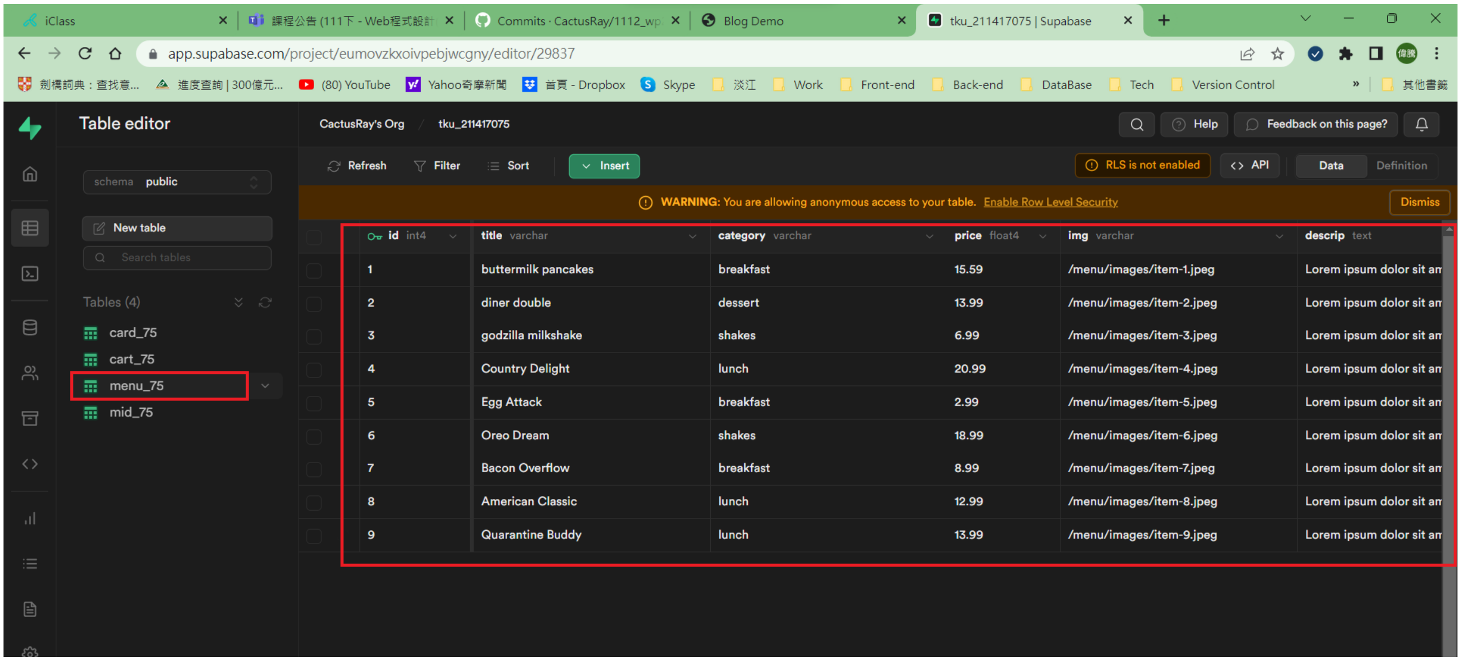
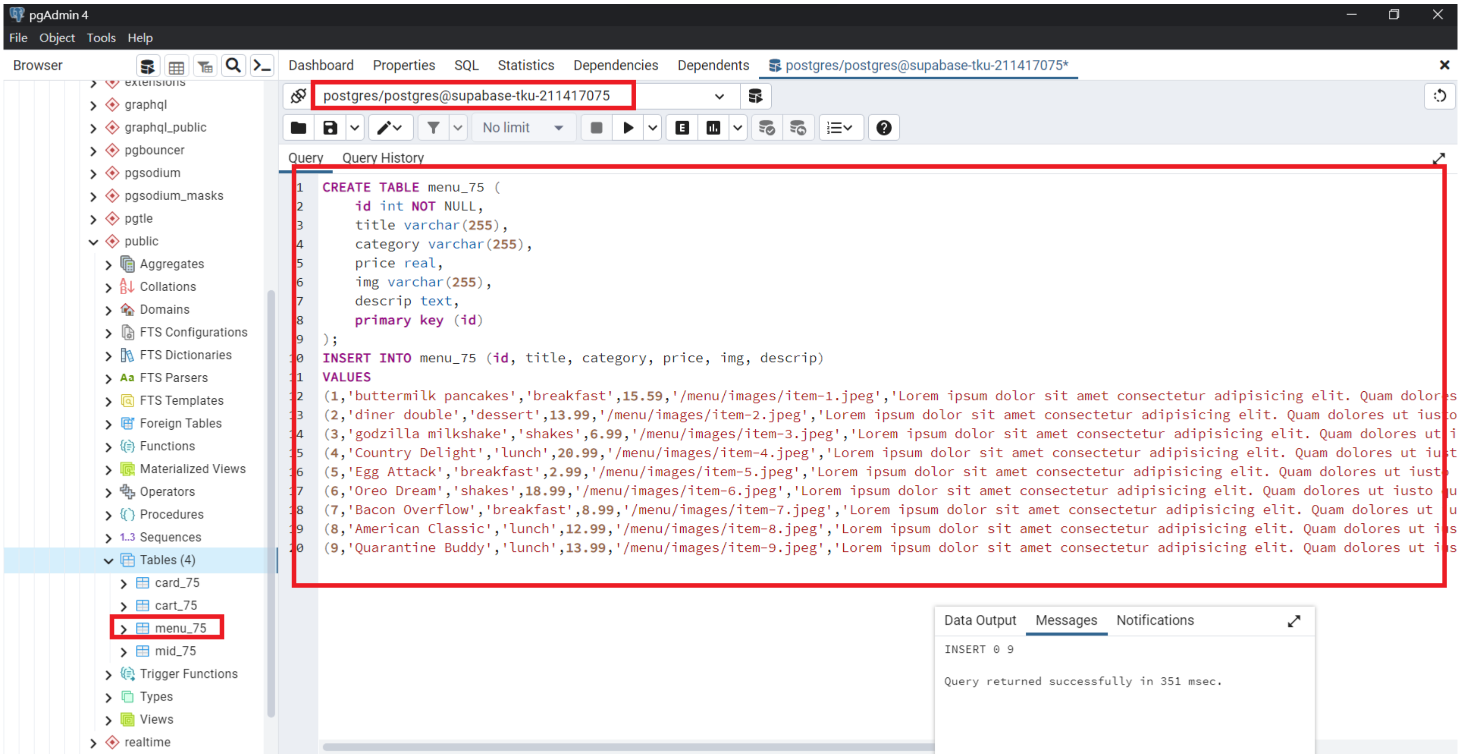
My Github URL

