

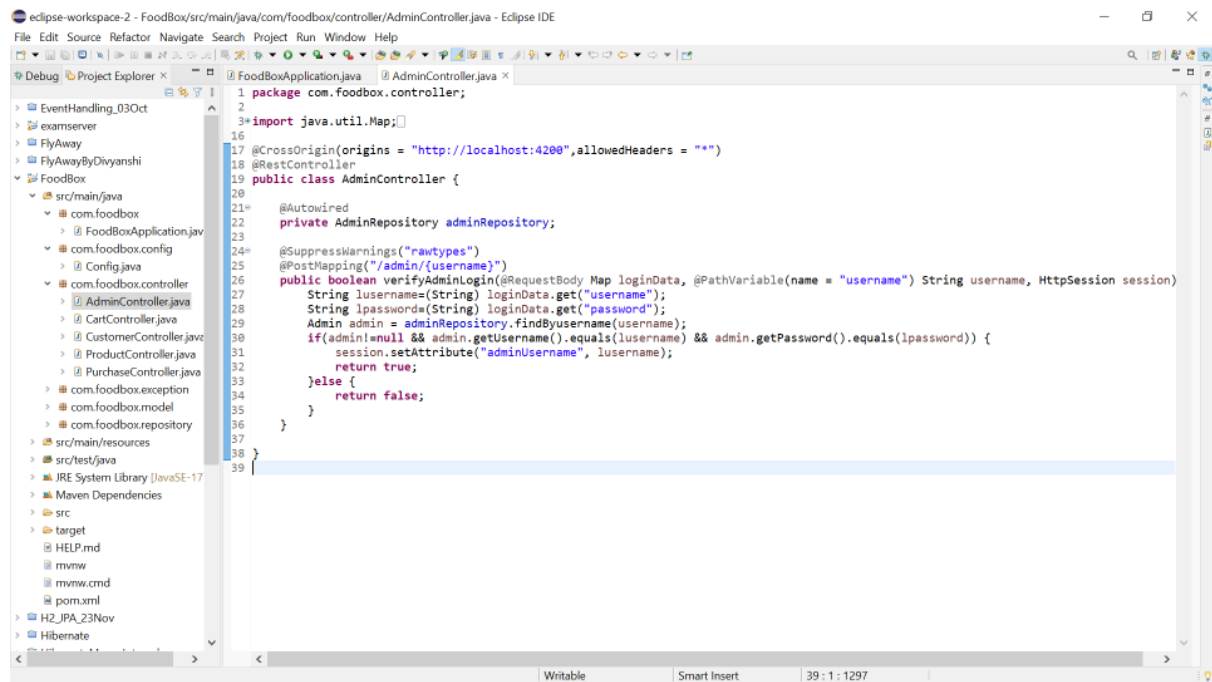
Project Screenshots

BACK-END

Project Structure



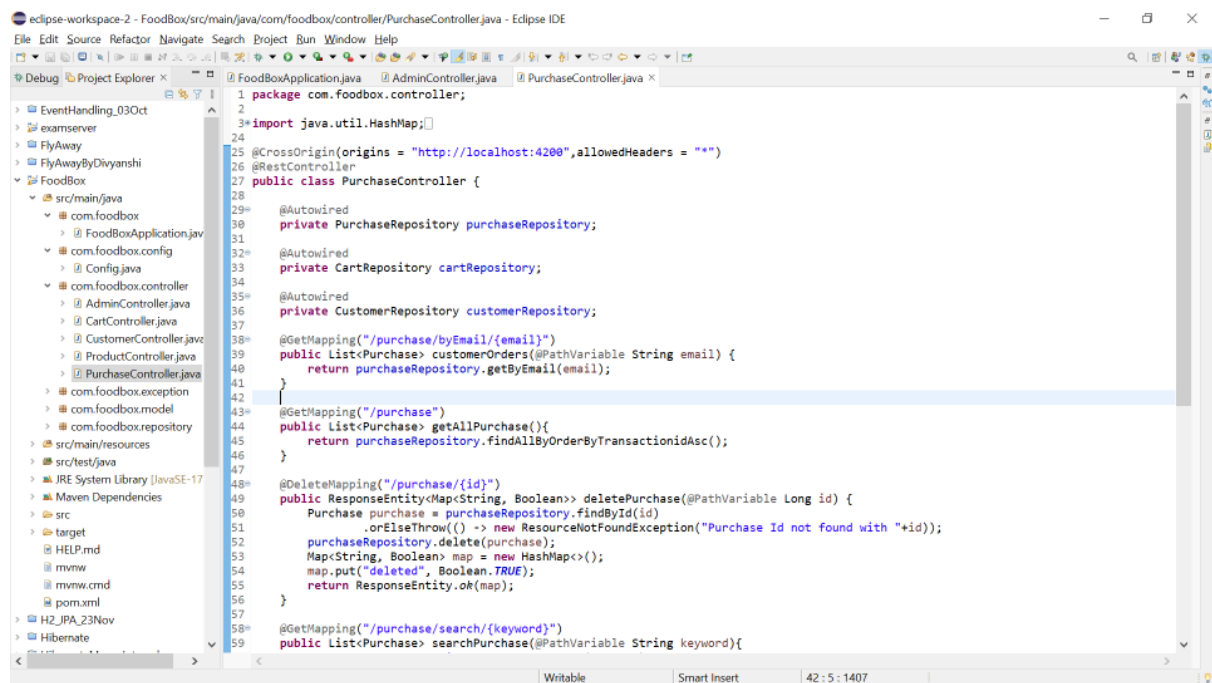
Few snapshots of the Controller classes –



```

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    @SuppressWarnings("rawtypes")
13    @PostMapping("/admin/{username}")
14    public boolean verifyAdminLogin(@RequestBody Map loginData, @PathVariable(name = "username") String username, HttpSession session) {
15        String username = loginData.get("username");
16        String password = loginData.get("password");
17        Admin admin = adminRepository.findByUsername(username);
18        if (admin != null && admin.getUsername().equals(username) && admin.getPassword().equals(password)) {
19            session.setAttribute("adminUsername", username);
20            return true;
21        } else {
22            return false;
23        }
24    }
25 }

```



```

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.findAllByOrderByTransactionIdAsc();
26    }
27
28    @DeleteMapping("/purchase/{id}")
29    public ResponseEntity<Map<String, Boolean>> deletePurchase(@PathVariable Long id) {
30        Purchase purchase = purchaseRepository.findById(id)
31            .orElseThrow(() -> new ResourceNotFoundException("Purchase Id not found with " + id));
32        purchaseRepository.delete(purchase);
33        Map<String, Boolean> map = new HashMap<>();
34        map.put("deleted", Boolean.TRUE);
