Lesson 06 Demo 01

Facilitating Language Conversion Using Generative AI

Objective: To demonstrate the migration of the login functionality of an online retail platform from Java to JavaScript

Tools required: ChatGPT, Visual Studio, and GitHub Copilot

Prerequisites: Install MySQL in your system

Steps to be followed:
1. Convert the login functionality written in Java to JavaScript

Step 1: Convert the login functionality written in Java to JavaScript

1.1 Install MySQL and run it on the local machine. Create a database to store user data
Database Name: RetailPlatform
Below command is use to create the database

create database RetailPlatform;

Below command is use to move or switch inside a database.

use RetailPlatform;

```
mysql> create database RetailPlatform;
Query OK, 1 row affected (0.07 sec)

mysql> use RetailPlatform
Database changed
mysql>
```

Now we will create the table

Table Name: users
create table users(
    emailId varchar(25) primary key,
   **password varchar(25));**

Insert some test data directly via MySQL as follows:

INSERT INTO users (emailId, password) VALUES
   ('admin@gmail.com', 'admin@123'),
   ('user1@gmail.com', 'user1@123');

```
mysql> create table users(
    ->       emailId varchar(25) primary key,
    ->       password varchar(25));
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO users (emailId, password) VALUES
    ->       ('admin@gmail.com', 'admin@123'),
    ->       ('user1@gmail.com', 'user1@123');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
```

1.2 Download Java Project folder RetailPlatform_java_app, extract and open inside a VS code Editor
   After open you can see the project structure as
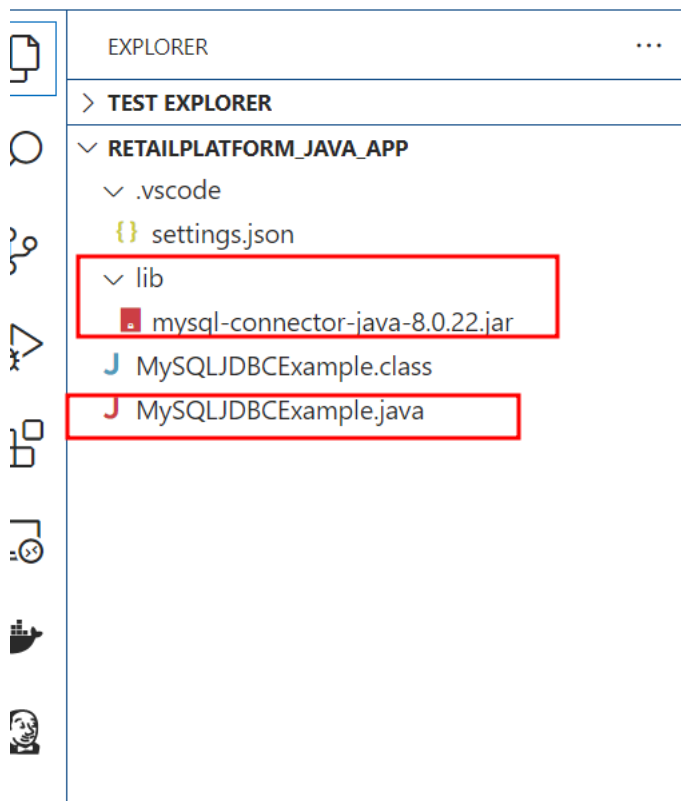
MySQLJDBCExample.java code

```java
// Source code is decompiled from a .class file using FernFlower decompiler.
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class MyJavaFile {
  private Connection connection;

  public MyJavaFile() {
    try {
      System.out.println("I came her");
      this.connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/RetailPlatform", "root",
"root@123");
      System.out.println("Connected to MySQL database!");
    } catch (Exception var2) {
      System.out.println("Error connecting to MySQL: " + var2);
    }

  }

  public static void main(String[] var0) {
    MyJavaFile var1 = new MyJavaFile();
    Scanner var2 = new Scanner(System.in);
    System.out.print("Enter emailId: ");
    String var3 = var2.nextLine();
    System.out.print("Enter password: ");
    String var4 = var2.nextLine();
    if (var1.authenticate(var3, var4)) {
      System.out.println("Login successful!");
    } else {
      System.out.println("Login failed!");
    }

    var2.close();
  }

  private boolean authenticate(String var1, String var2) {
    if (this.connection == null) {
      System.out.println("MySQL connection not established");
```

```java
        return false;
      } else {
        PreparedStatement var3 = null;
        ResultSet var4 = null;

        try {
          String var5 = "SELECT * FROM users WHERE emailId = ? AND password = ?";
          var3 = this.connection.prepareStatement(var5);
          var3.setString(1, var1);
          var3.setString(2, var2);
          var4 = var3.executeQuery();
          boolean var6 = var4.next();
          return var6;
        } catch (SQLException var16) {
          System.out.println("Error executing SQL query: " + var16.getMessage());
        } finally {
          try {
            if (var4 != null) {
              var4.close();
            }

            if (var3 != null) {
              var3.close();
            }
          } catch (SQLException var15) {
            System.out.println("Error closing resources: " + var15.getMessage());
          }

        }

        return false;
      }
    }
  }
```

