https://shorturl.at/3lCHe 12 Jun 2024

Capstone Project

****

Implementation of Backend for Capstone Project (Guideline only)

****

```
POC
Backend
        Functionality
                - on sign in create a new user in users
                - on login compare username and password with database and proceed accordingly
                - on add to cart insert a record in cart collection
                - on buy now update total quantity in products collection

        API calls
                - Create user                   -> insertuser
                - Login                         -> login
                - Show all products             -> fetch
                - Add to cart                   -> insertproduct / updateproduct
                - Reduce from cart              -> updateproduct / deleteproduct
                - Buy now                       -> updateproducts

collections
                - products
                        [
                                {
                                        product_1 details
                                },
                                {
                                        product_2 details
                                },
                                ...
                        ]
                - users (u_name, upwd)
                        [
                                {
                                        user_1 details
                                },
                                {
                                        user_2 details
                                },
                                ...
                        ]
                - cart
                        [
                                {
```

```
                    id:
                    user:
                        product_details:
                },
                ...
        ]
```

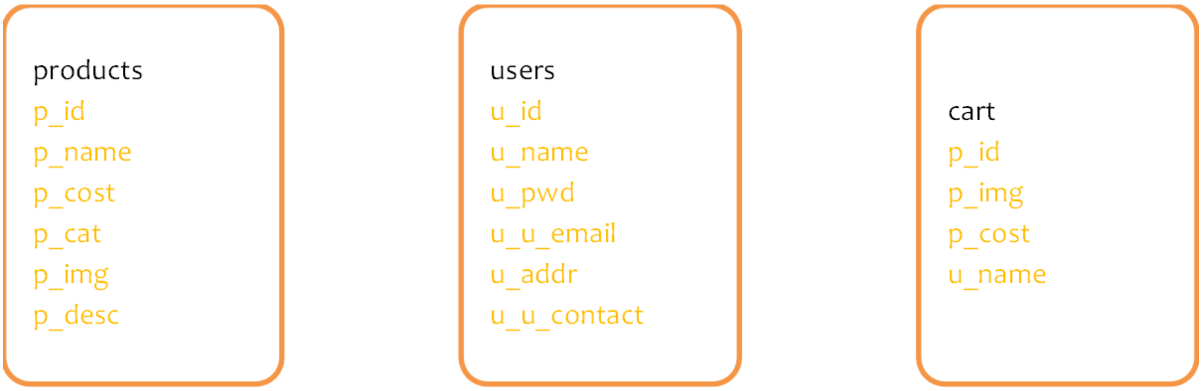
Front end

- Design e-commerse website
    - options are (two or more)
        - Medicins
        - Clothing
        - Workout
        - Mobiles
        - Mobile accessories
        - Computer equipments
        - Computer accessories
        - Books
        - Groceries
        - Car accessories
        - Jewelleries
        - watches
        - Sun glasses
- Website should have rich user interface
- There should be card layout for each product
- on mouse hover on product image that product should be enlarged
- There should be 'learn more', 'add to cart' and 'buy now' options
- There should be login page for sigining in
- There should be cart page where we will get number of added items
- In buy now page show complete cost of purchase

Enhancements (Optional)

- Try user interface without using Bootstrap CDN
- Use different GST rates for various products

Create mongodb database ‘miniprj’



products
p_id
p_name
p_cost
p_cat
p_img
p_desc

users
u_id
u_name
u_pwd
u_u_email
u_addr
u_u_contact

cart
p_id
p_img
p_cost
u_name

Create three collections

- products
  - p_id
  - p_name
  - p_cat
  - p_img
  - p_cost
  - p_desc
  - p_gst
- users
  - u_serid
  - u_name
  - u_pwd
  - u_u_email
  - u_addr
  - u_contact
- cart