35        return ResponseEntity.ok(map);
36    }
37
38    @GetMapping("/purchase/search/{keyword}")
39    public List<Purchase> searchPurchase(@PathVariable String keyword) {
40
41    }
42 }

```

```

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("/carts")
14    public Cart addToCart(@RequestBody Cart cart, HttpSession session) {
15        float grandTotal=0;
16        if(session.getAttribute("grandTotal")==null) {
17            grandTotal=0;
18        }
19        else {
20            grandTotal=(float) session.getAttribute("grandTotal");
21        }
22        List<Cart> cartList = cartRepository.findAll();
23        for(Cart temp:cartList) {
24            if(temp.getProduct().getId()==cart.getProduct().getId()) {
25                int tempQuantity = 1+temp.getQuantity();
26                grandTotal=grandTotal+temp.getPrice();
27                session.setAttribute("grandTotal", grandTotal);
28                temp.setQuantity(tempQuantity);
29                temp.setPrice(temp.getProduct().getPrice()*tempQuantity);
30                return cartRepository.save(temp);
31            }
32        }
33        int min = 100;
34        int max = 999;
35        int b = (int) (Math.random() * (max - min + 1) + min);
36        cart.setId(b);
37        cart.setQuantity(1);
38        cart.setPrice(cart.getProduct().getPrice());
39        grandTotal=grandTotal+cart.getProduct().getPrice();
40    }
41 }

