IT 6203 IT Design Studio  
Michael Young  
February 25, 2018  
[southernmiss@gmail.com](mailto:southernmiss@gmail.com)

Individual Project

# Table of Contents

[Project Overview 3](#_Toc507435863)

[Instructions for Installation 4](#_Toc507435864)

[Database Design 6](#_Toc507435865)

[Code and Screenshots 9](#_Toc507435866)

[**Client Side Validation** 9](#_Toc507435867)

[**Server Side Validation** 12](#_Toc507435868)

[Conclusion 19](#_Toc507435869)

# Project Overview

Users within the CCSE community will have to register and fill out their profiles that include a list of services they can provide to the community (Java Tutoring, SQL Tutoring, Basic PC Troubleshooting, etc) and their availability (weekday morning, weekends, etc).

The idea is that members in the CCSE community can be both service providers and service requesters. For instance, Student A has taken advance Java classes and feels confident with his/her Java skills and would like to offer Java Tutoring to the CCSE community. On the registration page, Student A will fill out his/her profile and select Java Tutoring as one of the services that he/she can provide.

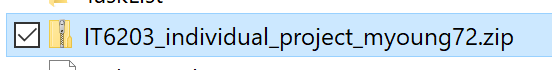
**Features**

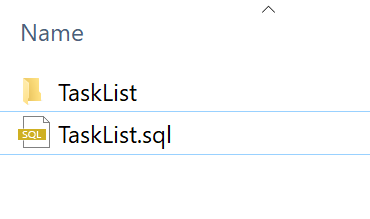
Users can:

* Register and fill out their profile, which include a list of services that they can provide and a list of date/time that they can provide the services.
* Search services offered by the registered users.

# Instructions for Installation

A zip file named **IT6203\_individual\_project\_myoung72.zip** is provided in order to get started. After extracting the zip file to a temporary location, you will notice one folder named **TaskList** and file named TaskList.sql exist. The folder **TaskList** contains the PHP source code for the project. The TaskList.sql file contains the SQL code that installs the database, creates a proxy user, creates the tables, inserts data into the tables, and grants privileges to the proxy user in order to query and insert into the tables.



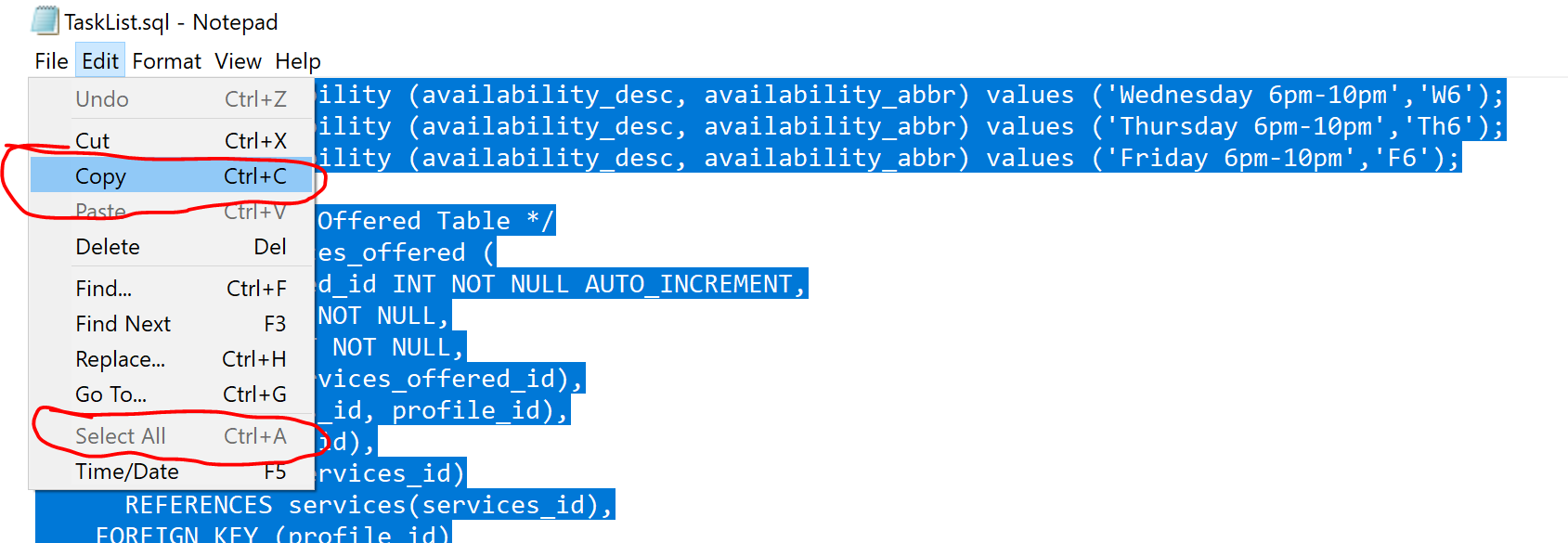


The steps to install this project are as follows:

1. Extract IT6203\_individual\_project\_myoung72.zip to a temporary folder.
2. Copy **TaskList** folder that you extracted to your Document Root in Apache (i.e. C:/xampp/htdocs ).
3. Make sure XAMPP Apache and MySQL client are running.
4. Connect to your MySQL instance with your root user and your root password that you created in a previous lab …

mysql -u root –p

1. Once you are connected to MySQL, open the file TaskList.sql in any editor like notepad or TextEdit. Highlight everything in the file and copy.

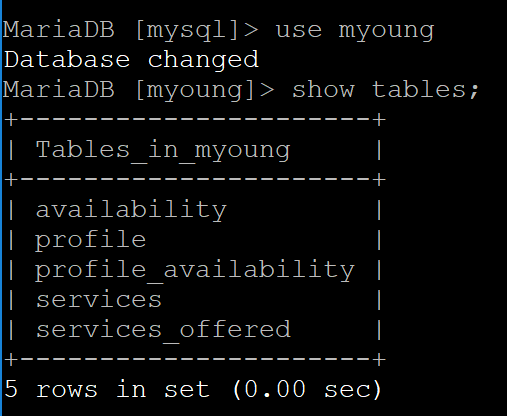


1. Paste into your mysql window (try right-clicking). This should run all the code necessary to set up the database.
2. Once finished. In mysql type:

use myoung;

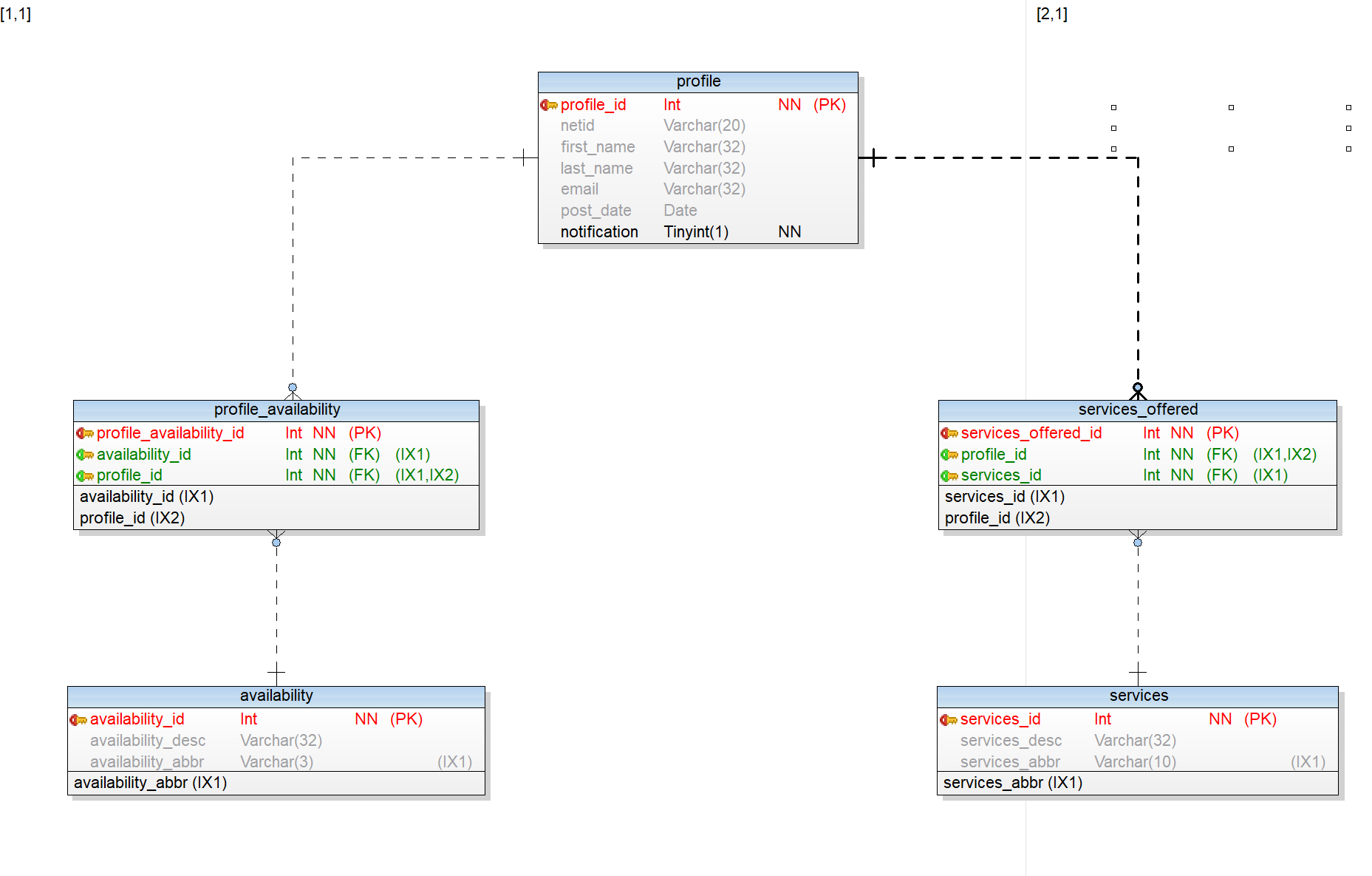
show tables;

