

Want to learn coding through videos for free?
Check my youtube channel => [HERE!](https://www.youtube.com/c/CoderMediAx)
(<https://www.youtube.com/c/CoderMediAx>)

SEARCH



LATEST ARTICLES



(<https://mfikri.com/en/blog/react-express-mysql-authentication>)
Register and Login

with JWT Authentication (React JS + Express + MySQL + Node JS) (<https://mfikri.com/en/blog/react-express-mysql-authentication>)
M Fikri Setiadi 23 Dec



(<https://mfikri.com/en/blog/react-crud-codeigniter>) Complete
CRUD CodeIgniter 4

and React JS Tutorial (Full-Stack)
(<https://mfikri.com/en/blog/react-crud-codeigniter>)
M Fikri Setiadi 09 Oct



(<https://mfikri.com/en/blog/node-express-react-mysql>) CRUD
Tutorial using Node

JS, Express, React JS, and MySQL (Full-Stack) (<https://mfikri.com/en/blog/node-express-react-mysql>)
M Fikri Setiadi 14 Sep

Steps to Create Simple CRUD Application using Node.js, Express, and MySQL (<https://mfikri.com/en/blog/nodejs-mysql-crud>)

NOVEMBER 27, 2018 M FIKRI SETIADI

NODE.JS ([HTTPS://MFIKRI.COM/EN/CATEGORY/NODEJS](https://mfikri.com/en/category/nodejs)) 57,378 VIEWS

7 COMMENTS



In this tutorial, I will share with you how to create a CRUD (Create Read Update Delete) application using node.js and mysql.

Not only that, I will also share how to use bootstrap modals for input and update form.

So that the CRUD application will be responsive and user friendly.

Let's get started.

Step #1. Introduction

This is important!

In this tutorial, you will learn how to create CRUD operations with node.js and mysql.

If you haven't known node.js before, I recommend you to read the previous post about: **Node.js tutorial for beginners** (<http://mfikri.com/en/blog/nodejs-tutorial>).



(<https://mfikri.com/en/blog/reactjs-beginners>) React JS Tutorial For Beginners

(React Hooks) (<https://mfikri.com/en/blog/reactjs-beginners>)

M Fikri Setiadi 09 Sep



(<https://mfikri.com/en/blog/codeigniter-login-jwt>) CodeIgniter 4 Login and Register

with JWT (JSON Web Token)

(<https://mfikri.com/en/blog/codeigniter-login-jwt>)

M Fikri Setiadi 01 Sep



(<https://mfikri.com/en/blog/codeigniter-vuejs-crud>) Complete CRUD CodeIgniter 4

and Vue JS Tutorial (Full Stack)

(<https://mfikri.com/en/blog/codeigniter-vuejs-crud>)

M Fikri Setiadi 28 Aug



(<https://mfikri.com/en/blog/restful-api-express-mongodb>) How to Build a

RESTful API Using Node js, Express, and MongoDB (<https://mfikri.com/en/blog/restful-api-express-mongodb>)

M Fikri Setiadi 29 Jul

In this tutorial I will use Bootstrap and JQuery.

If you don't have bootstrap and jquery, please download it on the official website **getbootstrap.com** (<http://getbootstrap.com/>) and **jquery.com** (<http://jquery.com/>).



(<https://mfikri.com/en/blog/vuejs-tutorial-beginners>) Vue JS 3 Tutorial For Beginners

(Options API) (<https://mfikri.com/en/blog/vuejs-tutorial-beginners>)
M Fikri Setiadi 08 Jul



(<https://mfikri.com/en/blog/restful-api-express-sequelize>)
How to Create a

RESTful API Using Node.js Express and Sequelize (<https://mfikri.com/en/blog/restful-api-express-sequelize>)
M Fikri Setiadi 07 May



(<https://mfikri.com/en/blog/codeigniter4-vanilla-javascript>)
CRUD Application

Using CodeIgniter 4 and Vanilla JavaScript (<https://mfikri.com/en/blog/codeigniter4-vanilla-javascript>)
M Fikri Setiadi 19 Jan

Step #2: Creating Database Structure

Create a new database, here I create a database named **crud_db**.

If you create it with the same name it's better.

To create a database in MySQL, it can be done by executing the following query:

```
1 | CREATE DATABASE crud_db;
```

The SQL command above will create a database with the name **crud_db**.

Next, create a table in the **crud_db** database.

Here I create a table named "**product**".

MOST POPULAR



(<https://mfikri.com/en/blog/nodejs-restful-api-mysql>) 5 Steps
Create a RESTful API

using Node.js and MySQL (<https://mfikri.com/en/blog/nodejs-restful-api-mysql>)
M Fikri Setiadi 17 Feb

If you create it with the same name it's better.

To create a **"product"** table, it can be done by executing the SQL command below:

```
1 CREATE TABLE product(  
2 product_id INT(11) PRIMARY KEY AUTO_INCREMENT,  
3 product_name VARCHAR(200),  
4 product_price INT(11)  
5 )ENGINE=INNODB;
```

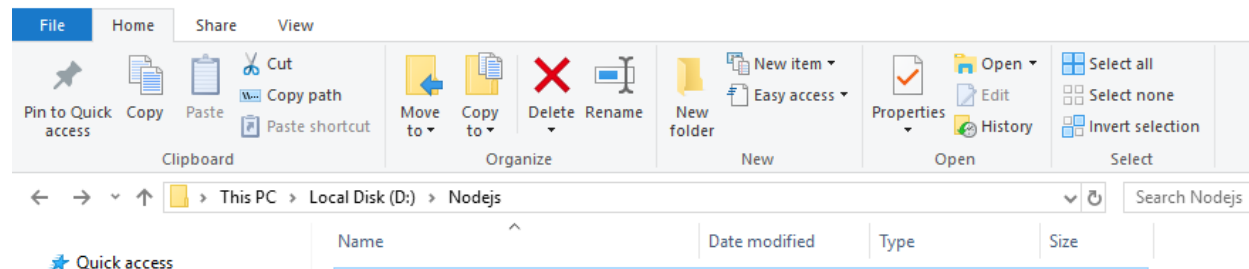
www.facebook.com
n2u=https
rter.com
atus=7%20Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js
com%2Fen%2Fblog%2Fnodejs-
s.google.com
and%20MySQL%20upt
https

Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js
com%2Fen%2Fblog%2Fnodejs-
Express
and%20MySQL)

Step #3. Install Dependencies

Before installing dependencies, please create a folder, here I create a folder and I named **crud**.

like the picture below:



(<https://mfikri.com/en/blog/nodejs-tutorial>) Learn Node.js from Scratch (Step by

Step) (<https://mfikri.com/en/blog/nodejs-tutorial>)

