



Express Generator

#Node.js Express Notes

Express Generator



Download Express Generator Package

C:\WINDOWS\system32\cmd.exe

```
D:\>npm i express-generator -g
```



Create new Project

C:\WINDOWS\system32\cmd.exe

```
D:\>express --view=ejs myapp
```



Project will Create various Folders

C:\WINDOWS\system32\cmd.exe

```
D:\>express --view=ejs myapp
```

```
create : myapp\  
create : myapp\public\  
create : myapp\public\javascripts\  
create : myapp\public\images\  
create : myapp\public\stylesheets\  
create : myapp\public\stylesheets\style.css  
create : myapp\routes\  
create : myapp\routes\index.js  
create : myapp\routes\users.js  
create : myapp\views\  
create : myapp\views\error.ejs  
create : myapp\views\index.ejs  
create : myapp\app.js  
create : myapp\package.json  
create : myapp\bin\  
create : myapp\bin\www
```



Navigate to Project

C:\WINDOWS\system32\cmd.exe

```
D:\>cd myapp
```

```
D:\myapp>
```



Download Dependency using npm install

C:\WINDOWS\system32\cmd.exe

```
D:\myapp>npm install
```



Install Successful

C:\WINDOWS\system32\cmd.exe

```
D:\myapp>npm install  
  
added 54 packages, and audited 55 packages in  
  
found 0 vulnerabilities  
  
D:\myapp>
```



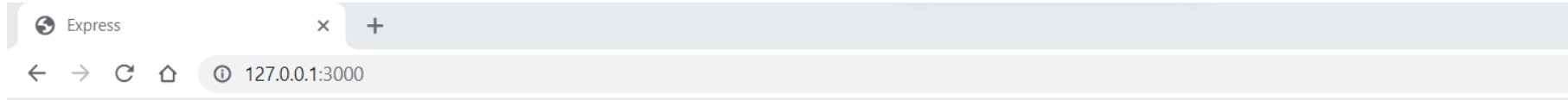
Run Project

Ctrl npm start

```
D:\myapp>npm start  
  
> myapp@0.0.0 start  
> node ./bin/www
```



Output Browser (<http://127.0.0.1:3000>)

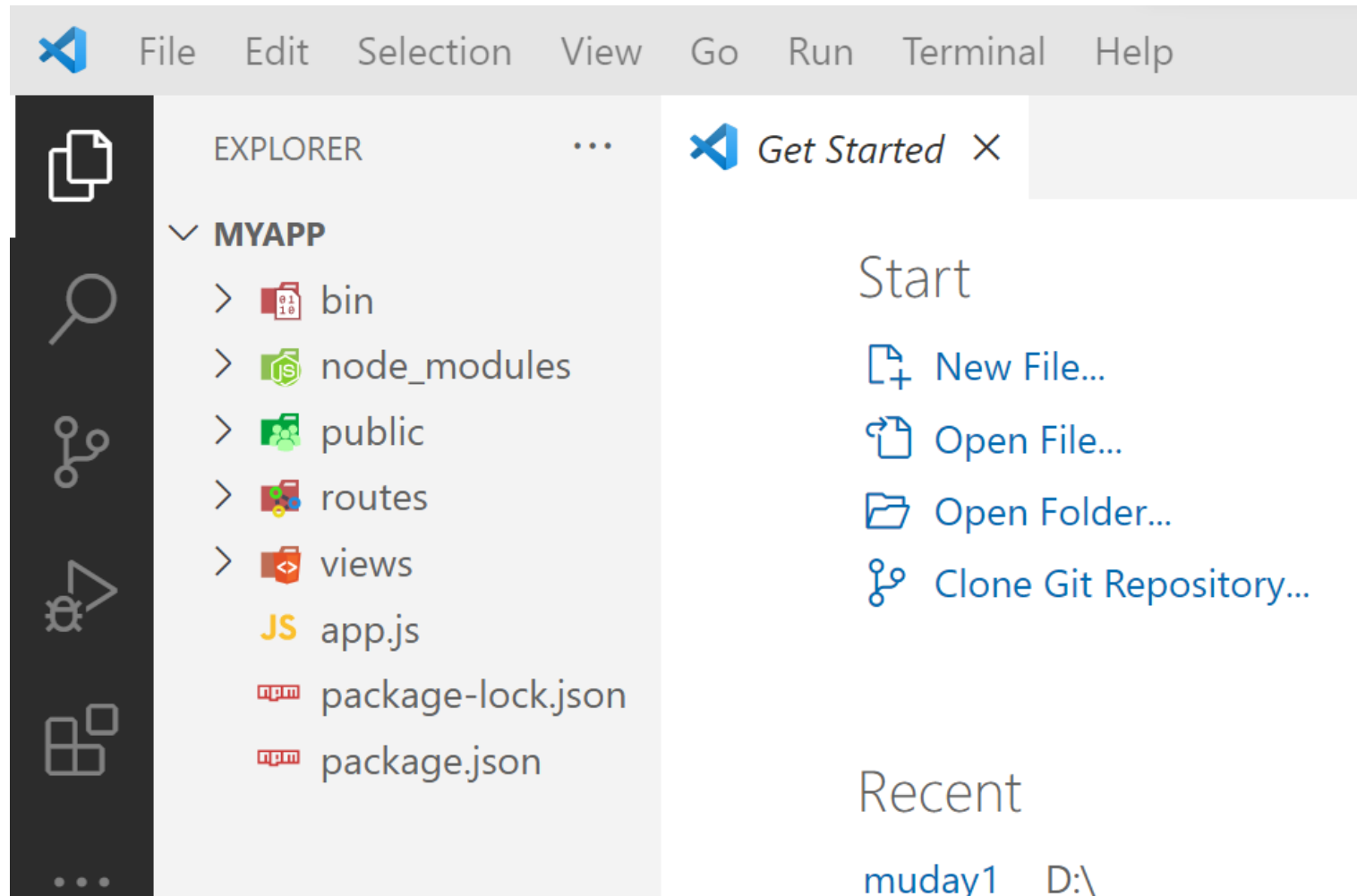


Express

Welcome to Express



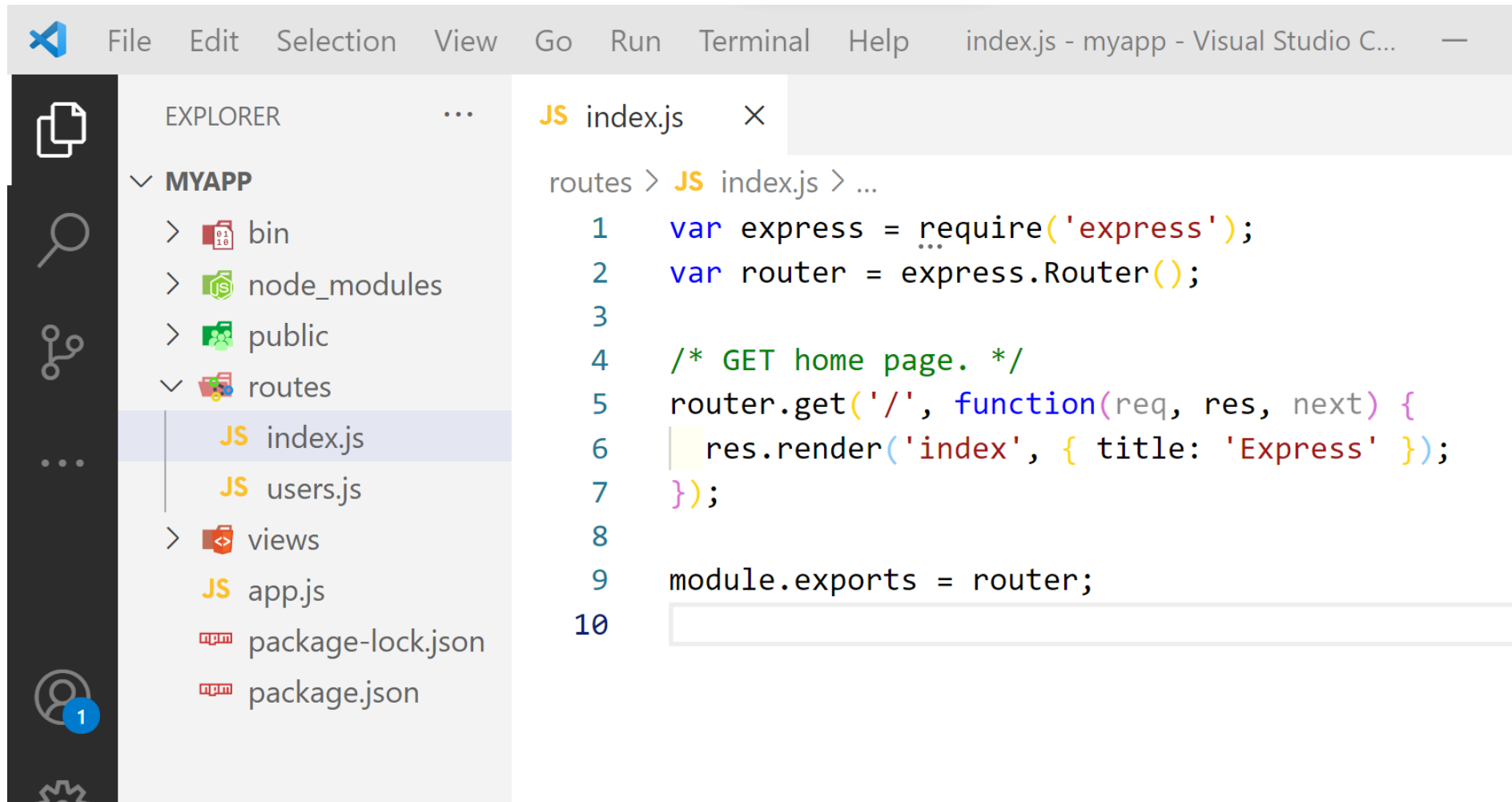
Folder Structure



Navigation (Routing)



