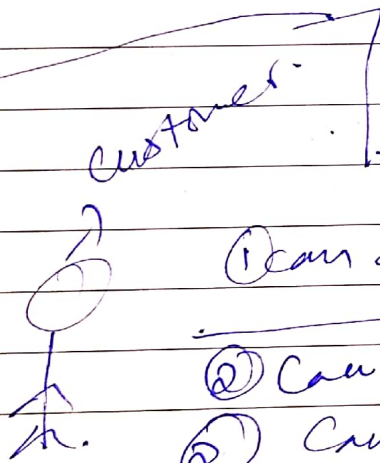
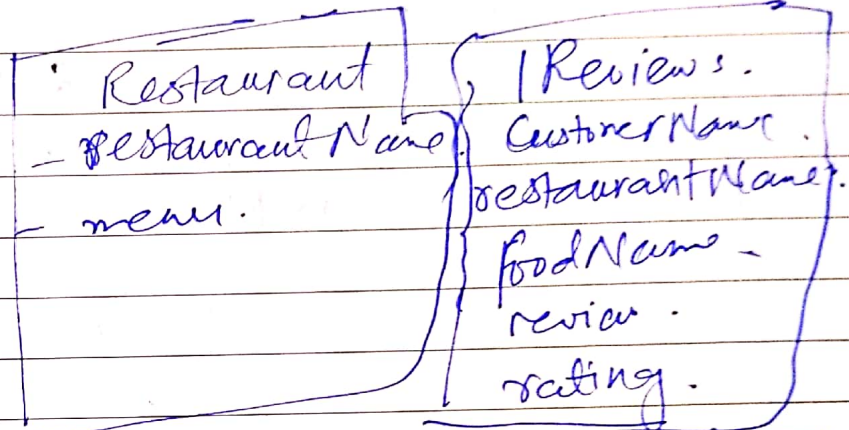


Date

Jan 11<sup>th</sup>.

Collections.



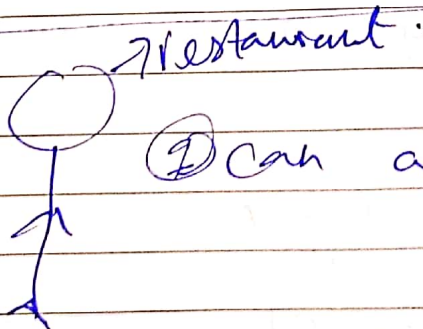
update/delete  
① Can add/delete

② Can view top restaurants.

③ Can view overall restaurant/food rating.

④ Can search for foods and see overall rating ~~other~~ their reviews.

⑤ Can view top restaurant.



① Can add/update/delete menu.

Date

Jan 11<sup>th</sup>

Req 1. <sup>Post Request</sup> Endpoint  
Add menu. <sup>v.</sup> `"/api/new-menu"`.

Request body.

Sto.  $\leftarrow$

restaurantName : "XYZ".

Array of Obj's  $\leftarrow$

menu : `"[{ 'Food1': 'Rating' },  
{ 'Food2': 'Rating2' }]"`.

response.

`"success : 'True/false'"`.

The `"/api/new-menu"` api.

`addMenu = (req, res) => {`

`const {restaurantName, menu} = req.body.`

$\rightarrow$  Saves. this to menu collection.

$\rightarrow$  Sends response of success.  
upon success.

Similar for update and Delete menu.  
only request type changes and Db operation

Date

Restaurant Name.

Req 2.

{Food :

load field.

new overall rating  
~~features~~

Reviews collection

- Food name
- Restaurant Name
- Customer name
- Rating

Add Review. (Food name, Restaurant Name, Customer Name, Rating) <sup>review</sup>

↳

Update Rev.. (Food -- same as above) { }

Delete Rev.

overall. Req 3.

getRestRev. (Rest Name)

Rest Aggregate. (lookup revs).

overall rating -  $\frac{R_1 + R_2 + R_3 + \dots + R_n}{\text{Total number of ratings/reviews}}$

get food name for a rest (rest name, food name)

Same mechanism as above but also add food filter.

Welcome Notes



Req 4.

(Check Other Cukatings (foods .  
have . filter Food }  
res . reviews . ? ?

Req 5.

top Restaurant.

check if  
a separate  
can be created for  
this.

repeated  
in Req 3.

- ← Calculate overall  
Rating for each rest.
- Sort in desc order.
  - get the first one.