SUNSTONE

“GROCERY WEBSITE” Database Modeling and Creation of API

GROCERY WEBSITE

➢An introduction to a grocery website should effectively convey the purpose, value, and unique selling points of the online platform.

➢The system decrease a much of work load for customer.

➢They can easily buy the grocery products from home through internet

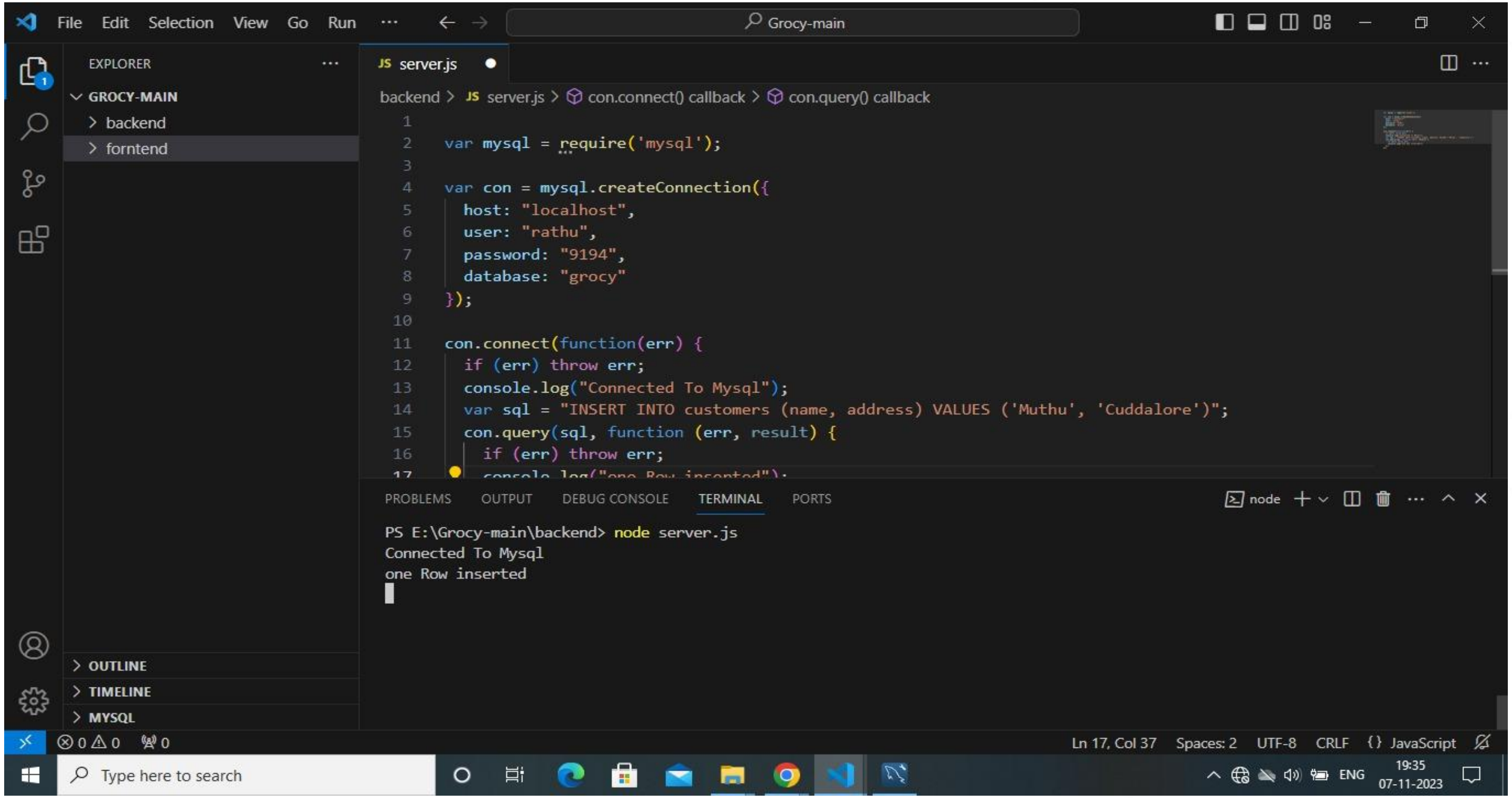
NM ID	Name	Batch
au421320104027	**PUGALENTHI. S**	**10**
au421320104029	**RABBANI MOHYADEEN.M**	**10**
au421320104039	**SASIKUMAR .R**	**10**
au421320104048	**THIYAGARAJAN .A**	**10**
au421320104701	**RATHISH .J**	**10**