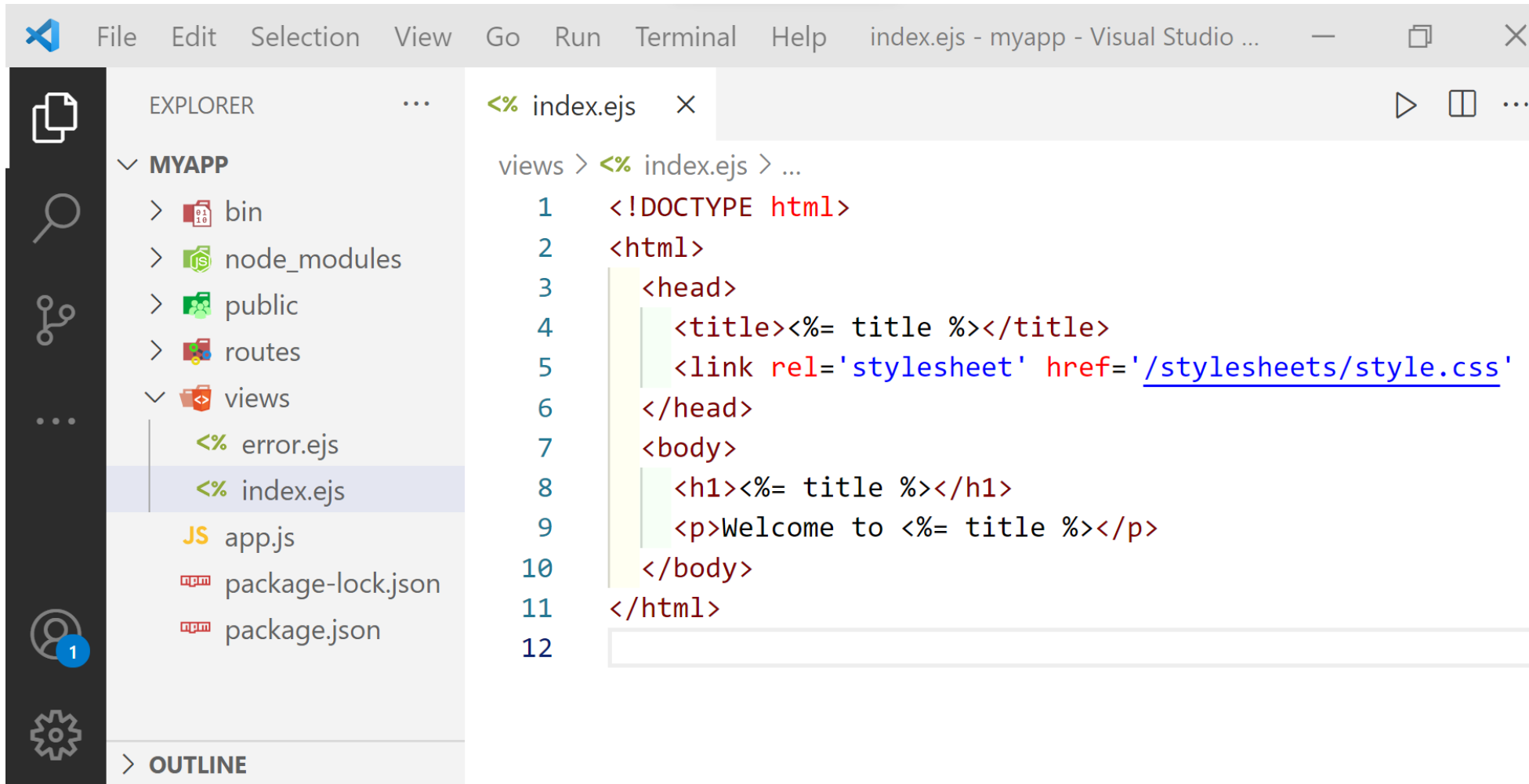
Default Index.js Route File



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows a project named 'MYAPP' with a 'routes' folder containing 'index.js' and 'users.js'. The editor window shows the content of 'index.js'.

```
File Edit Selection View Go Run Terminal Help index.js - myapp - Visual Studio C...  
EXPLORER  
▼ MYAPP  
  > bin  
  > node_modules  
  > public  
  ▼ routes  
    JS index.js  
    JS users.js  
  > views  
    JS app.js  
    package-lock.json  
    package.json  
routes > JS index.js > ...  
1  var express = require('express');  
2  var router = express.Router();  
3  
4  /* GET home page. */  
5  router.get('/', function(req, res, next) {  
6    res.render('index', { title: 'Express' });  
7  });  
8  
9  module.exports = router;  
10
```

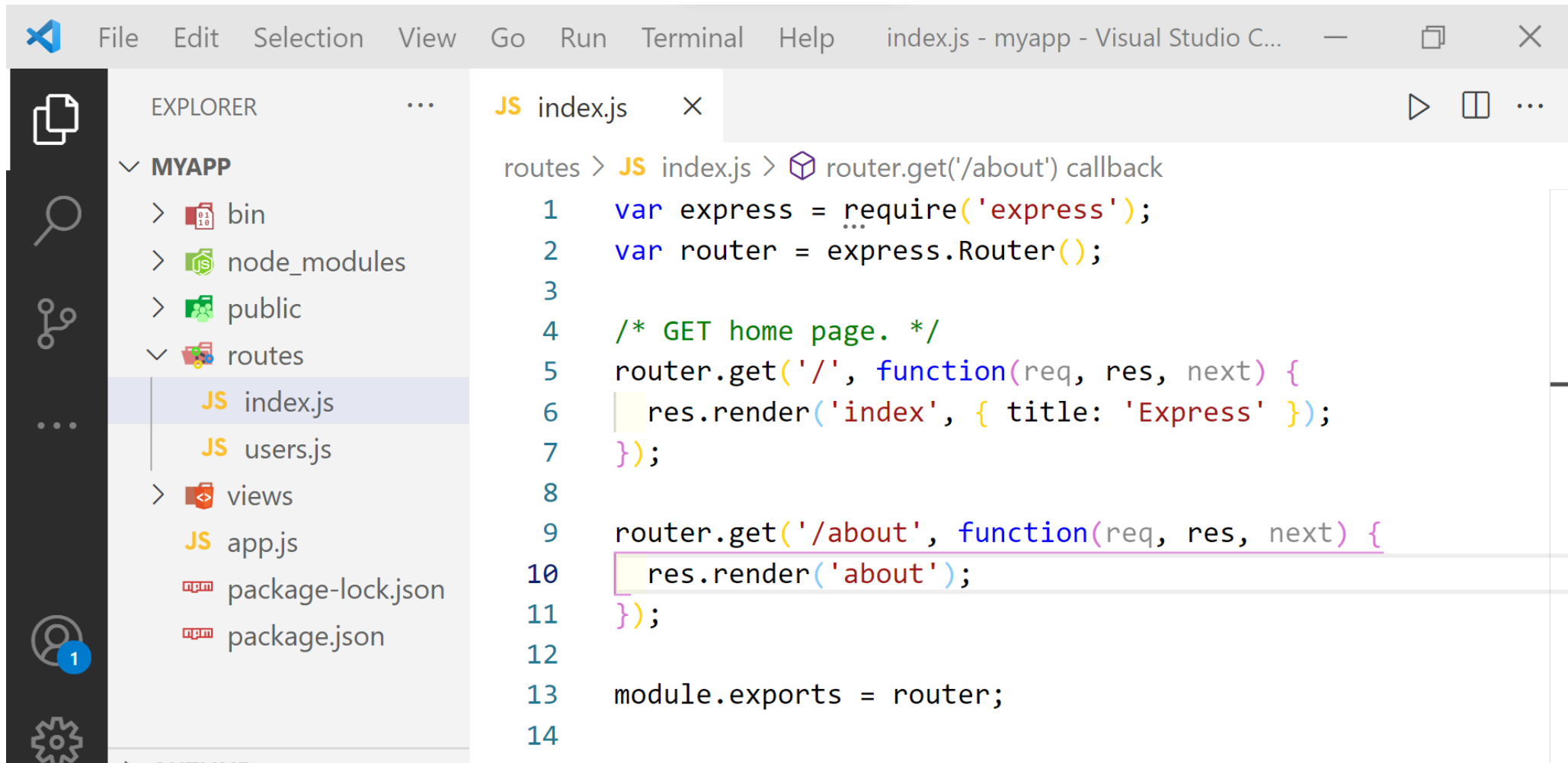
Default Index.ejs file



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure for 'MYAPP', including folders like 'bin', 'node_modules', 'public', 'routes', and 'views'. The 'views' folder is expanded, showing 'error.ejs' and 'index.ejs'. The 'index.ejs' file is selected and its content is displayed in the main editor. The code is an EJS template for a simple HTML page with a title and a welcome message.

```
File Edit Selection View Go Run Terminal Help index.ejs - myapp - Visual Studio ...
<% index.ejs
views > <% index.ejs > ...
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title><%= title %></title>
5     <link rel='stylesheet' href='/stylesheets/style.css'
6   </head>
7   <body>
8     <h1><%= title %></h1>
9     <p>Welcome to <%= title %></p>
10  </body>
11 </html>
12
```

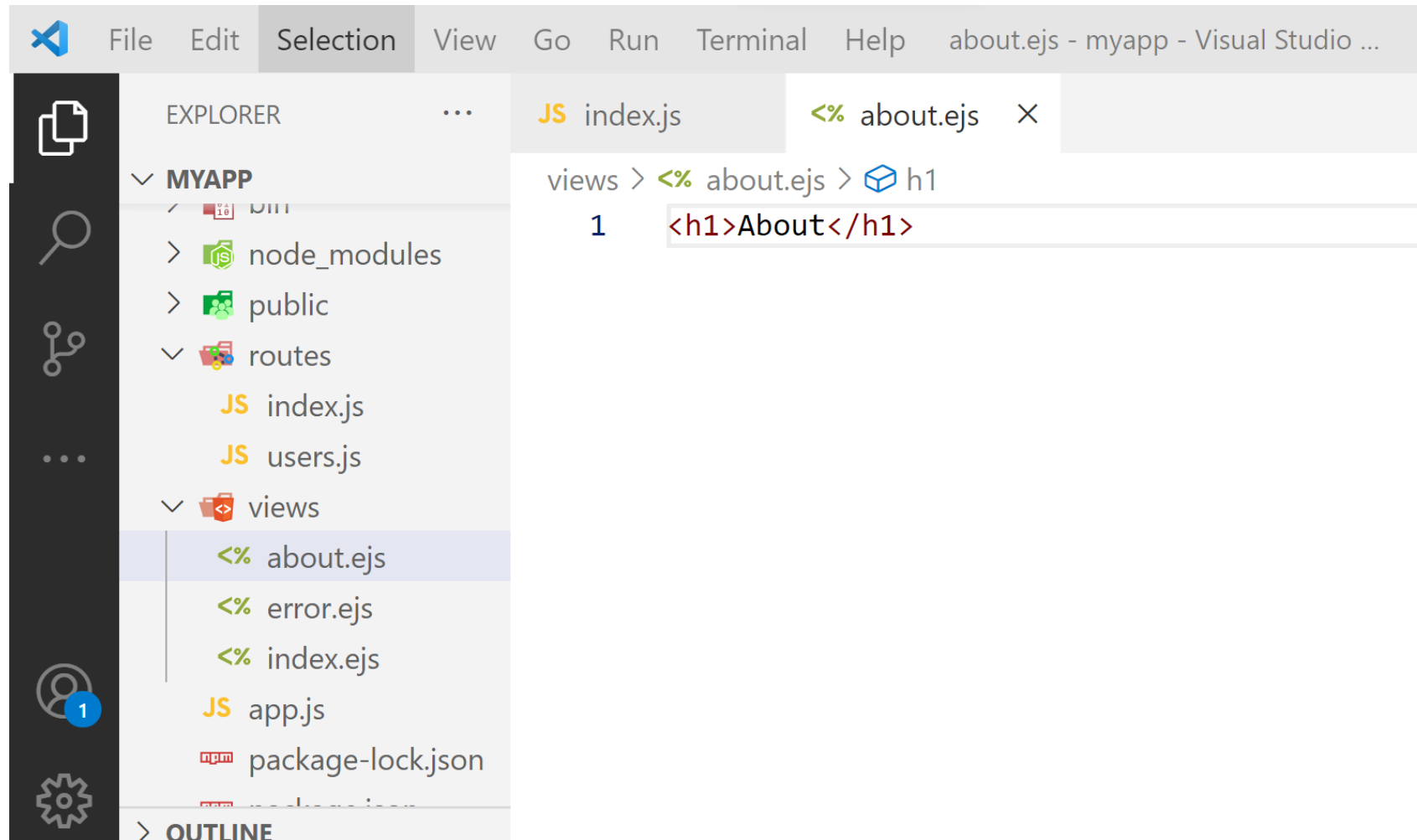
Create New Route named about in (Route->index.js)



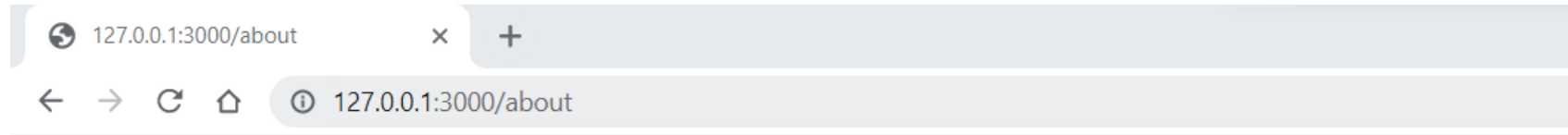
The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure for 'MYAPP'. The 'routes' folder is expanded, showing 'index.js' and 'users.js'. The 'index.js' file is selected. The main editor area shows the content of 'index.js'. The code defines an Express application with two routes: a home page at '/' and an 'about' route at '/about'. The 'about' route is currently empty, and the cursor is positioned to add a new route handler. The status bar at the bottom indicates the file is 'index.js' in the 'myapp' project.

```
routes > JS index.js > router.get('/about') callback
1  var express = require('express');
2  var router = express.Router();
3
4  /* GET home page. */
5  router.get('/', function(req, res, next) {
6    res.render('index', { title: 'Express' });
7  });
8
9  router.get('/about', function(req, res, next) {
10   res.render('about');
11 });
12
13 module.exports = router;
14
```

Create new **about.ejs** file in views



Open [http:127.0.0.1:3000/about](http://127.0.0.1:3000/about)



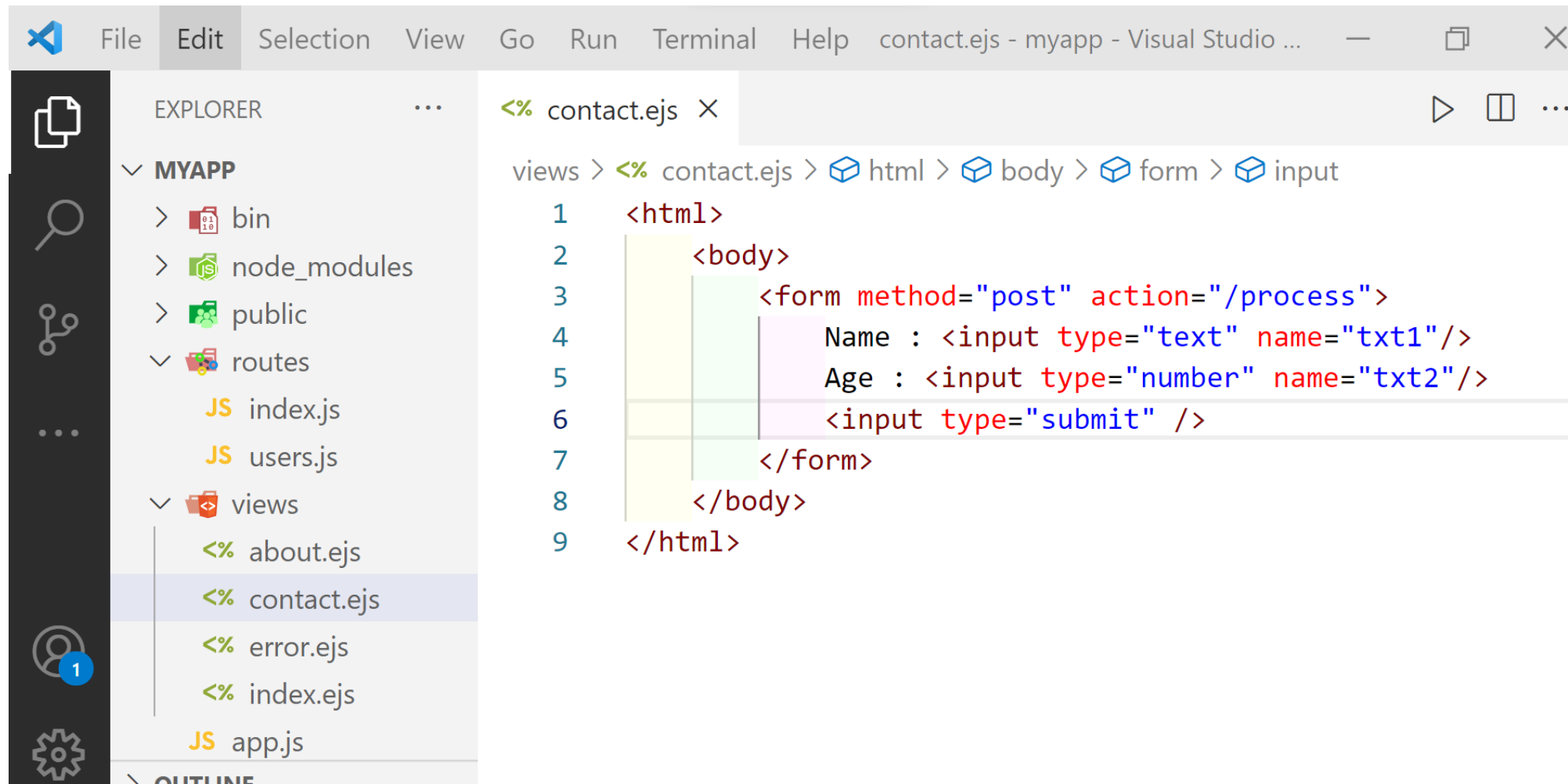
About



Get Post



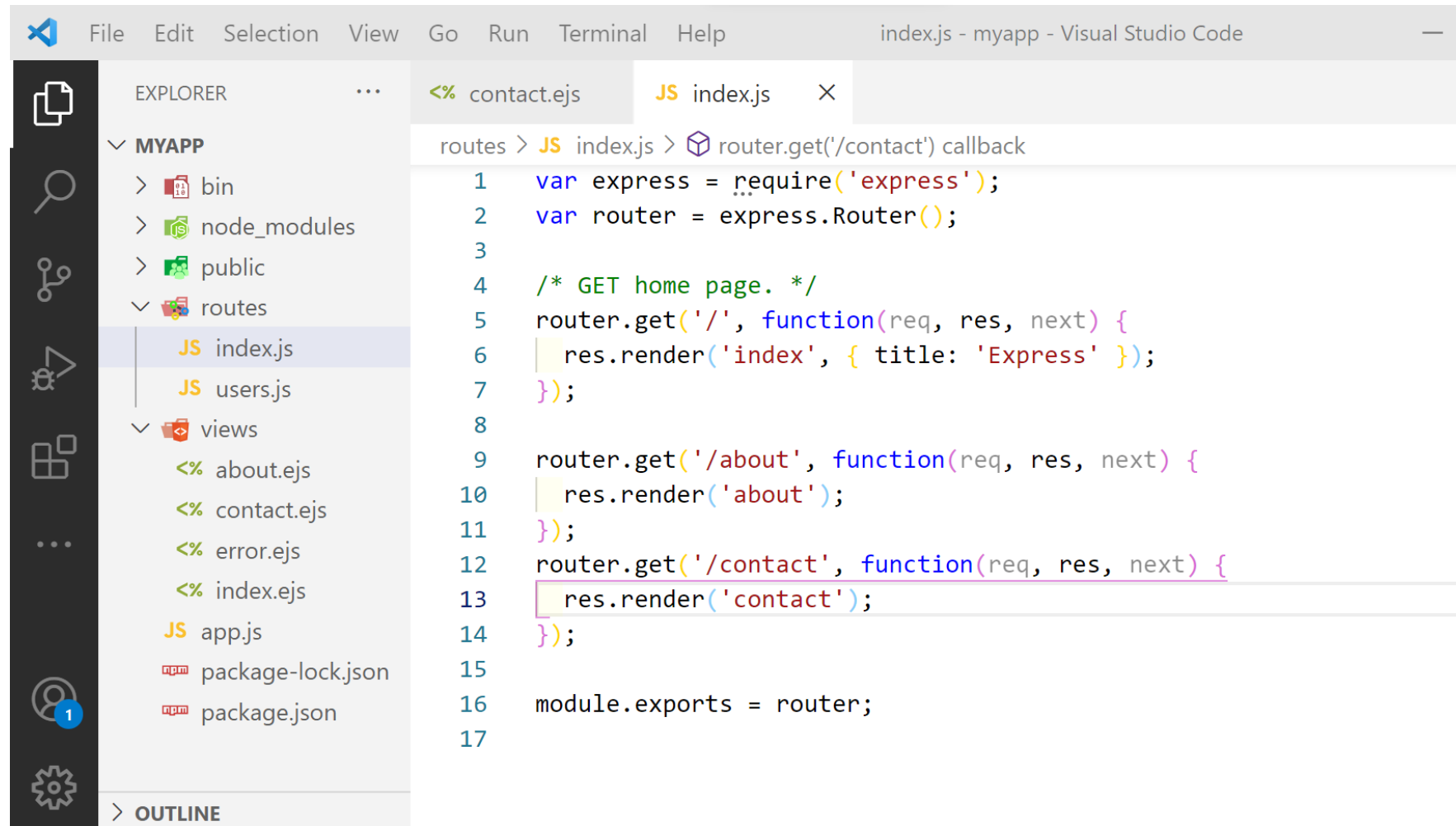
Create contact.ejs file in Index.js Route



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure for 'MYAPP', including folders like 'bin', 'node_modules', 'public', 'routes', and 'views'. The 'routes' folder is expanded, showing 'index.js' and 'users.js'. The 'views' folder is also expanded, showing 'about.ejs', 'contact.ejs', 'error.ejs', and 'index.ejs'. The 'contact.ejs' file is selected and highlighted. The Editor window shows the content of 'contact.ejs', which is an HTML template with a form for processing. The breadcrumb navigation at the top of the Editor window indicates the file path: 'views > contact.ejs > html > body > form > input'.

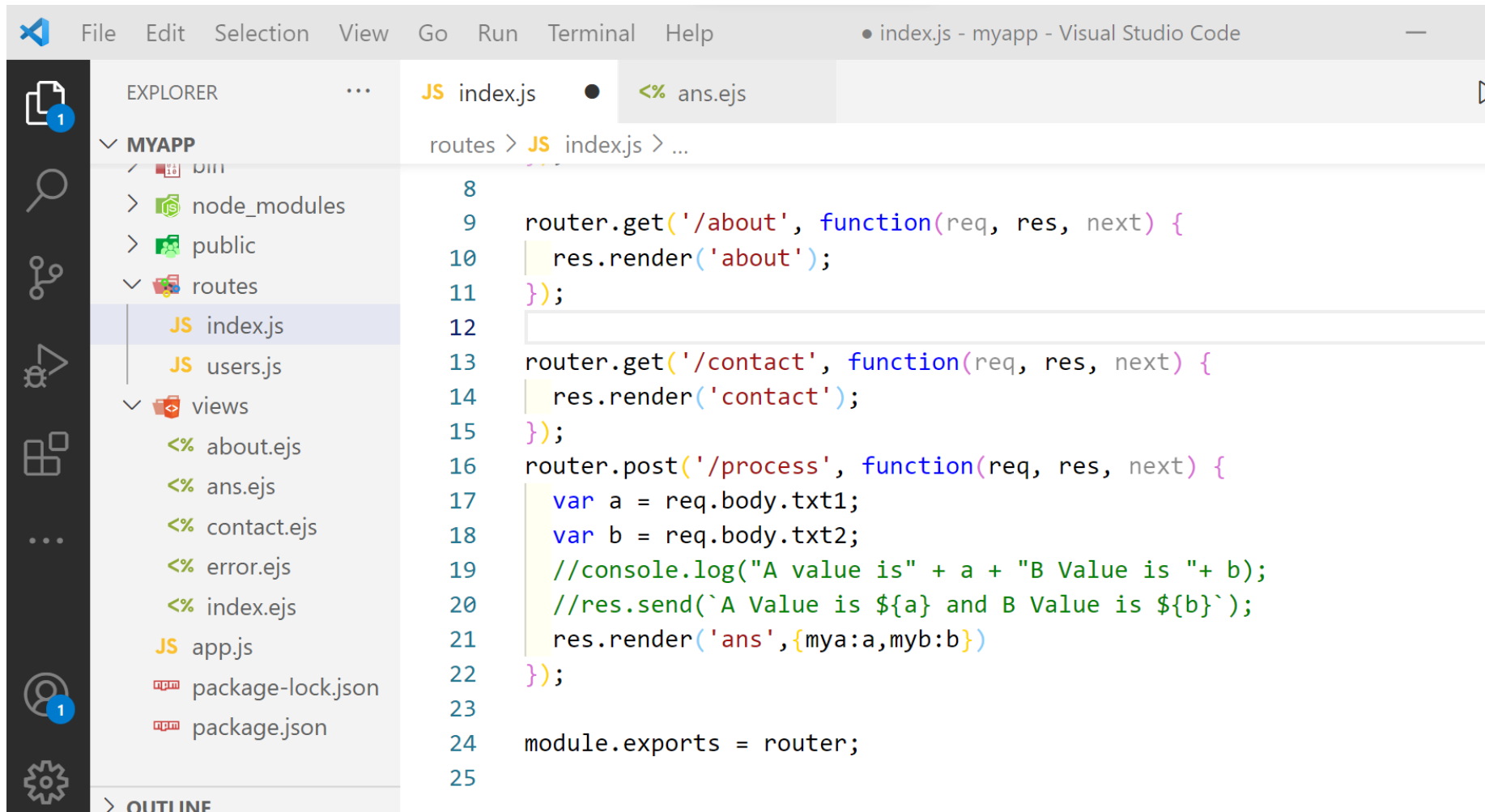
```
<% contact.ejs X
views > <% contact.ejs > html > body > form > input
1  <html>
2      <body>
3          <form method="post" action="/process">
4              Name : <input type="text" name="txt1"/>
5              Age : <input type="number" name="txt2"/>
6              <input type="submit" />
7          </form>
8      </body>
9  </html>
```

Create new Route called contact



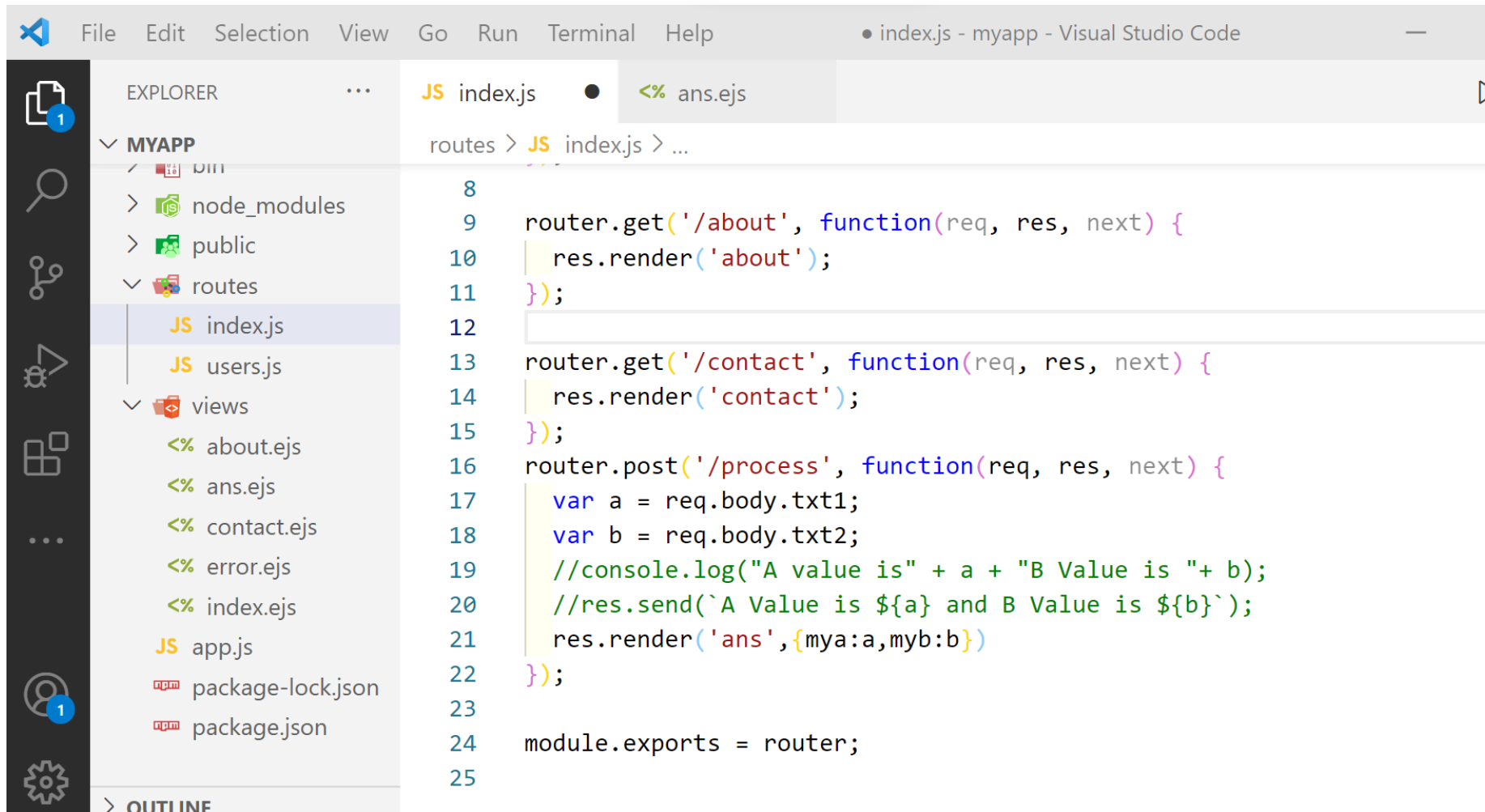
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Code Editor on the right. The Explorer sidebar shows the project structure for 'MYAPP', including folders like 'bin', 'node_modules', 'public', 'routes', and 'views', and files like 'index.js', 'users.js', 'about.ejs', 'contact.ejs', 'error.ejs', 'index.ejs', 'app.js', 'package-lock.json', and 'package.json'. The 'routes' folder is expanded, showing 'index.js' and 'users.js'. The 'contact.ejs' file is highlighted. The Code Editor shows the 'index.js' file with the following code:

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6   res.render('index', { title: 'Express' });
7 });
8
9 router.get('/about', function(req, res, next) {
10   res.render('about');
11 });
12 router.get('/contact', function(req, res, next) {
13   res.render('contact');
14 });
15
16 module.exports = router;
17
```



```
8
9  router.get('/about', function(req, res, next) {
10    res.render('about');
11  });
12
13 router.get('/contact', function(req, res, next) {
14   res.render('contact');
15 });
16 router.post('/process', function(req, res, next) {
17   var a = req.body.txt1;
18   var b = req.body.txt2;
19   //console.log("A value is" + a + "B Value is " + b);
20   //res.send(`A Value is ${a} and B Value is ${b}`);
21   res.render('ans', {mya:a, myb:b});
22 });
23
24 module.exports = router;
25
```

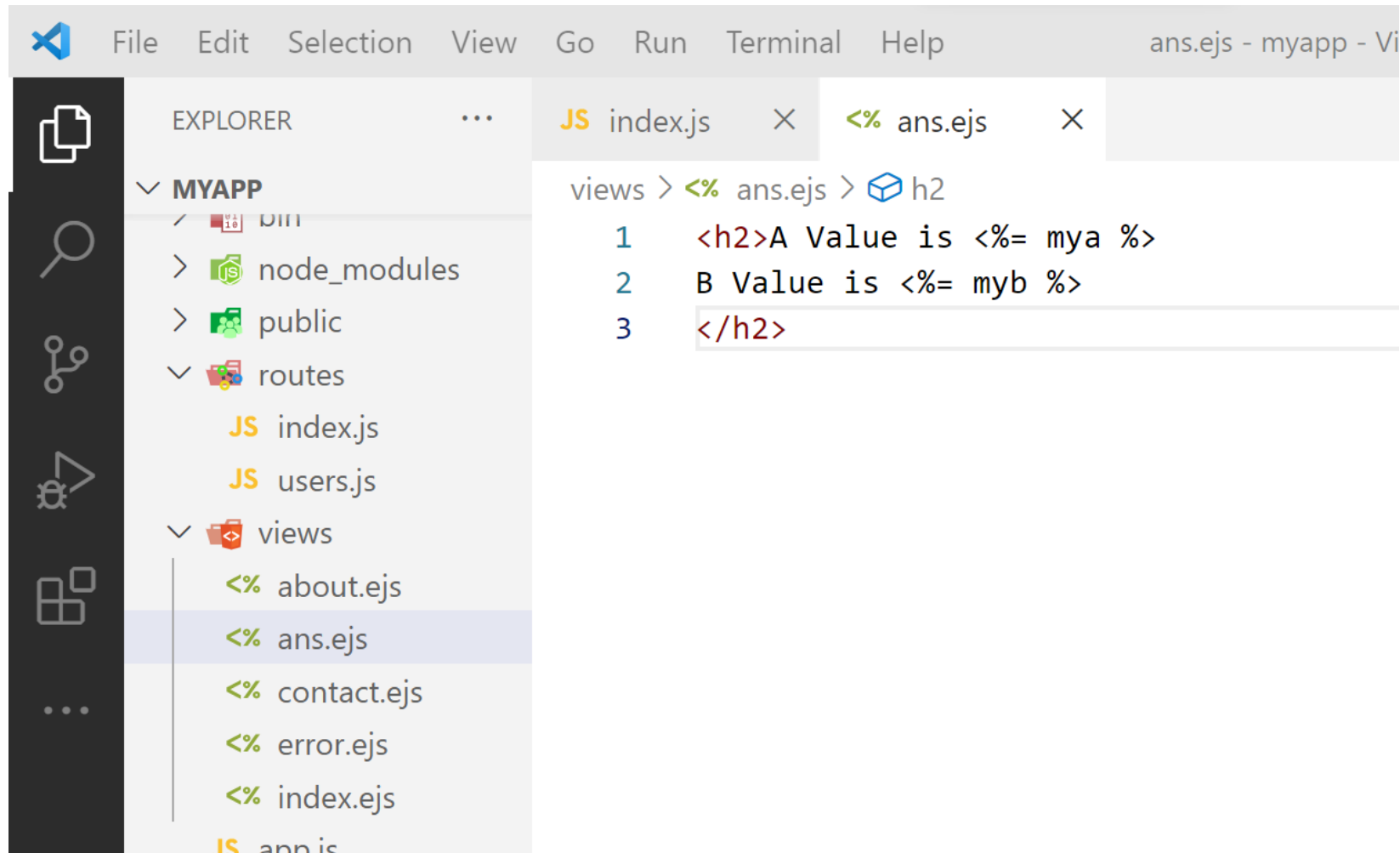
Create Process Route



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure for 'MYAPP'. The file 'index.js' is selected under the 'routes' directory. The main editor window shows the code for 'index.js'.

```
8
9  router.get('/about', function(req, res, next) {
10    res.render('about');
11  });
12
13 router.get('/contact', function(req, res, next) {
14   res.render('contact');
15 });
16 router.post('/process', function(req, res, next) {
17   var a = req.body.txt1;
18   var b = req.body.txt2;
19   //console.log("A value is" + a + "B Value is "+ b);
20   //res.send(`A Value is ${a} and B Value is ${b}`);
21   res.render('ans',{mya:a,myb:b})
22 });
23
24 module.exports = router;
25
```

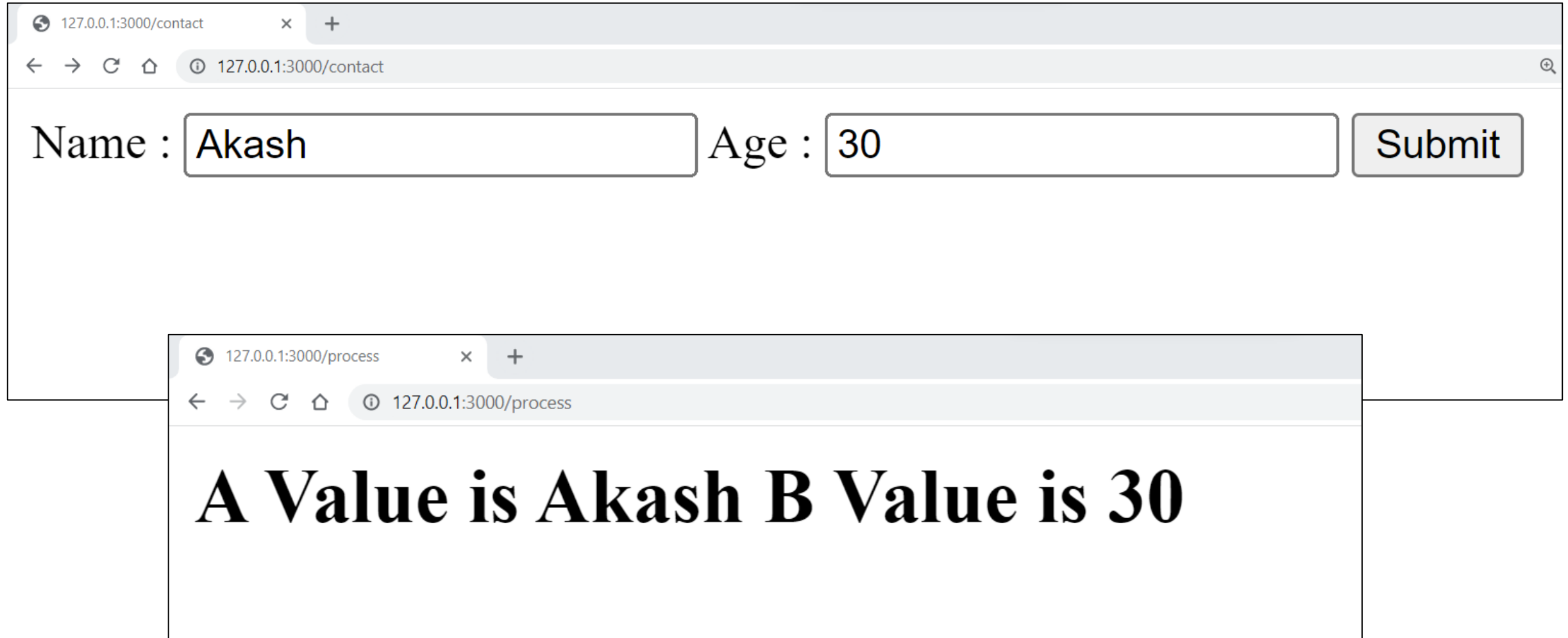
Create Ans.ejs file



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project named 'MYAPP' with a file tree including 'node_modules', 'public', 'routes', and 'views'. The 'views' directory is expanded, showing several files: 'about.ejs', 'ans.ejs' (highlighted), 'contact.ejs', 'error.ejs', 'index.ejs', and 'app.js'. The main editor area has two tabs: 'index.js' and 'ans.ejs'. The 'ans.ejs' tab is active, showing the following code:

```
views > <% ans.ejs > h2
1 <h2>A Value is <%= mya %>
2 B Value is <%= myb %>
3 </h2>
```

Output [http:127.0.0.1:3000/contact](http://127.0.0.1:3000/contact)



The image shows two browser windows. The top window, titled '127.0.0.1:3000/contact', displays a form with 'Name : Akash' and 'Age : 30', followed by a 'Submit' button. The bottom window, titled '127.0.0.1:3000/process', displays the output: 'A Value is Akash B Value is 30'.

127.0.0.1:3000/contact

Name : Akash Age : 30 Submit

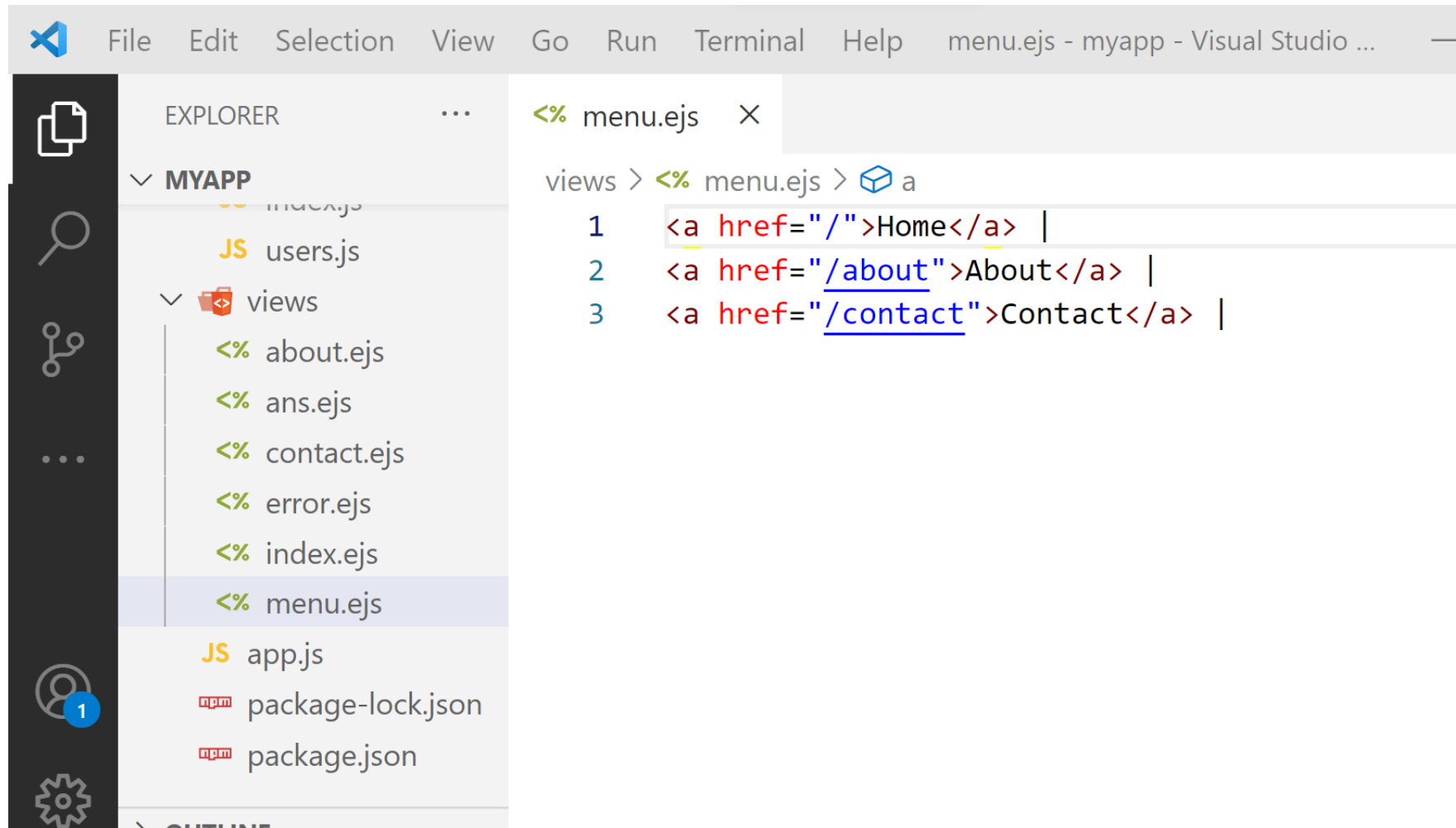
127.0.0.1:3000/process

A Value is Akash B Value is 30

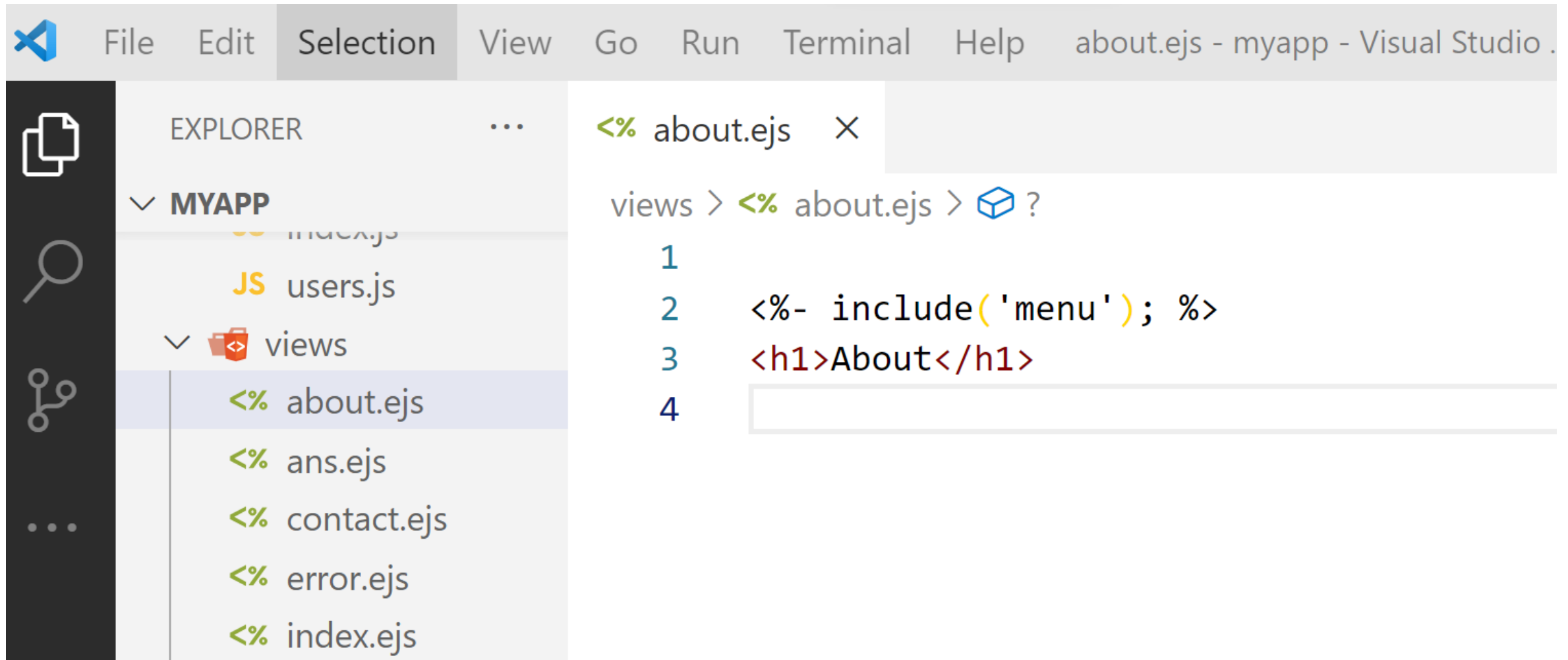
| Common Menu (include)



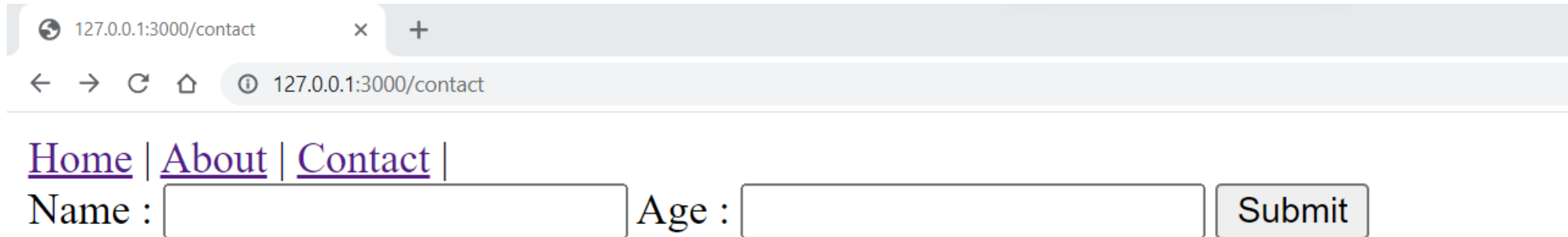
Create new file name menu.ejs



Add menu in every ejs file



Menu Output



A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:3000/contact". The page content includes a navigation menu with three links: "Home", "About", and "Contact", each underlined and separated by a vertical bar. Below the menu is a form with two input fields labeled "Name :" and "Age :", followed by a "Submit" button.

[Home](#) | [About](#) | [Contact](#) |

Name : Age :



| Mysql



mysql DT

2.18.1 • Public • Published 2 years ago

Readme

Explore BETA

4 Dependencies

6,588 Dependents

65 Versions

mysql

npm v2.18.1

downloads 3.74M/month

node >= 0.6

travis passing

windows success

coverage 97%

Table of Contents

- Install
- Introduction
- Contributors
- Sponsors
- Community
- Establishing connections
- Connection options
 - SSL options


Install

