Politechnika Rzeszowska

Katedra Informatyki i Automatyki

Aplikacje internetowe

**Połączenie aplikacji "Dziennik elektroniczny" z bazą danych MongoDB**

|  |  |
| --- | --- |
| Prowadzący:  dr inż. Tomasz Rak | Autor:  Radosław Olejarz, L04, olejarz.rad@gmail.com |

Aplikacja "Dziennik elektroniczny" korzysta z bazy danych MongoDB, a wraz z nią z Mongoose - bazującego na schematach formatu modelującego dane aplikacji. Do funkcjonalności aplikacji wykorzystujących bazę danych należą rejestracja uczniów i nauczycieli, logowanie użytkowników, dodawanie ocen przez nauczycieli oraz (założone, lecz niezrealizowane) przeglądanie ocen przez uczniów.

Połączenie z bazą danych zrealizowano w następujący sposób:

*var mongo = require('mongodb');*

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

*mongoose.connect('mongodb://127.0.0.1/database');*

*var db = mongoose.connection;*

Baza została nazwana "database" i zawiera jedną kolekcję o nazwie "users". Przygotowana baza danych zawiera dwóch przykładowych użytkowników, których dane podano w tabeli.

|  |  |  |
| --- | --- | --- |
| Nazwa użytkownika | Hasło użytkownika | Rola użytkownika |
| radol | olrad | uczeń |
| jarpad | padjar | nauczyciel |

Rejestracja nowych użytkowników przeprowadzana jest osobno dla uczniów oraz nauczycieli, ze względu na różne dane wymagane do ukończenia rejestracji. Każdy nowo zarejestrowany użytkownik zapisywany jest w bazie według następującego schematu:

*var UserSchema = mongoose.Schema({*

*username: {type: String, index:true},*

*password: { type: String},*

*email: { type: String},*

*name: { type: String},*

*lastname: { type: String},*

*klasa: { type: String},*

*subject: { type: String},*

*role: { type: String},*

*oceny: { type: Array}});*

Wysłanie formularza rejestracyjnego powoduje przeprowadzenie walidacji wprowadzonych danych. W przypadku braku błędów następuje utworzenie użytkownika (do pól schematu przypisywane są zmienne zawierające dane pobrane z formularza):

*var newUser = new User({ name: name, lastname: lastname, subject: subject, username: username, email: email, password: password, role: role}); //nauczyciel*

*var newUser = new User({ name: name, lastname: lastname, klasa: klasa, username: username, email: email, password: password, role: role}); //uczeń*

*User.createUser(newUser, function(err, user){*

*if(err) throw err;*

*console.log(user);});*

gdzie:

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

*module.exports.createUser = function(newUser, callback){*

*[...]//hashowanie hasła*

*newUser.save(callback);*

*[...]*

*}*

Rola podawana przy rejestracji pozwala na przeniesienie użytkownika na odpowiednią stronę po zalogowaniu. Eliminuje to sytuację, w której użytkownik nie posiadający odpowiednich uprawnień znajduje się na stronie, która powinna być dla niego niedostępna. Realizowane jest to w następujący sposób:

*router.post('/login',*

*passport.authenticate('local', {failureRedirect:'/users/login', failureFlash: true}),*

*function(req, res) {*

*if(req.user.role == 'uczen') { res.redirect('/users/uczen/');}*

*else if (req.user.role == 'nauczyciel') { res.redirect('/users/nauczyciel');}});*

Dodawanie ocen, dostępne wyłącznie dla nauczycieli, odbywa się poprzez wypełnienie formularza, w którym należy podać dane ocenianego ucznia. Wysłanie formularza powoduje przeprowadzenie weryfikację wprowadzonych danych (np. czy ocena mieści się w zadanym przedziale), po której ocena dodawana jest do pola "oceny" danego ucznia. Należy przy tym zaznaczyć, że przy wprowadzaniu danych nauczyciel nie ma prawa wyboru przedmiotu - zakładamy, że może nauczać tylko tego przedmiotu, który podał przy rejestracji.

W przypadku poprawnie podanych danych, dodanie oceny odbywa się następująco:

*User.findOneAndUpdate({name: name, lastname: lastname, klasa: klasa}, {$push: {oceny: {przedmiot: subject, ocena: grade}}}, function(err, doc) { [...] } );*

Przeglądanie ocen przez ucznia miało wyglądać następująco: uczeń po zalogowaniu powinien zobaczyć listę wszystkich przedmiotów, a obok odpowiednio wypisane wszystkie posiadane oceny. Pierwszą przeszkodą w tym celu było to, że wyszukiwanie ocen (z poziomu konsoli) dla danego ucznia, które zawierały zadaną nazwę przedmiotu zwracało wyłącznie pierwszą pozycję z tablicy, niezależnie od tego jak wiele elementów znajdowało się w tablicy z ocenami. Wyszukiwanie ocen w aplikacji zwracało natomiast wartość [Object Object].

*var dane = User.find({name: req.user.name, lastname: req.user.lastname, klasa: req.user.klasa}, {oceny:""});*

lub

*var dane = req.user.oceny;*