



SCHOOL OF COMPUTING (SoC)

ST0503 Back-end Web Development

2021/2022 SEMESTER 2

ASSIGNMENT 2

Admission Number: P2100803

Name: Haja Amir Rahman

Class: DAAA/FT/1B/01

Course: Diploma in Applied AI and Analytics

Content Page:

1. Instructions on how to setup the project on the lecturer's laptop [FOR BOTH BACKEND SERVER AND FRONT-END SERVER]
2. MYSQL database tables created and their linkage (including foreign keys)
3. Modified APIs screenshots from postman (can be accessed by importing postman json file from zip folder)

1. Instructions on how to set up the project:

1. Open MYSql Workbench and open the .sql script file from the downloaded project folder in the workbench. Run the file by clicking the first lightning symbol in the workspace area and refresh the schemas on the left to see the new database
2. Double click on the database 'sp_it' and then double click on 'Tables' to see the various table names. To view the actual tables double click on the table names and click the first lightning icon in the workspace on the left again
3. To start both servers and activate the website, right click on the parent folder and click 'Open Integrated Terminal'. Then type 'npm start' on both powershells to start the front end and backend servers simultaneously.
4. Go to your desired web browser (Google Chrome recommended) and type 'localhost:3001' in the url. The website will load and show.

Details and steps to navigate the website:

5. There are 5 main pages on this site:
 - i. index.html (main page),
 - ii. promotion page,
 - iii. category page,
 - iv. products page AKA Store,
 - v. Profile Page and
 - vi. Contact Page
6. If you click on the Profile Page you will see the Login Button at the top. Click it to login in as a registered Customer.
7. To create a new account, click on the Register Account Link and proceed to fill in your details
8. To login as Admin, click on the Admin Login Link to login in with Admin credentials and access the admin interface
9. In the store page, click on the view button under any product to view the product information, description, reviews and ratings
10. Go back to the store page and click on the reviews button under product to view it's specific reviews from customers.
11. If you wish to post a review about the product, you need to login in with Customer credentials and then go back to the reviews page from the store page to post a review about the specific product of choice.
12. To update your categories of interest, you can simply go to the profile page, scroll to the bottom and check the box belonging to the desired category of interest (refer to the category page for visual reference if required)
13. For admins, login in with admin credentials. When you reach the admin interface, you can click on add product to add a product and to check if the product is indeed added, logout of the admin interface and login to the general page to view the products page. (Same for adding categories and to check go to the categories page.
14. For registered users, you can login in with customer credentials and go to the profile page to view your details and update your profile picture. [Advanced Feature]

2. Tables in the SQL Workbench to support functionalities such as user registration, publication of product info, insertion of products and user reviews.

