

FoodBox Project Source Code

Version History

1	Author Name	Himanshu Yadav
2	Purpose	Source Code Screenshot
3	Date	22-12-2022
4	Version	1.8
5	Contact	himanshu.hby.yadav@gmail.com

Contents

User Flow Chart Diagram

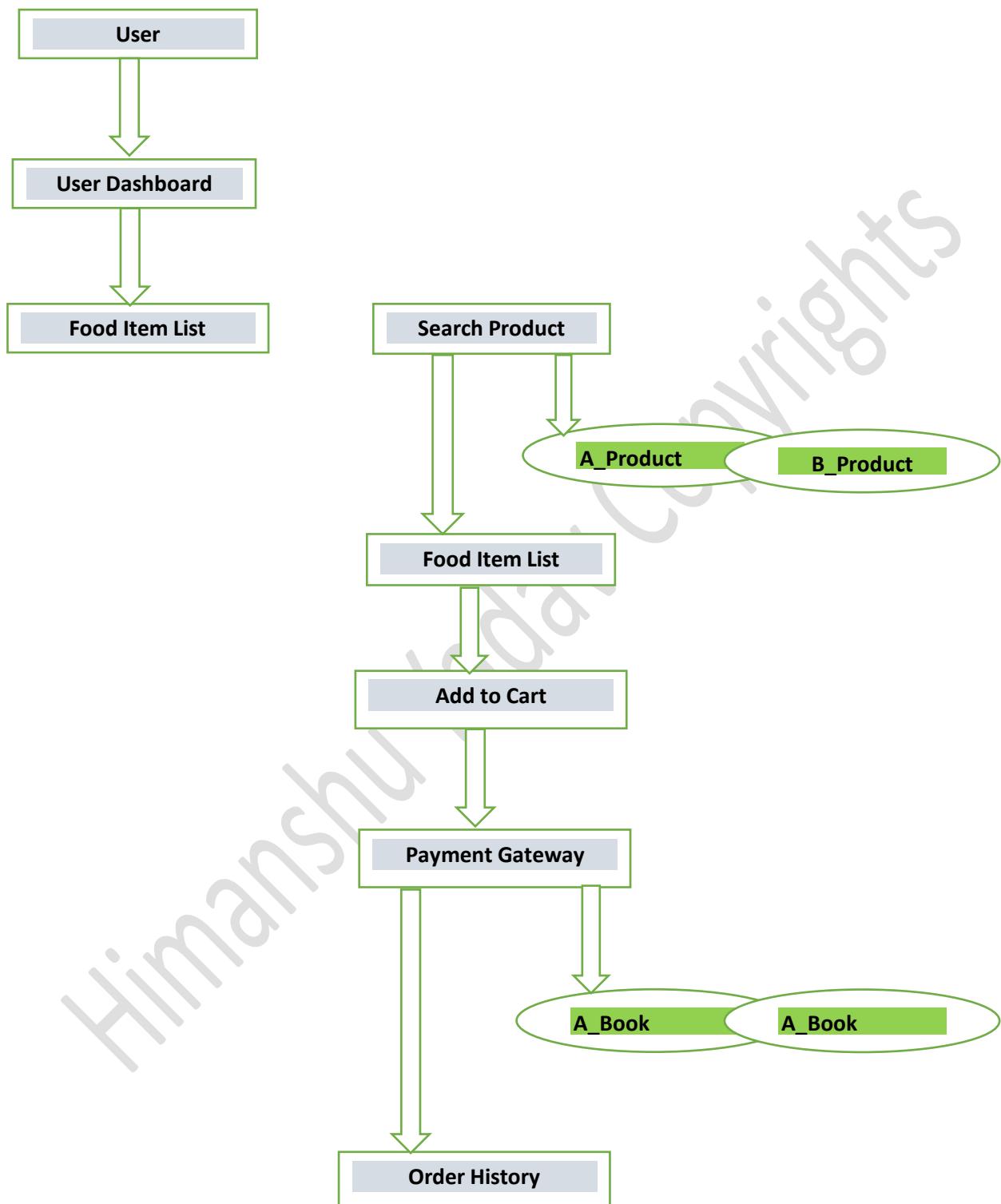
GitHub Link

Database Design

Folder Structure

Project Files

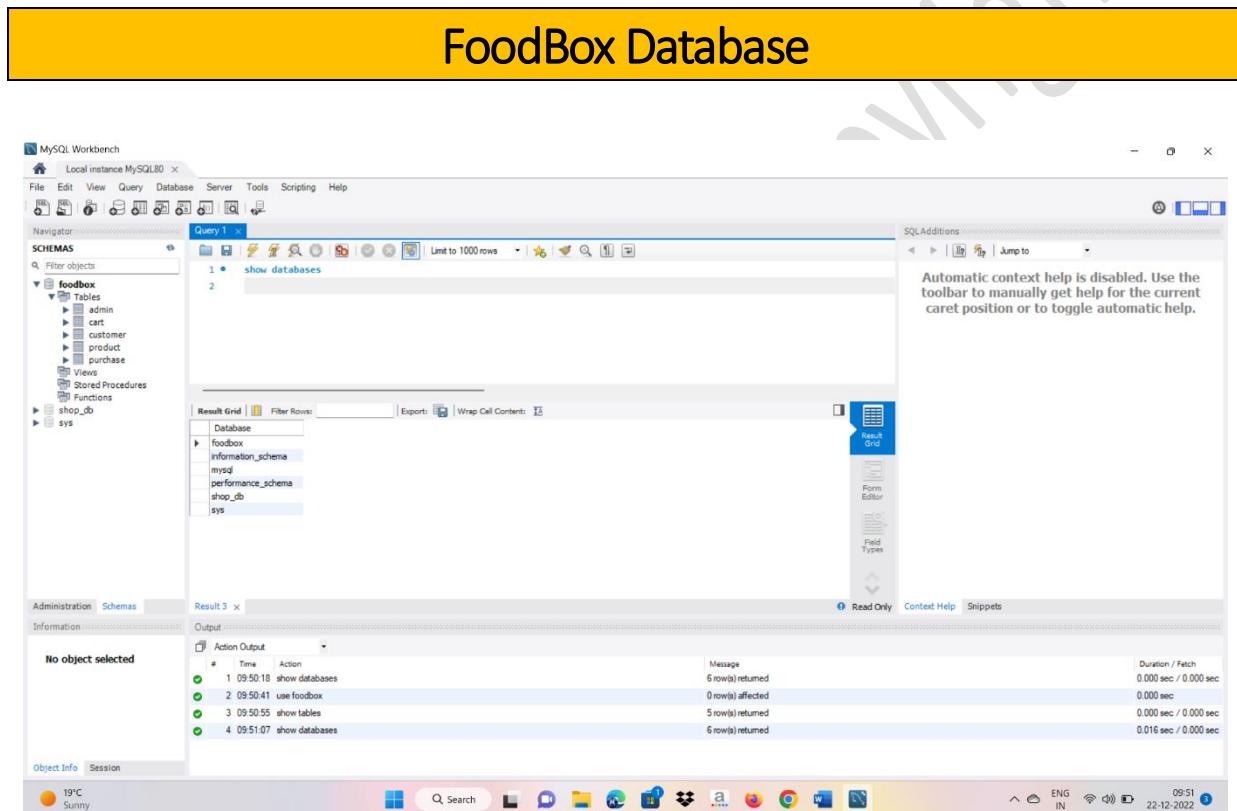
User Flow Diagram



GitHub Link

Repository Name	Capstone-Project-FoodBox
Link	https://github.com/ydvhimanshu/Capstone-Project-FoodBox

Database Screenshot



Tables in Foodbox Database

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **foodbox** schema with tables: admin, cart, customer, product, purchase.
- Query Grid:** Displays the results of the query `show tables`. The results are:

Tables_in_foodbox
admin
cart
customer
product
purchase
- Output:** Shows the execution history for the session, including the command `describe admin` at step 6.
- System Bar:** Shows the date and time as 22-12-2022 09:51:51.

Describe Admin Tables

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **foodbox** schema with tables: admin, cart, customer, product, purchase.
- Query Grid:** Displays the results of the query `describe admin`. The results are:

Field	Type	Null	Key	Default	Extra
username	varchar(255)	NO	PRI	NULL	
password	varchar(255)	YES	NULL	NULL	
- Output:** Shows the execution history for the session, including the command `describe admin` at step 6.
- System Bar:** Shows the date and time as 22-12-2022 09:52:09.

Describe Customer table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **SCHEMAS** section with the **foodbox** schema selected. Under **Tables**, the **customer** table is listed.
- Query Editor:** A query named **1 • describe customer** is running. The results are displayed in a **Result Grid** table:

Field	Type	Null	Key	Default	Extra
email	varchar(255)	NO	PRI	NULL	
address	varchar(255)	YES		NULL	
contact	varchar(255)	YES		NULL	
name	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	

- Result Grid:** Shows the execution history for the current session, including actions like using the schema, showing tables, and describing the customer table.
- System Bar:** Displays the date and time as 22-12-2022 09:52, and the weather as 19°C Sunny.

Describe Cart Table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **SCHEMAS** section with the **foodbox** schema selected. Under **Tables**, the **cart** table is listed.
- Query Editor:** A query named **1 • describe cart** is running. The results are displayed in a **Result Grid** table:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	
price	float	NO		NULL	
quantity	int	NO		NULL	
product_id	bigint	YES	MUL	NULL	

- Result Grid:** Shows the execution history for the current session, including actions like using the schema, showing tables, and describing the cart table.
- System Bar:** Displays the date and time as 22-12-2022 09:52, and the weather as 19°C Sunny.

Describe Product Table

The screenshot shows the MySQL Workbench interface with the title "Describe Product Table". The "Query 1" tab contains the SQL command "describe product". The "Result Grid" shows the following table structure:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI		
actual_price	float	NO			
avail	varchar(255)	YES			
category	varchar(255)	YES			
des	varchar(255)	YES			
discount	float	NO			
imagepath	varchar(255)	YES			
name	varchar(255)	YES			
price	float	NO			

The "Result 8" tab shows the session history with the following log entries:

#	Time	Action	Message	Duration / Fetch
4	09:51:07	show databases	6 row(s) returned	0.016 sec / 0.000 sec
5	09:51:34	show tables	5 row(s) returned	0.000 sec / 0.000 sec
6	09:52:09	describe admin	2 row(s) returned	0.000 sec / 0.000 sec
7	09:52:25	describe customer	5 row(s) returned	0.000 sec / 0.000 sec
8	09:52:46	describe cart	4 row(s) returned	0.000 sec / 0.000 sec
9	09:52:59	describe product	9 row(s) returned	0.000 sec / 0.000 sec

Describe Purchase Table

The screenshot shows the MySQL Workbench interface with the title "Describe Purchase Table". The "Query 1" tab contains the SQL command "describe purchase". The "Result Grid" shows the following table structure:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI		
dop	date	YES			
productname	varchar(255)	YES			
quantity	int	NO			
totalcost	float	NO			
transactionid	varchar(255)	YES			
customer_email	varchar(255)	YES	MUL		

The "Result 9" tab shows the session history with the following log entries:

#	Time	Action	Message	Duration / Fetch
5	09:51:34	show tables	5 row(s) returned	0.000 sec / 0.000 sec
6	09:52:09	describe admin	2 row(s) returned	0.000 sec / 0.000 sec
7	09:52:25	describe customer	5 row(s) returned	0.000 sec / 0.000 sec
8	09:52:46	describe cart	4 row(s) returned	0.000 sec / 0.000 sec
9	09:52:59	describe product	9 row(s) returned	0.000 sec / 0.000 sec
10	09:53:15	describe purchase	7 row(s) returned	0.000 sec / 0.000 sec

Admin Table Record

The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and "Local instance MySQL80". The "Query 1" tab contains the SQL query: "select * from admin". The results grid shows the following data:

username	password
admin	password
root	LinkedIn@12345
NULL	NULL

The "Information" pane shows the following log output:

#	Time	Action	Message	Duration / Fetch
6	09:52:09	describe admin	2 row(s) returned	0.000 sec / 0.000 sec
7	09:52:25	describe customer	5 row(s) returned	0.000 sec / 0.000 sec
8	09:52:46	describe cart	4 row(s) returned	0.000 sec / 0.000 sec
9	09:52:59	describe product	9 row(s) returned	0.000 sec / 0.000 sec
10	09:53:15	describe purchase	7 row(s) returned	0.000 sec / 0.000 sec
11	09:53:32	select * from admin LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

The system tray at the bottom right shows the date and time as "22-12-2022 09:53" and the weather as "19°C Sunny".

