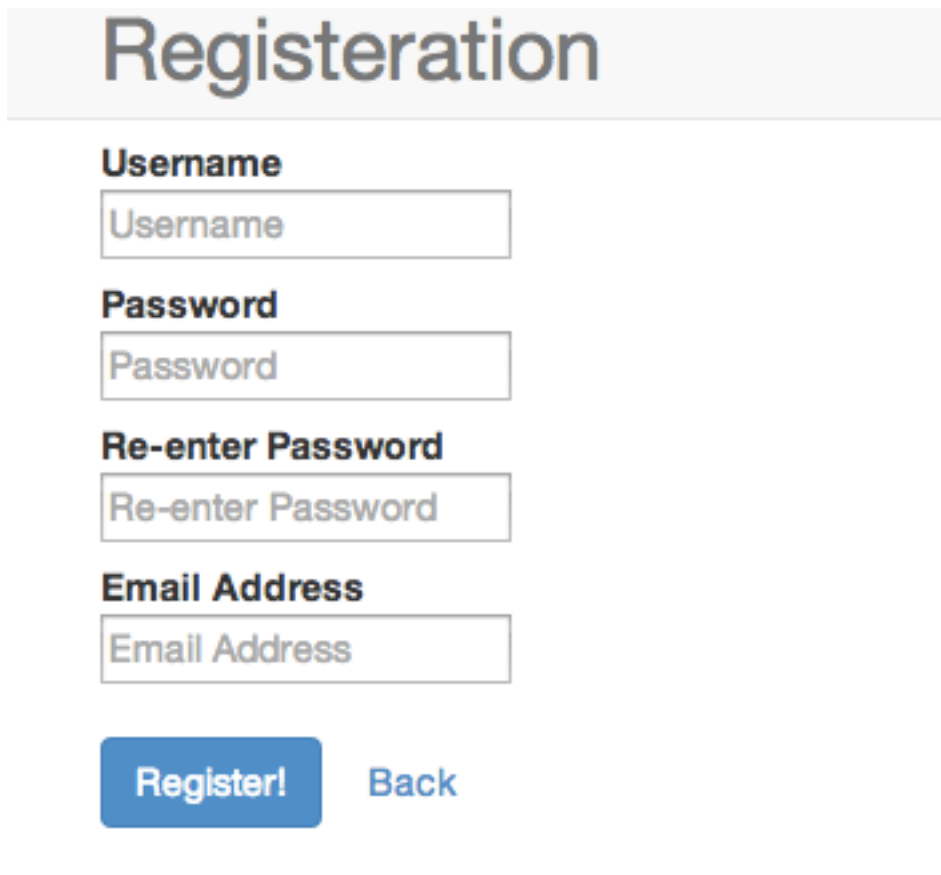


Final Project Manual Test-Plan

Test 1:

In our registration page, we allow users to fill out their username, password and email and register to the website. If the user submits invalid input, inline error messages are shown. On submitting valid input, the users are now allowed to login to the website.



The image shows a registration form with a light gray header containing the word "Registration" in a large, bold, dark gray font. Below the header, the form consists of four labeled input fields: "Username", "Password", "Re-enter Password", and "Email Address". Each label is in a bold, dark gray font, and the corresponding input field is a white rectangle with a thin gray border. The placeholder text for each field is the same as the label, in a lighter gray font. At the bottom of the form, there is a blue button with the text "Register!" in white, and a blue link with the text "Back" in white. A thin horizontal line is positioned below the "Back" link.

Registration

Username
Username

Password
Password

Re-enter Password
Re-enter Password

Email Address
Email Address

Register! [Back](#)

When the user submits a blank form, the registration page should print out inline error messages for each of the inputs:

Registration

Username

Please enter a valid username

Password

Please enter a password

Re-enter Password

Please reenter the password

Email Address

Please enter Email Address

Register!

[Back](#)

Test 2:

Incase the user enters invalid input, say text instead of email address, the page should still display an error message next to the email address.

Ex:

Registration

Username

Password

Re-enter Password

Email Address

Register!