1. The tables below should be shown:



# Database Design

The myoung Database has the following relationships:



**Profile** holds the users profile during registration.

**Profile\_Availability** holds the Available days and times of the person registering.

**Services\_Offered** holds the services the student checks in the registration page.

**Services** and **Availability** are lookup tables that have been prepopulated from the previous TaskList.sql script.

The SQL that produces these tables and inserts the data is below:

/\* Create Database \*/

**CREATE** DATABASE myoung**;**

/\* Change Database \*/

**use** myoung**;**

/\* Create Services Table \*/

**create** **table** services **(**

services\_id **int** **not** **null** auto\_increment**,**

services\_desc **varchar(**32**),**

services\_abbr **varchar(**10**),**

**primary** **key** **(**services\_id**),**

**INDEX** **(**services\_abbr**)**

**);**

/\* Insert into Services Table \*/

**insert** **into** services **(**services\_desc**,**services\_abbr**)** **values** **(**'PHP Tutoring'**,**'php'**);**

**insert** **into** services **(**services\_desc**,**services\_abbr**)** **values** **(**'C Tutoring'**,**'C'**);**

**insert** **into** services **(**services\_desc**,**services\_abbr**)** **values** **(**'Python Tutoring'**,**'python'**);**

**insert** **into** services **(**services\_desc**,**services\_abbr**)** **values** **(**'Database Training'**,**'database'**);**

**insert** **into** services **(**services\_desc**,**services\_abbr**)** **values** **(**'PC Repair'**,**'PC'**);**

**insert** **into** services **(**services\_desc**,**services\_abbr**)** **values** **(**'Website Development'**,**'web'**);**

/\* Create Profile Table \*/

**create** **table** profile **(**

profile\_id **INT** **NOT** **NULL** AUTO\_INCREMENT**,**

netid **varchar(**20**),**

first\_name **varchar(**32**),**

last\_name **varchar(**32**),**

email **varchar(**32**),**

post\_date **date,**

notification **tinyint(**1**)** **not** **null** **default** 0**,**

**PRIMARY** **KEY** **(**profile\_id**)**

**)** **;**

/\* Create Availability Table \*/

**create** **table** availability **(**

availability\_id **int** **not** **null** auto\_increment**,**

availability\_desc **varchar(**32**),**

availability\_abbr **varchar(**3**),**

**primary** **key** **(**availability\_id**),**

**INDEX** **(**availability\_abbr**)**

**);**

/\* Insert into Availability Table \*/

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Monday 8am-12pm'**,**'M8'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Tuesday 8am-12pm'**,**'T8'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Wednesday 8am-12pm'**,**'W8'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Thursday 8am-12pm'**,**'Th8'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Friday 8am-12pm'**,**'F8'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Monday 1pm-5pm'**,**'M1'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Tuesday 1pm-5pm'**,**'T1'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Wednesday 1pm-5pm'**,**'W1'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Thursday 1pm-5pm'**,**'Th1'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Friday 1pm-5pm'**,**'F1'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Monday 6pm-10pm'**,**'M6'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Tuesday 6pm-10pm'**,**'T6'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Wednesday 6pm-10pm'**,**'W6'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Thursday 6pm-10pm'**,**'Th6'**);**

**insert** **into** availability **(**availability\_desc**,** availability\_abbr**)** **values** **(**'Friday 6pm-10pm'**,**'F6'**);**

/\* Create Services Offered Table \*/

**CREATE** **TABLE** services\_offered **(**

services\_offered\_id **INT** **NOT** **NULL** AUTO\_INCREMENT**,**

profile\_id **INT** **NOT** **NULL,**

services\_id **INT** **NOT** **NULL,**

**PRIMARY** **KEY(**services\_offered\_id**),**

**INDEX** **(**services\_id**,** profile\_id**),**

**INDEX** **(**profile\_id**),**

**FOREIGN** **KEY** **(**services\_id**)**

**REFERENCES** services**(**services\_id**),**

**FOREIGN** **KEY** **(**profile\_id**)**

**REFERENCES** profile**(**profile\_id**)**

**)** **;**

/\* Create Profile Availability \*/

**CREATE** **TABLE** profile\_availability **(**

profile\_availability\_id **INT** **NOT** **NULL** AUTO\_INCREMENT**,**

availability\_id **INT** **NOT** **NULL,**

profile\_id **INT** **NOT** **NULL,**

**PRIMARY** **KEY(**profile\_availability\_id**),**

**INDEX** **(**availability\_id**,** profile\_id**),**

**INDEX** **(**profile\_id**),**

**FOREIGN** **KEY** **(**availability\_id**)**

**REFERENCES** availability**(**availability\_id**),**

**FOREIGN** **KEY** **(**profile\_id**)**

**REFERENCES** profile**(**profile\_id**)**

**)** **;**

/\* Create Proxy User \*/

**CREATE** **USER** 'tasklist\_user'@'localhost' **IDENTIFIED** **BY** 'my\*password'**;**

/\* Create Grants \*/

**GRANT** **SELECT,** **UPDATE,** **INSERT** **ON** myoung**.**profile **TO** 'tasklist\_user'@'localhost'**;**

**GRANT** **SELECT,** **UPDATE,** **INSERT** **ON** myoung**.**services\_offered **TO** 'tasklist\_user'@'localhost'**;**