Customer Table Record

The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and "Local instance MySQL80". The "Query 1" tab contains the SQL query: "select * from customer". The results grid shows the following data:

email	address	contact	name	password
diksha12@gmail.com	nagpur	8989998889	Diksha Yadav	diksha
krish23@gmail.com	varanasi	8989009090	Ish Kumar	krish
prush23@gmail.com	banglore	783343478	Prush Yadav	1234
NULL	NULL	NULL	NULL	NULL

The "Information" pane shows the following log output:

#	Time	Action	Message	Duration / Fetch
7	09:52:25	describe customer	5 row(s) returned	0.000 sec / 0.000 sec
8	09:52:46	describe cart	4 row(s) returned	0.000 sec / 0.000 sec
9	09:52:59	describe product	9 row(s) returned	0.000 sec / 0.000 sec
10	09:53:15	describe purchase	7 row(s) returned	0.000 sec / 0.000 sec
11	09:53:32	select * from admin LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
12	09:53:46	select * from customer LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

The system tray at the bottom right shows the date and time as "22-12-2022 09:53" and the weather as "19°C Sunny".

Product Table Record

The screenshot shows the MySQL Workbench interface with the 'product' table selected. The table has columns: id, actual_price, avail, category, des, discount, imagepath, name. The data grid shows 16 rows of product information, including items like Chicken Biryani, Ravioli, and Butter Chicken. The bottom pane displays the 'Action Output' log with various SQL statements and their execution details.

id	actual_price	avail	category	des	discount	imagepath	name
12203	350	yes	Indian	Rice Steamed with Chicken and spices	10	./assets/images/biryani.jpg	Chicken Biryani
12984	200	yes	Italian	Ravioli pasta filled with veg mince	18	./assets/images/ravioli.jpg	Ravioli
14612	298	yes	Indian	Indian chicken kawab	0	./assets/images/chicken_kawab.jpg	Chicken Kawab
29051	350	yes	Indian	Chicken infused with butter and spices	0	./assets/images/butterChicken.png	Butter Chicken
34420	247	yes	Indian	Indian chicken 65 Delicious food	4	./assets/images/chicken_65.jpg	Chicken 65
53430	435	yes	Italian	Pasta with Tomato and oregano	0	./assets/images/redPasta.jpg	Red Sause Pasta
56547	180	yes	Mexican	Corn topped with cream cheese and spice	7	./assets/images/elote.jpg	Elote de Corn
57004	350	yes	Mexican	Wrapped Tortilla with Meat mince and Mayo	0	./assets/images/tortilla.jpg	Burrito
63201	435	yes	Italian	Pizza topped with cottage cheese and veges	0	./assets/images/paneerpizza.jpg	Paneer Pizza
65118	288	yes	Indian	Indian Chicken Curry Masala	3	./assets/images/chicken_curry.jpg	Chicken Curry
65772	65	yes	Indian	Indian famous south Indian dish	3	./assets/images/dosa.jpg	Dosa

Purchase Table Record

The screenshot shows the MySQL Workbench interface with the 'purchase' table selected. The table has columns: id, dop, productname, quantity, totalcost, transactionid, customer_email. The data grid shows 10 rows of purchase history, including items like Chicken Biryani and Butter Chicken. The bottom pane displays the 'Action Output' log with various SQL statements and their execution details.

id	dop	productname	quantity	totalcost	transactionid	customer_email
193695	2022-12-21	Chicken Biryani	1	328.5	TFIB730579293306	kiran@gmail.com
296854	2022-12-21	Butter Chicken	1	328.5	TFIB730579293303	kiran@gmail.com
536206	2022-12-21	Chicken 65	1	237.12	TFIB120254217917	disha12@gmail.com
576760	2022-12-21	Butter Chicken	1	350	TFIB730579293306	kiran21@gmail.com
827730	2022-12-21	Butter Chicken	1	350	TFIB120254217917	disha12@gmail.com
829269	2022-12-21	Chicken Biryani	1	328.5	TFIB20781226008	piyush23@gmail.com
911800	2022-12-21	Elote de Corn	1	167.4	TFIB20186136663	disha12@gmail.com
911801	2022-12-21	Elote de Corn	1	167.4	TFIB20186136663	disha12@gmail.com

BackEnd Development Screenshot

The screenshot shows the Eclipse IDE interface with the project 'Capstone - foodbox' open. The 'FoodboxSpringApplication.java' file is selected in the editor. The code defines a main method that runs the FoodboxSpringApplication class. Below the code, the Java console displays the application's startup logs, which include Hibernate and Tomcat initialization messages.

```
package com.foodbox;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
public class FoodboxSpringApplication {
    public static void main(String[] args) {
        SpringApplication.run(FoodboxSpringApplication.class, args);
    }
}
```

```
2022-12-22 10:12:49.074 INFO 14936 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo : HHH000412: Hibernate Core Version 5.2.2.Final
2022-12-22 10:12:49.074 INFO 14936 --- [ restartedMain] o.hibernate.annotations.common.Version : HCAAN000001: Hibernate Commons Annotations Version 5.1.0.Final
2022-12-22 10:12:49.390 INFO 14936 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HHH000041: Hibernate Core Version 5.2.2.Final
2022-12-22 10:12:49.390 INFO 14936 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HHH000040: JDBC Driver Version 3.1.1
2022-12-22 10:12:49.728 INFO 14936 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.mysql.MySQLDialect
2022-12-22 10:12:49.728 INFO 14936 --- [ restartedMain] org.hibernate.id.IdentityGenerator : HHH000300: Using identity generator: org.hibernate.id.SequenceGenerator
2022-12-22 10:12:50.463 INFO 14936 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HHH000049: Using platform implementation: com.zaxxer.hikari.HikariPool-1
2022-12-22 10:12:51.106 WARN 14936 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'FoodboxWebConfiguration'
2022-12-22 10:12:51.440 INFO 14936 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 357
2022-12-22 10:12:51.474 INFO 14936 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http)
2022-12-22 10:12:51.486 INFO 14936 --- [ restartedMain] com.foodbox.FoodboxspringApplication : Started FoodboxspringApplication in 4.501 seconds (JVM running for 5.001)
```

The screenshot shows the Eclipse IDE interface with the project 'Capstone - foodbox' open. The 'ConfigCtrlJava.java' file is selected in the editor. The code defines a ConfigFilter class that implements the Filter interface. It overrides the doFilter method to handle HTTP requests and responses, setting various headers and statuses based on the request method.

```
package com.foodbox.config;
import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class ConfigFilter implements Filter {
    @Override
    public void doFilter(HttpServletRequest req, HttpServletResponse res, FilterChain chain) throws IOException, ServletException {
        final HttpServletResponse response = (HttpServletResponse) res;
        response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET, OPTIONS, DELETE");
        response.setHeader("Access-Control-Allow-Headers", "Authorization, Content-Type");
        response.setHeader("Access-Control-Max-Age", "3600");
        if ("OPTIONS".equalsIgnoreCase(((HttpServletRequest) req).getMethod())) {
            response.setStatus(HttpServletResponse.SC_OK);
        } else {
            chain.doFilter(req, res);
        }
    }
    @Override
    public void destroy() {
    }
    @Override
    public void init(FilterConfig config) throws ServletException {
    }
}
```

Capstone - foodbox/src/main/java/com/foodbox/controller/AdminController.java - Eclipse IDE

```
1 package com.foodbox.controller;
2
3 import java.util.Map;
4
5 @CrossOrigin(origins = "http://localhost:4200",allowedHeaders = "*")
6 @RestController
7 public class AdminController {
8
9     @Autowired
10    private AdminRepository adminRepository;
11
12    @PostMapping("/admin/login")
13    public boolean verifyAdminLogin(@RequestBody Map loginData, @PathVariable(name = "username") String username, HttpSession session) {
14        String lusername=(String) loginData.get("username");
15        String lpassword=(String) loginData.get("password");
16        Admin admin = adminRepository.findByUsername(username);
17        if(admin==null || admin.getUsername().equals(username) && admin.getPassword().equals(lpassword)) {
18            session.setAttribute("adminUsername", lusername);
19            return true;
20        }else {
21            return false;
22        }
23    }
24}
25
26
27
28
29
30
31
32
33
34
35
36
37
38 }
```

Project Explorer X

- com.foodbox
- com.foodbox.config
- com.foodbox.controller
 - AdminController.java
 - CartController.java
 - CustomerController.java
 - ProductController.java
 - PurchaseController.java
- com.foodbox.exception
- com.foodbox.model
- com.foodbox.repository

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

HELP.md

mvnw

mvnw.cmd

pom.xml

Servers [Capstone master]

Tomcat v10.0 Server at localhost-config

Markers Properties Servers Data Source Explorer Snippets Terminal Console

FoodboxSpringApplication [Java Application] [pid:14936]

22-12-22 10:22:49,000 INFO [main] org.springframework.boot.SpringApplication - Root WebApplicationContext: initialization completed in 1 ms

19°C Sunny

Search Smart Insert 37:5:1328

ENG IN 10:18 22-12-2022

Capstone - foodbox/src/main/java/com/foodbox/controller/CartController.java - Eclipse IDE

```
1 package com.foodbox.controller;
2
3 import java.util.List;
4
5 @CrossOrigin(origins = "http://localhost:4200")
6 @RestController
7 public class CartController {
8
9     @Autowired
10    private CartRepository cartRepository;
11
12    @CrossOrigin(origins = "http://localhost:4200")
13    @PostMapping("/cart")
14    public Cart addToCart(@RequestBody Cart cart, HttpSession session) {
15        float grandTotal=0;
16        if(session.getAttribute("grandTotal")==null) {
17            grandTotal=0;
18        }else {
19            grandTotal=(float) session.getAttribute("grandTotal");
20        }
21        List cartList = cartRepository.findAll();
22        for(Cart temp:cartList) {
23            if(temp.getId()==cart.getProduct().getId()) {
24                int tempQuantity = temp.getQuantity();
25                grandTotal+=temp.getPrice();
26                session.setAttribute("grandTotal", grandTotal);
27                temp.setQuantity(tempQuantity+1);
28                temp.setPrice((temp.getProduct().getPrice()*tempQuantity));
29                return cartRepository.save(temp);
30            }
31        }
32        int min = 100;
33        int max = 999;
34        int b = (int) (Math.random() * (max - min + 1) + min);
35        cart.setId(b);
36        cart.setQuantity(1);
37        cart.setPrice(cart.getProduct().getPrice());
38    }
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55 }
```

Project Explorer X

- com.foodbox
- com.foodbox.config
- com.foodbox.controller
 - AdminController.java
 - CartController.java
 - CustomerController.java
 - ProductController.java
 - PurchaseController.java
- com.foodbox.exception
- com.foodbox.model
- com.foodbox.repository

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

HELP.md

mvnw

mvnw.cmd

pom.xml

Servers [Capstone master]

Tomcat v10.0 Server at localhost-config

Markers Properties Servers Data Source Explorer Snippets Terminal Console

FoodboxSpringApplication [Java Application] [pid:14936]

22-12-22 10:22:49,000 INFO [main] org.springframework.boot.SpringApplication - Root WebApplicationContext: initialization completed in 1 ms

