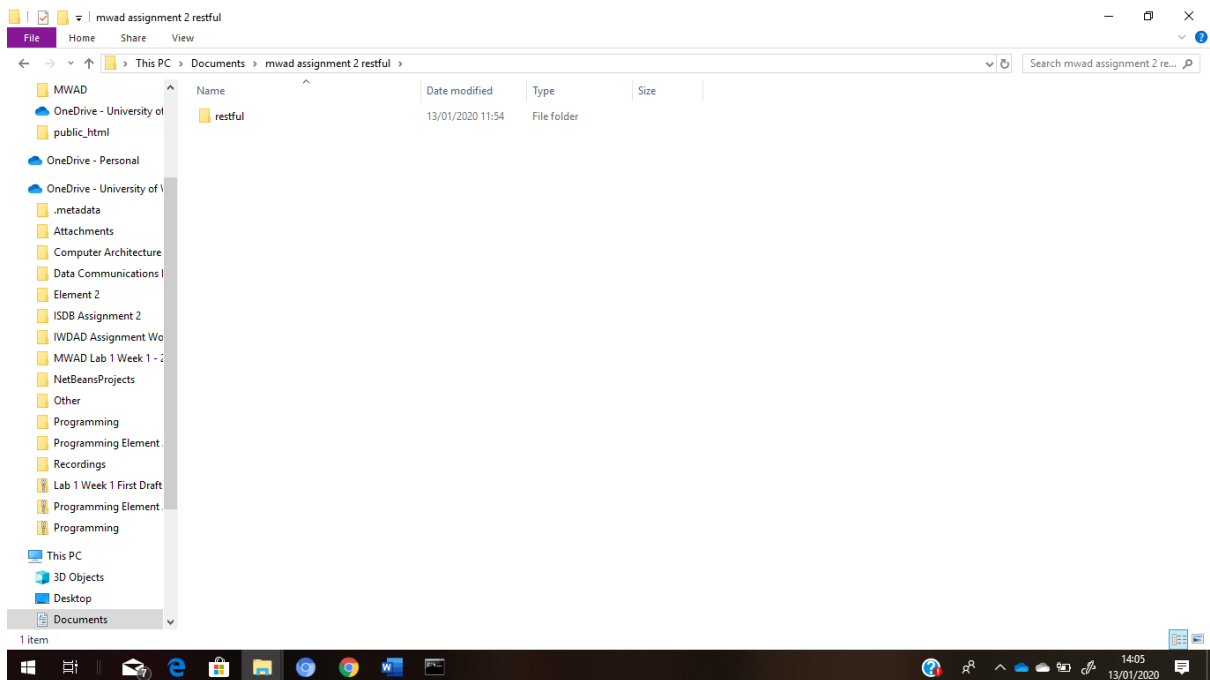
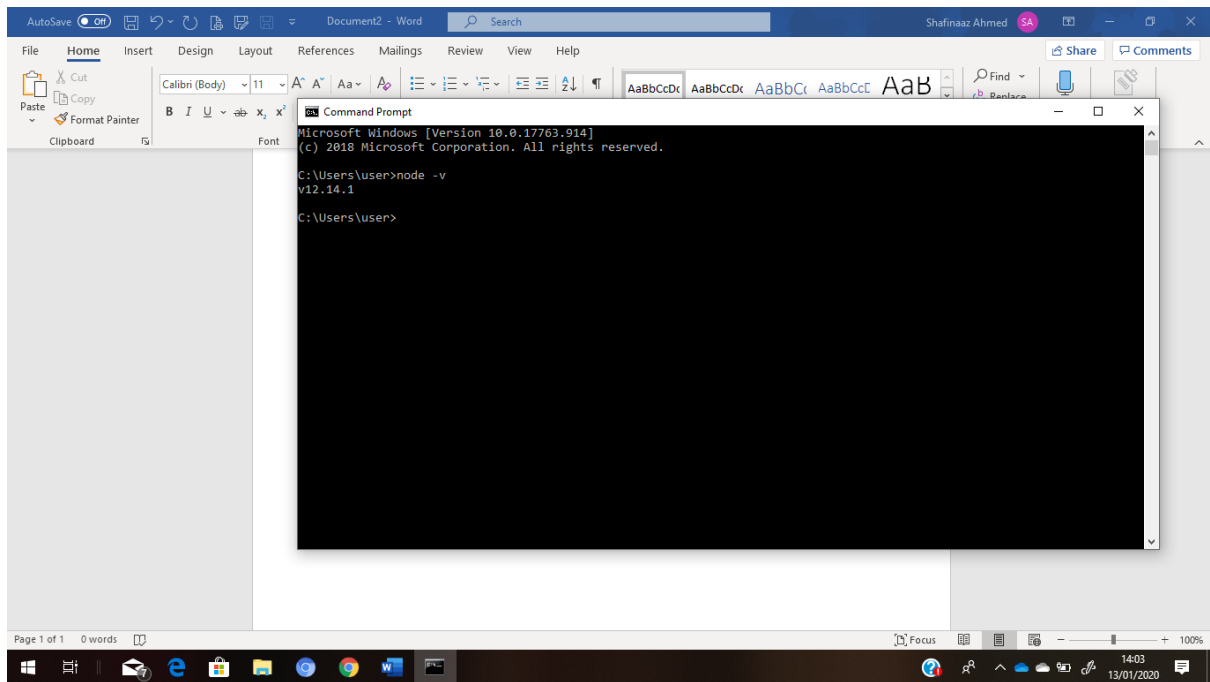


