Event Management System Report

Comp7480 Individual Assignment

Duan Sheng 15400441

Table of Contents:

[1. Introduction 1](#_Toc435655256)

[2. Bonus 1](#_Toc435655257)

[1) Check Code 1](#_Toc435655258)

[2) Change Password 3](#_Toc435655259)

[3) Delete Register 5](#_Toc435655260)

[4) Complete Pagination 5](#_Toc435655261)

[3. Reference 7](#_Toc435655263)

1. Introduction

Event management system is designed for HKBU staffs and students to help them manage events. In the system, there are three types of roles and they are administrator, staff and common member. Different role have different authority. Administrator of the system have the highest authority and he can do any operations. The role of staff can create new events and manipulate the events related to himself. While, common member can only register events and unregister events and view the contents launched by administrator and staff.

The system was divided into two part. One is designed for web and another is designed for mobile. Through sending request from mobile side to web side, the system on mobile can get the data responded by web side. Using this method integrated the two part together.

The system was designed with Node.js and use Bootstrap to control style. Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world [1]. Bootstrap is the world’s most popular framework for building responsive, mobile-first sites and applications. Inside you’ll find high quality HTML, CSS, and JavaScript to make starting any project easier than ever [2].

1. Bonus
2. Check Code



In login page, the function of checking valid code was added. Adding this function is for the purpose of hacker using a certain account to try to brute force with the help of some hacker software. Through inputting check code, can make sure that there is a real person is trying to login the system. The check code is 4 random digital number. Through click the number, the check code is able to change to another. Actually, in order to become more secure, the check code should be generated in background but using Node.js to generate random number need to import a module and it’s not convenience. So the system applied JavaScript to generate check code. The related code as follow:

<form action="/user/login" method="POST" onsubmit="return checkForm();">

Check code:<a href="javascript:valid()" id="valid"><a>

JS:

var code="";

function valid()

{

code="";

for(i=1;i<=4;i++)

{

code=code+Math.floor(Math.random()\*10);

}

document.getElementById("valid").innerHTML=code;

}

valid();

function checkForm(){

var validCode = document.getElementById("validCode").value;

if(validCode != code){

alert("wrong valid code");

return false;

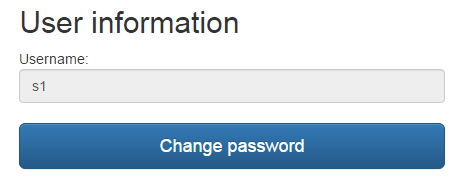
}

}

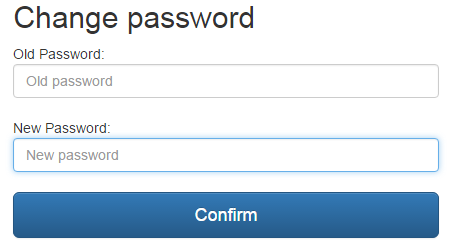
1. Change Password



Click Hi,s1 and then go to:



Click change password and then go to:



Inputting old password and new password, if old password is not correct the system will give promote message. If the old password is right, the user’s password will change to the new one and the session of login information will clean and the page will go to the login page. The main code:

changePassword: function(req, res) {

var oldPassword=req.body.oldPassword;

var newPassword=req.body.newPassword;

var password= req.session.password;

if(oldPassword != password)

{

return res.send("old password not correct");

}

else{

User.findOne(req.session.user.id).exec( function(err, u) {

u.password=newPassword;

u.save();

req.session=null;

return res.view("login");

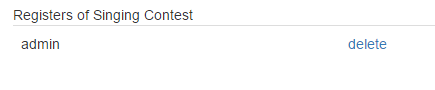
});

}

},

1. Delete Register

Adding the function of enabling administrator and staff can delete registed users of events.



Clikd delete, the register of a certain envet will be removed. Main code:

deleteRegister : function (req, res) {

var uid=req.query.uid;

User.findOne(uid).exec( function (err, model) {

if (model !== null) {

model.regists.remove(req.session.aid)

model.save();

Activity.findOne(req.session.aid).populateAll().exec( function (err, m) {

m.quota=m.quota+1;

return res.view("registedUsers",{"activity":m});

});

}

else {

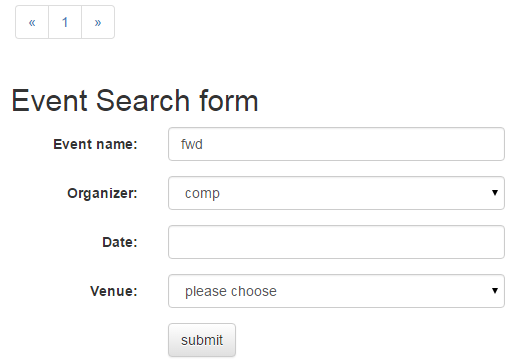
return res.send("User not found!");

}

})

},

1. Complete pagination



Click submit, The result of events will show on the left panels. Using Jquery control the query criterias in the form to keep it not change. Clikc pagination bar, the results will correctly show the evnets of current criterias. The background code is too long. Here just give main Js code:

$(function(){

$("#organizer").val("<%=organizer%>");

$("#venue").val("<%=venue%>");

if("<%=organizer%>" =="undefined")

{

$("#organizer").val("");

}

if("<%=venue%>" =="undefined")

{

$("#venue").val("");

}

if("<%=name%>" =="undefined")

{

$("#name").val("");

}

if("<%=date%>" =="undefined")

{

$("#date").val("");

}

})

1. References
2. Node.js: [URL: <https://nodejs.org/en/>]
3. Bootstrap: [URL: [*http://v4-alpha.getbootstrap.com/getting-started/introduction/*](http://v4-alpha.getbootstrap.com/getting-started/introduction/)]