19°C Sunny

Search Smart Insert 35:10:1232

ENG IN 10:18 22-12-2022

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer** (left sidebar): Shows the project structure for "Capstone - foodbox/src/main/java/com/foodbox/controller". The "CustomerController.java" file is selected.
- CustomerController.java** (right pane): The code for the CustomerController class. It includes annotations like @CrossOrigin, @RestController, and @Autowired. It has methods for adding a customer, verifying a login, and getting all customers.
- Bottom Bar**: Contains tabs for Markers, Properties, Servers, Data Source Explorer, Snippets, Terminal, and Console. The "Servers" tab is active.
- System Status Bar**: Shows the date (22-12-2022), time (10:18), battery level (19°C), and network status.

```
CustomerController.java
1 package com.foodbox.controller;
2
3 import java.util.HashMap;
4
5 @CrossOrigin(origins = "http://localhost:4200", allowedHeaders = "*")
6 @RestController
7 public class CustomerController {
8
9     @Autowired
10    private CustomerRepository customerRepository;
11
12    @CrossOrigin(origins = "http://localhost:4200")
13    @PostMapping("/customers")
14    public Customer addCustomer(@RequestBody Customer customer, HttpSession session) {
15        session.setAttribute("cust_email", customer.getEmail());
16        return customerRepository.save(customer);
17    }
18
19    @SuppressWarnings("rawtypes")
20    @CrossOrigin(origins = "http://localhost:4200")
21    @PostMapping("/customers/{email}")
22    public boolean verifyLogin(@RequestBody Map<String, String> loginData, @PathVariable(name = "email") String email, HttpSession session) {
23        String email = (String) loginData.get("email");
24        String ipAddress = (String) loginData.get("password");
25        Customer customer = customerRepository.findByEmail(email);
26        if (customer != null && customer.getEmail().equals(email) && customer.getPassword().equals(ipAddress)) {
27            session.setAttribute("cust_email", email);
28            return true;
29        } else {
30            return false;
31        }
32    }
33
34    @GetMapping("/customers")
35    public List<Customer> getAllCustomers(){
36        return customerRepository.findAll();
37    }
38}
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer**: Shows the project structure for "Capstone - foodbox/src/main/java/com/foodbox/controller/ProductController.java". The tree includes: foodbox (Capstone master), src/main/java, com.foodbox, com.foodbox.config, com.foodbox.controller (AdminController.java, CartController.java, CustomerController.java, ProductController.java, PurchaseController.java), com.foodbox.exception, com.foodbox.repository, src/main/resources, and pom.xml.
- Editor**: Displays the code for `ProductController.java`. The code defines a `ProductController` class with methods for getting admin products and customer products, and saving new admin users.
- Toolbars**: Standard Eclipse toolbars for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Bottom Status Bar**: Shows the status bar with "Writable", "Smart Insert", "35:46:1233", and system icons for battery, network, and time.

Capstone - foodbox/src/main/java/com/foodbox/controller/PurchaseController.java - Eclipse IDE

```
1 package com.foodbox.controller;
2
3 import java.util.HashMap;
4
5 @CrossOrigin(origins = "http://localhost:4200", allowedHeaders = "*")
6 @RestController
7 public class PurchaseController {
8
9     @Autowired
10    private PurchaseRepository purchaseRepository;
11
12    @Autowired
13    private CartRepository cartRepository;
14
15    @Autowired
16    private CustomerRepository customerRepository;
17
18    @GetMapping("/purchase/byEmail/{email}")
19    public List<Purchase> customerOrders(@PathVariable String email) {
20        return purchaseRepository.getByEmail(email);
21    }
22
23    @GetMapping("/purchase")
24    public List<Purchase> getAllPurchase() {
25        return purchaseRepository.findAllOrderByTransactionidAsc();
26    }
27
28    @DeleteMapping("/{id}")
29    public ResponseEntity<String, Boolean> deletePurchase(@PathVariable Long id) {
30        Purchase purchase = purchaseRepository.findById(id);
31        if (purchase == null) {
32            throw new ResourceNotFoundException("Purchase Id not found with " + id);
33        }
34        purchaseRepository.delete(purchase);
35        Map<String, Boolean> map = new HashMap<>();
36        map.put("deleted", Boolean.TRUE);
37        return ResponseEntity.ok(map);
38    }
39
40    @GetMapping("/purchase/search/{keyword}")
41}
```

Markers Properties Servers Data Source Explorer Snippets Terminal Console

FoodboxSpringApplication Java Application [pid:14936]

19°C Sunny

Capstone - foodbox/src/main/java/com/foodbox/exception/ResourceNotFoundException.java - Eclipse IDE

```
1 package com.foodbox.exception;
2
3 import org.springframework.http.HttpStatus;
4
5 @ResponseStatus(value=HttpStatus.NOT_FOUND)
6 public class ResourceNotFoundException extends RuntimeException{
7     private static final long serialVersionUID = 1L;
8
9     public ResourceNotFoundException(String message) {
10        super(message);
11    }
12}
```

Markers Properties Servers Data Source Explorer Snippets Terminal Console

FoodboxSpringApplication Java Application [pid:14936]

19°C Sunny

Capstone - foodbox/src/main/java/com/foodbox/model/Admin.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.exception
com.foodbox.model
Admin.java
Cart.java
Customer.java
Product.java
Purchase.java
com.foodbox.repository
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.mnd
mvnw
mvnw.cmd
pom.xml
Servers [Capstone master]
Tomcat v10.0 Server at localhost-config
```

```
1 package com.foodbox.model;
2
3import javax.persistence.Entity;
4
5 @Entity
6 public class Admin {
7     @Id
8     private String username;
9     private String password;
10
11     public Admin() {
12         super();
13     }
14
15     public Admin(String username, String password) {
16         super();
17         this.username = username;
18         this.password = password;
19     }
20
21     public String getUsername() {
22         return username;
23     }
24
25     public void setUsername(String username) {
26         this.username = username;
27     }
28     public String getPassword() {
29         return password;
30     }
31     public void setPassword(String password) {
32         this.password = password;
33     }
34
35
36 }
```

Markers Properties Data Source Explorer Snippets Terminal Console X

FootboxSpringApplication [Java Application] [pid:14936]

22-12-22 10:22:40:000 INFO [main] Root WebApplicationContext: initialization completed in 1 ms

19°C Sunny

Search

23:25:406

Writable Smart Insert

ENG IN 10:18 22-12-2022

Capstone - foodbox/src/main/java/com/foodbox/model/Cart.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.exception
com.foodbox.model
Admin.java
Cart.java
Customer.java
Product.java
Purchase.java
com.foodbox.repository
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.mnd
mvnw
mvnw.cmd
pom.xml
Servers [Capstone master]
Tomcat v10.0 Server at localhost-config
```

```
1 package com.foodbox.model;
2
3import javax.persistence.Entity;
4
5 @Entity
6 public class Cart {
7     @Id
8     private long id;
9     private int quantity;
10    private float price;
11    @OneToOne
12    private Product product;
13
14    public Cart(long id, int quantity, float price, Product product) {
15        super();
16        this.id = id;
17        this.quantity = quantity;
18        this.price = price;
19        this.product = product;
20    }
21
22    public Cart() {
23        super();
24    }
25
26    public long getId() {
27        return id;
28    }
29
30    public void setId(long id) {
31        this.id = id;
32    }
33
34    public int getQuantity() {
35        return quantity;
36    }
37
38    public void setQuantity(int quantity) {
39        this.quantity = quantity;
40    }
41
42    public float getPrice() {
43        return price;
44    }
45 }
```

Markers Properties Data Source Explorer Snippets Terminal Console X

FootboxSpringApplication [Java Application] [pid:14936]

22-12-22 10:22:40:000 INFO [main] Root WebApplicationContext: initialization completed in 1 ms

19°C Sunny

Search

25:5:464

Writable Smart Insert

ENG IN 10:19 22-12-2022

Capstone - foodbox/src/main/java/com/foodbox/model/Customer.java - Eclipse IDE

```
1 package com.foodbox.model;
2
3 import javax.persistence.Entity;
4
5 @Entity
6 public class Customer {
7     @Id
8     private String email;
9     private String password;
10    private String name;
11    private String contact;
12    private String address;
13
14    public Customer(String email, String password, String name, String contact, String address) {
15        super();
16        this.email = email;
17        this.password = password;
18        this.name = name;
19        this.contact = contact;
20        this.address = address;
21    }
22
23    public Customer() {
24        super();
25    }
26
27    public String getEmail() {
28        return email;
29    }
30
31    public void setEmail(String email) {
32        this.email = email;
33    }
34
35    public String getPassword() {
36        return password;
37    }
38
39    public void setPassword(String password) {
40        this.password = password;
41    }
42
43    public String getName() {
44    }
45}
```

Capstone - foodbox/src/main/java/com/foodbox/model/Product.java - Eclipse IDE

```
1 package com.foodbox.model;
2
3 import javax.persistence.Entity;
4
5 @Entity
6 public class Product {
7     @Id
8     private long id;
9     private String name;
10    private String des;
11    private String category;
12    private float actualPrice;
13    private float discount;
14    private float price;
15    private String avail;
16    private String imagepath;
17
18    public Product(long id, String name, String des, String category, float actualPrice, float discount, float price,
19                  String avail, String imagepath) {
20        super();
21        this.id = id;
22        this.name = name;
23        this.des = des;
24        this.category = category;
25        this.actualPrice = actualPrice;
26        this.discount = discount;
27        this.price = price;
28        this.avail = avail;
29        this.imagepath = imagepath;
30    }
31
32    public Product() {
33        super();
34    }
35
36    public long getId() {
37        return id;
38    }
39
40    public void setId(long id) {
41        this.id = id;
42    }
43}
```

Capstone - foodbox/src/main/java/com/foodbox/model/Purchase.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X CartJava CustomerJava ProductJava PurchaseJava X
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.exception
com.foodbox.model
AdminJava
CartJava
CustomerJava
ProductJava
PurchaseJava
com.foodbox.repository
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Servers [Capstone master]
Tomcat v10.0 Server at localhost-config

1 package com.foodbox.model;
2
3 import java.sql.Date;
4
5 @Entity
6 public class Purchase {
7     @Id
8     private long id;
9     private float totalcost;
10    private Date dop;
11    private int quantity;
12    private String productname;
13    private String transactionid;
14    @OneToOne
15    private Customer customer;
16
17    public Purchase() {
18        super();
19    }
20
21    public Purchase(long id, float totalcost, Date dop, int quantity, String productname, String transactionid, Customer customer) {
22        super();
23        this.id = id;
24        this.totalcost = totalcost;
25        this.dop = dop;
26        this.quantity = quantity;
27        this.productname = productname;
28        this.transactionid = transactionid;
29        this.customer = customer;
30    }
31
32    public long getId() {
33        return id;
34    }
35
36    public void setId(long id) {
37        this.id = id;
38    }
39
40    public float getTotalcost() {
41        return totalcost;
42    }
43
44    public Date getDop() {
45        return dop;
46    }
47 }
```

Capstone - foodbox/src/main/java/com/foodbox/repository/AdminRepository.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X AdminRepository.java X
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.exception
com.foodbox.model
com.foodbox.repository
AdminRepository.java
CartRepository.java
CustomerRepository.java
ProductRepository.java
PurchaseRepository.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Servers [Capstone master]
Tomcat v10.0 Server at localhost-config

1 package com.foodbox.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 public interface AdminRepository extends JpaRepository<Admin, String> {
6     Admin findByUsername(String username);
7 }
```

Capstone - foodbox/src/main/java/com/foodbox/repository/CartRepository.java - Eclipse IDE

```
File Edit Source Refactor Navigate Project Run Window Help
Project Explorer X CartRepository.java X
foodbox (Capstone master)
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.exception
com.foodbox.model
com.foodbox.repository
AdminRepository.java
CartRepository.java
CustomerRepository.java
ProductRepository.java
PurchaseRepository.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Servers (Capstone master)
Tomcat v10.0 Server at localhost-config
```

Markers Properties Servers Data Source Explorer Snippets Terminal Console X

FoodboxSpringApplication [Java Application] [pid:14936] 2022-12-22 10:11:40,000 INFO [main] o.s.boot.SpringApplication: application context initialized

Writable Smart Insert 9:2:206

19°C Sunny ENG IN 10:19 22-12-2022

Capstone - foodbox/src/main/java/com/foodbox/repository/CustomerRepository.java - Eclipse IDE