## RESTFUL code downloaded from Blackboard





```

Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

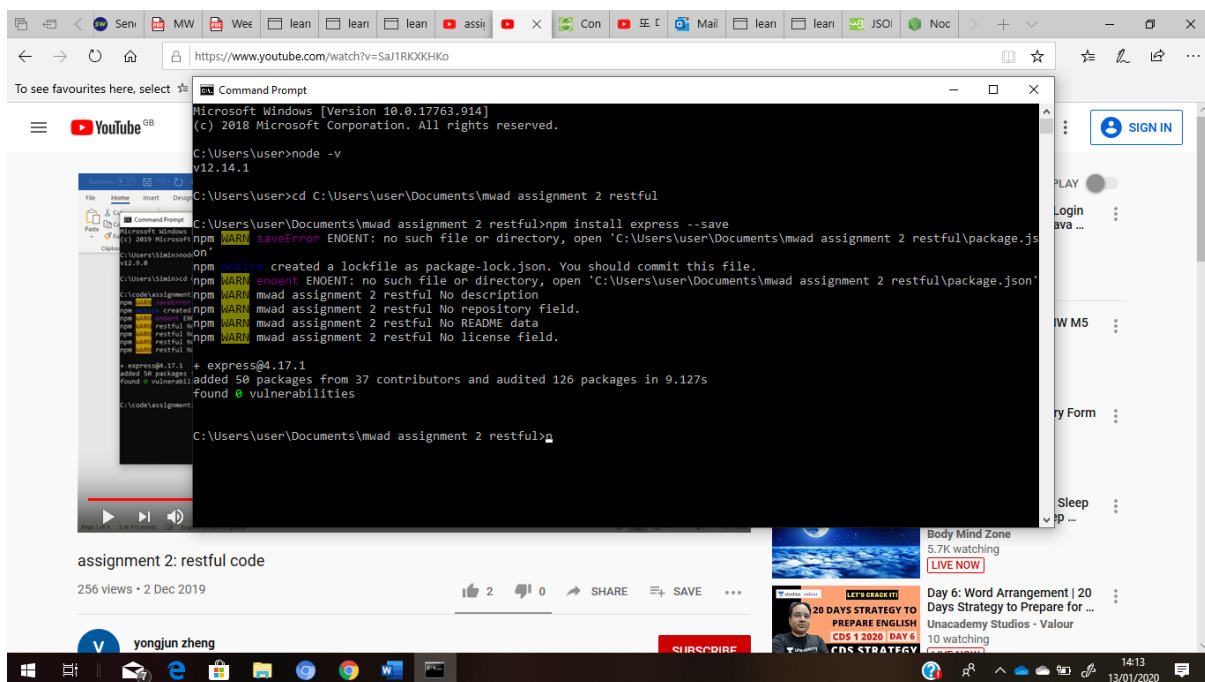
C:\Users\user>node -v
v12.14.1

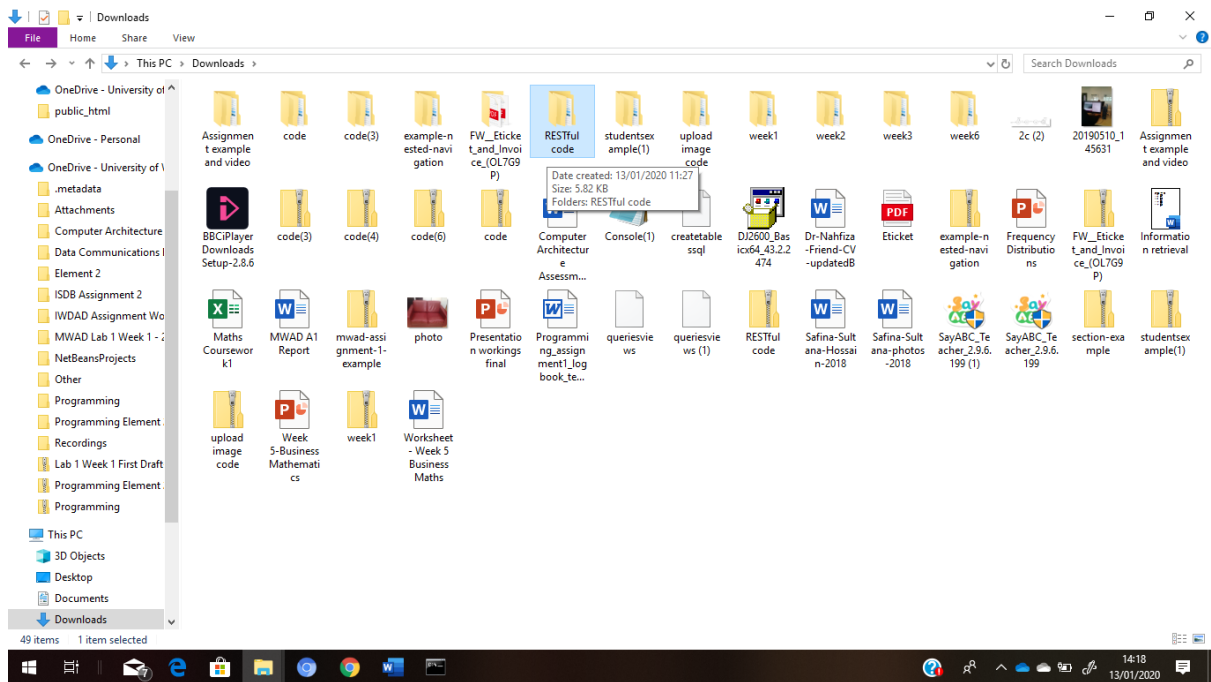
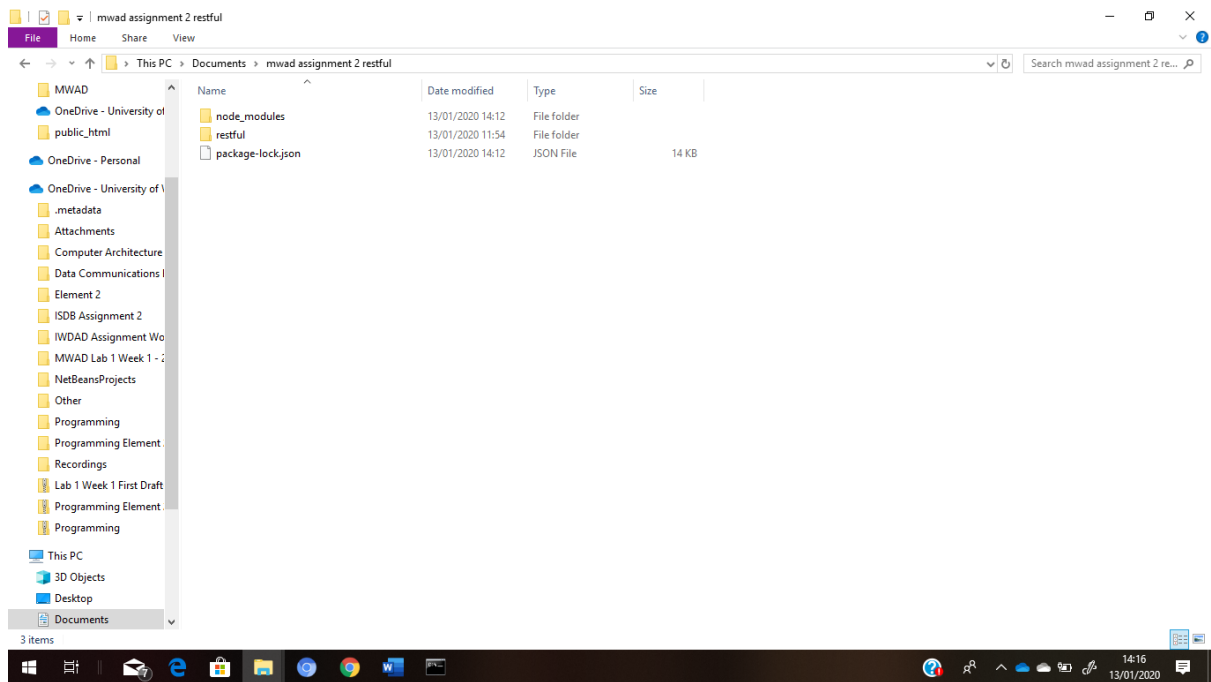
C:\Users\user>cd C:\Users\user\Documents\mwad assignment 2 restful

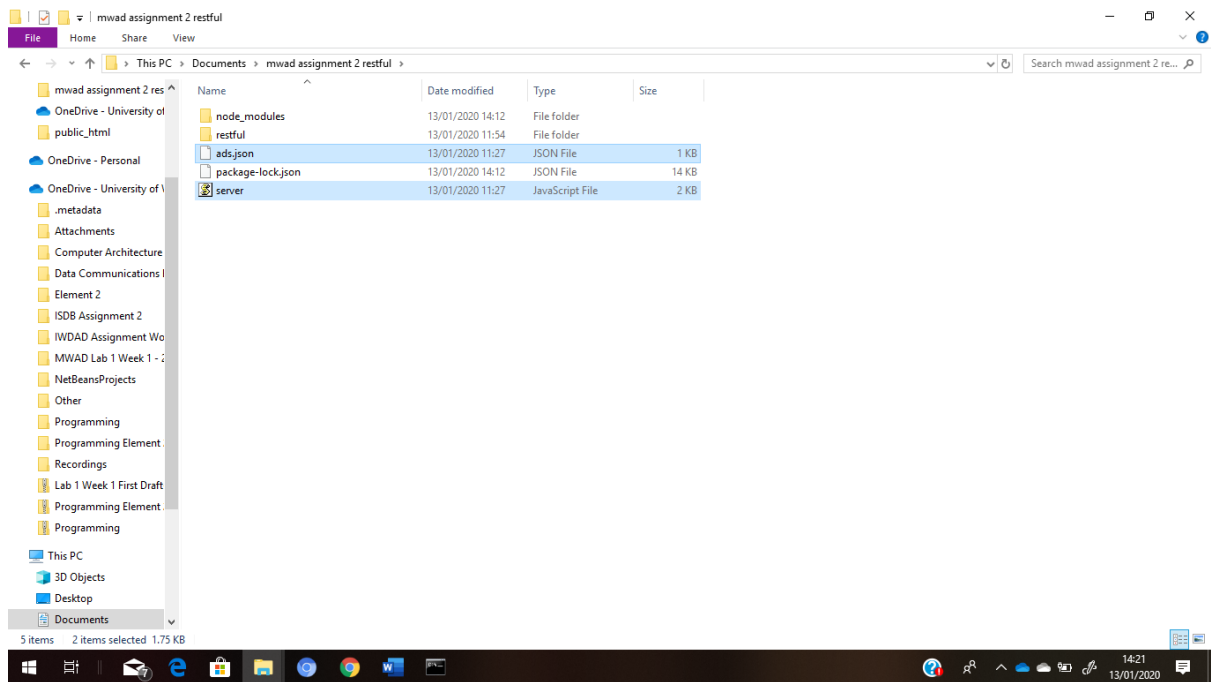
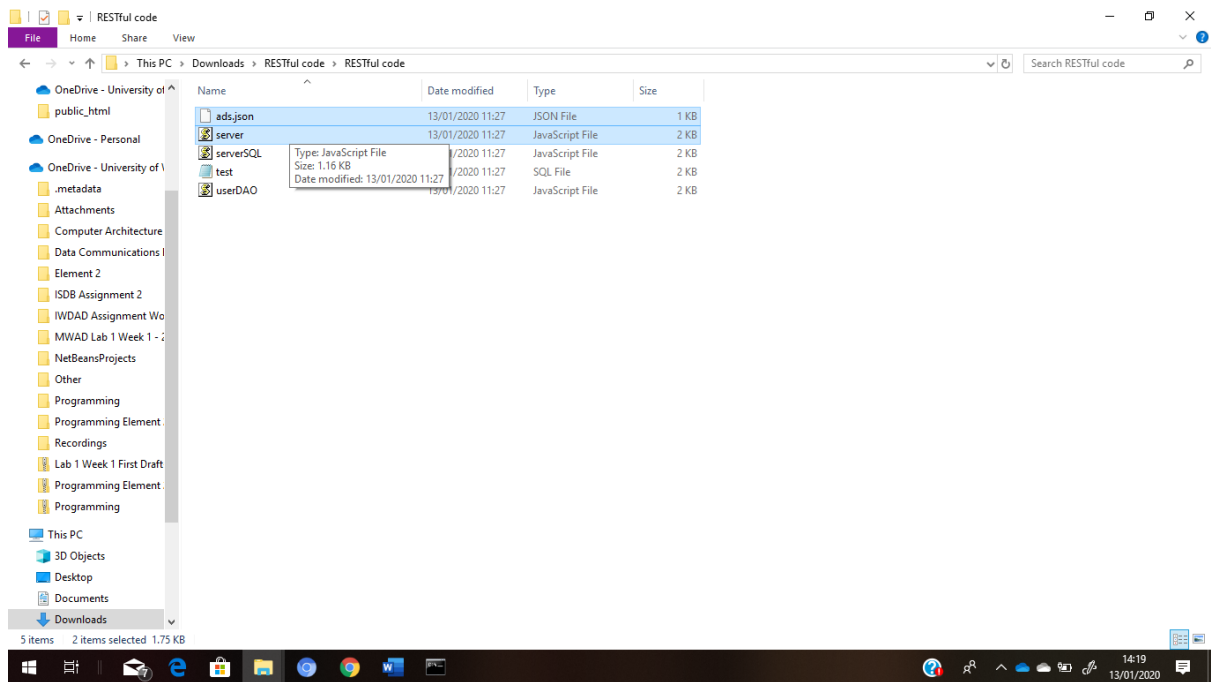
C:\Users\user\Documents\mwad assignment 2 restful>_

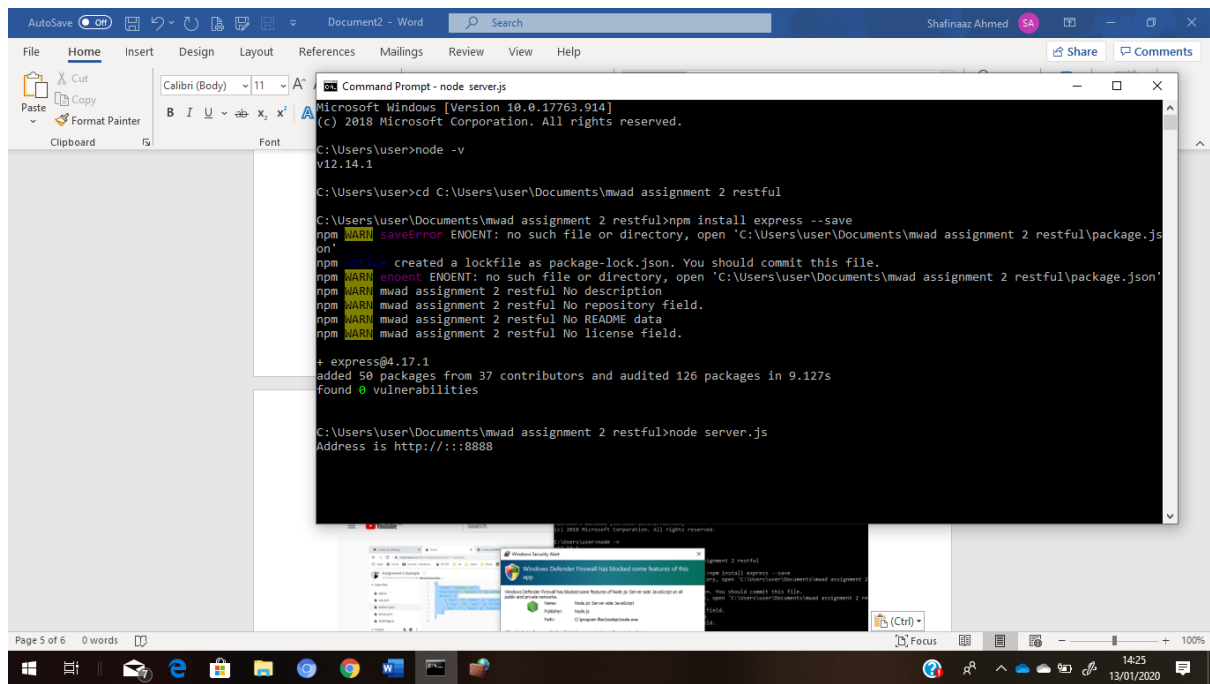
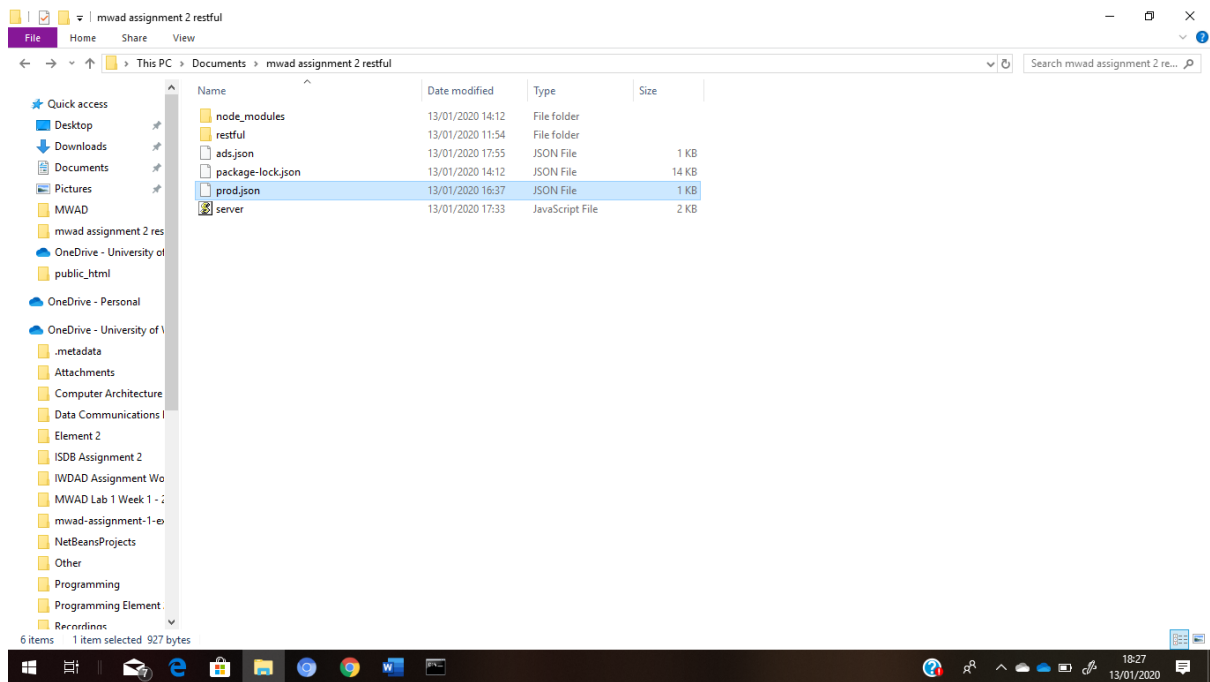
```

