

Assignment 1: CRUD and AMS in Express.js	
CLO 3: Develop applications and tools using various frameworks	
Maximum Marks: 10	Instructor: Dr. Sidra Sultana
Announcement Date: 15 <sup>th</sup> Oct., 2019	Due Date : 23 <sup>rd</sup> Oct., 2019 5:00 pm on LMS

## Scenario

Code Product manager with name, desc, price, category, image and reviews where only authentic user (the one who owns the product) can add, update and delete product.

Admin can add users and assign roles as saler and buyer.

2 dbs, 1 products , 1 user. Owners/users access products, admin access users.

Products schema: id, name, description, price, category, image, reviews, ownername

users schema: ownername, pwd, role

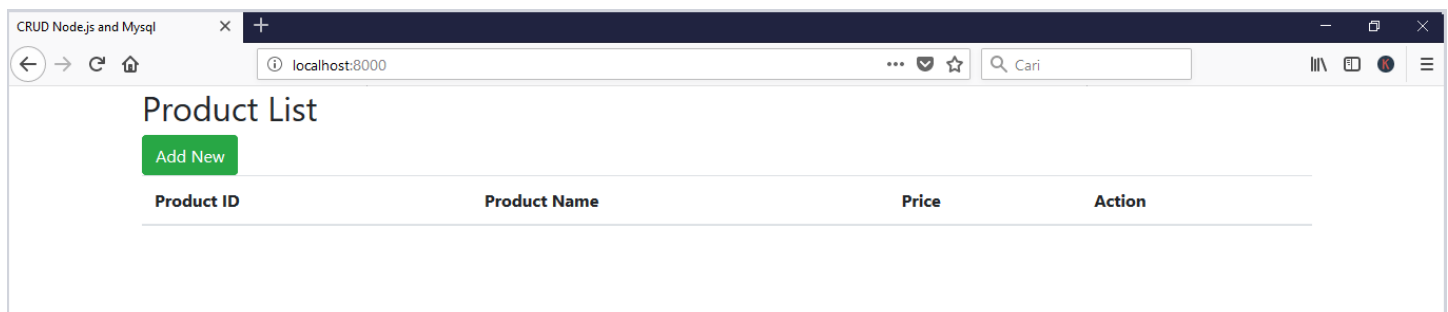
3 views

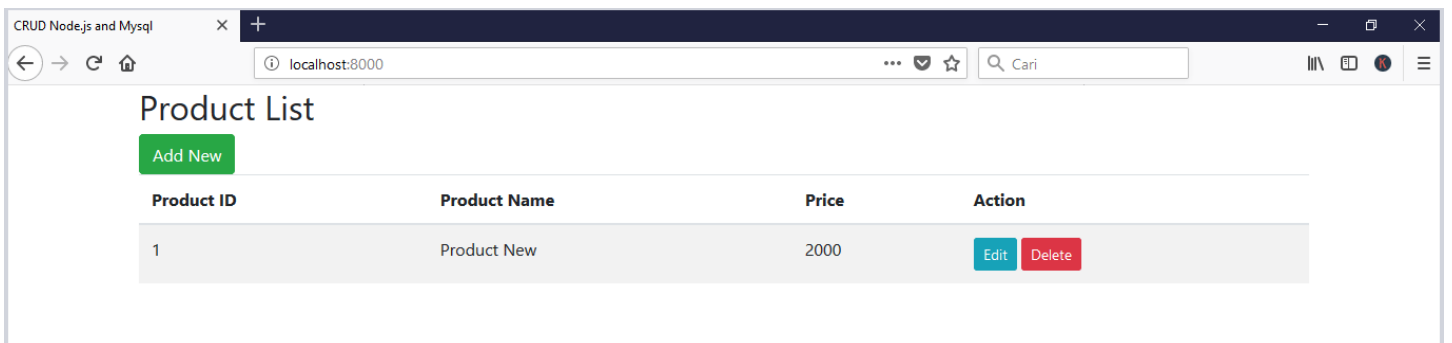
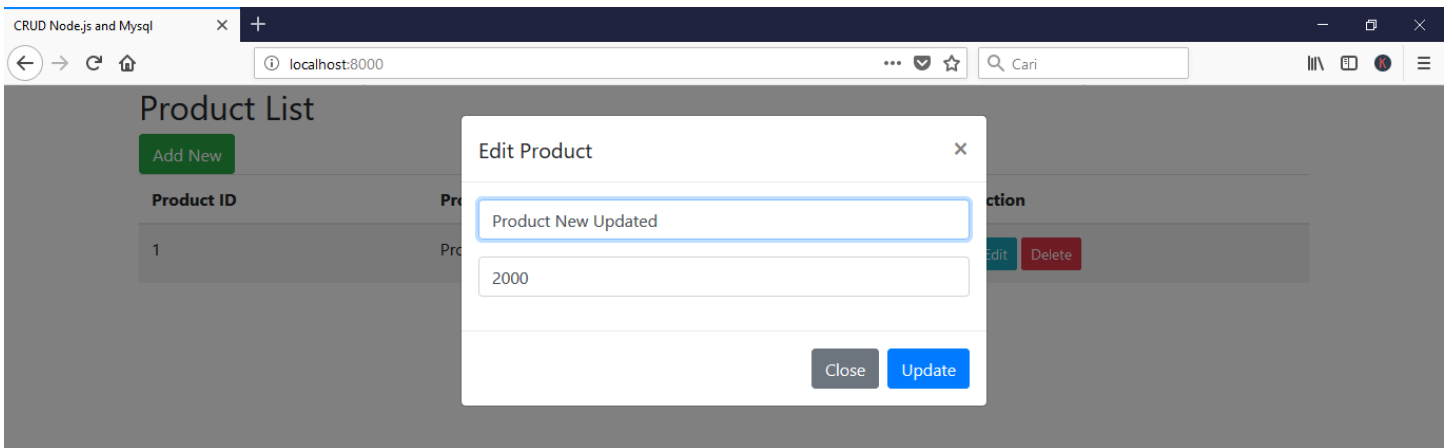
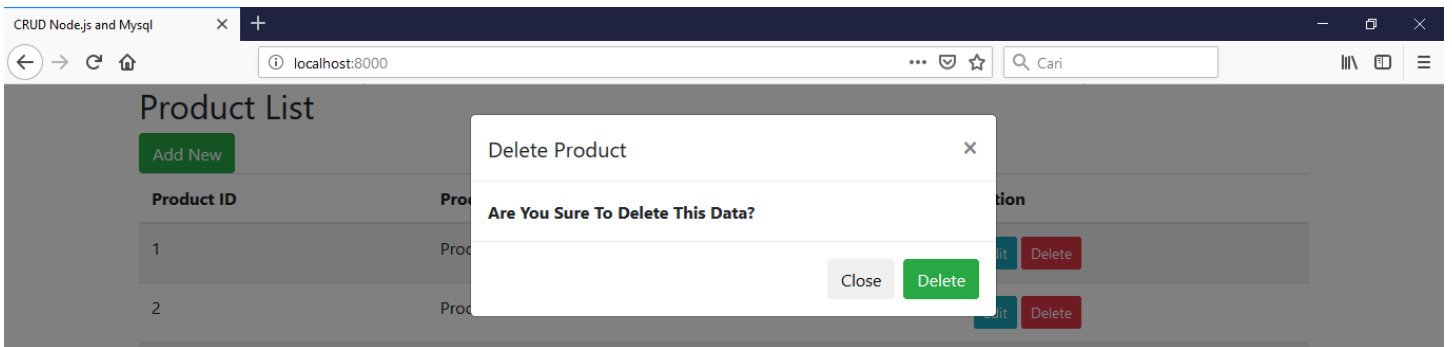
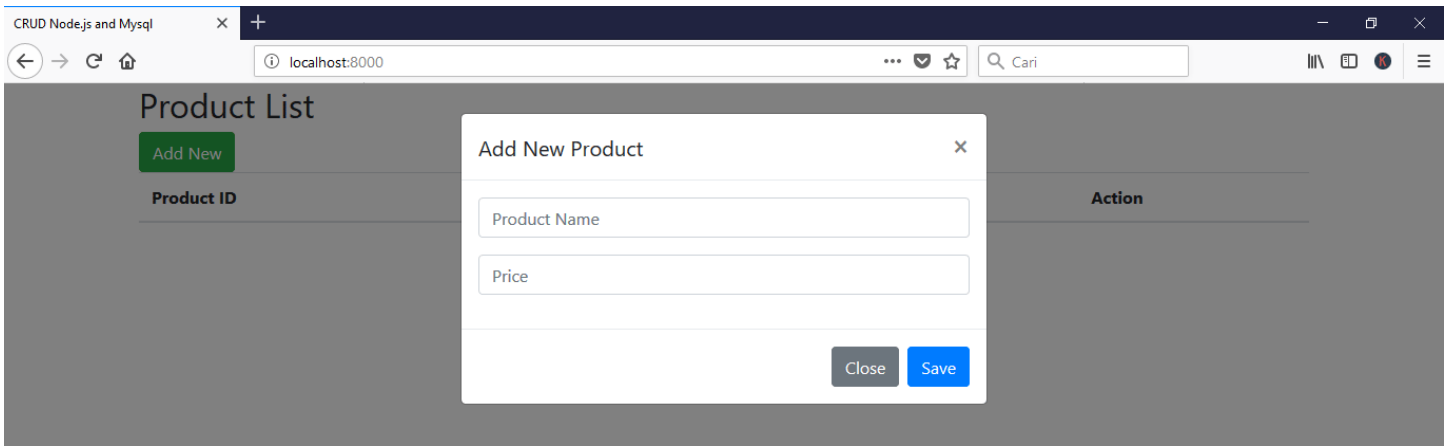
Login view choose admin/user- enter username,pwd