W011-P1: Midterm answer 6.a, 6.b



WARNING: You are allowing anonymous access to your table. Enable Row Level Security

id int4	title varchar	category varchar	price float4	img varchar	descrip text
1	buttermilk pancakes	breakfast	15.59	/menu/images/item-1.jpeg	Lorem ipsum dolor sit am
2	diner double	dessert	13.99	/menu/images/item-2.jpeg	Lorem ipsum dolor sit am
3	godzilla milkshake	shakes	6.99	/menu/images/item-3.jpeg	Lorem ipsum dolor sit am
4	Country Delight	lunch	20.99	/menu/images/item-4.jpeg	Lorem ipsum dolor sit am
5	Egg Attack	breakfast	2.99	/menu/images/item-5.jpeg	Lorem ipsum dolor sit am
6	Oreo Dream	shakes	18.99	/menu/images/item-6.jpeg	Lorem ipsum dolor sit am
7	Bacon Overflow	breakfast	8.99	/menu/images/item-7.jpeg	Lorem ipsum dolor sit am
8	American Classic	lunch	12.99	/menu/images/item-8.jpeg	Lorem ipsum dolor sit am
9	Quarantine Buddy	lunch	13.99	/menu/images/item-9.jpeg	Lorem ipsum dolor sit am

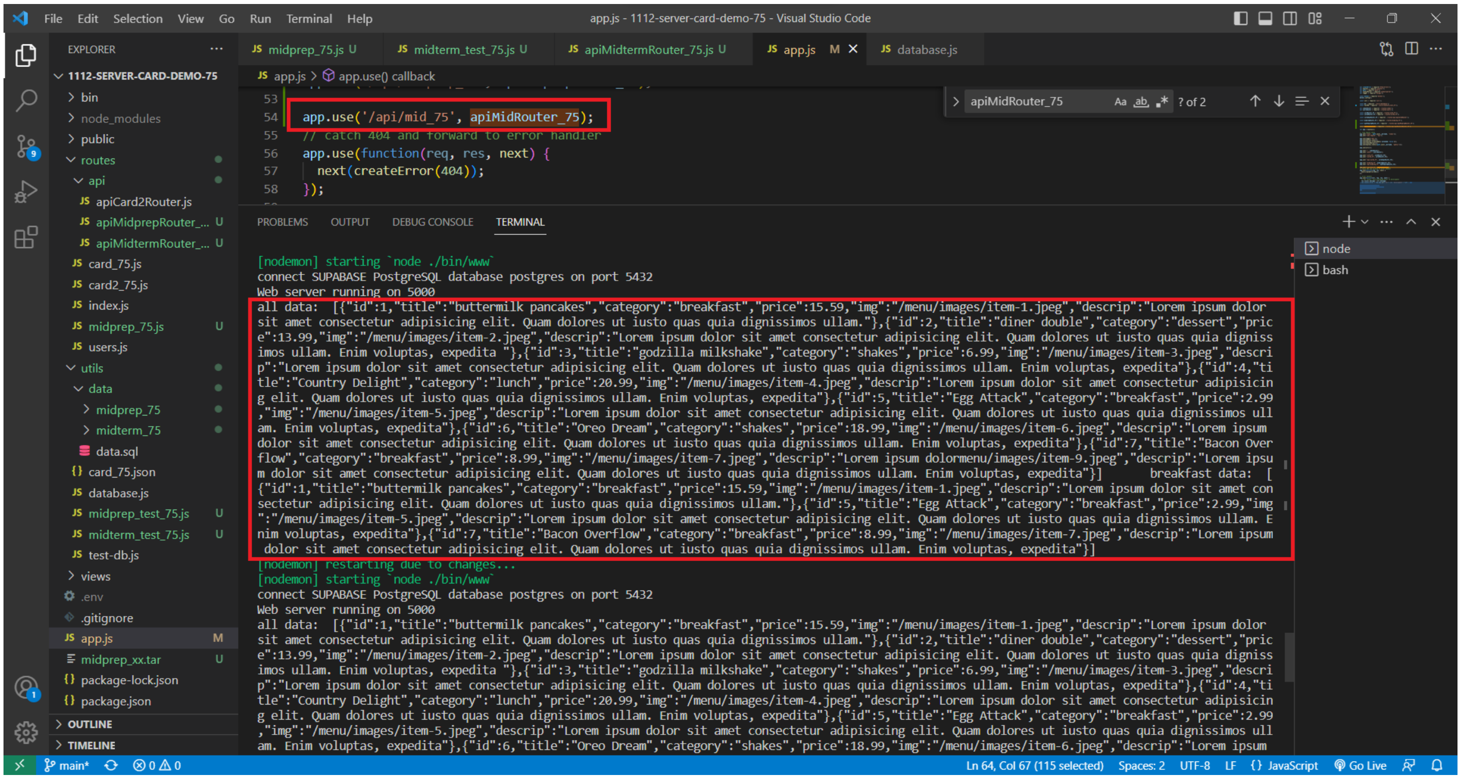


```
CREATE TABLE menu_75 (
    id int NOT NULL,
    title varchar(255),
    category varchar(255),
    price real,
    img varchar(255),
    descrip text,
    primary key (id)
);
INSERT INTO menu_75 (id, title, category, price, img, descrip)
VALUES
(1,'buttermilk pancakes','breakfast',15.59,'/menu/images/item-1.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores
(2,'diner double','dessert',13.99,'/menu/images/item-2.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto
(3,'godzilla milkshake','shakes',6.99,'/menu/images/item-3.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut i
(4,'Country Delight','lunch',20.99,'/menu/images/item-4.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iust
(5,'Egg Attack','breakfast',2.99,'/menu/images/item-5.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto
(6,'Oreo Dream','shakes',18.99,'/menu/images/item-6.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto qu
(7,'Bacon Overflow','breakfast',8.99,'/menu/images/item-7.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iu
(8,'American Classic','lunch',12.99,'/menu/images/item-8.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut ius
(9,'Quarantine Buddy','lunch',13.99,'/menu/images/item-9.jpeg','Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut ius
```

```
INSERT 0 9

Query returned successfully in 351 msec.
```

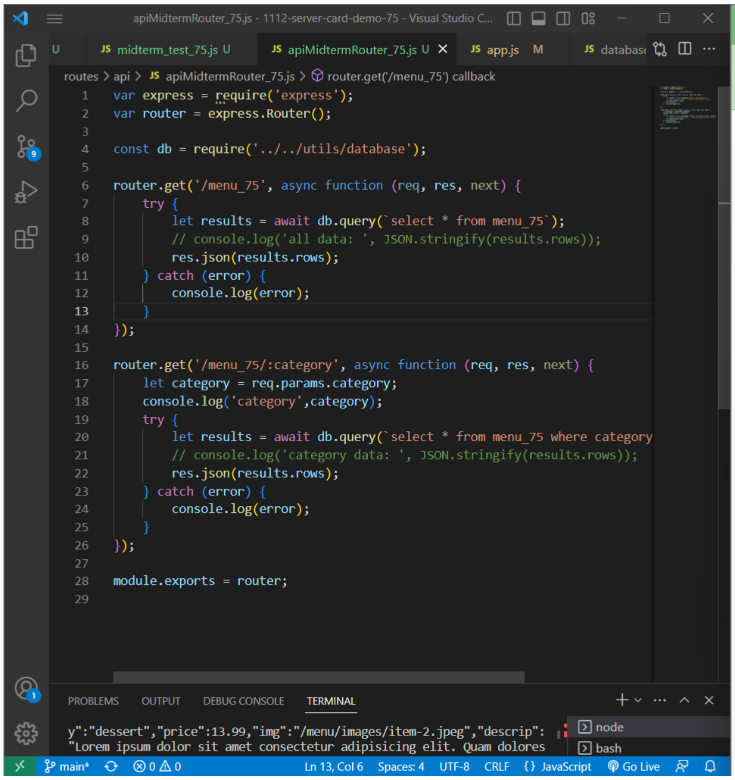
```
dd6bd68 CactusRay       Wed Apr 26 19:18:45 2023 +0800  ### W011-P1: Midterm answer 6.a, 6.b
```

W011-P2: Midterm answer 6.c, 7, 8



```javascript
app.use('/api/mid_75', apiMidRouter_75);
// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});
```

