

Node Server

Pages

2- Task Manager

3 – Node Server

4 – Node Server log

5 – Node Bid console/ Atom text editor

6 - Node File structure

7 – Node Routes

8 – Apache Server

9- Workbench

10- Query tables

11- MySQL File Structure

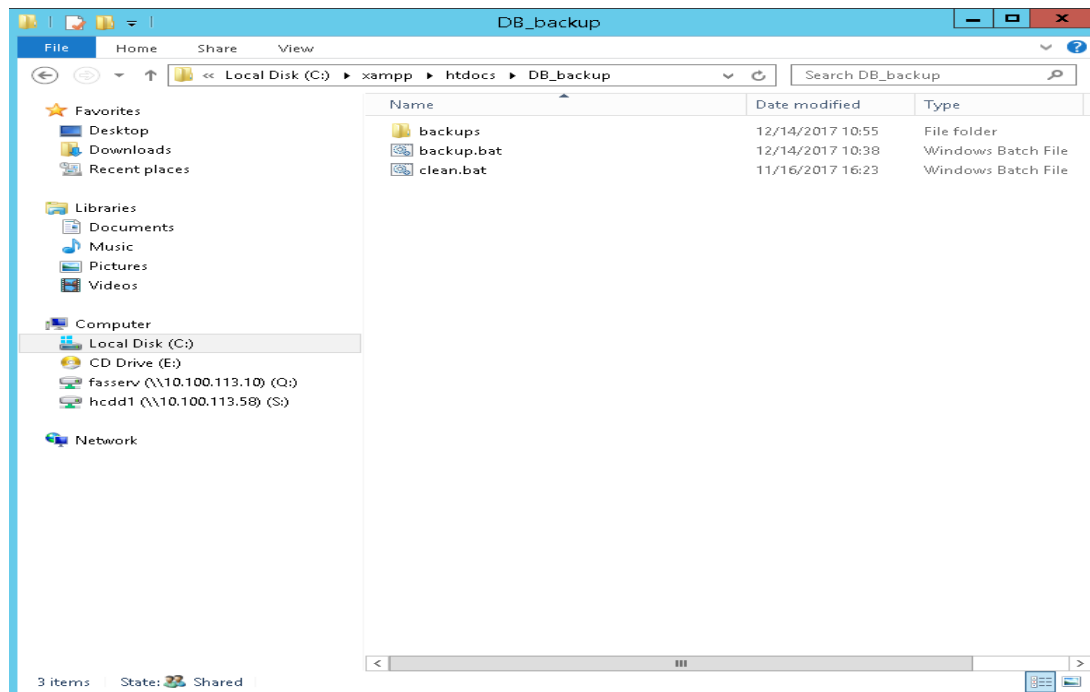
12- Final Notes

Here under **task scheduler** you can find the different tasks that get run automatically.

Task Scheduler					
Name	Status	Triggers	Next Run Time	Last Run Time	Last Ru
DB Backup	Ready	At 17:30 every day	12/14/2017 17:30:30	12/13/2017 17:30:30	The sy:
delete old files	Ready	At 17:30 every Monday of every week, starting 11/16/2017	12/18/2017 17:30:00	12/11/2017 17:30:00	The op
GoogleUpda...	Ready	Multiple triggers defined	12/14/2017 12:17:49	12/13/2017 12:17:49	The op
GoogleUpda...	Ready	At 12:17 every day - After triggered, repeat every 1 hour for a duration of 1 day.	12/14/2017 09:17:50	12/14/2017 08:17:50	The op
node	Ready	At 07:55 every day	12/15/2017 07:55:28	12/14/2017 07:55:28	Incorre
node bid	Run...	At 07:30 every day	12/15/2017 07:30:00	12/14/2017 07:30:00	Incorre
Optimize Sta...	Ready	When computer is idle		Never	
Optimize Sta...	Ready	When computer is idle		7/21/2015 10:32:06	The op
ParetoLogic ...	Ready	At 18:00 every day	12/14/2017 18:00:00	12/13/2017 18:00:00	The op
RemoteArch...	Ready			8/9/2013 13:51:25	The op

In descending order we start with **DB Backup** which is a batch file that backups the database 'time' from MySQL.

You can find the batch file in the following path `C:\xampp\htdocs\DB_backup`



backup.bat as the name implies, creates a backup of the database time every day at 5:30 pm .

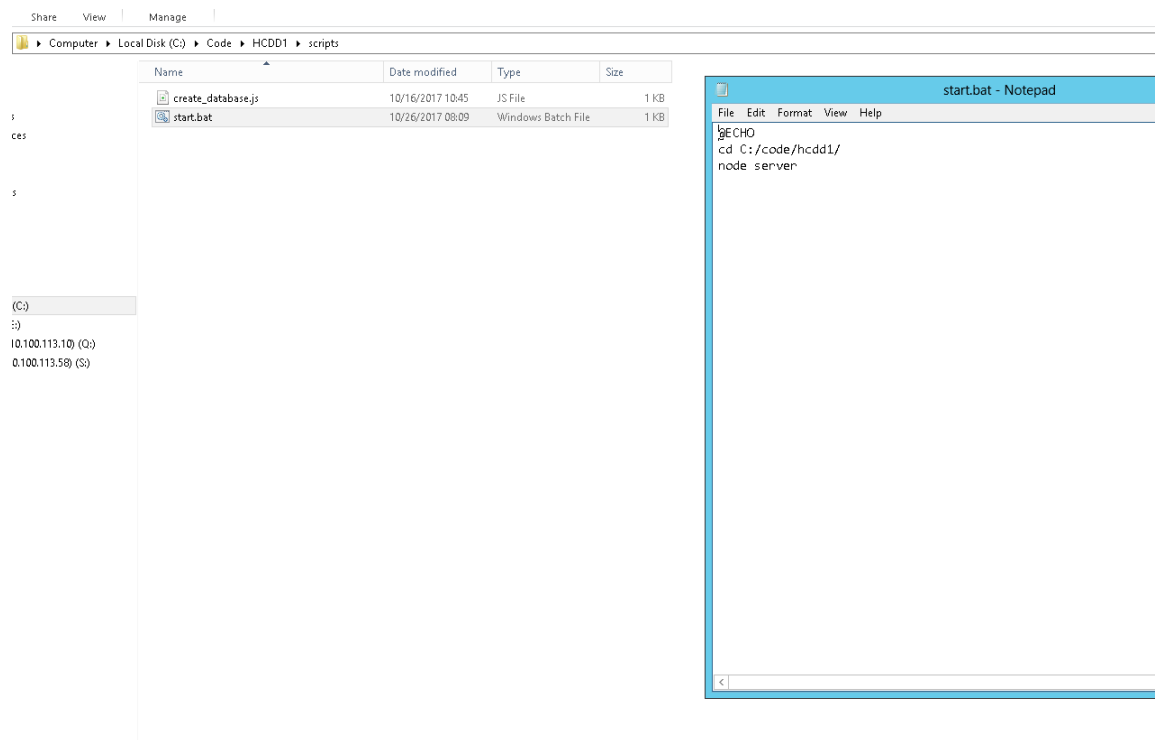
This files are created under the backup folder.