admin view: user db add button on top

user view: products table view edit/delete buttons on every row- add button on top

Sample screenshots with lesser fields are given as an example.





**main express app file:**  
**productmanager.js**

```
var express = require("express");
var bodyParser = require("body-parser");
var multer = require("multer");
var upload = multer();
var cookieParser = require("cookie-parser");
var app = express();
var session = require("express-session");
app.set('view engine', 'pug');
var mongoose = require('mongoose');
//mongoose.connect('mongodb://localhost/pdb');
const mongoURI = 'mongodb://localhost:27017/pdb'
app.set('views', './views');
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(upload.array());
app.use(cookieParser());
app.use(express.static('public'));
app.use(session({secret: "Your secret key"}));
var admin = {
  adminUsername : 'admin',
  adminPwd : 'root'
};
/*var users ={};
var products ={};*/
/*
"/" app.get render login view
"/login" app.post render adminview or productview depending on data recieved
"/products" app.get render productview
"/addproduct" app.post add product
"/editproduct" app.post edit product
"/deleteproduct" app.post delete product
"/admin" app.get render adminview
"/adduser" app.post add user
*/
/*
const conn = mongoose.createConnection(mongoURI, {useNewUrlParser: true,useUnifiedTopology: true});
conn.on('open', function () {
  conn.db.listCollections({name: 'products'})
    .next(function(err, collinfo) {
      if (collinfo) {
        console.log(collinfo);
        products = collinfo[0];
        // The collection exists
      }
    })
  else{
```

```

    var productsSchema = mongoose.Schema({ pname: String,description: String, price
: Number, category: String, image:{ data:Buffer, contentType: String}, reviews: Strin
g });
    products = mongoose.model("products", productsSchema);
    //create collection
    console.log("collection created");
  }
});
conn.db.listCollections({name: 'users'})
.next(function(err, collinfo) {
  if (collinfo) {
    // The collection exists
    console.log(collinfo);
    //users = collinfo[0];
    //var userModel = mongoose.model('users', usersSchema);
    users = mongoose.model('users');
  }
  else{
    var usersSchema = mongoose.Schema({name: String, pwd: String, role: String});
    users = mongoose.model("users", usersSchema);
    //create collection
    console.log("collection created");
    console.log(users);
  }
});
*/

```

```

mongoose.connect('mongodb://localhost/pdb');
var usersSchema = mongoose.Schema({name: String, pwd: String, role: String});
var users = mongoose.model("users", usersSchema);
console.log("collection created");
var productsSchema = mongoose.Schema({ pname: String,description: String, price: Stri
ng, category: String, image:{ data:Buffer, contentType: String}, reviews: String, own
erID: String });
var products = mongoose.model("products", productsSchema);
console.log("collection created");
var Users ={};
users.find(function(err, response){
  console.log(response);
  console.log('here');
  Users = response;
});
console.log(Users);

