Assignment4 | Coursera

Important Information
It is especially important to submit this assignment before the deadline, April 10, 11:59 PM PDT, because it must be graded by others. If you submit late, there may not be enough classmates around to review your work. This makes it difficult - and in some cases, impossible - to produce a grade. Submit on time to avoid these risks.

Instructions

Assignment Overview
In this assignment, you will be extending the router to support the ability to save and retrieve a list of favorite dishes by each of the registered users. All registered users in the system should have the ability to save any dish as their favorite dish, retrieve all their favorite dishes and remove one or all their favorite dishes. At the end of this assignment, your should have completed the following:

- Allowed users to select a dish as their favorite, and add it to the list of favorites that are saved on the server.
- Allowed users to retrieve the list of their favorite dishes from the server
- Delete one or all of their favorite dishes from their favorites list on the server.

Assignment Requirements
In this assignment, you will be supporting a new route https://localhost:3443/favorites, where the users can do a GET to retrieve all their favorite dishes, a POST to add a dish to their favorites, and a DELETE to delete the list of their favorites. In addition, the users should have the ability to issue a DELETE request to https://localhost:3443/favorites/dishObjectId and delete the specific dish from the list of their favorite dishes.

Furthermore, when the user's token is checked in *verifyOrdinaryUser()* function, it will load a new property named *decoded* to the *request* object. From this *req* object, you can obtain the user's ObjectId by using the following expression. You can use this to identify the user:

```
req.decoded._doc._id
```

This assignment consists of the following three tasks:

Task 1
In this task you will be implementing a new Mongoose schema named *favoriteSchema*, and a model named *Favorites* in the file named f*avorites.js* in the *models* folder. This schema should take advantage of

the mongoose population support to populate the information about the user and the list of dishes when the user does a GET operation.

Task 2

In this task, you will implement the Express router() for the '/favorites' URI such that you support GET, POST and DELETE operations

- When the user does a GET operation on '/favorites', you will populate the user information and the dishes information before returning the favorites to the user.
- When the user does a POST operation on '/favorites' by including {"_id":"dish ObjectId"} in the body of the message, you will (a) create a favorites document if such a document corresponding to this user does not already exist in the system, (b) add the dish specified in the body of the message to the list of favorite dishes for the user, if the dish does not already exists in the list of favorites.
- When the user performs a DELETE operation on '/favorites', you will delete the list of favorites corresponding to the user
- When the user performs a DELETE operation on '/favorites/dishObjectId', then you will remove the specified dish from the list of the user's favorite dishes.

Task 3

You will update app.js to support the new '/favorites' route.

Review criterialess

Your assignment will be graded on the basis of the following review criteria:

- A new favoriteSchema and Favorites model has been correctly implemented to take advantage of Mongoose Population support to track the users and the list of favorite dishes using their ObjectIds in the favoriteSchema and Favorites model.
- The GET, POST and DELETE operations are well supported as per the specifications above
- The app.js has been updated to support the new route.