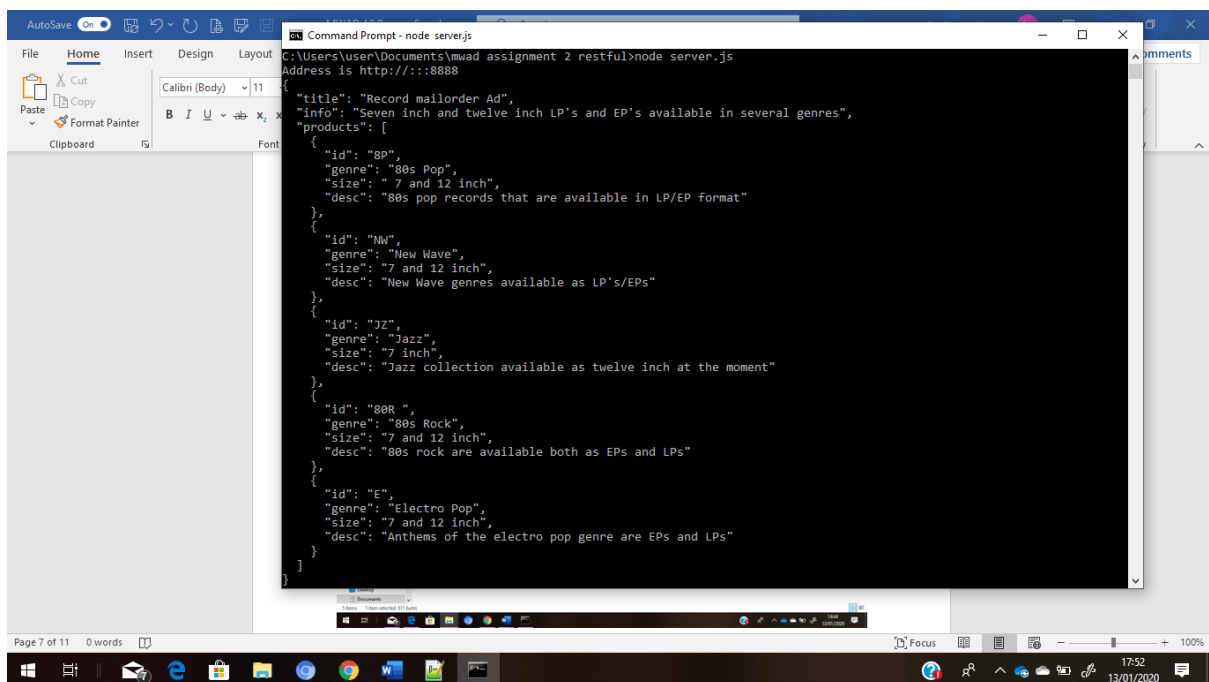
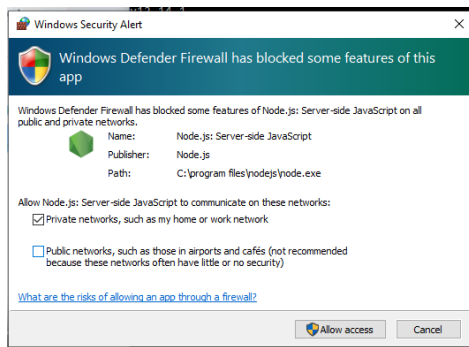
Having installed node.js on our machine I check it on command prompt using `node -v`. The version of the one installed is shown on the prompt `v12.14.1`



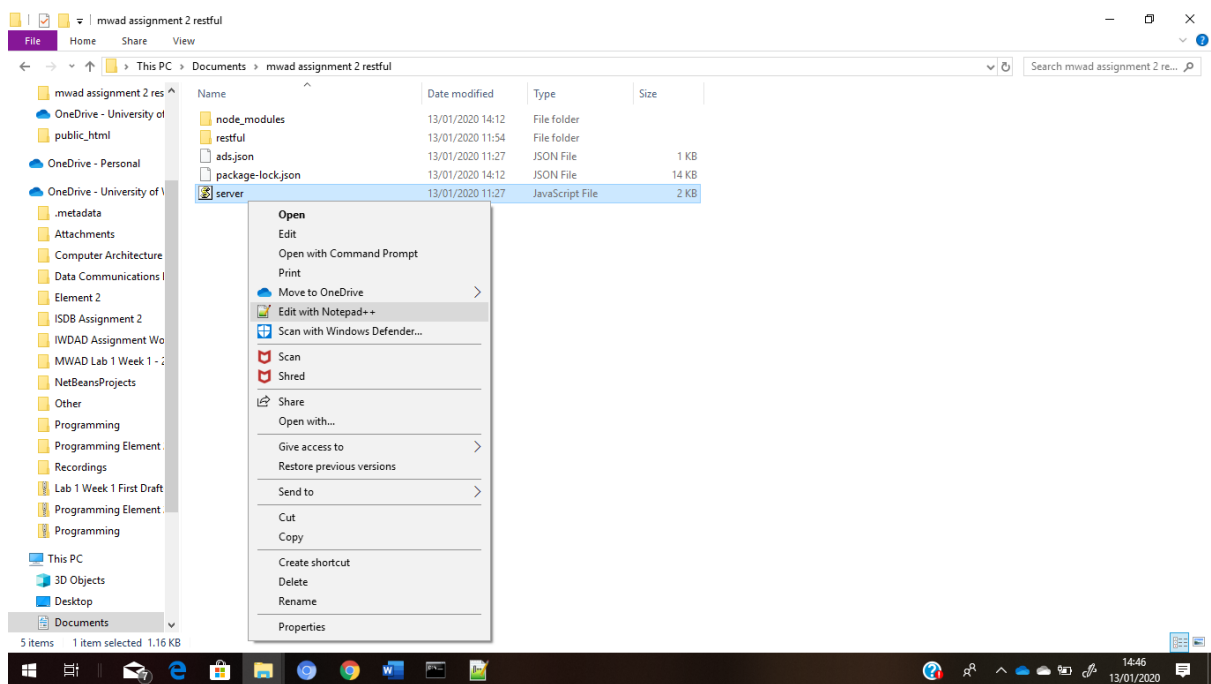
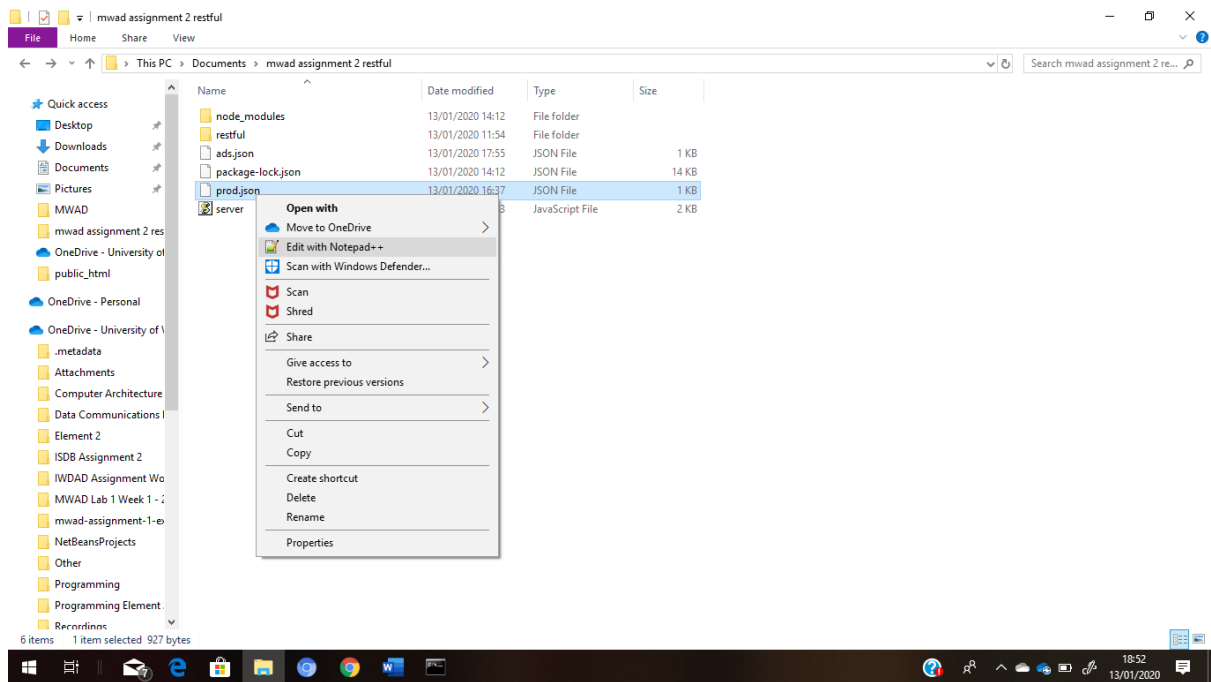