clean.bat cleans old database copies every Monday, it should delete anything older than a week .

Node and **node bid** tasks, are for starting the 2 node servers. Each one is a different application in different ports. Node hccd1 runs on port **8081** and node bid runs on port **8082**.

Task Scheduler					
Name	Status	Triggers	Next Run Time	Last Run Time	Last Run
DB Backup	Ready	At 17:30 every day	12/14/2017 17:30:30	12/13/2017 17:30:30	The sy:
delete old files	Ready	At 17:30 every Monday of every week, starting 11/16/2017	12/18/2017 17:30:00	12/11/2017 17:30:00	The op
GoogleUpda...	Ready	Multiple triggers defined	12/14/2017 12:17:49	12/13/2017 12:17:49	The op
GoogleUpda...	Ready	At 12:17 every day - After triggered, repeat every 1 hour for a duration of 1 day.	12/14/2017 09:17:50	12/14/2017 08:17:50	The op
node	Ready	At 07:55 every day	12/15/2017 07:55:28	12/14/2017 07:55:28	Incorre
node bid	Run...	At 07:30 every day	12/15/2017 07:30:00	12/14/2017 07:30:00	Incorre
Optimize Sta...	Ready	When computer is idle		Never	
Optimize Sta...	Ready	When computer is idle		7/21/2015 10:32:06	The op
ParetoLogic ...	Ready	At 18:00 every day	12/14/2017 18:00:00	12/13/2017 18:00:00	The op
RemoteArch...	Ready			8/9/2013 13:51:25	The op

The node task runs the following batch file *C:\Code\HCDD1\scripts\start.bat*



As you can see, the batch file is quite simple, it just changes the folder to *C:\code\hccd1* and then executes node with the argument *server*. All the node code exists under the folder *C:\Code\HCDD1*

If you wish you can enter the folder in *cmd* and just type *server node* and press enter and the server should start on port **8081**.

```

MINGW64:/c/Code/HCDD1

Administrator@EDI-1-DD-GAUGES MINGW64 /c/Code/HCDD1 (master)
$ ls
app/      gulpfile.js  node_modules/  README.md  server.js
config/   img/         package.json   scripts/   views/

Administrator@EDI-1-DD-GAUGES MINGW64 /c/Code/HCDD1 (master)
$ node server
The magic happens on port 8081
GET /timesheet 304 180.036 ms - -
GET /static/styles.css 304 9.792 ms - -
GET /static/css/portfolio-item.css 304 14.895 ms - -
GET /styles.css 404 13.194 ms - 149
GET /static/head.png 304 0.812 ms - -
GET /styles.css 404 1.905 ms - 149
GET /static/tin2.png 404 4.372 ms - 154
GET /favicon.ico 404 2.124 ms - 150
Jorge Gonzalez 152043
Project Id:
Type of Project Id: string
Week number:
[ RowDataPacket {
  id: 1,
  User_ID: 152043,

```

You can see the port and also different kinds of logs coming from the JavaScript code.

It's the exact same thing for the second batch file *bid.bat* located at

C:\Code\bid\scripts\bid.bat

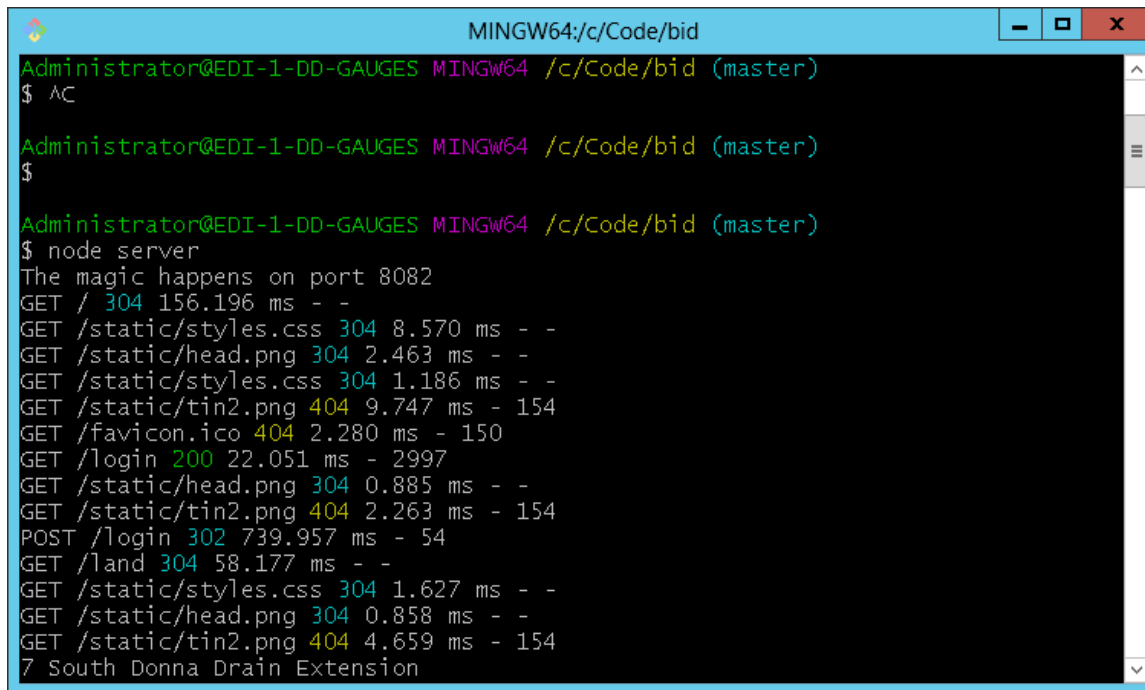
bid.bat	11/30/2017 11:45	Windows Batch File	1 KB
create_database.js	6/28/2015 23:27	JS File	1 KB

```

bid.bat - Notepad
File Edit Format View Help
ECHO
cd C:/code/bid/
node server