```
File Edit Source Refactor Navigate Project Run Window Help
Project Explorer X CustomerRepository.java X
foodbox (Capstone master)
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.exception
com.foodbox.model
com.foodbox.repository
AdminRepository.java
CartRepository.java
CustomerRepository.java
ProductRepository.java
PurchaseRepository.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Servers (Capstone master)
Tomcat v10.0 Server at localhost-config
```

Markers Properties Servers Data Source Explorer Snippets Terminal Console X

FoodboxSpringApplication [Java Application] [pid:14936] 2022-12-22 10:11:40,000 INFO [main] o.s.boot.SpringApplication: application context initialized

Writable Smart Insert 18:2:547

19°C Sunny ENG IN 10:19 22-12-2022

Capstone - foodbox/src/main/java/com/foodbox/repository/ProductRepository.java - Eclipse IDE

```
File Edit Source Refactor Navigate Project Run Window Help
Project Explorer X CartRepository.java CustomerRepository.java ProductRepository.java X
src/main/java
  com.foodbox
    com.foodbox.config
    com.foodbox.controller
    com.foodbox.exception
    com.foodbox.model
    com.foodbox.repository
      AdminRepository.java
      CartRepository.java
      CustomerRepository.java
      ProductRepository.java
      PurchaseRepository.java
    src/main/resources
    src/test/java
    JRE System Library [JavaSE-1.8]
    Maven Dependencies
    src
    target
    HELP.mnd
    mvnw
    mvnw.cmd
    pom.xml
Servers [Capstone master]
  Tomcat v10.0 Server at localhost-config
```

CartRepository.java

```
package com.foodbox.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface CartRepository extends JpaRepository<Cart, Long> {
    @Query("Select p FROM Cart p WHERE p.avail='yes' ORDER BY 'category'")
    List<Cart> findIfAvail();
    @Query("SELECT p FROM Cart p WHERE (p.avail LIKE 'yes') AND (p.name LIKE ?1%"
            + " OR p.des LIKE ?1%" +
            " OR p.price LIKE ?1%" +
            " OR p.category LIKE ?1%)")
    public List<Cart> homeSearch(String keyword);
    @Query("SELECT p FROM Cart p WHERE p.category LIKE 'Chinese' AND p.avail LIKE 'yes'")
    public List<Cart> getChinese();
    @Query("SELECT p FROM Cart p WHERE p.category LIKE 'Indian' AND p.avail LIKE 'yes'")
    public List<Cart> getIndian();
    @Query("SELECT p FROM Cart p WHERE p.category LIKE 'Mexican' AND p.avail LIKE 'yes'")
    public List<Cart> getMexican();
    @Query("SELECT p FROM Cart p WHERE p.category LIKE 'Italian' AND p.avail LIKE 'yes'")
    public List<Cart> getItalian();
}
```

ProductRepository.java

```
package com.foodbox.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface ProductRepository extends JpaRepository<Product, Long> {
    @Query("Select p FROM Product p WHERE p.avail='yes' ORDER BY 'category'")
    List<Product> findIfAvail();
    @Query("SELECT p FROM Product p WHERE (p.avail LIKE 'yes') AND (p.name LIKE ?1%"
            + " OR p.des LIKE ?1%" +
            " OR p.price LIKE ?1%" +
            " OR p.category LIKE ?1%)")
    public List<Product> homeSearch(String keyword);
    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Chinese' AND p.avail LIKE 'yes'")
    public List<Product> getChinese();
    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Indian' AND p.avail LIKE 'yes'")
    public List<Product> getIndian();
    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Mexican' AND p.avail LIKE 'yes'")
    public List<Product> getMexican();
    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Italian' AND p.avail LIKE 'yes'")
    public List<Product> getItalian();
}
```

Capstone - foodbox/src/main/java/com/foodbox/repository/PurchaseRepository.java - Eclipse IDE

```
File Edit Source Refactor Navigate Project Run Window Help
Project Explorer X CartRepository.java CustomerRepository.java ProductRepository.java PurchaseRepository.java X
src/main/java
  com.foodbox
    com.foodbox.config
    com.foodbox.controller
    com.foodbox.exception
    com.foodbox.model
    com.foodbox.repository
      AdminRepository.java
      CartRepository.java
      CustomerRepository.java
      ProductRepository.java
      PurchaseRepository.java
    src/main/resources
    src/test/java
    JRE System Library [JavaSE-1.8]
    Maven Dependencies
    src
    target
    HELP.mnd
    mvnw
    mvnw.cmd
    pom.xml
Servers [Capstone master]
  Tomcat v10.0 Server at localhost-config
```

PurchaseRepository.java

```
package com.foodbox.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface PurchaseRepository extends JpaRepository<Purchase, Long> {
    @Query("Select p FROM Purchase p WHERE p.customer_email LIKE ?1%")
    public List<Purchase> getByEmail(String email);
    @Query("Select p FROM Purchase p WHERE p.transactionid LIKE ?1%"
            + " OR p.dop LIKE ?1%" +
            " OR p.productname LIKE ?1%" +
            " OR p.customer LIKE ?1%")
    public List<Purchase> searchPurchase(String keyword);
}
```

Capstone - foodbox/src/main/resources/application.properties - Eclipse IDE

```

server.port=8086
spring.datasource.url= jdbc:mysql://localhost:3306/foodbox
spring.datasource.username=root
spring.datasource.password=LinkedIn12345
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.hibernate.ddl-auto = update
static
templates
application.properties
src/main/java
com.foodbox
com.foodbox.config
com.foodbox.controller
com.foodbox.model
com.foodbox.repository
src/main/resources
src
templates
application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.nd
mvnw
mvnw.cmd
pom.xml
Servers [Capstone master]
Tomcat v10.0 Server at localhost-config

```

Markers Properties Servers Data Source Explorer Snippets Terminal Console

FoodboxSpringApplication [Java Application] [pid:14936]

2022-12-22 10:11:40,000 INFO [main] - Resource loading: C:\Users\Acer\IdeaProjects\Capstone\src\main\resources\application.properties

14|

Writable Insert 14:1:418

ENG IN 10:20 22-12-2022

19°C Sunny

Capstone - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer Maven Dependencies

- spring-boot-starter-data-jpa-2.7.6
- spring-boot-starter-aop-2.7.6.jar
- spring-aop-5.3.24.jar
- aspectjweaver-1.9.7.jar
- spring-boot-starter-jdbc-2.7.6.jar
- HikarICP-4.0.3.jar
- spring-jdbc-5.3.24.jar
- jakarta.transaction-api-1.3.3.jar
- jakarta.persistence-api-2.2.3.jar
- hibernate-core-5.6.14.Final.jar
- jboss-logging-3.4.3.Final.jar
- byte-buddy-1.12.19.jar
- antlr-2.7.7.jar
- jandex-2.4.2.Final.jar
- cassert-1.5.1.jar
- hibernate-commons-annotations-5.1.4.Final.jar
- lombok-runtime-2.37.jar
- b2w-2.3.7.jar
- stack-commons-runtime-3.0.12.jar
- jakarta.activation-1.2.2.jar
- spring-data-jpa-2.7.6.jar
- spring-data-commons-2.7.6.jar
- spring-orm-5.3.24.jar
- spring-context-5.3.24.jar
- spring-tx-5.3.24.jar
- spring-beans-5.3.24.jar
- slf4j-api-1.7.36.jar
- spring-aspects-5.3.24.jar
- spring-boot-starter-web-2.7.6.jar
- spring-boot-starter-2.7.6.jar
- spring-boot-starter-logging-2.7.6.jar
- logback-classic-1.2.11.jar
- logback-core-1.2.11.jar
- log4j-to-slf4j-2.17.2.jar
- log4j-api-2.17.2.jar
- jul-to-slf4j-1.7.36.jar

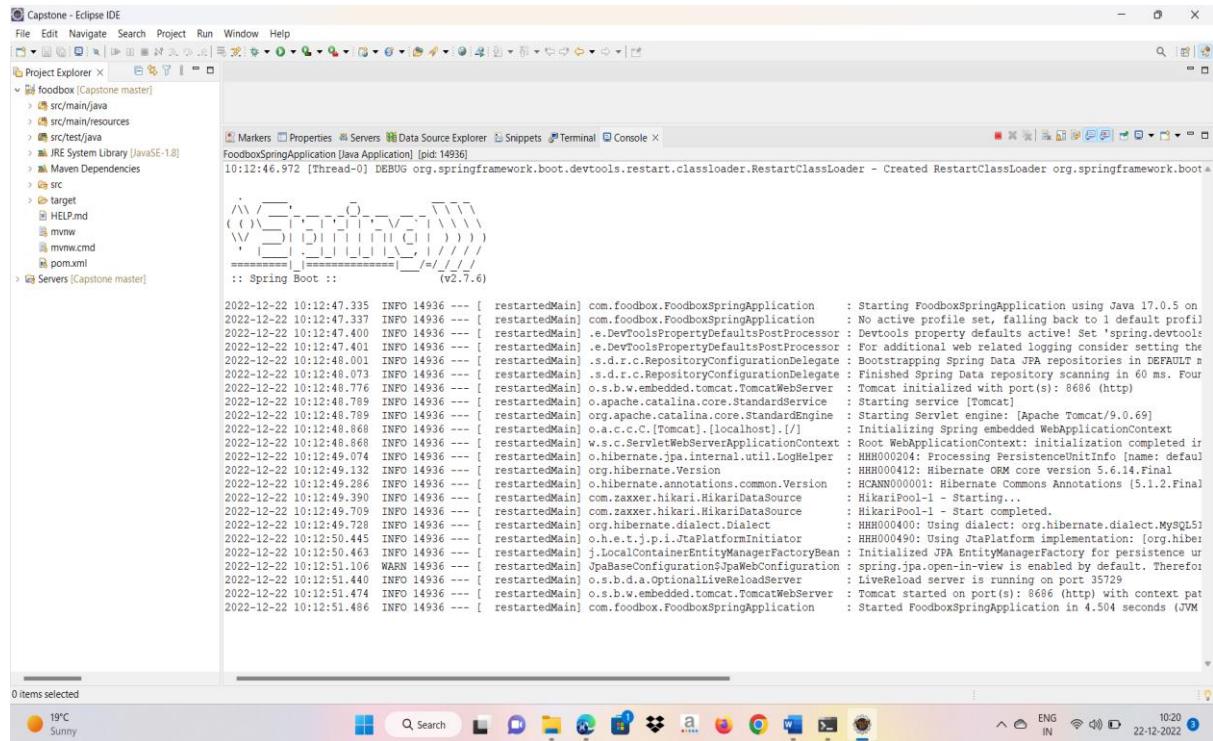
Markers Properties Servers Data Source Explorer Snippets Terminal Console

FoodboxSpringApplication [Java Application] [pid:14936]

2022-12-22 10:11:40,000 INFO [main] - Resource loading: C:\Users\Acer\IdeaProjects\Capstone\src\main\resources\application.properties

0 items selected

19°C Sunny



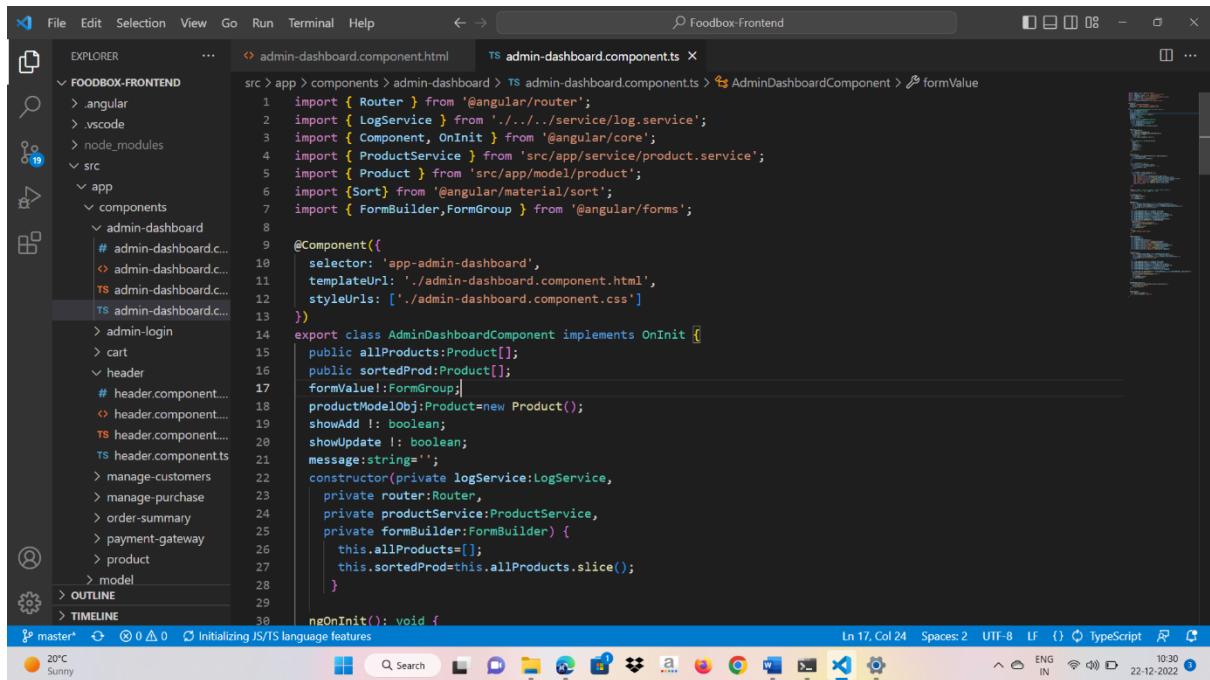
FrontEnd Source Code

This screenshot shows the VS Code interface with the file `admin-dashboard.component.css` open in the editor. The code defines styles for various components of an admin dashboard, including a logo, header title, and navigation items. A tooltip from the 'Angular Language Service' extension is visible, prompting the user to install it for better support.

