

FOOD ORDERING PLATFORM

-Sakshi Jaiswal

SOFTWARE REQUIREMENTS:

- Java Development Kit (JDK): Version 8 or above.
- Apache Maven: Version 3.x.
- MySQL: Version 5.x or above (with JDBC driver).
- Visual Studio Code: Latest version.
- Tomcat Server: Version 9.x or compatible.
- Git: Optional, for version control.

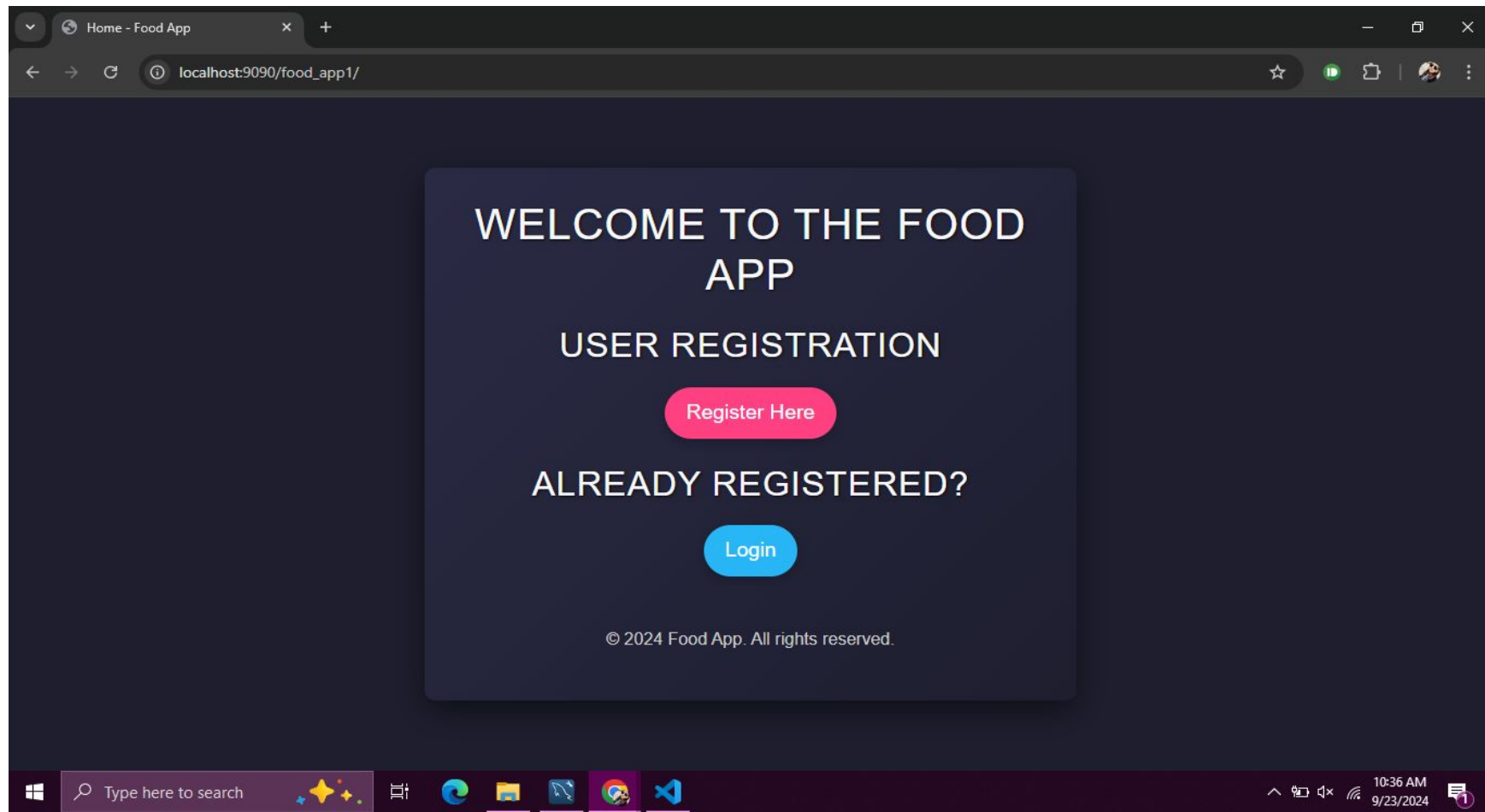
Requirements:

- JAVA HOME PATH (JDK8 and above)
- MAVEN HOME PATH
- set up database schema
- Servlet api, mysqlconnector dependency, apache tomcat server plugin : pom.xml

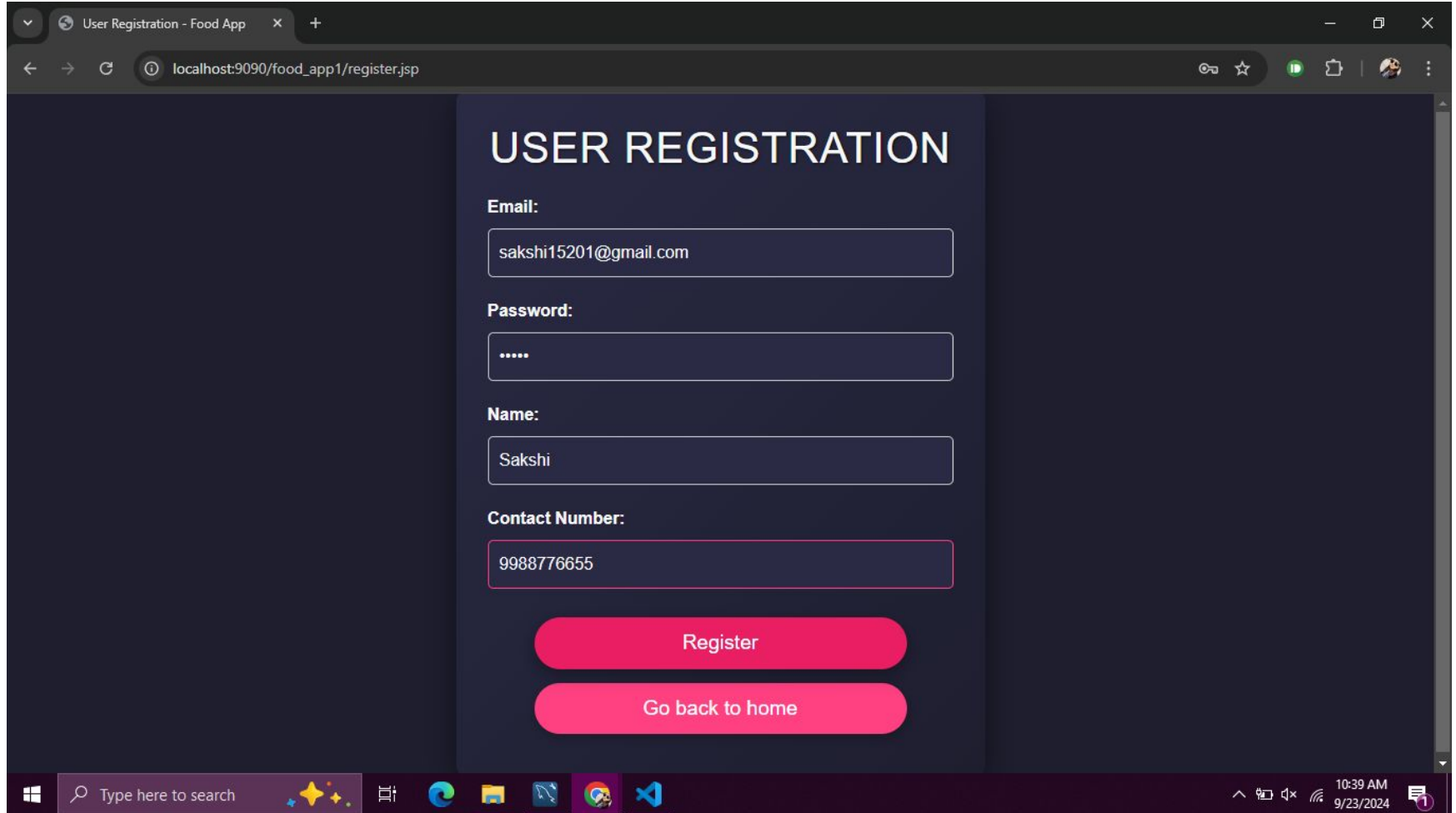
Required Dependencies and plugins present in 'pom.xml' file

```
pom.xml
3  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
22  <dependencies>
33  </dependencies>
34  <dependency>
35  |   <groupId>mysql</groupId>
36  |   <artifactId>mysql-connector-java</artifactId>
37  |   <version>8.0.32</version>
38  </dependency>
39  <dependency>
40  |   <groupId>javax.servlet</groupId>
41  |   <artifactId>javax.servlet-api</artifactId>
42  |   <version>4.0.1</version> <!-- You can use a different version if necessary -->
43  |   <scope>provided</scope>
44  </dependency>
45  </dependencies>
46  <build>
47  |   <finalName>food_app1</finalName>
48  |   <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
49  |   <plugins>
50  |   |   <plugin>
51  |   |   |   <groupId>org.apache.tomcat.maven</groupId>
52  |   |   |   <artifactId>tomcat7-maven-plugin</artifactId>
53  |   |   |   <version>2.2</version>
54  |   |   |   <configuration>
55  |   |   |   |   <server>Tomcat</server>
56  |   |   |   |   <port>9090</port>
57  |   |   |   </configuration>
58  |   |   </plugin>
59  |   </plugins>
```

