***Завдання:*** Створити адміністративну частину сайту з можливістю реєстрації користувачів в системі. Додати можливість додавання товарів (книжок, людей, згідно вашої теми). Додати можливість додавання, видалення, редагування. Організувати авторизацію на сайті за допомогою логіна і паролю.

***Тема:*** книжковий менеджер.

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

Для робити авторизації на стороні сервера необхідно прописати роути для роботи з даними користувачів (*логіни і паролі*) за допомогою *Passport.js*:

var express = require('express');

var router = express.Router();

var passport = require('passport');

var User = require('../models/user.js');

router.post('/register', function(req, res) {

User.register(new User({ username: req.body.username }),

req.body.password, function(err, account) {

if (err) {

return res.status(500).json({

err: err

});

}

passport.authenticate('local')(req, res, function () {

return res.status(200).json({

status: 'Registration successful!'

});

});

});

});

router.post('/login', function(req, res, next) {

passport.authenticate('local', function(err, user, info) {

if (err) {

return next(err);

}

if (!user) {

return res.status(401).json({

err: info

});

}

req.logIn(user, function(err) {

if (err) {

return res.status(500).json({

err: 'Could not log in user'

});

}

res.status(200).json({

status: 'Login successful!'

});

});

})(req, res, next);

});

router.get('/logout', function(req, res) {

req.logout();

res.status(200).json({

status: 'Bye!'

});

});

router.get('/status', function(req, res) {

if (!req.isAuthenticated()) {

return res.status(200).json({

status: false

});

}

res.status(200).json({

status: true

});

});

module.exports = router;

Mongoose-модель даних користувача (GitHub: /models/user.js):

var mongoose = require('mongoose');

var Schema = mongoose.Schema;

var passportLocalMongoose = require('passport-local-mongoose');

var User = new Schema({

username: String,

password: String

});

User.plugin(passportLocalMongoose);

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

Функції для оновлення, запису і видалення даних, які прописані в Book.js (*GitHub: /models/Book.js*):

Скріншоти роботи системи:

//Отримання книг

module.exports.getBooks = function (callback, limit) {

Book.find(callback).limit(limit);

};

//Отримання певної книги за id

module.exports.getBookById = function (id, callback, limit) {

Book.findById(id, callback);

};

//Додавання книг

module.exports.addBook = function (book, callback) {

Book.create(book, callback);

};

//Оновлення книг

module.exports.updateBook = function (id, book, options, callback) {

var query = {\_id: id};

var update = {

title: book.title,

genre: book.genre,

description: book.description,

author: book.author,

publisher: book.publisher,

pages: book.pages,

image\_url: book.image\_url,

buy\_url: book.buy\_url

}

Book.findOneAndUpdate(query, update, options, callback);

};

//Видалення книг

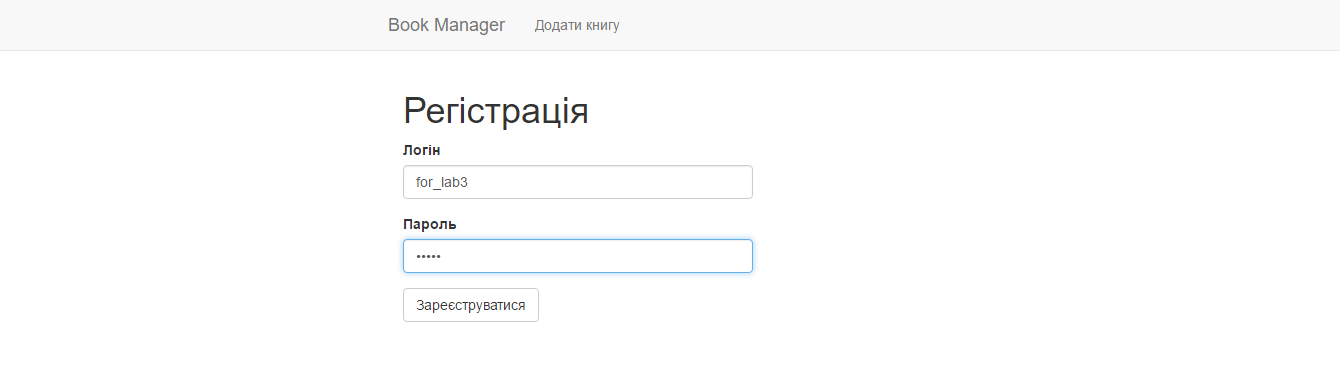
module.exports.removeBook = function (id, callback) {