```
# admin-dashboard.component.css
src > app > components > admin-dashboard > # admin-dashboard.component.css > .header-title
1 .logo{
2     position: relative;
3     top: 8px;
4     height: 40px;
5     width: 52px;
6     cursor: pointer;
7 }
8 .header-title{
9     position: relative;
10    top: 8px;
11    left: 25px;
12    font-family: 'WindSong', cursive;
13    font-size: 32px;
14    color: #bf1e2e;
15 }
16 .bthHead{
17    margin-left: 10px;
18 }
19 .active, .bthHead:hover{
20    color: #bf1e2e;
21 }
22 }
23 hr{
24    margin: 0;
25 }
26 .prodLogo{
27    position: relative;
28    top: 20px;
29    font-size: 36px;
30 }
```

This screenshot shows the VS Code interface with the file `admin-dashboard.component.html` open in the editor. The code defines the structure of the admin dashboard's header, including a navigation bar with links for products, customers, purchases, and logout, and a container for the main content.

```
<div>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div>
      
      <strong class="header-title">The Foodbox</strong>
    </div>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <button class="btn bthHead active"><i class="fas fa-tasks fa-lg"></i> Manage Products</button>
        </li>
        <li class="nav-item">
          <button class="btn bthHead" routerLink="/manageCustomer"><i class="fas fa-users fa-lg"></i> Manage Customers</button>
        </li>
        <li class="nav-item">
          <button class="btn bthHead" routerLink="/managePurchase"><i class="fas fa-shopping-basket fa-lg"></i> Manage Purchase</button>
        </li>
        <li class="nav-item">
          <button class="btn bthHead float-right" (click)="adminLogout()">Logout</button>
        </li>
      </ul>
    </div>
  </nav>
  <hr>
</div>
<div>
  <div class="container">
    <span class="text-center prodLogo">Product List</span>
    <button type="button" (click)="clickAddProduct()" data-toggle="modal" data-target="#exampleModal">Add Product</button>
  </div>
</div>
```



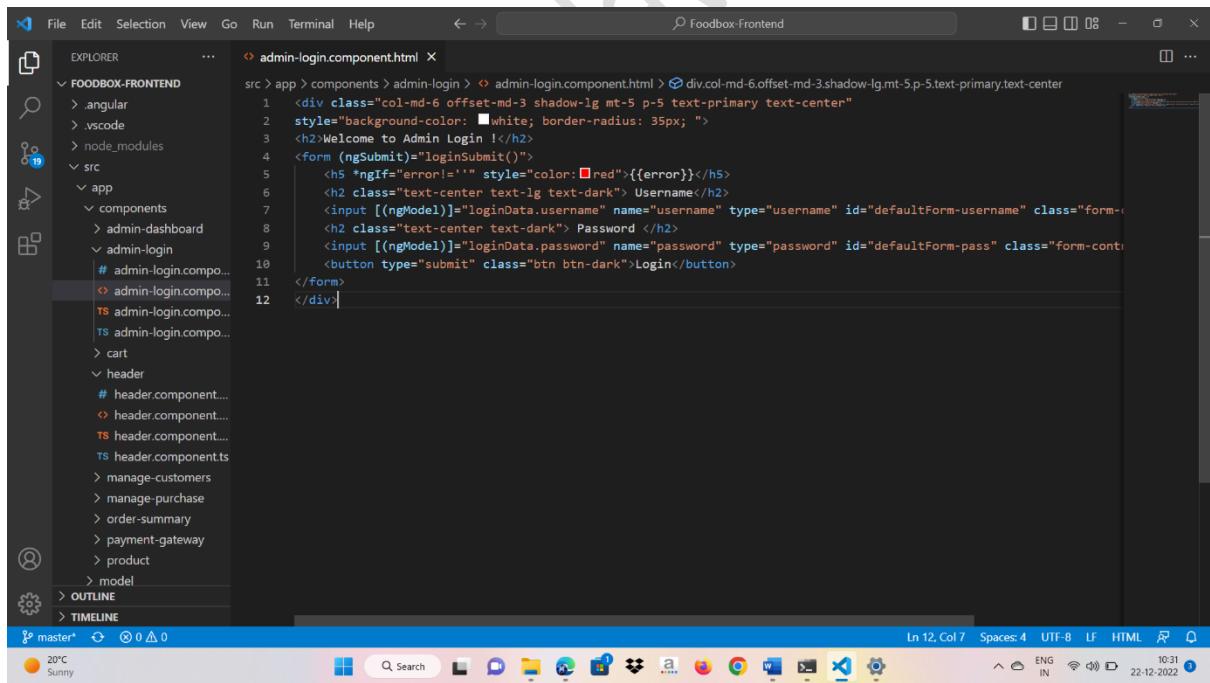
The screenshot shows a code editor window for a file named `admin-dashboard.component.ts`. The code is written in TypeScript and defines an `AdminDashboardComponent` that implements the `OnInit` interface. The component has a selector of `'app-admin-dashboard'`, a template URL of `'./admin-dashboard.component.html'`, and style URLs pointing to `'./admin-dashboard.component.css'`. It imports various services and components from the application's structure. The code includes logic for managing products, adding new products, updating existing ones, and handling errors. The editor interface includes an Explorer sidebar, a search bar, and a status bar at the bottom.

```
src > app > components > admin-dashboard > TS admin-dashboard.component.ts > AdminDashboardComponent > formValue
1 import { Router } from '@angular/router';
2 import { LogService } from '../../../../../service/log.service';
3 import { Component, OnInit } from '@angular/core';
4 import { ProductService } from 'src/app/service/product.service';
5 import { Product } from 'src/app/model/product';
6 import { Sort } from '@angular/material/sort';
7 import { FormBuilder, FormGroup } from '@angular/forms';

@Component({
  selector: 'app-admin-dashboard',
  templateUrl: './admin-dashboard.component.html',
  styleUrls: ['./admin-dashboard.component.css']
})
export class AdminDashboardComponent implements OnInit {
  public allProducts:Product[];
  public sortedProd:Product[];
  formValue!:FormGroup;
  productModelObj:Product=new Product();
  showAdd!: boolean;
  showUpdate!: boolean;
  message:string='';
  constructor(private logService:LogService,
  private router:Router,
  private productService:ProductService,
  private formBuilder:FormBuilder) {
    this.allProducts=[];
    this.sortedProd=this.allProducts.slice();
  }
}

ngOnInit(): void {
}

Ln 17, Col 24 Spaces: 2 UTF-8 LF {} TypeScript
20°C Sunny 10:30 22-12-2022
```



The screenshot shows a code editor window for a file named `admin-login.component.html`. The code is an HTML template for a login page. It features a central `div` with a class of `col-md-6 offset-md-3 shadow-lg mt-5 p-5 text-primary text-center` and a background color of white. Inside this div, there is a `<h2>Welcome to Admin Login !</h2>`, a form with a submit button labeled `Login`, and two input fields for `username` and `password`. The form also includes error handling logic using the `*ngIf` directive. The editor interface includes an Explorer sidebar, a search bar, and a status bar at the bottom.

```
src > app > components > admin-login > admin-login.component.html > div.col-md-6.offset-md-3.shadow-lg.mt-5.p-5.text-primary.text-center
1 <div class="col-md-6 offset-md-3 shadow-lg mt-5 p-5 text-primary text-center"
2   style="background-color: white; border-radius: 35px; "
3   <h2>Welcome to Admin Login !</h2>
4   <form (ngSubmit)="loginSubmit()">
5     <ngIf="error" style="color: red">{{error}}</ngIf>
6     <h2 class="text-center text-lg text-dark"> Username</h2>
7     <input [(ngModel)]="loginData.username" name="username" type="username" id="defaultForm-username" class="form-control" />
8     <h2 class="text-center text-dark"> Password </h2>
9     <input [(ngModel)]="loginData.password" name="password" type="password" id="defaultForm-pass" class="form-control" />
10    <button type="submit" class="btn btn-dark">Login</button>
11  </form>
12 </div>

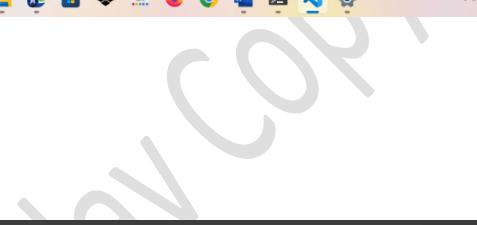
Ln 12, Col 7 Spaces: 4 UTF-8 LF HTML
20°C Sunny 10:31 22-12-2022
```

The screenshot shows the VS Code interface with the title bar "Foodbox-Frontend". The Explorer sidebar on the left shows the project structure under "FOODBOX-FRONTEND". The current file being edited is "admin-login.component.ts" in the "src/app/components/admin-login" folder. The code implements an "AdminLoginComponent" that extends "OnInit". It imports "AdminService", "LogService", "FormBuilder", "FormGroup", and "Router". The component has properties for "formValue", "adminData", "error", "loginData", "username", "password", and "auth". It includes methods for "ngOnInit" and "loginSubmit". The status bar at the bottom shows "Ln 16, Col 14" and "TypeScript".