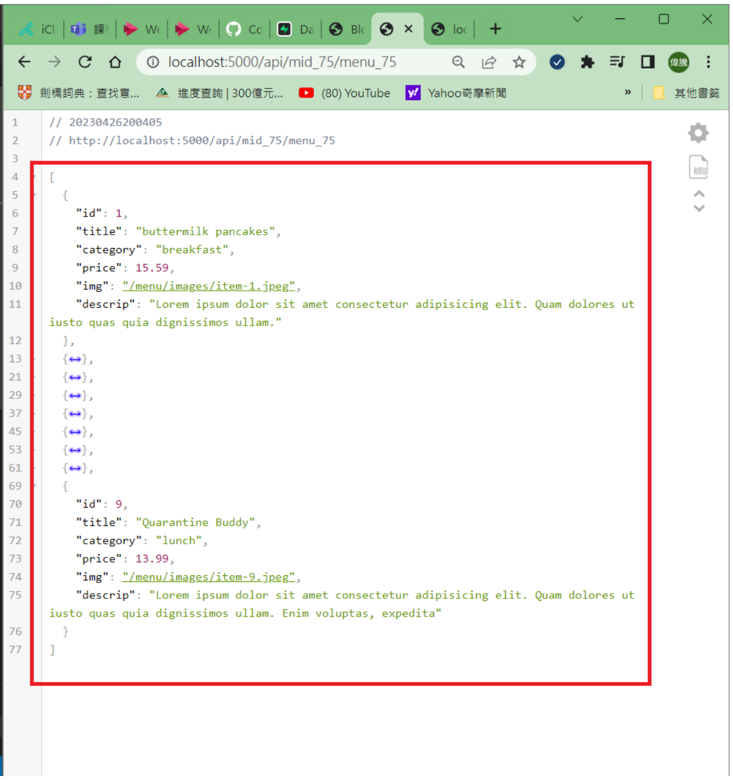
```
[nodemon] starting `node ./bin/www`
connect SUPABASE PostgreSQL database postgres on port 5432
Web server running on 5000
all data:  [{"id":1,"title":"buttermilk pancakes","category":"breakfast","price":15.59,"img":"/menu/images/item-1.jpeg","descrip":"Lorem ipsum dolor
sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."},{"id":2,"title":"diner double","category":"dessert","pric
e":13.99,"img":"/menu/images/item-2.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia digniss
imos ullam. Enim voluptas, expedita "},{"id":3,"title":"godzilla milkshake","category":"shakes","price":6.99,"img":"/menu/images/item-3.jpeg","descri
p":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"},{"id":4,"ti
tle":"Country Delight","category":"lunch","price":20.99,"img":"/menu/images/item-4.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicin
g elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"},{"id":5,"title":"Egg Attack","category":"breakfast","price":2.99
,"img":"/menu/images/item-5.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ull
am. Enim voluptas, expedita"},{"id":6,"title":"Oreo Dream","category":"shakes","price":18.99,"img":"/menu/images/item-6.jpeg","descrip":"Lorem ipsum
dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"},{"id":7,"title":"Bacon Over
flow","category":"breakfast","price":8.99,"img":"/menu/images/item-7.jpeg","descrip":"Lorem ipsum dolormenu/images/item-9.jpeg","descrip":"Lorem ipsu
m dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"}]       breakfast data:  [
{"id":1,"title":"buttermilk pancakes","category":"breakfast","price":15.59,"img":"/menu/images/item-1.jpeg","descrip":"Lorem ipsum dolor sit amet con
sectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."},{"id":5,"title":"Egg Attack","category":"breakfast","price":2.99,"img
":"/menu/images/item-5.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. E
nim voluptas, expedita"},{"id":7,"title":"Bacon Overflow","category":"breakfast","price":8.99,"img":"/menu/images/item-7.jpeg","descrip":"Lorem ipsum
 dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"}]
[nodemon] restarting due to changes...
[nodemon] starting `node ./bin/www`
connect SUPABASE PostgreSQL database postgres on port 5432
Web server running on 5000
all data:  [{"id":1,"title":"buttermilk pancakes","category":"breakfast","price":15.59,"img":"/menu/images/item-1.jpeg","descrip":"Lorem ipsum dolor
sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."},{"id":2,"title":"diner double","category":"dessert","pric
e":13.99,"img":"/menu/images/item-2.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia digniss
imos ullam. Enim voluptas, expedita "},{"id":3,"title":"godzilla milkshake","category":"shakes","price":6.99,"img":"/menu/images/item-3.jpeg","descri
p":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"},{"id":4,"ti
tle":"Country Delight","category":"lunch","price":20.99,"img":"/menu/images/item-4.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicin
g elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"},{"id":5,"title":"Egg Attack","category":"breakfast","price":2.99
,"img":"/menu/images/item-5.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ull
am. Enim voluptas, expedita"},{"id":6,"title":"Oreo Dream","category":"shakes","price":18.99,"img":"/menu/images/item-6.jpeg","descrip":"Lorem ipsum
```