[Back](#)

After clicking “Register” the page should display an error message for the email address:

Registration

Username

Password

Re-enter Password

Email Address

Please enter a valid Email Address

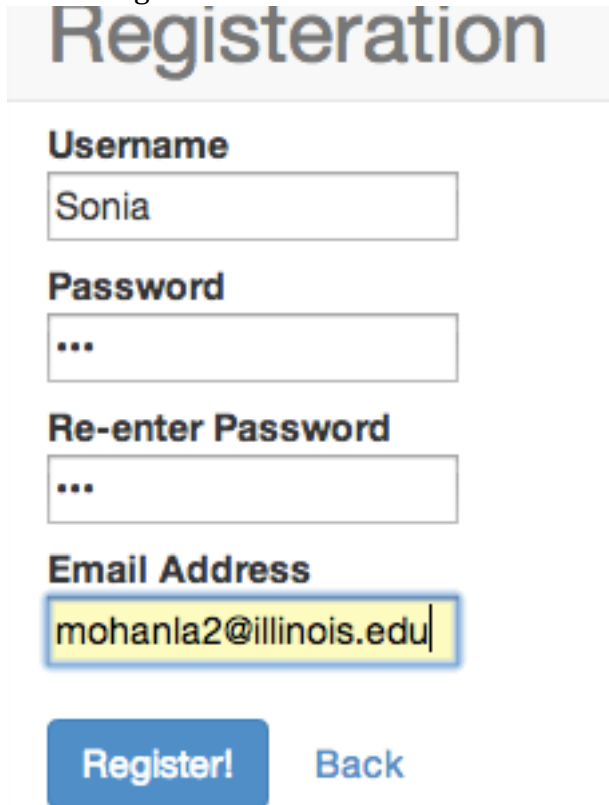
Register!

[Back](#)

Test 3:

If the user puts in valid values for all the textboxes, the user should be redirected to the main page and registered to the website!

Before registration:



The registration form is titled "Registration" in a large, bold, blue font. Below the title, there are four input fields: "Username" with the value "Sonia", "Password" with three dots, "Re-enter Password" with three dots, and "Email Address" with the value "mohanla2@illinois.edu". The email field is highlighted with a yellow background and a blue border. At the bottom, there are two buttons: a blue "Register!" button and a grey "Back" button.

Registration

Username
Sonia

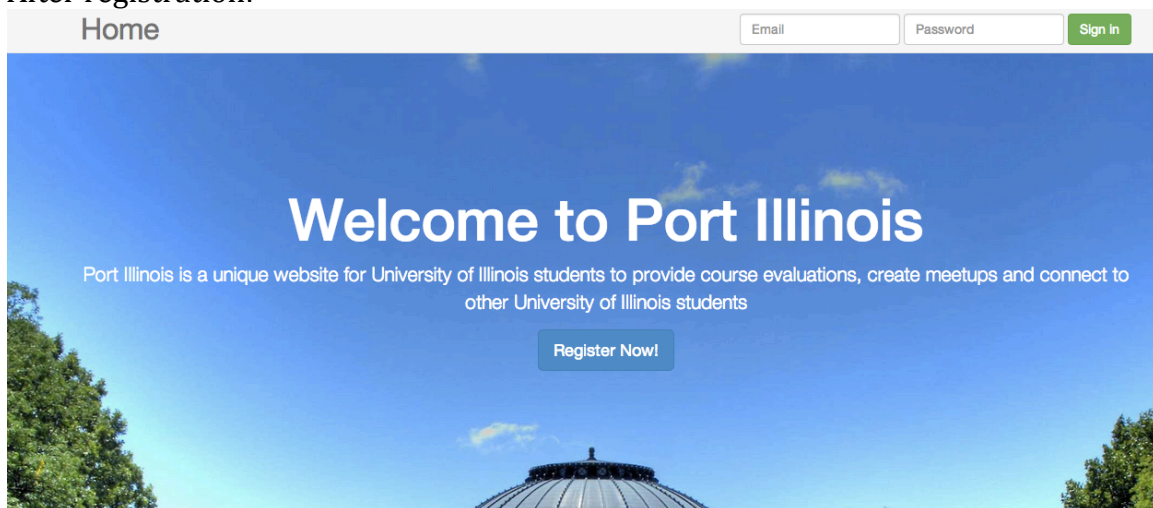
Password
...

