Node.js and Express.js

Lab 09 - Solution

In this lab tutorial, we will step by step show you how to create a simple web application for a restaurant. We already created the front-end in the previous labs. We will follow MVC model and use Node.js and Express.js to create the backed. First, follow these instructions to get the application running:

Download the Zip file from Moodle and unzip it. In the Lab9 folder, there is a SQL script (lab9.sql) which must be loaded and executed in MySQL to prepare the lab9 database. We need this database for our application.

Open a command window and type this command:

```
npm start
```

Go to a browser and type ***localhost:3000***

There are two users in the database at the moment:

Email: admin@gmail.com	Password: test	Role: admin
Email: james@gmail.com	Password: test1234	Role: customer

You can sign up as a new customer.

If you want, to create an application like this, follow these steps:

1- Preparing MySQL Database Structure
2- Install Node Express Application
3- Make MySQL Database Connection with Node Express Application
4- Configure Session
5- Create Route
6- Create Views
7- Start Node Server & Check Output in browser

Step 1- Preparing MySQL Database Structure

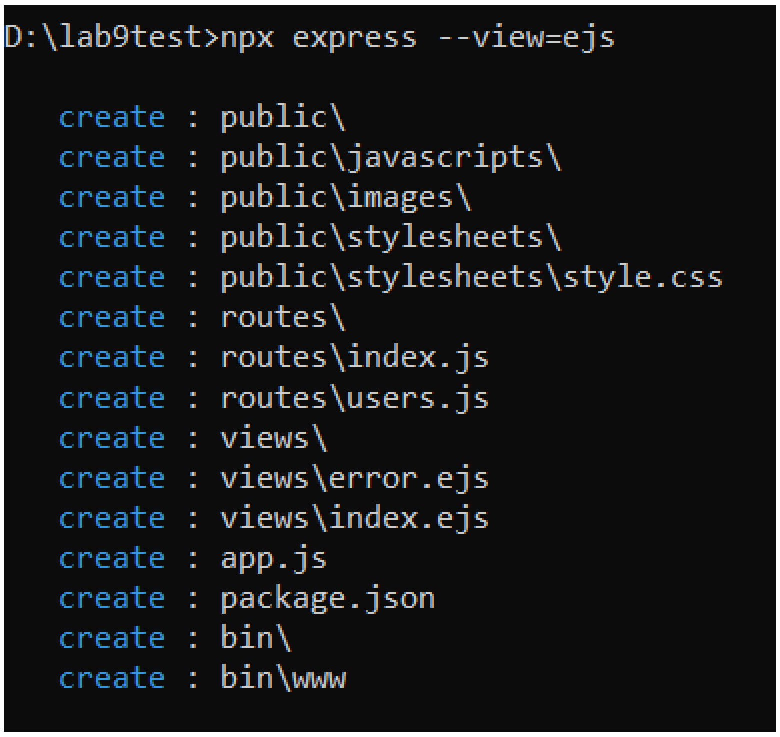
Use a DBMS like MySQL and design a database that can store your data (similar to the one I prepared for this lab).

Step 2- Install Node Express Application

User the command line, create a folder and run the following commands:

```
npm install -g express-generator
npx express --view=ejs
```

This command prepares directory structure for your project and help you to follow the MVC approach. A sample of the directory structure is shown in the below.



