DATABASE MANAGEMENT SYSTEMS PROJECT

Ahmed Shamsudeen (106118006), Brijesh D (106118024), Naren Sairam (106118062)

NOSQL MINI PROJECT:

Personalized Note Taker

Description:
NoteApp is a simple notepad app. It gives you a quick and simple notepad editing experience where you can write notes, shopping lists and to-do lists.

Project GitHub link: https://github.com/nsi319/Notes-App.git

Project Features:
- Organize notes by date of modification
- View in grid format
- Edit saved notes
- Delete notes

Experimental setup:

Mobile App Technology Stack:
- Java
- XML
- mongo-java-driver (3.4.0) (for database connectivity)

Database:
- MongoDB

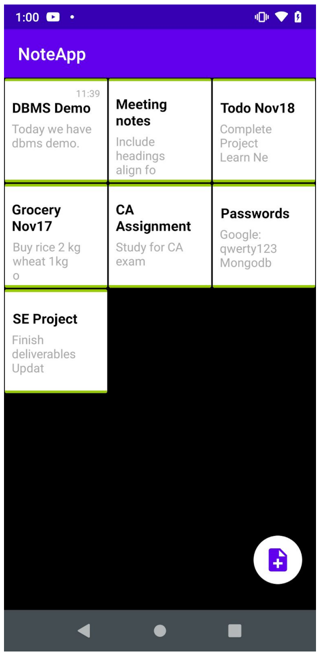
Database Description:
The database used for this project is MongoDB which is a NoSQL database that is a non-relational database. MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON- like documents with optional schemas.

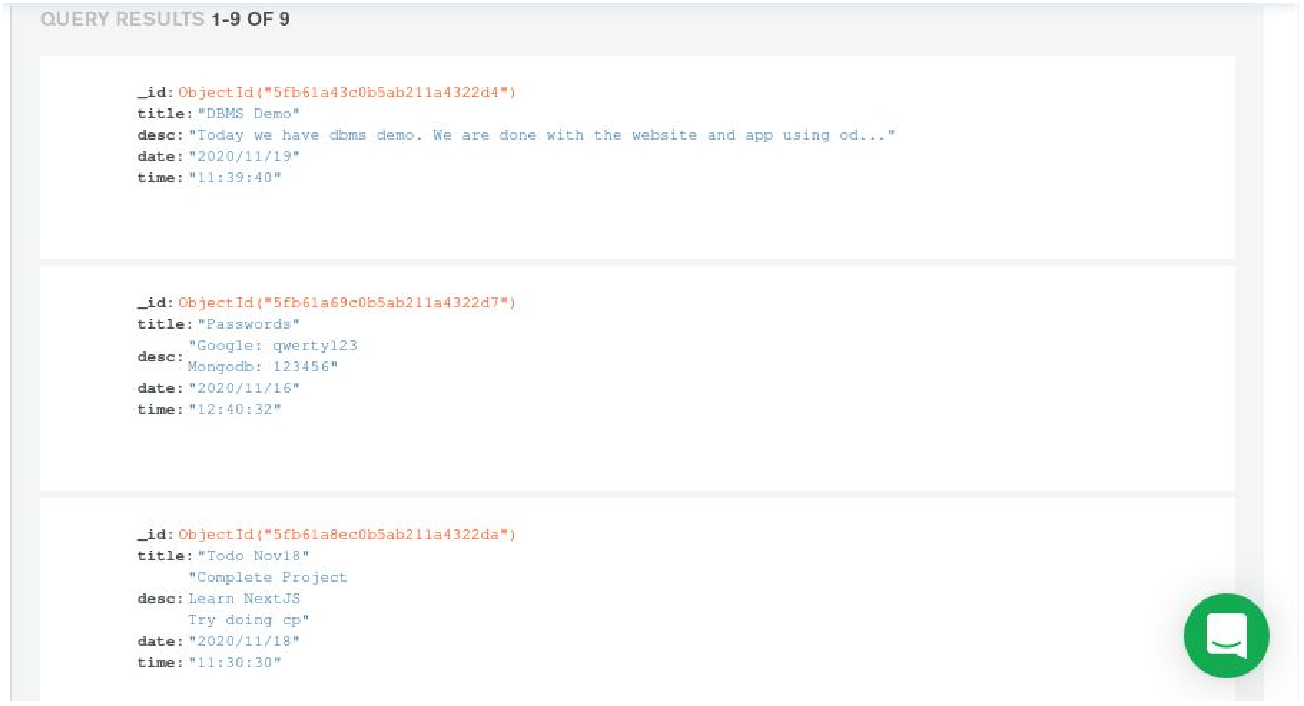
Database Models:

Note Schema:
```
{
"_id":{"$oid":"5fb61b5ec0b5ab211a4322e8"},
"title":"Meeting notes",
"desc":"Include headings \n align fonts \n work on new homepage \n sprint week 3",
"date":"2020/11/18",
"time":"12:44:10"
}
```

Implementation details:
With the help of the database schema, we have developed an app that can carry out all the features of a Note Making App. Some basic implementations are shown below:

1) Home Screen:
The user can view the saved notes (title and description) in a grid format. The user can also create a new note by clicking on the add-note present at the bottom right corner. The notes are sorted in descending order of modified time and the ones edited today will display the time of modification.