Re-enter Password
...

Email Address
mohanla2@illinois.edu

Register! **Back**

After registration:



The home page features a navigation bar with a "Home" link, an "Email" input field, a "Password" input field, and a green "Sign In" button. The main content area has a blue background with a large white heading "Welcome to Port Illinois". Below the heading, there is a paragraph of text: "Port Illinois is a unique website for University of Illinois students to provide course evaluations, create meetups and connect to other University of Illinois students". A blue "Register Now!" button is centered below the text. The background image shows a blue sky with a building dome and trees.

Home **Sign In**

Welcome to Port Illinois

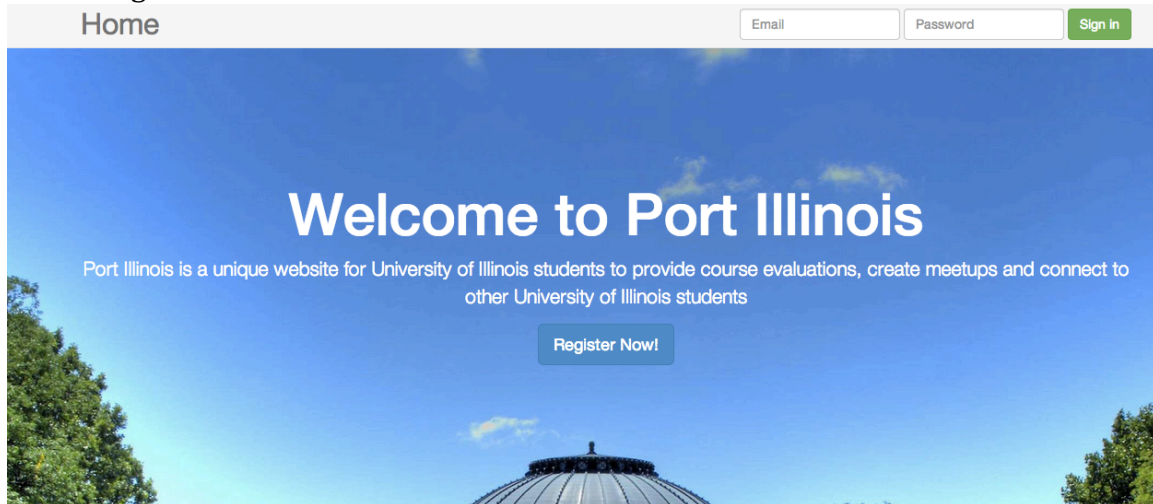
Port Illinois is a unique website for University of Illinois students to provide course evaluations, create meetups and connect to other University of Illinois students

Register Now!

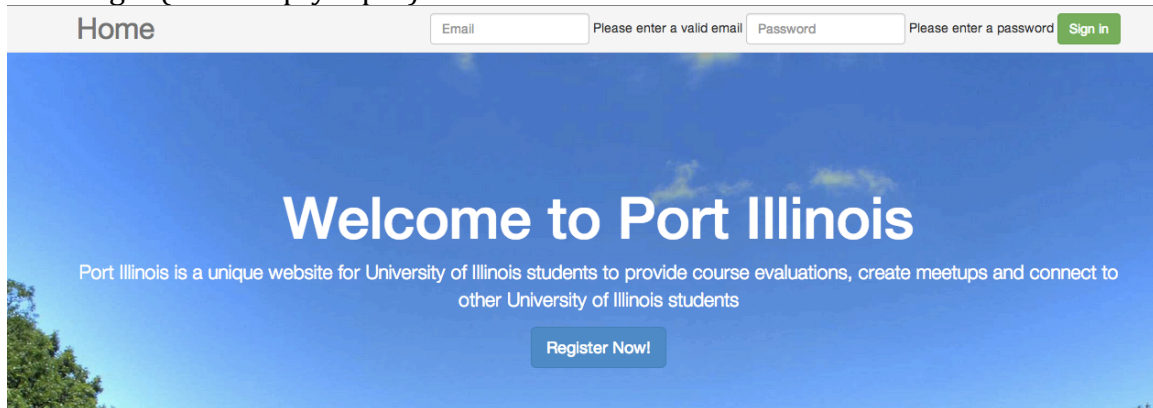
Test 4:

On our main home page, when a user inputs either empty or incorrect email/password combination and clicks the signin button, the webpage should display an error message:

Before login:



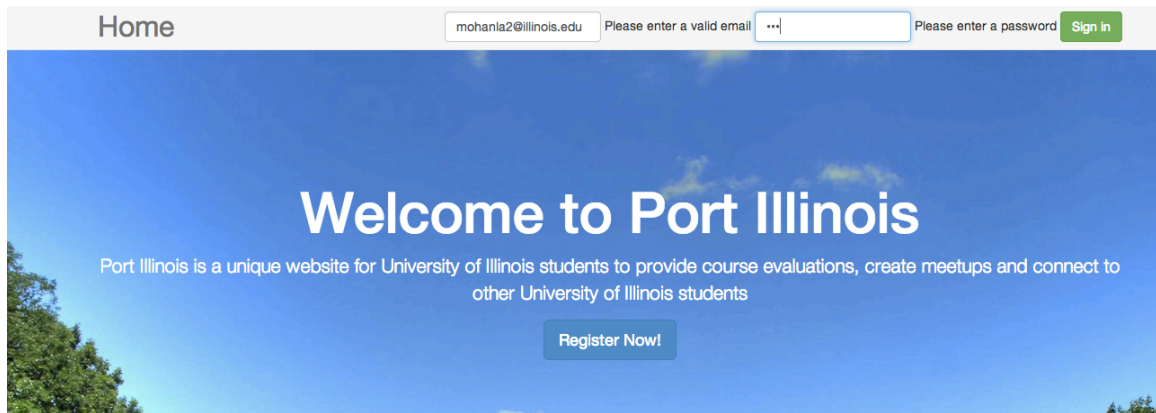
After login (with empty input):



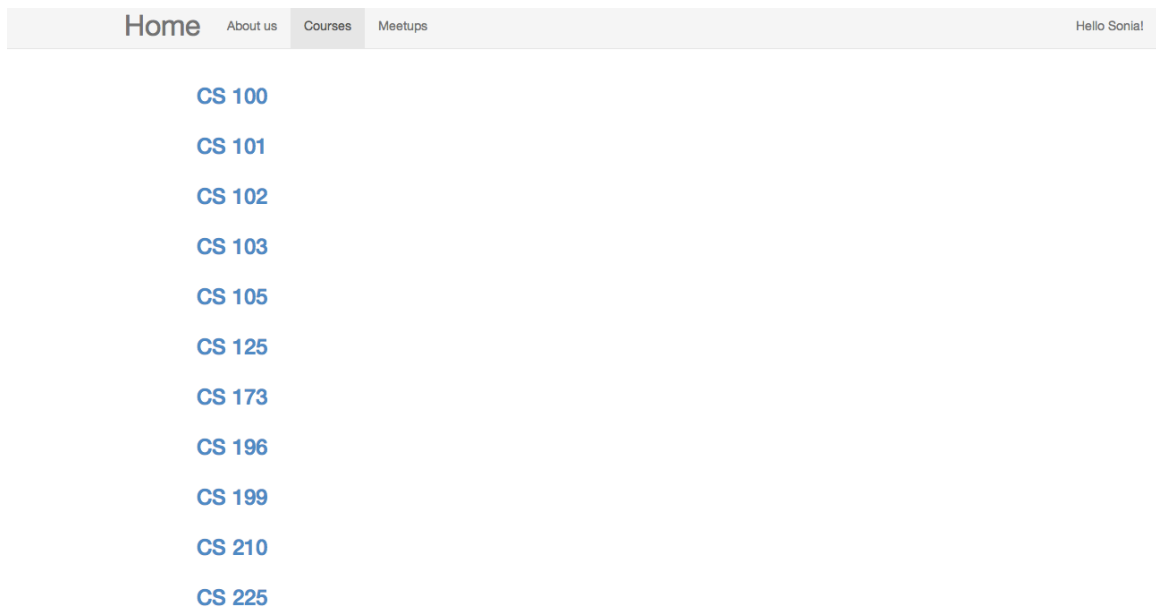
Test5:

If the user enters correct login-info and clicks the sign-in button, he should be redirected to the courses page with a header on the top welcoming him!