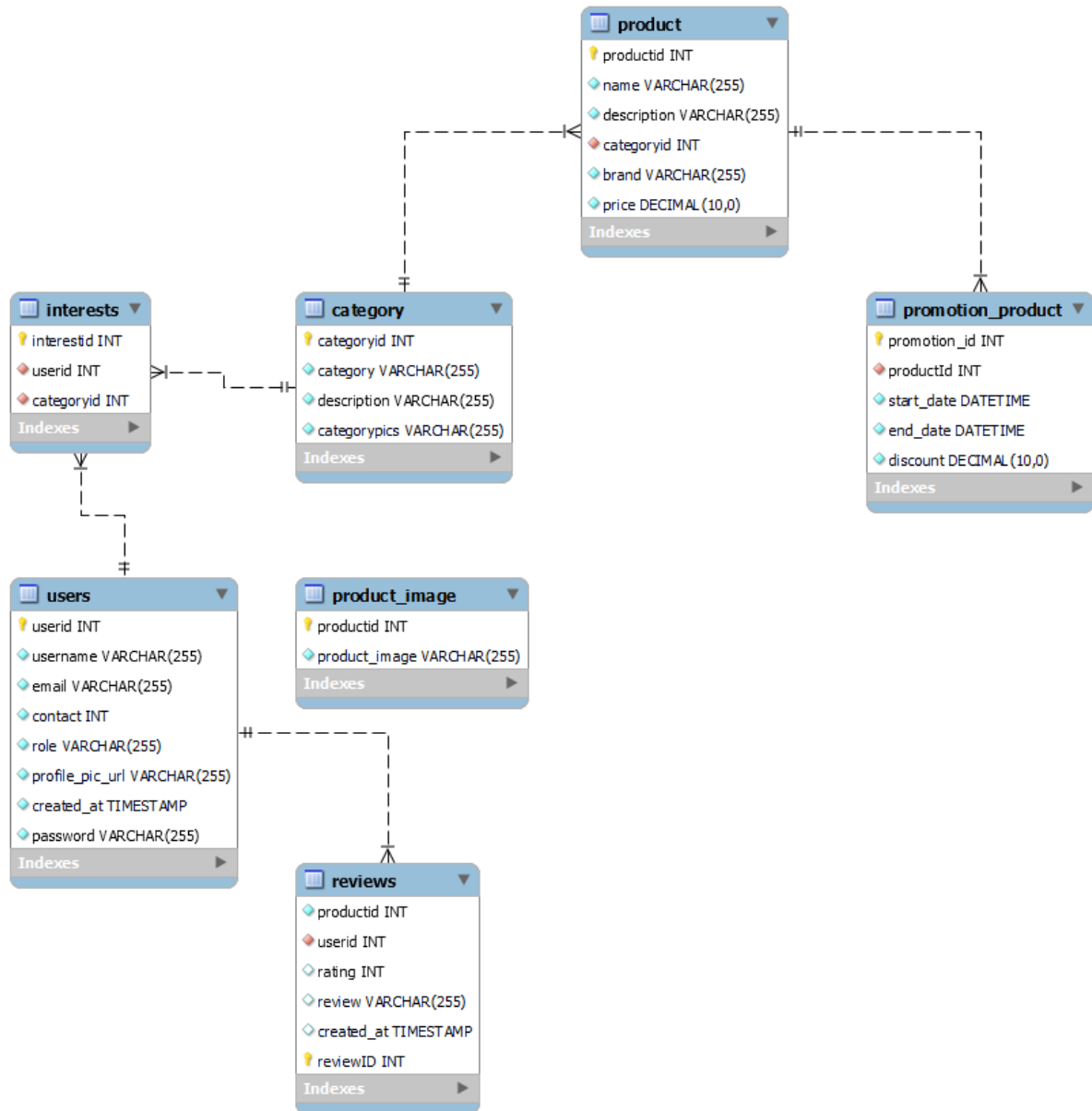
Context:

Background

SP IT! is considering setting up an online e-store to allow the public to purchase IT products it is selling. Before it launches its online e-store, it requires a web application to computerize its inventory management module and also allow the public to view the product details online.

As such, SP IT! has tasked you to design the backend API Specs the website. The API specs would support functionalities such as user registration, publication of product info, insertion of products and user reviews

ER Diagram:



3. Proof of APIs

API 1:

GET localhost:8081/users/ Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body Cookies Headers (8) Test Results Status: 200 OK Time: 98 ms Size: 1.23 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "userid": 1,
4     "username": "Terry Tan",
5     "email": "terry@gmail.com",
6     "contact": 98765432,
7     "role": "Customer",
8     "profile_pic_url": "1644143554865-IMG_20210902_145141.jpg",
9     "created_at": "2021-12-15 20:27:21",
10    "password": "123"
11  },
12  {
13    "userid": 2,
14    "username": "Amir Rahman",
15    "email": "amirramen@gmail.com",
16    "contact": 96209012,
17    "role": "Admin",
18    "profile_pic_url": "1644136761240-WhatsApp Image 2021-07-02 at 19.25.04 (5).jpeg",
19    "created_at": "2021-12-15 20:58:33",
20  }
21 }
```

API 2: Getting token for admin credentials

POST localhost:8081/admin/login Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body Cookies Headers (8) Test Results Status: 200 OK Time: 42 ms Size: 648 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjojYXQ1OjE2NDQxNDkwMDh9.Q09KF1JgSE9C1uend-GXULIqJ9TH7-BK7yA10T28-LY",
3   "userInfo": {
4     "userid": 2,
5     "username": "Amir Rahman",
6     "email": "amirramen@gmail.com",
7     "contact": 96209012,
8     "role": "Admin",
9     "profile_pic_url": "1644136761240-WhatsApp Image 2021-07-02 at 19.25.04 (5).jpeg",
10    "created_at": "2021-12-15 20:58:33",
11    "password": "123"
12  }
13 }
```