```

```

var Products ={};
products.find(function(err, response){

```

```

    console.log(response);
    console.log('here');
    Products = response;
  });
  console.log(Products);

  app.get('/app', function(req, res){

    res.render('loginview');

  });
  var Products1 = {};
  app.post('/login', function(req, res){
    console.log('here');
    if(!req.body.username || !req.body.pwd){
      res.status("400");
      res.send("Invalid details!");
    }
    else {
      console.log('here');
      console.log(req.body.mode);
      if(req.body.mode==="Admin"){
        //authenticate superuser
        if(req.body.username === admin.adminUsername && req.body.pwd === admin.adminPwd)
      ){
        console.log('hereadmin');
        req.session.user = admin;
        console.log('here');
        res.redirect('/admin');

      }

    }
    else{
      Users.filter(function(user){
        console.log('here2');
        if(user.name === req.body.username && user.pwd === req.body.pwd){
          req.session.user = user;
          products.find({ownerID: req.session.user._id}, function(err,response){
            console.log("here3");
            console.log(response);
            Products1=response;
          });
          res.redirect('/products');
          return 0;
        }
      });
    }
  });
}

```

```

    }
  });

  app.get('/admin', function(req, res){
    console.log('here');
    res.render('adminview', {userslist: Users});

  });

  app.post('/addUser', function(req, res){
    console.log('here');
    /*var uname = req.body.uname;
    var pwd = req.body.pwd;
    var role = req.body.role;*/
    var userInfo = req.body;
    if(!userInfo.uname || !userInfo.pwd || !userInfo.role){
      console.log("sorry fill all feilds");
    }
    else{
      console.log(userInfo);
      var newUser = new users({name: userInfo.uname, pwd: userInfo.pwd, role: userInfo.
role});
      console.log(newUser);
      newUser.save(function(err, users){
        if(err) throw err;
        else console.log('user added');
      });
      Users.push(newUser);
      console.log('here');
      res.render('adminview', {message: "User has been added successfully", userslist:
Users});
    }

  });

  app.get('/products', function(req, res){
    console.log('here');
    console.log(req.session.user._id);
    console.log("here4");
    console.log(Products1);
    res.render('productsview', {productslist: Products1});

  });

  app.post('/addProduct', function(req, res){
    var product = req.body;
    console.log(req.body);
    console.log(req.session.user);

```

```

    if(!product.pname || !product.price ||!product.category){
        res.render('productsview', {message:"Sorry enter full details",productslist: Products1});
    }
    else{
        //description, price, category, image, reviews, ownername
        var newProduct = new products({pname:product.pname, description:product.description,price:product.price,category:product.category,reviews:"", ownerID: req.session.user._id});
        console.log(newProduct);
        newProduct.save(function(err, users){
            if(err) throw err;
            else console.log('product added');
        });
        Products1.push(newProduct);
        Products.push(newProduct);
        res.render('productsview',{message: "your product has been added", productslist:Products1});
    }

});
var pid="";

app.get('/edit/:id',function(req,res, next){
    console.log(req.params.id);
    console.log("here5");
    pid = req.params.id;
    res.render("editform", {id:pid});

});

app.get('/delete/:id',function(req,res, next){
    console.log(req.params.id);
    console.log("here5");
    pid = req.params.id;
    res.render("deleteitem", {id:pid});

});

app.post('/editProduct', function(req,res){
    var product = req.body;

    console.log(req.session.user);

    products.findByIdAndUpdate(pid,{pname:product.pname, description:product.description,price:product.price,category:product.category}, function(err, response)
    {
        console.log(response);
    }
    );
}

```

```

    });
    products.find({ownerID: req.session.user._id}, function(err,response){
        console.log("here3");
        console.log(response);
        Products1=response;
    });
    products.find(function(err, response){
        console.log(response);
        console.log('here');
        Products = response;
    });
    res.redirect('/updatedProducts');

```

```

});

```

```

app.get('/updatedProducts', function(req,res){
    res.render('productsview',{message: "your products list has been updated", products
list:Products1});
});

```

```

app.post('/deleteProduct',function(req,res){
    console.log(req.session.user._id);
    products.findByIdAndRemove(pid, function(err,response){
        console.log(response);
        console.log(req.session.user._id);
    });
    products.find({ownerID: req.session.user._id}, function(err,response2){
        console.log("here3");
        console.log(response2);
        Products1=response2;
    });
    products.find(function(err, response3){
        console.log(response3);
        console.log('here');
        Products = response3;
    });
    res.redirect('/updatedProducts');
    /*products.findByIdAndRemove(pid, function(err,response){
        console.log(response);
    });
    products.find({ownerID: req.session.user._id}, function(err,response){
        console.log("here3");
        console.log(response);
        Products1=response;
    });
    products.find(function(err, response){

```



```

        console.log(response);
        console.log('here');
        Products = response;
    });
    console.log("/updatedProducts");
    */
});

```

```

app.get('/logout', function(req,res){
    req.session.destroy(function(){
        console.log("user logged out");
        res.redirect("/app");
    })
});

```

```

app.use('/products', function(err, req, res, next){
    console.log(err);
    res.redirect("/app");
});
app.use('/admin', function(err, req, res, next){
    console.log(err);
    res.redirect("/app");
});

```

```

app.listen(3000);

```

## loginview.pug

```

html
  head
    title Product Manager
  body
    h1 LOGIN
    form(action='/login',method = 'POST', enctype='multipart/form-data')
      label(for = "mode") Choose Mode
      input(type="radio" name="mode" value="Admin" checked)
      label Admin
      input(type="radio" name="mode" value="Owner")
      label Owner
      br
      label(for = "username") Enter Username:
      input(type= "text" name = "username")
      br
      label(for = "password") Enter Password:
      input(type= "password" name = "pwd")
      br
      button(type = "submit") Login

```

## adminview.pug

```
html
  head
    style(type='text/css').
      .form-popup {
        display: none;
        position: fixed;
        top: 500;
        left: 700;
        border: 3px solid #f1f1f1;
        z-index: 9;
      }

      .form-container {
        max-width: 300px;
        padding: 10px;
        background-color: white;
        border: 1px solid teal;
      }

      .form-container input[type=text], .form-container input[type=file] {
        width: 100%;
        padding: 15px;
        margin: 5px 0 22px 0;
        border: none;
        background: #f1f1f1;
      }

      a {
        color: black;
        text-decoration: none;
      }

      table {
        width: 100%;
        border-collapse: collapse;
      }

      td, th {
        text-align: center;
        border: 1px solid black;
      }

    script(type='text/javascript').
      function openAddForm() {
        document.getElementById("addform").style.display = "block";
      }

      function closeAddForm() {
        document.getElementById("addform").style.display = "none";
      }

      function openeditForm() {
        document.getElementById("editform").style.display = "block";
      }
```

```

function closeeditForm() {
    document.getElementById("editform").style.display = "none";
}

function opendeleteForm() {
    document.getElementById("deleteform").style.display = "block";
}

function closedeleteForm() {
    document.getElementById("deleteform").style.display = "none";
}

title Product Manager
body
    h1 Products List
    //description, price, category, image, reviews, ownername
    button(style="position:absolute; right:20px")
        a( href="/logout" ) Logout
    button(class='popup', id='addbutton', onclick='openAddForm()') Add New
    div(class='form-popup', id='addform')
        form(action='/addUser',method = 'POST', class='form-container', enctype='multipart/form-data')
            label Enter Username
            input(type= "text" name = "uname" )
            br
            label Enter password
            input(type= "password" name = "pwd")
            br
            label(for = "role") Assign role
            input(type="radio" name="role" value="buyer" checked)
            label Buyer
            input(type="radio" name="role" value="seller")
            label Seller
            br
            button(type = "submit") Add User
            button(type="button" ,onclick='closeAddForm()') Close
        p=message
    table
        tr
            th User ID
            th Name
            th Password
            th Role
        each user in userslist
            tr#user_list_item
                td #{user._id}
                td #{user.name}
                td #{user.pwd}
                td #{user.role}

```

## productsview.pug

```
html
  head
    style(type='text/css').
      .form-popup {
        display: none;
        position: fixed;
        top: 200;
        left: 500;
        border: 3px solid #f1f1f1;
        z-index: 9;
      }

      .form-container {
        max-width: 500px;
        padding: 10px;
        background-color: white;
        border: 1px solid teal;
      }

      .form-container input[type=text], .form-container input[type=file] {
        width: 100%;
        padding: 15px;
        margin: 5px 0 22px 0;
        border: none;
        background: #f1f1f1;
      }

      a {
        color: black;
        text-decoration: none;
      }

      table {
        width: 100%;
        border-collapse: collapse;
      }

      th, td {
        border: 1px solid black;
      }

    script(type='text/javascript').
      function openAddForm() {
        document.getElementById("addform").style.display = "block";
      }

      function closeAddForm() {
        document.getElementById("addform").style.display = "none";
      }

      function openeditForm(id) {
        document.getElementById("editform").style.display = "block";
        console.log("#{id}");
      }
```

```

function closeeditForm() {
    document.getElementById("editform").style.display = "none";
}

function opendeleform() {
    document.getElementById("deleteform").style.display = "block";
}

function closedeleform() {
    document.getElementById("deleteform").style.display = "none";
}

title Product Manager
body
    h1 Products List
    //description, price, category, image, reviews, ownername
    button(style="position:absolute; right:20px")
        a( href="/logout" ) Logout
    button(class='popup', id='addbutton' ,onclick='openAddForm()') Add New
    div(class='form-popup', id='addform')
        form(action='/addProduct',method = 'POST', class='form-container', enctype='multipart/form-data')
            label Enter Product name
            input(type= "text" name = "pname" )
            br
            label Enter Product description
            input(type= "text" name = "description" )
            br
            label Enter price
            input(type= "text" name = "price" )
            br
            label Enter Product category
            input(type= "text" name = "category" )
            br
            label insert image
            input(type="file" name="pic" )
            br
            button(type = "submit") Add product
            button(type="button" ,onclick='closeAddForm()') Close
    p=message
    table
        tr
            th Product ID
            th Name
            th Description
            th Price
            th Category
            th Image
            th Reviews
            th Action
        each product in productslist

```

```
|  |
| --- |
|#product_list_item
  |

```

### editform.pug

```

doctype html
html
  head
    title Product List
  body
    h1.header Edit item #{id}
    form(action='/editProduct',method = 'POST', class='form-container', enctype='multipart/form-data')
      div
        label(for = 'name') Product Name:
        input(name='pname', value='', type='text')
        br
      div
        label(for = 'price') Price:
        input(name='price', type='text')
        br
        label(for = 'desc') Description:
        div
          textarea(name='description', rows="10", cols="60", value='text')
        br
      div
        label(for = 'Image') Image:
        input(name='image', type='file')
        br
      div
        label(for = 'Category') Category:
        input(type= "text" name = "category")
        br
        button(type = "submit") Update product

```

### deleteitem.pug

```

doctype html
html

```

```

<head>
  <title>Product List</title>
</head>
<body>
  <h1>header delete item #{id}</h1>
  <form(action='/deleteProduct',method = 'POST', class='form-container', enctype='multipart/form-data')>
    <button(type = "submit")>Delete product
  </form>
</body>

```

**LOGIN**

Choose Mode ☒ Admin ☐ Owner

Enter Username:

Enter Password:

**Products List**

User ID	Name	Password	Role
5daec51eb593de213f7f2e8b	mariam	aslam	buyer
5daec596b593de213f7f2e8c	mariam	aslam	buyer
5daec734e937d0225a2a9439	umaid	zaffar	buyer
5daf227f56742a2a7064408e	sabayna	ali	seller
5daf232cd3241d2aca01eddf	sarah	aslam	seller
5daf235ab699ec2ae50feed7	saad	aslam	seller
5daf239448f8242afd8d20a8	aisha	aslam	buyer
5daf23a448f8242afd8d20a9	aisha	aslam	buyer
5db1482e694fdf7de2a9ea6e	sultan	fateh	seller

## Products List

User ID	Name	Password	Role
5daec51eb593de213f7f2e8b	mariam	aslam	buyer
5daec596b593de213f7f2e8c	mariam	aslam	buyer
5daec734e937d0225a2a9439	umaid	zaffar	buyer
5daf227f56742a2a7064408e	sabayna	ali	seller
5daf232cd3241d2aca01eddf	sarah	aslam	seller
5daf235ab699ec2ae50feed7	saad	aslam	seller
5daf239448f8242afd8d20a8	aisha	aslam	buyer
5daf23a448f8242afd8d20a9	aisha	aslam	buyer
5db1482e694fdf7de2a9ea6e	sultan	fateh	seller