```

The console will look the same only that we are running on port **8082**.

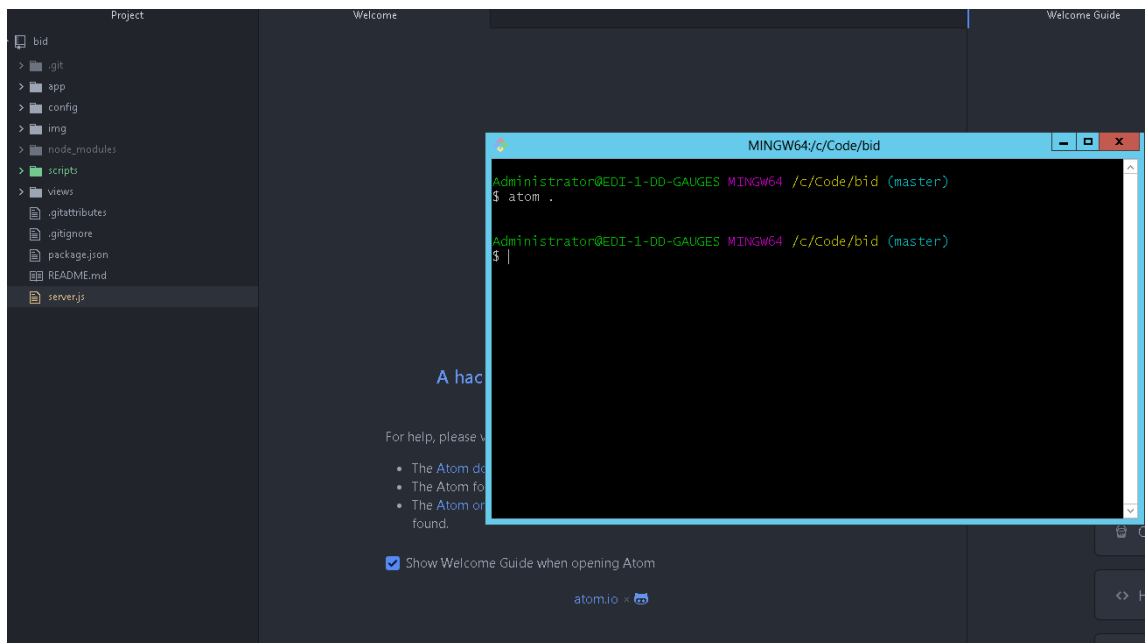


```

MINGW64:/c/Code/bid
Administrator@EDI-1-DD-GAUGES MINGW64 /c/Code/bid (master)
$ AC
Administrator@EDI-1-DD-GAUGES MINGW64 /c/Code/bid (master)
$
Administrator@EDI-1-DD-GAUGES MINGW64 /c/Code/bid (master)
$ node server
The magic happens on port 8082
GET / 304 156.196 ms - -
GET /static/styles.css 304 8.570 ms - -
GET /static/head.png 304 2.463 ms - -
GET /static/styles.css 304 1.186 ms - -
GET /static/tin2.png 404 9.747 ms - 154
GET /favicon.ico 404 2.280 ms - 150
GET /login 200 22.051 ms - 2997
GET /static/head.png 304 0.885 ms - -
GET /static/tin2.png 404 2.263 ms - 154
POST /login 302 739.957 ms - 54
GET /land 304 58.177 ms - -
GET /static/styles.css 304 1.627 ms - -
GET /static/head.png 304 0.858 ms - -
GET /static/tin2.png 404 4.659 ms - 154
7 South Donna Drain Extension
  