```
> npm i mysql
```

Repository

 github.com/mysqljs/mysql

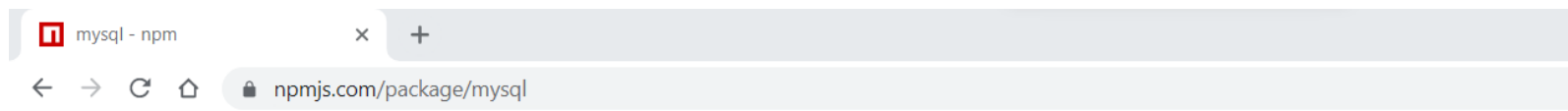
Homepage

 github.com/mysqljs/mysql#readme

Weekly Downloads



Version	License
2.18.1	MIT



Here is an example on how to use it:

```
var mysql      = require('mysql');
var connection = mysql.createConnection({
  host      : 'localhost',
  user      : 'me',
  password  : 'secret',
  database  : 'my_db'
});

connection.connect();

connection.query('SELECT 1 + 1 AS solution', function (error, results, fields) {
  if (error) throw error;
  console.log('The solution is: ', results[0].solution);
});

connection.end();
```



Create Database and Table in phpmyadmin

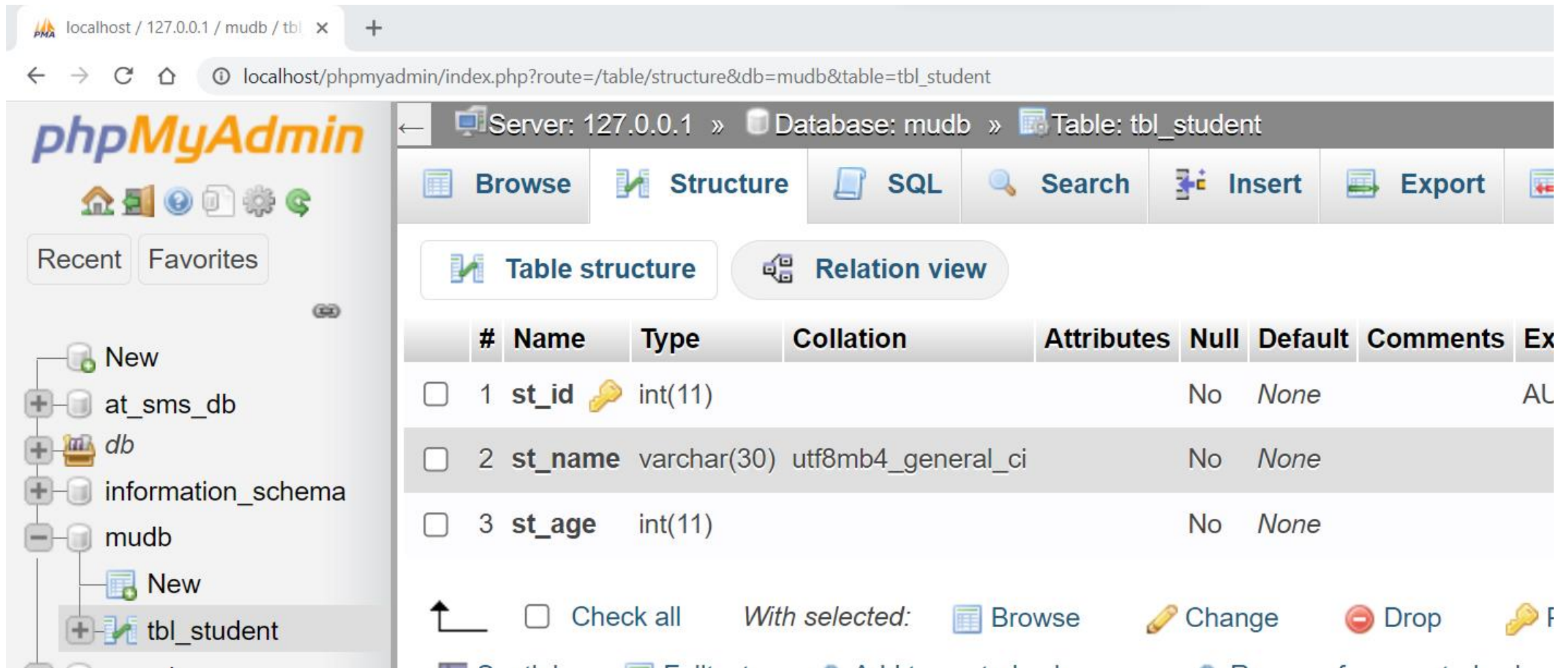
The screenshot shows the phpMyAdmin web interface in a browser. The address bar indicates the URL: `localhost/phpmyadmin/index.php?route=/database/structure&server=1&db=mudb`. The interface is for the 'mudb' database on server '127.0.0.1'. The left sidebar shows a tree view of databases, with 'mudb' selected and its tables listed: 'tbl_student' and 'tbl_sms_db'. The main panel shows the 'Structure' tab for the 'tbl_student' table. It displays a table with 1 row and 1 column. The table name is 'tbl_student' and the action is 'Sum'. The table is using the InnoDB engine and the utf8mb4_general_ci collation. The table structure is as follows:

Table	Action	Rows	Type	Collation
tbl_student	Sum	1	InnoDB	utf8mb4_general_ci

Below the table structure, there are options to 'Check all' and 'With selected:'. At the bottom, there are links for 'Print' and 'Data dictionary'.



Tbl_student in phpmyadmin



The screenshot shows the phpMyAdmin interface in a web browser. The address bar indicates the URL: localhost/phpmyadmin/index.php?route=/table/structure&db=mudb&table=tbl_student. The interface includes a sidebar with a database tree on the left and a main content area on the right. The main content area displays the table structure for 'tbl_student' in the 'mudb' database. The table has three columns: 'st_id' (int(11), primary key), 'st_name' (varchar(30), utf8mb4_general_ci), and 'st_age' (int(11)). The 'Table structure' tab is selected, and the 'Relation view' tab is also visible. Below the table structure, there are options to 'Check all', 'Browse', 'Change', 'Drop', and 'Refresh'.

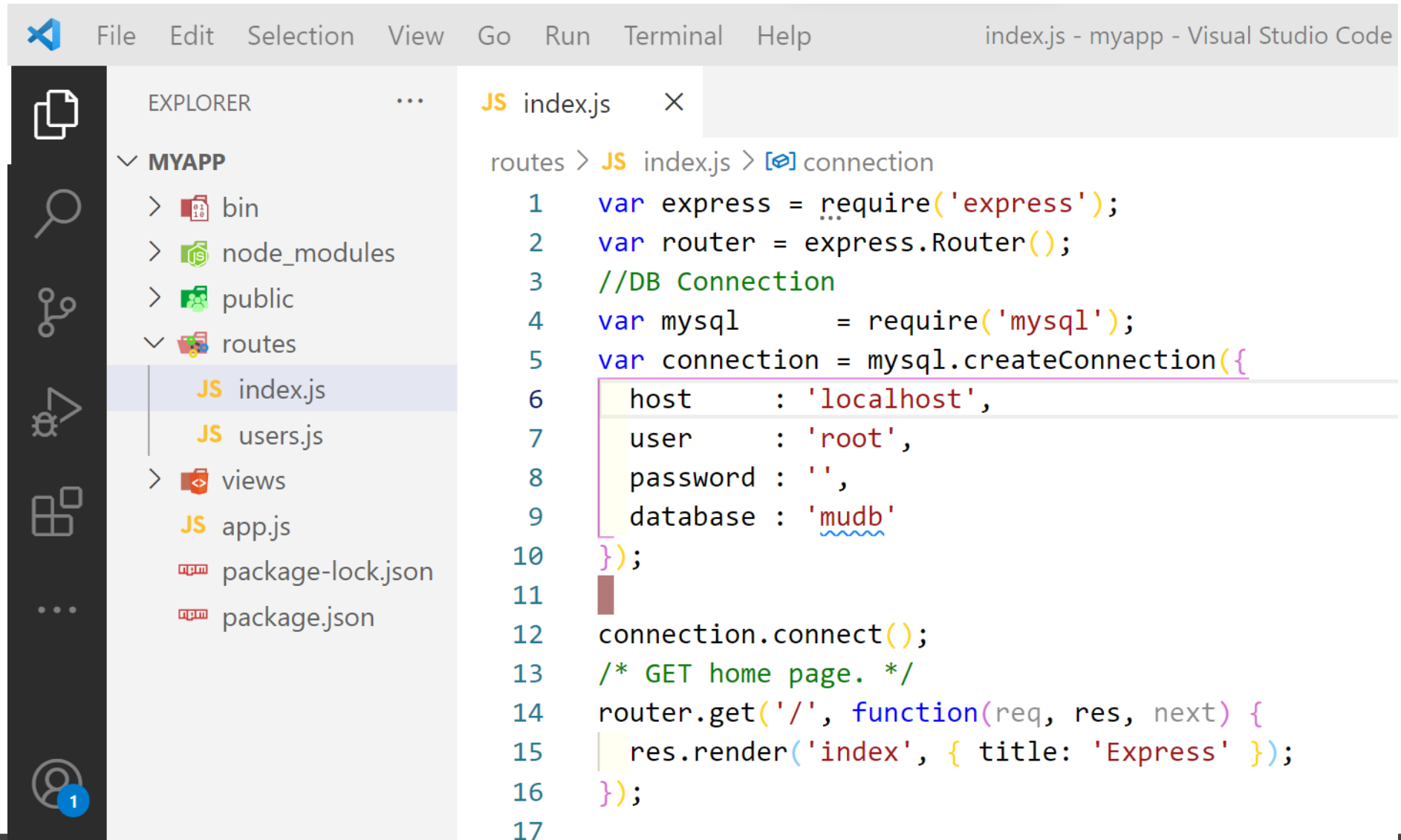
Server: 127.0.0.1 » Database: mudb » Table: tbl_student

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Ex
<input type="checkbox"/>	1 st_id	int(11)			No	None		AL
<input type="checkbox"/>	2 st_name	varchar(30)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	3 st_age	int(11)			No	None		

Check all With selected: Browse Change Drop Refresh

DB Connection



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project named 'MYAPP' with the following structure:

- bin
- node_modules
- public
- routes
 - index.js (selected)
 - users.js
- views
- app.js
- package-lock.json
- package.json

The main editor window shows the 'index.js' file with the following code:

```
routes > JS index.js > [🔗] connection
1  var express = require('express');
2  var router = express.Router();
3  //DB Connection
4  var mysql    = require('mysql');
5  var connection = mysql.createConnection({
6      host      : 'localhost',
7      user      : 'root',
8      password  : '',
9      database  : 'mudb'
10 });
11
12 connection.connect();
13 /* GET home page. */
14 router.get('/', function(req, res, next) {
15     res.render('index', { title: 'Express' });
16 });
17
```

Query

JS index.js

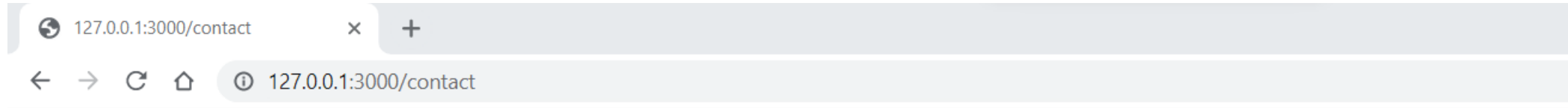


routes > JS index.js > router.post('/process') callback

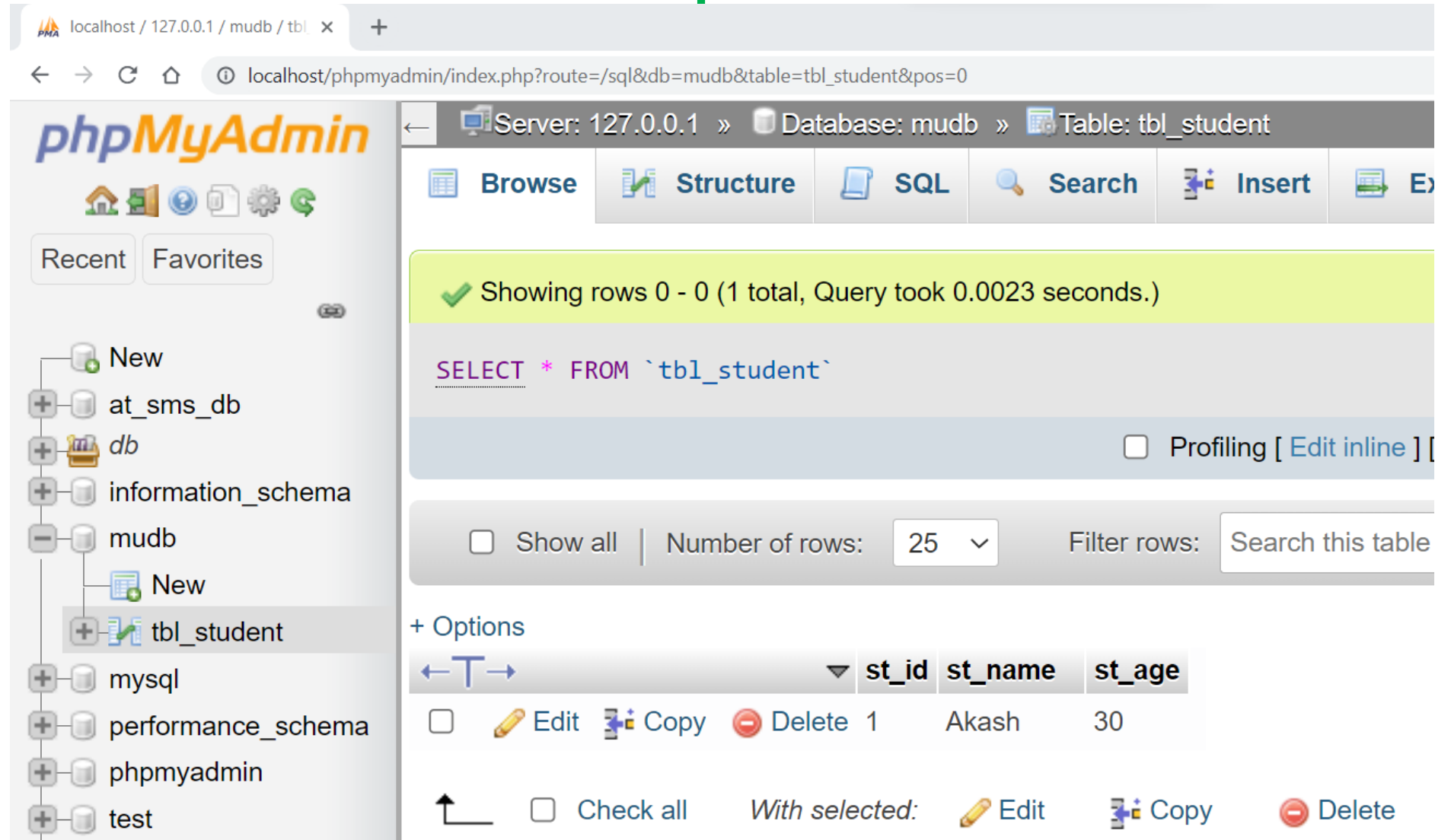
```
21
22 router.get('/contact', function(req, res, next) {
23   res.render('contact');
24 });
25
26 router.post('/process', function(req, res, next) {
27   var a = req.body.txt1;
28   var b = req.body.txt2;
29
30   //Query
31   connection.query("insert into tbl_student (st_name,st_age) values(?,?)",[a,b],function(err){
32     res.send("Record Added");
33   });
34
35 });
36
37 module.exports = router;
38
```



Run Code



Output in Table



The screenshot shows the phpMyAdmin interface for a local database. The left sidebar displays a tree view of databases, with 'tbl_student' selected under the 'mudb' database. The main panel shows the 'Table: tbl_student' view. A green message bar indicates 'Showing rows 0 - 0 (1 total, Query took 0.0023 seconds.)'. Below this, the SQL query 'SELECT * FROM `tbl_student`' is displayed. The 'Filter rows' section shows 'Show all' selected, 'Number of rows' set to 25, and a search box. The '+ Options' section shows a table with columns 'st_id', 'st_name', and 'st_age'. The table contains one row with values 1, Akash, and 30. Below the table, there are checkboxes for 'Check all' and 'With selected:', and buttons for 'Edit', 'Copy', and 'Delete'.

localhost / 127.0.0.1 / mudb / tbl x +

localhost/phpmyadmin/index.php?route=/sql&db=mudb&table=tbl_student&pos=0

phpMyAdmin

Recent Favorites

Server: 127.0.0.1 » Database: mudb » Table: tbl_student

Browse Structure SQL Search Insert Ex

Showing rows 0 - 0 (1 total, Query took 0.0023 seconds.)

SELECT * FROM `tbl_student`

Profiling [Edit inline]

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

	st_id	st_name	st_age
<input type="checkbox"/> Edit Copy Delete	1	Akash	30

Check all With selected: Edit Copy Delete



Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



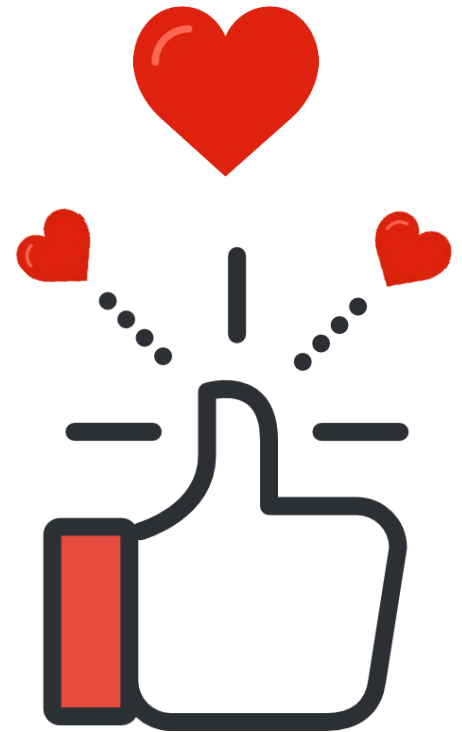
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com

www.akashtechlabs.com

www.akashpadhiyar.com

www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar

#apptutorials