Index Page



Register



The screenshot shows a web browser window with the title "User Registration - Food App". The address bar displays "localhost:9090/food_app1/register.jsp". The page content is a dark-themed registration form with the title "USER REGISTRATION" in large white letters. The form includes four input fields: "Email:" with the value "sakshi15201@gmail.com", "Password:" with masked characters "*****", "Name:" with the value "Sakshi", and "Contact Number:" with the value "9988776655". Below the input fields are two red buttons: "Register" and "Go back to home". The Windows taskbar is visible at the bottom, showing the search bar and several application icons. The system clock in the bottom right corner indicates the time is 10:39 AM on 9/23/2024.

User Registration - Food App

localhost:9090/food_app1/register.jsp

USER REGISTRATION

Email:

sakshi15201@gmail.com

Password:

Name:

Sakshi

Contact Number:

9988776655

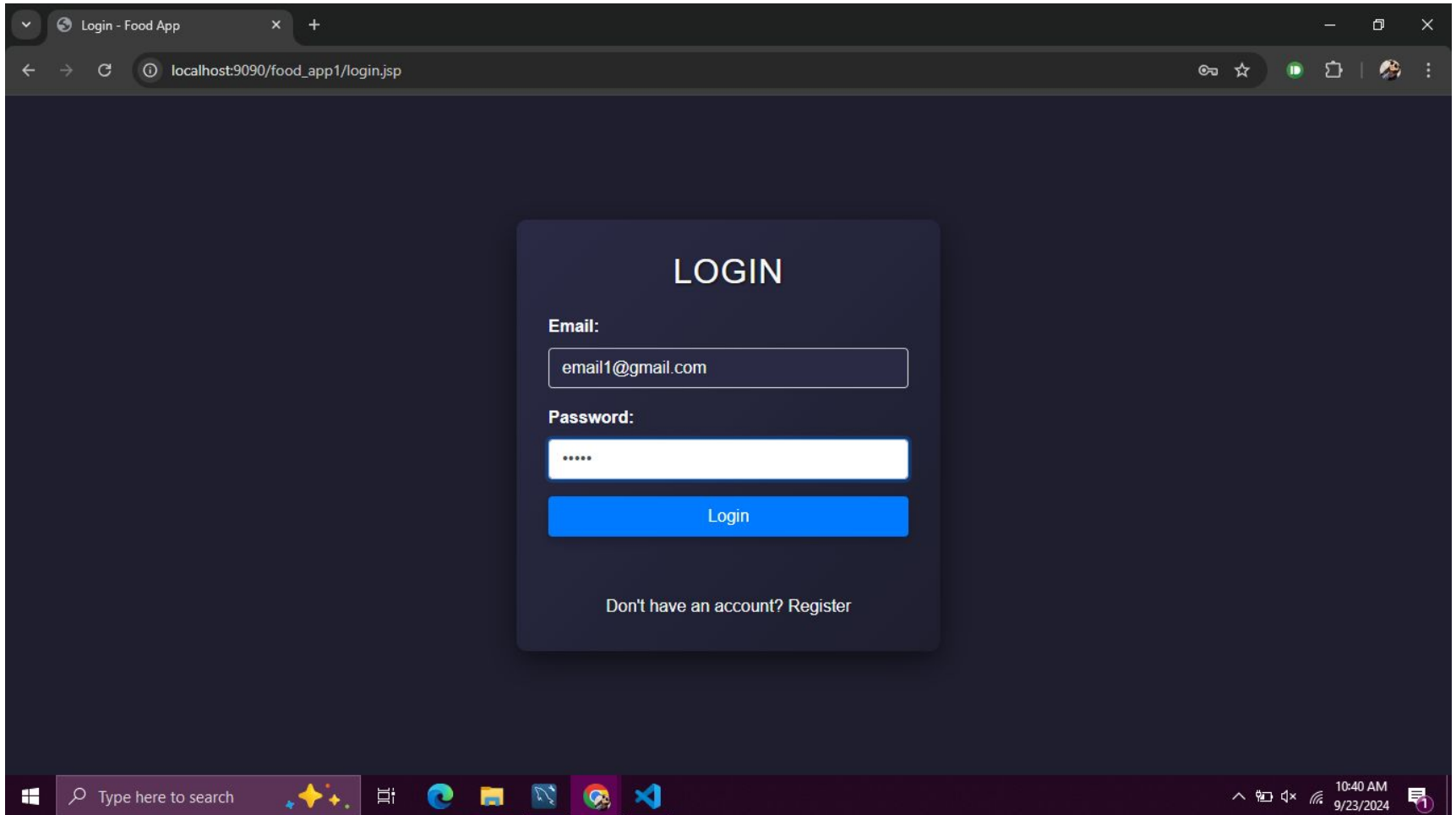
Register

Go back to home

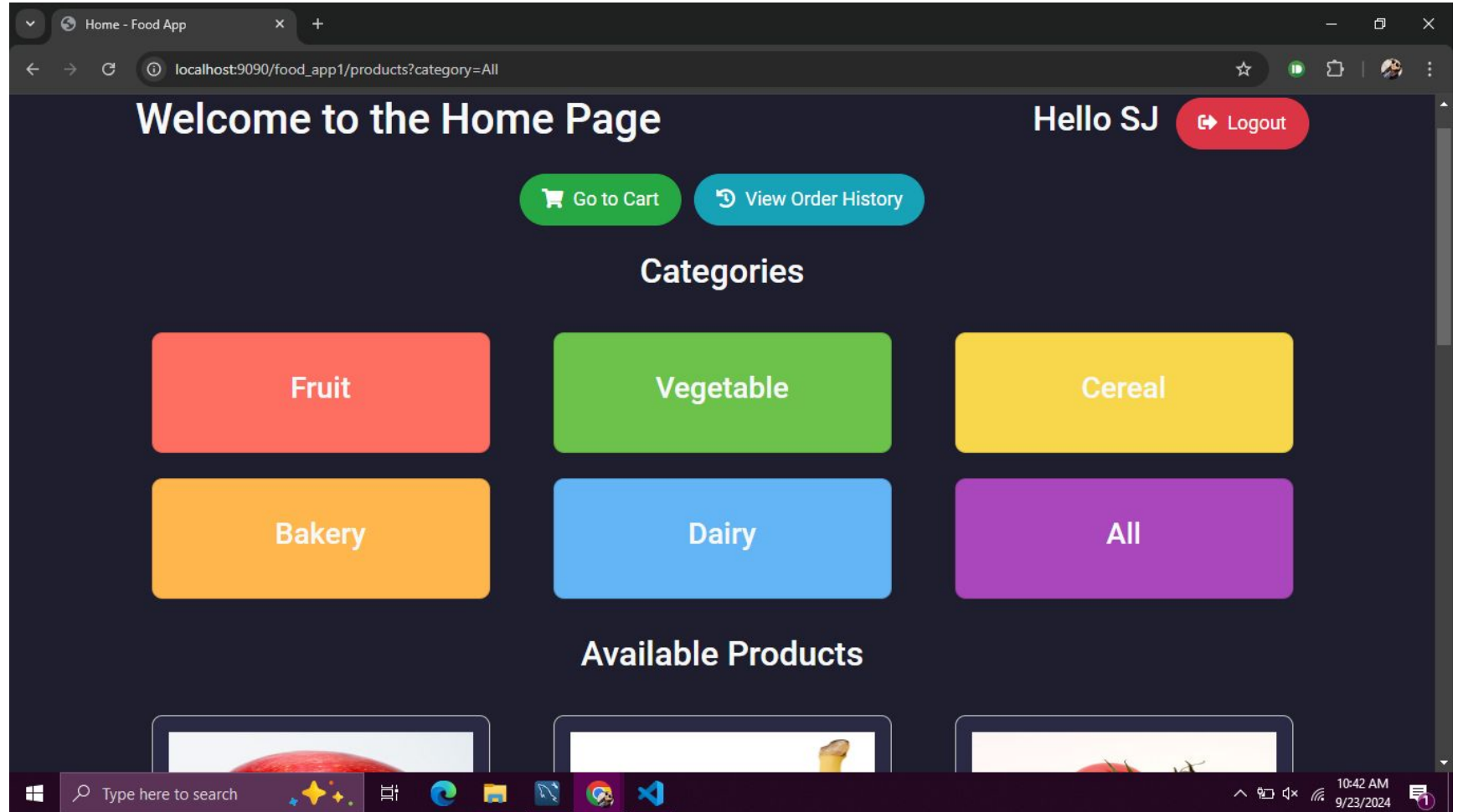
Type here to search

10:39 AM
9/23/2024

Login



Home



Fruit Products

Home - Food App


localhost:9090/food_app1/products?category=Fruit

Bakery

Dairy

All

Available Products




Apple

Price: \$0.5

In Cart: 1

Add to Cart



Banana

Price: \$0.3

In Cart: 0

Add to Cart

Type here to search

10:44 AM

9/23/2024

Cart




Your Cart

localhost:9090/food_app1/cart.jsp

Your Cart

Continue Shopping

Cart Summary

	Product: Apple (x1) Price: \$.50 x 1 = \$.50	Remove
	Product: Banana (x2) Price: \$.30 x 2 = \$.60	Remove
	Product: Milk (x1) Price: \$1.00 x 1 = \$1.00	Remove