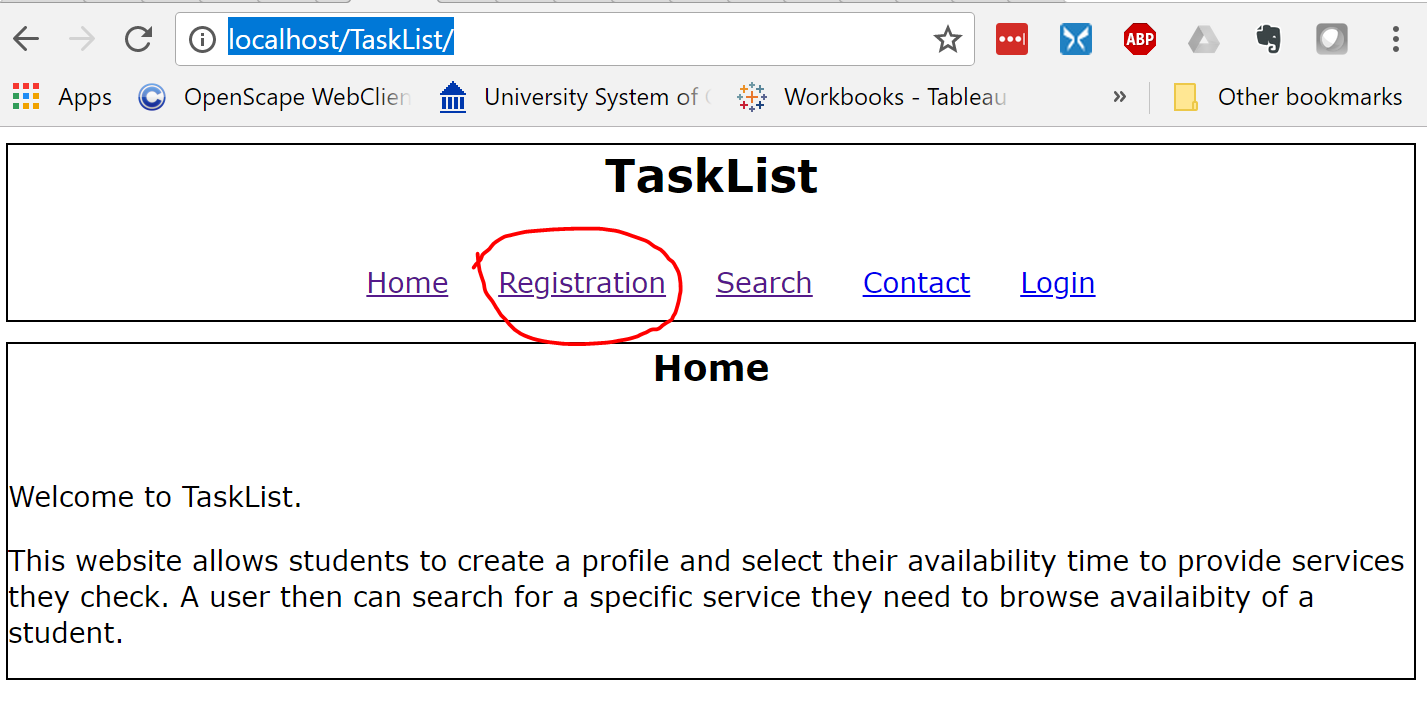
**GRANT** **SELECT,** **UPDATE,** **INSERT** **ON** myoung**.**profile\_availability **TO** 'tasklist\_user'@'localhost'**;**

**GRANT** **SELECT** **ON** myoung**.**services **to** 'tasklist\_user'@'localhost'**;**

**GRANT** **SELECT** **ON** myoung**.**availability **to** 'tasklist\_user'@'localhost'**;**

# Code and Screenshots

Browse to <http://localhost/TaskList/> then click on Registration up top on the menu.



### **Client Side Validation**

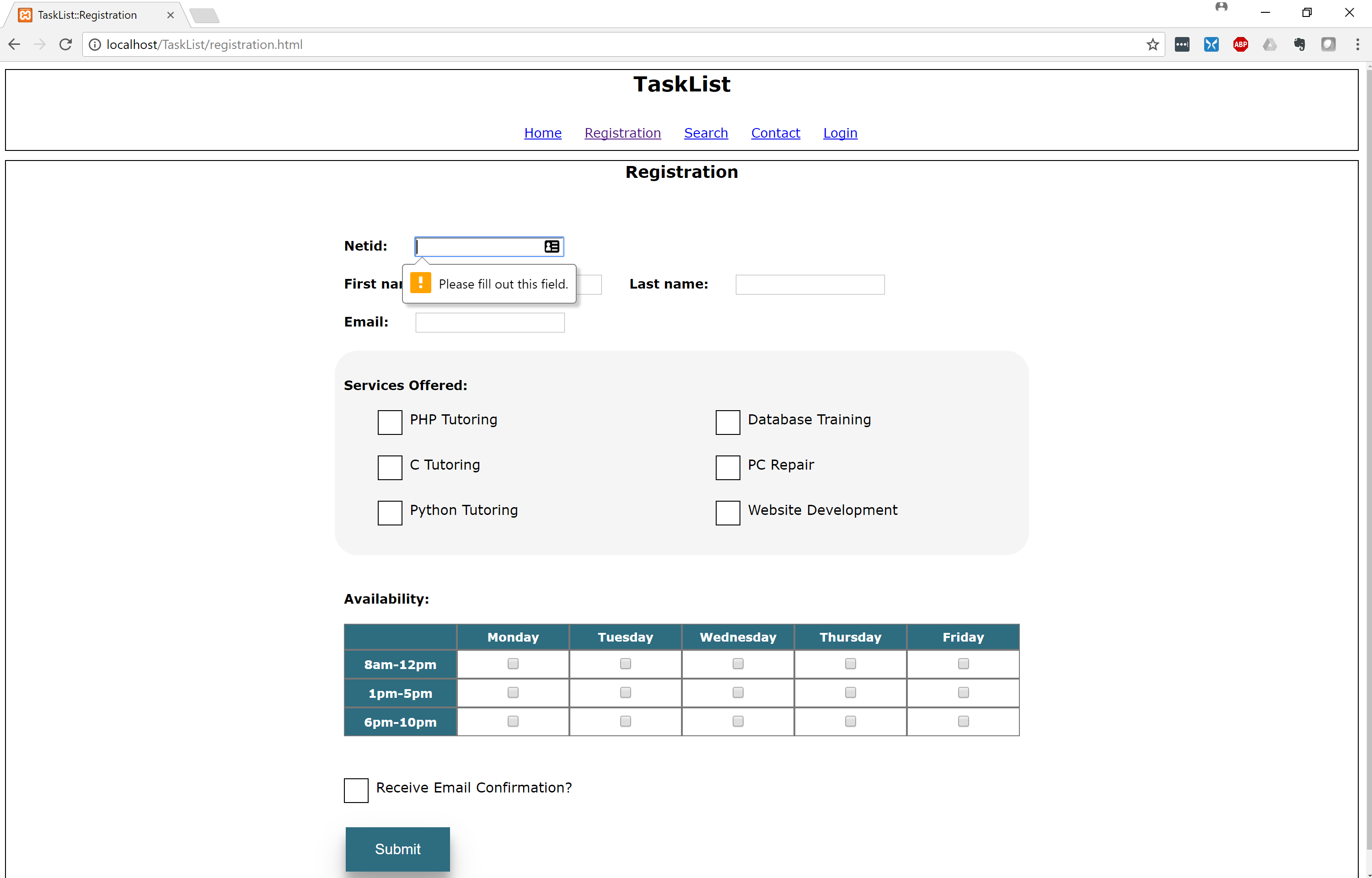
Fields empty. I used **required** for the first fields:

