

**SCHOOL OF COMPUTING (SoC)**

**ST0503 Back-end Web Development**

**2021/2022 SEMESTER 2**

**ASSIGNMENT 1**

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
2. MYSQL database tables created and their linkage (including foreign keys)
3. Screenshots as proof of all 12 successful APIs in chronogical order (inclusive of advanced feature APIs
4. **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. Open Postman and go to import to import the postman path collection file from the downloaded project folder. You can test all APIs from here more conveniently
   4. Open VScode and right click on the main folder which was download and click integrated terminal
   5. Once terminal is open type ‘npm start’ and press enter. The server will start running in the terminal.
   6. From here you can test all the endpoints. I recommend that you have postman and mysql workbench open side by side so you can see the changes happening in real time for POST, PUT, GET and DELETE.
5. **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:

Diagram

Description automatically generated

User Table for user registration:

![Graphical user interface, application

Description automatically generated]()

Product Table for publication of product:

![Graphical user interface, text, application

Description automatically generated]()

Foreign Key for categoryid to category table:

![Graphical user interface, text, application, email

Description automatically generated]()

Category Table for category of devices:

![Graphical user interface, text

Description automatically generated]()

Reviews Table for reviews from users on the various products:

![Graphical user interface, text, application, email

Description automatically generated]()

Foreign Key for productid to product table:

![Graphical user interface, text, application, email

Description automatically generated]()

Foreign Key for userid to user table:

![Graphical user interface, text, application, email

Description automatically generated]()

Interests Table for users who are interested in multiple categories:

![Graphical user interface, application, table

Description automatically generated]()

Foreign key for userid to user table:

![Graphical user interface, text, application, email

Description automatically generated]()

Foreign key for categoryid to category table:

![Graphical user interface, text, application, email

Description automatically generated]()

Advanced Feature:

Promotion\_product Table for the promotion period and discount for the respective products:

![Graphical user interface

Description automatically generated]()

Foreign Key for productid to product table:

![Graphical user interface, text, email

Description automatically generated]()

1. Proof that the endpoints work with the sql database when tested on postman.

Endpoint 1

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 2

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 3

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 4

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 5

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 6

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 7

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 8

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 9

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 10

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 11![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 12![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 13![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 14

![Graphical user interface, text, application, email

Description automatically generated]()

Endpoint 15

![Graphical user interface, text, application, email

Description automatically generated]()