```
src > app > components > admin-login > TS admin-login.component.ts < AdminLoginComponent > loginData
1 import { AdminService } from '../../../../../service/admin.service';
2 import { LogService } from '../../../../../service/log.service';
3 import { Component, OnInit } from '@angular/core';
4 import { FormBuilder, FormGroup } from '@angular/forms';
5 import { Router } from '@angular/router';
6
7 @Component({
8   selector: 'app-admin-login',
9   templateUrl: './admin-login.component.html',
10  styleUrls: ['./admin-login.component.css']
11})
12 export class AdminLoginComponent implements OnInit {
13   formValue!: FormGroup;
14   adminData: any;
15   error: String '';
16   loginData: [];
17   username: '';
18   password: '';
19 }
20 auth: any;
21
22 constructor(private logService: LogService, private adminService: AdminService, private router: Router) { }
23
24 ngOnInit(): void {
25   this.logService.sendHeader(0);
26 }
27 loginSubmit(){
28   if(this.loginData.username.trim()==''||this.loginData.username==null){
29     this.error="Enter Username";
30     return;
31   }
32 }
```

The screenshot shows the VS Code interface with the title bar "Foodbox-Frontend". The Explorer sidebar on the left shows the project structure under "FOODBOX-FRONTEND". The current file being edited is "cart.component.css" in the "src/app/components/cart" folder. The code defines styles for ".card", ".center", ".center img", ".card-table", and ".card-table tr". The status bar at the bottom shows "Ln 23, Col 19" and "CSS".

```
# cartcomponent.css < TS cart.component.ts
src > app > components > cart > # cartcomponent.css > .center img
1 .card{
2   height:60vh;
3   margin:25px;
4   padding:25px;
5 }
6
7 .center{
8   position: absolute;
9   top:50%;
10  left:50%;
11  transform: translate(-50%,-50%);
12  text-align:center;
13 }
14 h4,h6{
15   font-weight: 400;
16 }
17 .center img{
18   text-decoration: none;
19   color: inherit;
20   border:none;
21   outline: none;
22   height: 162px;
23   width: 250px;
24   margin : 20px 0px;
25 }
26
27 .card-table{
28   position: relative;
29   display: flex;
30 }
```



Foodbox-Frontend

File Edit Selection View Go Run Terminal Help

EXPLORER

FOODBOX-FRONTEND

- > .angular
- > .vscode
- > node_modules
- < src
 - < app
 - < components
 - > admin-dashboard
 - > admin-login
 - < cart
 - # cart.component.css
 - cart.component.html
 - TS cart.component.ts
 - < header
 - # header.component...
 - header.component...
 - TS header.component...
 - TS header.components.ts
 - > manage-customers
 - > manage-purchase
 - > order-summary
 - > payment-gateway
 - > product
 - > model

> model

OUTLINE

TIMELINE

cart.component.ts TS cart.component.ts

```
src > app > components > cart > TS cart.component.ts > CartComponent > cart
1 import { LogService } from '../../../../../service/log.service';
2 import { Cart } from '../../../../../model/cart';
3 import { CartService } from '../../../../../service/cart.service';
4 import { Component, OnInit } from '@angular/core';
5 import { Router } from '@angular/router';
6 import { Sort } from '@angular/material/sort';
7
8 @Component({
9   selector: 'app-cart',
10  templateUrl: './cart.component.html',
11  styleUrls: ['./cart.component.css']
12})
13 export class CartComponent implements OnInit {
14   public cart:Cart=new Cart();
15   public products:Cart[];
16   public sortedCart:Cart[];
17   public grandTotal : number = 0;
18   constructor(private cartService:CartService, private router:Router, private logService:LogService) {
19     this.products=[];
20     this.sortedCart=this.products.slice();
21   }
22
23 ngOnInit(): void {
24   this.logService.sendHeader(1);
25   this.logService.loggedUserId$.subscribe(
26     (id)=>{
27       if(id==''){
28         this.router.navigate(['']);
29       }
30     });
31   this.getCartItemList();
32 }
```

Ln 14, Col 31 Spaces: 2 UTF-8 LF () TypeScript

master* 20°C Sunny 10:31 22-12-2022



Foodbox-Frontend

File Edit Selection View Go Run Terminal Help

EXPLORER

FOODBOX-FRONTEND

- > .angular
- > .vscode
- > node_modules
- < src
 - < app
 - < components
 - < header
 - # header.component...
 - header.component...
 - TS header.component...
 - TS header.components.ts
 - > manage-customers
 - > manage-purchase
 - > order-summary
 - > payment-gateway
 - > product
 - > model
 - > service
 - TS app-routing.module.ts
 - # app.component.css
 - app.component.html

> model

OUTLINE

TIMELINE

header.component.css

```
src > app > components > header > # header.component.css > .header-title
1 .logo{
2   position: relative;
3   top: 8px;
4   height: 80px;
5   width: 185px;
6   cursor: pointer;
7 }
8 .header-title{
9   position: relative;
10  top: 48px;
11  left: 20px;
12  font-family: "Papyrus", Fantasy;
13  font-style: italic;
14  font-size: 55px;
15  color: #332f41;
16 }
17 .btnMain{
18   font-size: 20px;
19 }
20 .btnMain:hover{
21   color: #bf1e2e;
22   background-color: aqua;
23 }
24 .modal-footer button{
25   background-color: #bf1e2e;
26   color: white;
27 }
28 th{
29   color: #bf1e2e;
30 }
```

Ln 14, Col 21 Spaces: 4 UTF-8 CRLF CSS

master* 20°C Sunny 10:31 22-12-2022

File Edit Selection View Go Run Terminal Help

Foodbox-Frontend

EXPLORER

FOODBOX-FRONTEND

- angular
- vscode
- node_modules
- src
 - app
 - components
 - admin-dashboard
 - admin-login
 - cart
 - header
 - # header.component...
 - header.component...
 - TS header.component.ts**
 - header.component.ts
 - manage-customers
 - manage-purchase
 - order-summary
 - payment-gateway
 - product
 - model
 - service
 - app-routing.module.ts
 - # app.component.css
 - app.component.html

OUTLINE

TIMELINE

TS header.component.ts

```
src > app > components > header > TS header.component.ts > HeaderComponent > auth
1 import { LogService } from '../../../../../service/log.service';
2 import { Cart } from '../../../../../model/cart';
3 import { CustomerService } from '../../../../../service/customer.service';
4 import { CartService } from '../../../../../service/cart.service';
5 import { Customer } from '../../../../../model/customer';
6 import { Component, OnInit } from '@angular/core';
7 import { FormBuilder, FormGroup } from '@angular/forms';
8 import { Router } from '@angular/router';
9 import { ProductService } from 'src/app/service/product.service';
10 import { Purchase } from 'src/app/model/purchase';
11 import { PurchaseService } from 'src/app/service/purchase.service';

12 @Component({
13   selector: 'app-header',
14   templateUrl: './header.component.html',
15   styleUrls: ['./header.component.css']
16 })
17 export class HeaderComponent implements OnInit {
18   showHeader:number=0;
19   message:string='';
20   formValue!: FormGroup;
21   customer:Customer = new Customer();
22   auth:any;
23   loginData={
24     email:'',
25     password:'';
26   }
27   login:number=0;
28   productList:any;
29   public cart:Cart=new Cart();
30   public products:Cart[];
```

Ln 23, Col 12 Spaces: 2 UTF-8 LF () TypeScript

20°C Sunny

ENG IN 1031 22-12-2022

File Edit Selection View Go Run Terminal Help

Foodbox-Frontend

EXPLORER

FOODBOX-FRONTEND

- angular
- vscode
- node_modules
- src
 - app
 - components
 - admin-dashboard
 - admin-login
 - cart
 - header
 - # header.component...
 - header.component...
 - TS header.component...**
 - header.component.ts
 - manage-customers
 - manage-purchase
 - order-summary
 - payment-gateway
 - product
 - model
 - service
 - app-routing.module.ts
 - # app.component.css
 - app.component.html

OUTLINE

TIMELINE

TS header.component.html

```
src > app > components > header > TS header.component.html > header > div.navbar.shadow-sm
1 <header *ngIf="showHeader==1">
2   <div class="navbar shadow-sm">
3     <div class="container">
4       <div class="navbar-brand align-item-center d-flex">
5         <div>
6           
7         </div>
8         <strong class="header-title">FoodBox Delicious Restaurent</strong>
9       </div>
10      <div *ngIf="login==0" class="text-right">
11        <button class="btn btnMain" type="button" data-toggle="modal" data-target="#loginModal" id="custLogin">Login</button>
12        <button class="btn btnMain" type="button" data-toggle="modal" data-target="#registerModal" id="cusRegister">Register</button>
13      </div>
14      <div *ngIf="login==1" class="text-right">
15        <button class="btn btnMain" (click)="getActiveOrders()" type="button" data-toggle="modal" data-target="#activeOrdersModal">My Orders</button>
16        <button class="btn btnMain" (click)="customerLogout()">Logout</button>
17      </div>
18    </div>
19    <button *ngIf="login==1" routerLink="cart" class="btn cart">
20      <i style="font-size: 50px;color: #372861;" class="fas fa-cart-plus"></i>
21    </button>
22  </div>
23 <div class="modal fade" id="loginModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
24   <div class="modal-dialog" role="document">
25     <div class="modal-content">
26       <div class="modal-header text-center">
27         <h4 class="modal-title w-100 font-weight-bold">Sign in</h4>
28         <button type="button" class="close" id="cancel" data-dismiss="modal" aria-label="Close"></button>
29       </div>
30     </div>
31   </div>
32 </div>
```

Ln 19, Col 1 Spaces: 4 UTF-8 LF HTML

20°C Sunny

ENG IN 1032 22-12-2022

File Edit Selection View Go Run Terminal Help

manage-customers.component.html

```
src > app > components > manage-customers > manage-customers.component.html > div > nav.navbar.navbar-expand-lg.navbar-light.bg-light
1  <div>
2    <nav class="navbar navbar-expand-lg navbar-light bg-light">
3      <div>
4        
5        <strong class="header-title">The Foodbox</strong>
6      </div>
7      <div class="collapse navbar-collapse" id="navbarNav">
8        <ul class="navbar-nav ml-auto">
9          <li class="nav-item">
10            <button class="btn btnHead" routerLink="/adminDashboard"><i class="fas fa-tasks fa-lg"></i> Ma...
11            Products</button>
12          </li>
13          <li class="nav-item">
14            <button class="btn btnHead active"><i class="fas fa-users fa-lg"></i> Manage Users</button>
15          </li>
16          <li class="nav-item">
17            <button class="btn btnHead" routerLink="/managePurchase"><i class="fas fa-shopping-basket fa-lg"></i> Manage Purchases</button>
18          </li>
19          <li class="nav-item">
20            <button class="btn btnHead float-right" (click)="adminLogout()">Logout</button>
21          </li>
22        </ul>
23      </div>
24    </div>
25    <hr>
26  </div>
27  <div class="container">
28    <span class="text-center prodLogo">Users List</span>
29  </div>
30
```

Ln 25, Col 11 Spaces: 4 UTF-8 LF HTML

master* 20°C Sunny 10:32 22-12-2022

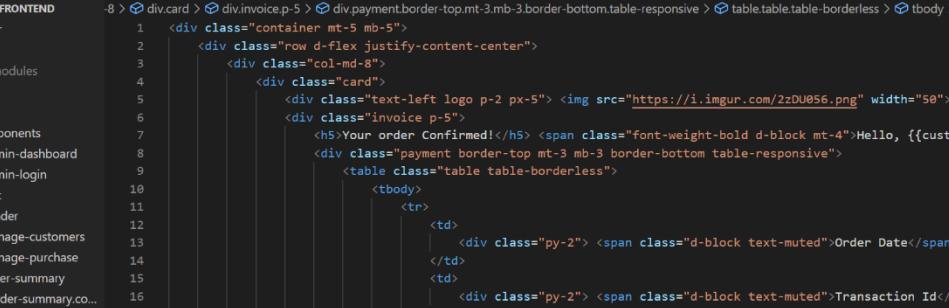
File Edit Selection View Go Run Terminal Help

manage-purchase.component.html