<div>Netid:</div>

<div>

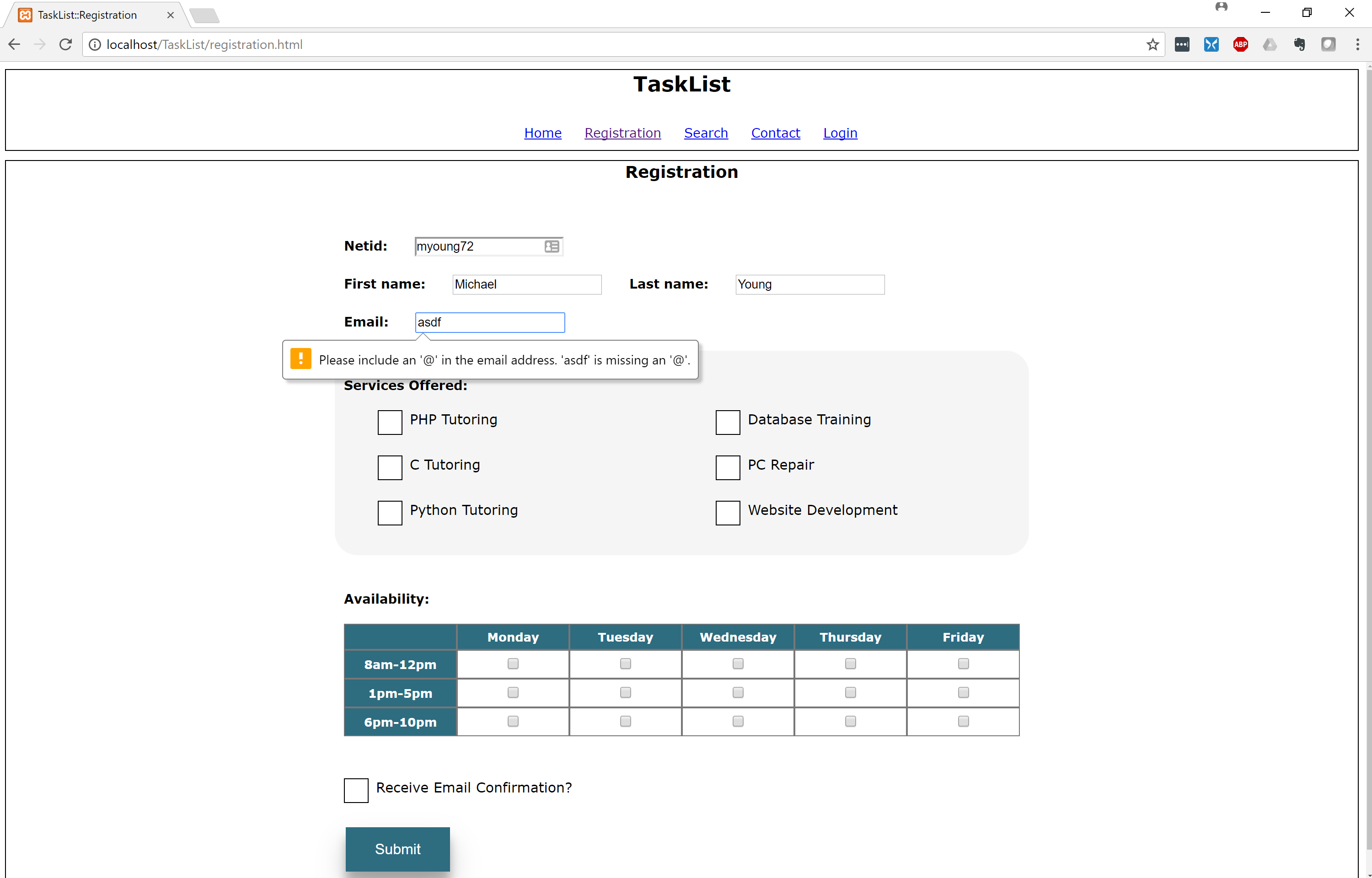
<input type="text" name="netid" required>

</div>



Email not in right format. I used input type = email for the check:

<input type="email" name="email" required>



No Services Selected. I used JavaScript functions for this check:

<form name="myForm" action="registration\_submit.php" onsubmit="return validate(this);" method="post" >

function validate\_services(form) {

var serv = document.getElementsByName("services[]");

var checkCount = 0;

for(i=0; i < serv.length; i++) {

if(serv[i].checked) {

checkCount++;

}

}

if (checkCount < 1) {

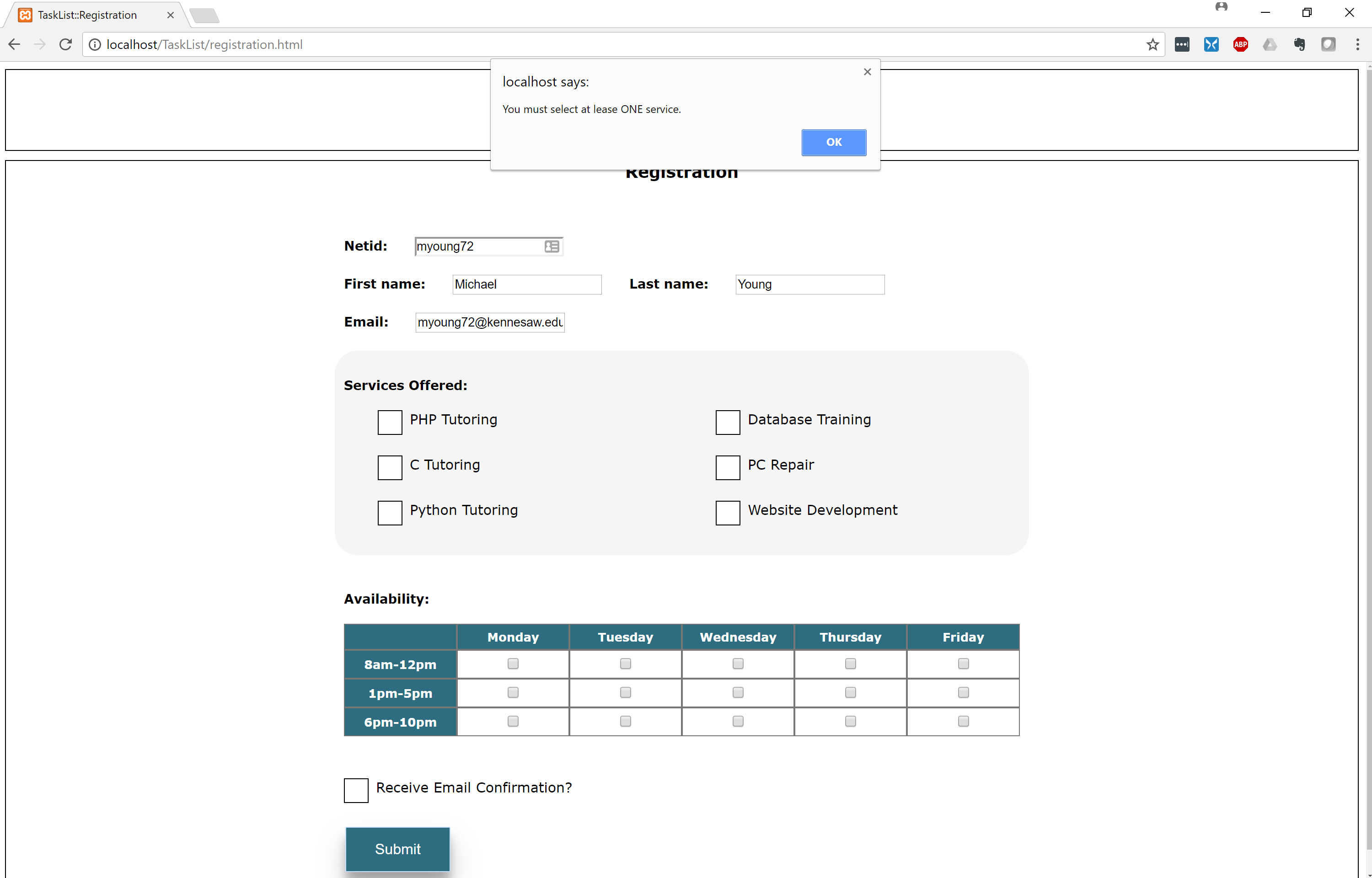
alert('You must select at lease ONE service.');

return false;

}

return true;

}



No time slot selected. I used JavaScript function for this check as well:

function validate\_times(form) {

var time = document.getElementsByName("times[]");

var checkCount = 0;

for(i=0; i < time.length; i++) {

if(time[i].checked) {

checkCount++;