QUERY RESULTS 1-9 OF 9

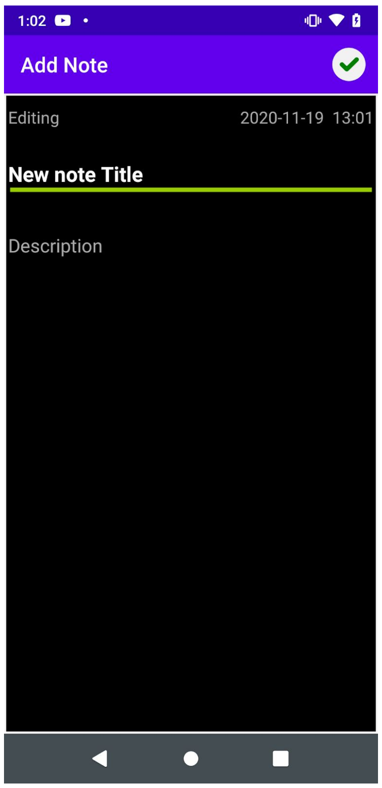
```
_id: ObjectId("5fb61a43c0b5ab211a4322d4")
title: "DBMS Demo"
desc: "Today we have dbms demo. We are done with the website and app using od..."
date: "2020/11/19"
time: "11:39:40"
```

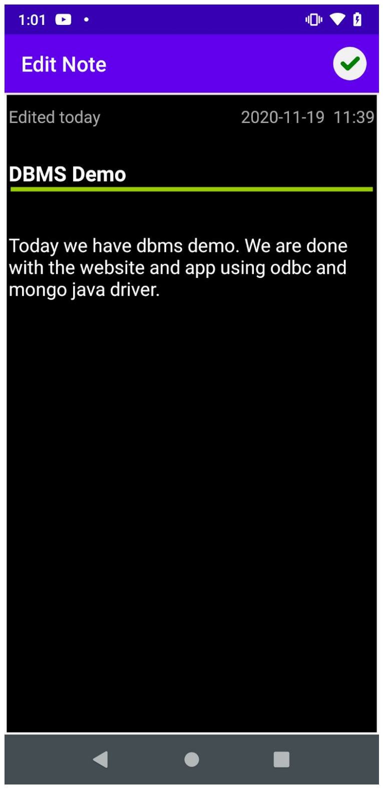
```
_id: ObjectId("5fb61a69c0b5ab211a4322d7")
title: "Passwords"
      "Google: qwerty123
desc: Mongodb: 123456"
date: "2020/11/16"
time: "12:40:32"
```

```
_id: ObjectId("5fb61a8ec0b5ab211a4322da")
title: "Todo Nov18"
      "Complete Project
desc: Learn NextJS
      Try doing cp"
date: "2020/11/18"
time: "11:30:30"
```

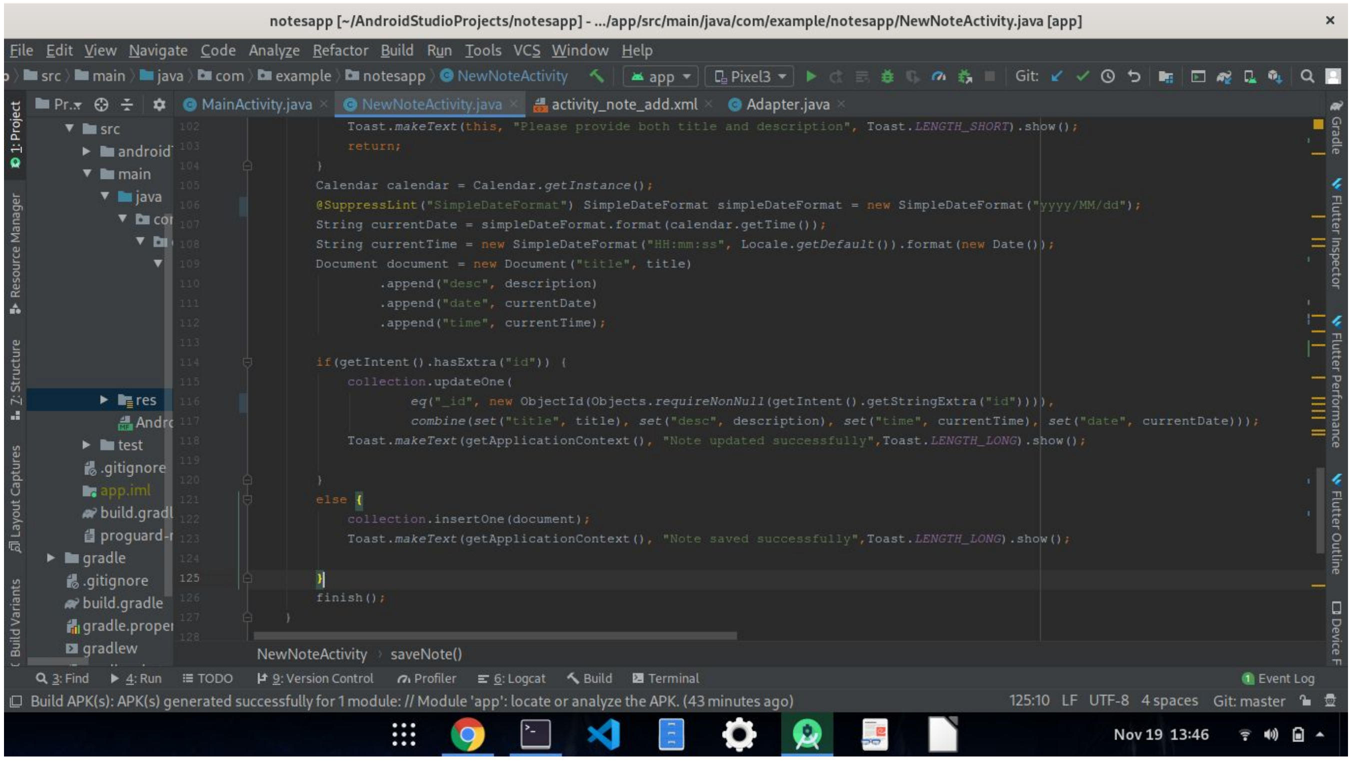
2) Add / Edit Note Screen:

In this screen, the user can add a new note by entering its title and description. The status is showing at the top corner and the current date and time on the right corner.





Today we have dbms demo. We are done with the website and app using odbc and mongo java driver.

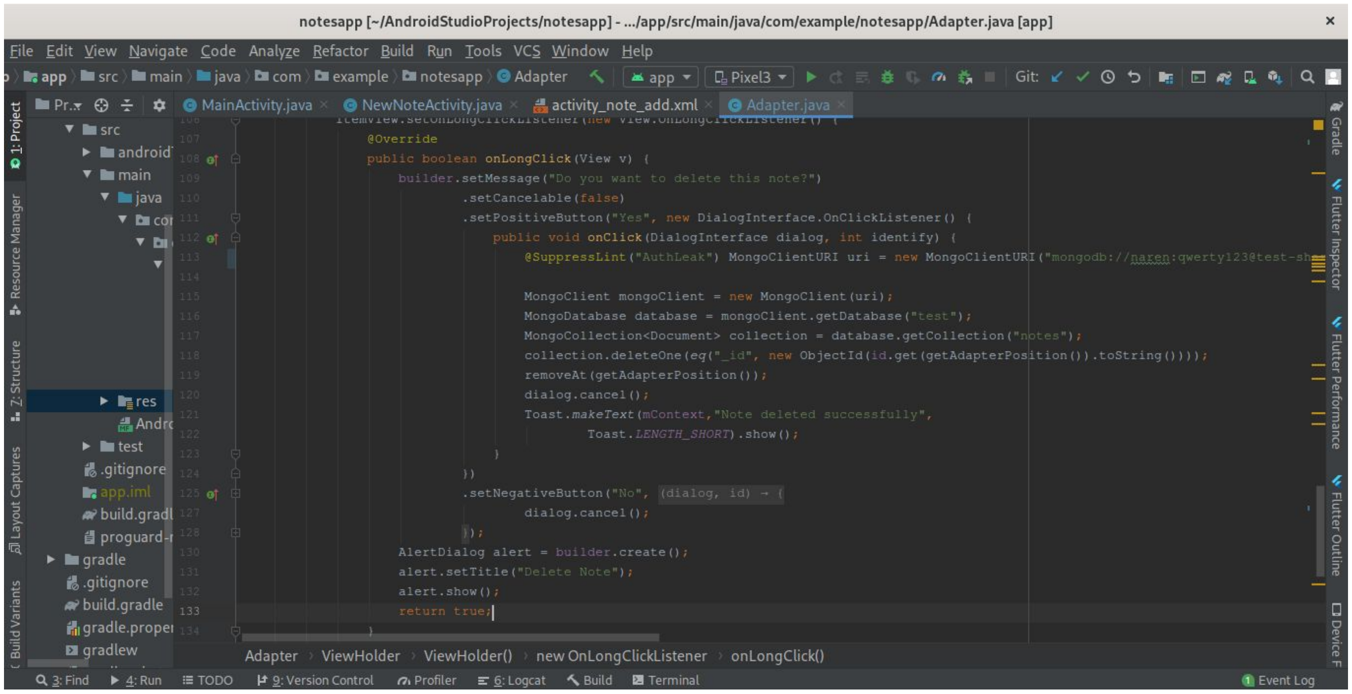

notesapp [~/AndroidStudioProjects/notesapp] - .../app/src/main/java/com/example/notesapp/NewNoteActivity.java [app]

```java
            Toast.makeText(this, "Please provide both title and description", Toast.LENGTH_SHORT).show();
            return;
        }
        Calendar calendar = Calendar.getInstance();
        @SuppressLint("SimpleDateFormat") SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy/MM/dd");
        String currentDate = simpleDateFormat.format(calendar.getTime());
        String currentTime = new SimpleDateFormat("HH:mm:ss", Locale.getDefault()).format(new Date());
        Document document = new Document("title", title)
                .append("desc", description)
                .append("date", currentDate)
                .append("time", currentTime);

        if(getIntent().hasExtra("id")) {
            collection.updateOne(
                    eq("_id", new ObjectId(Objects.requireNonNull(getIntent().getStringExtra("id")))),
                    combine(set("title", title), set("desc", description), set("time", currentTime), set("date", currentDate)));
            Toast.makeText(getApplicationContext(), "Note updated successfully",Toast.LENGTH_LONG).show();

        }
        else {
            collection.insertOne(document);
            Toast.makeText(getApplicationContext(), "Note saved successfully",Toast.LENGTH_LONG).show();

        }
        finish();
    }
```

Build APK(s): APK(s) generated successfully for 1 module: // Module 'app': locate or analyze the APK. (43 minutes ago)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src main java com example notesapp NewNoteActivity
app Pixel3 Git:
MainActivity.java
NewNoteActivity.java
activity_note_add.xml
Adapter.java
NewNoteActivity saveNote()
3: Find 4: Run TODO 9: Version Control Profiler 6: Logcat Build Terminal Event Log
125:10 LF UTF-8 4 spaces Git: master
Nov 19 13:46

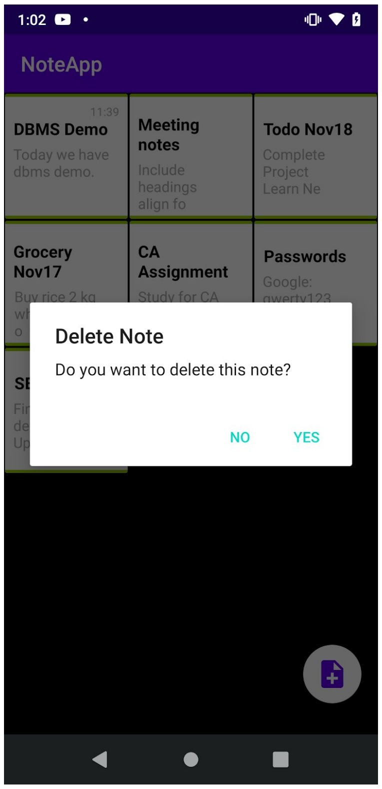
3) Delete Note:
When the user long presses any note, an alert dialog is displayed to delete the note.