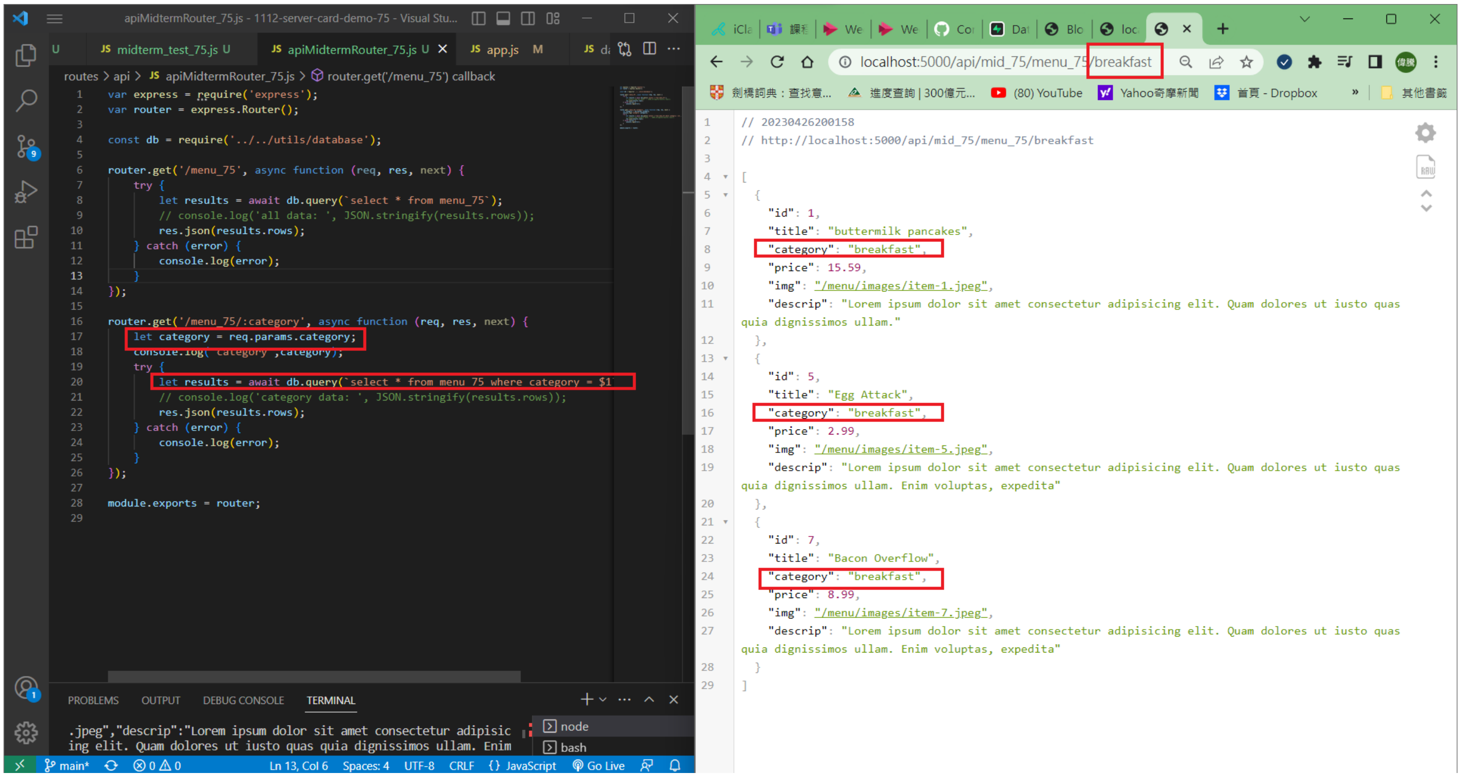
apiMidtermRouter_75.js - 1112-server-card-demo-75 - Visual Studio C...
midterm_test_75.js U
apiMidtermRouter_75.js U
app.js M
routes > api > apiMidtermRouter_75.js > router.get('/menu_75') callback
Ln 13, Col 6
Spaces: 4
UTF-8
CRLF
{} JavaScript
Go Live

```javascript
var express = require('express');
var router = express.Router();

const db = require('../../utils/database');

router.get('/menu_75', async function (req, res, next) {
    try {
        let results = await db.query(`select * from menu_75`);
        // console.log('all data: ', JSON.stringify(results.rows));
        res.json(results.rows);
    } catch (error) {
        console.log(error);
    }
});

router.get('/menu_75/:category', async function (req, res, next) {
    let category = req.params.category;
    console.log('category',category);
    try {
        let results = await db.query(`select * from menu_75 where category
        // console.log('category data: ', JSON.stringify(results.rows));
        res.json(results.rows);
    } catch (error) {
        console.log(error);
    }
});

module.exports = router;
```

```
y":"dessert","price":13.99,"img":"/menu/images/item-2.jpeg","descrip":
"Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores
```



```
// 20230426200405
// http://localhost:5000/api/mid_75/menu_75

[
  {
    "id": 1,
    "title": "buttermilk pancakes",
    "category": "breakfast",
    "price": 15.59,
    "img": "/menu/images/item-1.jpeg",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."
  },
  {↔},
  {↔},
  {↔},
  {↔},
  {↔},
  {↔},
  {↔},
  {
    "id": 9,
    "title": "Quarantine Buddy",
    "category": "lunch",
    "price": 13.99,
    "img": "/menu/images/item-9.jpeg",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
  }
]
```



apiMidtermRouter_75.js - 1112-server-card-demo-75 - Visual Stu...

routes > api > apiMidtermRouter_75.js > router.get('/menu_75') callback

```javascript
var express = require('express');
var router = express.Router();

const db = require('../../utils/database');

router.get('/menu_75', async function (req, res, next) {
    try {
        let results = await db.query(`select * from menu_75`);
        // console.log('all data: ', JSON.stringify(results.rows));
        res.json(results.rows);
    } catch (error) {
        console.log(error);
    }
});

router.get('/menu_75/:category', async function (req, res, next) {
    let category = req.params.category;
    console.log('category',category);
    try {
        let results = await db.query(`select * from menu_75 where category = $1`
        // console.log('category data: ', JSON.stringify(results.rows));
        res.json(results.rows);
    } catch (error) {
        console.log(error);
    }
});

module.exports = router;
```

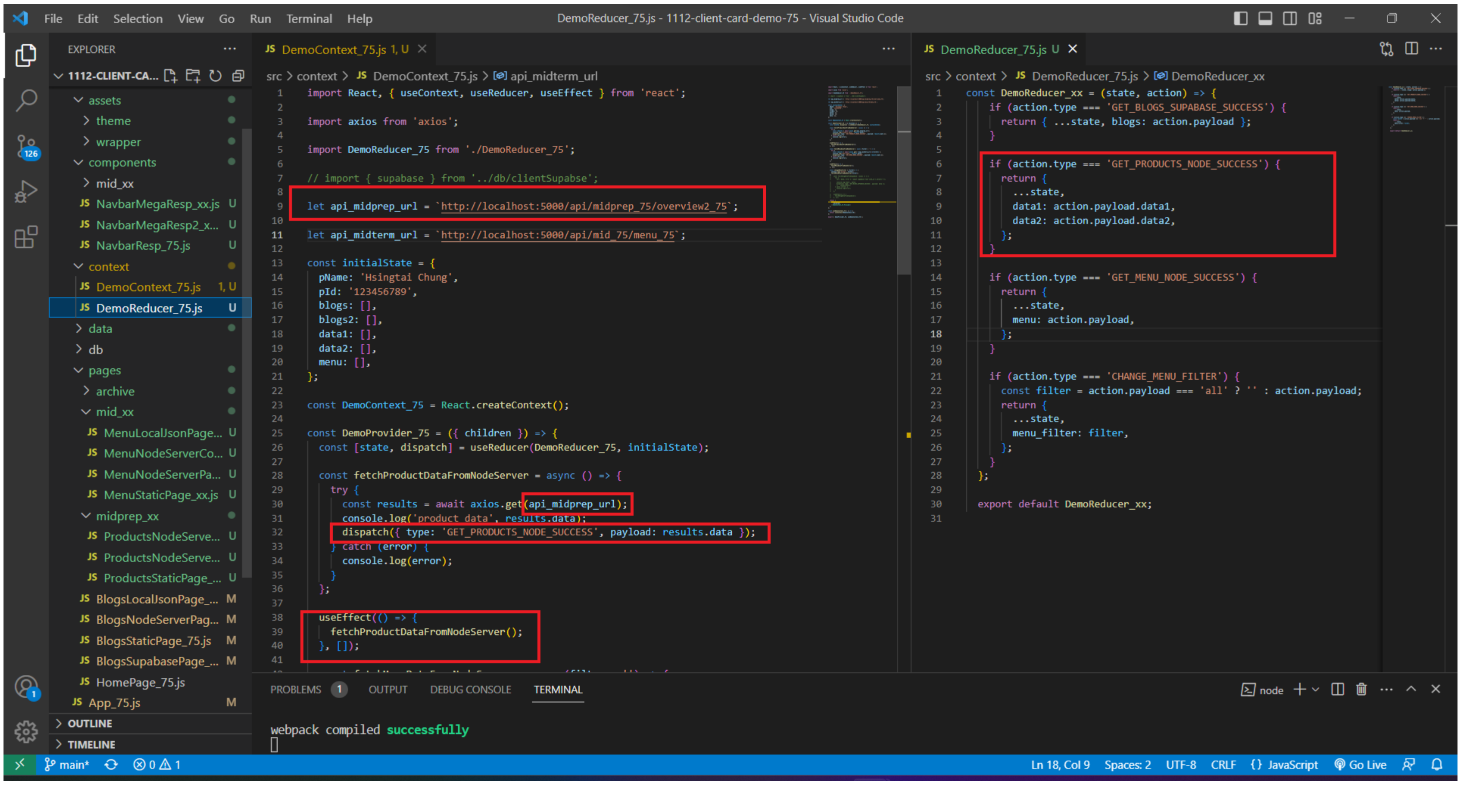
```
.jpeg","descrip":"Lorem ipsum dolor sit amet consectetur adipisic
ing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim
```

localhost:5000/api/mid_75/menu_75/breakfast

```json
// 20230426200158
// http://localhost:5000/api/mid_75/menu_75/breakfast

[
  {
    "id": 1,
    "title": "buttermilk pancakes",
    "category": "breakfast",
    "price": 15.59,
    "img": "/menu/images/item-1.jpeg",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."
  },
  {
    "id": 5,
    "title": "Egg Attack",
    "category": "breakfast",
    "price": 2.99,
    "img": "/menu/images/item-5.jpeg",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
  },
  {
    "id": 7,
    "title": "Bacon Overflow",
    "category": "breakfast",
    "price": 8.99,
    "img": "/menu/images/item-7.jpeg",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
  }
]
```

```
c2239f4 CactusRay       Wed Apr 26 20:08:31 2023 +0800  ### W011-P2: Midterm answer 6.c, 7, 8
```

W011-P3: Show Midprep answers of node server with context for rourte /midp_node_context_xx



