***Завдання:*** Створити клієнтську частину сайту з можливістю каталогів, товарів (згідно вашого завдання). Організувати авторизацію на сайті за допомогою логіна і паролю.

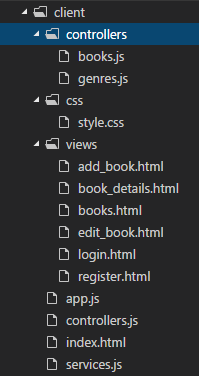
***Тема:*** книжковий менеджер.

***Повний лістинг за посиланням -*** https://github.com/BabalaV/Book-Manager

***Виконання***

Для реалізації клієнтської частини було обрано AngularJS, а також Bootstrap.

Структура:



Оскільки рішення односторінкове (*SPA – Self Page App*), потрібно було прописати необхідні views для перегляду книжкового каталогу, додавання книг, редагування, а також для реєстрації та авторизації користувачів. Приклад view для перегляду детального опису певної книги:

<div class="panel panel-default" ng-init="getBook()">

<div class="panel-heading">

<h3 class="panel-title">{{book.title}}</h3>

</div>

<div class="panel-body">

<div class="row">

<div class="col-md-4">

<img src="{{book.image\_url}}">

</div>

<div class="col-md-4">

<p>{{book.description}}</p>

<ul class="list-group">

<li class="list-group-item">Жанр: {{book.genre}}</li>

<li class="list-group-item">Автор: {{book.author}}</li>

<li class="list-group-item">Видавець: {{book.publisher}}</li>

<li class="list-group-item">К-сть сторінок: {{book.pages}}</li>

</ul>

<a target="\_blank" href="{{book.buy\_url}}" class="btn btn-primary">Купити книгу</a>

</div>

</div>

</div>

</div>

<div class="pull-right">

<a href="#/books/edit/{{book.\_id}}">Редагувати</a> | <a href="#" ng-click="removeBook(book.\_id)">Видалити</a>

</div>

Також необхідні контролери для «відслідковування» дій користувача (*перехід по різним посиланням*), а також контролери для опрацювання даних з api (*book.js, user.js*). Приклад контролера, який відповідає за опрацювання даних з книжкового каталогу:

var myApp = angular.module('myApp');

myApp.controller('BooksController', ['$scope', '$http', '$location', '$routeParams', function($scope, $http, $location, $routeParams){

console.log('BooksController loaded...');

$scope.getBooks = function(){

$http.get('/api/books').success(function(response){

$scope.books = response;

});

}

$scope.getBook = function(){

var id = $routeParams.id;

$http.get('/api/books/'+id).success(function(response){

$scope.book = response;

});

}

$scope.addBook = function(){

console.log($scope.book);

$http.post('/api/books/', $scope.book).success(function(response){

window.location.href='#/books';

});

}

$scope.updateBook = function(){

var id = $routeParams.id;

$http.put('/api/books/'+id, $scope.book).success(function(response){

window.location.href='#/books';

});

}

$scope.removeBook = function(id){

$http.delete('/api/books/'+id).success(function(response){

window.location.href='#/books';

});

}

}]);

Також був написаний сервіс з необхідними для реалізації користувацької авторизації, в якому прописані функції для реєстрації, авторизації та виходу з облікового запису:

angular.module('myApp').factory('AuthService',

['$q', '$timeout', '$http',

function ($q, $timeout, $http) {

var user = null;

return ({

isLoggedIn: isLoggedIn,

getUserStatus: getUserStatus,

login: login,

logout: logout,

register: register

});

function isLoggedIn() {

if(user) {

return true;

} else {

return false;

}

}

function getUserStatus() {

return $http.get('/user/status')

.success(function (data) {

if(data.status){

user = true;

} else {

user = false;

}

})

.error(function (data) {

user = false;

});

}

function login(username, password) {

var deferred = $q.defer();

$http.post('/user/login',

{username: username, password: password})