```

Few snapshots of the model classes -

```

1 package com.foodbox.model;
2
3 import jakarta.persistence.Entity;
4
5 @Entity
6 public class Admin {
7
8     @Id
9     private String username;
10    private String password;
11
12    public Admin() {
13        super();
14    }
15
16    public Admin(String username, String password) {
17        super();
18        this.username = username;
19        this.password = password;
20    }
21
22    public String getUsername() {
23        return username;
24    }
25
26    public void setUsername(String username) {
27        this.username = username;
28    }
29
30    public String getPassword() {
31        return password;
32    }
33
34    public void setPassword(String password) {
35        this.password = password;
36    }
37 }

```

The screenshot shows the Eclipse IDE with the 'Cart.java' file open. The file is located in the package 'com.foodbox.model'. The code defines a 'Cart' class that implements the 'Entity' interface. It has private fields for 'id', 'quantity', 'price', and 'product'. The class includes a constructor that takes these fields as parameters and initializes them. It also has getter and setter methods for 'id' and 'quantity'.

```

1 package com.foodbox.model;
2
3 import jakarta.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

```

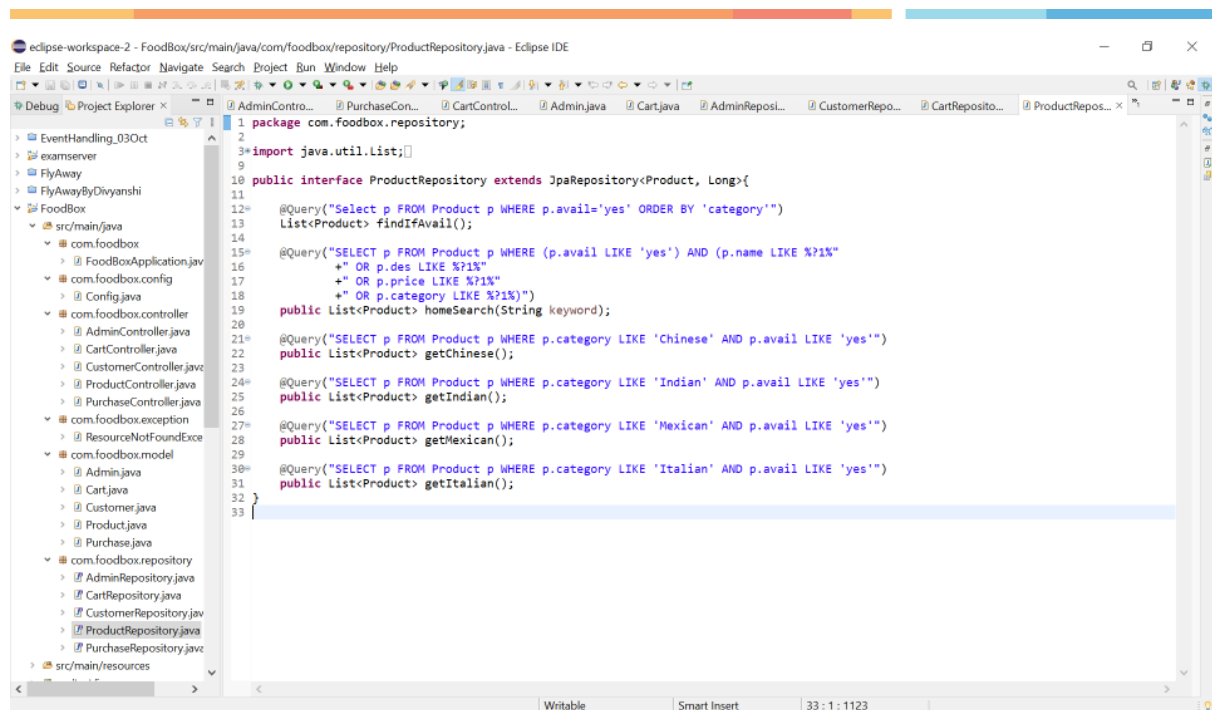
Few snapshots of the repository classes -