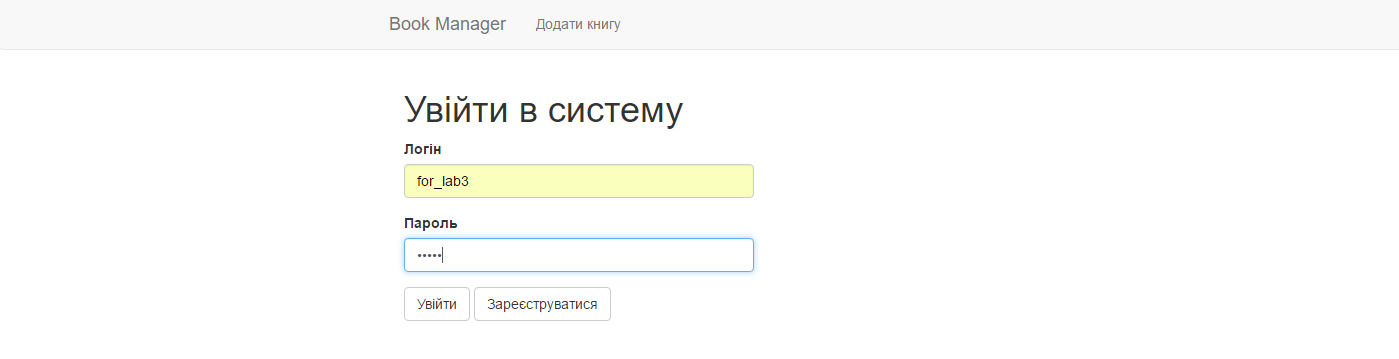
var query = {\_id: id};

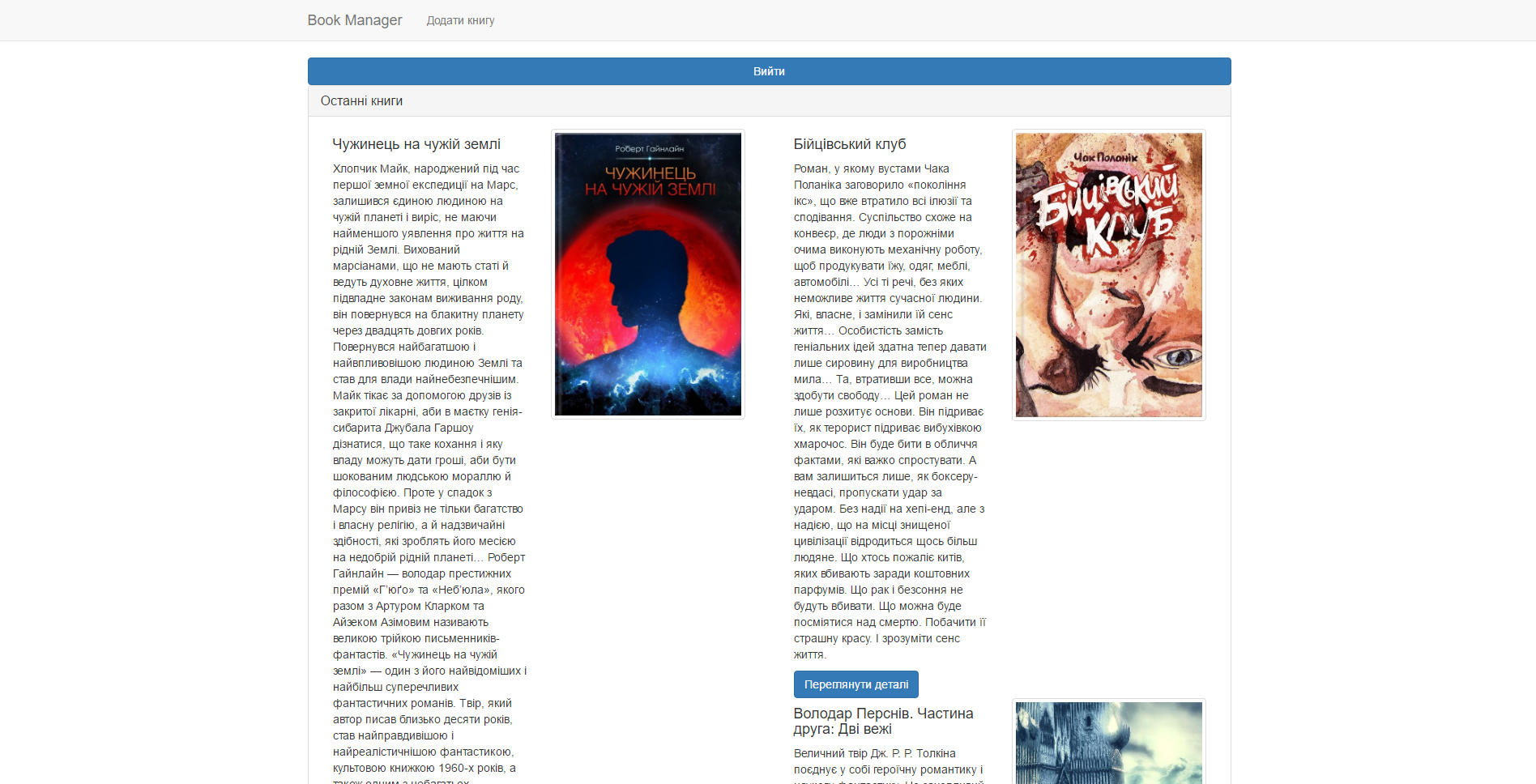
Book.remove(query,callback);