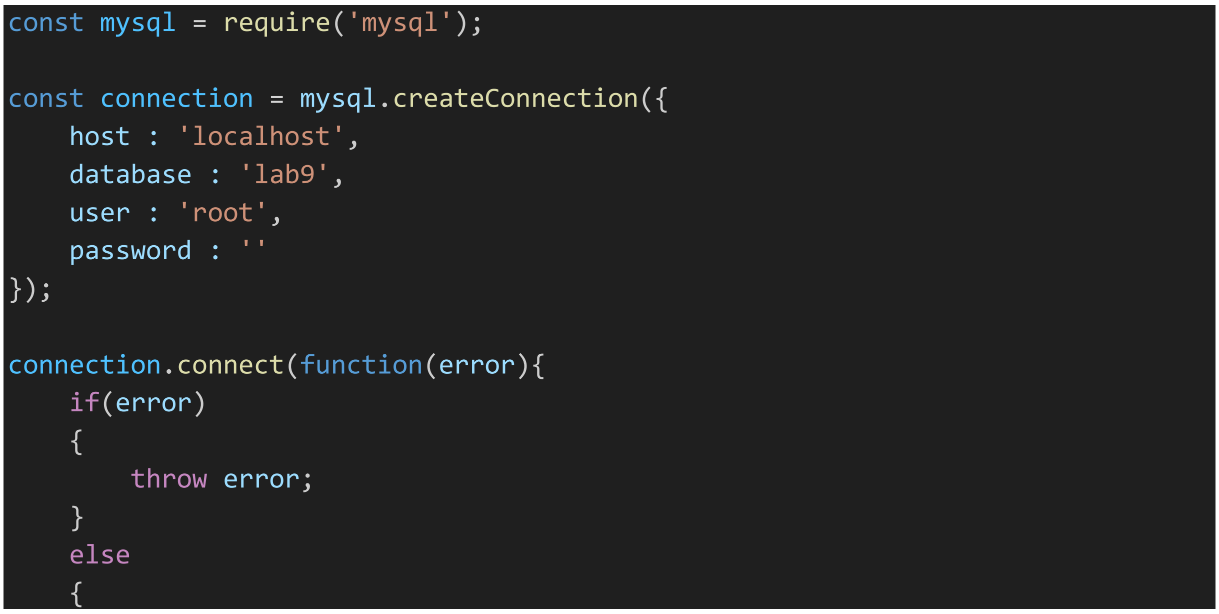
```
D:\lab9test>npx express --view=ejs

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

npm install

After run this command here our download and install of Node JS Express Application process has been completed and we are ready to build our application in Node js.

Step 3- Make MySQL Database Connection with Node Express Application

To connect with the database, create a file named database.js (or any name you would like) in the root directory and copy pate the following configurations:



```
const mysql = require('mysql');

const connection = mysql.createConnection({
    host : 'localhost',
    database : 'lab9',
    user : 'root',
    password : ''
});

connection.connect(function(error){
    if(error)
    {
        throw error;
    }
    else
    {
        console.log('MySQL Database is connected Successfully');
    }
});

module.exports = connection;
```

You need to install mysql package:

npm install mysql

Step 4- Configure Session

Since we use Session for storing user's information, the following package must be installed:

npm install express-session –save

Then, we need to configure the session in app.js (in the root directory):

```
app.use(session({
    secret : 'webslesson',
    resave : true,
    saveUninitialized : true
  }));
```

There are several other things that must be configured. I would recommend to copy/paste the whole app.js from the Zip file to your project.

Step 5- Create Route

Similar to Laravel API and any other REST API framework, you need to define the routes for your API functions. You need to work on routes/index.js. You can copy/paste the whole file from the Zip file.

Step 6- Create Views

Similarly to Laravel, you can create views for your project. We already used ejs engine which helps you rendering the views from ejs files. You can simply change the extension of your HTML files and put them in the views directory. For this project, we need three files:

index.ejs: This is actually the login form

signup.ejs: The form for signup

menu.ejs: The menu page (and contact us)

You can use the session variables in the HTML codes using <% and %> tags:

```
<div class="row mt-5">
                <div class="col-md-3">&nbsp;</div>
                <div class="col-md-6">
                <% if(session.user_id) { %>
                    <meta http-equiv="refresh" content="7; url=/menu" />