The screenshot shows the Eclipse IDE with the 'CustomerRepository.java' file open. The file is located in the package 'com.foodbox.repository'. It defines a 'CustomerRepository' interface that extends 'JpaRepository<Customer, String>'. It includes a method 'findByEmail' and a 'searchCustomer' method that uses a JPQL query to search for customers by keyword.

```

1 package com.foodbox.repository;
2
3 import java.util.List;
4
5 public interface CustomerRepository extends JpaRepository<Customer, String> {
6     Customer findByEmail(String email);
7
8     @Query("SELECT c FROM Customer c WHERE c.email LIKE ?1%"+
9         " OR c.name LIKE ?1%"+
10        " OR c.contact LIKE ?1%"+
11        " OR c.address LIKE ?1%")
12     public List<Customer> searchCustomer(String keyword);
13

```

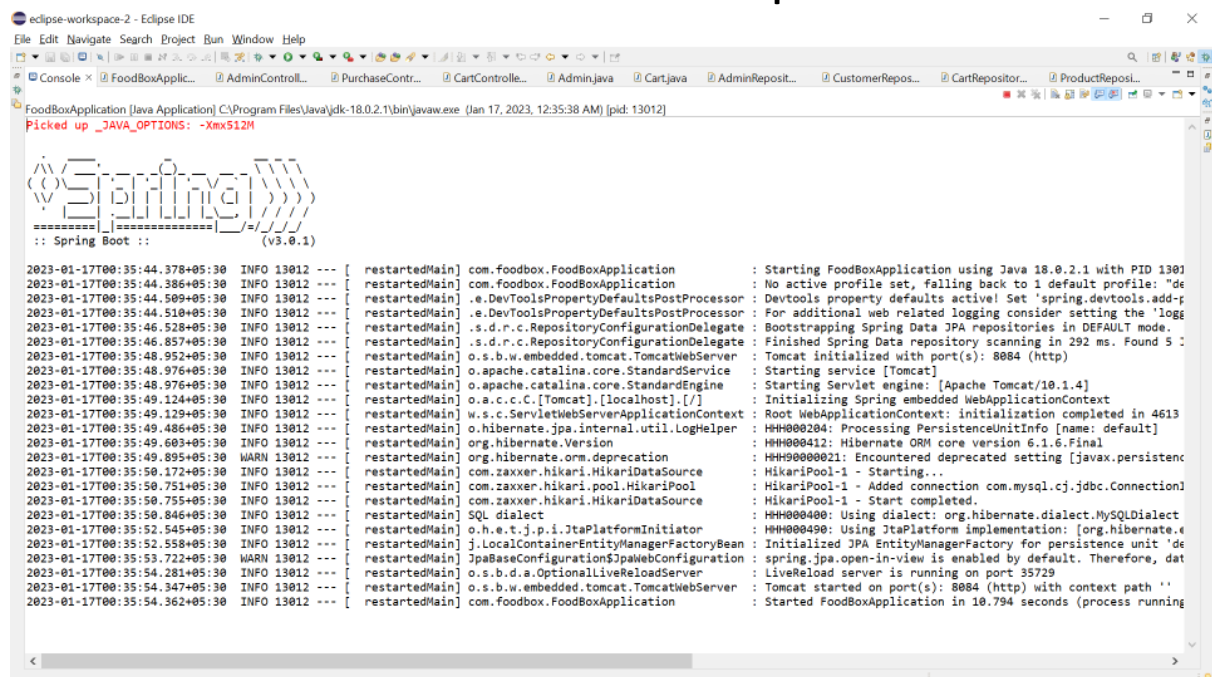


```

1 package com.foodbox.repository;
2
3 import java.util.List;
4
5 public interface ProductRepository extends JpaRepository<Product, Long>{
6
7     @Query("SELECT p FROM Product p WHERE p.avail='yes' ORDER BY 'category'")
8     List<Product> findIfAvail();
9
10    @Query("SELECT p FROM Product p WHERE (p.avail LIKE 'yes') AND (p.name LIKE %?1%"+
11        " OR p.des LIKE %?1%"+
12        " OR p.price LIKE %?1%"+
13        " OR p.category LIKE %?1%")
14    public List<Product> homeSearch(String keyword);
15
16    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Chinese' AND p.avail LIKE 'yes'")
17    public List<Product> getChinese();
18
19    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Indian' AND p.avail LIKE 'yes'")
20    public List<Product> getIndian();
21
22    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Mexican' AND p.avail LIKE 'yes'")
23    public List<Product> getMexican();
24
25    @Query("SELECT p FROM Product p WHERE p.category LIKE 'Italian' AND p.avail LIKE 'yes'")
26    public List<Product> getItalian();
27
28 }