DemoReducer_75.js - 1112-client-card-demo-75 - Visual Studio Code

src > context > DemoContext_75.js > api_midterm_url

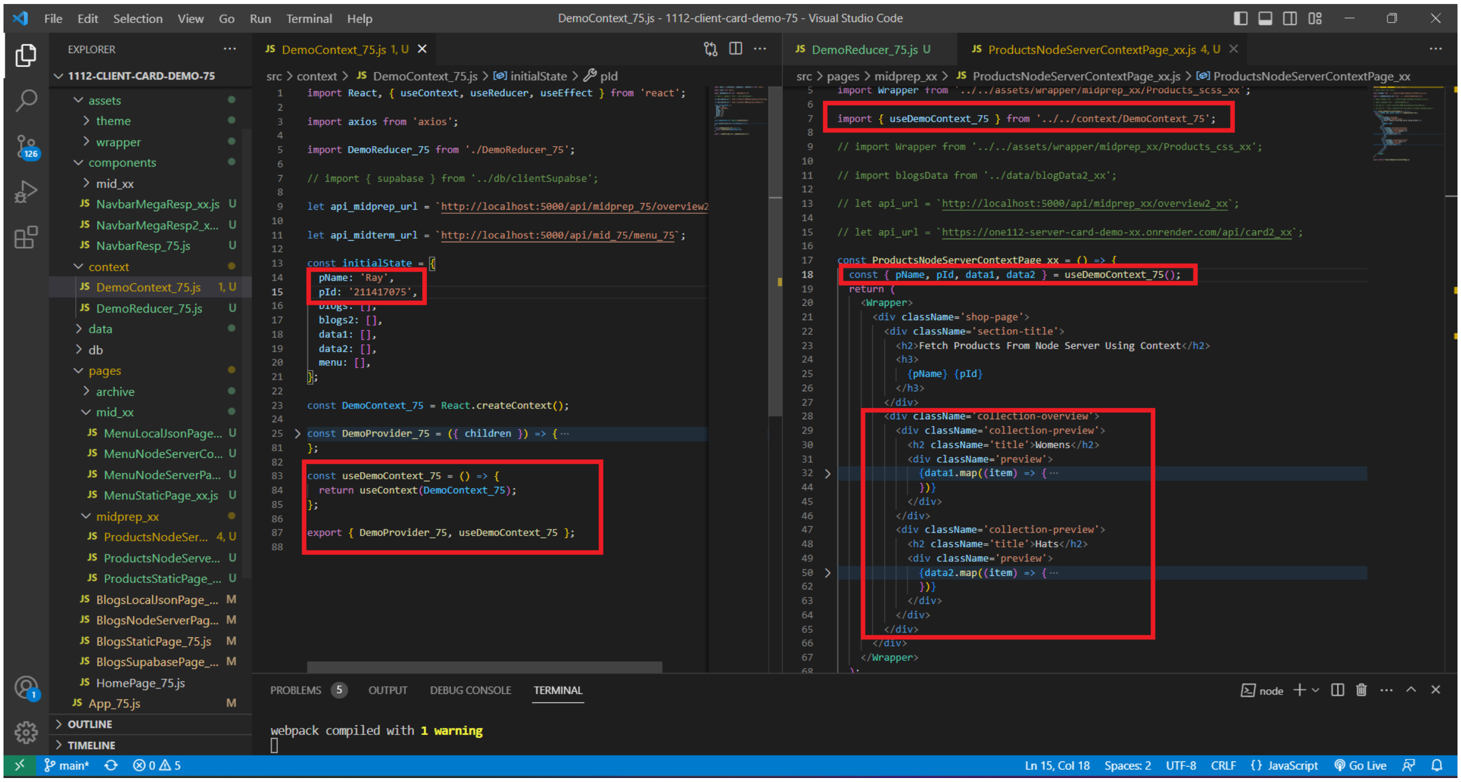
```javascript
import React, { useContext, useReducer, useEffect } from 'react';

import axios from 'axios';

import DemoReducer_75 from './DemoReducer_75';

// import { supabase } from '../db/clientSupabse';

let api_midprep_url = `http://localhost:5000/api/midprep_75/overview2_75`;

let api_midterm_url = `http://localhost:5000/api/mid_75/menu_75`;

const initialState = {
  pName: 'Hsingtai Chung',
  pId: '123456789',
  blogs: [],
  blogs2: [],
  data1: [],
  data2: [],
  menu: [],
};

const DemoContext_75 = React.createContext();

const DemoProvider_75 = ({ children }) => {
  const [state, dispatch] = useReducer(DemoReducer_75, initialState);

  const fetchProductDataFromNodeServer = async () => {
    try {
      const results = await axios.get(api_midprep_url);
      console.log('product data', results.data);
      dispatch({ type: 'GET_PRODUCTS_NODE_SUCCESS', payload: results.data });
    } catch (error) {
      console.log(error);
    }
  };

  useEffect(() => {
    fetchProductDataFromNodeServer();
  }, []);

```

src > context > DemoReducer_75.js > DemoReducer_xx

```javascript
const DemoReducer_xx = (state, action) => {
  if (action.type === 'GET_BLOGS_SUPABASE_SUCCESS') {
    return { ...state, blogs: action.payload };
  }

  if (action.type === 'GET_PRODUCTS_NODE_SUCCESS') {
    return {
      ...state,
      data1: action.payload.data1,
      data2: action.payload.data2,
    };
  }

  if (action.type === 'GET_MENU_NODE_SUCCESS') {
    return {
      ...state,
      menu: action.payload,
    };
  }

  if (action.type === 'CHANGE_MENU_FILTER') {
    const filter = action.payload === 'all' ? '' : action.payload;
    return {
      ...state,
      menu_filter: filter,
    };
  }
};

export default DemoReducer_xx;
```

```
webpack compiled successfully
```