```

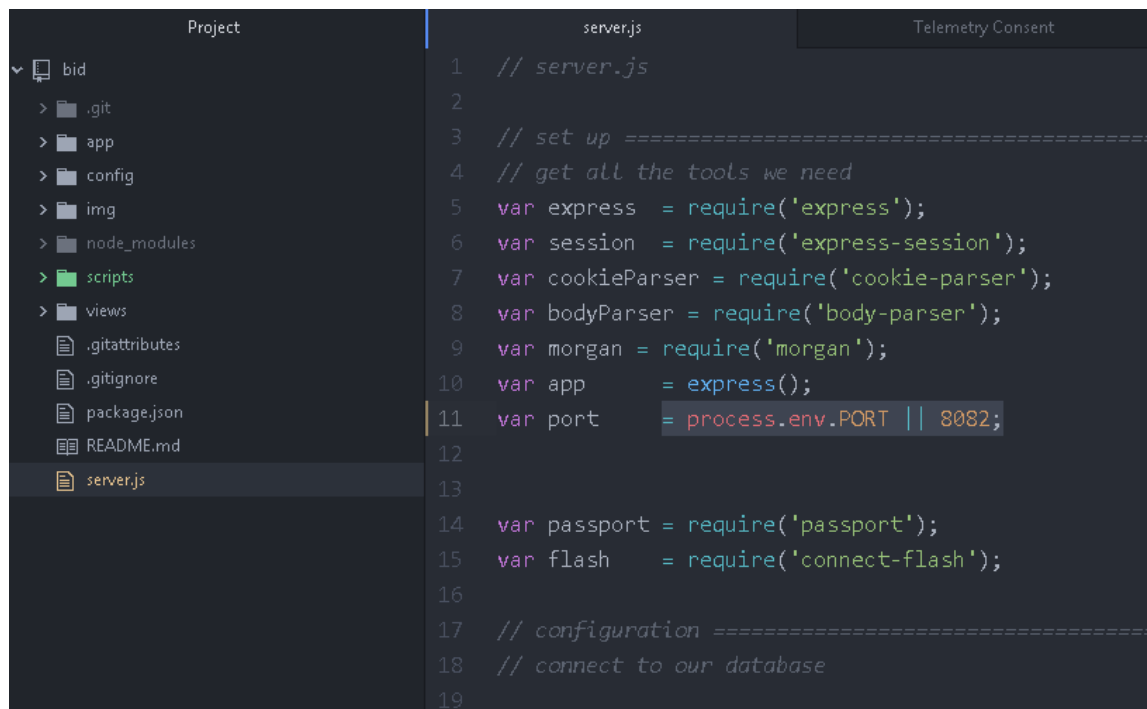
The way I edit code is with the **Atom** word processor by GitHub.

I do that by right clicking inside the folder and then typing *atom* . To open the whole folder in **Atom**.



To the left you can see the file structure which is almost identical in every node project. *server.js* is the file we run using node and starts the whole server.

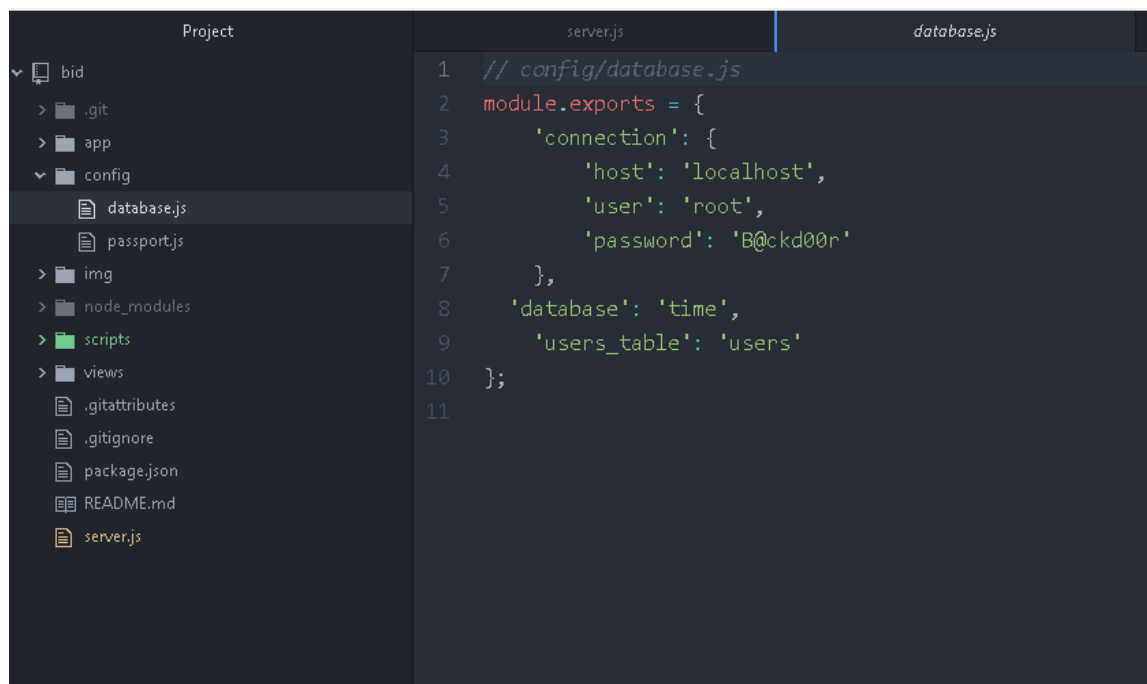
Inside *server.js* you can change the port in which to listen.



```
1 // server.js
2
3 // set up =====
4 // get all the tools we need
5 var express = require('express');
6 var session = require('express-session');
7 var cookieParser = require('cookie-parser');
8 var bodyParser = require('body-parser');
9 var morgan = require('morgan');
10 var app = express();
11 var port = process.env.PORT || 8082;
12
13
14 var passport = require('passport');
15 var flash = require('connect-flash');
16
17 // configuration =====
18 // connect to our database
19
```