With the server running as indicated by the port number 8888, the json file code is loaded on the command prompt as it is loaded on the web browser by localhost with this port number.





```
C:\Users\user\Documents\mwad assignment 2 restful\server.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

1  /*The example show below: the response is a string, if you want to send the response prettified,
2  you could use something like JSON.stringify(anObject, null, 3)
3  It's important that you set the Content-Type header to application/json, too.
4  */
5
6  var express = require('express');
7  var app = express();
8  var fs = require("fs");
9
10 app.get('/listProd', function (req, res) {
11   fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
12     console.log( data );
13     res.end( data );
14   });
15 }
16 //-----list the user by id
17
18
19 app.get('/listProd/:id', function (req, res) {
20   // retrieve all the users
```

Looking at the first three lines declaring variables: First line lets us know the express library will be brought i.e. imported. An instance of the express library is then created by declaring the variable app. Line 8 shows the file system being required for the variable fs.

```
C:\Users\user\Documents\mwad assignment 2 restful\server.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

1  /*The example show below: the response is a string, if you want to send the response prettified,
2  you could use something like JSON.stringify(anObject, null, 3)
3  It's important that you set the Content-Type header to application/json, too.
4  */
5
6  var express = require('express');
7  var app = express();
8  var fs = require("fs");
9
10 app.get('/listProd', function (req, res) {
11   fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
12     console.log( data );
13     res.end( data );
14   });
15 }
16 //-----list the user by id
17
18
19 app.get('/listProd/:id', function (req, res) {
20   // retrieve all the users
21   fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
22     data = JSON.parse( data );
23     // var user = data.prod[req.params.id-1].title //get user by key, which is Ad 1, ...
24     var user = data.prod[req.params.id-1]//get user by key, which is Ad 1, ...
25     console.log( user );
26     res.end( JSON.stringify(user));
27   });
28 }
29 }
```

The code with this variable shows the local JSON file being read from. So this is fs which will read the prod.json and then I will pass this file to the client.

```
C:\Users\user\Documents\mwad assignment 2 restful\server.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

1 /*The example show below: the response is a string, if you want to send the response prettified,
2 you could use something like JSON.stringify(anObject, null, 3)
3 It's important that you set the Content-Type header to application/json, too.
4 */
5
6 var express = require('express');
7 var app = express();
8 var fs = require("fs");
9
10 app.get('/listProd', function (req, res) {
11     fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
12         console.log( data );
13         res.end( data );
14     });
15 })
16 //-----list the user by id
17
18
19 app.get('/listProd/:id', function (req, res) {
20     // retrieve all the users
21     fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
22         data = JSON.parse( data );
23         // var user = data.prod[req.params.id-1].title //get user by key, which is Ad 1, ...
24         var user = data.prod[req.params.id-1]//get user by key, which is Ad 1, ...
25         console.log( user );
26         res.end( JSON.stringify(user));
27     });
28 })
29
```

In the line that makes the file system read the json file the file prod.json is written after the + so that it is accessed and read to be run on the browser by localhost as the RESTful API.

```
C:\Users\user\Documents\mwad assignment 2 restful\server.js - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

1 /*The example show below: the response is a string, if you want to send the response prettified,
2 you could use something like JSON.stringify(anObject, null, 3)
3 It's important that you set the Content-Type header to application/json, too.
4 */
5
6 var express = require('express');
7 var app = express();
8 var fs = require("fs");
9
10 app.get('/listProd', function (req, res) {
11     fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
12         console.log( data );
13         res.end( data );
14     });
15 })
16 //-----list the user by id
17
18
19 app.get('/listProd/:id', function (req, res) {
20     // retrieve all the users
21     fs.readFile( __dirname + "/" + "prod.json", 'utf8', function (err, data) {
22         data = JSON.parse( data );
23         // var user = data.prod[req.params.id-1].title //get user by key, which is Ad 1, ...
24         var user = data.prod[req.params.id-1]//get user by key, which is Ad 1, ...
25         console.log( user );
26         res.end( JSON.stringify(user));
27     });
28 })
29
```

We need the listProd as a parameter in the get method to make the prod.json file be displayed as a web page by localhost. The function with parameters req and res complete this task.

C:\Users\user\Documents\mwad assignment 2 restful\prod.json - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

ads.json prices.json server.js userDAO.js serverSQL.js crew members.json prices.json prod.json server.js

```
1 {
2   "title": "Record mailorder Ad",
3   "info": "Seven inch and twelve inch LP's and EP's available in several genres",
4   "products": [
5     {
6       "id": "1",
7       "genre": "80s Pop",
8       "size": "7 and 12 inch",
9       "desc": "80s pop records that are available in LP/EP format"
10    },
11    {
12      "id": "2",
13      "genre": "New Wave",
14      "size": "7 and 12 inch",
15      "desc": "New Wave genres available as LP's/EPs"
16    },
17    {
18      "id": "3",
19      "genre": "Jazz",
20      "size": "7 inch",
21      "desc": "Jazz collection available as twelve inch at the moment"
22    },
23    {
24      "id": "4 ",
25      "genre": "80s Rock",
26      "size": "7 and 12 inch",
27      "desc": "80s rock are available both as EPs and LPs"
28    },
29    {
30      "id": "5",
31      "genre": "Electro Pop",
32      "size": "7 and 12 inch",
33      "desc": "Anthems of the electro pop genre are EPs and LPs"
34    }
35  ]
36 }
```

Snacks by stu115 x | Snack x | Snack x | localhost:8888/listProd x

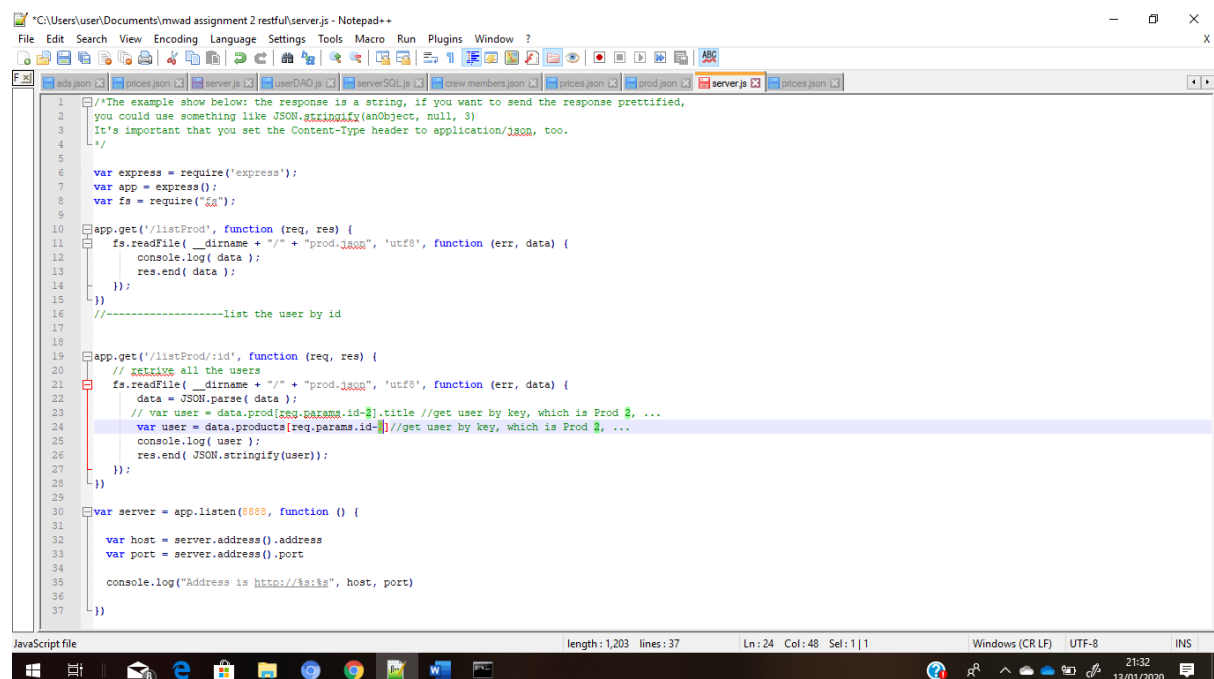
← → ↻ ⓘ localhost:8888/listProd

```
{
  "title": "Record mailorder Ad",
  "info": "Seven inch and twelve inch LP's and EP's available in several genres",
  "products": [
    {
      "id": "8p",
      "genre": "80s Pop",
      "size": "7 and 12 inch",
      "desc": "80s pop records that are available in LP/EP format"
    },
    {
      "id": "Nw",
      "genre": "New Wave",
      "size": "7 and 12 inch",
      "desc": "New Wave genres available as LP's/EPs"
    },
    {
      "id": "JZ",
      "genre": "Jazz",
      "size": "7 inch",
      "desc": "Jazz collection available as twelve inch at the moment"
    },
    {
      "id": "80R ",
      "genre": "80s Rock",
      "size": "7 and 12 inch",
      "desc": "80s rock are available both as EPs and LPs"
    },
    {
      "id": "E",
      "genre": "Electro Pop",
      "size": "7 and 12 inch",
      "desc": "Anthems of the electro pop genre are EPs and LPs"
    }
  ]
}
```

Here is the prod.json file open in notepad ++ and then on the client side on the browser by localhost with the port number 8888 and the listProd parameter. The indication of this is in the screenshot of the node.js being run on command prompt after the port number is shown.