**Enter Username**

**Enter password**

Assign role ☒ Buyer ☐ Seller

←→↺🏠

localhost:3000/addUser

90%⋮🔍☆

⌵⌵⌵📄🔴⋮

Products List

Add New

Logout

User has been added successfully

User ID	Name	Password	Role
5daec51eb593de213f7f2e8b	marium	aslam	buyer
5daec596b593de213f7f2e8c	marium	aslam	buyer
5daec734e937d0225a2a9439	umaid	zaffar	buyer
5daf227f56742a2a7064408e	sabayna	ali	seller
5daf232cd3241d2aca01eddf	sarah	aslam	seller
5daf235ab699ec2ae50feed7	saad	aslam	seller
5daf239448f8242afd8d20a8	aisha	aslam	buyer
5daf23a448f8242afd8d20a9	aisha	aslam	buyer
5db1482e694fdf7de2a9ea6e	sultan	fateh	seller
5db154d6940ae00303f59796	aksa	mustafa	buyer

user logged out

Products List

Add New

Logout

Product ID	Name	Description	Price	Category	Image	Reviews	Action
5daf41d8c40762399ac67f0b	Dell Laptop bag	essential black 15.6 inch	\$35	Laptop accessories	{}		<div>Edit</div> <div>Delete</div>
5daf3cddc40762399ac67f0a	OnePlus 5t	3500 mah battery, 6.01 inch display, 162gm, snapdragon 835, 6gb ram, 64gb storage	\$500	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>
5daf2eef703c09312e8c5374	Google Pixel 4	2800 mah battery, 5.7inch hd+smooth display, qualcomm snapdragon 855, 6gb ram, 64gb	\$799	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>

er

Enter Product name

Water Bottle

\$50

er

Enter Product description

metallic with straw

\$79

Enter price

\$12

Enter Product category

Supplies

insert image

Browse...

No file selected.

Add product

Close



localhost:3000/addProduct

90%

Products List

Add New

Logout

your product has been added

Product ID	Name	Description	Price	Category	Image	Reviews	Action
5daf41d8c40762399ac67f0b	Dell Laptop bag	essential black 15.6 inch	\$35	Laptop accessories	{}		<div>Edit</div> <div>Delete</div>
5daf3cddc40762399ac67f0a	OnePlus 5t	3500 mah battery, 6.01 inch display, 162gm, snapdragon 835, 6gb ram, 64gb storage	\$500	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>
5daf2eef703c09312e8c5374	Google Pixel 4	2800 mah battery, 5.7inch hd+smooth display, qualcomm snapdragon 855, 6gb ram, 64gb	\$799	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>
5db15651940ae00303f59797	Water Bottle	metalic with straw	\$12	Supplies	{}		<div>Edit</div> <div>Delete</div>

localhost:3000/edit/5db15651940ae00303f59797

Edit item 5db15651940ae00303f59797

Product Name: 

Water Bottle

Price: 

\$15

Description: 

smart metallic water bottle

Image: 

Browse...

 No file selected.

Category: 

Supplies

Update product

## Products List

Add New

Logout

your products list has been updated

Product ID	Name	Description	Price	Category	Image	Reviews	Action
5daf41d8c40762399ac67f0b	Dell Laptop bag	essential black 15.6 inch	\$35	Laptop accessories	{}		<div>Edit</div> <div>Delete</div>
5daf3cddc40762399ac67f0a	OnePlus 5t	3500 mah battery, 6.01 inch display, 162gm, snapdragon 835, 6gb ram, 64gb storage	\$500	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>
5daf2eef703c09312e8c5374	Google Pixel 4	2800 mah battery, 5.7inch hd+smooth display, qualcomm snapdragon 855, 6gb ram, 64gb	\$799	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>
5db15651940ae00303f59797	Water Bottle	smart metallic water bottle	\$15	Supplies	{}		<div>Edit</div> <div>Delete</div>

# delete item 5db15651940ae00303f59797

Delete product

## Products List

Add New

Logout

Product ID	Name	Description	Price	Category	Image	Reviews	Action
5daf41d8c40762399ac67f0b	Dell Laptop bag	essential black 15.6 inch	\$35	Laptop accessories	{}		<div>Edit</div> <div>Delete</div>
5daf3cddc40762399ac67f0a	OnePlus 5t	3500 mah battery, 6.01 inch display, 162gm, snapdragon 835, 6gb ram, 64gb storage	\$500	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>
5daf2eef703c09312e8c5374	Google Pixel 4	2800 mah battery, 5.7inch hd+smooth display, qualcomm snapdragon 855, 6gb ram, 64gb	\$799	Phones and Tablets	{}		<div>Edit</div> <div>Delete</div>