Use mongodbcrud code and update the same, dont change url.js and server.js

fetch.js

```
//import modules
const express = require('express')
let mongodb = require('mongodb')
//import url
```

```
const url = require('../url')
//create mongoclient
let mcl = mongodb.MongoClient
//create router instance
let router = express.Router()
//database name
let dbName = 'miniprj'
//create restapi
router.get("/", (req, res) => {
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection')
      else {
            let db = conn.db(dbName)
            db.collection('products').find().toArray((err, array) => {
                if (err)
                    console.log('Error:- ', err)
                else {
                    console.log('Data Sent')
                    res.json(array)
                    conn.close()
                }
            })
      }
      })
})

//User login Authentication
router.post('/auth', (req, res) => {
      let u_name = req.body.u_name
      let upwd = req.body.upwd
      let obj = { u_name, upwd }
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection:- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('users').find(obj).toArray((err, array) => {
                if (err)
                    console.log(err)
                else {
                    if (array.length > 0)
                         res.json({ 'auth': 'success', 'user': u_name })
                    else
                         res.json({ 'auth': 'failed' })
                    console.log('Auth response sent')
                    conn.close()
                }
            })
      }
      })
})

//Fetch cart data
```

```
router.post("/fetchCart", (req, res) => {
      let u_name = req.body.u_name
      let obj = { u_name }
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection:- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('cart').find(obj).toArray((err, array) => {
                if (err)
                    console.log(err)
                else {
                    res.json(array)
                    console.log(`Cart response for ${obj.u_name} sent`)
                    conn.close()
                }
            })
      }
      })
})

//export router
module.exports = router
```

insert.js

```
//import modules
const express = require('express')
let mongodb = require('mongodb')
//import url
const url = require('../url')
//create mongoclient
let mcl = mongodb.MongoClient
//create router instance
let router = express.Router()
//database name
let dbName = 'miniprj'
//create restapi
router.post("/", (req, res) => {
      let obj = req.body
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection :- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('products').insertOne(obj, (err) => {
              if (err)
                    res.json({ 'insert': 'Error ' + err })
                else {
                    console.log("Data inserted")
                    res.json({ 'insert': 'success' })
                    conn.close()
                }
            })
      }
      })
})

//Insert User
router.post("/createUser", (req, res) => {
      let obj = {
      "userid": req.body.userid,
      "u_name": req.body.u_name,
      "upwd": req.body.upwd,
      "u_email": req.body.u_email,
      "u_addr": req.body.u_addr,
      "u_contact": req.body.u_contact
      }
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection :- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('users').insertOne(obj, (err) => {
                if (err)
                    res.json({ 'userInsert': 'Error ' + err })
                else {
                    console.log("User inserted")
                    res.json({ 'userInsert': 'success' })
                    conn.close()
                }
            })
      }
      })
})

//insert product into cart
router.post("/cartInsert", (req, res) => {
      let obj = {
      "p_id": req.body.p_id,
      "p_cost": req.body.p_cost,
      qty: 1,
      "p_img": req.body.p_img,
      "u_name": req.body.u_name
      }
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection :- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('cart').insertOne(obj, (err) => {
                if (err)
                    res.json({ 'cartInsert': 'Error ' + err })
                else {
                    console.log("Prouct in Cart inserted")
                    res.json({ 'cartInsert': 'success' })
                    conn.close()
                }
            })
      }
      })
})

//export router
module.exports = router
```