Integrate the API to the frontend to ensure dynamic feature

- Choose a Database System
- Set Up the Database
- Backend Integration
- Connect Backend to the Database
- Test and Debug
- Implement Security Measures
- Optimize and Scale
- Document and Maintain

Execution and Connecting to Database

VS Code execution



File Edit Selection View Go Run ···
Grocy-main
EXPLORER
GROCY-MAIN
backend
forntend
server.js
backend > server.js > con.connect() callback > con.query() callback
OUTLINE
TIMELINE
MYSQL
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
Ln 17, Col 37 Spaces: 2 UTF-8 CRLF {} JavaScript
Type here to search
ENG 19:35 07-11-2023

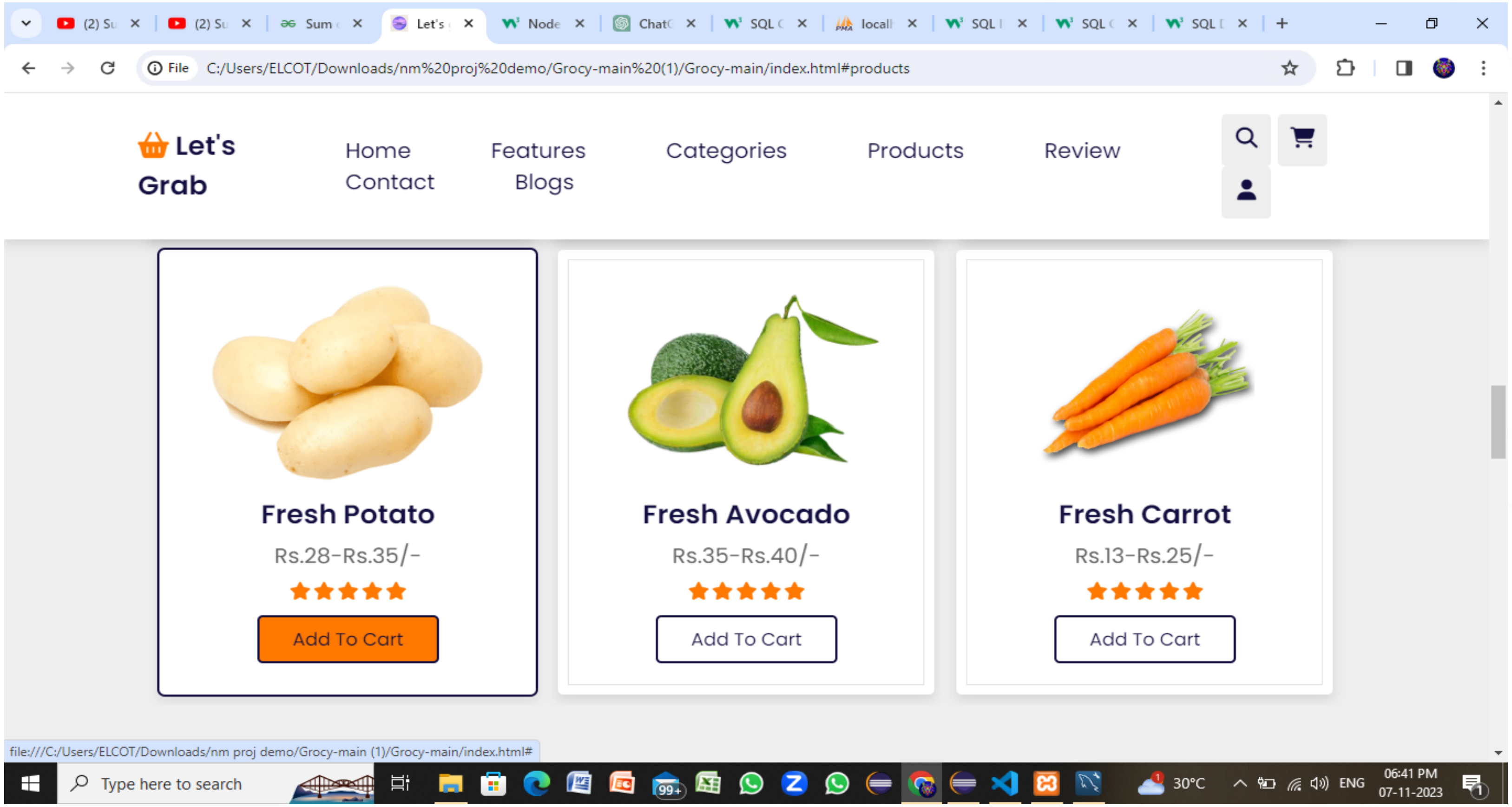
```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "localhost",
  user: "rathu",
  password: "9194",
  database: "grocy"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected To Mysql");
  var sql = "INSERT INTO customers (name, address) VALUES ('Muthu', 'Cuddalore')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("one Row inserted");
```

```
PS E:\Grocy-main\backend> node server.js
Connected To Mysql
one Row inserted
```