                <% } else { %>

                    <div class="card">
                        <div class="card-header">Login</div>
                        <div class="card-body">
```

If you already had any PHP codes from the files of previous lab for this application, please remove them. However, you can keep the JS codes.

Note: You need an upload directory to store the image files of menu items.

Step 7- Start Node Server & Check Output in browser

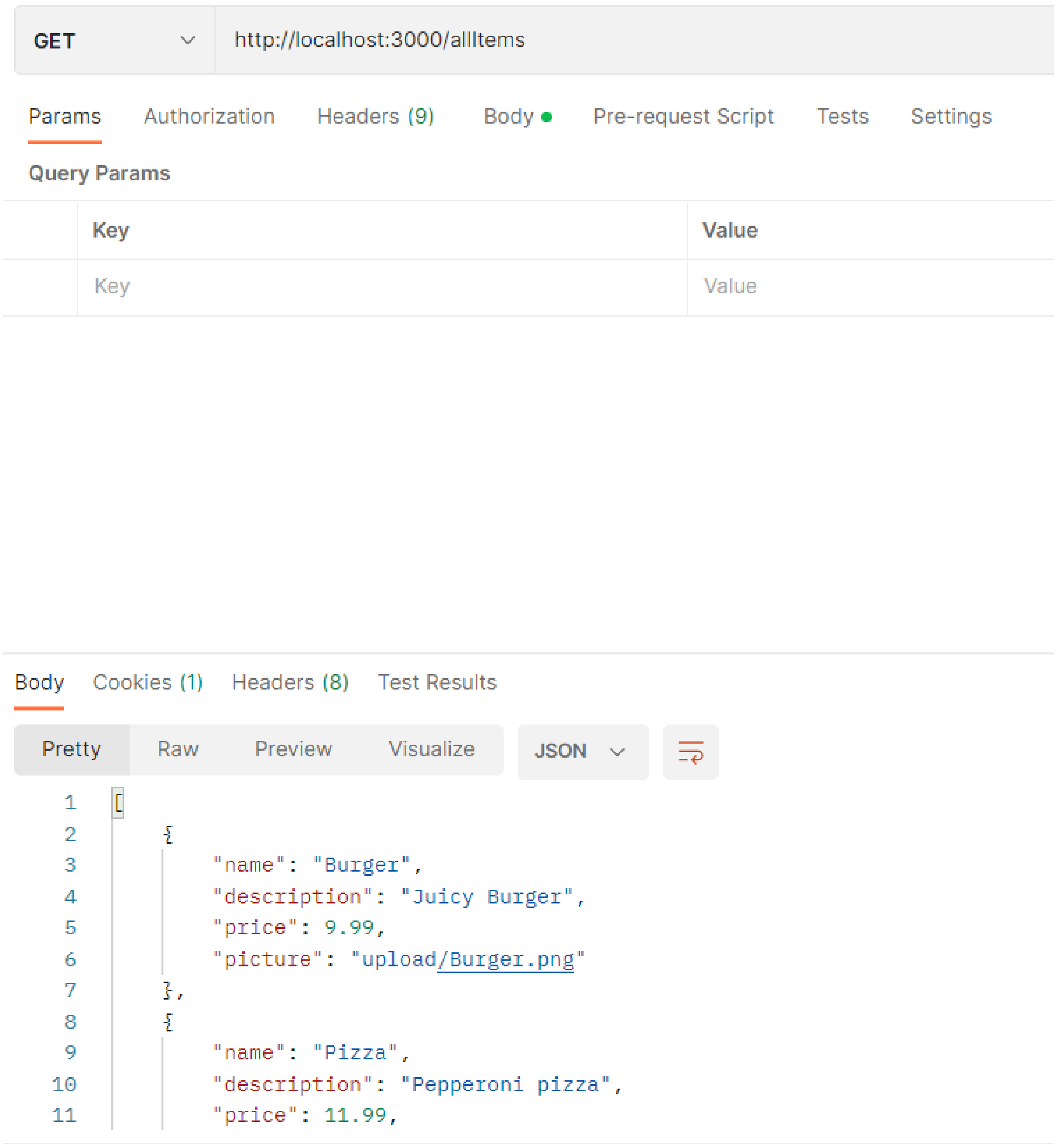
Run the following command to start the server:

npm start

Note: You may need to install several other modules before start the server. For example:

npm install multer
npm install cors

You can test APIs in Postman:


GET http://localhost:3000/allItems

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Query Params

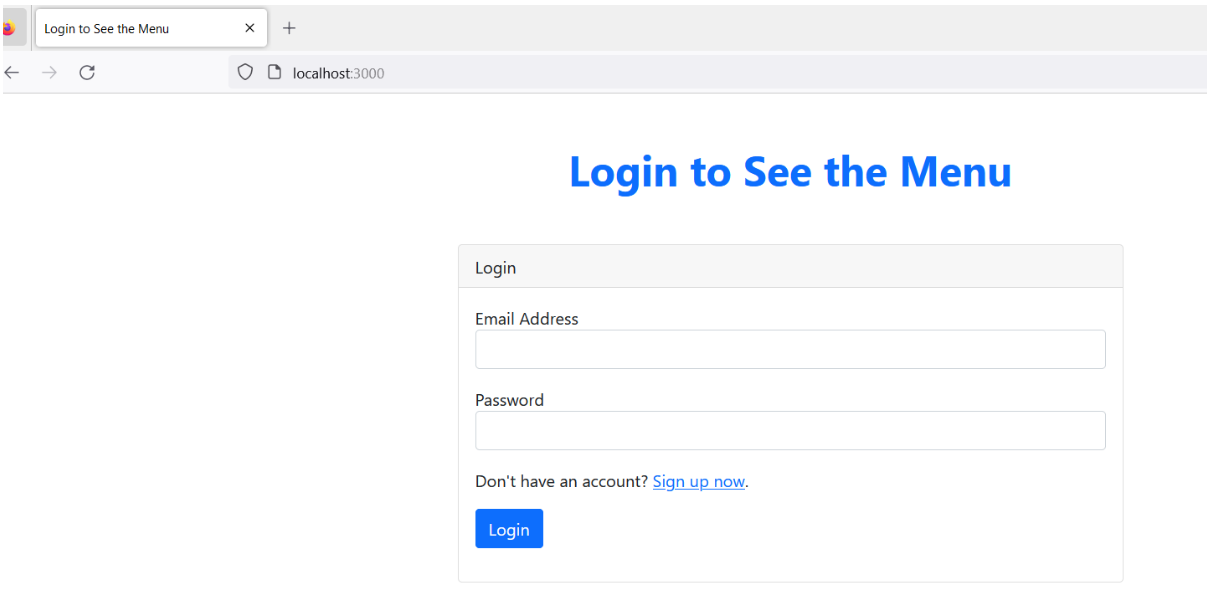
Key	Value
Key	Value

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
[
    {
        "name": "Burger",
        "description": "Juicy Burger",
        "price": 9.99,
        "picture": "upload/Burger.png"
    },
    {
        "name": "Pizza",
        "description": "Pepperoni pizza",
        "price": 11.99,
```