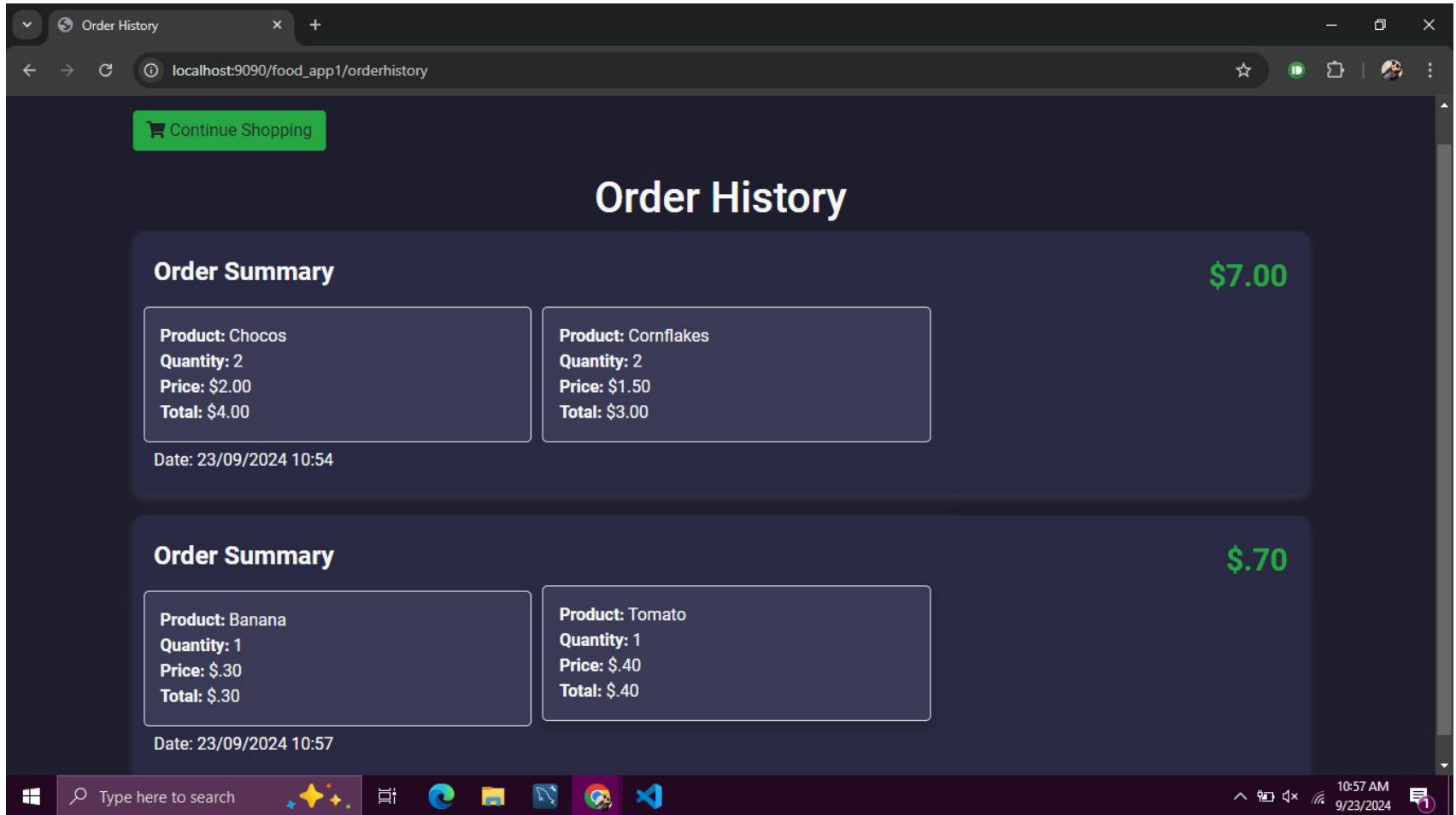
Total Amount Payable: \$2.10

Proceed to Checkout

Windows Taskbar

10:48 AM 9/23/2024

Order Hist



MySQL Database

```
• CREATE TABLE user_cred (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    name VARCHAR(100) NOT NULL,  
    contact_number VARCHAR(15) NOT NULL  
);
```

```
• CREATE TABLE products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    category VARCHAR(100) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    image_filename VARCHAR(255) NOT NULL  
);
```

```
• CREATE TABLE order_summaries (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT,  
    products JSON,  
    total_amount DECIMAL(10, 2),  
    order_date DATETIME,  
    FOREIGN KEY (id) REFERENCES user_cred(id)  
);
```

```
• CREATE TABLE user_carts (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT,  
    product_id INT, -- Add this column for product IDs  
    quantity INT, -- Add this column for product quantities  
    FOREIGN KEY (user_id) REFERENCES user_cred(id)  
);
```