notesapp [~/AndroidStudioProjects/notesapp] - .../app/src/main/java/com/example/notesapp/Adapter.java [app]

```java
            itemView.setOnLongClickListener(new View.OnLongClickListener() {
                @Override
                public boolean onLongClick(View v) {
                    builder.setMessage("Do you want to delete this note?")
                            .setCancelable(false)
                            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialog, int identify) {
                                    @SuppressLint("AuthLeak") MongoClientURI uri = new MongoClientURI("mongodb://naren:qwerty123@test-sh

                                    MongoClient mongoClient = new MongoClient(uri);
                                    MongoDatabase database = mongoClient.getDatabase("test");
                                    MongoCollection<Document> collection = database.getCollection("notes");
                                    collection.deleteOne(eq("_id", new ObjectId(id.get(getAdapterPosition()).toString())));
                                    removeAt(getAdapterPosition());
                                    dialog.cancel();
                                    Toast.makeText(mContext,"Note deleted successfully",
                                            Toast.LENGTH_SHORT).show();
                                }
                            })
                            .setNegativeButton("No", (dialog, id) → {
                                    dialog.cancel();
                            });
                    AlertDialog alert = builder.create();
                    alert.setTitle("Delete Note");
                    alert.show();
                    return true;
                }
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app src main java com example notesapp Adapter
MainActivity.java NewNoteActivity.java activity_note_add.xml Adapter.java
Adapter › ViewHolder › ViewHolder() › new OnLongClickListener › onLongClick()
3: Find 4: Run TODO 9: Version Control Profiler 6: Logcat Build Terminal Event Log



1:02
NoteApp
11:39
DBMS Demo
Today we have dbms demo.
Meeting notes
Include headings align fo
Todo Nov18
Complete Project Learn Ne
Grocery Nov17
Buy rice 2 kg
CA Assignment
Study for CA
Passwords
Google: qwerty123

Delete Note