In the file change your database name, username and password depending upon MySQL Database configuration details

```
J MySQLJDBCExample.java ×

J MySQLJDBCExample.java > 🐾 MySQLJDBCExample > ⊘ main(String[])
 1    import java.sql.*;
 2    import java.util.Scanner;
 3
 4    public class MySQLJDBCExample {
      Run | Debug
 5        public static void main(String[] args) {
 6            String url = "jdbc:mysql://localhost:3306/RetailPlatform";  // Change this to your database URL
 7            String user = "root";                                      // Change this to your database username
 8            String password = "root@123";                              // Change this to your database password
 9
10            try {
11                // Load MySQL JDBC Driver
12                Class.forName(className:"com.mysql.cj.jdbc.Driver");
13
14                // Establish connection
15                Connection conn = DriverManager.getConnection(url, user, password);
16                System.out.println(x:"Connected to MySQL successfully!");
17
18                // Execute a query
19                PreparedStatement preparedStatement = conn.prepareStatement(sql:"SELECT * FROM users WHERE emailId = ? AN
20                Scanner sc = new Scanner(System.in);
21                System.out.println(x:"Enter emailId: ");
22                String emailId = sc.nextLine();
23                System.out.println(x:"Enter password: ");
```

Run the following Java code in Visual studio code to connect with MySQL to perform authentication:

Then click on run option

```
 2    import java.util.Scanner;
 3
 4    public class MySQLJDBCExample {
      Run | Debug
 5        public static void main(String[] args) {
 6            String url = "jdbc:mysql://localhost:3306/RetailPlatform";  // Change this to your database URL
 7            String user = "root";                                      // Change this to your database username
 8            String password = "root@123";                              // Change this to your database password
 9
10            try {
11                // Load MySQL JDBC Driver
12                Class.forName(className:"com.mysql.cj.jdbc.Driver");
13
14                // Establish connection
15                Connection conn = DriverManager.getConnection(url, user, password);
16                System.out.println(x:"Connected to MySQL successfully!");
17
18                // Execute a query
19                PreparedStatement preparedStatement = conn.prepareStatement(sql:"SELECT * FROM users WHERE emailId = ? AN
20                Scanner sc = new Scanner(System.in);
```

It will ask you emailId and password

```
∨ TERMINAL

  PS C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\RetailPlatform_java_ap>
  & 'C:\Program Files\Java\jdk-17.0.9\bin\java.exe' '@C:\Users\akash\AppData\Local\Temp\cp_579p0wen30lsq7gt2wwk90ql
  c.argfile' 'MySQLJDBCExample'
  Connected to MySQL successfully!
  Enter emailId:
```

admin@gmail.com   as emailid
admin@123   as password

```
PS C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\RetailPlatform_java_ap>
& 'C:\Program Files\Java\jdk-17.0.9\bin\java.exe' '@C:\Users\akash\AppData\Local\Temp\cp_579p0wen30lsq7gt2wwk90ql
c.argfile' 'MySQLJDBCExample'
Connected to MySQL successfully!
Enter emailId:
admin@gmail.com
Enter password:
admin@123
Login successful!
PS C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\RetailPlatform_java_app>
```

If you run once again with wrong emailid and password.

```
c
PS C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MER
PS C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MER
c:; cd 'c:\Users\akash\Desktop\Project with Gen AI\Demos - Updated wit
app'; & 'C:\Program Files\Java\jdk-17.0.9\bin\java.exe' '@C:\Users\aka
wk90qlc.argfile' 'MySQLJDBCExample'
Connected to MySQL successfully!
Enter emailId:
user4@gmail.com
Enter password:
user4@123555
Invalid emailId or password.
PS C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MER
```