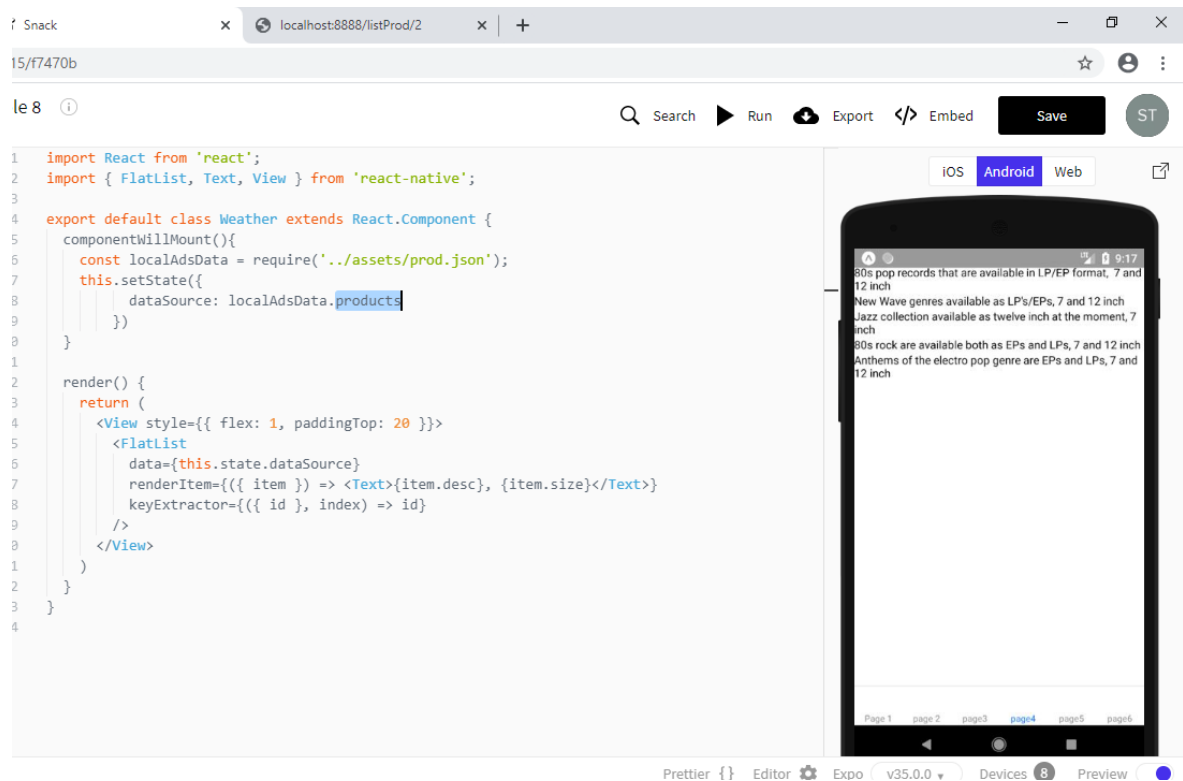
```
17
18
19 app.get('/listProd/:id', function (req, res) {
20     // retrieve all the users
21     fs.readFile(__dirname + "/" + "prod.json", 'utf8', function (err, data) {
22         data = JSON.parse( data );
23         // var user = data.prod[req.params.id-2].title //get user by key, which is Prod 2, ...
24         var user = data.products[req.params.id-2]//get user by key, which is Prod 2, ...
25         console.log( user );
26         res.end( JSON.stringify(user));
27     });
28 })
29
30 var server = app.listen(8888, function () {
31
32     var host = server.address().address
33     var port = server.address().port
34
35     console.log("Address is http://%s:%s", host, port)
36
37 })
```

Since there is the id as data for the products we may want to use that to list a particular detail about a product. This is by having it after /listProd/ to make details of a particular product be listed by its id number with localhost on the web browser.

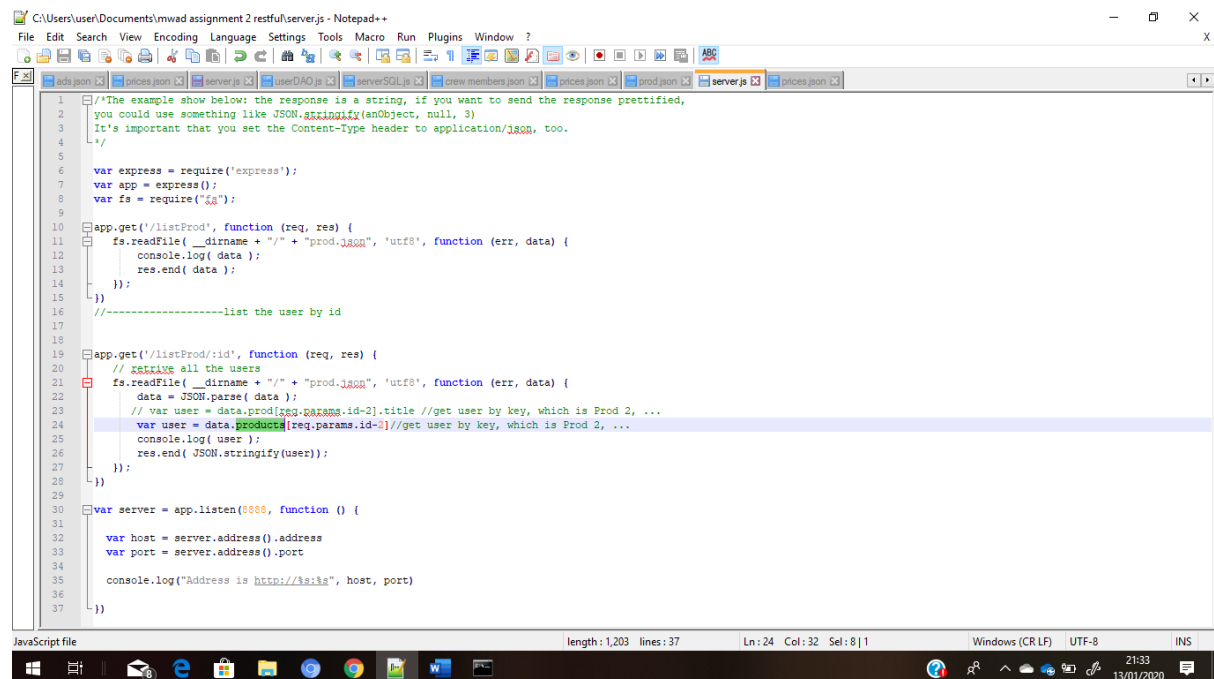


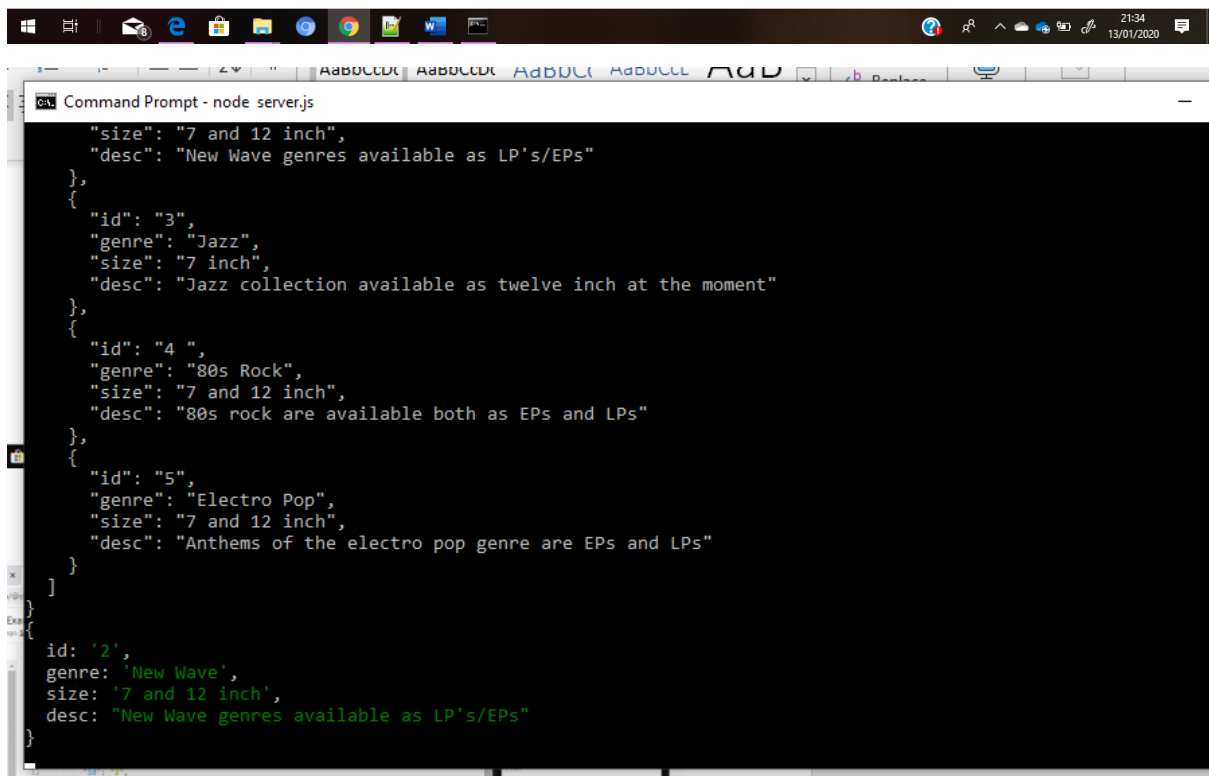
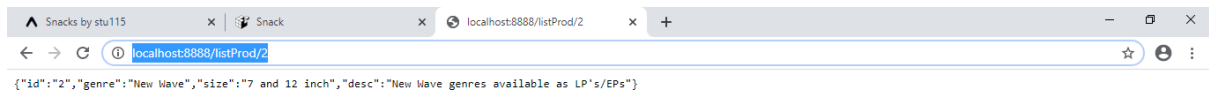
```
1  /*The example show below: the response is a string, if you want to send the response prettified,
2  you could use something like JSON.stringify(anObject, null, 3)
3  It's important that you set the Content-Type header to application/json, too.
4  */
5
6  var express = require('express');
7  var app = express();
8  var fs = require("fs");
9
10 app.get('/listProd', function (req, res) {
11     fs.readFile(__dirname + "/" + "prod.json", 'utf8', function (err, data) {
12         console.log( data );
13         res.end( data );
14     });
15 })
16 //-----list the user by id
17
18
19 app.get('/listProd/:id', function (req, res) {
20     // retrieve all the users
21     fs.readFile(__dirname + "/" + "prod.json", 'utf8', function (err, data) {
22         data = JSON.parse( data );
23         // var user = data.prod[req.params.id-2].title //get user by key, which is Prod 2, ...
24         var user = data.products[req.params.id-2]//get user by key, which is Prod 2, ...
25         console.log( user );
26         res.end( JSON.stringify(user));
27     });
28 })
29
30 var server = app.listen(8888, function () {
31
32     var host = server.address().address
33     var port = server.address().port
34
35     console.log("Address is http://%s:%s", host, port)
36
37 })
```

