Web_Lesson6: REST APIs with Angular

Please do not forget to submit your feedback after the class. This feedback helps a lot in increasing the effectiveness of the course. Use the related Canvas survey to submit your ICP # and feedback

Lesson Overview:

In this lesson, we are going to discuss routers, services, HTTP, and RESTful APIs in Angular

Use Case Description:

Wikipedia Viewer: A sample webpage which will extract data from Wikipedia and display it to the user based on the request with the help of REST calls

Programming elements:

Angular (routers, services, HTTP, and RESTful APIs)

Source Code:

https://umkc.box.com/s/senkugg3z37gb74n5qw2p4kak5jd1wje

In Class Programming (ICP):

There is no limit on details and creativity. You can add as many details as you can and create an attractive and interactive web page.

1. Create an application in Angular which displays nearby restaurants (Hint: Use Foursquare API)
2. Create an application in Angular which displays recipes (Hint: Use EDAMAM API)

If possible, combine both the APIs in a single application. (Given starter code has both together. Change the UI if you are going to use this code)

Note: You must use the **Angular (not the AngularJS)**

ICP Submission Guidelines

1. ICP submission is an individual contribution
2. Submit your source code and documentation to GitHub and represent the work through the wiki page accurately (submit your screenshots as well. The screenshot should have both the code and the output)
3. Comment your code appropriately
4. Video submission (3 to 5 min video showing the demo of the ICP, with brief voiceover on the code explanation)
5. Submission after the due date is considered as a late submission. (Check the 'Late Submission Policy on Assignments' in the syllabus)
6. Use the related Canvas survey to submit your ICP # and feedback

ICP Rubric Details

You can find ICP Rubric Details in both the Syllabus and Canvas ICP assignment.

Criteria	Novice	Competent	Proficient
Wiki page (25)	Basic wiki page. (>=0 to <=5)	Wiki page with the required details. (>5 to <=15)	Wiki page with all details and making it easy to follow and understand. Visually looking good. (>15 to <=25)
Video (25)	Basic video. (>=0 to <=5)	Video with the required details. (>5 to <=15)	Video with all details and making it easy to follow and understand. Annotated with the subtitles. (>15 to <=25)
Completeness of given assignment (25)	It is partially solved. (>=0 to <=5)	Completely solved. (>5 to <=15)	It is solved efficiently. (>15 to <=25)
Code Quality (It is relative) (10)	Refer to the best coding practices page. (>=0 to <=5)	Refer to the best coding practices page. (>5 to <=8)	Refer to the best coding practices page. (>8 to <=10)
Commenting the code (10)	Not useful comments. (>=0 to <=5)	Slightly appropriate comments. (>5 to <=8)	Appropriate comments. (>8 to <=10)
Time of submission	Submission after the due date. Check the 'Late Submission Policy on Assignments' section in the syllabus	Submission on the deadline. No score will deduct from the obtained score.	Submission before the deadline. No score will deduct from the obtained score.
Submission (including feedback) (5)	Submission with partial details. (>=0 to <=3)	Submission with the essential details. (>3 to <=4)	Submission with all the details. (>4 to <=5)
Total	**Minimum = 0**		**Maximum = 100**

Note: Cheating, plagiarism, disruptive behavior, and other forms of unacceptable conduct are subject to strong sanctions under university policy. See detailed description of university policy at the following URL: https://catalog.umkc.edu/special-notices/academic-honesty/