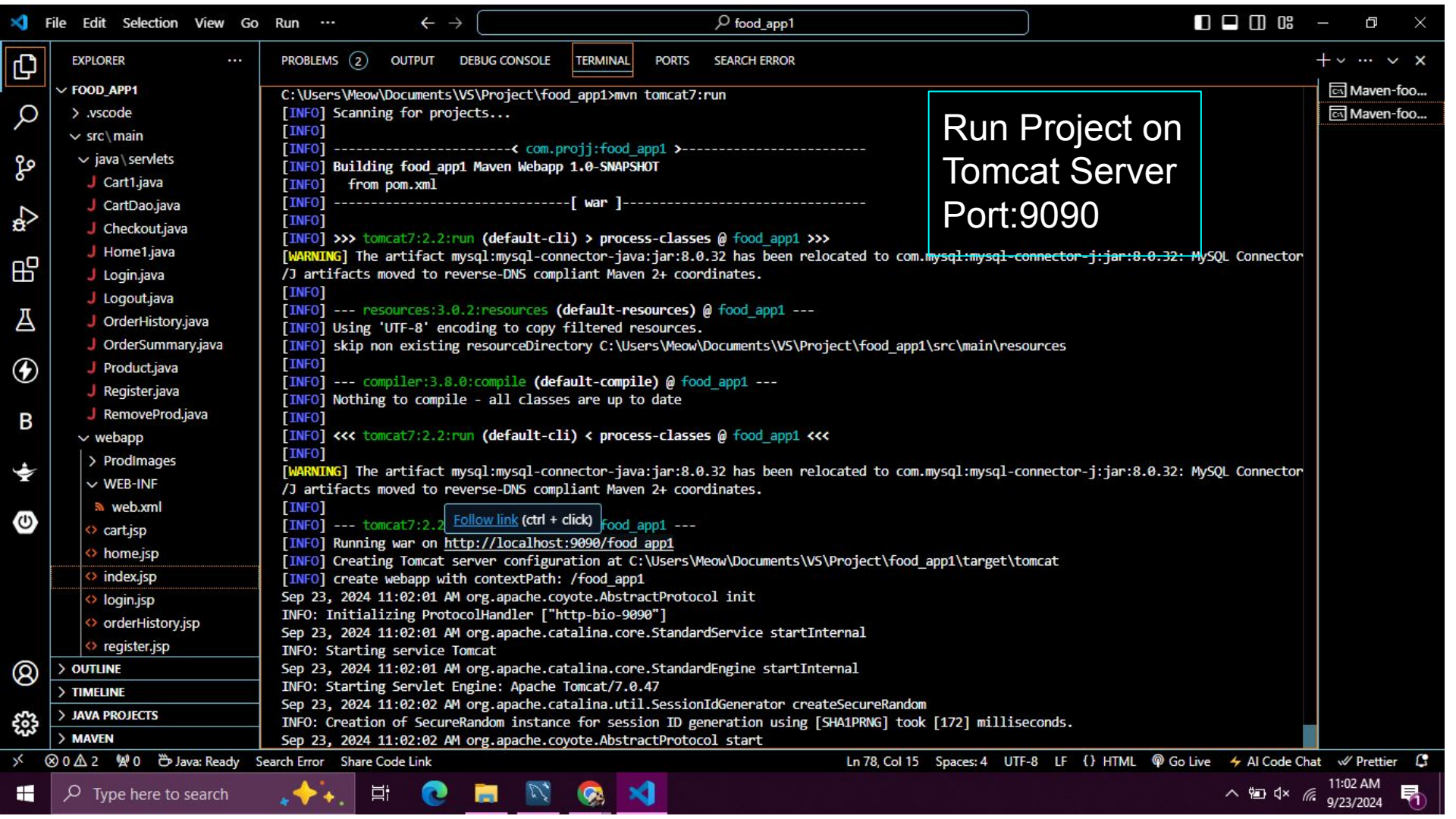
EXPLORER

- FOOD_APP1
 - .vscode
 - src/main
 - java/servlets
 - Cart1.java
 - CartDao.java
 - Checkout.java
 - Home1.java
 - Login.java
 - Logout.java
 - OrderHistory.java
 - OrderSummary.java
 - Product.java
 - Register.java
 - RemoveProd.java
 - webapp
 - ProdImages
 - WEB-INF
 - web.xml
 - cart.jsp
 - home.jsp
 - index.jsp
 - login.jsp
 - orderHistory.jsp
 - register.jsp
 - OUTLINE
 - TIMELINE
 - JAVA PROJECTS
 - MAVEN

```
C:\Users\Meow\Documents\VS\Project\food_app1>mvn clean install
[INFO] Scanning for projects...
[INFO]
-----< com.proj:food_app1 >-----
[INFO] Building food_app1 Maven Webapp 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO]
-----[ war ]-----
[WARNING] The artifact mysql:mysql-connector-java:jar:8.0.32 has been relocated to com.mysql:mysql-connector-j:jar:8.0.32: MySQL Connector
/J artifacts moved to reverse-DNS compliant Maven 2+ coordinates.
[INFO]
--- clean:3.1.0:clean (default-clean) @ food_app1 ---
[INFO] Deleting C:\Users\Meow\Documents\VS\Project\food_app1\target
[INFO]
--- resources:3.0.2:resources (default-resources) @ food_app1 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Meow\Documents\VS\Project\food_app1\src\main\resources
[INFO]
--- compiler:3.8.0:compile (default-compile) @ food_app1 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 11 source files to C:\Users\Meow\Documents\VS\Project\food_app1\target\classes
[INFO] /C:/Users/Meow/Documents/VS/Project/food_app1/src/main/java/servlets/OrderHistory.java: Some input files use unchecked or unsafe op
erations.
[INFO] /C:/Users/Meow/Documents/VS/Project/food_app1/src/main/java/servlets/OrderHistory.java: Recompile with -Xlint:unchecked for details
.
[INFO]
--- resources:3.0.2:testResources (default-testResources) @ food_app1 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Meow\Documents\VS\Project\food_app1\src\test\resources
[INFO]
--- compiler:3.8.0:testCompile (default-testCompile) @ food_app1 ---
[INFO] No sources to compile
[INFO]
--- surefire:2.22.1:test (default-test) @ food_app1 ---
[INFO] No tests to run.
[INFO]
--- war:3.2.2:war (default-war) @ food_app1 ---
[INFO] Packaging webapp
```

