

# LAB INVENTORY

Thomas Van Havere  
Lorenz Put

# VOORBEREIDING

- Laptop
- Account op mongoLabs
- MongoDB server
- Account op Heroku
- Hosting server



# PRIJSKAARTJE

Wat	Hoeveelheid	Kost
Innolux 7'' touchscreen	1	31 €
Barcode Scanner	1	16 €
RPI GPIO extension board	1	3 €
RPI B+	1	40 €
WiPi	1	20 €
Bluetooth stick	1	10 €
SD kaart	1	10 €
Sigaren kistje	1	0 €
Licentie MongoDB	1	0 €
Totaal	7	130 €

# DE MEAN STACK

- **M** for MongoDB
- **E** for Express
- **A** for AngularJS
- **N** for NodeJS



# M FOR MONGODB

- Connectie via NodeJS

```
var mongoose = require("mongoose");  
mongoose.connect('mongodb://<dbusername>:<dbpassword>@ds054308.mongolab.com  
:54308/labinventory');
```

- Document structuur

```
var componentScheme = mongoose.Schema({Type: String, Value: String,  
Quantity: Number, Barcode: String, Note: String});  
  
var componentsmodel = mongoose.model('component', componentScheme);
```



# DE MEAN STACK

- **M** for MongoDB
- **E** for Express
- **A** for AngularJS
- **N** for NodeJS

# E FOR EXPRESS

- Initialiseren Express app

```
var express = require("express");  
var app = express();
```

- Related express modules

```
//set up static routes  
app.use(express.static(__dirname + "/public"));  
  
//Use json body-parser  
app.use(bodyParser.json());  
  
app.use(passport.initialize());
```



# E FOR EXPRESS : LOGIN

- Initialisatie users

```
var user = mongoose.model('User', userSchema);

user.find({}).exec(function(err, collection)
{
  if(collection.length == 0)
  {
    var salt, hash;

    salt = createSalt();
    hash = hashPwd(salt, 'Lorenz');
    user.create({schoolId: "s079368", firstName: "Lorenz",
    lastName: "Put", userName : "Lorenz", salt : salt,
    hashed_pwd: hash});
```



# E FOR EXPRESS : LOGIN

- Authentication

```
userSchema.methods = {  
  authenticate: function (passwordToMatch){  
    return hashPwd(this.salt, passwordToMatch) === this.hashPwd;  
  }  
}
```

```
app.post('/login', function (req, res, next)  
{  
  var auth = passport.authenticate('local', function (err, user)  
  {  
    if (err)  
    {  
      return next(err);  
    }  
    if (!user)  
    {  
      res.send({success: false});  
    }  
    req.logIn(user, function (err)  
    {  
      if (err)  
      {  
        return next(err);  
      }  
      res.send({success : true, user: user})  
    })  
  })  
  auth(req, res, next);  
});
```

# E FOR EXPRESS : ADMINISTRATIE

- Uitloggen en registreren

```
app.post('/logout', function (req,res) {  
  req.logout();  
  res.end();  
});
```

```
app.post('/register', function (req,res) {  
  salt = createSalt();  
  hash = hashPwd(salt, req.body.password);  
  user.create({  
    schoolId : req.body.SchoolId,  
    firstName: req.body.firstname,  
    lastName: req.body.lastname,  
    userName: req.body.username,  
    salt: salt,  
    hashed_pwd: hash  
  });  
  res.end();  
});
```



# DE MEAN STACK

- **M** for MongoDB
- **E** for Express
- **A** for AngularJS
- **N** for NodeJS

# A FOR ANGULARJS

- Laden pagina

```
$http.get('/componentlist').success(function(response)
{
    $scope.componentList = response;
    $scope.component = "";
});
```

- Search bar

```
<tr ng-repeat = "component in componentList | filter: searchText |
orderBy: 'Type' " >
```

```
<label>Search : <input ng-model="searchText"></label>
```



# A FOR ANGULARJS

- Add knop

```
$scope.addComponent = function (type,value)
{
    $http.post('/componentlist', $scope.component)
    .success(function (response) {
        refresh();
    });
};
```

# A FOR ANGULARJS

- Update / edit knop

```
$scope.edit = function(id)
{
    $http.get('/componentlist/' + id).success(function(response)
    {
        $scope.component = response;
    });
};
```

```
$scope.update = function()
{
    $http.put("/componentlist" + $scope.component._id,
    $scope.component).success(function(response)
    {
        refresh();
    });
};
```



# A FOR ANGULARJS

- Remove knob

```
$scope.remove = function(id)
{
    $http.delete("/componentlist/" + id).success(function(response) {
        refresh();
    });
};
```

# A FOR ANGULARJS

- Barcode generator

```
<div id="barcode" barcode-generator="{converttype(component.Type) +  
component.Value}" style="height:100px; text-align: center; width: auto;">  
</div>
```

```
$scope.downloadbarcode = function(id) {  
  html2canvas($("#" + id), {  
    onrendered: function (canvas) {  
      Canvas2Image.saveAsPNG(canvas);  
    }  
  });  
});
```



# DE MEAN STACK

- **M** for MongoDB
- **E** for Express
- **A** for AngularJS
- **N** for NodeJS

# N FOR NODEJS

- Javascript engine
- Basis Express
- I/O events





M

E

A

N