update.js

```
//import modules
const express = require('express')
let mongodb = require('mongodb')
//import url
const url = require('../url')
//create mongoclient
let mcl = mongodb.MongoClient
//create router instance
let router = express.Router()
//database name
let dbName = 'miniprj'
//create restapi
router.post("/", (req, res) => {
      let p_id = req.body.p_id
      let obj = {
      "p_name": req.body.p_name,
      "p_cost": req.body.p_cost
      }
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection:- ', err)
      else {
             let db = conn.db(dbName)
            db.collection('products').updateOne({ p_id }, { $set: obj }, (err,
result) => {
                if (err)
                    res.json({ 'update': 'Error ' + err })
                else {
                    if (result.matchedCount != 0) {
                         console.log("Data updated ")
                         res.json({ 'update': 'success' })
                    } else {
                         console.log("Data Not updated ")
                         res.json({ 'update': 'Record Not found' })
                    }
                    conn.close()
                }
             })
      }
      })
})
//Update product in cart
router.post("/updateCart", (req, res) => {
      let p_id = req.body.p_id
      let u_name = req.body.u_name
      let obj = { "qty": req.body.qty }
      //connect to mongodb
    mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection:- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('cart').updateOne({ p_id, u_name }, { $set: obj },
                (err, result) => {
                    if (err)
                         res.json({ 'cartUpdate': 'Error ' + err })
                    else {
                         if (result.matchedCount != 0) {
                         console.log(`Cart data for ${u_name} updated`)
                         res.json({ 'cartUpdate': 'success' })
                         }
                         else {
                            console.log(`Record not updated`)
                         res.json({ 'cartUpdate': 'Record Not found' })
                         }
                         conn.close()
                    }
                })
      }
      })
})

//Update user
//? ? ?
//export router
module.exports = router
```

delete.js

```
//import modules
const express = require('express')
let mongodb = require('mongodb')
//import url
const url = require('../url')
//create mongoclient
let mcl = mongodb.MongoClient
//create router instance
let router = express.Router()
//database name
let dbName = 'miniprj'
//create restapi
router.post("/", (req, res) => {
      let obj = {
      "p_id": req.body.p_id
      }
      //connect to mongodb
      mcl.connect(url, (err, conn) => {
      if (err)
            console.log('Error in connection:- ', err)
      else {
            let db = conn.db(dbName)
            db.collection('products').deleteOne(obj, (err, result) => {
```

```
        if (err)
           res.json({ 'delete': 'Error ' + err })
        else {
          if (result.deletedCount != 0) {
            console.log("Data deleted ")
            res.json({ 'delete': 'success' })
          } else {
            console.log("Data Not deleted ")
            res.json({ 'delete': 'Record Not found' })
          }
          conn.close()
        }
      })
   }
   })
})
//Delete product from cart
router.post("/deleteCart", (req, res) => {
   let obj = {
   "p_id": req.body.p_id,
   "u_name": req.body.u_name
   }
   //connect to mongodb
  mcl.connect(url, (err, conn) => {
   if (err)
      console.log('Error in connection:- ', err)
   else {
      let db = conn.db(dbName)
      db.collection('cart').deleteOne(obj, (err, result) => {
        if (err)
          res.json({ 'cartDelete': 'Error ' + err })
        else {
          if (result.deletedCount != 0) {
            console.log(`Cart data from ${obj.u_name} deleted`)
            res.json({ 'cartDelete': 'success' })
          }
          else {
            console.log('Cart Data Not deleted')
            res.json({ 'cartDelete': 'Record Not found' })
          }
          conn.close()
        }
      })
   }
   })
})

//Delete user
//? ? ?
//export router
module.exports = router
```

Hosting the application
>npm init
1. create '.gitignore' file
    >npx gitignore node

2. login to github.com and create repository
3. copy url

4. initialise local repository

```
>git init
```

5. add files to repository

```
>git add .
```

6. check status

```
>git status
```

7. commit

```
>git commit -m "initial Commit"
```

8. add to remote repository

```
>git remote add origin
```

9. push code to repository

```
>git push -u origin master
```

Deploying nodejs on render.com

*Login to render.com

1. goto render dashboard
2. click on new+
   choose web service
3. Go down to public repository
4. paste url of github repository
5. Click on continue
6. Choose name for web service
7. leave region
8. Branch -> Master
9. root directory ./ -> path of server.js file
10. Runtime   Node
11. Build command   npm install
12. Start command   node server
13. Click on create web service and wait
14. in command prompt of render will get port no
15. now ur url on upper left part of page is ready

'<web_service_name>.onrender.com'