M Fikri Setiadi 23 Aug



(<https://mfikri.com/en/blog/nodejs-express-mysql-vue>) Full Stack Web App

Using Vue.Js, Express Js and MySQL

(Complete Guide) (<https://mfikri.com/en/blog/nodejs-express-mysql-vue>)

M Fikri Setiadi 02 Dec



(<https://mfikri.com/en/blog/restful-api-express-sequelize>) How to Create a

RESTful API Using Node.js Express and Sequelize (<https://mfikri.com/en/blog/restful-api-express-sequelize>)

M Fikri Setiadi 07 May



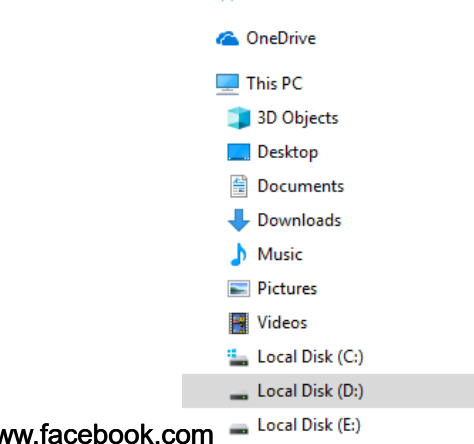
(<https://mfikri.com/en/blog/express-tutorial-beginners>) Express Js Tutorial

For Beginners (<https://mfikri.com/en/blog/express-tutorial-beginners>)

M Fikri Setiadi 16 Nov

VIDEO TUTORIALS:





www.facebook.com

https://www.facebook.com

status=7%20Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs- The crud folder is our project folder in this tutorial.

Express and%20MySQL%20https

Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs- Express and%20MySQL)

In this tutorial, we need 4 dependencies below:

1. **Express** (<http://expressjs.com/>) (node.js framework)

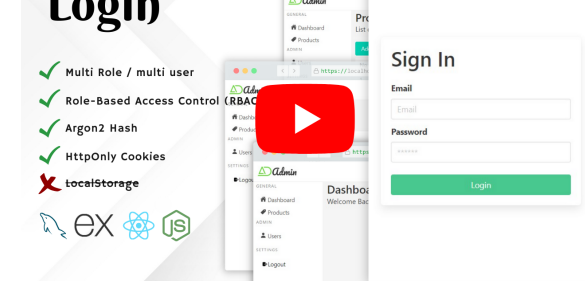
2. **MySQL** (mysql driver for node.js)

3. **Body-parser** (<https://www.npmjs.com/package/body-parser>)

(middleware for handling post body request)

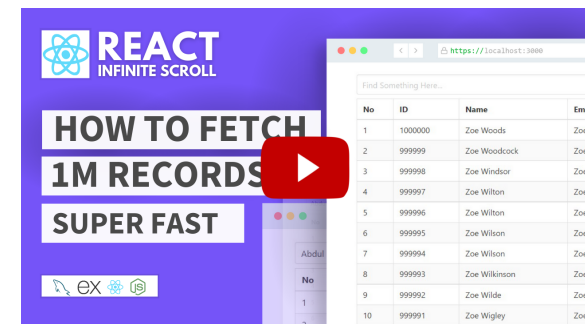
4. **Handlebars** (<https://www.npmjs.com/package/handlebars>)

(template engine)



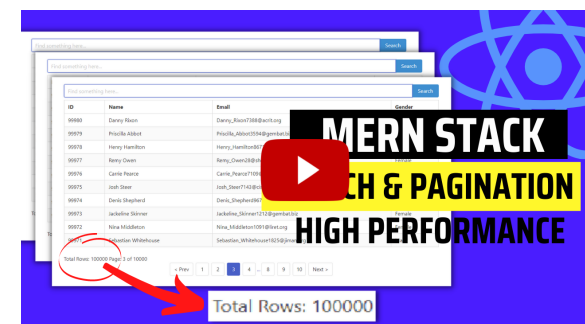
(<https://youtu.be/OJU0L9D-Zdo>)

Login Multi User with Node JS, Express, MySQL, and React JS
(<https://youtu.be/OJU0L9D-Zdo>)



(https://youtu.be/Gv_A7jHsMmk)

How to Create Infinite Scroll with Node JS, Express, React JS, MySQL (Full Stack)
(https://youtu.be/Gv_A7jHsMmk)



(<https://youtu.be/XkRkUSkEgxl>)

Search & Pagination with Node, Express, React, MySQL (Best

To install dependencies on node.js can be done easily using "NPM" (Node Package Manager).

You can run NPM in "Terminal" or "Command Prompt".

But, in this tutorial I don't use Command Prompt, instead I use Git Bash

www.facebook.com

in?u=https

status=7%20Steps%20to%20Create%20Simple%20CRUD%20Application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs-

js.google.com

and%20MySQL%20https

Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL)

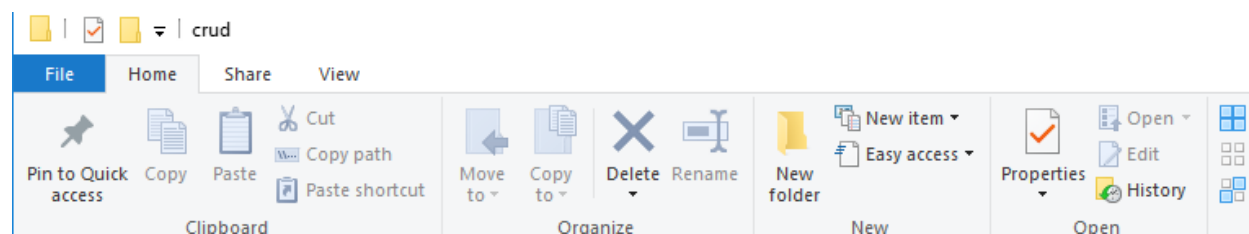
You can download Git Bash in the url below:

<https://git-scm.com/downloads> (<https://git-scm.com/downloads>)

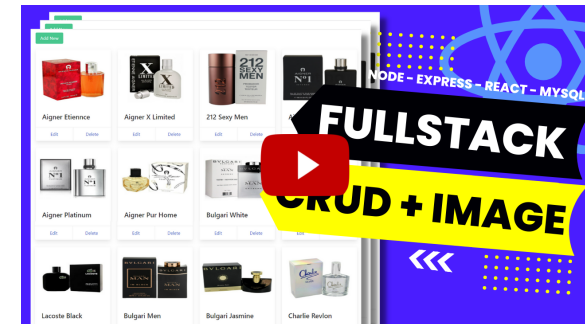
Download it according to your platform, then install it on your computer.

If you have installed "Git Bash", please open the **crud** folder that was created before.