Do you want to delete this note?
NO YES

Result:
With the help of the non-relational database (MongoDB) and mongo-java-driver, we have successfully developed a Note Making App which connects to MongoDB database, enabling users to create,update and manage notes, shopping lists and to-do lists.

SQL MINI PROJECT:

Online buying and selling platform

Description:
This is an online shopping platform where people can buy and sell services and goods such as electronics, fashion items, furniture, household goods, cars and bikes.
Project Features:
- Sign and Register for both customers and sellers.
- Provision to add, edit and delete products.
- Sort products by price and brands.
- Product purchase history.

Project Github link: https://github.com/ahmed-28/dbms-SQL-project/tree/master

Experimental setup:

Backend Technology Stack:

- Nodejs
- Express (Web Framework)
- ODBC (Database connectivity)

Frontend Technology Stack:

- HTML / CSS
- EJS template engine
- Semantic UI

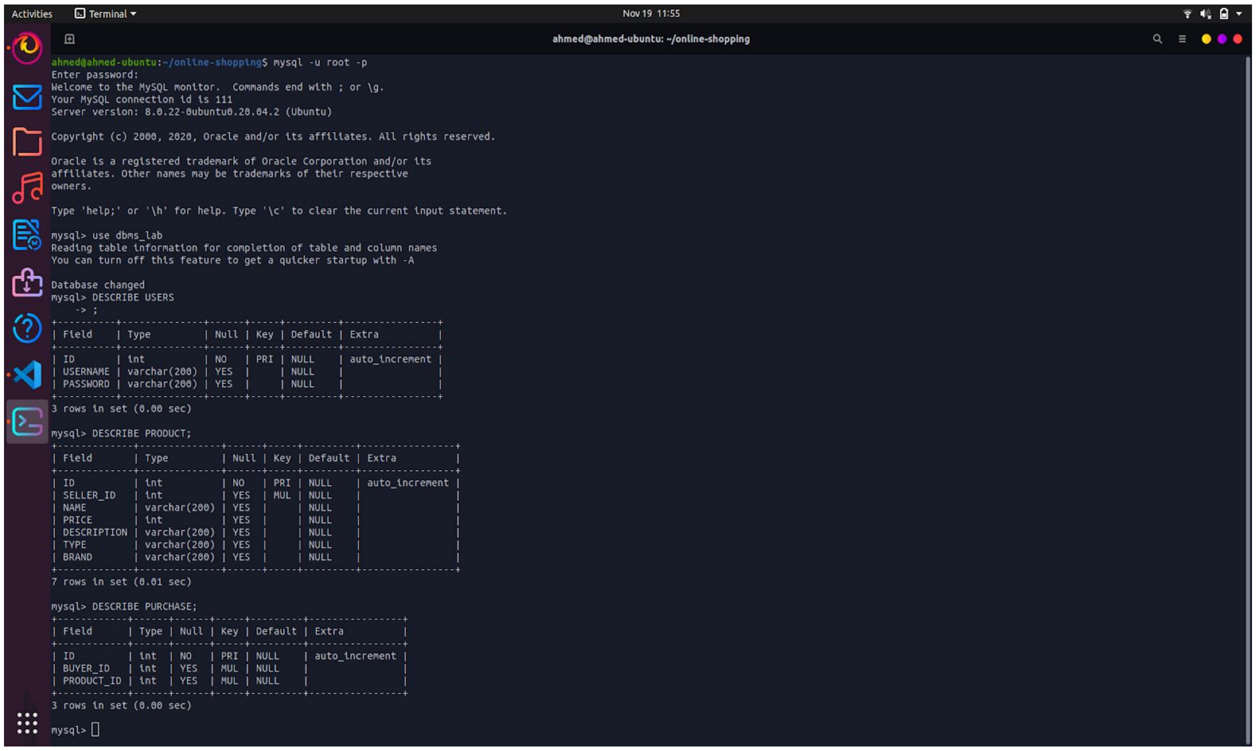
Database used:

- MySQL

The steps for complete setup and running of project is given in the github link

Database Description:

The tables used in the project are given in the below screenshots:



```
ahmed@ahmed-ubuntu:~/online-shopping$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 111
Server version: 8.0.22-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbms_lab
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> DESCRIBE USERS
    -> ;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| ID       | int          | NO   | PRI | NULL    | auto_increment |
| USERNAME | varchar(200) | YES  |     | NULL    |                |
| PASSWORD | varchar(200) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> DESCRIBE PRODUCT;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| ID          | int          | NO   | PRI | NULL    | auto_increment |
| SELLER_ID   | int          | YES  | MUL | NULL    |                |
| NAME        | varchar(200) | YES  |     | NULL    |                |
| PRICE       | int          | YES  |     | NULL    |                |
| DESCRIPTION | varchar(200) | YES  |     | NULL    |                |
| TYPE        | varchar(200) | YES  |     | NULL    |                |
| BRAND       | varchar(200) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)