Go to the Browser and type: localhost:3000



Login to See the Menu

Login

Email Address

Password

Don't have an account? Sign up now.

Login

Your application is ready to use.

Final Note

Some of you asked me about the philosophy of using REST APIs and MVC approach instead of using PHP codes in the backend to handle the database and requests from the front-end. Here are some thoughts:

Why APIs?

Please read the following article to know about advantages of using REST APIs:

https://www.quora.com/What-is-REST-API-and-its-advantages

If you ask Chat GPT about the advantages of MVC, you will get these:

1. Separation of Concerns: MVC separates the application into three interconnected components: Model, View, and Controller. This separation allows each component to focus on its specific functionality, making the codebase more modular, maintainable, and easier to understand.
2. Reusability of Code: With a clear separation of concerns, developers can easily reuse the same code across multiple views and controllers. This can save time and effort and improve the overall quality of the application.
3. Testability: Since each component of the application is independent, it can be tested separately, making the testing process more efficient and effective.
4. Flexibility: MVC provides developers with greater flexibility and control over the application's behavior and user interface. They can easily modify or replace a component without affecting the others, which makes the application more adaptable to changing requirements.
5. Improved Collaboration: Because the MVC pattern separates the application into three components, it is easier for multiple developers to work on the same application

simultaneously. They can work on different components without interfering with each other's work.

Overall, using the MVC pattern can result in an application that is easier to develop, maintain, test, and extend.