



Exercise 15: Node.js Mongo Backend

Upload a JavaScript file named `USPnumber1.USPnumber2.js`. If you need to upload a file after the deadline, use `late.USPnumber1.USPnumber2.js`

Exceptionally, if you already made the backend in multiple files, you may use a .zip file.

1. Write a CRUD implementation for the products in your Online Store using MongoDB. A product must be specified using Mongoose and have a name, slug, description, price, and quantity. There is no need for user authentication.

To test, you can use Insomnia or modify the client code below. You can use this implementation with the client code of your store already on GitHub, adding users, orders, etc.

Tip:

The [video lesson](#) shows how to implement a CRUD backend (with the GET, PUT, POST, and DELETE HTTP methods) using MongoDB.

Test app (`test.html`):

```
<!DOCTYPE html>
<html>
  <meta>
    <meta charset="UTF-8">
    <title>User Info</title>
  </meta>

  <body>

    <div id="app">
      <h2>User Information</h2>

      <label>Username</label> <input v-model="username" type="text">
      <br> <br>
      <label>Password</label> <input v-model="password" type="text">
      <br> <br>
      <label>Email</label>      <input v-model="email"      type="text">
      <br> <br>
      <label>Address</label>   <input v-model="address"   type="text">
      <br> <br>
      <input type="button" value="Read" @click="read">
```



UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO
SCC0219 - Introduction to Web Development

```
<input type="button" value="Delete" @click="read">
<input type="button" value="Create" @click="read">
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.1.6/vue.min.js">
</script>

<script>
  new Vue({
    el: "#app",
    data: {
      username: "john",
      password: "",
      email: "",
      address: ""
    },
    methods: {
      read: async function() {
        try {
          let resp = await
fetch("http://www.mocky.io/v2/5ecc1c79300000fceaaddb15");
          resp = await resp.json();
          console.log("resp: "+JSON.stringify(resp));

          this.username= resp.username;
          this.password= resp.password;
          this.email = resp.email;
          this.address= resp.address;
        }
        catch (e) {alert("Error: " + e);}
      }
    }
  });
</script>
</body>
</html>
```

Good Work!