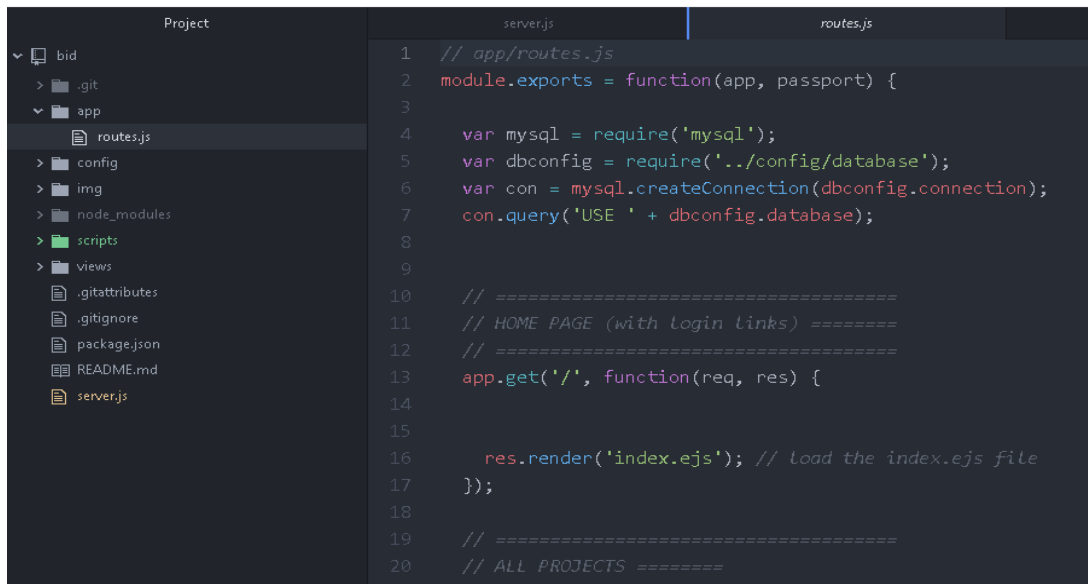
After you modify the file just save the changes and restart the server. First stop it by going to the console and pressing *ctrl+c* and then type *node server* again. Then, changes should take effect.

Under *./bid/config/database.js* you can find the connection file that setups the MySQL connection.



```
1 // config/database.js
2 module.exports = {
3   'connection': {
4     'host': 'localhost',
5     'user': 'root',
6     'password': 'B@ckd00r'
7   },
8   'database': 'time',
9   'users_table': 'users'
10 };
11
```

Under `./bid/app/routes.js` you can find the routes that the server handles.



The screenshot shows a code editor with a project explorer on the left and a code editor on the right. The project explorer shows a folder named 'bid' containing subfolders like '.git', 'app', 'config', 'img', 'node_modules', 'scripts', and 'views'. The 'app' folder is expanded, showing 'routes.js'. The code editor shows the content of 'routes.js' with the following code:

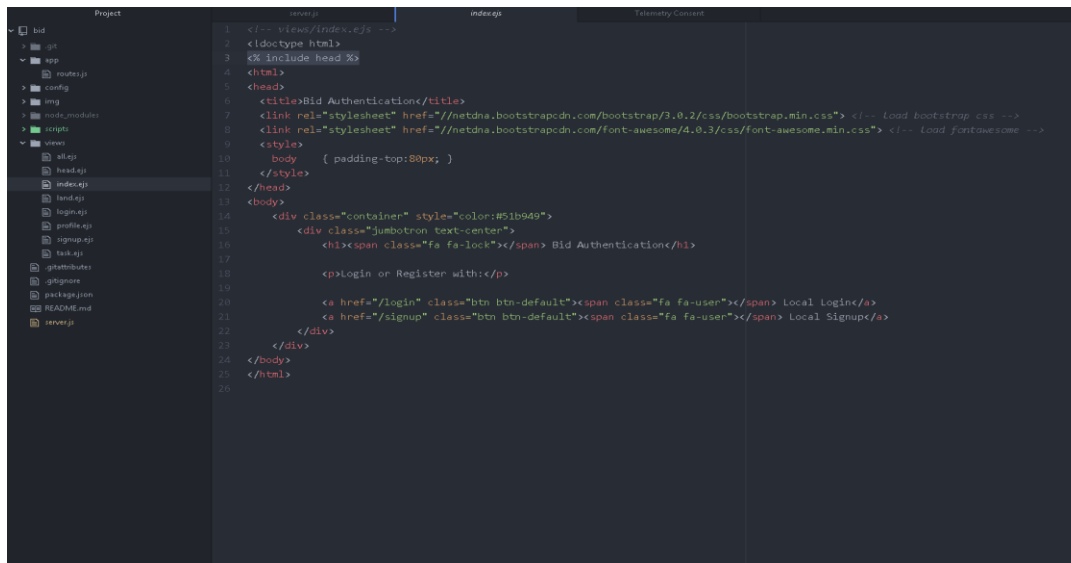
```

1 // app/routes.js
2 module.exports = function(app, passport) {
3
4     var mysql = require('mysql');
5     var dbconfig = require('../config/database');
6     var con = mysql.createConnection(dbconfig.connection);
7     con.query('USE ' + dbconfig.database);
8
9
10    // =====
11    // HOME PAGE (with login links) =====
12    // =====
13    app.get('/', function(req, res) {
14
15
16        res.render('index.ejs'); // Load the index.ejs file
17    });
18
19    // =====
20    // ALL PROJECTS =====

```

In this case you can see how `app.get` gets `'/'` AKA homepage and renders `index.ejs`.

`ejs` is almost identical to `html` but it allows you to insert vanilla JavaScript.