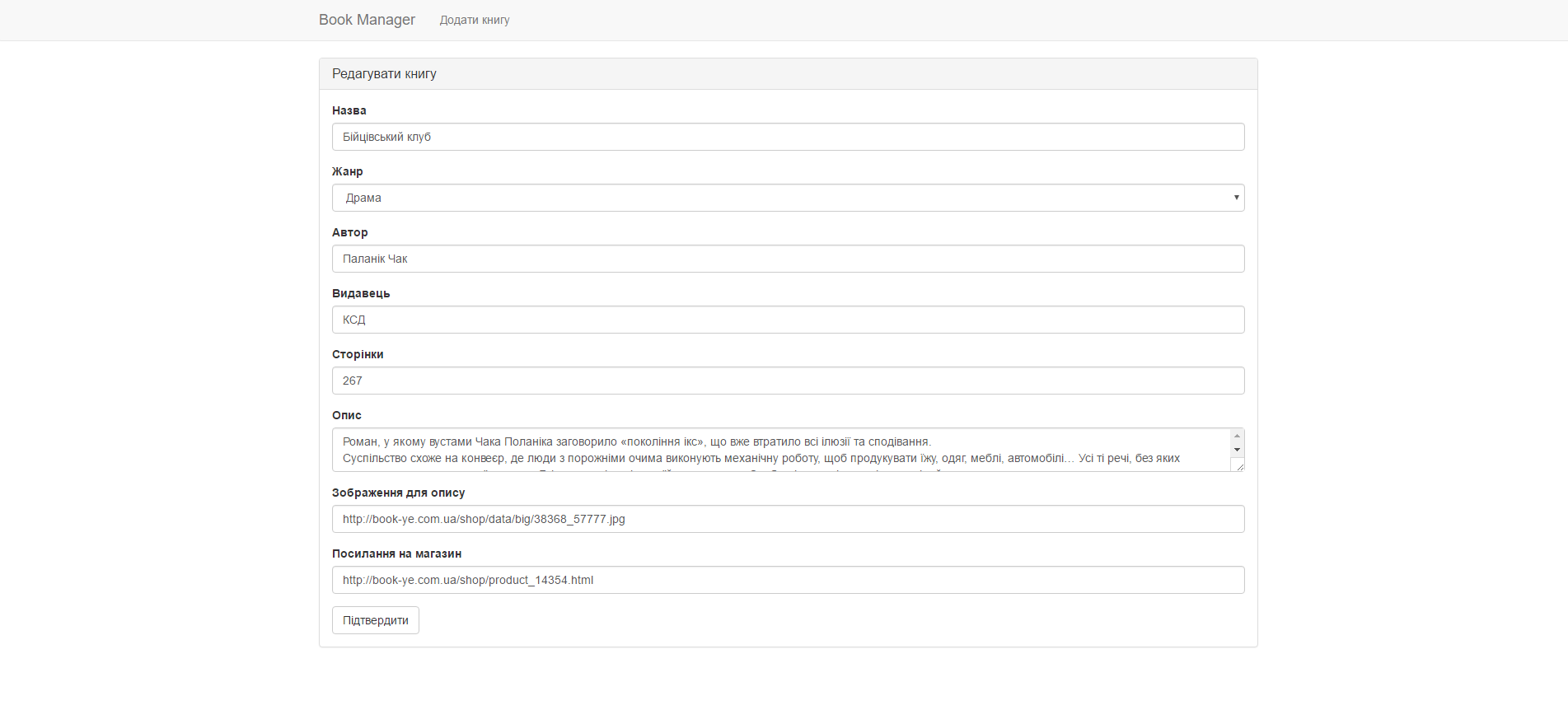
};

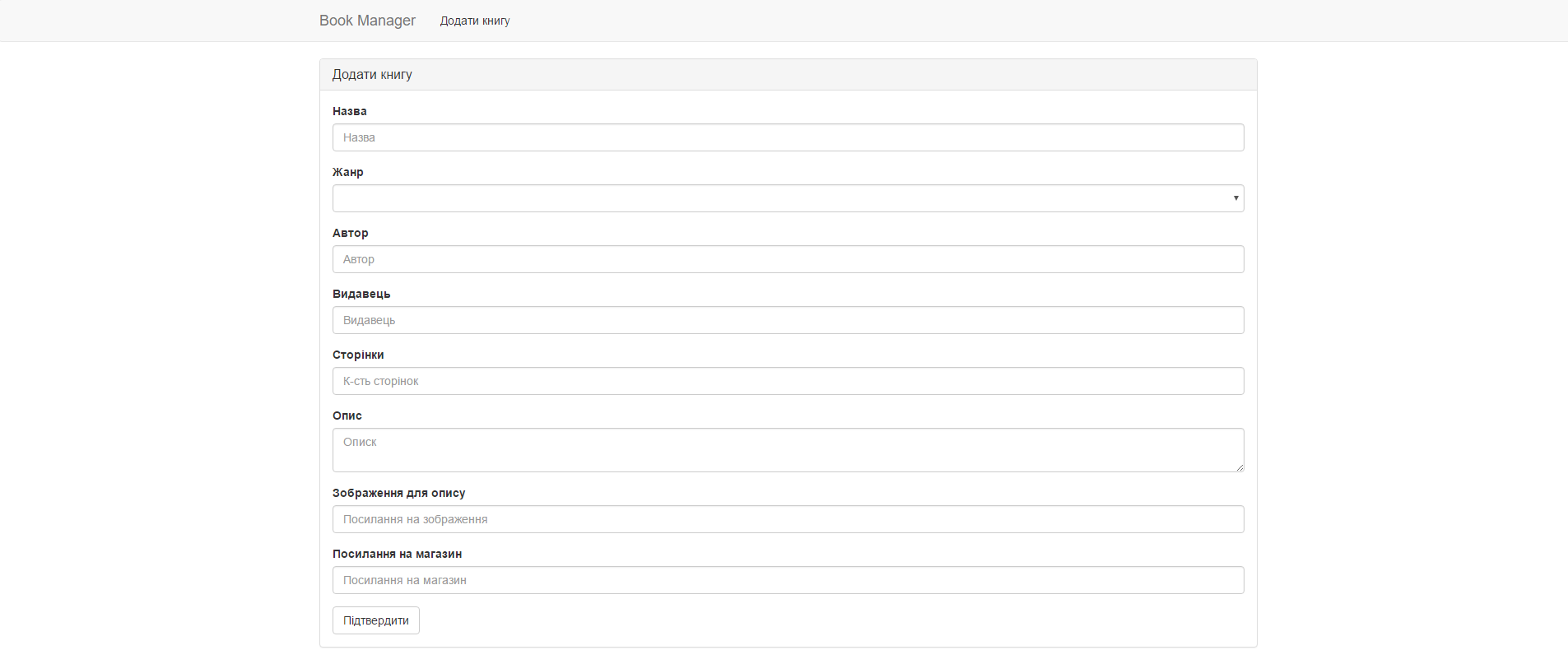
1. ***Реєстрація:***

******

1. ***Вхід:*** ******

******

1. ***Редагування книги:*** ******
2. ***Додавання книги:***

******

***Висновок:*** на цій лабораторній роботі було створено систему авторизації користувачів з шифруванням, а також можливість додавання, видалення або редагування інформацію про книги.

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

Кафедра програмного   
забезпечення автоматизованих систем

Лабораторна робота №3

«Розроблення адміністративної частини сайту»

Виконав: ст. гр. ПІ-13-1

Бабала Василь

Перевірив:   
Броновський І. В.

м. Івано-Франківськ

2016