```
src > app > components > manage-purchase > manage-purchase.component.html > div > div.container > table.table.table-striped.table-bordered > tbody > tr
1  <div>
2    <nav class="navbar navbar-expand-lg navbar-light bg-light">
3      <div>
4        
5        <strong class="header-title">The Foodbox</strong>
6      </div>
7      <div class="collapse navbar-collapse" id="navbarNav">
8        <ul class="navbar-nav ml-auto">
9          <li class="nav-item">
10            <button class="btn btnHead" routerLink="/adminDashboard"><i class="fas fa-tasks fa-lg"></i> Ma...
11            Products</button>
12          </li>
13          <li class="nav-item">
14            <button class="btn btnHead active" routerLink="/manageCustomer"><i class="fas fa-users fa-lg"></i> Ma...
15            Users</button>
16          </li>
17          <li class="nav-item">
18            <button class="btn btnHead" routerLink="/managePurchase"><i class="fas fa-shopping-basket fa-lg"></i> Ma...
19            Manage Purchases</button>
20          </li>
21          <li class="nav-item">
22            <button class="btn btnHead float-right" (click)="adminLogout()">Logout</button>
23          </li>
24        </ul>
25      </div>
26      <hr>
27    </div>
28    <div class="container">
29
```

Ln 50, Col 54 Spaces: 4 UTF-8 LF HTML

master* 20°C Sunny 10:32 22-12-2022



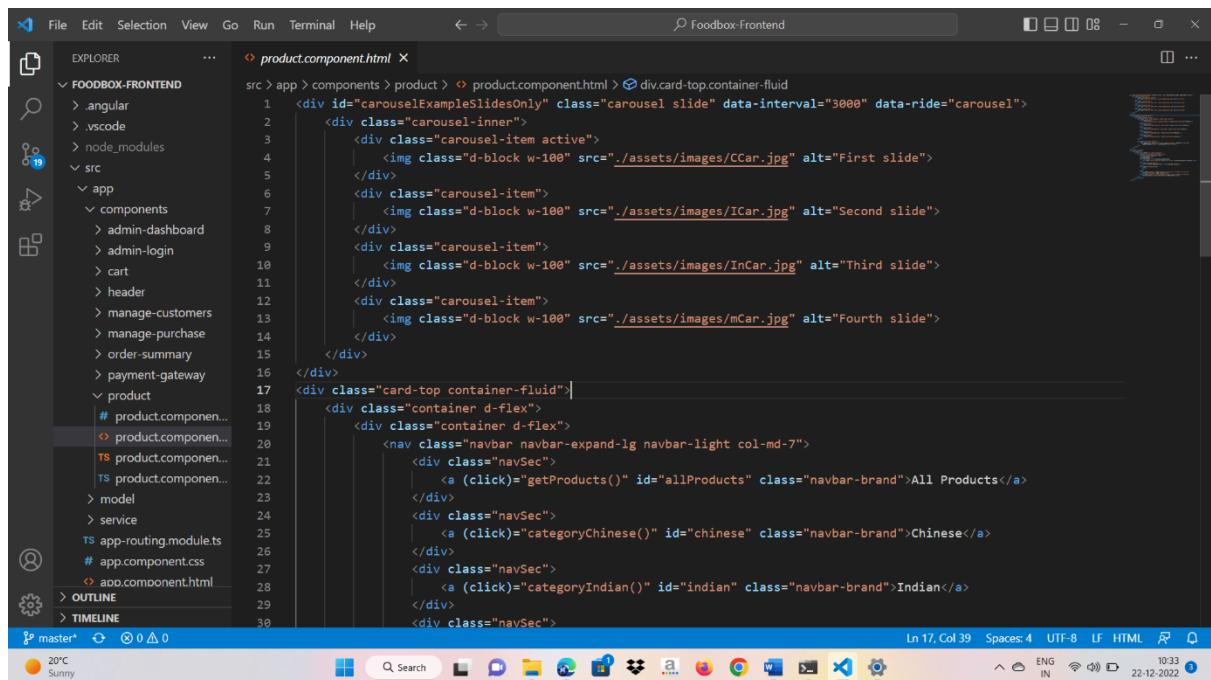
The screenshot shows a browser window with the title "Foodbox-Frontend". The page content is an order summary. At the top, there's a header with a logo and some text. Below it is a table with two rows. The first row contains columns for Order Date, Transaction Id, Payment, and Shipping Address. The second row contains a single column with a long text string. The code editor on the left shows the HTML and CSS files for this component.

```
<div class="container mt-5 mb-5">
  <div class="row d-flex justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="text-left logo p-2 px-5">  </div>
        <div class="invoice p-5">
          <h5>Your order Confirmed!</h5> <span class="font-weight-bold d-block mt-4">Hello, {{customerName}}!</span>
          <div class="payment border-top mt-3 mb-3 border-bottom table-responsive">
            <table class="table table-borderless">
              <tbody>
                <tr>
                  <td>Order Date</td>
                  <td>Transaction Id</td>
                  <td>Payment</td>
                  <td>Shipping Address</td>
                </tr>
              </tbody>
            </table>
          </div>
          <div class="product border-bottom table-responsive">
            <table class="table table-borderless">
              <tbody>
```

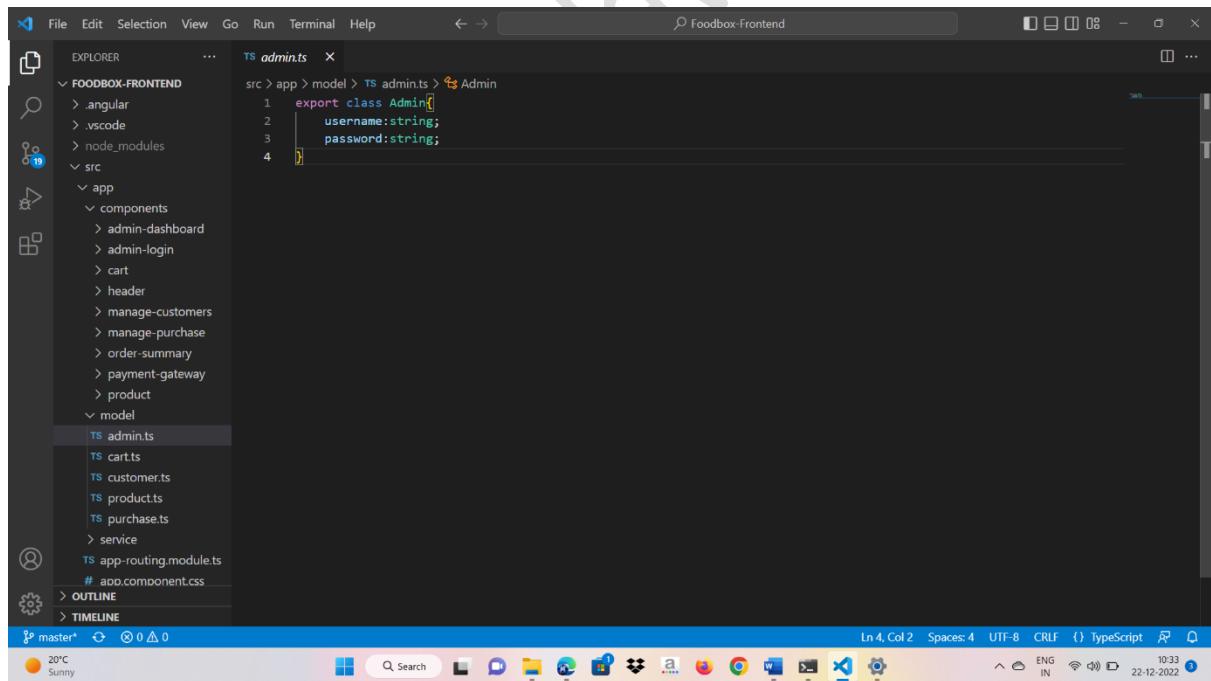
The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "FOODBOX-FRONTEND". The "payment-gateway" folder is expanded, showing files like payment-gateway.c, payment-gateway.ts, payment-gateway.css, product, model, service, app-routing.module.ts, app.component.css, and app.component.html.
- Editor:** The main editor area displays the content of "payment-gateway.component.html".
- Search Bar:** At the top, it says "Foodbox-Frontend".
- Bottom Status Bar:** Shows "master" as the current branch, file statistics (18, Col 16, Spaces: 4, UTF-8), and icons for HTML, CSS, JS, and a refresh button.
- Bottom Taskbar:** Includes icons for File, Edit, Selection, View, Go, Run, Terminal, Help, and various system icons.

```
<div>.row.d-flex justify-content-center > <div>.col-1xl-7 col-lg-8 col-md-9 col-sm-11 > <div>.card.border-0 > <div>.row > <div>.col-sm-7 border-line.pb-3 > <div>.form-group<br/><div>.container-fluid px-1 px-md-2 px-lg-4 py-5 mx-auto<br/><div>.row d-flex justify-content-center<br/><div>.col-1xl-7 col-lg-8 col-md-9 col-sm-11 > <div>.card border-0<br/><div>.row justify-content-center<br/>| <h3>Credit Card Checkout</h3></div><div>.row<br/>| <div>.col-sm-7 border-line pb-3 > <div>.form-group<br/>| <p>Name on the card</p> <input type="text" name="name"></div><div>.form-group<br/>| <p>Card Number</p> <input type="text" name="card-num" placeholder="0000 0000 0000 0000" ><div>.mb-0 ml-3 ></div> </div></div><div>.form-group<br/>| <p>Expiry date</p> <input type="text" name="exp" placeholder="MM/YY" ></div><div>.form-group<br/>| <p>CVV/CVC</p> <input type="password" name="cvv" placeholder="CVV" ></div></div><div>.col-sm-5 text-sm-center justify-content-center pt-4 pb-4 > <small>Payment amount</small><br/><h5>{{transId}}</h5> <small>Payment amount</small></div><div>.row px-3 justify-content-sm-center<br/>| <h2><span>₹{{totalCost}}</span><span>₹{{totalCost}}</span></h2></div>
```



```
product.component.html
src > app > components > product > product.component.html > div.card-top.container-fluid
1   <div id="carouselExampleSlidesOnly" class="carousel slide" data-interval="3000" data-ride="carousel">
2     <div class="carousel-inner">
3       <div class="carousel-item active">
4         
5       </div>
6       <div class="carousel-item">
7         
8       </div>
9       <div class="carousel-item">
10        
11      </div>
12      <div class="carousel-item">
13        
14      </div>
15    </div>
16  </div>
17  <div class="card-top container-fluid">
18    <div class="container d-flex">
19      <div class="container d-flex">
20        <nav class="navbar navbar-expand-lg navbar-light col-md-7">
21          <div class="navSec">
22            <a (click)="getProducts()" id="allProducts" class="navbar-brand">All Products</a>
23          </div>
24          <div class="navSec">
25            <a (click)="categoryChinese()" id="chinese" class="navbar-brand">Chinese</a>
26          </div>
27          <div class="navSec">
28            <a (click)="categoryIndian()" id="indian" class="navbar-brand">Indian</a>
29          </div>
30        </div>
31      </div>
32    </div>
33  </div>
```



```
admin.ts
src > app > model > admin.ts > Admin
1  export class Admin {
2    username:string;
3    password:string;
4 }
```

The screenshot shows the Visual Studio Code interface with the title bar "Foodbox-Frontend". The left sidebar displays the project structure under "EXPLORER" with "FOODBOX-FRONTEND" selected. The "src" folder contains "app" and "model" subfolders. In the "app" folder, "cart.ts" is currently open in the editor. The code defines a class "Cart" with properties: id: number, quantity: number, price: number, and product: Product.

```
src > app > model > TS cart.ts > Cart
1 import { Product } from "./product";
2 export class Cart{
3     id:number;
4     quantity:number;
5     price:number;
6     product:Product;
7 }
```

The screenshot shows the Visual Studio Code interface with the title bar "Foodbox-Frontend". The left sidebar displays the project structure under "EXPLORER" with "FOODBOX-FRONTEND" selected. The "src" folder contains "app" and "model" subfolders. In the "app" folder, "customer.ts" is currently open in the editor. The code defines a class "Customer" with properties: email: string, password: string, name: string, contact: string, and address: string.