The screenshot shows a code editor with a project explorer on the left and a code editor on the right. The project explorer shows a folder named 'bid' containing subfolders like '.git', 'app', 'config', 'img', 'node_modules', 'scripts', and 'views'. The 'views' folder is expanded, showing 'index.ejs'. The code editor shows the content of 'index.ejs' with the following code:

```

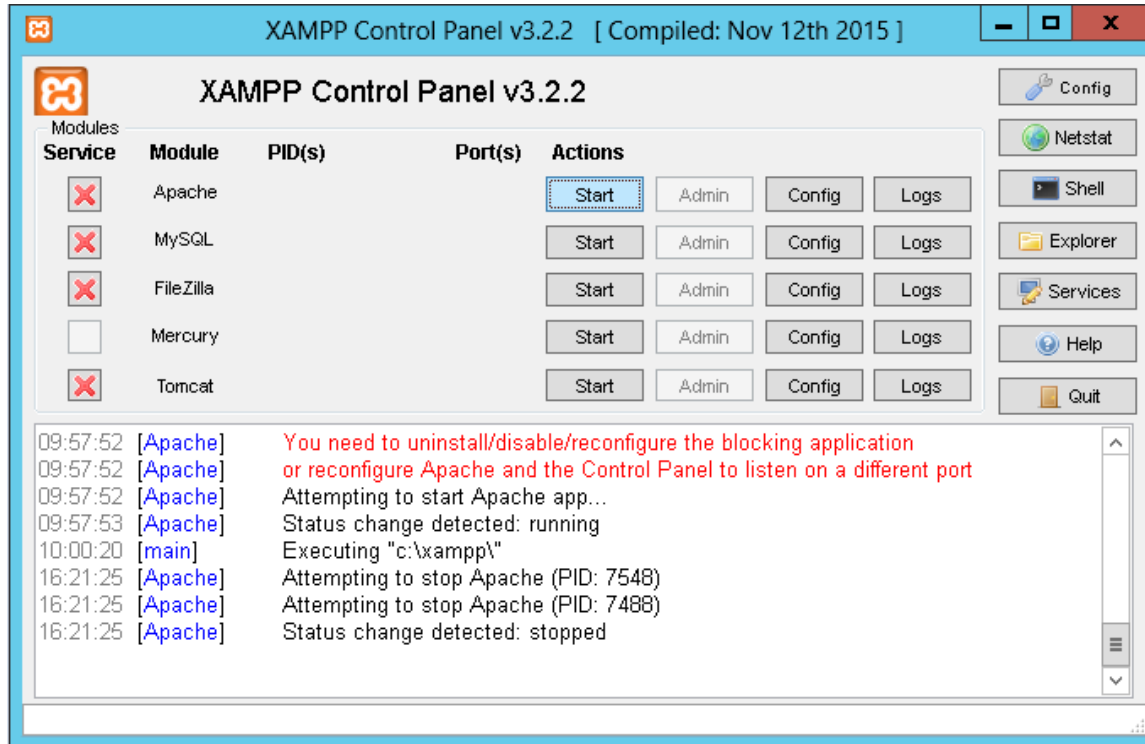
1 <!-- views/index.ejs -->
2 <!doctype html>
3 <% include head.ejs %>
4 <html>
5 <head>
6 <title>Bid Authentication</title>
7 <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.2/css/bootstrap.min.css"> <!-- Load bootstrap css -->
8 <link rel="stylesheet" href="//netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.min.css"> <!-- Load fontawesome -->
9 <style>
10     body { padding-top: 80px; }
11 </style>
12 </head>
13 <body>
14 <div class="container" style="color: #51b949">
15 <div class="jumbotron text-center">
16 <h1><span class="fa fa-lock"></span> Bid Authentication</h1>
17
18 <p>Login or Register with:</p>
19
20 <a href="/login" class="btn btn-default"><span class="fa fa-user"></span> Local Login</a>
21 <a href="/signup" class="btn btn-default"><span class="fa fa-user"></span> Local Signup</a>
22 </div>
23 </div>
24 </body>
25 </html>
26

```

In this case you can see, we insert JavaScript by using `<%` to open and `%>` to close. The code is including head, which means include the `head.ejs` file, which has most of the styling and the header. This way you can include the header easily in any page with one line of code.

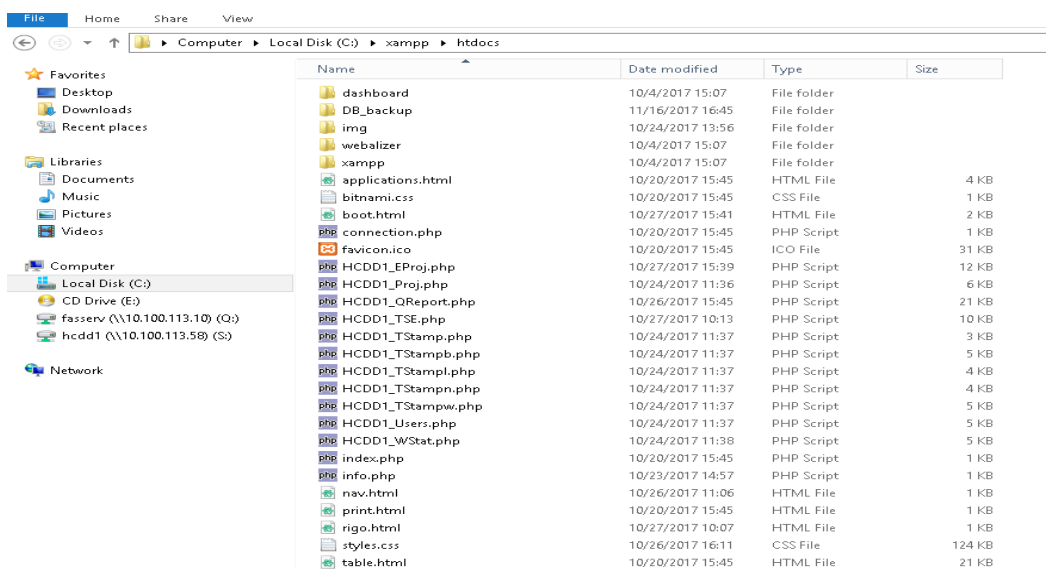
That covers most of the vital parts of the node structure. Any other project should follow a very similar format.

The apache server serves php and there is nothing needed to do. The server should start automatically, however if needed you can always access the xampp controler by pressing the windows key and searching for xampp.