Inputting the data on the Website



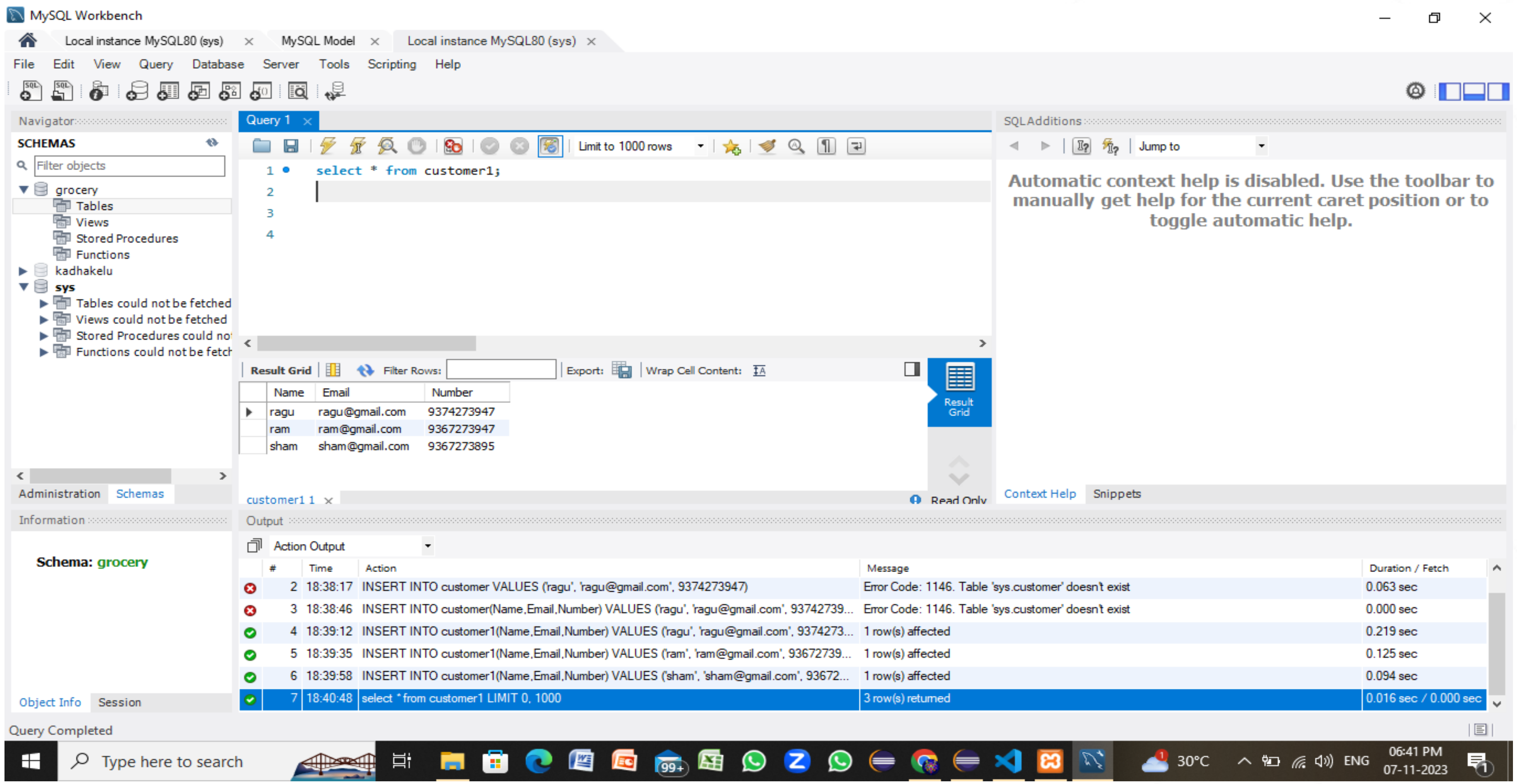
Fresh Potato
Rs.28-Rs.35/-
Add To Cart

Fresh Avocado
Rs.35-Rs.40/-
Add To Cart

Fresh Carrot
Rs.13-Rs.25/-
Add To Cart

file:///C:/Users/ELCOT/Downloads/nm proj demo/Grocy-main (1)/Grocy-main/index.html#

