RentZen – The week 08 challenge (the final challenge document)

This is the last challenge document for the MIS3502 class project. By this time, you should have completed all the previously assigned challenges, and you should also have acquired some competence in using the MVC pattern to develop your application.

The goal of this week 08 challenge is simple… finish the project. That means that your top priority now is to complete the server-side scripting necessary to provide the minimum viable RentZen product. With that in place, we can move on to other things.

In the weeks ahead we will build on top of our MVC solution. We will add in some improved form validation (using JavaScript / jQuery) and some advanced features that will be accomplished by making Ajax calls. Much of that activity (JavaScript/jQuery/Ajax) will be covered in class. But in order to make the most benefit from the in-class activity, you will need to first complete as much of the MVC part of the project as you can.

Your instructor's expectations are documented on the next page – the RentZen Rubric. At this point in the semester you should possess all the skills necessary to complete the "Part 1" required tasks. This is what you should be working on now.

Helps and Hints,

1. To assist you in the project, our web designer (Nancy) completed **all the view** files. They are provided to you in the rentzen_week08.zip file. Nancy also went so far as to put some PHP code into her views… this means that **most of the coding remaining to be done for "Part 1" will take place in the controller (index.php) files of your project.**

2. Personally, I think Nancy has done some fine work here. She deserves a raise! However, if you want to do things slightly differently than the way Nancy did, or if you want to change/adapt/rename/delete one or more of the files she created, that's fine. Just make sure that you achieve the functionality described in the rubric. Side note, the exception to this is **property_add.php** and **property_view.php** … you will want to use those "as is" later in the semester!

3. A working copy of RentZen can be found here: https://misdemo.temple.edu/fall901/rentzen

   Feel free to sign up, sign in, and try it out. This example should give you a vision for what a good solution to the "part 1" tasks would look like. Give it a try. Please note, this version was implemented with nothing but server-side scripting and Bootstrap.

   Also, please be advised that I will periodically flush and rebuild the fall901 database, returning it to its initial state.

4. There is an opportunity to earn some PRO points for the project. See the rubric.

5. A good place to start would be write the code necessary to see the landlord dashboard (person_menu_landlord.php) and the renter dashboard (person_menu_renter.php). Tackle each link, one at time. Don't forget the links that appear in the header and the footer. Hint: Even though they are technically optional, it might be smart to start with the "help" and "about" links.

6. Nancy wanted to add a few custom css styles. So, there is a new file, rentzen2.css, found in the css folder.

The due date for the RentZen project is 11:59 PM on Friday 4/26. On the day of 4/26 (not before, not after) I expect your team to email me (jeremy@temple.edu) a link to your project as found on the class server. If you do not email me, I will assume that your project URL is https://misdemo.temple.edu/YOUR_TEAM_NAME/rentzen.

RentZen Grading Rubric

The fundamental questions behind the items is this rubric are as follows: Does the final RentZen solution provided by the student demonstrate knowledge of all three parts of the course: MVC, PDO, Bootstrap, JavaScript / jQuery, Ajax, APIs? Did the student team complete the challenges in such a way that a minimum viable product was created that was (at least) on par with the working example provided by the instructor? Did the student team invest a sustained effort in the project over time?

Required Tasks (Part 1)

Item	Point
Are the model files and model test pages complete?	1
Do they run without error?	1
Does the solution show working knowledge of Bootstrap? The sample views, provided by the instructor, have been used effectively. No glaring layout problems. Solution is responsive.	1
Does the solution show a working knowledge of MVC? Are all client-initiated requests consistently run through a controller (index.php) or, in the case of Ajax, a service call?	1
Do the contents of the navigation bar icon ☰ change appropriately when the user is signed in / signed out? Hint! Use the $_SESSION variables and the header.php file to do this.	1
Is Landlord functionality complete?	
Can a user sign in / sign out as a landlord?	1
Can a user sign up as a landlord?	1
Can a landlord list rental applications and approve / reject renters?	1
Can a landlord add a property?	1
Can a landlord manage properties?	
Can a landlord see a list of properties managed by his/herself?	1
Mark properties as “listed” (status 403) or Occupied (status 402) with a single button click	1
View the property information by clicking on a view button / link?	1
Note that the edit property feature is *not* required. (SEE PRO POINTS)	NA
Is Renter functionality complete?	
Can a user sign in / sign out as a renter?	1
Can a user sign up as a renter?	1
Can a renter see a list of rental properties that are marked as “listed” (Status 402)	1
Can a renter view the details of a specific property?	1
Can a renter apply to rent a property?	1
Can a renter see a list of their rental applications and their status?	1
Total points	18

Required tasks (part 2)

Item	Point
All input pages have rudimentary, server-side, validation.	1
The Landlord’s add property view has JavaScript / jQuery scripting (client-side) to provide detailed error messages to users and prevent the form post event when required data is missing or invalid. *****	3
The Renter’s rental application view has JavaScript / jQuery scripting (client-side) to provide detailed error messages to users and prevent the form post event when required data is missing or invalid.	3
Total Points	7

Required tasks (part 3)

Item	Point
An application-specific Ajax service is used to generate a list of state codes on the add a property page. *****	3
When a renter views a property, he/she can see an image of Google Street view. *****	3
When a renter views a property, he/she can see a Google map pinpointing the property's location	3
~~When a landlord adds a property, capture the properties latitude and longitude in the database. Use an API to determine the latitude and longitude based on the address. * –~~ **THREE "FREE" POINTS**	3
Total Points	12

Optional tasks / PRO points

Teams that complete all the required tasks above to the instructor's satisfaction are eligible to make the following improvements to earn PRO points. **These optional tasks are also due by 11:59 PM Friday 4/26.**

Item
1. Allow Landlords to edit a property by clicking on the "edit" button provided in the property list.
2. Allow renters to complete their renter profile by clicking on "Update Profile" as found in the renter dashboard.
3. Allow landlords to complete their landlord profile by clicking on "Update Profile" as found in the landlord dashboard.
4. When a landlord adds a property, permit the landlord to upload a photo to Flickr. Store the photo URL in the database. Use the URL in place of no_image.png so that pictures of the properties are shown to the renters. *****
~~5. Use a messaging API to send a text an SMS text message to a renter when their application is approved by a Landlord. (A messaging API will be provided to you)~~
6. The Help and About us links work and are directed to appropriate locations.

If all the above items are completed in a satisfactory manner, each team member will earn 50 PRO points. This is an all-or-nothing PRO point award. Either all six tasks are done, or not.

* These items will be completed either all or in part during class.