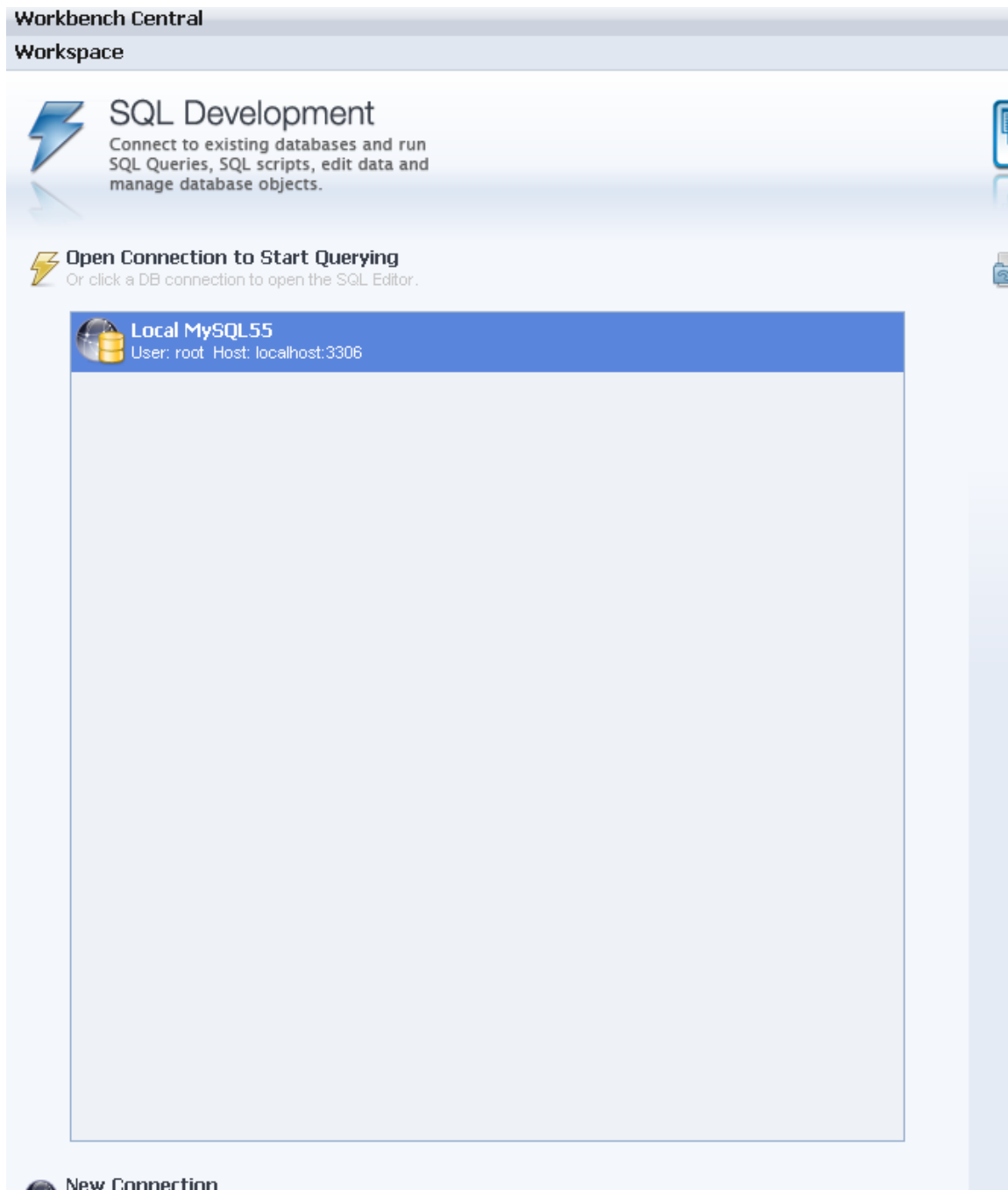


Click on start apache server and it should begin serving php on port **81**

The php and html files served by apache can be found at *C:\xampp\htdocs*



Arguably, the easiest way to check and modify the database is through workbench. To open, is just like **xampp**. Just press the window key and type **workbench** and open the program.



Select local MySQL and enter user=*root* password=*B@ckd00r*

You can also connect from a different computer by specifying the IP address or the computers name instead of 'localhost'.

The database we use, is called time and there are several tables inside it .You can click on any table and then press the lightning button. This will display the contents of the table.

From here you can modify indiviual cells or delete and create whole rows.

The screenshot displays a database management application interface. On the left, the 'Object Browser' shows a tree structure under the 'time' schema, with 'Tables' expanded and 'hcdd1prj' selected. The 'Information' panel at the bottom left provides details for the 'hcdd1prj' table, including its columns and data types. The central 'SQL Editor' shows a query: `SELECT * FROM time.hcdd1prj;`. The bottom right features an 'Output' panel with a dropdown menu set to 'Action Output'.

Object Browser

SCHEMAS

- sys
- time
 - Tables
 - hcdd1prj
 - hcdd1swt
 - hcdd1task
 - hcdd1user
 - users
 - users_image
 - Views
 - Routines

Information

Table: **hcdd1prj**

Columns:

Column Name	Data Type
Proj_id	int(11) PK AI
ProjNo	text
Owner	char(8)
Pct	char(6)
ProjOfficial	varchar(8000)
ProjDesc	varchar(8000)
EngName	text
Estimate	decimal(12,2)
AcctNo	text
ContractNo	text
StartDate	text
EndDate	text
PList	char(1)
Status	varchar(5)