Then right-click and select **Git Bash Here**, like picture below:

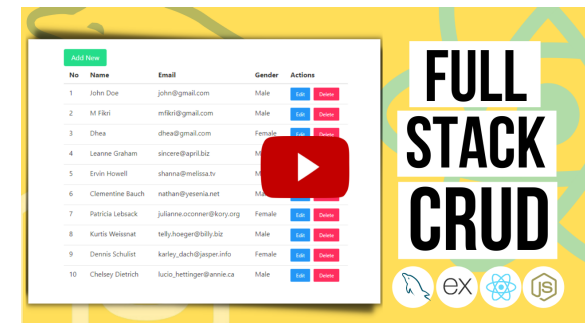


Practice) (<https://youtu.be/XkRkUSkEgxl>)



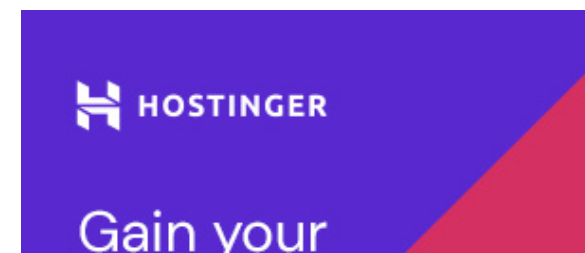
(<https://youtu.be/jPjPGAQOMac>)

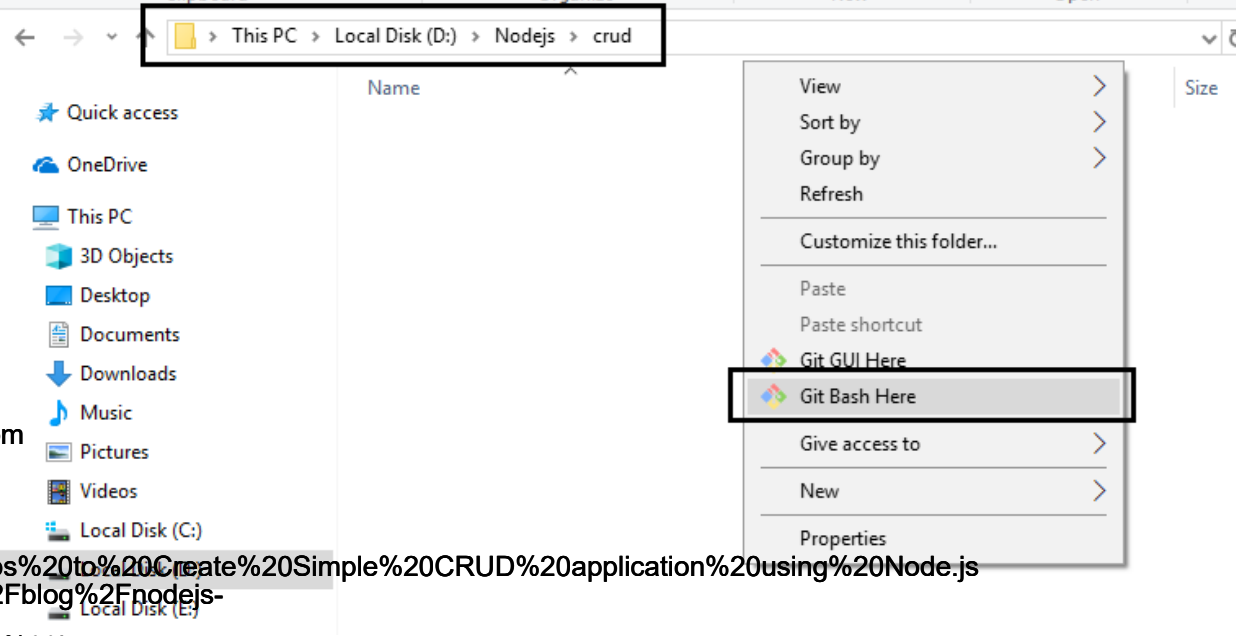
CRUD + Image tutorial with Node JS, Express, React JS, and MySQL (MERN STACK) (<https://youtu.be/jPjPGAQOMac>)



(https://youtu.be/es9_6RFR7wk)

FULL CRUD tutorial with Node JS, Express, React JS and MySQL (Full Stack) (https://youtu.be/es9_6RFR7wk)





www.facebook.com

ter.com

status=7%20Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs-

us.google.com

and%20MySQL%20https

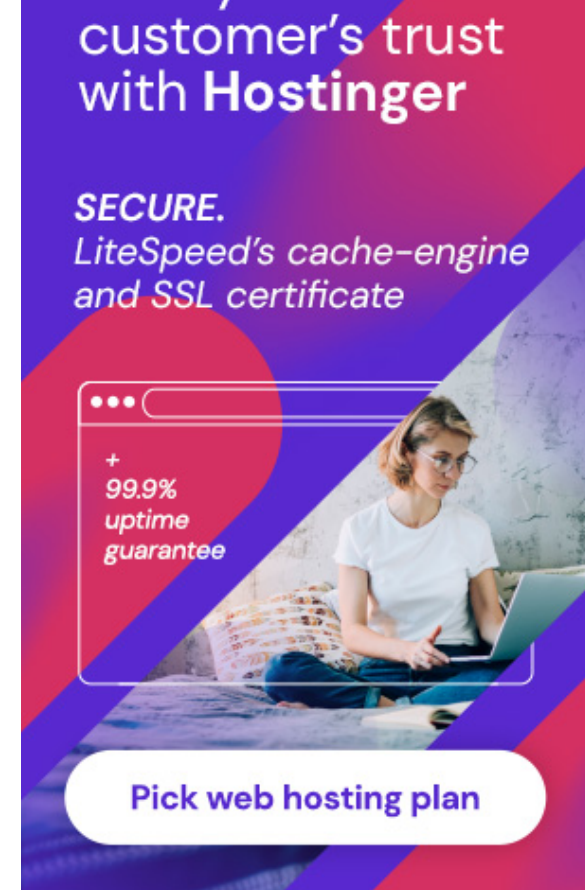
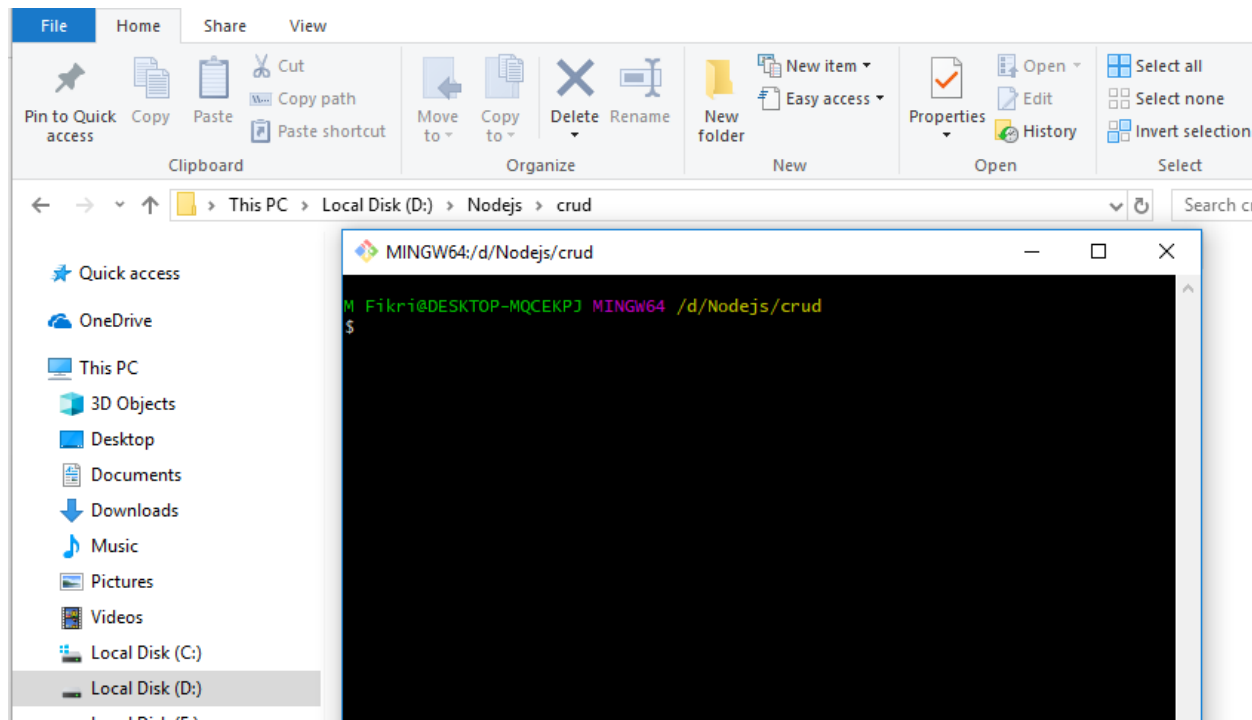
F=https

Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL) Then will appear the git bash terminal like the picture below:



(<https://www.hostg.xyz/SH96M>)

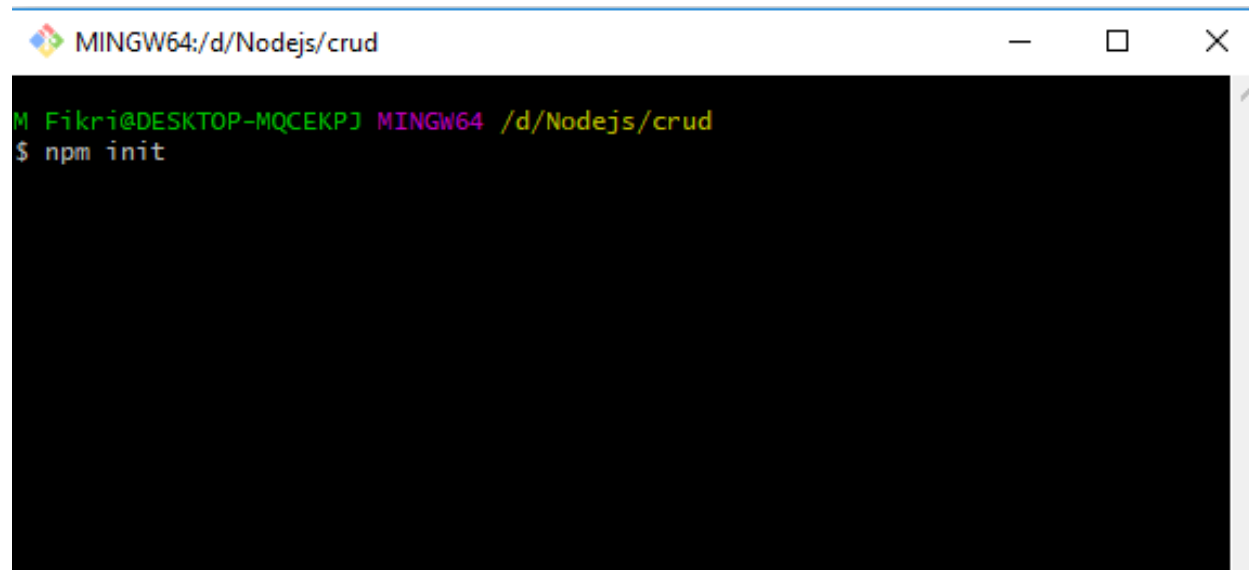
In the terminal above, you can install all the dependencies that you need for your project.

Before installing dependencies, we need to create **package.json** file to our project.

To create package.json file, you can run the following command on the

```
1 | npm init
```

Like the picture below:



```
MINGW64:/d/Nodejs/crud
M Fikri@DESKTOP-MQCEKPJ MINGW64 /d/Nodejs/crud
$ npm init
```


The above command will automatically create a package.json file on your

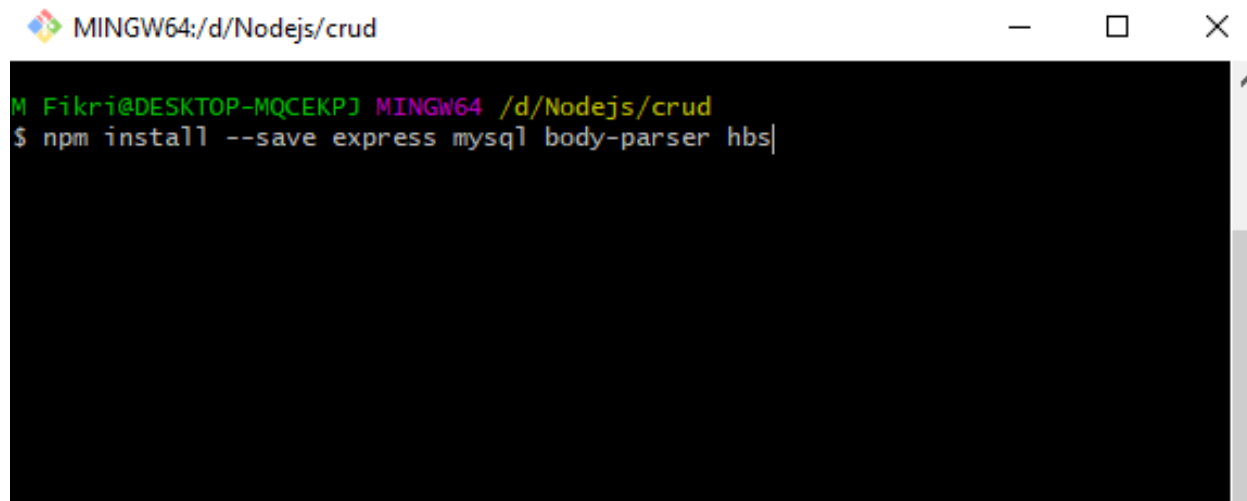
project.

