Bagian Ke 12 NodeJS Fullstack Dev

Passport & Login

Disini kita akan menambahkan library baru yaitu passport:

```
npm install passport passport-local --save
```

Kemudian require passport tsb dibawah MongoStore require :

```
var passport = require('passport');
```

Buat folder baru beri nama "models"

Buat file baru beri nama models/user.js (organize file)

Update user.js sbb:

```
var mongoose = require('mongoose');

var UserSchema = new mongoose.Schema({
        email: { type: String, unique: true,
lowercase: true},
        password: String
});

module.exports = mongoose.model('User', UserSchema);
```

Kemudian buat file baru pada folder root, beri nama "passport.js", kemudian update file passport.js sbb:

```
var passport = require("passport");
var localStrategy = require("passport-local");
var User = require('./models/user');

passport.serializeUser(function(user, done){
        done(null, user._id);
});

passport.deserializeUser(function(id, done){
        done(err, user);
});

// Sign in
passport.use('local-login', new localStrategy({
        usernameField: 'email',
        passwordField: 'password',
        passReqToCallback: true
}, function(req, email, password, done){
        User.findOne({email: email}, function(err,
user){
                if (err){
                        return done(err);
                }

                if (!user) {
                        return done(null, false);
                }

                if (!user.comparePassword(password)){
                        return done(null, false);
                }
                return done(null, user);
        });
}));
```

Keterangan Code

- Require, seperti biasa.

- serializeUser, penjelasan secara sederhana adalah menyimpan user._id pada "session".
- deserializeUser, adalah fetching user._id dari database.
- LocalStrategy secara default adalah username & password dan kita menggantikannya dengan email dan password yang digunakan untuk login user.
- passReqToCallback true adalah memperbolehkan kita untuk melewatkan kembali (pass) seluruh request untuk (di) callback. Itu sebabnya kita memasukan request pada callback:

```
function(req, email, done){
        ...
}
```

- if (!user) -> jika user tidak exist kita akan mengembalikan null dan false.
- if (!user.comparePassword(password)){ -> kita membuat custome method "comparePassword", gunanya untuk apabila user memasukan password yang berbeda makan akan di match dengan password yang tersedia. Kemudian akan return null dan false.
- If login sukses -> Kemudian akan di return "user object" yang kemudian akan di store pada serializeUser dan deserializeUser.

Kemudian selanjutnya kita akan membuat URL (path) untuk user login:

=> Kembali ke file server.js, update baris kode ini dibawah middleware:

```
        app.get('/', function(req, res, next){
                res.render("home")
        });
```

```
        app.get('/login', function(req, res, next){
                if (req.user) return
res.redirect('/')
                res.sender('login')
        });
```

Selanjutnya instal template engine: ejs dan ejs-mate:

```
npm install ejs ejs-mate --save
```

Require seperti biasa:

```
var ejs = require('ejs');
var engine = require('ejs-mate');
```

Kemudian setup ejs middleware:

```
app.engine('ejs', engine);
app.set('view engine', 'ejs');
```

Selanjutnya kita harus membuat method POST, dibawah method GET untuk mengirimkan inputan login ke server sbb:

```
app.post('/login', passport.authentication('local-
login', {
        successRedirect: '/profile',
        failureRedirect: '/login'
}))
```

Continue to next video 😃