SQL Editor

SQL File 1* Query 1 Query 2 Query 3 Query 4 Query 5 Query 6 Query 7

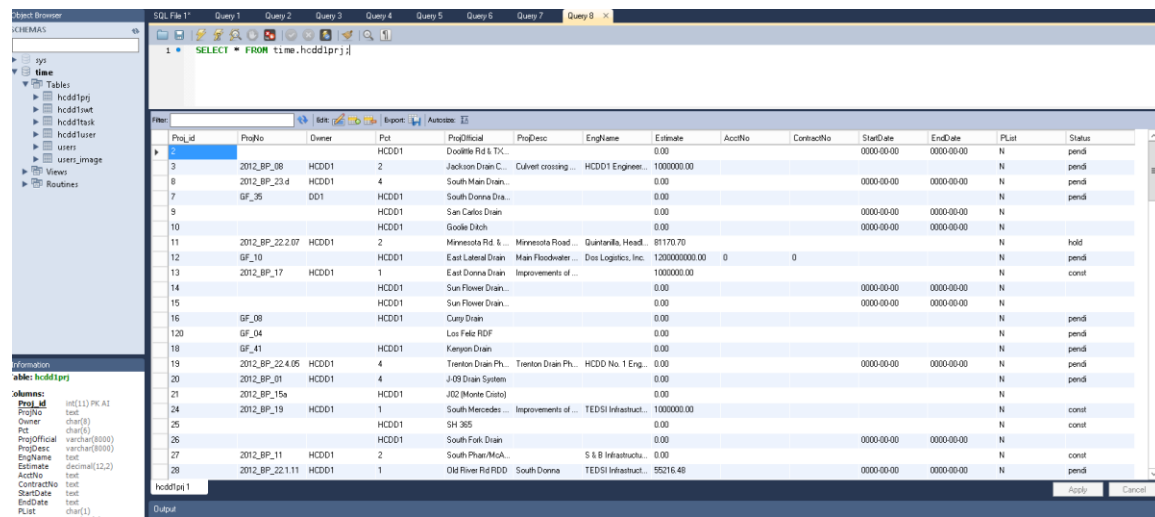
1 • `SELECT * FROM time.hcdd1prj;`

Output

Action Output

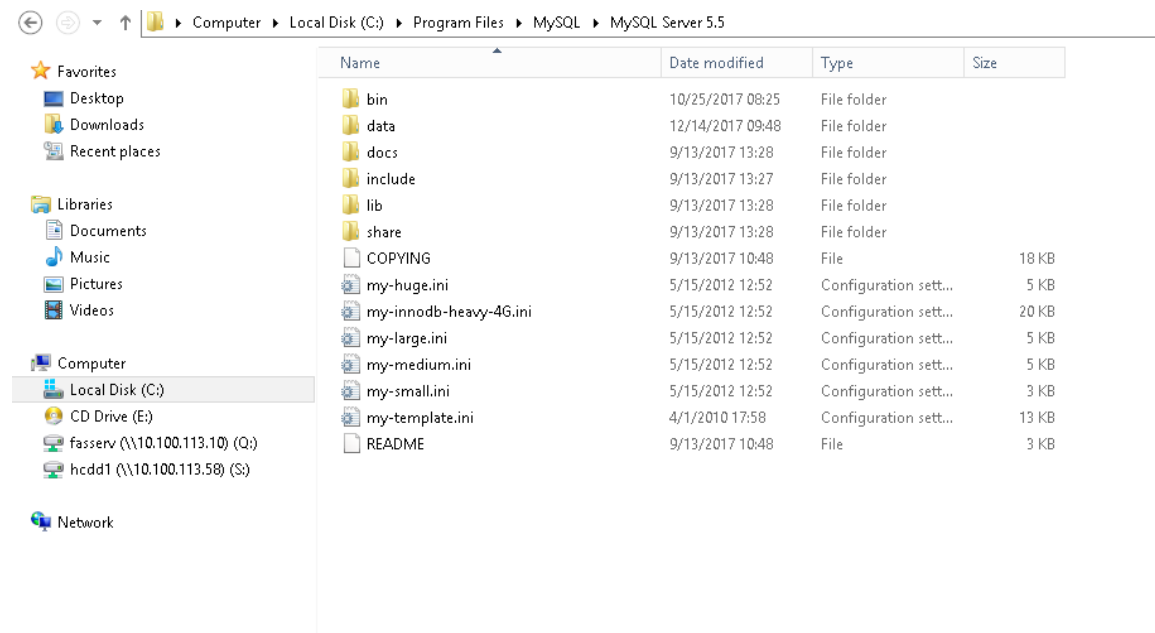
Pretty much any operation that can be done through cml, can be done through workbench.

All the typical CRUD operations.



Finally the MySQL version running in the server is the 5.5 this is for compatibility purposes.

You can find the software under C:\Program Files\MySQL\MySQL Server 5.5



Binary files and executables are under bin.

So if the MySQL server is down, you can run it from there

★ Favorites				
Desktop				
Downloads				
Recent places				
Libraries				
Documents				
Music				
Pictures				
Videos				
Computer				
Local Disk (C:)				
CD Drive (E:)				
fasserv (\\10.100.113.10) (Q:)				
hcdd1 (\\10.100.113.58) (S:)				
Network				

Name	Date modified	Type	Size
echo.exe	9/13/2017 13:14	Application	23 KB
innochecksum.exe	9/13/2017 13:15	Application	4,716 KB
libmecab.dll	9/6/2017 17:16	Application extens...	1,797 KB
lz4_decompress.exe	9/13/2017 13:14	Application	105 KB
my_print_defaults.exe	9/13/2017 13:14	Application	4,189 KB
myisam_ftdump.exe	9/13/2017 13:15	Application	4,486 KB
myisamchk.exe	9/13/2017 13:15	Application	4,598 KB
myisamlog.exe	9/13/2017 13:15	Application	4,436 KB
myisampack.exe	9/13/2017 13:15	Application	4,510 KB
mysql.exe	9/13/2017 13:15	Application	4,957 KB
mysql_client_test_embedded.exe	9/13/2017 13:20	Application	24,566 KB
mysql_config.pl	9/13/2017 13:06	PL File	8 KB
mysql_config_editor.exe	9/13/2017 13:15	Application	4,593 KB
mysql_convert_table_format.pl	5/15/2012 12:52	PL File	5 KB
mysql_embedded.exe	9/13/2017 13:20	Application	24,125 KB
mysql_plugin.exe	9/13/2017 13:15	Application	4,203 KB
mysql_secure_installation.exe	9/13/2017 13:15	Application	4,827 KB
mysql_secure_installation.pl	5/15/2012 12:52	PL File	10 KB
mysql_ssl_rsa_setup.exe	9/13/2017 13:14	Application	4,272 KB
mysql_tzinfo_to_sql.exe	9/13/2017 13:14	Application	4,128 KB

I think this should cover most of the inner works of the server, if there is any remaining questions please contact me at rigoberto.resendez@hcdd1.org.