```

Execution of the backend part -



```

Picked up _JAVA_OPTIONS: -Xmx512M

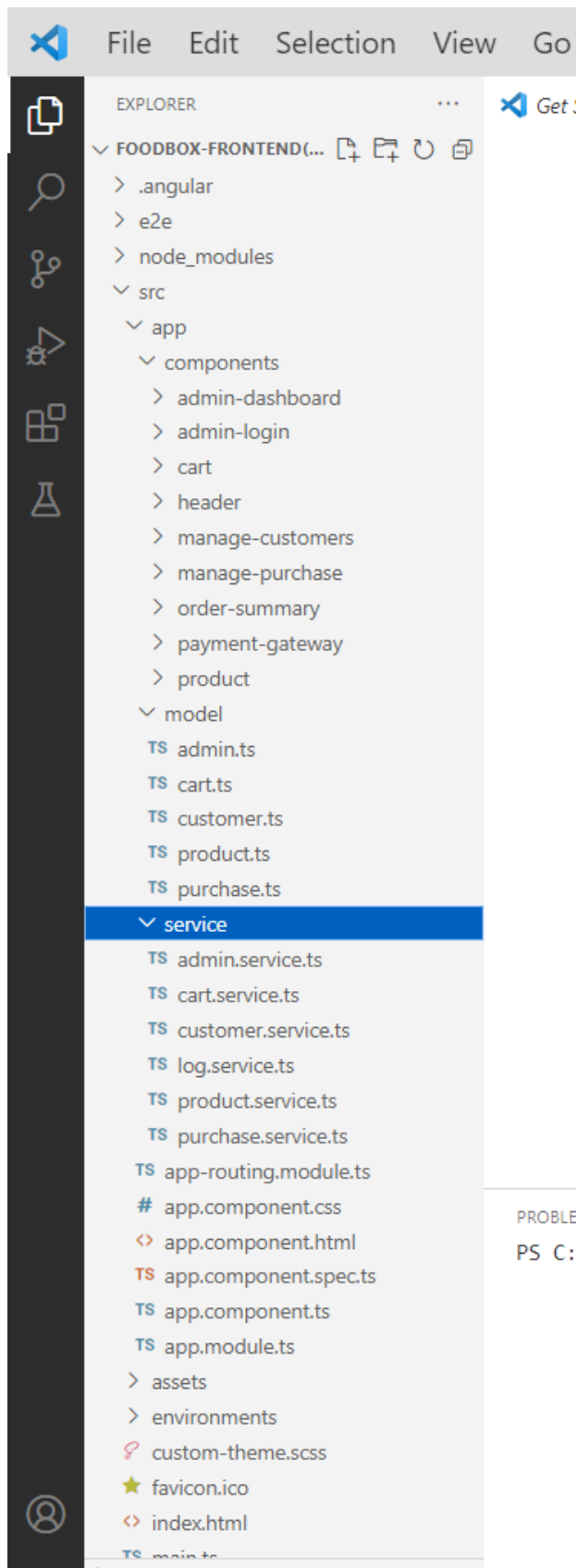
:: Spring Boot ::
(v3.0.1)

2023-01-17T00:35:44.378+05:30 INFO 13012 --- [ restartedMain ] com.foodbox.FoodBoxApplication : Starting FoodBoxApplication using Java 18.0.2.1 with PID 13012
2023-01-17T00:35:44.386+05:30 INFO 13012 --- [ restartedMain ] com.foodbox.FoodBoxApplication : No active profile set, falling back to 1 default profile: "default"
2023-01-17T00:35:44.509+05:30 INFO 13012 --- [ restartedMain ] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-p
2023-01-17T00:35:44.518+05:30 INFO 13012 --- [ restartedMain ] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logg
2023-01-17T00:35:46.528+05:30 INFO 13012 --- [ restartedMain ] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-01-17T00:35:46.857+05:30 INFO 13012 --- [ restartedMain ] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 292 ms. Found 5 :
2023-01-17T00:35:48.952+05:30 INFO 13012 --- [ restartedMain ] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8084 (http)
2023-01-17T00:35:48.976+05:30 INFO 13012 --- [ restartedMain ] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-01-17T00:35:48.976+05:30 INFO 13012 --- [ restartedMain ] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.4]
2023-01-17T00:35:49.124+05:30 INFO 13012 --- [ restartedMain ] o.s.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-01-17T00:35:49.129+05:30 INFO 13012 --- [ restartedMain ] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 4613
2023-01-17T00:35:49.486+05:30 INFO 13012 --- [ restartedMain ] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-01-17T00:35:49.603+05:30 INFO 13012 --- [ restartedMain ] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.1.6.Final
2023-01-17T00:35:49.895+05:30 WARN 13012 --- [ restartedMain ] org.hibernate.orm.deprecation : HHH90000021: Encountered deprecated setting [javax.persistence
2023-01-17T00:35:50.172+05:30 INFO 13012 --- [ restartedMain ] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-01-17T00:35:50.751+05:30 INFO 13012 --- [ restartedMain ] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionI
2023-01-17T00:35:50.755+05:30 INFO 13012 --- [ restartedMain ] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-01-17T00:35:50.846+05:30 INFO 13012 --- [ restartedMain ] SQL dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
2023-01-17T00:35:52.545+05:30 INFO 13012 --- [ restartedMain ] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.e
2023-01-17T00:35:52.558+05:30 INFO 13012 --- [ restartedMain ] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'de
2023-01-17T00:35:53.722+05:30 WARN 13012 --- [ restartedMain ] jpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, dat
2023-01-17T00:35:54.281+05:30 INFO 13012 --- [ restartedMain ] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-01-17T00:35:54.347+05:30 INFO 13012 --- [ restartedMain ] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8084 (http) with context path ''
2023-01-17T00:35:54.362+05:30 INFO 13012 --- [ restartedMain ] com.foodbox.FoodBoxApplication : Started FoodBoxApplication in 10.794 seconds (process running

```

FRONT-END

Project Structure



The screenshot shows the Visual Studio Code editor with the 'app.component.html' file open. The Explorer sidebar on the left shows the project structure for 'FOODBOX-FRONTEND(ANGULARHTMLCSS)'. The file 'app.component.html' is selected, and its content is displayed in the editor. The content is a simple Angular component template with an app-header and a router-outlet. The terminal at the bottom shows the command prompt for the project directory.

```

src > app > > app.component.html > ...
1 <app-header></app-header>
2 <router-outlet></router-outlet>
3

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\foodbox\FoodBox-FrontEnd(AngularHTMLCSS)>

The screenshot shows the Visual Studio Code editor with the 'index.html' file open. The Explorer sidebar on the left shows the project structure for 'FOODBOX-FRONTEND(ANGULARHTMLCSS)'. The file 'index.html' is selected, and its content is displayed in the editor. The content is a full HTML document with a head section containing meta tags, a title, a base href, and various links for fonts, CSS, and JavaScript. The body section contains an app-root and a script for jQuery.

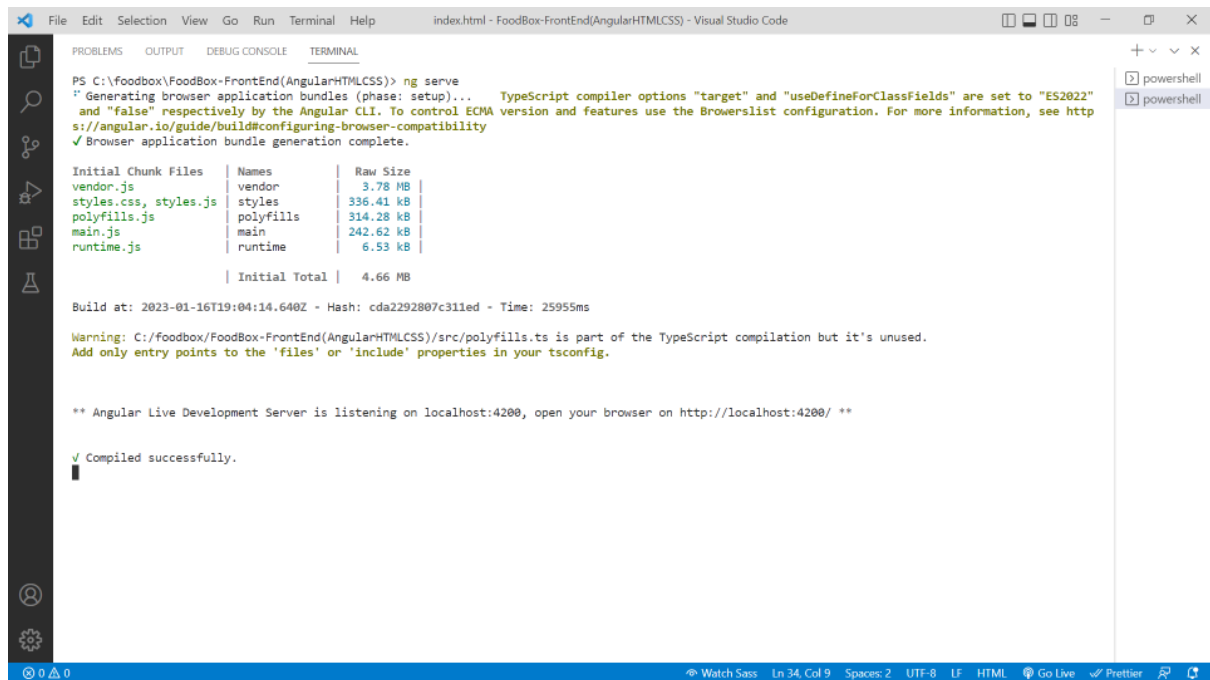
```

src > index.html > > html > > body.mat-typography > > hr
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5 <meta charset="utf-8">
6 <title>Foodbox</title>
7 <base href="/">
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="icon" type="image/x-icon" href="favicon.ico">
10 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
11 integrity="sha384-B0vP5xmATw1+K9KRQJQERJvTumQw0nPEzvF6L/Z6nronJ3oUOFpCjEUQouq2+1" crossorigin="anonymous">
12 <script src="https://kit.fontawesome.com/3fe90d9367.js" crossorigin="anonymous"></script>
13 <link
14 href="https://fonts.googleapis.com/css?family=Lato:100,300,400,700,900"
15 rel="stylesheet"
16 />
17 <link rel="preconnect" href="https://fonts.googleapis.com">
18 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
19 <link href="https://fonts.googleapis.com/css2?family=WindSong&display=swap" rel="stylesheet">
20 <link href="https://fonts.googleapis.com/css2?family=Roboto:300,400,500&display=swap" rel="stylesheet">
21 <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
22 <link rel="preconnect" href="https://fonts.gstatic.com">
23 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap" rel="stylesheet">
24 </head>
25
26 <body class="mat-typography">
27 <app-root></app-root>
28 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
29 integrity="sha384-DfXdz2htPH0lsSS55nCTpuj/zy4C/OgpmoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
30 crossorigin="anonymous"></script>
31 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js"
32 integrity="sha384-Piv4xVNRyMGpQkS2by6br4gNj7DXjQk09RmUpJ8jG6tD7zP9yug3goQf6I10yAns"

```

Execution of the above project

On terminal (In the project directory) – ng serve



```

PS C:\foodbox\FoodBox-FrontEnd(AngularHTMLCSS)> ng serve
* Generating browser application bundles (phase: setup)...   TypeScript compiler options "target" and "useDefineForClassFields" are set to "ES2022"
and "false" respectively by the Angular CLI. To control ECMA version and features use the Browserslist configuration. For more information, see http
s://angular.io/guide/build#configuring-browser-compatibility
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor         | 3.78 MB
styles.css, styles.js | styles        | 336.41 kB
polyfills.js        | polyfills      | 314.28 kB
main.js             | main           | 242.62 kB
runtime.js          | runtime        | 6.53 kB
                    | Initial Total  | 4.66 MB

Build at: 2023-01-16T19:04:14.640Z - Hash: cda2292807c311ed - Time: 25955ms

Warning: C:/foodbox/FoodBox-FrontEnd(AngularHTMLCSS)/src/polyfills.ts is part of the TypeScript compilation but it's unused.
Add only entry points to the 'files' or 'include' properties in your tsconfig.

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
  
```

Database –