.success(function (data, status) {

if(status === 200 && data.status){

user = true;

deferred.resolve();

} else {

user = false;

deferred.reject();

}

})

.error(function (data) {

user = false;

deferred.reject();

});

return deferred.promise;

}

function logout() {

var deferred = $q.defer()

$http.get('/user/logout')

.success(function (data) {

user = false;

deferred.resolve();

})

.error(function (data) {

user = false;

deferred.reject();

});

return deferred.promise;

}

function register(username, password) {

var deferred = $q.defer();

$http.post('/user/register',

{username: username, password: password})

.success(function (data, status) {

if(status === 200 && data.status){

deferred.resolve();

} else {

deferred.reject();

}

})

.error(function (data) {

deferred.reject();

});

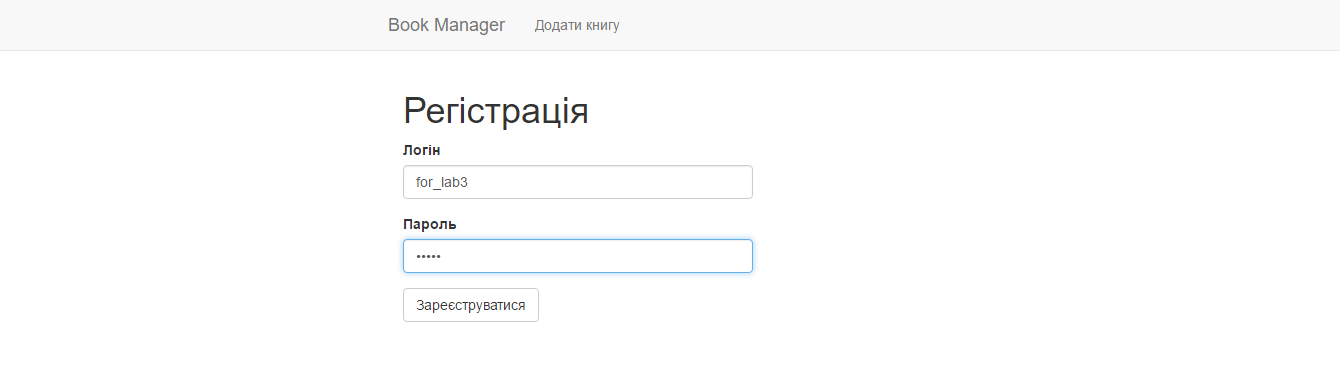
return deferred.promise;

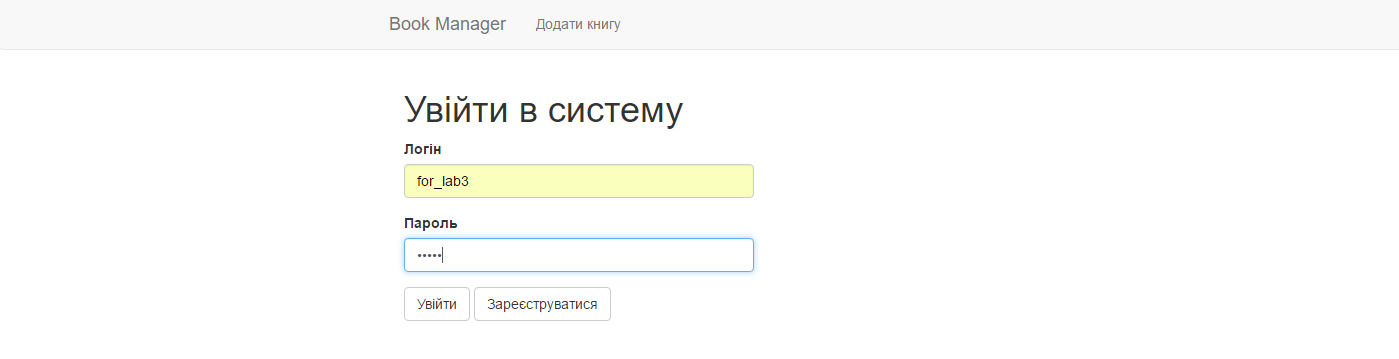
}

}]);