mysql> DESCRIBE PURCHASE;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| ID         | int  | NO   | PRI | NULL    | auto_increment |
| BUYER_ID   | int  | YES  | MUL | NULL    |                |
| PRODUCT_ID | int  | YES  | MUL | NULL    |                |
+------------+------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> 
```

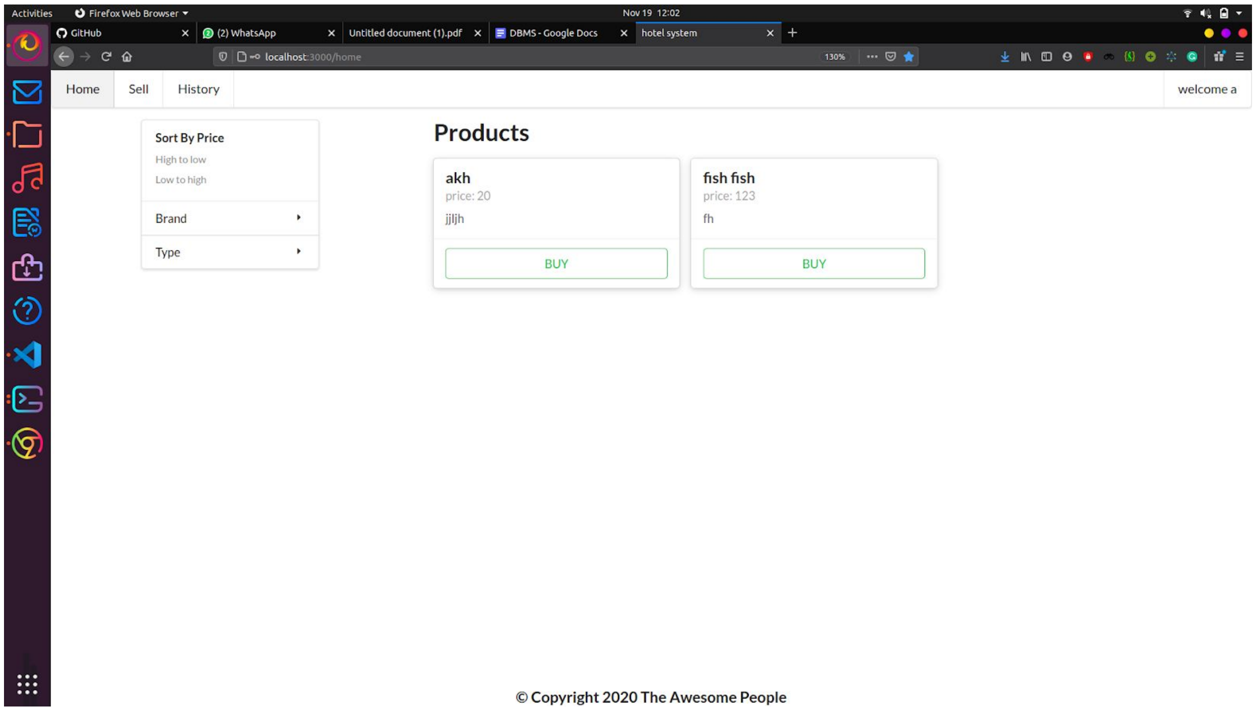
Creation of required tables:

```
const userTableQuery = `CREATE TABLE USERS(
                                ID INT PRIMARY KEY AUTO_INCREMENT,
                                USERNAME VARCHAR(200),
                                PASSWORD VARCHAR(200)
                                );`;
db.query(userTableQuery)
.then(res => console.log("users table created!!"))
.catch(err => console.log(err));


const productTableQuery = `CREATE TABLE PRODUCT(
                                ID INT PRIMARY KEY AUTO_INCREMENT,
                                SELLER_ID INT,
                                NAME VARCHAR(200),
                                PRICE INT,
                                DESCRIPTION VARCHAR(200),
                                TYPE VARCHAR(200),
                                BRAND VARCHAR(200),
                                FOREIGN KEY (SELLER_ID) REFERENCES
USERS(ID)
                                );`;

db.query(productTableQuery)
   .then(res => console.log("product table created!!"))
   .catch( (err) => {throw err} );

const purchaseTableQuery = `CREATE TABLE PURCHASE(
                                ID INT PRIMARY KEY AUTO_INCREMENT,
                                BUYER_ID INT,
                                SELLER_ID INT,
                                PRODUCT_ID INT,
                                FOREIGN KEY (PRODUCT_ID) REFERENCES
PRODUCT(ID) ON DELETE CASCADE,
                                FOREIGN KEY (BUYER_ID) REFERENCES USERS(ID)
ON DELETE CASCADE
                                );`;


