CS568 Web Application Development I

Final Project

Assistant Professor Umur Inan

During the pandemic COVID-19, farmers are no longer able to sell their products and customers are no longer able to buy fresh local food.

You will design:

1. A Web application (React Application) to add products, browse orders, shop products, and contact customers when orders are ready for pickup.
2. Backend REST API (NodeJS, MongoDB) to support the application above.

Requirements

1. Farmers and Customers will sign up/sign in before they use the application.
2. Only Farmers can:
    a. Add, delete, update, retrieve products.
    b. See all orders and filter them by status:
        i. Orders have 3 statuses; Pending, Ready, Complete
    c. Once the order is prepared and ready for pick up, the farmer will update the order status to 'ready', and an automatic email will be sent to customers with pick up date and time.
        i. You can refer to https://www.w3schools.com/nodejs/nodejs_email.asp to send an email.
    d. Once the order is picked up, farmers will update the order status to 'complete'.
3. Only Customers can:
    a. Browse all farmers. Farmers should be sorted by reputation. Check the farmer's reputation algorithm below.
    b. Browse all products in the inventory for the selected farmer.
    c. Add products from one farmer at a time to their shopping cart. (Shopping cart cannot contain products from more than 1 farmer).
    d. Check out the shopping cart, then an order should be created with the status 'pending' and update the inventory, an automatic email should be sent to both farmer and customer with the order details.
    e. Pick up their orders, and pay the farmers.
    f. After the order status is changed to 'complete', customers may leave an optional rating and feedback. The rating can be one of three values: excellent, good, and bad.
4. Farmers Reputation Algorithm:
    a. An excellent rating adds 1 point to the farmer's reputation score.
    b. A bad rating deducts 1 point from the farmer's reputation score.
    c. A good rating does not change the farmer's reputation score.
5. SuperUser Account
    a. Application has 1 superuser account.
    b. Superuser can
        i. List all farmers' and customers' accounts.
        ii. Activates/deactivates accounts.
        iii. Reset password.
        iv. List all transactions and filter them by date.
        v. List all requests in the log file.

Technical Details

1. Use JWT for authentication and authorization.
2. Rest convention needs to be followed.
3. Create API documentation with Swagger for the following:
    a. Entity, verb, request body, response body.
4. Use feature branches for each task/issue/feature.
5. Log all the backend API requests to a file.

Technical Video Tutorial

- Every group is required to record 1 technical video no longer than 30 minutes.
    - Each member has 10 minutes to talk about features that are implemented by him/her.
    - For example: https://www.youtube.com/watch?v=Ur6MNStwXlQ
    - Upload the video tutorial to YouTube and submit the video link.
        - Videos can be published as ‘public’, or ‘unlisted’ so only those with the link can watch it.

Submission

- Daily push is required. (At least one push for each team member)
- https://classroom.github.com/g/V22LpuKH
- Deadline: 09/17/2021 11:55 am.

Important Notes

- You are not allowed to share codes between other groups. If detected, all group members get NC.
- Remember to respect the code honor submission policy. All written code must be original. Presenting something as one’s own work when it came from another source is plagiarism and is forbidden.
- Plagiarism is a very serious thing in all American academic institutions and is guarded against vigilantly by every professor.