API 4: Getting token for user credentials

CA1 / localhost:8081/user/login

POST localhost:8081/user/login [Send](#)

Params Authorization Headers (9) Body **Pre-request Script** Tests Settings [Cookies](#)

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body [Cookies](#) [Headers \(8\)](#) [Test Results](#) [Status: 200 OK](#) [Time: 12 ms](#) [Size: 622 B](#) [Save Response](#)

[Pretty](#) [Raw](#) [Preview](#) [Visualize](#) [JSON](#) [...](#)

```
1  {
2    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxCjYXQj0jE2NDQxNDkwMzF9.N0_dYF6ECZjVPMRnWT5BviSaLIRKcMPtHv9bci6aj8",
3    "userInfo": {
4      "userid": 1,
5      "username": "Terry Tan",
6      "email": "terry@gmail.com",
7      "contact": 98765432,
8      "role": "Customer",
9      "profile_pic_url": "1644143554865-IMG_20210902_145141.jpg",
10     "created_at": "2021-12-15 20:27:21",
11     "password": "123"
12   }
13 }
```

[Capture requests and cookies](#) [Bootcamp](#) [Runner](#) [Trash](#)

API 4: Registering of user and creating of admin

```
app.post("/users/", (req, res, next) => {
  User.insertuser(req.body, (error, userID) => {
    if (error) {
      console.log(error);
      if (error.code === "ER_DUP_ENTRY") {
        return res.status(422).send()
      }
      res.status(500).send("What is the error?");
      return;
    }
    res.status(201).send({ userID });
  });
});

app.post("/admin/", (req, res, next) => {
  User.insertadmin(req.body, (error, userID) => {
    if (error) {
      console.log(error);
      if (error.code === "ER_DUP_ENTRY") {
        return res.status(422).send()
      }
      res.status(500).send("What is the error?");
      return;
    }
    res.status(201).send({ userID });
  });
});
```


API 5:

POST localhost:8081/admin/category/new

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "category": "Chargers",
3   "description": "Portable"
4 }
```

Body Cookies Headers (6) Test Results Status: 204 No Content Time: 25 ms Size: 207 B Save Response

Pretty Raw Preview Visualize Text

1

API 6:

GET localhost:8081/retrieve/category

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body Cookies Headers (8) Test Results Status: 200 OK Time: 10 ms Size: 1.78 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "categoryid": 1,
4     "category": "Macbook Pro",
5     "description": "Light and high performance laptops for school",
6     "categorypics": "tile_macos_large.jpg"
7   },
8   {
9     "categoryid": 2,
10    "category": "iPhone",
11    "description": "Portable, high performance and easy for different modes of communication",
12    "categorypics": "1643118396486-iphone_x_large.jpg"
13  },
14  {
15    "categoryid": 3,
16    "category": "iPad",
17    "description": "Portable, high performance and larger screen",
18    "categorypics": "Ipad.jpg"
19  }
20 ]
```

API 7:

PUT localhost:8081/admin/category/update Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   ... "category": "iMac",
3   ... "description": "Desktop PC with amazing display and color clarity, big screen"
4 }
5
```

Body Cookies Headers (6) Test Results Status: 204 No Content Time: 17 ms Size: 208 B Save Response

Pretty Raw Preview Visualize Text Copy Search

1

API 8:

POST localhost:8081/admin/product/new Send

Params Authorization **Headers (10)** **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   ... "name": "Smart Watch Charger",
3   ... "description": "Portable",
4   ... "categoryid": 14,
5   ... "brand": "Apple",
6   ... "price": "50.00"
7 }
8
```

API 9:

GETlocalhost:8081/retrieve/product/id/

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Path Variables

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| id | 50 | Description | | |

Body

Cookies

Headers (8)

Test Results

Status: 200 OKTime: 8 msSize: 371 BSave Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "productId": 50,
3    "name": "iphone10",
4    "description": "iphone10",
5    "categoryId": 2,
6    "brand": "Apple",
7    "price": 1000
8  }
```

API 10:

PUTlocalhost:8081/admin/productid/update

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

2

3

4

5

6

7

```
1  {
2    "name": "iPhone10",
3    "description": "iPhone10",
4    "categoryId": 2,
5    "brand": "Apple",
6    "price": 1000
7  }
```

Body

Cookies

Headers (6)

Test Results

Status: 204 No ContentTime: 15 msSize: 208 BSave Response

Pretty

Raw

Preview

Visualize

Text

```
1
```

API 11:

DELETE

localhost:8081/product/:id/

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

| KEY | VALUE | DESCRIPTION | | Bulk Edit |
|-----|-------|-------------|--|-----------|
| Key | Value | Description | | |

Path Variables

| KEY | VALUE | DESCRIPTION | | Bulk Edit |
|-----|-------|-------------|--|-----------|
| id | 48 | Description | | |

Body

Cookies

Headers (5)

Test Results

Status: 204 No Content

Time: 119 ms

Size: 166 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

API 12:

GET

localhost:8081/retrieve/category/:categoryid

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

| KEY | VALUE | DESCRIPTION | | Bulk Edit |
|-----|-------|-------------|--|-----------|
| Key | Value | Description | | |

Path Variables

| KEY | VALUE | DESCRIPTION | | Bulk Edit |
|------------|-------|-------------|--|-----------|
| categoryid | 6 | Description | | |

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 19 ms

Size: 584 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "productid": 10,
3    "name": "Oppo Find X Pro 3",
4    "description": "Portable, high performance, easy to use for transferring data to the cloud for backup",
5    "categoryid": 6,
6    "brand": "Oppo",
7    "price": 1000
8  },
9  {
10   "productid": 49,
11   "name": "samsung galaxy s10",
12   "description": "samsung galaxy s10",
13 }
```

API 13:

API 13 configuration in Postman. The 'Params' tab is active, showing Query Params and Path Variables. The 'Body' tab is also visible, showing a JSON response.

Query Params

| KEY | VALUE | DESCRIPTION |
|-----|-------|-------------|
| Key | Value | Description |

Path Variables

| KEY | VALUE | DESCRIPTION |
|--------|-------|-------------|
| userid | 1 | Description |

Body (JSON)

```
1 {
2   "interest": [
3     {
4       "interestid": 11,
5       "userid": 1,
6       "categoryid": 5
7     },
8     {
9       "interestid": 12,
10      "userid": 1,
11      "categoryid": 8
12     },
13   ]
14 }
```

API 14:

API 14 configuration in Postman. The 'POST' method is selected, and the URL is `http://localhost:8081/admin/upload/:categoryid`. The 'Body' tab is active, showing a form-data body type with a file upload field.

POST `http://localhost:8081/admin/upload/:categoryid` **Send**

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Body (form-data)

| KEY | VALUE | DESCRIPTION |
|--|------------------------------------|-------------|
| <input checked="" type="checkbox"/> productImage | <code>accessories_large.jpg</code> | |
| Key | Value | Description |

Body (HTML)

```
1 File uploaded successfully
```

Several endpoints that have been added and modified cannot be shown here as they only work from the website. So, to test, test with the website.