After that, you need to type the following

command in the terminal:

```
1 | npm install --save express mysql body-parser hbs
```

Like the picture below:

A screenshot of a Windows terminal window. The title bar shows 'MINGW64:/d/Nodejs/crud'. The terminal content shows the prompt 'M Fikri@DESKTOP-MQCEKPJ MINGW64 /d/Nodejs/crud' followed by the command '\$ npm install --save express mysql body-parser hbs|'.

```
MINGW64:/d/Nodejs/crud
M Fikri@DESKTOP-MQCEKPJ MINGW64 /d/Nodejs/crud
$ npm install --save express mysql body-parser hbs|
```

The above command will install all the dependencies that you need:

express, mysql, body-parser, and handlebars.

Open the **package.json** file, it will look like this:

```
1 {
2   "name": "crud",
3   "version": "1.0.0",
4   "description": "Crud Node.js",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "M Fikri",
10  "license": "ISC",
11  "dependencies": {
12    "body-parser": "^1.18.3",
13    "express": "^4.16.3",
14    "hbs": "^4.0.1",
15    "mysql": "^2.16.0"
16  }
17 }
```

Step #4. Project Structure

Open the crud folder using text editor, here I use **ATOM** (<https://atom.io/>) as my text editor.

You can use **Sublime Text** (<http://www.sublimetext.com/>), **IntelliJ IDEA** (<https://www.jetbrains.com/idea/>), or any other editors.

Then create a new folder in the crud folder. Here I create two folders, **public** and **views** folder.

Like the picture below:

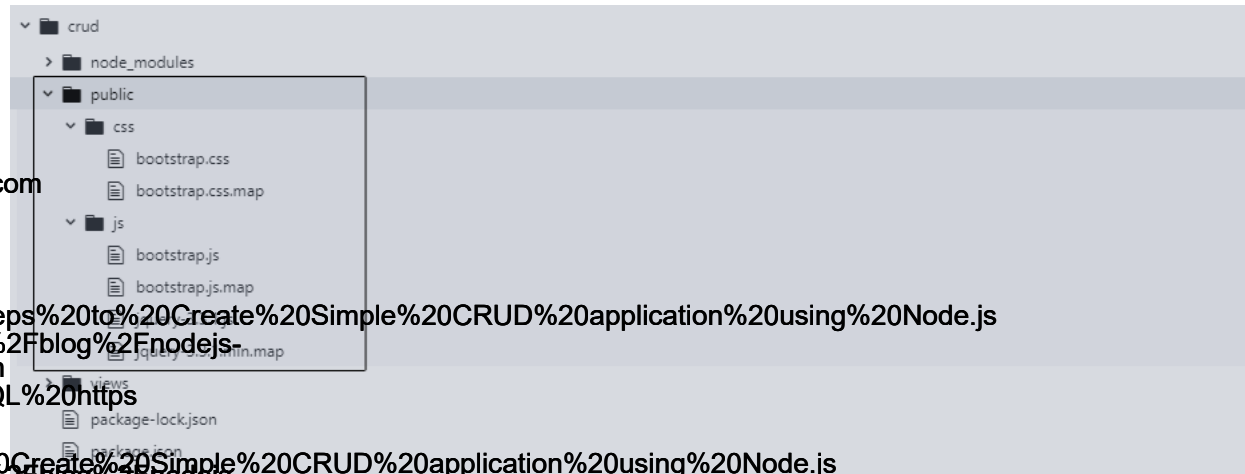


After that create **css** and **js** folder inside the **public** folder.

Then copy the **bootstrap.css** file that was previously downloaded into the **public/css/** folder.

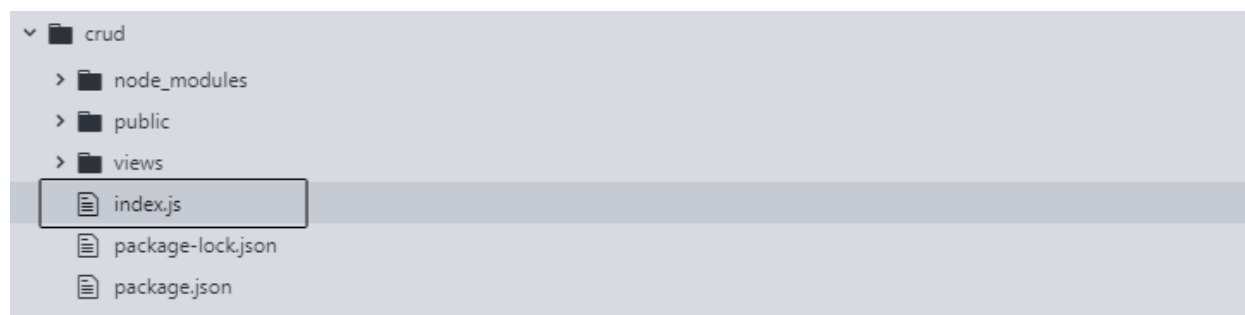
And copy the bootstrap.js and jquery files in the **public/js/** folder.

So that our project structure looks like the picture below:



Step #5. Index.js

Create a file named **index.js**. like the picture below:



And then open **index.js** and type the code below:

www.facebook.com

u=https
r.com

atus=7%20Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

L.com%2Fen%2Fblog%2Fnodejs-

us.google.com

and%20MySQL%20https

F=https

eps%20te%20Create%20Simple%20CRUD%20application%20using%20Node.js

L.com%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL)

```

1 //use path module
2 const path = require('path');
3 //use express module
4 const express = require('express');
5 //use hbs view engine
6 const hbs = require('hbs');
7 //use bodyParser middleware
8 const bodyParser = require('body-parser');
9 //use mysql database
10 const mysql = require('mysql');
11 const app = express();
12
13 //Create connection
14 const conn = mysql.createConnection({
15   host: 'localhost',
16   user: 'root',
17   password: '',
18 });
19
20 //connect to database
21 conn.connect((err) => {
22   console.log('Mysql Connected...');
23 });
24
25 //set views file
26 app.set('views', path.join(__dirname, 'views'));
27 //set view engine
28 app.set('view engine', 'hbs');
29 app.use(bodyParser.json());
30 app.use(bodyParser.urlencoded({ extended: false }));
31 //set public folder as static folder for static file
32 app.use('/assets', express.static(__dirname + '/public'));
33
34 //route for homepage
35 app.get('/', (req, res) => {
36   let sql = "SELECT * FROM product";
37   let query = conn.query(sql, (err, results) => {
38     if(err) throw err;
39     res.render('product_view', {
40       results: results
41     });
42   });
43 });
44
45 //route for insert data
46 app.post('/save', (req, res) => {
47   let data = {product_name: req.body.product_name, product_price: req.body.
48   let sql = "INSERT INTO product SET ?";
49   let query = conn.query(sql, data, (err, results) => {

```



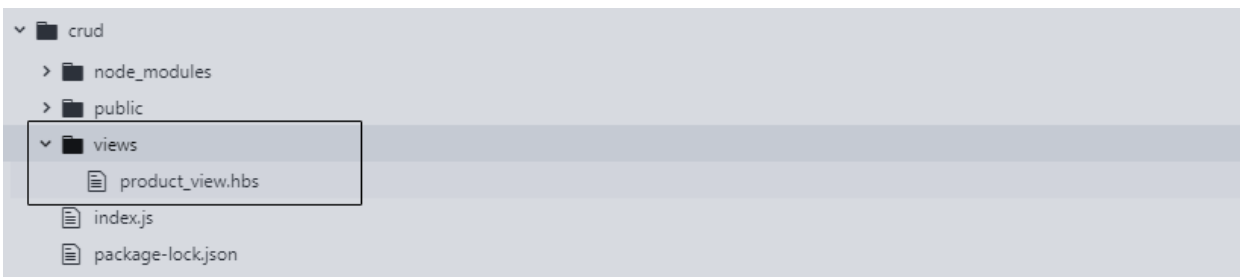
```

52     if(err) throw err;
53     res.redirect('/');
54   });
55 });
56
57 //route for update data
58 app.post('/update',(req, res) => {
59   let sql = "UPDATE product SET product_name='"+req.body.product_name+"', {
60   let query = conn.query(sql, (err, results) => {
61     if(err) throw err;
62     res.redirect('/');
63   });
64 });
65
66 //route for delete data
67 app.post('/delete',(req, res) => {
68   let sql = "DELETE FROM product WHERE product_id='"+req.body.product_id+"'";
69   let query = conn.query(sql, (err, results) => {
70     if(err) throw err;
71     res.redirect('/');
72   });
73 });
74
75 //listening
76 app.listen(8000, () => {
77   console.log('Server is running at port 8000');
78 });

```

Step #6. View

Create a file named **product_view.hbs** in the views folder. Like the following picture:



And then open **product_view.hbs** and type the following code:

www.facebook.com

u=https
r.com

atus=7%20Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

Lcom%2Fen%2Fblog%2Fnodejs-

us.google.com

and%20MySQL%20https

F=https

eps%20te%20Create%20Simple%20CRUD%20application%20using%20Node.js

Lcom%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL)

```

1  <html lang="en">
2  <head>
3      <meta charset="utf-8">
4      <title>CRUD Node.js and Mysql</title>
5      <link href="/assets/css/bootstrap.css" rel="stylesheet" type="text/css"/>
6  </head>
7  <body>
8      <div class="container">
9          <h2>Product List</h2>
10         <button class="btn btn-success" data-toggle="modal" data-target="#>
11         <table class="table table-striped" id="mytable">
12             <thead>
13                 <tr>
14                     <th>Product ID</th>
15                     <th>Product Name</th>
16                     <th>Price</th>
17                     <th>Action</th>
18             </thead>
19             <tbody>
20                 {{#each results}}
21                 <tr>
22                     <td>{{ product_name }}</td>
23                     <td>{{ product_price }}</td>
24                     <td>
25                         <a href="javascript:void(0);" class="btn btn-sm btn-info edit">
26                         <a href="javascript:void(0);" class="btn btn-sm btn-danger del">
27                     </td>
28                 </tr>
29             </tbody>
30             {{/each}}
31         </table>
32     </div>
33
34     <!-- Modal Add Product-->
35     <form action="/save" method="post">
36         <div class="modal fade" id="myModalAdd" tabindex="-1" role="dialog">
37             <div class="modal-dialog" role="document">
38                 <div class="modal-content">
39                     <div class="modal-header">
40                         <h5 class="modal-title" id="exampleModalLabel">Add New Proc
41                         <button type="button" class="close" data-dismiss="modal" ar
42                             <span aria-hidden="true">&times;</span></button>
43                     </div>
44                     <div class="modal-body">
45                         <div class="form-group">
46                             <input type="text" name="product_name" class="form-cont
47                             </div>

```

```

52     <div class="form-group">
53         <input type="text" name="product_price" class="form-cor
54     </div>
55 </div>
56 <div class="modal-footer">
57     <button type="button" class="btn btn-secondary" data-dismiss
58     <button type="submit" class="btn btn-primary">Save</button>
59 </div>
60 </div>
61 </div>
62 </div>
63 </form>
64
65 <!-- Modal Update Product-->
66 <form action="/update" method="post">
67     <div class="modal fade" id="EditModal" tabindex="-1" role="dialog"
68     <div class="modal-dialog" role="document">
69         <div class="modal-content">
70             <div class="modal-header">
71                 <h5 class="modal-title" id="exampleModalLabel">Edit Product<
72                 <button type="button" class="close" data-dismiss="modal" ari
73                     <span aria-hidden="true">&times;</span>
74                 </button>
75             </div>
76             <div class="modal-body">
77                 <div class="form-group">
78                     <input type="text" name="product_name" class="form-contr
79                 </div>
80
81                 <div class="form-group">
82                     <input type="text" name="product_price" class="form-cont
83                 </div>
84             </div>
85             <div class="modal-footer">
86                 <input type="hidden" name="id" class="product_id">
87                 <button type="button" class="btn btn-secondary" data-dismiss
88                 <button type="submit" class="btn btn-primary">Update</butt
89             </div>
90         </div>
91     </div>
92 </div>
93 </form>
94
95 <!-- Modal Delete Product-->
96 <form id="add-row-form" action="/delete" method="post">
97     <div class="modal fade" id="DeleteModal" tabindex="-1" role="dial
98     <div class="modal-dialog">
99         <div class="modal-content">
100             <div class="modal-header">
101                 <h5 class="modal-title" id="myModa
102                 <button type="button" class="close" data-dismiss="mc

```

```

103         </div>
104         <div class="modal-body">
105             <strong>Are you sure to c
106         </div>
107         <div class="modal-footer">
108             <input type="hidden" name="pro
109             <button type="button" class="btn btn-default" data
110             <button type="submit" class="btn btn-success">Dele
111         </div>
112     </div>
113 </div>
114 </div>
115 </form>
116
117 <script src="/assets/js/jquery-3.3.1.js"></script>
118 <script src="/assets/js/bootstrap.js"></script>
119 <script>
120     $(document).ready(function(){
121         //showing delete record modal
122         $('#mytable').on('click', '.de
123         var product_id = $(this).data('id');
124         var product_name = $(this).data('product_name');
125         var product_price = $(this).data('product_price');
126         $('#EditModal').modal('show');
127         $('#product_name').val(product_name);
128         $('#price').val(product_price);
129         $('#product_id').val(product_id);
130     });
131     //showing delete record modal
132     $('#mytable').on('click', '.delete',function(){
133         var product_id = $(this).data('id');
134         $('#DeleteModal').modal('show');
135         $('#product_id2').val(product_id);
136     });
137 });
138 </script>
139 </body>
140 </html>

```

Step #7. Testing

Test the application to make sure the crud application that we create runs

well.