db.query(purchaseTableQuery,(err,res)=>{
   if(err) throw err;
   else console.log("PURCHASE tbale created !!");
});
```

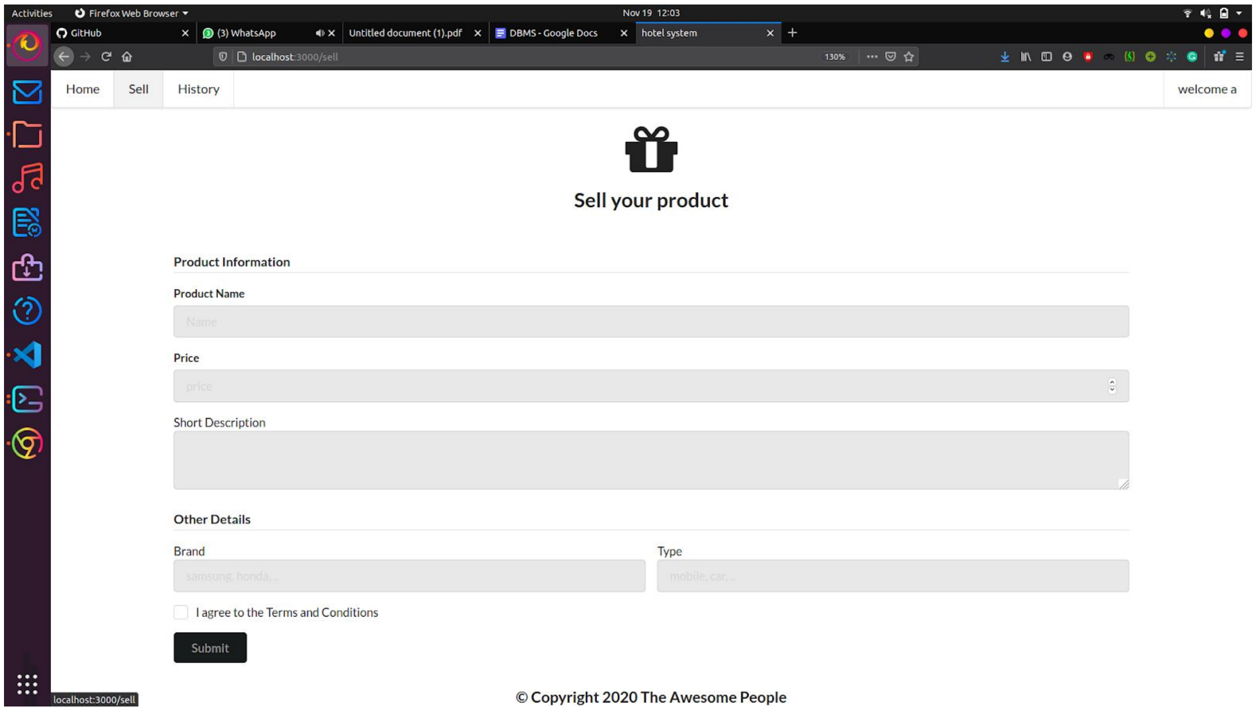
Implementation details:

1) Home page:
   In the home page, all the products are displayed to the buyer with options to sort the product list by price, brand and type.

   There are three options on the navigation bar.
   On clicking Sell, the user will be navigated to the product selling page where he/she will enter product specific details to upload the product.
   History page shows the user their purchase history and product sales.



2) Product selling page:
In this page, the seller will enter his/her product specific details such as name, price, description, brand and type to upload his product to the platform. The seller can later modify the fields of the product as per the needs.



Sell your product

Product Information

Product Name

Name

Price

price

Short Description

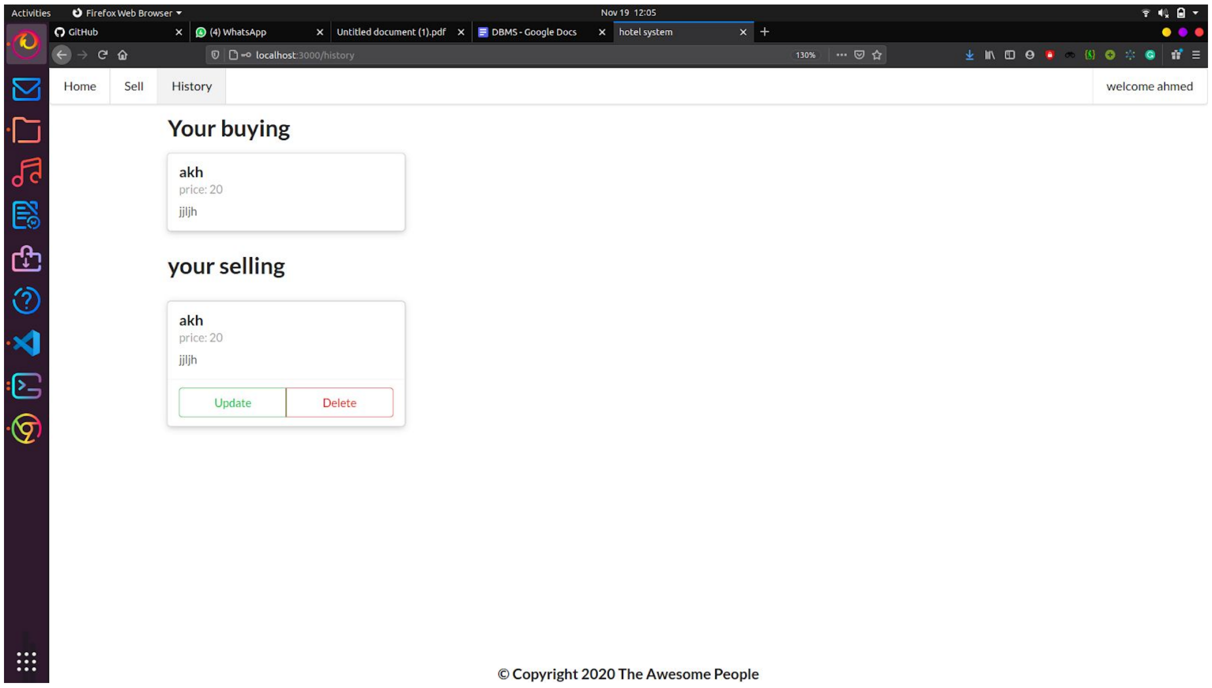
Other Details

Brand	Type
samsung, honda,...	mobile, car,...

I agree to the Terms and Conditions

© Copyright 2020 The Awesome People

3) History page:
In the history page, the user can view his purchase and selling history.



Your buying

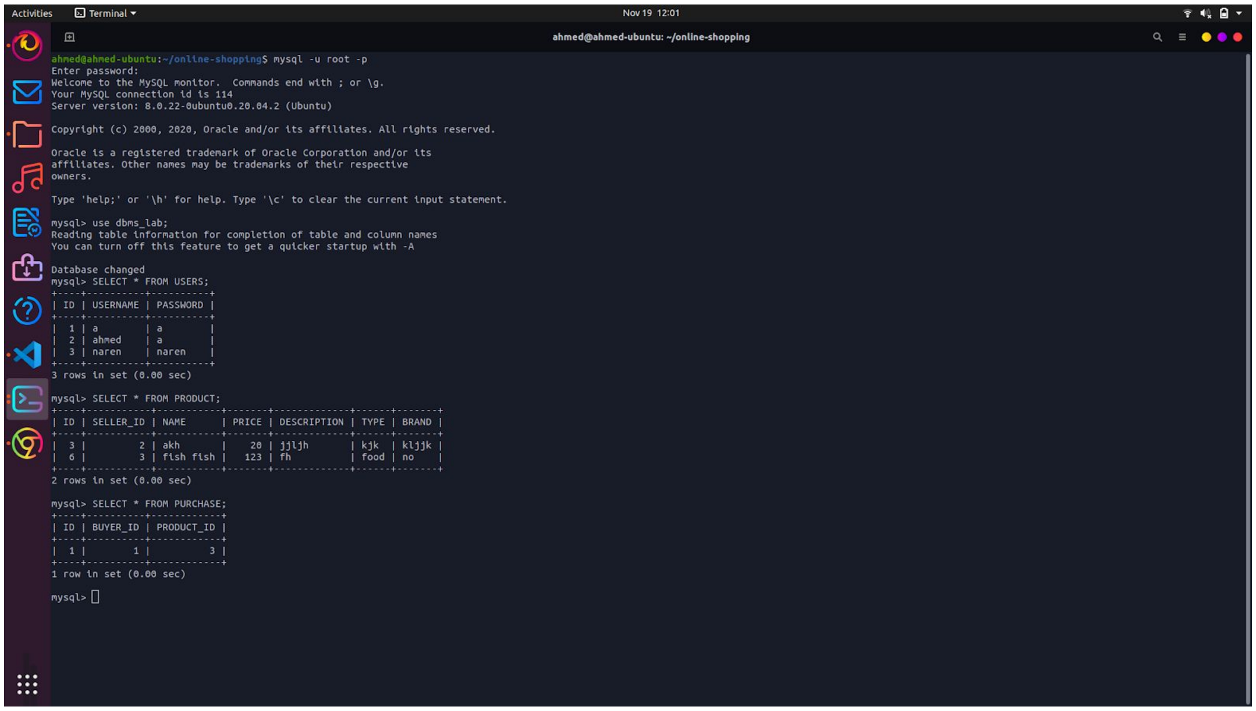
akh
price: 20
jjljh

your selling

akh
price: 20
jjljh

© Copyright 2020 The Awesome People

4) Database entries:



```
ahmed@ahmed-ubuntu:~/online-shopping$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 114
Server version: 8.0.22-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbms_lab;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM USERS;
+----+----------+----------+
| ID | USERNAME | PASSWORD |
+----+----------+----------+
|  1 | a        | a        |
|  2 | ahmed    | a        |
|  3 | naren    | naren    |
+----+----------+----------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM PRODUCT;
+----+-----------+-----------+-------+-------------+------+-------+
| ID | SELLER_ID | NAME      | PRICE | DESCRIPTION | TYPE | BRAND |
+----+-----------+-----------+-------+-------------+------+-------+
|  3 |         2 | akh       |    20 | jjljh       | kjk  | kljjk |
|  6 |         3 | fish fish |   123 | fh          | food | no    |
+----+-----------+-----------+-------+-------------+------+-------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM PURCHASE;
+----+----------+------------+
| ID | BUYER_ID | PRODUCT_ID |
+----+----------+------------+
|  1 |        1 |          3 |
+----+----------+------------+
1 row in set (0.00 sec)

mysql> 
```

Results:
With the help of the relational database (MySQL) and ODBC, we have successfully developed an online shopping platform which connects to MySQL database, enabling users to create accounts, buy/sell products and view purchase history.