Rendering output of API’s to table components



```
select * from customer1;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Name	Email	Number
ragu	ragu@gmail.com	9374273947
ram	ram@gmail.com	9367273947
sham	sham@gmail.com	9367273895

Schema: grocery

	#	Time	Action	Message	Duration / Fetch
✗	2	18:38:17	INSERT INTO customer VALUES ('ragu', 'ragu@gmail.com', 9374273947)	Error Code: 1146. Table 'sys.customer' doesn't exist	0.063 sec
✗	3	18:38:46	INSERT INTO customer(Name,Email,Number) VALUES ('ragu', 'ragu@gmail.com', 93742739...	Error Code: 1146. Table 'sys.customer' doesn't exist	0.000 sec
✓	4	18:39:12	INSERT INTO customer1(Name,Email,Number) VALUES ('ragu', 'ragu@gmail.com', 9374273...	1 row(s) affected	0.219 sec
✓	5	18:39:35	INSERT INTO customer1(Name,Email,Number) VALUES ('ram', 'ram@gmail.com', 93672739...	1 row(s) affected	0.125 sec
✓	6	18:39:58	INSERT INTO customer1(Name,Email,Number) VALUES ('sham', 'sham@gmail.com', 93672...	1 row(s) affected	0.094 sec
✓	7	18:40:48	select * from customer1 LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec

LEARNING OUTCOME

- Connecting a database in a cultivates a comprehensive set of learning outcomes. Individuals develop crucial skills in database management by grasping the fundamentals of setting up and managing databases such as MySQL Proficiency in Structured Query Language (SQL) is acquired, empowering the execution of critical operations like inserting, updating, querying, and deleting data.
- Learners gain an understanding of data modeling and design, allowing the creation of efficient and normalized database structures tailored for inventory management.
- Connecting backend applications to databases becomes second nature, enabling the implementation of essential Create, Read, Update, and Delete (CRUD) functionalities.
- Emphasis on data security measures, error handling, and optimization techniques fortifies skills in maintaining the integrity, security, and performance of the database.
- Validating and sanitizing user inputs to practical application through hands-on projects, this learning journey equips individuals with practical and theoretical knowledge for proficiently managing and manipulating inventory data within a database system.

STEP-WISE DESCRIPTION

Step 1: Choose a Database Management System (DBMS)
Select a DBMS: Choose a suitable database system such as MySQL, based on the project's requirements and scalability needs.
Step 2: Set Up the Database
Install and Configure: Install the MySQL and configure it on your server or local machine.
Create the Database: Using the DBMS tools or command-line interface, create a database specifically for managing inventory data.
Step 3: Define Database Schema
Design Database Schema: Plan and design the structure of the database, including tables for products, orders, users, etc.
Step 4: Backend Integration
Choose Backend Technology: Select a backend technology or framework Node.js for your application.
Install Database Drivers: Install necessary database drivers or ORM libraries to connect your chosen backend technology to the database.

Step 5: Connect the Backend to the Database

Configure Database Connection: Set up a connection to the database within your backend code, providing the necessary connection details such as host, port, username, password, and database name.

Create CRUD Operations: Write code to perform CRUD operations (Create, Read, Update, and Delete) within your backend code, allowing your application to interact with the database.

Step 6: Test and Debug

Test Database Connectivity: Ensure that your backend application can connect to the database and perform CRUD operations accurately.

Handle Errors: Implement error handling mechanisms for database-related errors, ensuring robustness and data integrity.

Step 7: Implement Security Measures

Secure Data Inputs: Implement data validation and parameterized queries to prevent SQL injection attacks and maintain data security.

SUMMARY OF YOUR TASK

- The process of establishing database connectivity for a Job Search website involves several key steps. It begins by selecting an appropriate Database Management System (DBMS) such as MySQL followed by the setup and configuration of the database.
- Planning the database schema, defining relationships between different data entities, and choosing a suitable backend technology are crucial intermediate steps.
- The integration between the backend and the database involves configuring the connection and implementing Create, Read, Update, and Delete (CRUD) operations to enable interaction.
- Subsequent testing, debugging, and error handling ensure a stable and error-resistant connection. Lastly, implementing stringent security measures, including data validation and protection against vulnerabilities like SQL injection, safeguards the integrity and security of the inventory data.
- This systematic process equips the stock inventory website with a robust and secure foundation for managing, retrieving, and manipulating inventory-related information. Adjustments or additional security protocols might be necessary based on the unique requirements and scale of the project.

Assessment Parameter

Check-List

- Gather requirements for the project
- Prepare database design schemas
- Get your initial project Structure ready
- Initiate a git repository
- add Readme.md file with description of the project
- Commit all changes with "first commit"
- create a repository on github realted to project
- Push your changes to github

Submission Github

https://github.com/rishabh1723/NM-Batch10/tree/main/Assessment%204

SUNSTONE

Thank you!