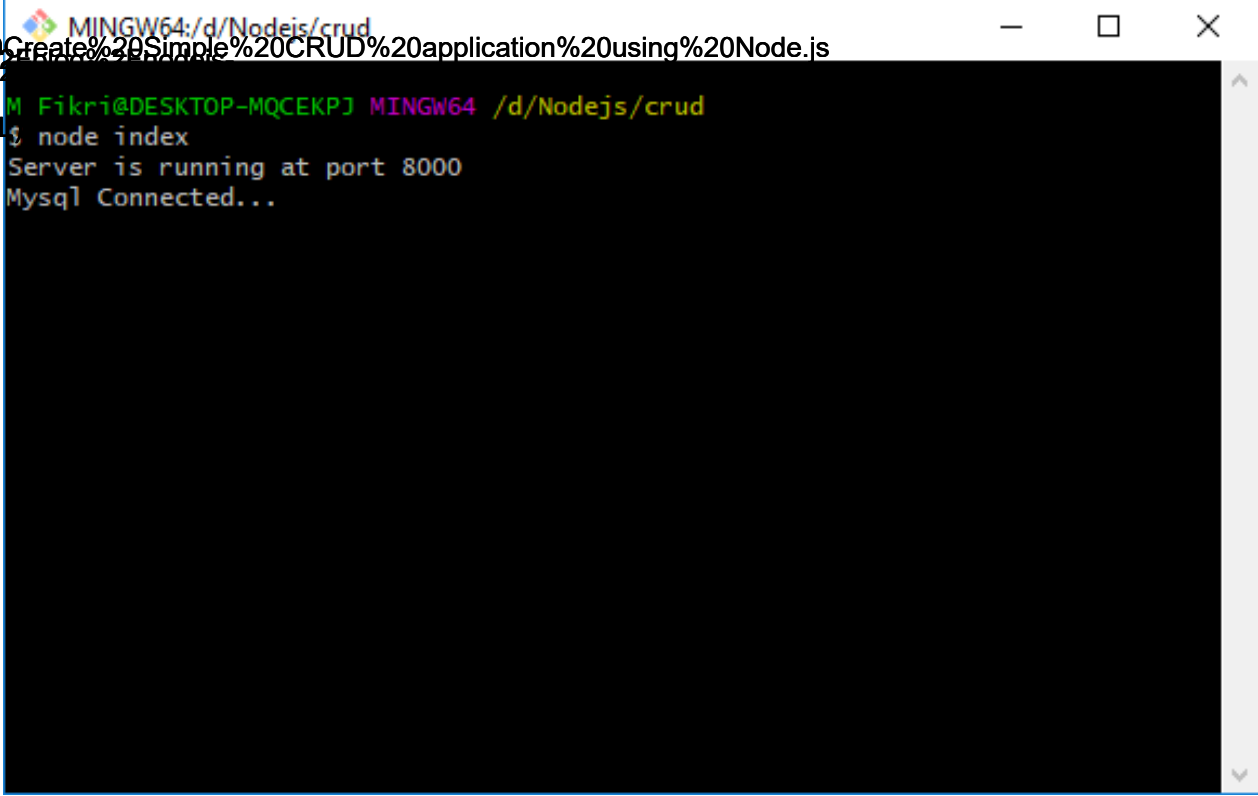
To run the application, type the following command on the terminal.

```
1 | node index
```

Then it will appear on the console message "Server is running at port 8000"

and "Mysql Connected".

Like the picture below.

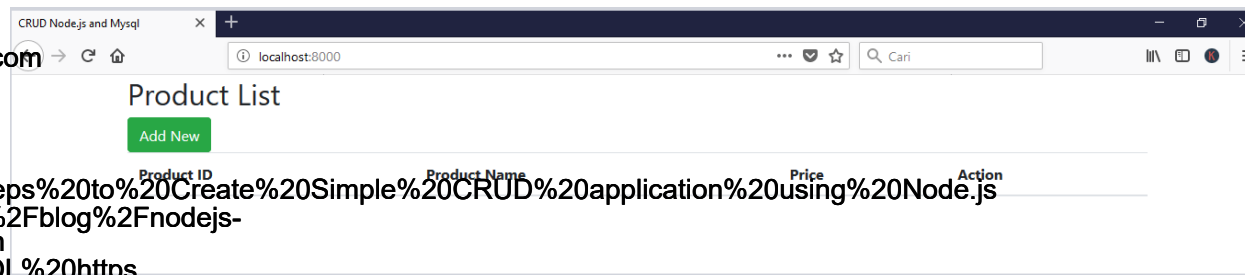


The screenshot shows a terminal window titled 'MINGW64: d/Nodejs/crud'. The prompt is 'M Fikri@DESKTOP-MQCEKPJ MINGW64 /d/Nodejs/crud'. The user has entered the command 'node index'. The output displayed is 'Server is running at port 8000' followed by 'Mysql Connected...' on the next line.

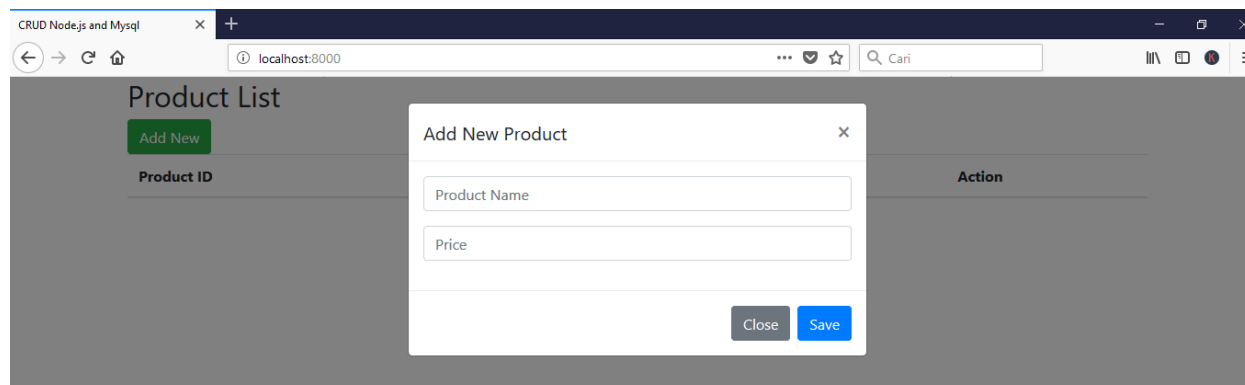
Then open your browser and type the following URL:

http://localhost:8000/ (http://localhost:8000/)