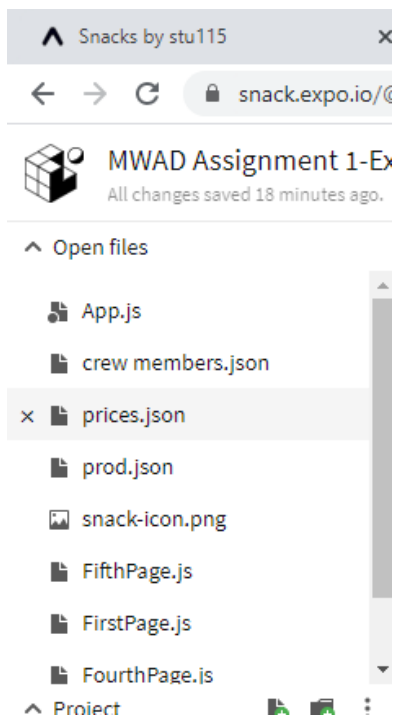
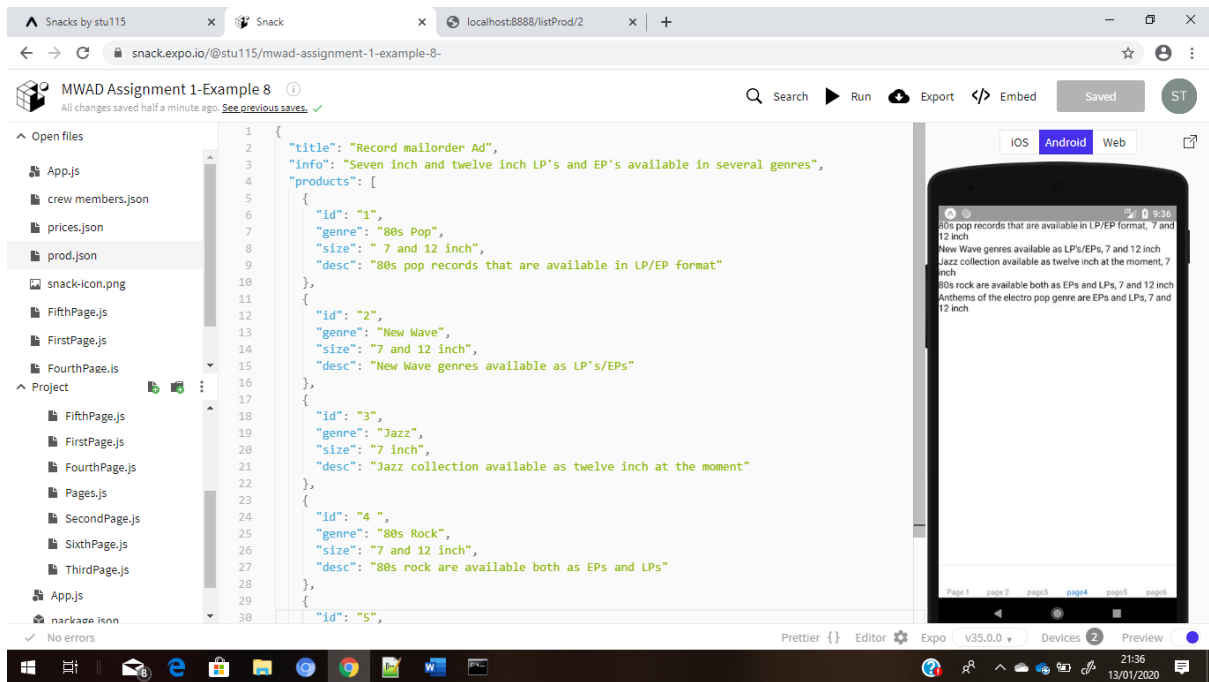
JavaScript file length: 1,203 lines: 37 Ln: 24 Col: 48 Sel: 1 | 1 Windows (CR LF) UTF-8 INS 21:32 13/01/2020

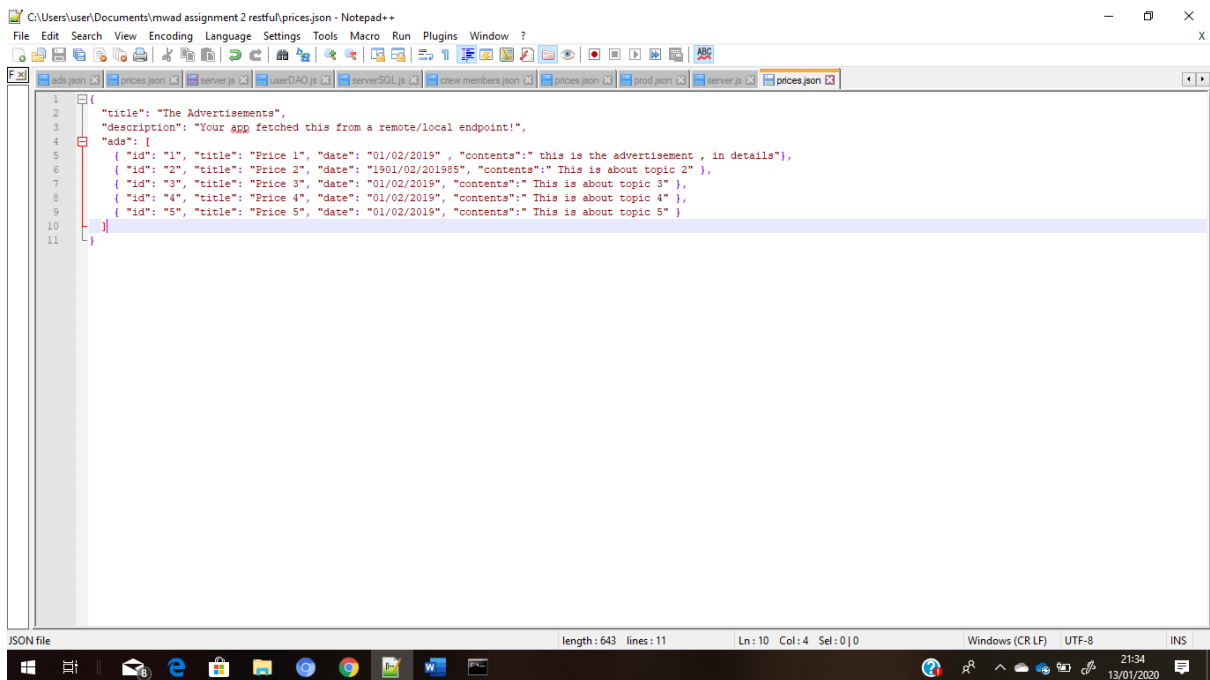
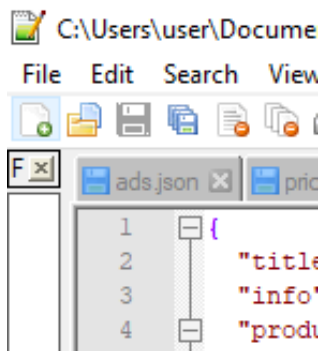
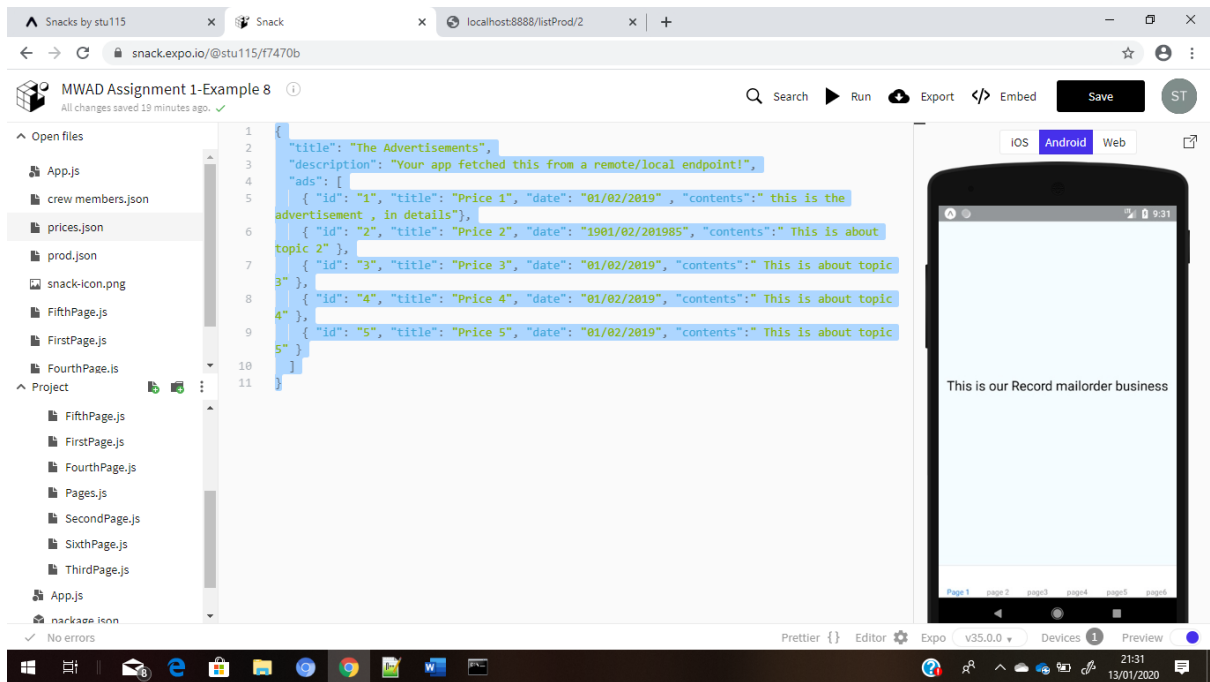


It is important to make sure the word products is in the server code to make a particular product be listed by its id.

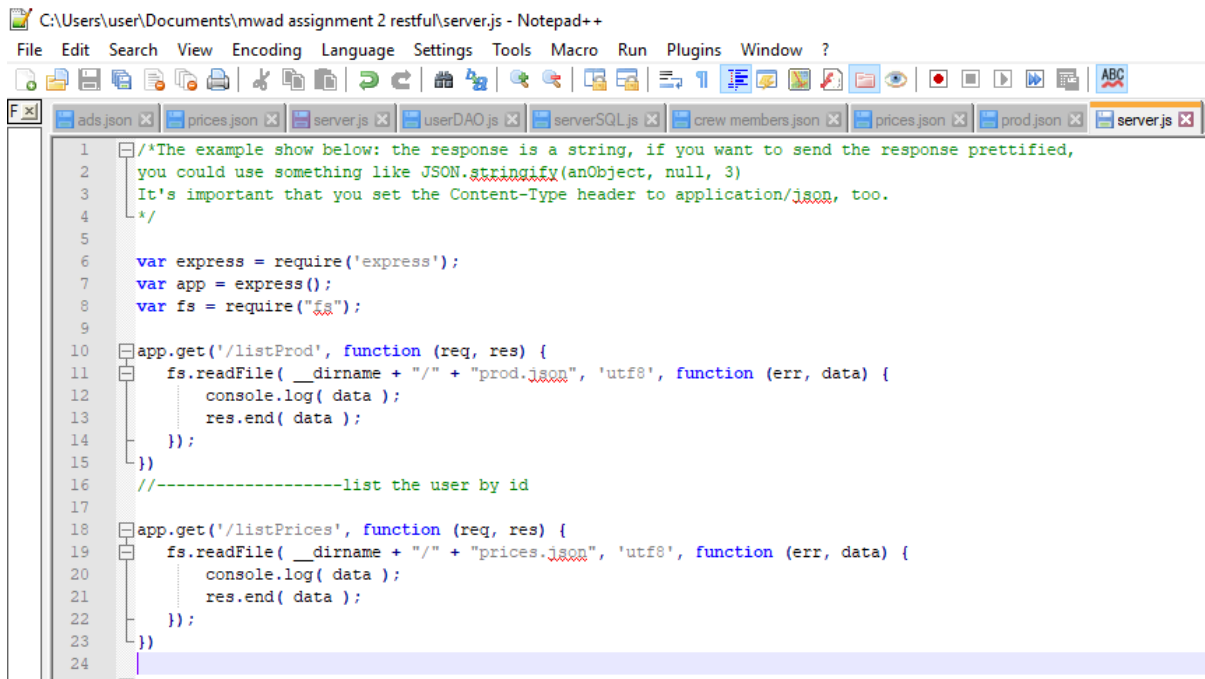
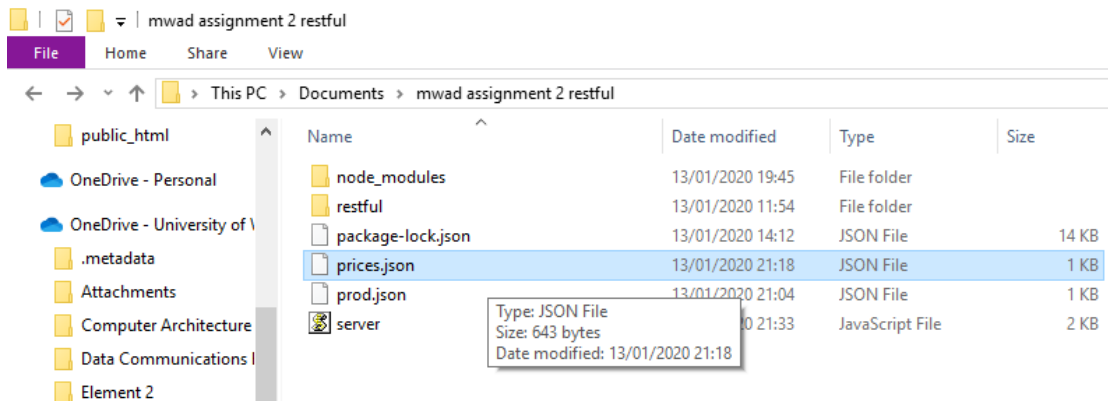