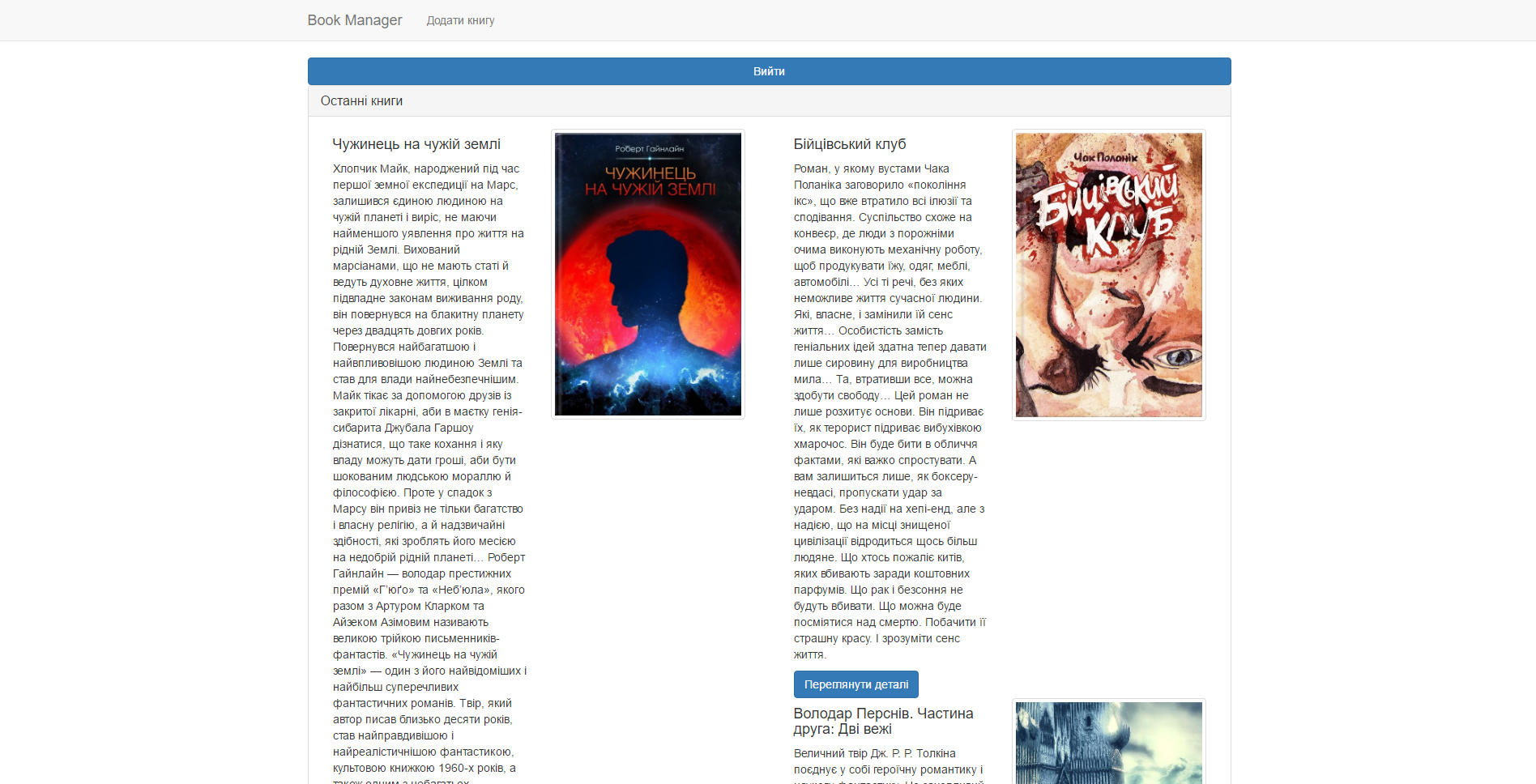
Далі приведені скріншоти роботи рішення, а саме авторизація, перегляд книжкового каталогу і детального опису певної книги:

*Реєстрація та вхід:*

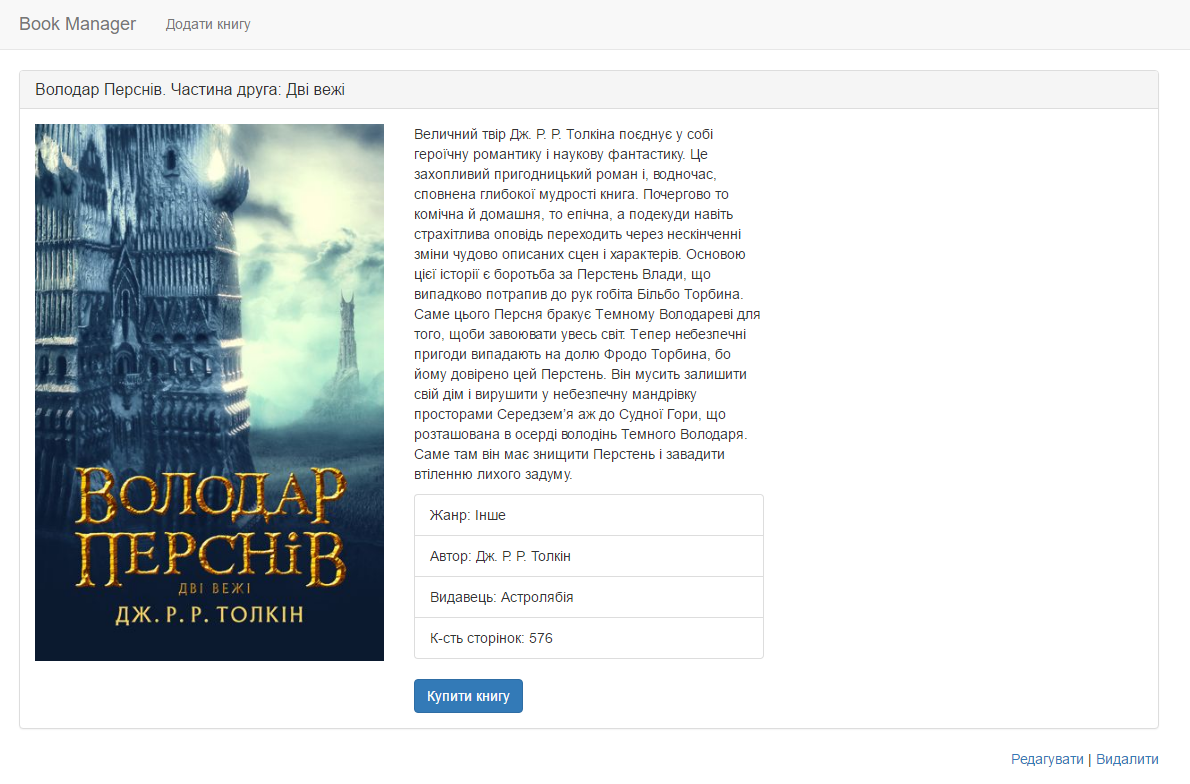
******

******

*Каталог:*

******

*Детальний опис книги і кнопка переходу на інтернет магазин.*

**

***Висновок:***на цій лабораторній роботі було розроблено клієнтську частину за допомогою AngularJS*.*

Міністерство освіти і науки України  
Івано-Франківський національний технічний університет нафти і газу

Кафедра програмного   
забезпечення автоматизованих систем

Лабораторна робота №4

«Розроблення клієнтської частини сайту»

Виконав: ст. гр. ПІ-13-1

Бабала Василь

Перевірив:   
Броновський І. В.

м. Івано-Франківськ

2016