If running well, then it will look like the following picture:

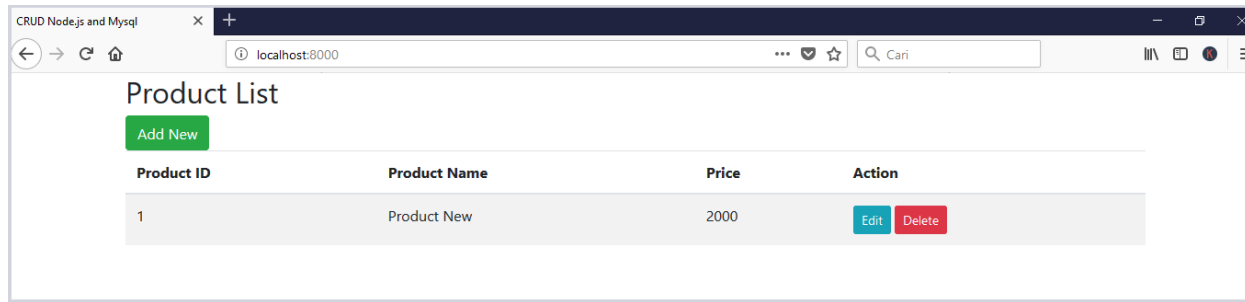


Click the "**Add New**" button, then will appear "**Add New Product**" modal like the picture below:



Then Input Product Name and Price then click **Save** button.

Then you can see the results like the picture below:



www.facebook.com

an?u=https
rter.com

atus=7%20Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js "Edit Product"

L.com%2Fen%2Fblog%2Fnodejs-

ts.google.com

and%20MySQL%20https

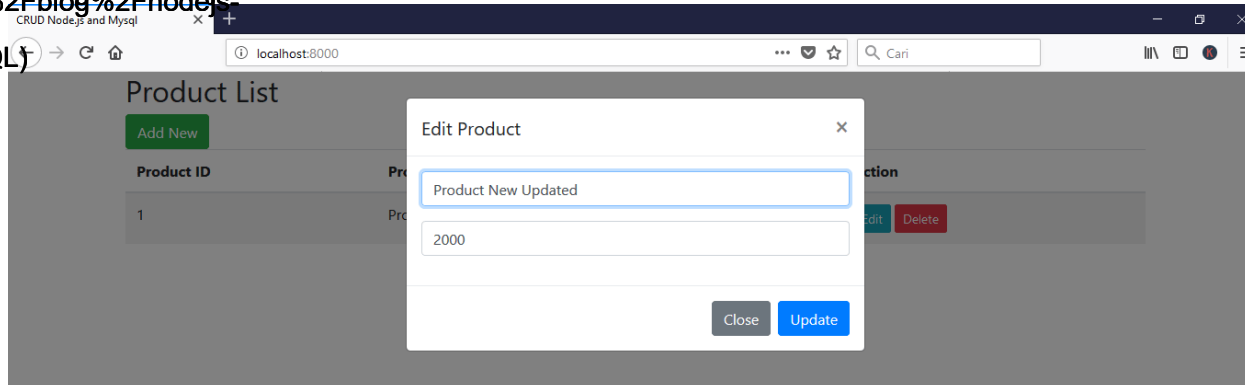
F=https

Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js

L.com%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL)



Edit the data you want to edit, then click the **Update** button.

Then you can see the results like the picture below:



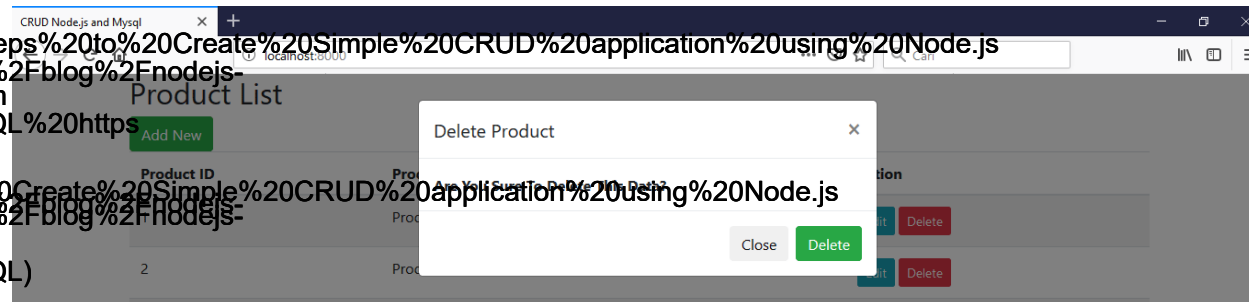
Product List

[Add New](#)

Product ID	Product Name	Price	Action
1	Product New Updated	2000	Edit Delete

To delete a record, click the **Delete** button, then will appear the

confirmation modal as follows:



Then click the **Delete** button to delete the record.

Done.

Conclusion:

Today's tutorial is about how to build a CRUD application (Create Read

Update Delete) using node.js and mysql.

In this tutorial, you have learned how to make CRUD applications with node.js and mysql.

Not only that, you have also learned how to use bootstrap modals for input and update forms.

www.facebook.com

in?u=https
fiter.com

status=7%20Steps%20to%20create%20Simple%20CRUD%20application%20using%20Node.js

com%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL%20https

F=https

Steps%20to%20Create%20Simple%20CRUD%20application%20using%20Node.js


com%2Fen%2Fblog%2Fnodejs-

Express

and%20MySQL)

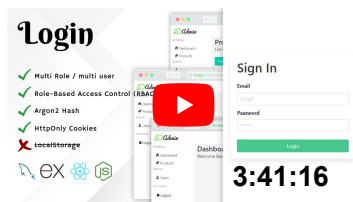
DOWNLOAD SOURCE CODE (HTTPS://PASSIONEAGLE.COM/DL/OQ)

 Twitter  Facebook  Google+  Linked In  Pinterest

 Whats App (whatsapp://send?text=https%3A%2F%2Fmfikri.com%2Fen%2Fblog%2Fnodejs-mysql-crud

Learn%20how%20to%20create%20a%20simple%20CRUD%20application%20using%20node.js%2C%20express%2C%20bootstrap%2C%20and%20mysql%20step%20by%20step...)

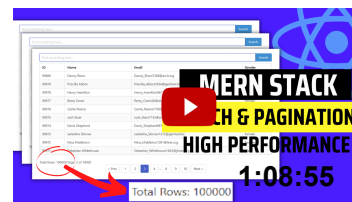
Video Tutorials:



(https://youtu.be/OJU0L9D-



(https://youtu.be



(https://youtu.be