```
src > app > model > TS customer.ts > Customer
1 export class Customer{
2     email:string;
3     password:string;
4     name:string;
5     contact:string;
6     address:string;
7 }
```

The screenshot shows the VS Code interface with the title bar "Foodbox-Frontend". The left sidebar (EXPLORER) shows the project structure under "FOODBOX-FRONTEND". The main editor tab is "product.ts" with the following code:

```
src > app > model > TS products > Product
1  export class Product {
2    id:number;
3    name:string;
4    desc:string;
5    category:string;
6    actualPrice:number;
7    discount:number;
8    price:number;
9    avail:string;
10   imagepath:string;
11 }
```

The status bar at the bottom shows "Ln 11, Col 2" and "TypeScript".

The screenshot shows the VS Code interface with the title bar "Foodbox-Frontend". The left sidebar (EXPLORER) shows the project structure under "FOODBOX-FRONTEND". The main editor tab is "purchase.ts" with the following code:

```
src > app > model > purchase.ts > Purchase
1  import { Customer } from "./customer";
2  export class Purchase {
3    id:number;
4    dop:Date;
5    productname:string;
6    quantity:number;
7    totalcost:number;
8    transactionid:string;
9    customer:Customer;
10 }
```

The status bar at the bottom shows "Ln 10, Col 2" and "TypeScript".

VS Code interface showing the `customer.service.ts` file in the `FOODBOX-FRONTEND` project. The code defines a `CustomerService` class with methods for adding customers, logging in, getting customers, searching for customers, and deleting customers.

```
src > app > service > TS customer.service.ts < CustomerService > baseURL
1 import { Observable } from 'rxjs';
2 import { Customer } from './model/customer';
3 import { HttpClient } from '@angular/common/http';
4 import { Injectable } from '@angular/core';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class CustomerService {
10   private baseURL="http://localhost:8686/customers";
11   constructor(private httpClient:HttpClient) { }
12
13   addCustomer(customer:Customer):Observable<Object>{
14     return this.httpClient.post(`${this.baseURL}`,customer);
15   }
16
17   customerLogin(loginData:any):Observable<Object>{
18     return this.httpClient.post(`${this.baseURL}/${loginData.email}`,loginData);
19   }
20
21   getCustomers():Observable<Customer[]>{
22     return this.httpClient.get<Customer[]>(`${this.baseURL}`);
23   }
24
25   searchCustomer(keyword:any):Observable<Customer[]>{
26     return this.httpClient.get<Customer[]>(`${this.baseURL}/search/${keyword}`);
27   }
28
29   deleteCustomer(email:string):Observable<Customer>{
30     return this.httpClient.delete<Customer>(`${this.baseURL}/${email}`);
31   }
32 }
```

VS Code interface showing the `admin.service.ts` file in the `FOODBOX-FRONTEND` project. The code defines an `AdminService` class with a method for logging in administrators.

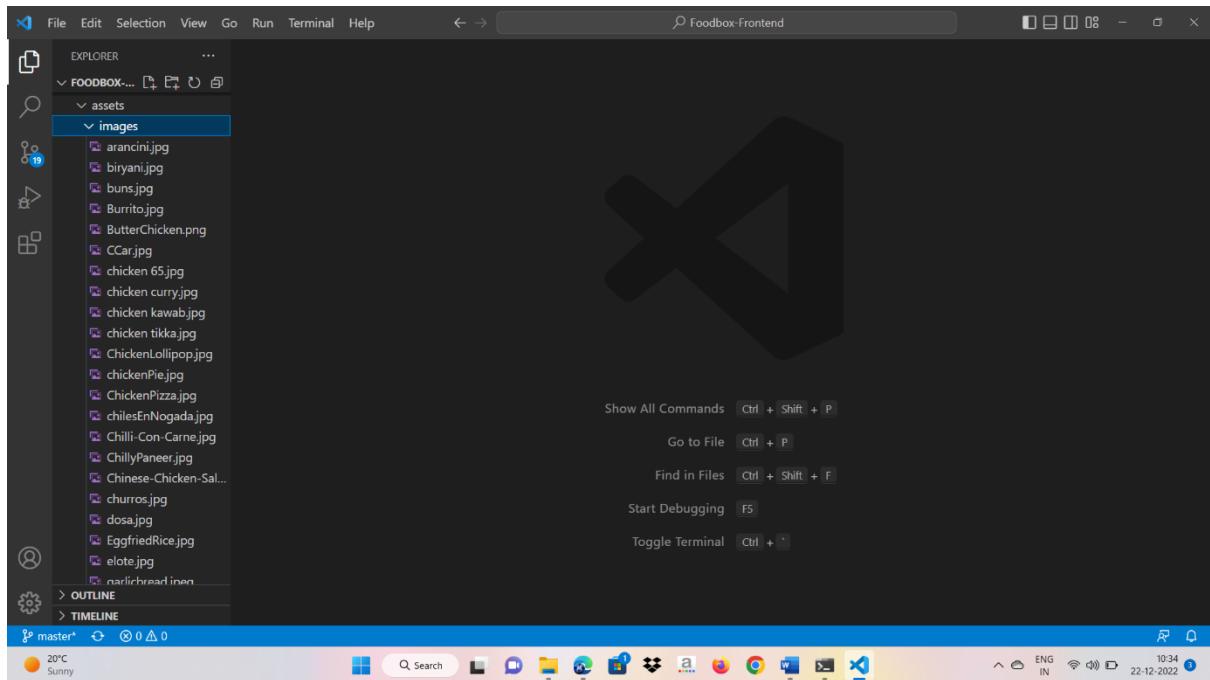
```
src > app > service > TS admin.service.ts < AdminService
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class AdminService {
9   private baseURL="http://localhost:8686/admin";
10  constructor(private httpClient:HttpClient) { }
11
12  adminLogin(loginData:any):Observable<Object>{
13    return this.httpClient.post(`${this.baseURL}/${loginData.username}`,loginData);
14  }
15 }
```

VS Code screenshot showing the Explorer sidebar with the 'FOODBOX-FRONTEND' folder selected. The 'app.module.ts' file is open in the editor. The code defines the AppModule with declarations, imports, and exports.

```
src > app > TS app.module.ts > AppModule
1.4 import { AdminLoginComponent } from './components/admin-login/admin-login.component';
1.5 import { AdminDashboardComponent } from './components/admin-dashboard/admin-dashboard.component';
1.6 import { ManageCustomersComponent } from './components/manage-customers/manage-customers.component';
1.7 import { ManagePurchaseComponent } from './components/manage-purchase/manage-purchase.component';
1.8 import { PaymentGatewayComponent } from './components/payment-gateway/payment-gateway.component';
1.9 import { OrderSummaryComponent } from './components/order-summary/order-summary.component';
20
21 @NgModule({
22   declarations: [
23     AppComponent,
24     HeaderComponent,
25     ProductComponent,
26     CartComponent,
27     AdminLoginComponent,
28     AdminDashboardComponent,
29     ManageCustomersComponent,
30     ManagePurchaseComponent,
31     PaymentGatewayComponent,
32     OrderSummaryComponent,
33   ],
34   imports: [
35     BrowserModule,
36     AppRoutingModule,
37     HttpClientModule,
38     FormsModule,
39     ReactiveFormsModule,
40     BrowserAnimationsModule,
41     MatSnackBarModule,
42     MatSortModule
43   ],
44 })
45 export class AppModule { }
```

VS Code screenshot showing the Explorer sidebar with the 'FOODBOX-FRONTEND' folder selected. The 'app-routing.module.ts' file is open in the editor. The code defines the AppRoutingModule with routes and exports the AppRoutingModule.

```
src > app > TS app-routing.module.ts > ...
1 import { ManageCustomersComponent } from './components/manage-customers/manage-customers.component';
2 import { AdminDashboardComponent } from './components/admin-dashboard/admin-dashboard.component';
3 import { AdminLoginComponent } from './components/admin-login/admin-login.component';
4 import { ProductComponent } from './components/product/product.component';
5 import { NgModule } from '@angular/core';
6 import { Routes, RouterModule } from '@angular/router';
7 import { CartComponent } from './components/cart/cart.component';
8 import { ManagePurchaseComponent } from './components/manage-purchase/manage-purchase.component';
9 import { PaymentGatewayComponent } from './components/payment-gateway/payment-gateway.component';
10 import { OrderSummaryComponent } from './components/order-summary/order-summary.component';
11
12 const routes: Routes = [
13   {path: '', redirectTo: 'products', pathMatch: 'full'},
14   {path: 'products', component: ProductComponent},
15   {path: 'cart', component: CartComponent},
16   {path: 'admin', component: AdminLoginComponent},
17   {path: 'adminDashboard', component: AdminDashboardComponent},
18   {path: 'manageCustomer', component: ManageCustomersComponent},
19   {path: 'managePurchase', component: ManagePurchaseComponent},
20   {path: 'paymentGateway', component: PaymentGatewayComponent},
21   {path: 'orderSummary', component: OrderSummaryComponent}
22 ];
23
24 @NgModule({
25   imports: [RouterModule.forRoot(routes)],
26   exports: [RouterModule]
27 })
28 export class AppRoutingModule { }
```



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Foodbox-most trusted</title>
    <meta href="/" content="width=device-width, initial-scale=1">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="favicon.ico">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1K9KRQjQERJvTumQh0nPEzvF6L/Z6rnvJ3oUOFUpCjeEUQuaq2+l" crossorigin="anonymous">
    <script src="https://kit.fontawesome.com/3fe99d9367.js" crossorigin="anonymous"></script>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link rel="preconnect" href="https://fonts.googleapis.com/css?family=WindSong&display=swap" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap" rel="stylesheet">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Roboto:wght@300;400;500&display=swap" rel="stylesheet">
  <ad>
    <y class="mat-typography">
      <pp-root></pp-root>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zY4C+OgpamoFV38MVBnE+IbbVVUew+OrCxRkfj" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-Piv4xVRNyNqPqkSzby6br4ghJ7DXjk09RmUpJ8jgGtD7zP9yug3goQfGII0yAns" crossorigin="anonymous"></script>
  </ad>
<hr>
```

This screenshot shows the code editor for the 'index.html' file. The code is an Angular template for a landing page. It includes meta tags for the title and viewport, a link to a favicon, and several links to CSS and JS files from Google Fonts and CDNs. A comment block is present for advertising.

The screenshot shows a code editor window with the title "Foodbox-Frontend". The left sidebar displays the project structure under "FOODBOX-FRONTEND". The main editor area shows the content of "styles.css". The status bar at the bottom indicates "Ln 9, Col 1" and "Spaces: 4". The taskbar at the bottom of the screen shows various application icons.

```
src > # styles.css > ...
1  /* You can add global styles to this file, and also import other style files */
2  html, body { height: 100%; }
3  body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif; background-color:
4      #f8fcff }
5
6  hr { border-top: 1px dashed grey; }
7  html, body { height: 100%; }
8
9
```

The screenshot shows a terminal window titled "npm" with the command "ng serve" run. The output shows the browser application bundle generation complete and lists the initial chunk files with their sizes. It also shows the build time and hash, and the fact that the angular live development server is listening on port 4200. The status bar at the bottom indicates "10:57" and "22-12-2022".

```
Microsoft Windows [Version 10.0.22621.963]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Capstone\Foodbox-Frontend>ng serve
/ Browser application bundle generation complete.

Initial Chunk Files    | Names          | Raw Size
vendor.js              | vendor        | 3.58 MB
styles.css, styles.js  | styles        | 328.00 kB
polyfills.js           | polyfills    | 314.29 kB
main.js                | main         | 206.23 kB
runtime.js             | runtime      | 6.53 kB
                           | Initial Total | 4.41 MB

Build at: 2022-12-22T05:26:37.656Z - Hash: 2e57e542578a5e52 - Time: 8718ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

/ Compiled successfully.
/ Browser application bundle generation complete.

5 unchanged chunks

Build at: 2022-12-22T05:26:39.329Z - Hash: 2e57e542578a5e52 - Time: 1361ms
/ Compiled successfully.
```