}

}

if (checkCount < 1) {

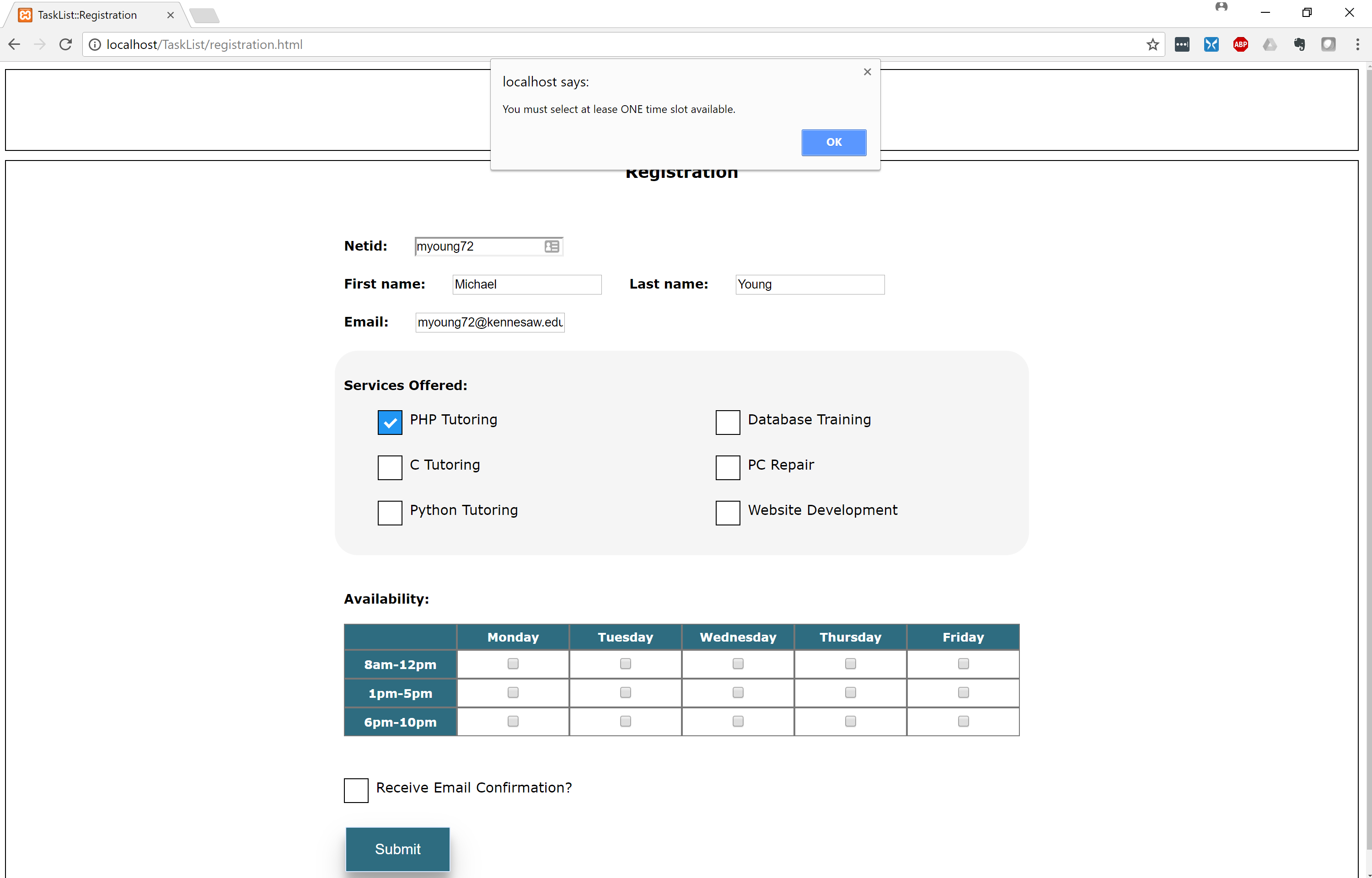
alert('You must select at lease ONE time slot available.');

return false;

}

return true;

}



Call both functions on submit:

function validate(form) {

if (validate\_services(form) == false || validate\_times(form) == false ) {

return false;

}

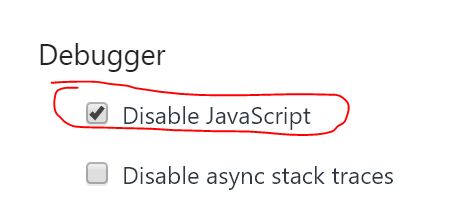
return true;

}

</script>

### **Server Side Validation**

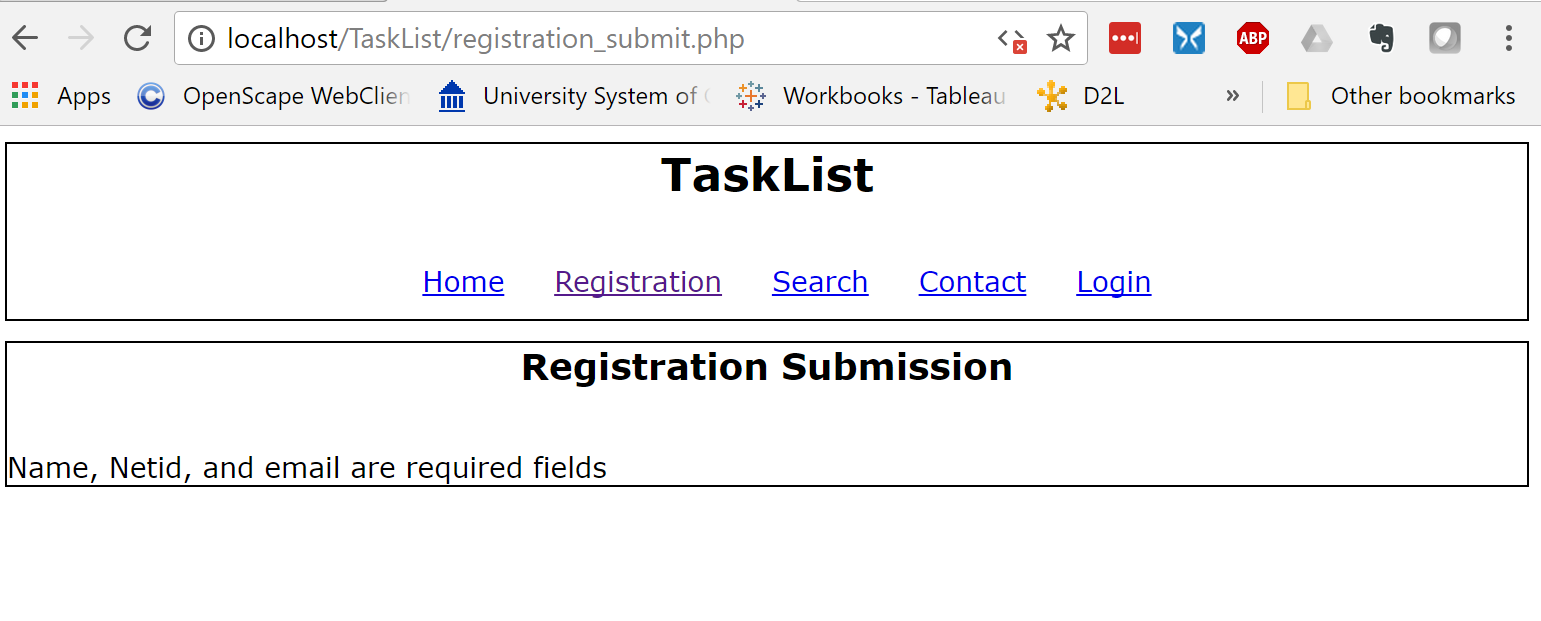
\*\* Note: For this example, I will temporarily disable JavaScript in order to test server-side\*\*



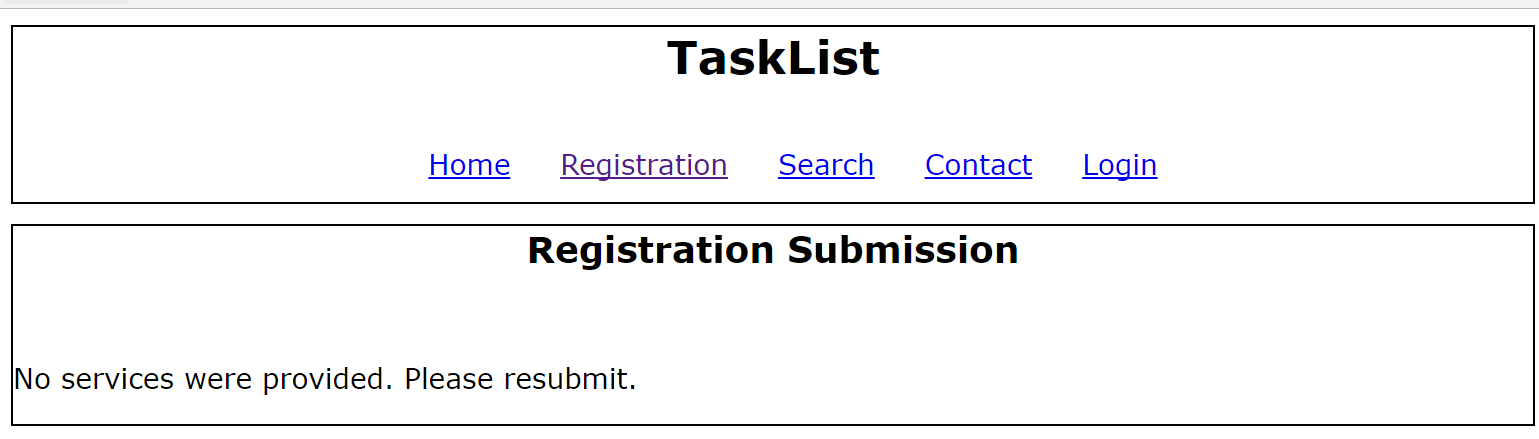
And temporarily add novalidate to the form submission:



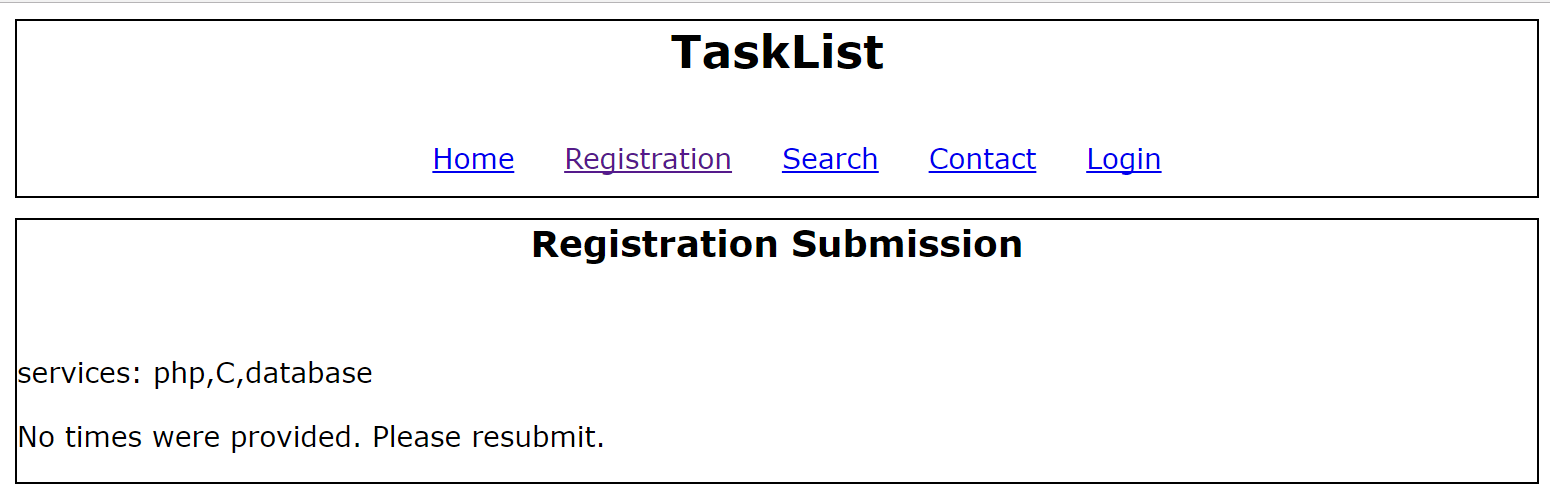
**Empty Text Fields:**



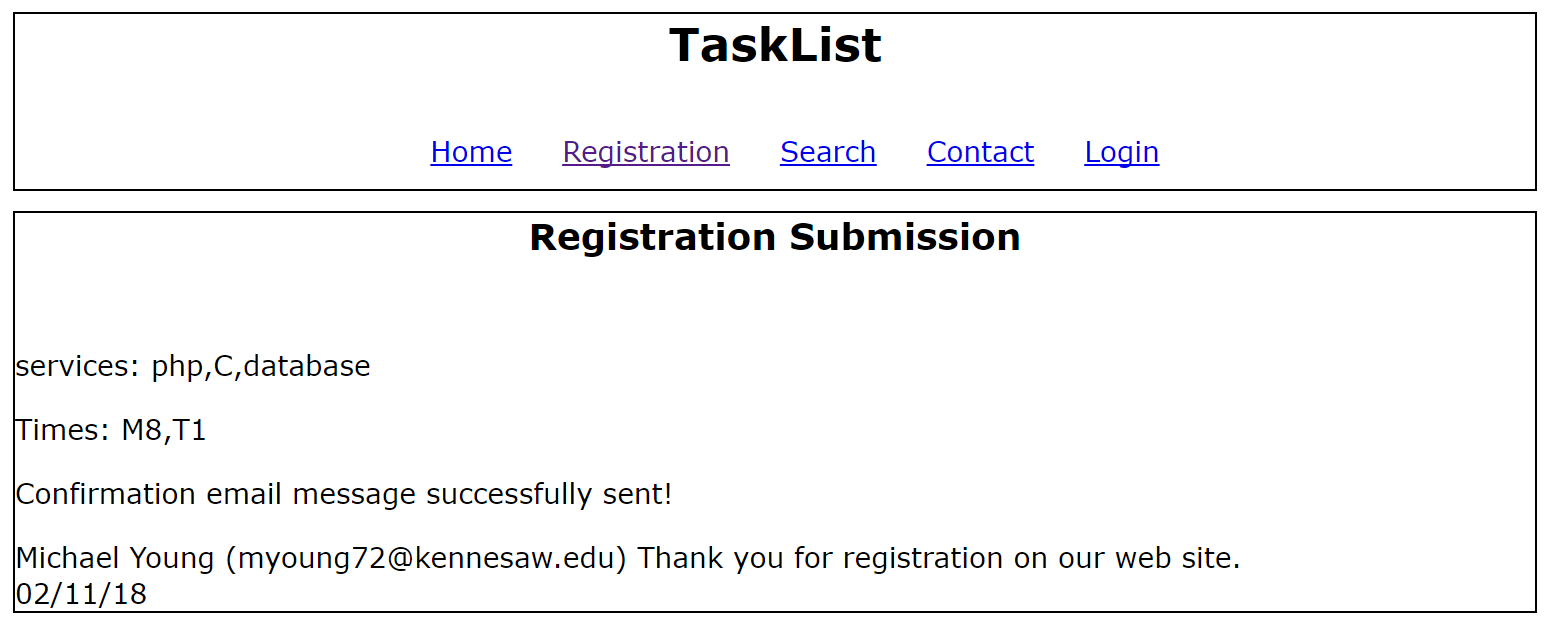
**No services checked:**



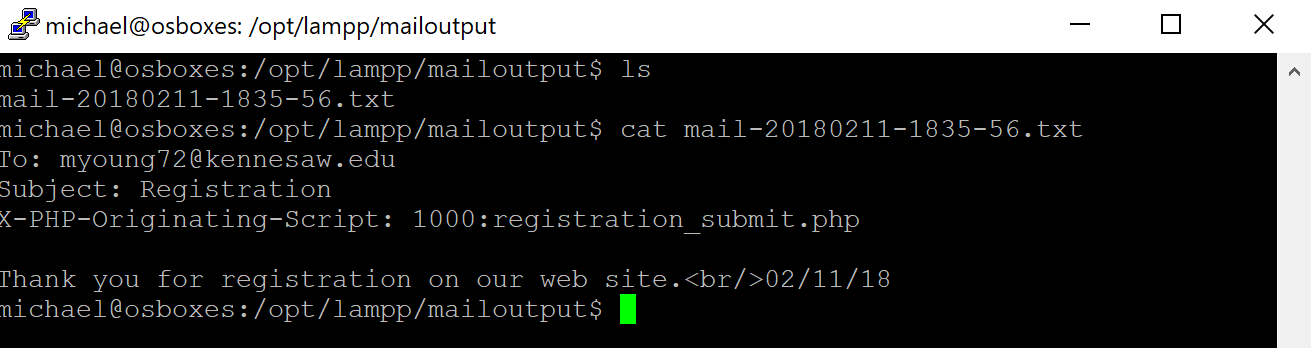
**No Availability checked:**



**Successful submit and email confirmation:**



To check mail output, you will need to check your mail output folder that is set in your php.ini file:



**Menu:**

<div class="menu">

<nav>

<a href="index.php">Home</a>

<a href="registration.php">Registration</a>

<a href="search.php">Search</a>

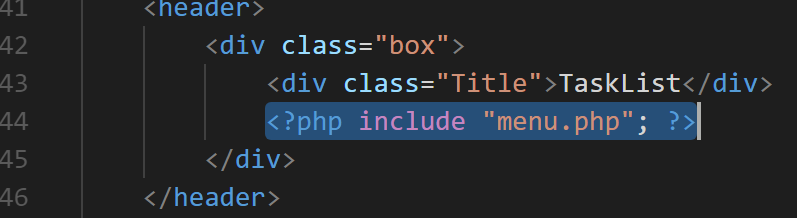
<a href="#">Contact</a>

<a href="#">Login</a>

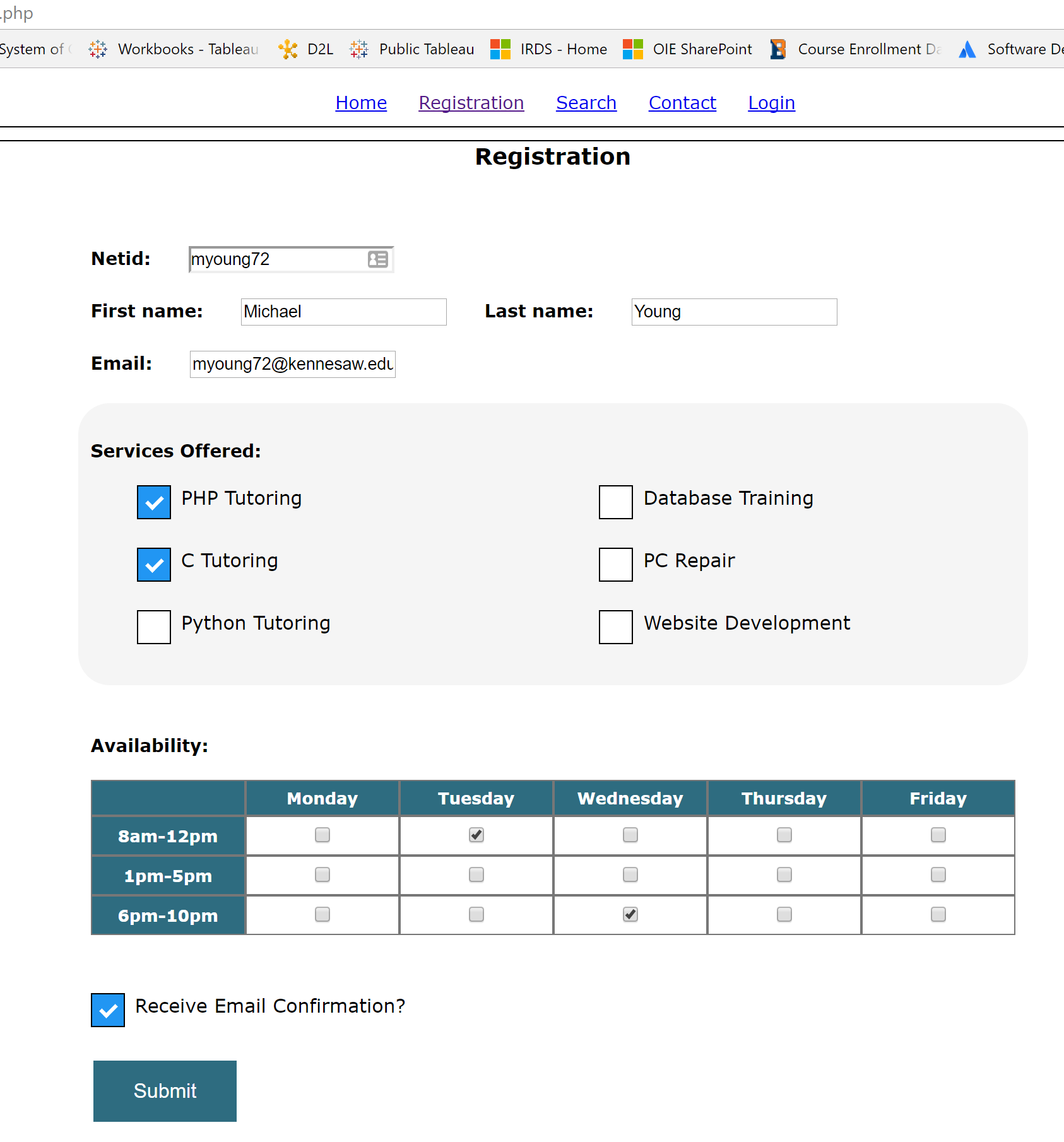
</nav>

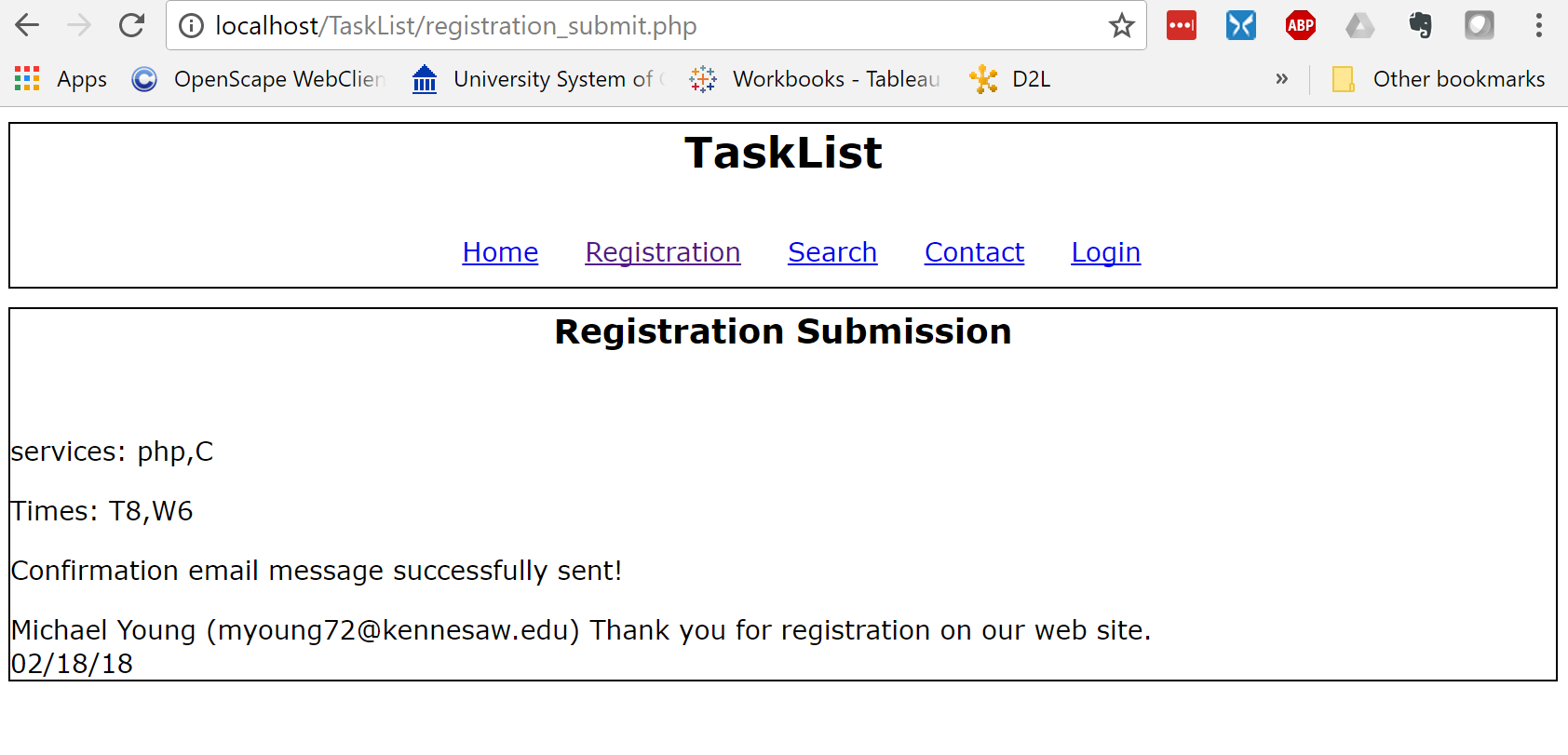
</div>

**Include:**



**Functionality and Database Screenshots**

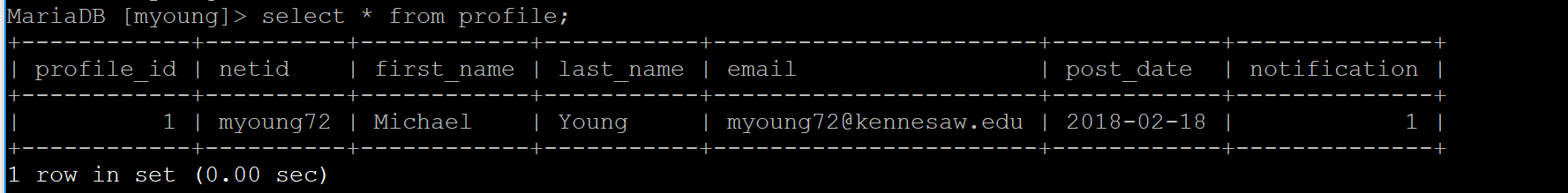


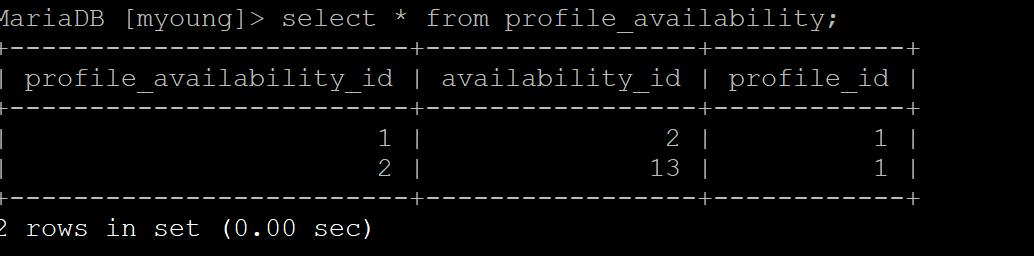


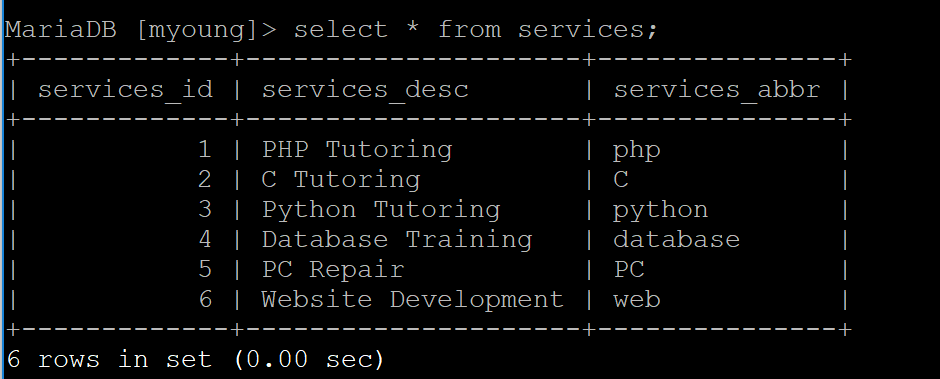
**Availability**

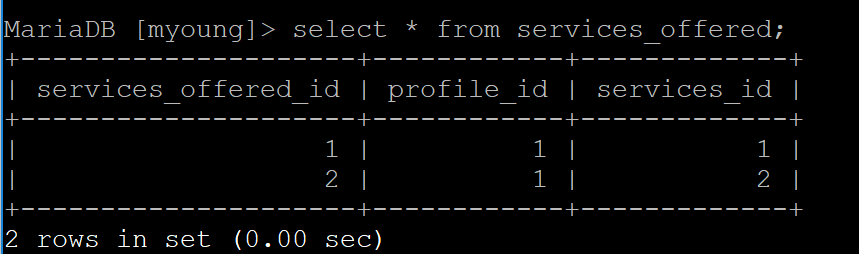


**Profile**



**Profile Availability**

**Services**

**Services Offered**

**Search:**

$query = mysqli\_prepare(

$conn,

"select a.first\_name, a.last\_name, availability\_desc

from profile a, profile\_availability b, services c, services\_offered d, availability e

where c.services\_id = d.services\_id

and d.profile\_id = a.profile\_id

and e.availability\_id = b.availability\_id

and b.profile\_id = a.profile\_id

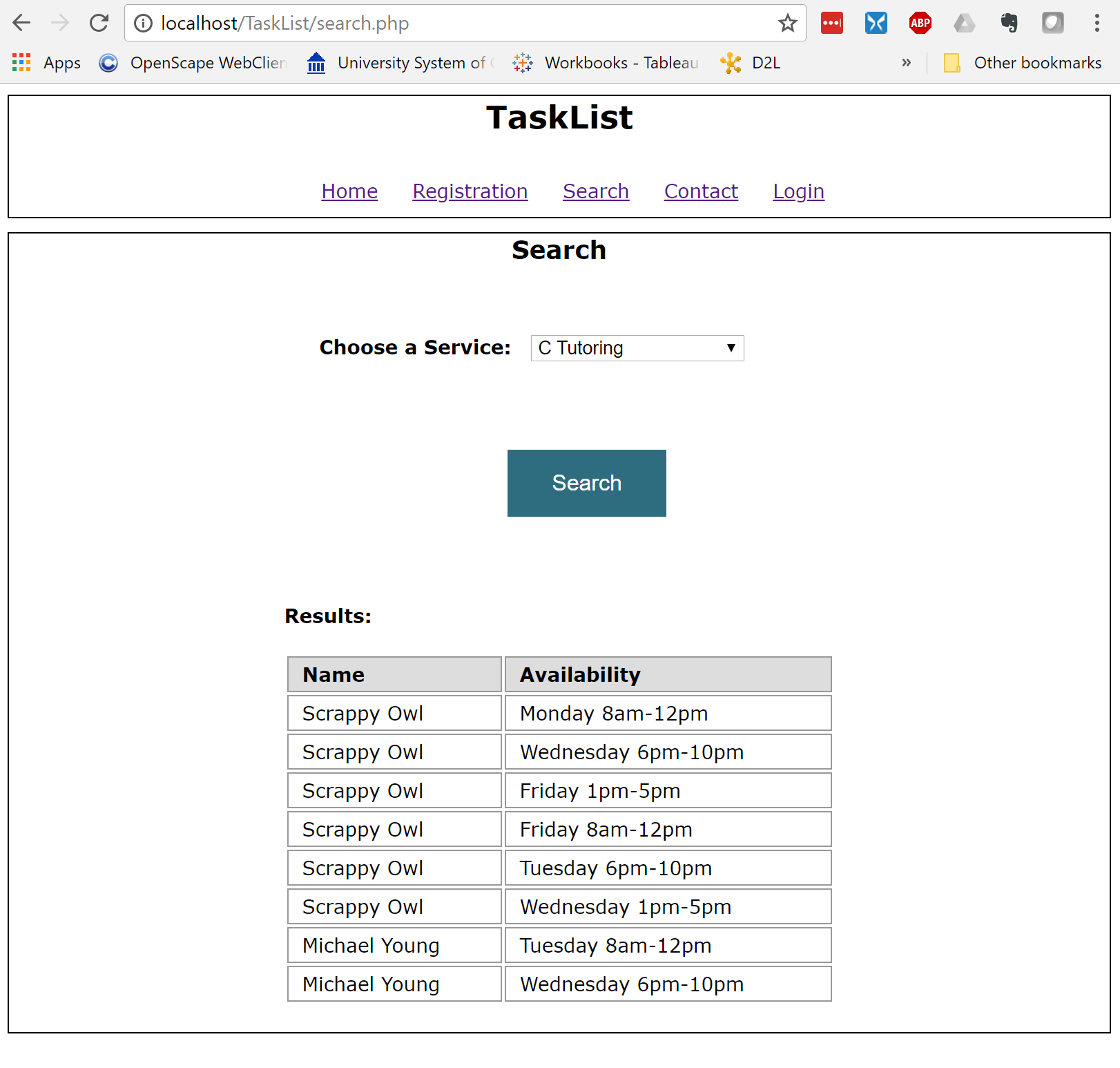
and c.services\_abbr= ?

order by last\_name"

)

**Bind:**

mysqli\_stmt\_bind\_param ($query, "s", $services);



# Conclusion

This project met the requirements for:

1. The site must have a navigation menu.

2. User registration page:

a. KSU NetID

b. First Name

c. Last Name

d. Email

e. Services Offered (the list of services on the form for user to choose from has to come

from a prepopulated “services” table in the database)

f. Availability (the date and time slots that the user is available to provide the above

services to the CSE community)

g. Email Confirmation (ask if the user would like to receive an email confirmation)

h. DO NOT ask for a PASSWORD. This feature will be implemented in the group project

i. Once the Register button is clicked, the user data will be written into a “profiles” table

along with any additional tables depending on the design

3. A search page to find users who can provide a service. The search results should come from the “profiles” table along with any additional tables depending on the design.

Feel free to email with any questions: [southernmiss@gmail.com](mailto:southernmiss@gmail.com)