show how to get data1, data2 from api, and store with useContext and useReducer

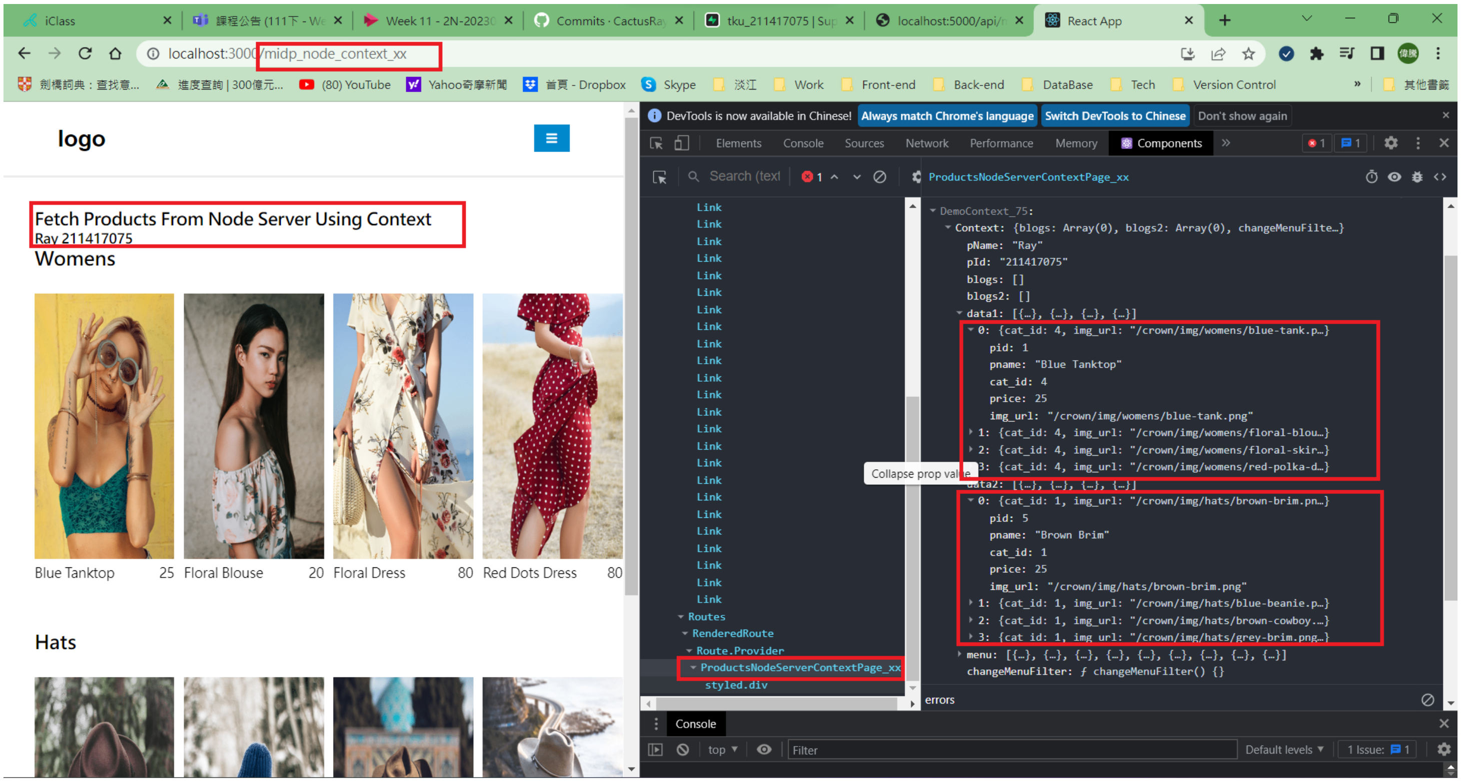


```
import React, { useContext, useReducer, useEffect } from 'react';

import axios from 'axios';

import DemoReducer_75 from './DemoReducer_75';

// import { supabase } from '../db/clientSupabse';

let api_midprep_url = `http://localhost:5000/api/midprep_75/overview2

let api_midterm_url = `http://localhost:5000/api/mid_75/menu_75`;

const initialState = {
  pName: 'Ray',
  pId: '211417075',
  blogs: [],
  blogs2: [],
  data1: [],
  data2: [],
  menu: [],
};

const DemoContext_75 = React.createContext();

const DemoProvider_75 = ({ children }) => {…
};

const useDemoContext_75 = () => {
  return useContext(DemoContext_75);
};

export { DemoProvider_75, useDemoContext_75 };
```

```
import Wrapper from '../../assets/wrapper/midprep_xx/Products_scss_xx';

import { useDemoContext_75 } from '../../context/DemoContext_75';

// import Wrapper from '../../assets/wrapper/midprep_xx/Products_css_xx';

// import blogsData from '../data/blogData2_xx';

// let api_url = `http://localhost:5000/api/midprep_xx/overview2_xx`;

// let api_url = `https://one112-server-card-demo-xx.onrender.com/api/card2_xx`;

const ProductsNodeServerContextPage_xx = () => {
  const { pName, pId, data1, data2 } = useDemoContext_75();
  return (
    <Wrapper>
      <div className='shop-page'>
        <div className='section-title'>
          <h2>Fetch Products From Node Server Using Context</h2>
          <h3>
            {pName} {pId}
          </h3>
        </div>
        <div className='collection-overview'>
          <div className='collection-preview'>
            <h2 className='title'>Womens</h2>
            <div className='preview'>
              {data1.map((item) => {…
              })}
            </div>
          </div>
          <div className='collection-preview'>
            <h2 className='title'>Hats</h2>
            <div className='preview'>
              {data2.map((item) => {…
              })}
            </div>
          </div>
        </div>
      </div>
    </Wrapper>
```

```
webpack compiled with 1 warning
```

show how to get data1, data2 from context in ProductsNodeServerContextPage_xx



logo

Fetch Products From Node Server Using Context
Ray 211417075

Womens

Blue Tanktop 25 Floral Blouse 20 Floral Dress 80 Red Dots Dress 80

Hats

```
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Link
Routes
 RenderedRoute
  Route.Provider
   ProductsNodeServerContextPage_xx
    styled.div
```

```
ProductsNodeServerContextPage_xx
DemoContext_75:
  Context: {blogs: Array(0), blogs2: Array(0), changeMenuFilte…}
    pName: "Ray"
    pId: "211417075"
    blogs: []
    blogs2: []
    data1: [{…}, {…}, {…}, {…}]
      0: {cat_id: 4, img_url: "/crown/img/womens/blue-tank.p…}
        pid: 1
        pname: "Blue Tanktop"
        cat_id: 4
        price: 25
        img_url: "/crown/img/womens/blue-tank.png"
      1: {cat_id: 4, img_url: "/crown/img/womens/floral-blou…}
      2: {cat_id: 4, img_url: "/crown/img/womens/floral-skir…}
      3: {cat_id: 4, img_url: "/crown/img/womens/red-polka-d…}
    data2: [{…}, {…}, {…}, {…}]
      0: {cat_id: 1, img_url: "/crown/img/hats/brown-brim.pn…}
        pid: 5
        pname: "Brown Brim"
        cat_id: 1
        price: 25
        img_url: "/crown/img/hats/brown-brim.png"
      1: {cat_id: 1, img_url: "/crown/img/hats/blue-beanie.p…}
      2: {cat_id: 1, img_url: "/crown/img/hats/brown-cowboy.…}
      3: {cat_id: 1, img_url: "/crown/img/hats/grey-brim.png…}
    menu: [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
    changeMenuFilter: ƒ changeMenuFilter() {}
errors
```

```
d777f6d CactusRay       Wed Apr 26 21:51:10 2023 +0800  ### W011-P3: Show Midprep answers of node server with cor
```