Maven Clean and Compile Project

- Maven-foo...
- Maven-foo...



Run Project on
Tomcat Server
Port:9090

Design Structure of the Food Ordering Application

The design structure of the Food Ordering Application follows a standard Model-View-Controller (MVC) architecture, which separates the application into three interconnected components, enhancing modularity, maintainability, and scalability.

1. Model

The Model represents the application's data and business logic. It is responsible for accessing and managing data, including interaction with the database.

- **Product.java:** Represents the product entity with properties such as ID, name, price, and image filename
- **CartDao.java:** Handles operations related to the user's cart, including loading and saving cart data to the database.
- **OrderSummaryDao.java:** Manages the storage and retrieval of order summaries for users.
- **UserServlet.java:** Handles user-related operations, including registration and login.

2. View

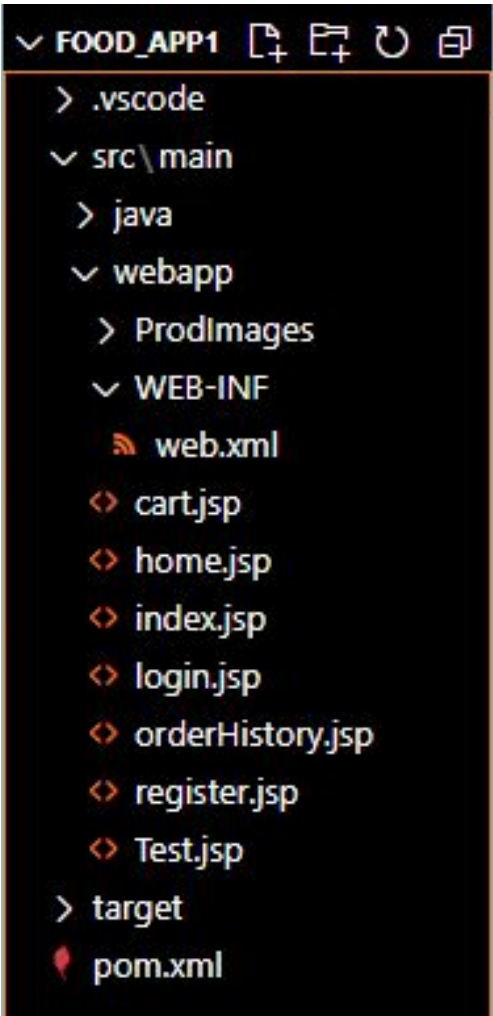
The View is responsible for displaying the user interface and presenting data to the user. In this application, the Views are implemented using JSP (JavaServer Pages).

- **home.jsp:** Displays the welcome message, available categories, and products. It also shows the user's cart and order history options.
- **login.jsp:** Provides a login interface for users to access their accounts.
- **registration.jsp:** Allows new users to register for an account.
- **cart.jsp:** Displays the items in the user's cart and allows modifications.
- **orderHistory.jsp:** Shows all past orders made by the user, regardless of the session.

3. Controller

The Controller acts as an intermediary between the Model and the View, processing user inputs and interactions. It handles the application's flow, managing requests and responses.

- **ProductServlet.java:** Controls the flow of product-related requests, such as displaying products based on categories.
- **UserServlet.java:** Manages user authentication, including login and registration processes.
- **CartDao.java:** Although primarily a data access object, it also contains logic for managing the cart state.



Project Structure Overview

- **src/main/java/servlets**: Contains the Java classes for the Model and Controllers, including DAO classes and servlets.
- **src/main/webapp**: Contains all JSP files for the View, along with any images used in the application.
- **src/main/webapp/WEB-INF**: Contains the **web.xml** file, which defines the application configuration and servlet mappings.