Command Prompt - node server.js

```
{
  "id": "3",
  "genre": "Jazz",
  "size": "7 inch",
  "desc": "Jazz collection available as twelve inch at the moment"
},
{
  "id": "4 ",
  "genre": "80s Rock",
  "size": "7 and 12 inch",
  "desc": "80s rock are available both as EPs and LPs"
},
{
  "id": "5",
  "genre": "Electro Pop",
  "size": "7 and 12 inch",
  "desc": "Anthems of the electro pop genre are EPs and LPs"
}
]
}
{
  id: '2',
  genre: 'New Wave',
  size: '7 and 12 inch',
  desc: "New Wave genres available as LP's/EPs"
}
^C
C:\Users\user\Documents\mwad assignment 2 restful>node server.js
Address is http://:::8888
```

Snacks by stu115

Snack

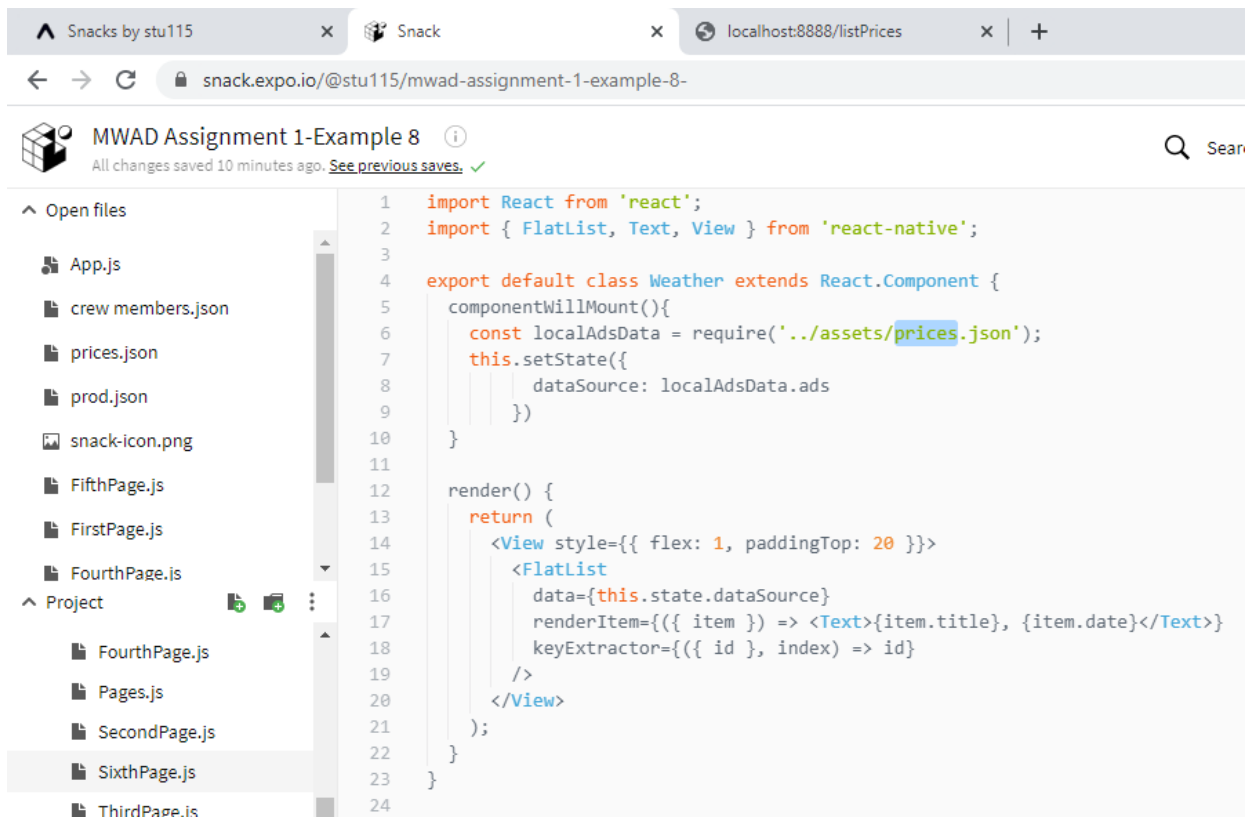
localhost:8888/listPrices

localhost:8888/listPrices

```
{
  "title": "The Advertisements",
  "description": "Your app fetched this from a remote/local endpoint!",
  "ads": [
    { "id": "1", "title": "Price 1", "date": "01/02/2019", "contents": "this is the advertisement , in details"},
    { "id": "2", "title": "Price 2", "date": "1901/02/201985", "contents": " This is about topic 2" },
    { "id": "3", "title": "Price 3", "date": "01/02/2019", "contents": " This is about topic 3" },
    { "id": "4", "title": "Price 4", "date": "01/02/2019", "contents": " This is about topic 4" },
    { "id": "5", "title": "Price 5", "date": "01/02/2019", "contents": " This is about topic 5" }
  ]
}
```

```
Command Prompt - node server.js

},
{
  "id": "5",
  "genre": "Electro Pop",
  "size": "7 and 12 inch",
  "desc": "Anthems of the electro pop genre are EPs and LPs"
}
]
}
id: '2',
genre: 'New Wave',
size: '7 and 12 inch',
desc: 'New Wave genres available as LP's/EPs'
}
C:\Users\user\Documents\mwad assignment 2 restful>node server.js
Address is http://:::8888
{
  "title": "The Advertisements",
  "description": "Your app fetched this from a remote/local endpoint!",
  "ads": [
    { "id": "1", "title": "Price 1", "date": "01/02/2019", "contents": "this is the advertisement, in detail" },
    { "id": "2", "title": "Price 2", "date": "1901/02/201985", "contents": "This is about topic 2" },
    { "id": "3", "title": "Price 3", "date": "01/02/2019", "contents": "This is about topic 3" },
    { "id": "4", "title": "Price 4", "date": "01/02/2019", "contents": "This is about topic 4" },
    { "id": "5", "title": "Price 5", "date": "01/02/2019", "contents": "This is about topic 5" }
  ]
}
```



Snacks by stu115

snack.expo.io/

MWAD Assignment 1-Ex  
All changes saved 10 minutes ago, 5

Open files

- App.js
- crew members.json
- prices.json
- prod.json
- snack-icon.png
- FifthPage.js
- FirstPage.js
- FourthPage.js

Project

- FourthPage.js
- Pages.js
- SecondPage.js
- SixthPage.js
- ThirdPage.js
- App.js
- package.json
- README.md

```
    })  
  
    app.get('/listPrices/:id', function (req, res) {  
      // retrieve all the users  
      fs.readFile(__dirname + "/" + "prices.json", 'utf8', function (err, data) {  
        data = JSON.parse( data );  
        // var user = data.ads[req.params.id-5].title //get user by key, which is Price 5, ...  
        var user = data.ads[req.params.id-5]//get user by key, which is Price 5, ...  
        console.log( user );  
        res.end( JSON.stringify(user));  
      });  
    })  
  })  
}
```

Snacks by stu115 | Snack | localhost

localhost:8888/listPrices/5

localhost:8888/listPrices/5

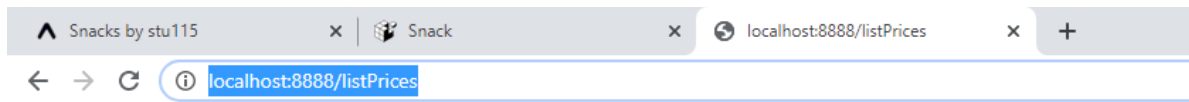
The image shows a screenshot of a Visual Studio Code editor window. The title bar at the top indicates the file path: C:\Users\User\Documents\1m2w assignment 2 restful server.js - Notepad+. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar contains various icons for file operations and development tools. The editor area displays the content of 'server.js', which is a JavaScript file using Express.js to create a REST API. The code includes comments explaining the purpose of each route and the use of JSON.stringify for prettifying responses. The routes are defined for GET, POST, and PUT methods. The status bar at the bottom shows 'JavaScript file', 'length: 1,396 lines: 43', and 'Ln: 10 Col: 19 Sel: 8|1'. The system tray at the very bottom shows the date and time as 21:48 on 13/01/2020.

The image shows a Visual Studio Code editor window with a file named `server.js` open. The code is a JavaScript file for a REST API. It includes comments about JSON responses and headers, and implements endpoints for listing products and users.