Before pressing "sign in":



After pressing sign-in:



Test 6:

On the main homepage, when you click on the “view courses” button, you get redirected to a page containing the list of all CS classes:

Before pressing the “View courses” button:

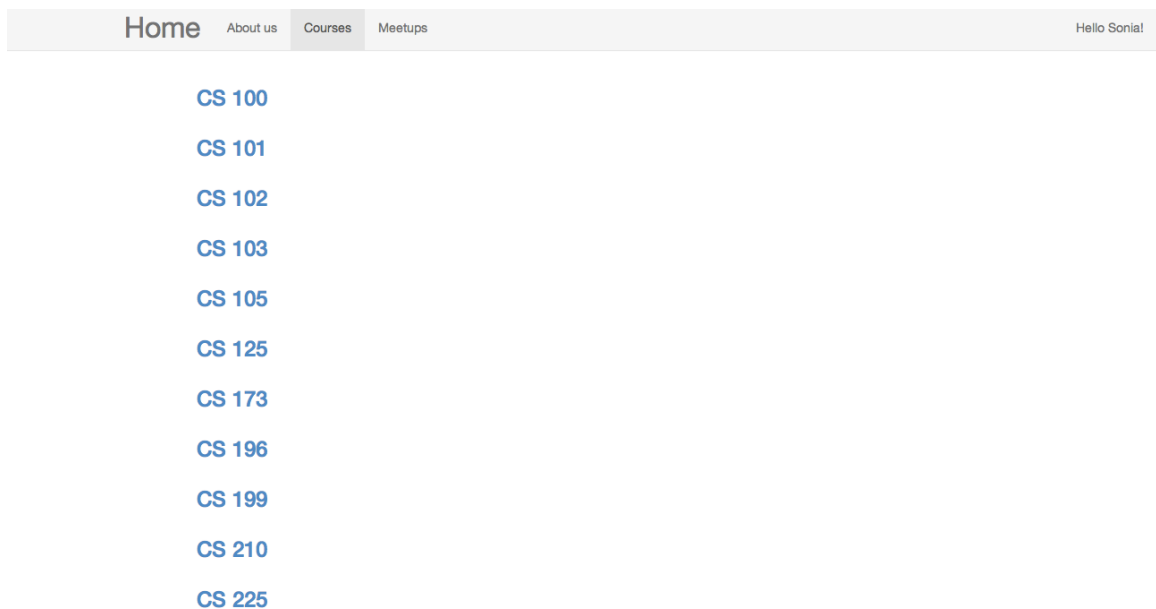
Courses

Browse and review the classes offered in the Department of Computer Science. Get course information and reviews posted by your peers to get feedbacks on the course. Don't forget to add your own reviews!!!

[View courses »](#)

[Update courses »](#)

After pressing the “view courses” button:

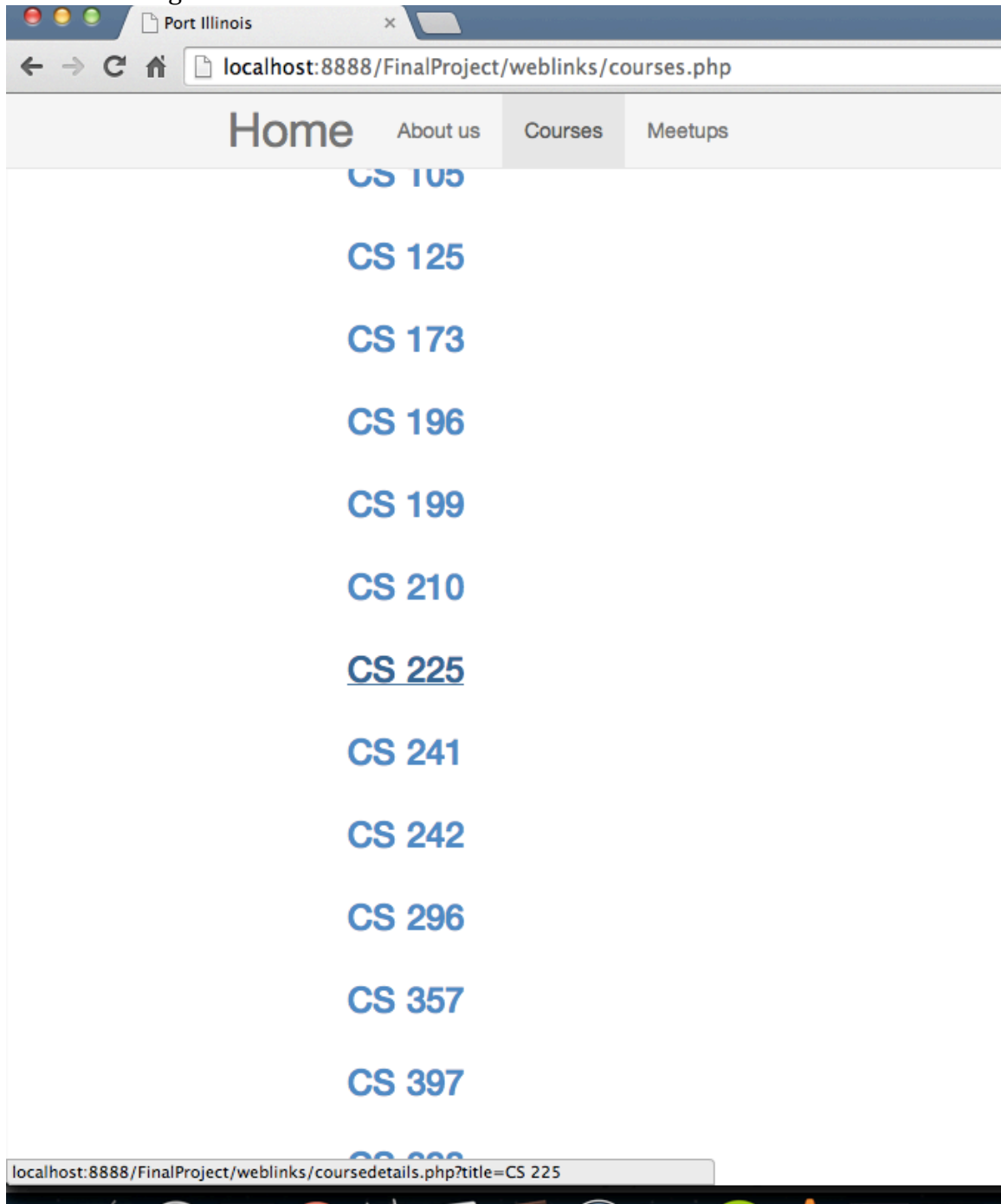


Test7:

Each course is a link, which when clicked, should display 3 things:

- Course Title
- Course Description
- Course Credit Hour
- An “add review” section

Before clicking “CS 225”:




After clicking “CS 225”:

CS 225

Data Structures

Credit: 4 hours. Data abstractions: elementary data structures (lists, stacks, queues, and trees) and their implementation using an object-oriented programming language. Solutions to a variety of computational problems such as search on graphs and trees. Elementary analysis of algorithms. Prerequisite: CS 125 or ECE 190; CS 173 or MATH 213.

Add a Review

Professor name:	<input type="text"/>
TA:	<input type="text"/>
Semester	<input type="text"/>
Expected Grade/Grade:	<div>Select option...</div>
Part of:	<input type="radio"/> College Core <input type="radio"/> Major Requirement <input type="radio"/> Elective

Qualitative Reviews

Difficulty:	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Time Commitment:	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5