

W

Account

Dashboard

Courses

Calendar

Inbox

Help

←

FA19 COMPSCI 639 002 > Assignments > 09. React-Native 3

Fall 2019-2020

Home

Announcements

Assignments

Piazza

Grades

Pages

Files

09. React-Native 3

Due Nov 25, 2019 by 11:59pm Points 6 Submitting a text entry box or a website url
Available Nov 16, 2019 at 12pm - Dec 9, 2019 at 11:59pm 24 days

This assignment was locked Dec 9, 2019 at 11:59pm.

Homework: React-Native 3 (6 Points + 1 Point EC)

This assignment is meant to allow you to add the final bits of functionality to your application, as well as a bit of finesse and polish to your app. You also have the option to earn a point of EXTRA CREDIT for implementing one of your accessible designs from the design homework.

You can generate your empty repository [HERE](#) . You should move your code from React Native 2 into this empty repository, and build from there.

Potential Gotchas/Helpful Suggestions:

- We highly suggest reviewing the rubric and specification below. Information on using the API can be found in your project README.
- A list of approved third-party NPM packages is curated [HERE](#) . If you find a package you would like added, send a message to one of the TA's on Teams or Piazza to request permission. Assignments using unauthorized packages may receive scores of zero.
- If you install third-party packages, make sure to run `npm install --save [package name]` , so as to make sure you update the package.json.
- Double-check that the code you have submitted builds correctly. One way to do this is to clone the repository in a new folder after committing/pushing your working code, and running `npm install` and then `npm start` in the new repository.
- Compare the components and navigation patterns (if applicable) that you made in React Native 2 to the requirements in React Native 3. There are a lot of similarities in these patterns. This is intentional. Use this to your advantage as you build out React Native 3. If you do, you will realize that you should be able to spend a great deal less time on building out functionality, and more time adding the polish or getting the extra credit.**

Submission Details:

To submit the assignment, you will need to do two things.

- Make sure that you commit your final version to the personal repository that gets created for you in GitHub classrooms. The last commit you make before the deadline is the one that will be graded.
- In the homework submission, provide the [LINK TO YOUR REPOSITORY](#) in the website field.
- If you want to use a late day, **indicate this in the text field**. Otherwise, the last commit before the deadline in the GitHub repository is the one that will be graded.

Specification:

- Create an Meals reviewer/editor (1.2 points)
 - Users should be able to create meals, and these should be added to the server. When created, meal times should be able to be configured as either the current time, or some other time.
 - Users should be able to add, modify, and delete meals.
 - Users should be able to see the stats for each meal (aggregated data from all foods in that meal). In other words, total Calories, Protein, etc.
- Create a Foods reviewer/editor (0.8 points)
 - Users should be able to create foods in each meal, and these should be added to the server. You can base these off the list of foods provided by the endpoint `/Foods` on the server if you like, but you are not limited to that list.
 - Users should be able to add, modify, and delete foods in a given meal.
- Hook up the stats for the current day's meals with the randomly/statically chosen values you had from React 2. (1 point)
 - For example, if you used charts to display the total calories consumed versus the user's goals, the total calories eaten should now be based on the foods for the day.
- Allow the user to view their stats for the past 7 days (1 point).
 - One such way may be a line/bar chart for their daily stats compared to each goal. For example, there might be a line graph with 7 (one for each day) points showing the total calories consumed. You do not need to use graphs for this, but they are generally good methods for showing this type of data.
- Clean and clear code/interface (2 points)
 - Example things we will consider:
 - User feedback is reasonable
 - Design is clear and visually appealing (Note: check the [Approved Packages](#) page for styling components)
 - Code is understandable
- EXTRA CREDIT: Implement one of the designs/concepts you had for improving accessibility from the design homework. (1 point possible)
 - If you do this, make sure to describe in your text entry of this canvas assignment what you did and how to enable/disable it.
 - If you have another feature you would like to add to your design that is not part of the requirements, check with the TA's and we may allow you to do that for the point of extra credit instead.

Submission

✓ Submitted!

Nov 27, 2019 at 7:16pm (late)
[Submission Details](#)

Grade: 6.2 (6 pts possible)
Graded Anonymously: no
[View Rubric Evaluation](#)

Comments:
No Comments

React-Native 3											
Criteria	Ratings										Pts
Meals: Creating Users should be able to add new meals to their current day (0.1). They should be able to specify all fields of the meals, including date/time (0.2). As a shortcut, users should be able to click something like "Use current time" (0.1).	0.4 pts Full Marks	0.3 pts Partial Credit		0.2 pts Partial Credit		0.1 pts Partial Credit		0.0 pts No Marks			0.4 pts
Meals: Review, Edit, Delete Users should be able to review all their meals for the current day (0.2), make modifications to meals (0.2), and delete them (0.1). Each meal should also show the aggregated values for Calories, Fat, Protein, and Carbohydrates based on all foods for that meal (0.3).	0.8 pts Full Marks	0.7 pts Partial Credit	0.6 pts Partial Credit	0.5 pts Partial Credit	0.4 pts Partial Credit	0.3 pts Partial Credit	0.2 pts Partial Credit	0.1 pts Partial Credit	0.0 pts No Marks		0.8 pts
Foods: Creating	0.4 pts					0.3 pts					

Review/Editing Users should be able to add new foods to a given meal (0.4).	0.4 pts Full Marks	0.0 pts No Marks				0.4 pts
Foods: Review, Edit, Delete Users should be able to review all the foods for a given meal (0.2), make modifications to foods (0.1), and delete them (0.1).	0.4 pts Full Marks	0.3 pts Partial Credit	0.2 pts Partial Credit	0.1 pts Partial Credit	0.0 pts No Marks	0.4 pts
All Stats Aggregated in Today View All stats (Calories, Protein, Carbohydrates, Fat, and Activity) are aggregated in the main view (0.8). This should be done in a way that allows the user to compare them to their goals without flipping back and forth between that view and their profile (0.2).	1.0 pts Full Marks	0.75 pts Partial Credit	0.5 pts Partial Credit	0.25 pts Partial Credit	0.0 pts No Marks	1.0 pts
Stats for Past 7 Days All stats (Calories, Protein, Carbohydrates, Fat, and Activity) can be viewed from the past 7 days. This can be in the form of a graph, but graphs are not required.	1.0 pts Full Marks	0.75 pts Partial Credit	0.5 pts Partial Credit	0.25 pts Partial Credit	0.0 pts No Marks	1.0 pts
Clean/Clear Code and Interface Consider general usability and code implementation. Interface should be clean, appealing, and abide by the heuristics/principles discussed in class.	2.0 pts Full Marks	1.5 pts Partial Credit	1.0 pts Partial Credit	0.5 pts Partial Credit	0.0 pts No Marks	2.0 pts
Accessibility Implement one of the designs/concepts you had for improving accessibility from the design homework. (1 point possible)	1.0 pts Full Marks	0.0 pts No Marks				1.0 pts
Total Points: 7.0						