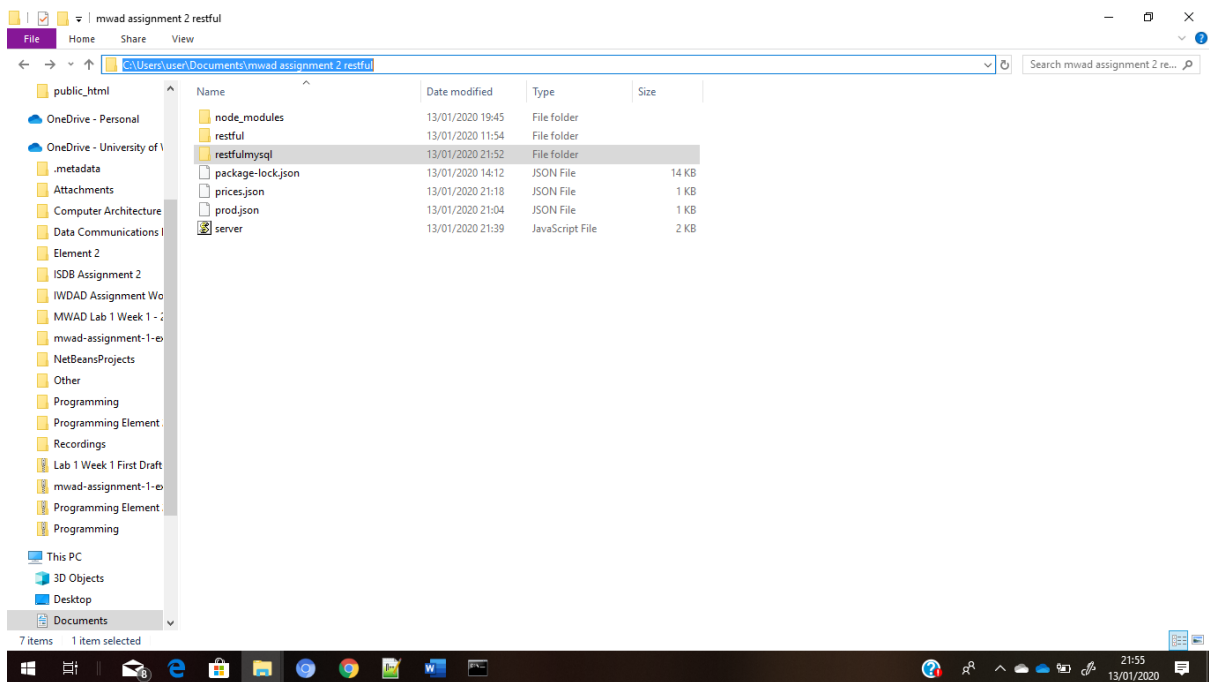
```
1  //The example show below: the response is a string, if you want to send the response prettified,
2  //you could use something like JSON.stringify(anObject, null, 3)
3  //It's important that you set the Content-Type header to application/json, too.
4  //
5
6  var express = require('express');
7  var app = express();
8  var fs = require("fs");
9
10 app.get('/listProd', function (req, res) {
11   fs.readFile(__dirname + "/" + "prod.json", 'utf8', function (err, data) {
12     console.log( data );
13     res.end( data );
14   });
15 })
16 //-----list the user by id
17
18 app.get('/:id', function (req, res) {
19   fs.readFile(__dirname + "/" + "prices.json", 'utf8', function (err, data) {
20     console.log( data );
21     res.end( data );
22   });
23 })
24
25 app.get('/listProd/:id', function (req, res) {
26   // retrieve all the users
27   fs.readFile(__dirname + "/" + "prod.json", 'utf8', function (err, data) {
28     data = JSON.parse( data );
29     // var user = data.prod[req.params.id-2].title //get user by key, which is Prod 2, ...
30     var user = data.products[req.params.id-2]//get user by key, which is Prod 2, ...
31     console.log( user );
32     res.end( JSON.stringify(user));
33   });
34 })
35
36 var server = app.listen(8888, function () {
37
38   var host = server.address().address
```

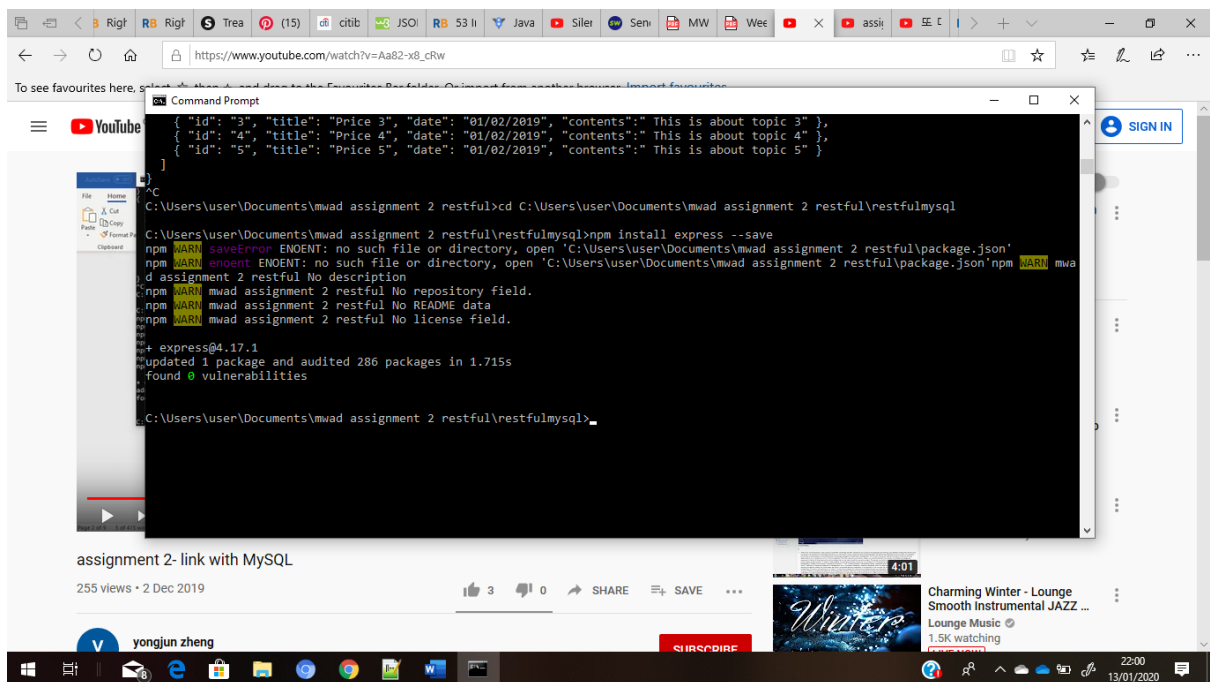
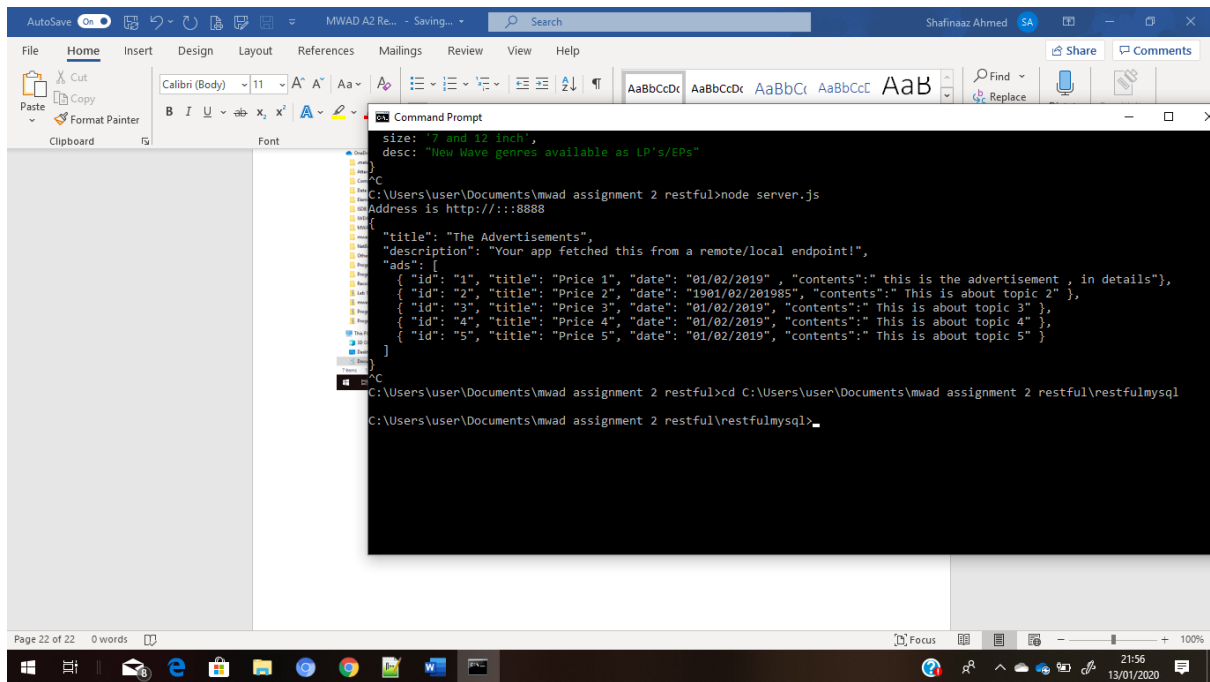
The status bar at the bottom of the editor shows the following information:

- JavaScript file
- length: 1,396 lines: 43
- Ln: 18 Col: 21 Sel: 10 | 1
- Windows (CR LF)
- UTF-8
- INS



```
{
  "title": "The Advertisements",
  "description": "Your app fetched this from a remote/local endpoint!",
  "ads": [
    { "id": "1", "title": "Price 1", "date": "01/02/2019", "contents": "this is the advertisement , in details"},
    { "id": "2", "title": "Price 2", "date": "1901/02/201985", "contents": "This is about topic 2" },
    { "id": "3", "title": "Price 3", "date": "01/02/2019", "contents": "This is about topic 3" },
    { "id": "4", "title": "Price 4", "date": "01/02/2019", "contents": "This is about topic 4" },
    { "id": "5", "title": "Price 5", "date": "01/02/2019", "contents": "This is about topic 5" }
  ]
}
```





YouTube video player showing a tutorial titled "assignment 2- link with MySQL". The video content displays a Windows Command Prompt window with the following commands and output:

```
C:\Users\user\Documents\mwad assignment 2 restful>npm install express --save
npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\user\Documents\mwad assignment 2 restful\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\user\Documents\mwad assignment 2 restful\package.json'
npm WARN mwad assignment 2 restful No description
npm WARN mwad assignment 2 restful No repository field.
npm WARN mwad assignment 2 restful No README data
npm WARN mwad assignment 2 restful No license field.

+ express@4.17.1
updated 1 package and audited 286 packages in 1.715s
found 0 vulnerabilities

C:\Users\user\Documents\mwad assignment 2 restful>npm install mysql
npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\user\Documents\mwad assignment 2 restful\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\user\Documents\mwad assignment 2 restful\package.json'
npm WARN mwad assignment 2 restful No description
npm WARN mwad assignment 2 restful No repository field.
npm WARN mwad assignment 2 restful No README data
npm WARN mwad assignment 2 restful No license field.

+ mysql@2.17.1
added 9 packages from 14 contributors and audited 299 packages in 2.607s
found 0 vulnerabilities

C:\Users\user\Documents\mwad assignment 2 restful>
```

The video player interface shows 255 views, a date of 2 Dec 2019, and a channel named "yongjun zheng". The video progress bar indicates 1:51 / 10:52.

Screenshot of a Windows desktop environment showing a WAMP server setup. The background is a Microsoft Word document titled "MWAD A2 SQL steps - Saved". In the foreground, a web browser window displays the phpMyAdmin interface. The phpMyAdmin page includes a "Welcome to phpMyAdmin" message, a "Language" dropdown menu set to "English", and a "Log in" section with fields for "Username", "Password", and "Server Choice" (set to "MySQL"). A "Go" button is located at the bottom right of the login section. The Windows taskbar at the bottom shows the system clock as 23:16 on 13/01/2020.