1.3 Install the GitHub Copilot extension. Once it is installed, it appears as follows:
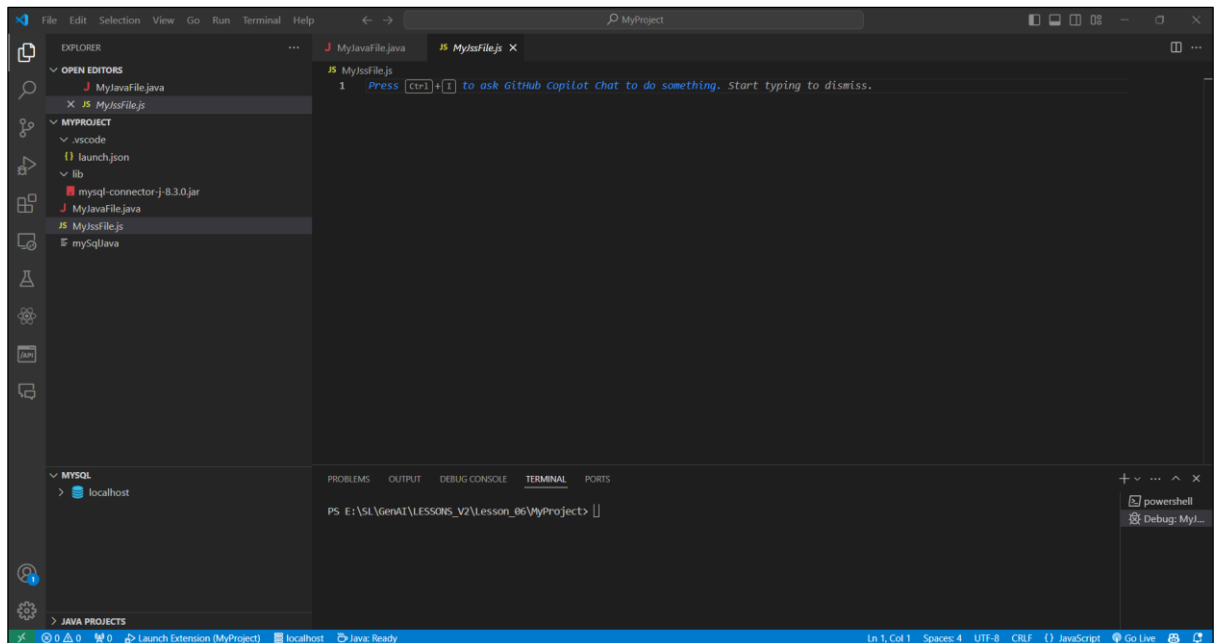
1.4 Use **Ctrl+I** to open the GitHub Copilot prompt

1.5 Use the following prompt along with the Java code to convert it into JavaScript:
Convert the provided java code, which interacts with a MySQL database to JavaScript using Node.js and the mysql package
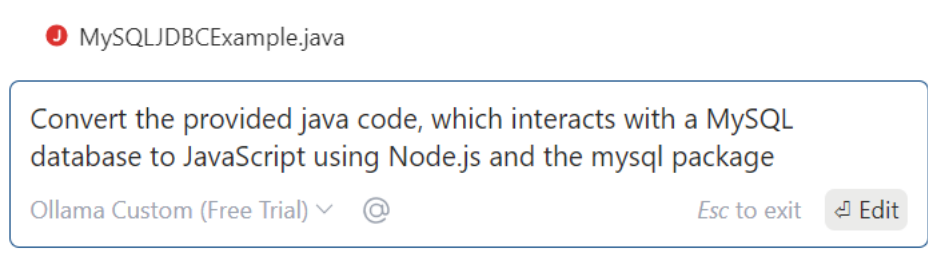
> `{}` package.json   📋 Copy   ▷ Apply

     2. Now, let's create the main Node.js file:

> **JS** mysqlExample.js   📋 Copy   ▷ Apply

     3. Create a .gitignore file to exclude node_modules:

> ◆ .gitignore   📋 Copy   ▷ Apply

Summary of key changes:

     1. Replaced JDBC with Node.js mysql2 package for better async support
     2. Converted Java's Scanner to readline-sync for synchronous console input
     3. Implemented promise-based MySQL connection and query execution
     4. Simplified error handling and connection management

To run this code:

     1. Run `npm install` to install dependencies
     2. Execute `node mysqlExample.js` to run the application

📋 👍 👎

1.6 apply all these changes or follow the instruction which help to create Java Code into node js to connect mysql database.

Summary of key changes:

Replaced JDBC with Node.js mysql2 package for better async support
Converted Java's Scanner to readline-sync for synchronous console input
Implemented promise-based MySQL connection and query execution
Simplified error handling and connection management
To run this code:

Run **npm install** to install dependencies
Execute **node mysqlExample.js** to run the application

Open the external terminal and execute above two commands.

1.7 Execute the converted JavaScript code and verify the output of both Java and
    converted JavaScript using the following command: