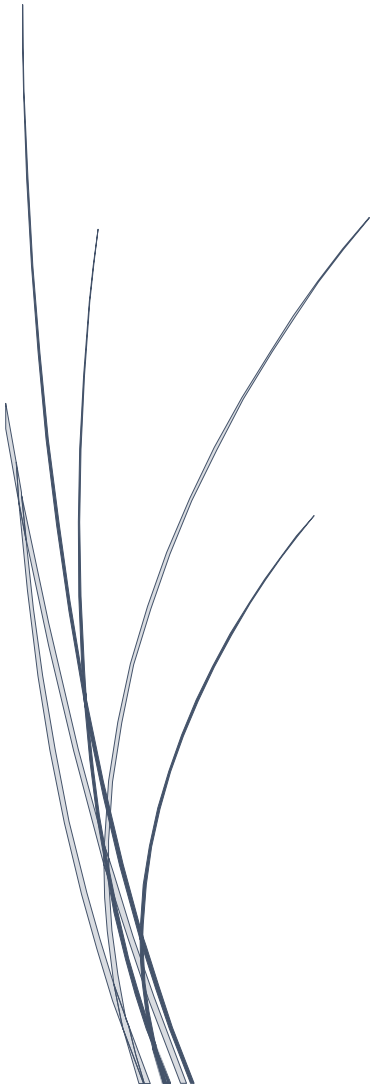


# Full Stack Java Developer

## Phase-6: Capstone Project

- **Project:** Foodbox
- **Developer:** Sunny Mishra



## Project and Developer Details

This project aims to design and develop an E-commerce website that lets people shop food items of different cuisines at affordable prices and deliver the products to their addresses. It was developed using Angular and Spring boot It was created as the Capstone Project for the Full Stack Java Developer course.

## Sprints and Tasks.

Product Backlog:

### Programming:

1. Create database and tables.
2. Add some rows and metadata to the tables
3. Initialize a Spring Boot project for the Back-End side.
4. Create REST APIs with spring Data JPA Repositories.
5. Create desired DAO methods for the Back-End side
6. Create a new Angular project for the Front-End side.
7. Create login and register pages and components.
8. Add cache to the login user
9. Logout user and remove cache
10. Show all products to the home page.
11. Show all products as cards.
12. Create a product details component.
13. Search a product by a category.
14. Search a product by a keyword.
15. Add products pages
16. Filter by page number
17. Sort product by different options
18. Add products to the cart.
19. Update total price in the cart status.
20. Show the payment gate and review the list
21. Add and remove products from the review list
22. Update the total price in the payment gate
23. Create the admin view
24. Update/Remove a product for the admin
25. Add a new product for the admin
26. Update the CSS design
27. Add bootstrap and font awesome to the components.
28. Debug and test the project.

### Deployment:

29. Upload project to GitHub.
30. Create a t3.medium instance for master.
31. Create a t3.micro instance for slave.

32. Connect the two instances to the system
33. Create a Pipeline project on Jenkins.
34. Create a Jenkins file.
35. Generate a SSH key for GitHub.
36. Build the pipeline project.
37. Deploy the project.

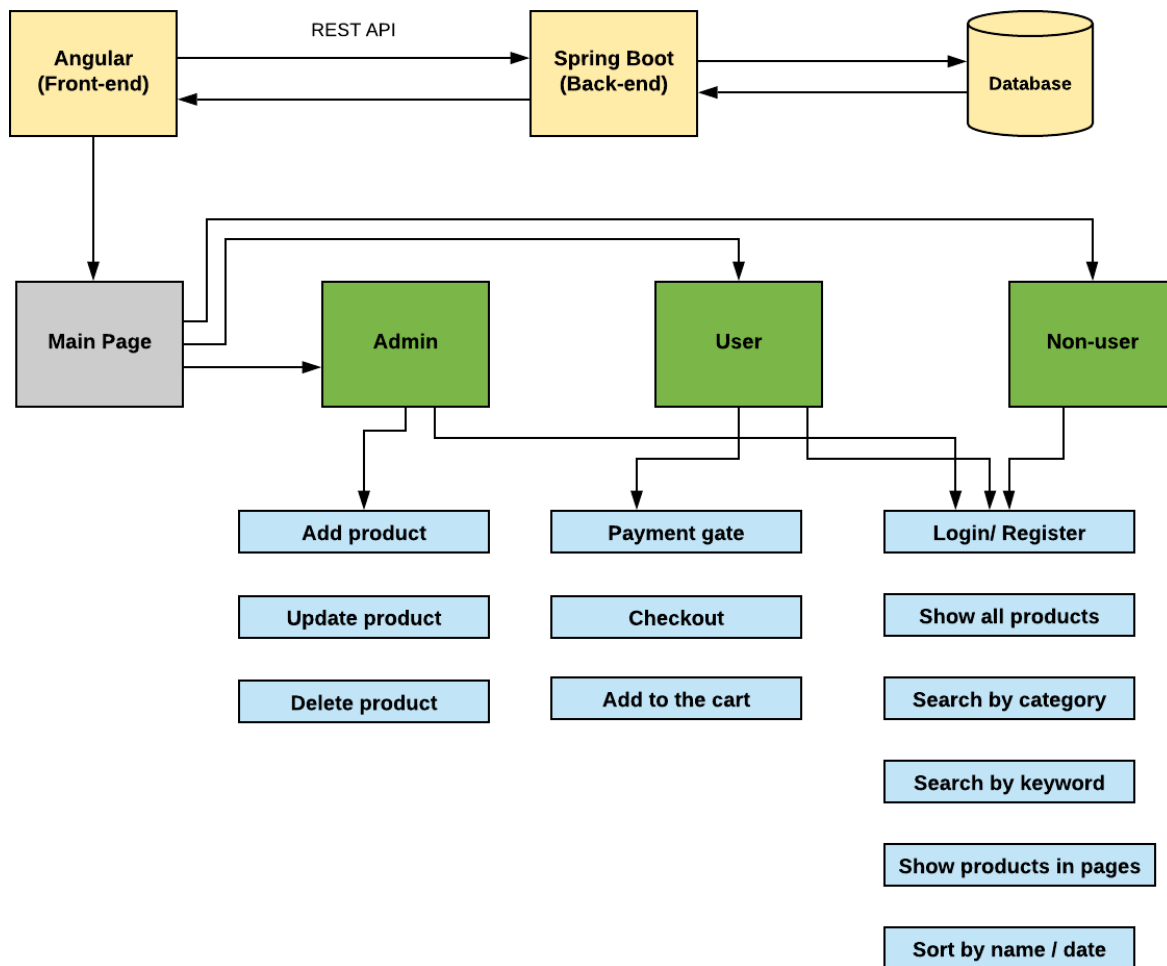
Sprints	Tasks	Days
Sprint 1	<ul style="list-style-type: none"> <li>○ Create database and tables.</li> <li>○ Add some rows and metadata to the tables</li> <li>○ Initialize a Spring Boot project for the Back-End side.</li> <li>○ Create REST APIs with spring Data JPA Repositories.</li> <li>○ Create desired DAO methods for the Back-End side</li> </ul>	1 day
Sprint 2	<ul style="list-style-type: none"> <li>○ Create a new Angular project for the Front-End side.</li> <li>○ Create login and register pages and components.</li> <li>○ Add cache to the login user</li> <li>○ Logout user and remove cache</li> </ul>	1 day
Sprint 3	<ul style="list-style-type: none"> <li>○ Show all products to the home page.</li> <li>○ Show all products as cards.</li> <li>○ Create a product details component.</li> <li>○ Search a product by a category.</li> <li>○ Search a product by a keyword.</li> </ul>	1 day
Sprint 4	<ul style="list-style-type: none"> <li>○ Add products pages</li> <li>○ Filter by page number</li> <li>○ Sort product by different options</li> <li>○ Add products to the cart.</li> <li>○ Update total price in the cart status.</li> </ul>	1 day
Sprint 5	<ul style="list-style-type: none"> <li>○ Show the payment gate and review the list</li> <li>○ Add and remove products from the review list</li> <li>○ Update the total price in the payment gate</li> </ul>	1 day
Sprint 6	<ul style="list-style-type: none"> <li>○ Create the admin view</li> <li>○ Update/Remove a product for the admin</li> <li>○ Add a new product for the admin</li> </ul>	1 day
Sprint 7	<ul style="list-style-type: none"> <li>○ Update the CSS design</li> <li>○ Add bootstrap and font awesome to the components.</li> <li>○ Debug and test the project.</li> </ul>	1 day
Sprint 8	<ul style="list-style-type: none"> <li>○ Upload project to GitHub.</li> <li>○ Create a t3.medium instance for master.</li> <li>○ Create a t3.micro instance for slave.</li> <li>○ Connect the two instances to the system</li> <li>○ Create a Pipeline project on Jenkins.</li> </ul>	1 day

	<ul style="list-style-type: none"> <li>○ Create a Jenkins file.</li> <li>○ Generate a SSH key for GitHub.</li> </ul>	
Sprint 9	<ul style="list-style-type: none"> <li>○ Build the pipeline project.</li> <li>○ Debug problems</li> <li>○ Deploy the project.</li> </ul>	1 day

## Technologies and tools Used

- **Angular:** used in the front-end side to build modern single-page applications
- **Spring Boot:** used in the back-end side to create the REST API and retrieve data from a database.
- **AWS EC2 instance:** to use the instances as a VM and deploy the application
- **AWS RDS:** to upload the database online.
- **Jenkins:** to build the project from GitHub.
- **GitHub:** to upload the source code of the project.
- **MobaXterm:** to the instance from Windows OS.
- **Selenium:** for automation and testing.
- **Apache:** to use it as a web server.
- **HTML/CSS:** to create and format the content of the pages.
- **Bootstrap:** to use some CSS and JavaScript designs.
- **Maven:** to manage the project.
- **Visual Studio Code:** to write and run the Angular code.
- **IntelliJ:** to write and run the Spring Boot code.
- **MySQL:** to use it as database management system.
- **phpMyAdmin:** to administrate and manage the database manually.

## Flowcharts of The Application

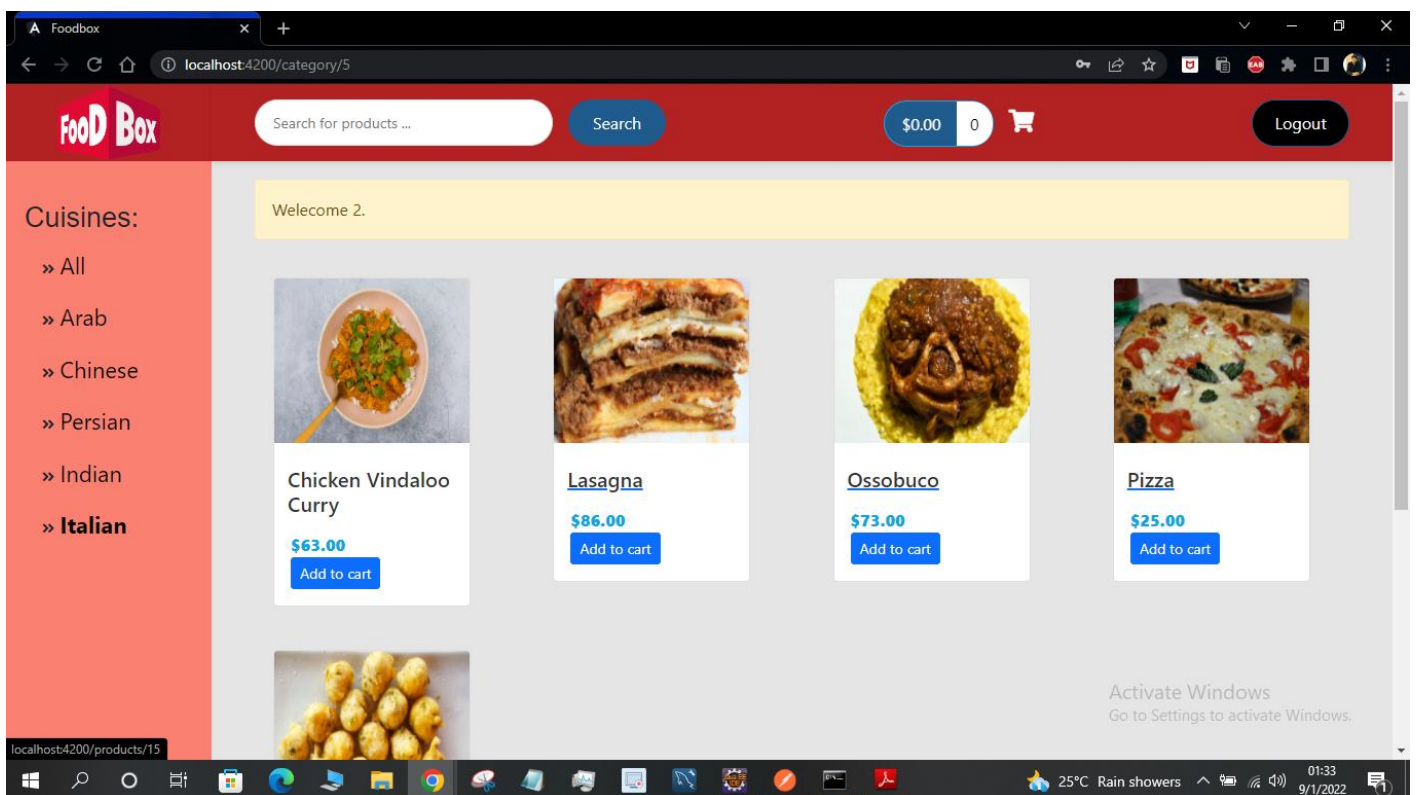
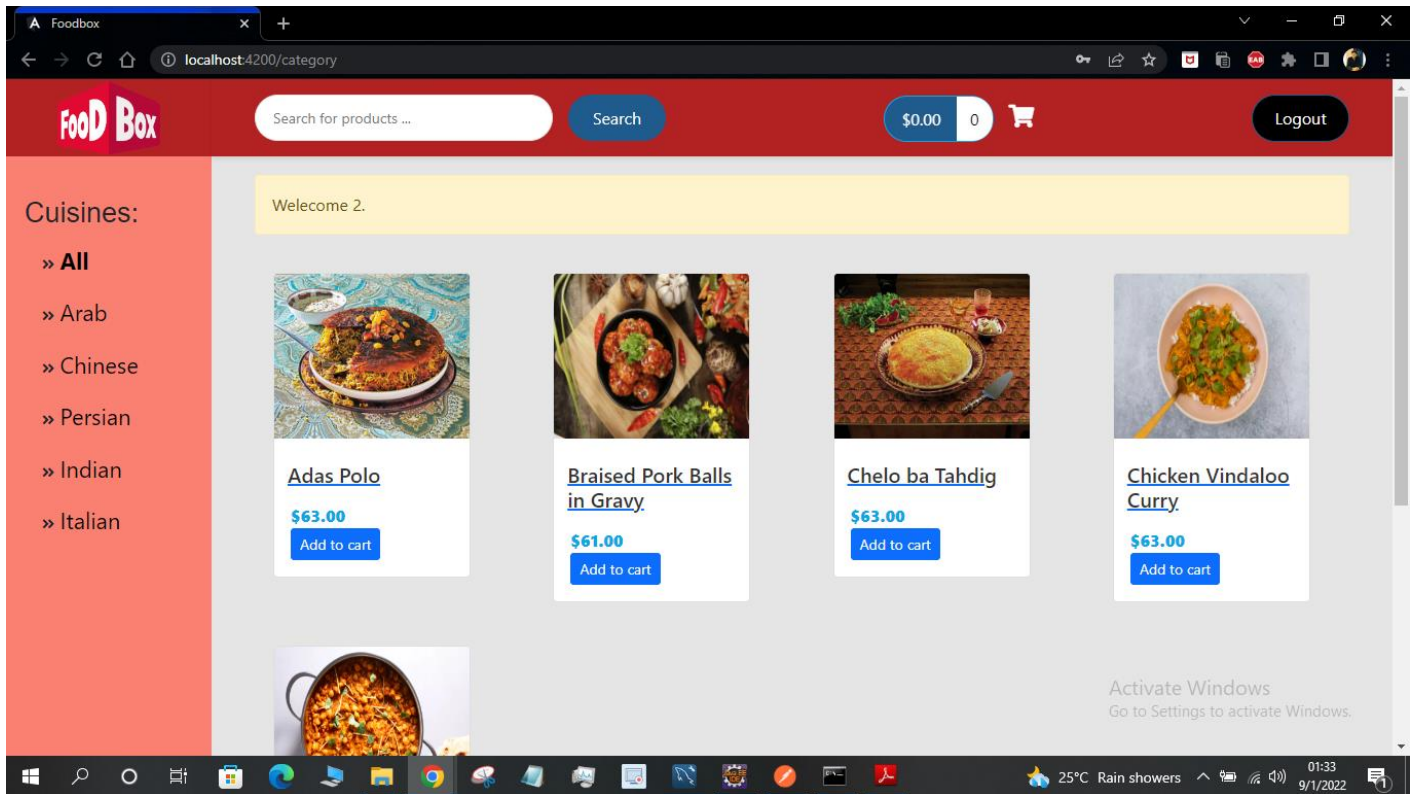


## Core concepts used in the project.

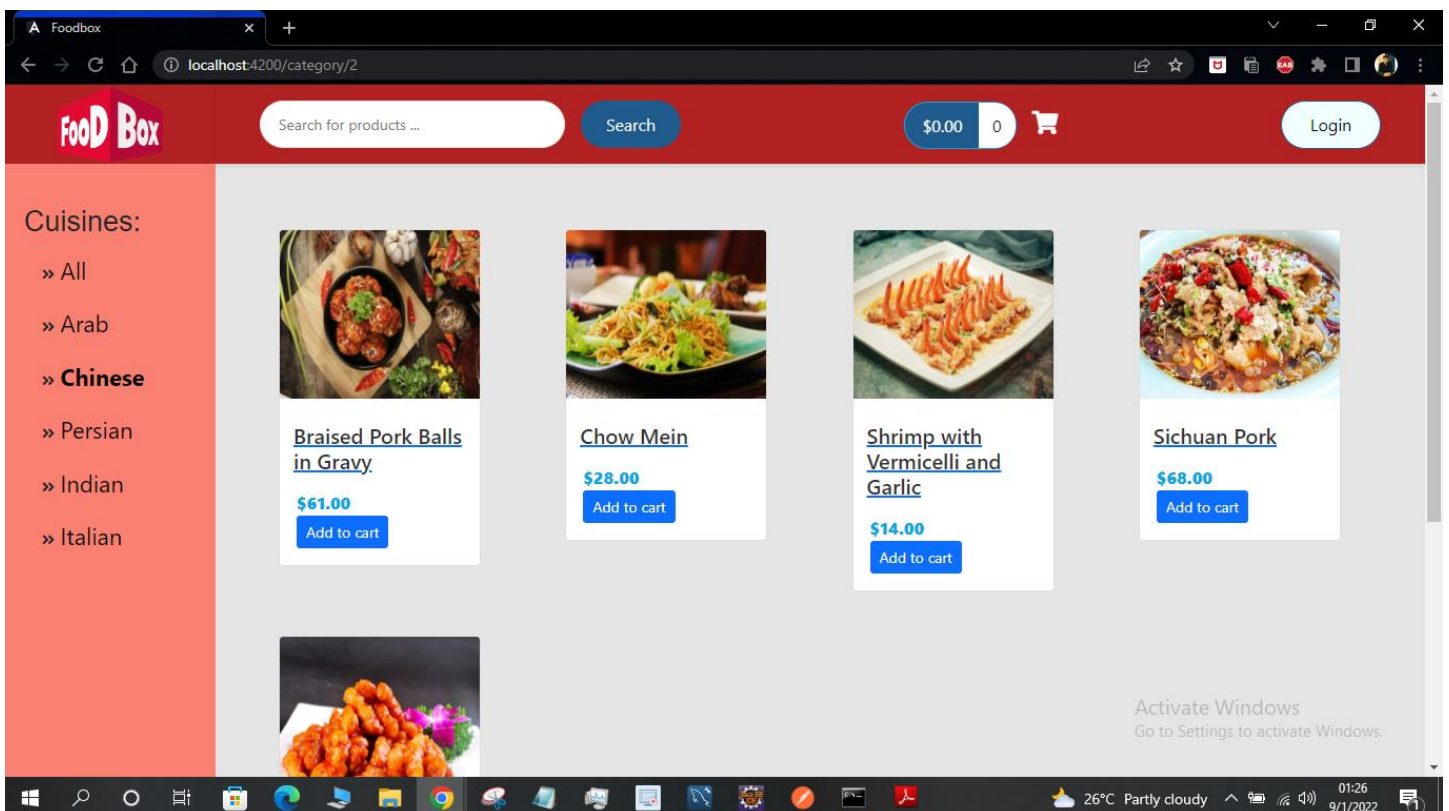
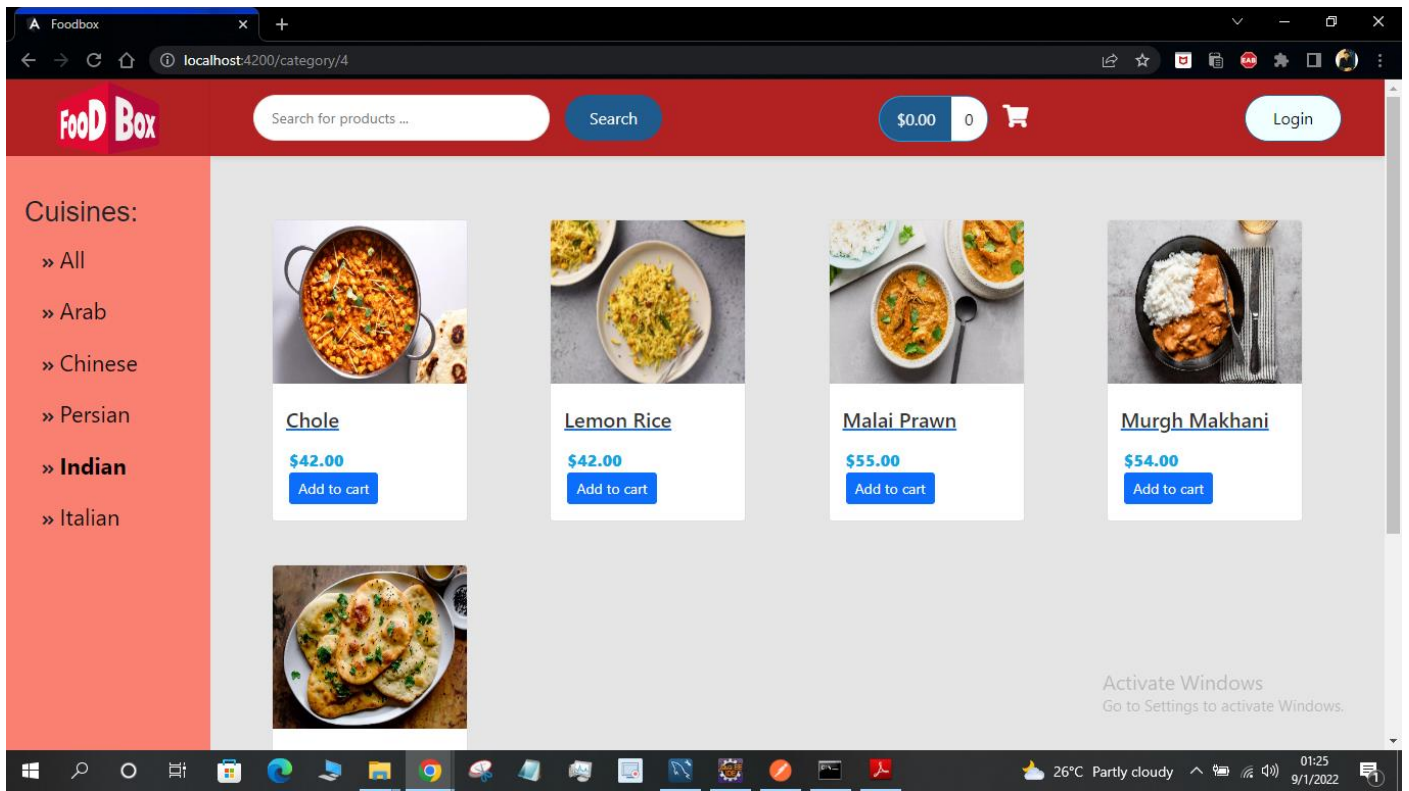
- **Object-Oriented:** used to create and model objects for users and their credentials.
- **REST API:** used to communicate between the back-end and the front-end sides.
- **Data Access Object:** to abstract and encapsulate all access to the data source.
- **Object-Relational Mapping:** to map the objects to the database.
- **Databases:** used to store and retrieve data.
- **Data Sources:** used to define a set of properties required to identify and access the database.
- **Collections:** used some collections such as **ArrayList** to store collection of data.
- **Deployment:** to deploy the local project to the end-users.
- **Virtual Machine:** use virtual instances to help to build, deploy and manage websites.
- **Exception Handling:** used to catch problems that arises in the code especially in I/O blocks.
- **Single Web Page:** apply the concept of a website that only contains one HTML page.

## Screenshots

- General look of the store:

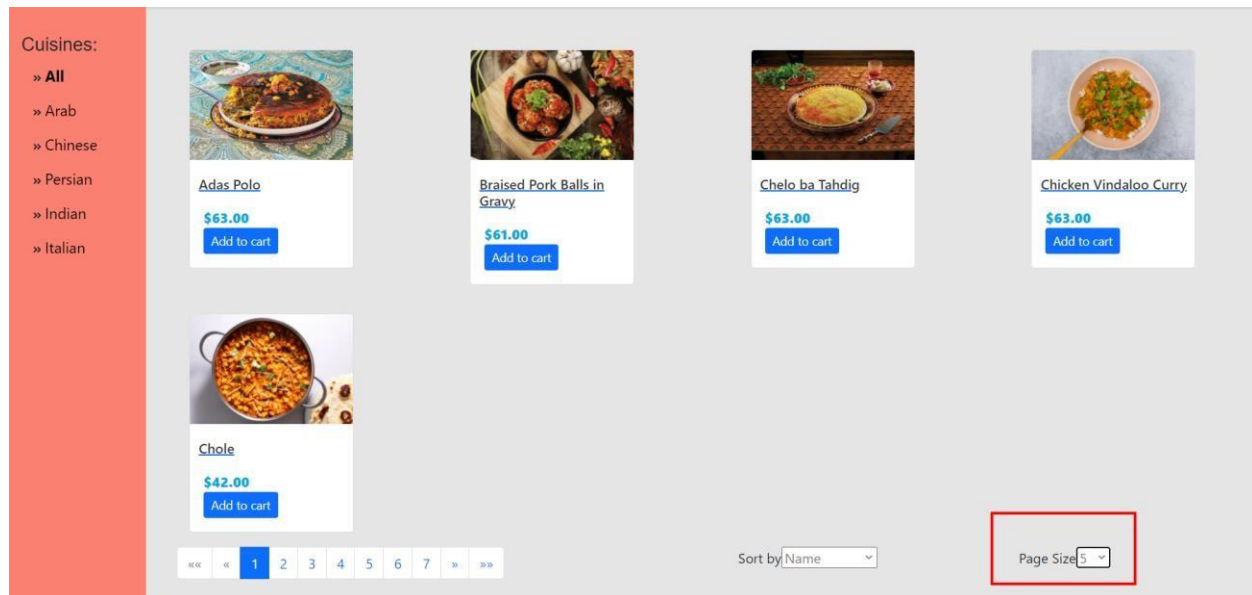


- Filter the product according to the cuisine

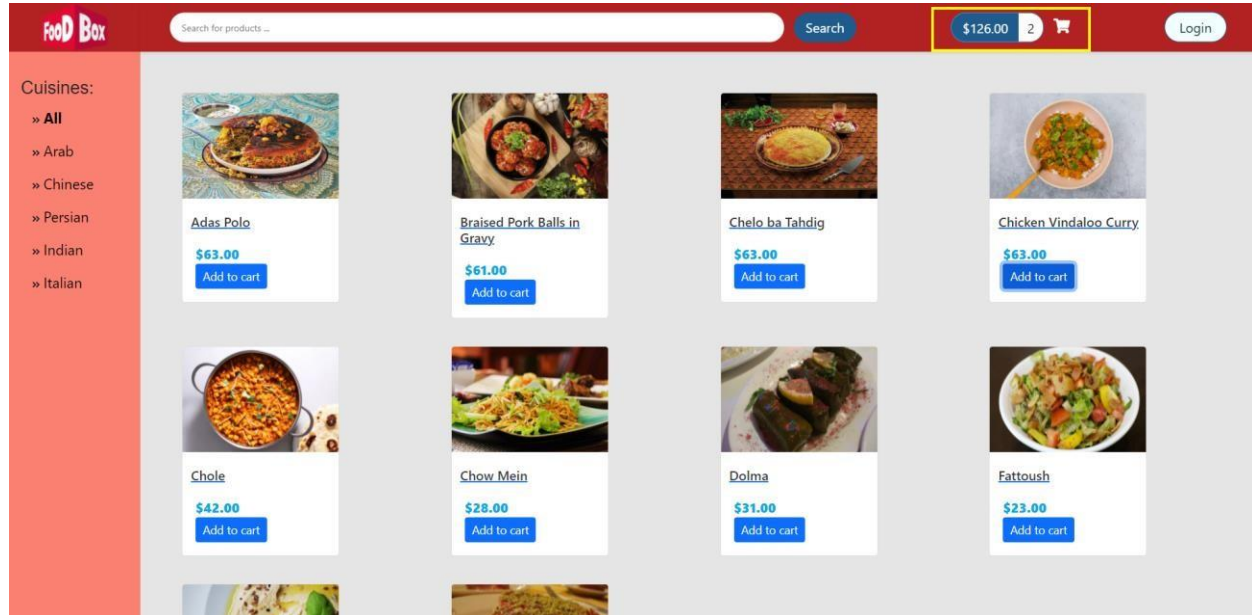




- Choosing the number of each displayed products in each page

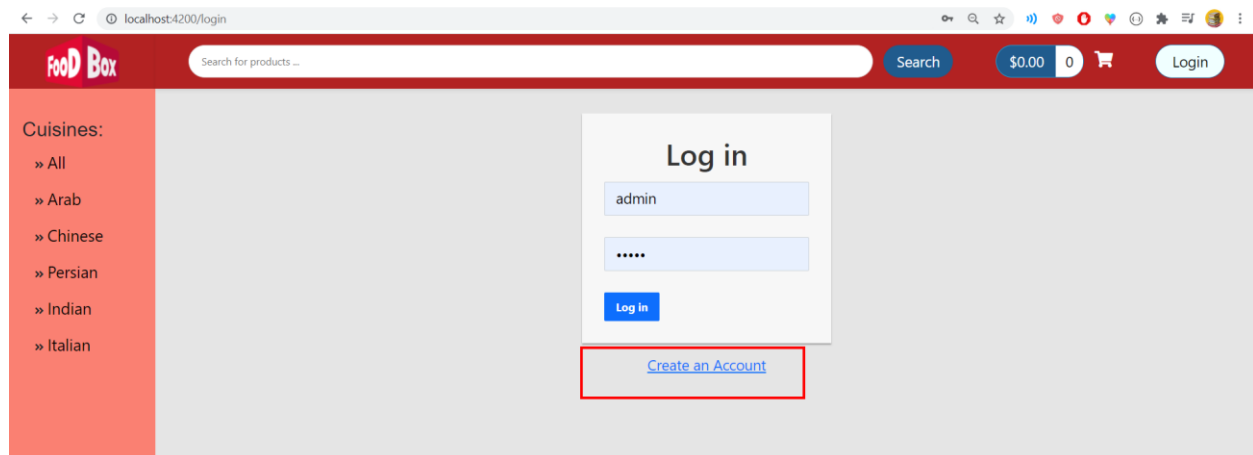
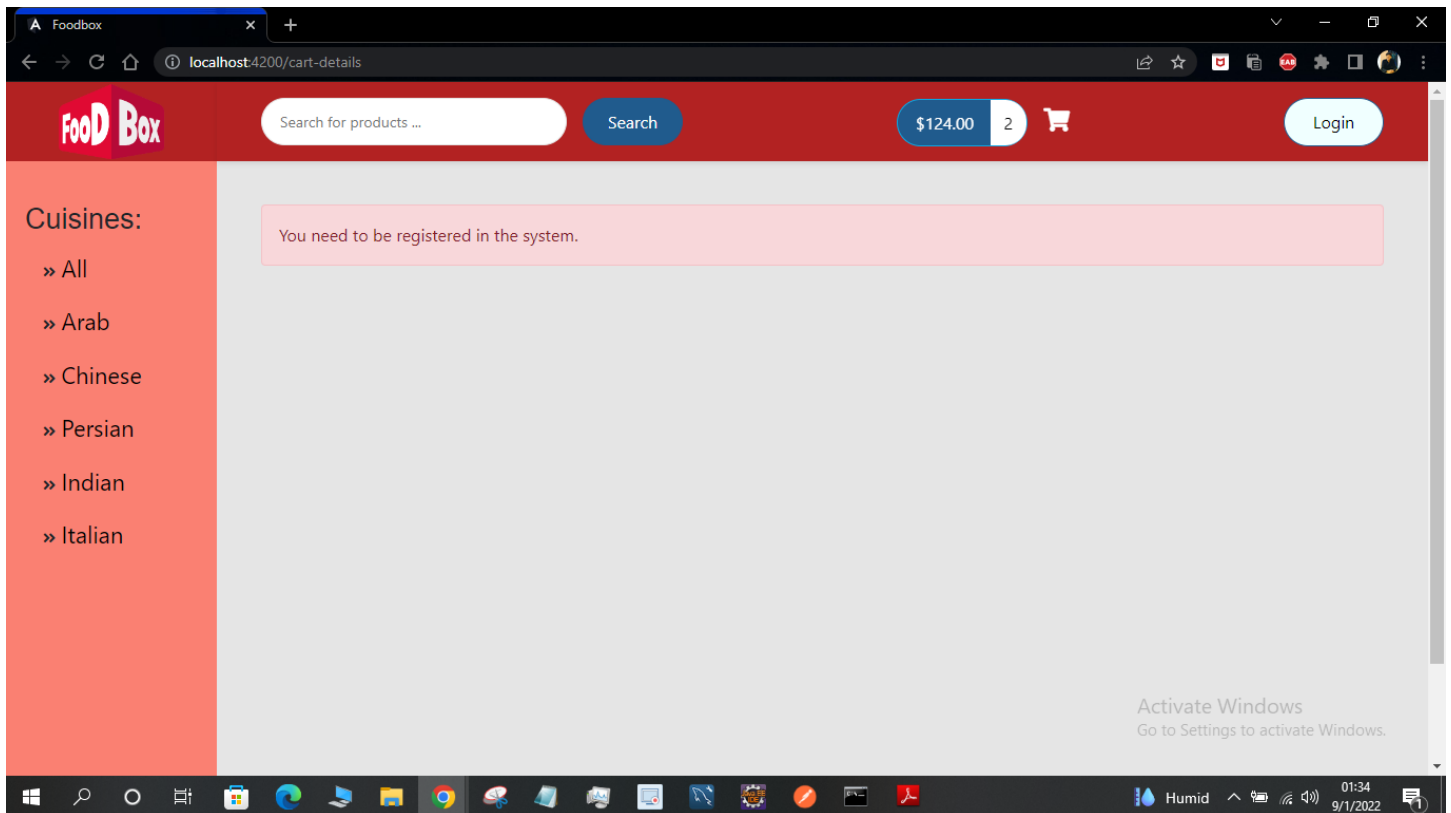


- You can add products to the cart





- However, you need to be registered to buy



- Register form

localhost:4200/register

Food Box

Search for products ... Search

\$0.00 0 Login

Cuisines:

- » All
- » Arab
- » Chinese
- » Persian
- » Indian
- » Italian

### Register

Username

Email

Password

phone

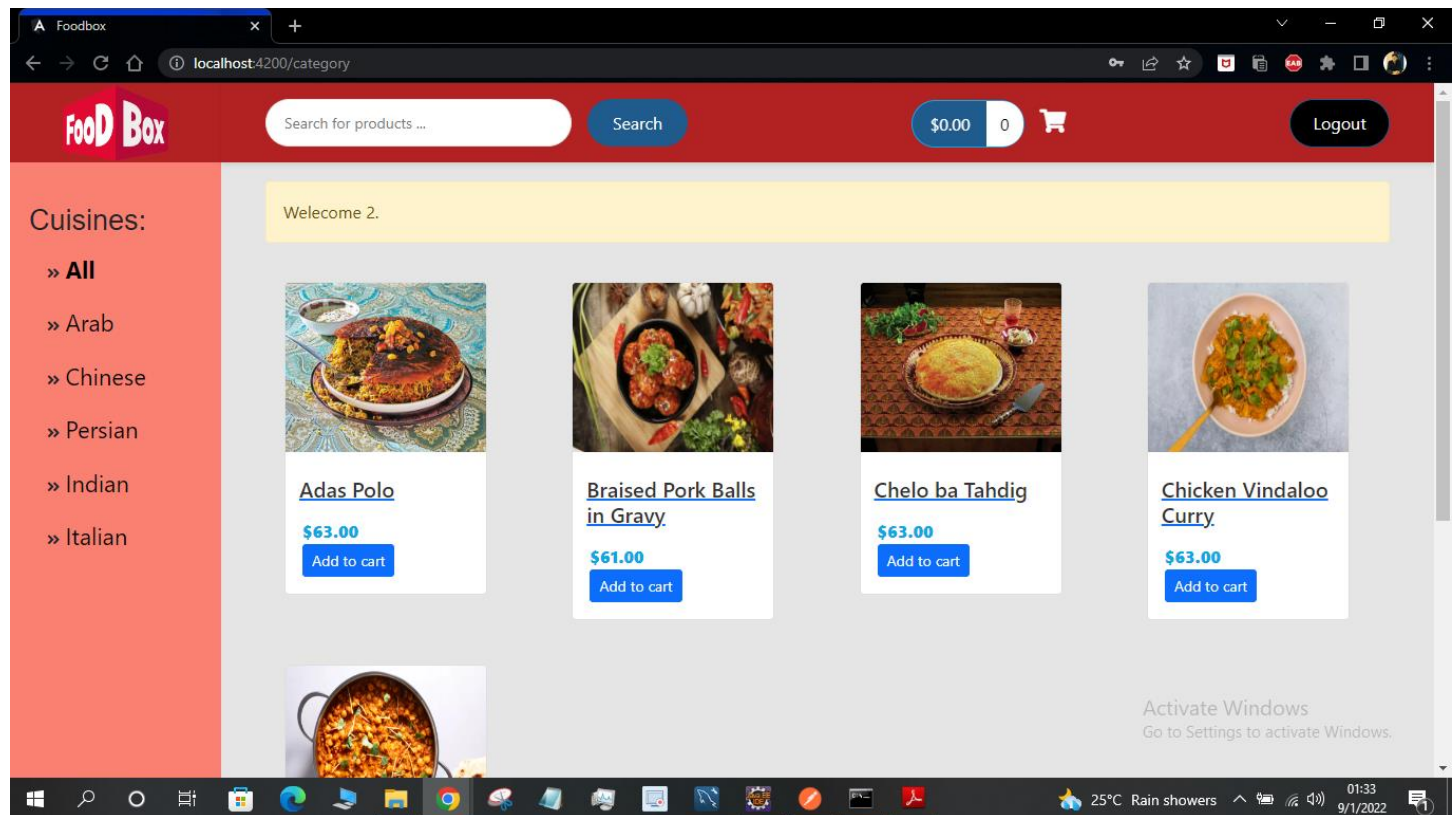
Address

city

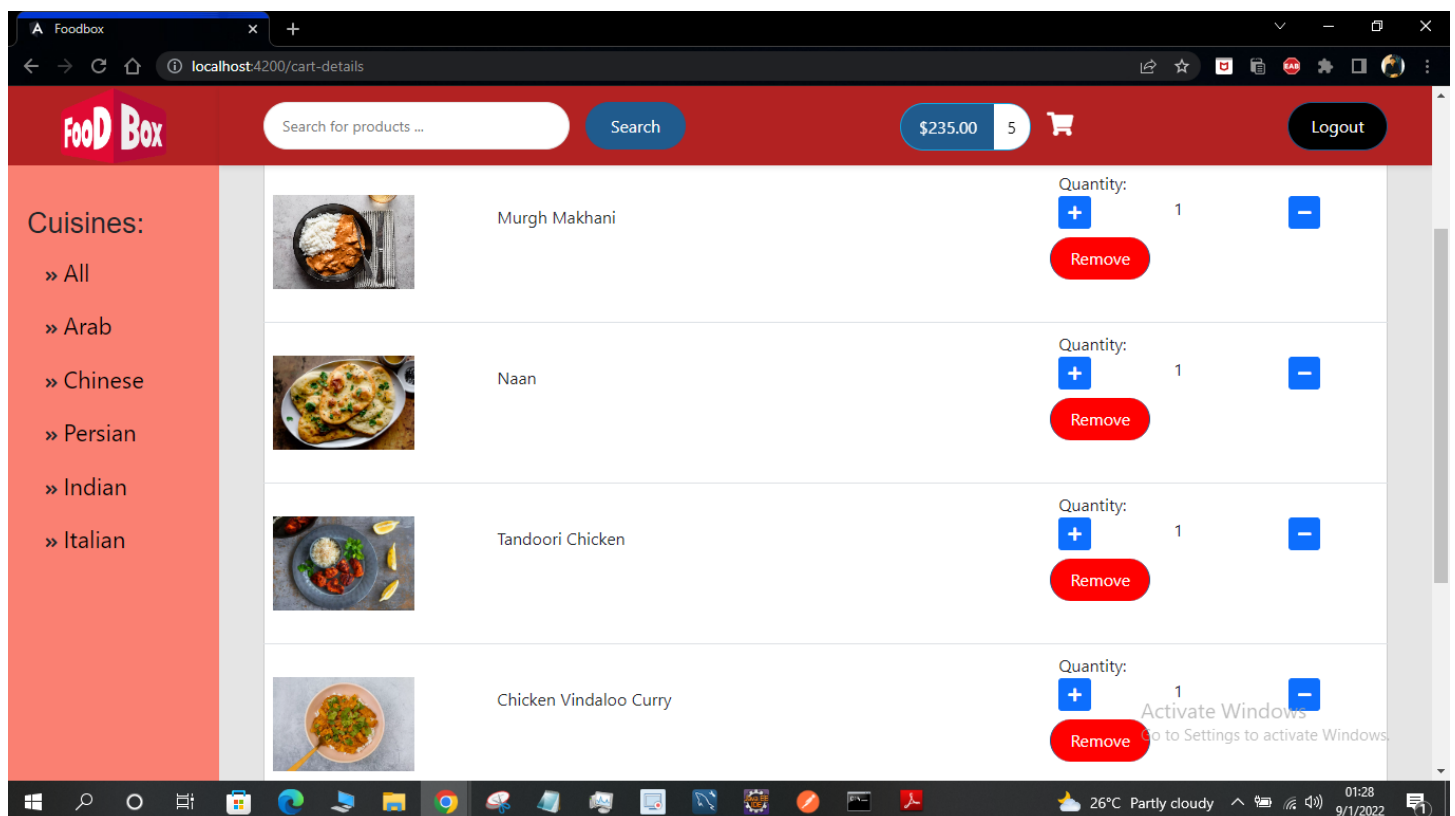
street

zipCode

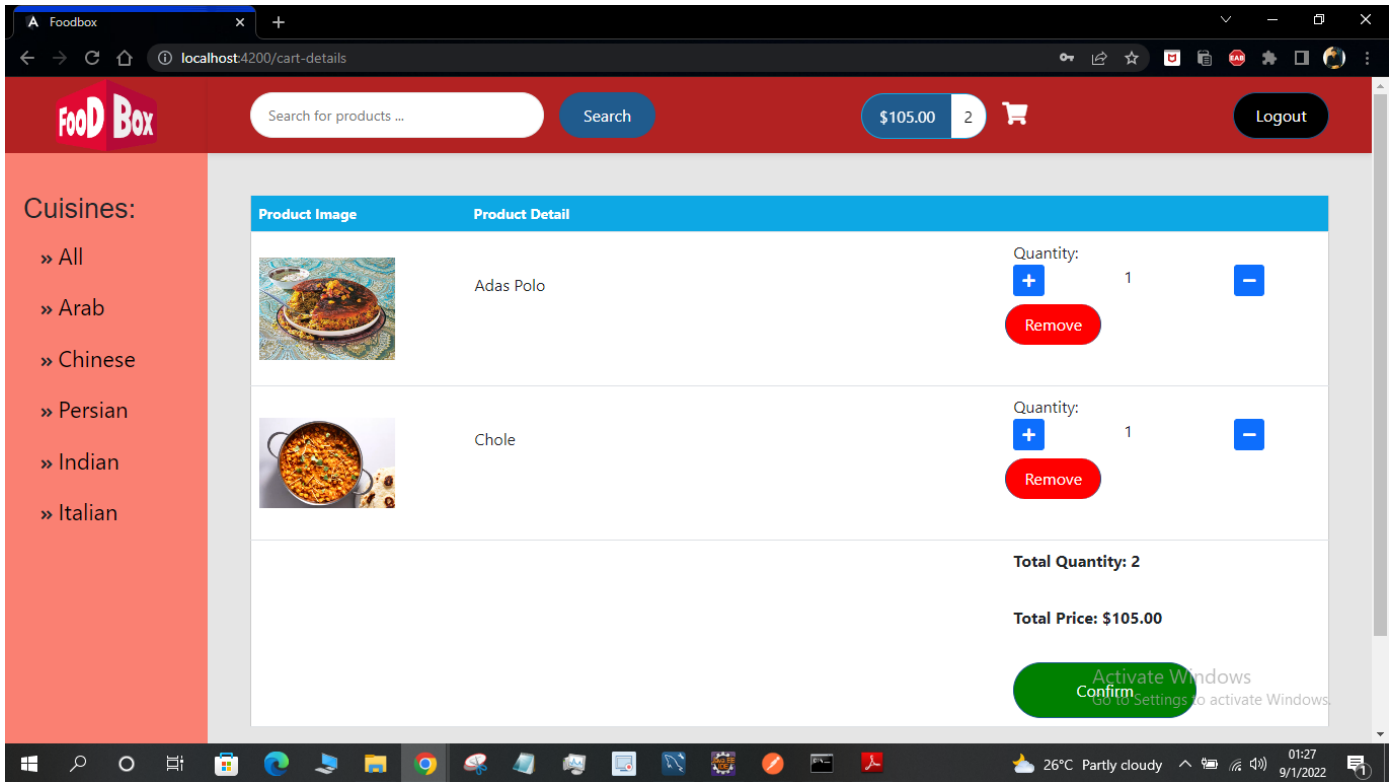
Register



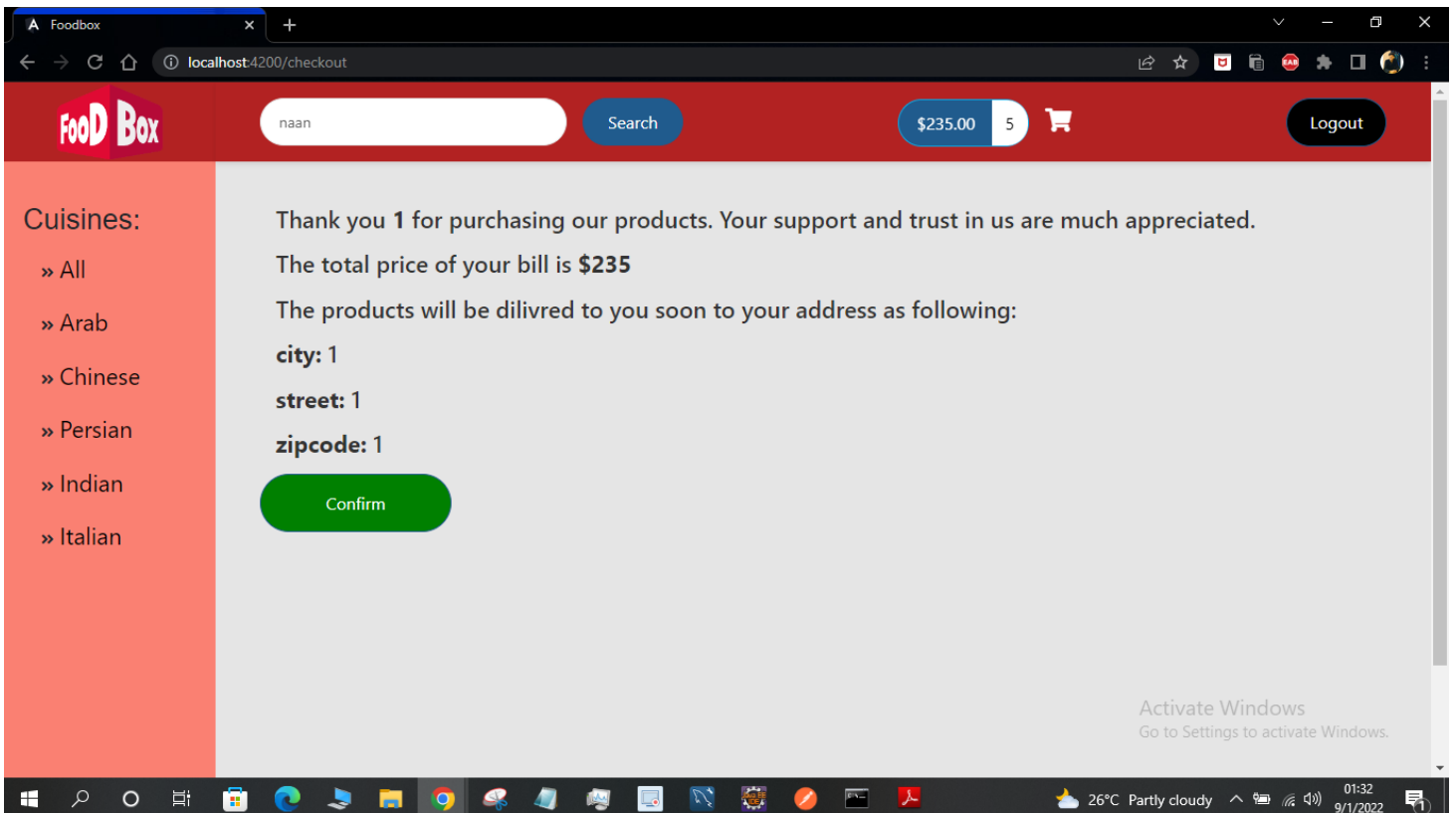
- User can now buy (Payment gateway)



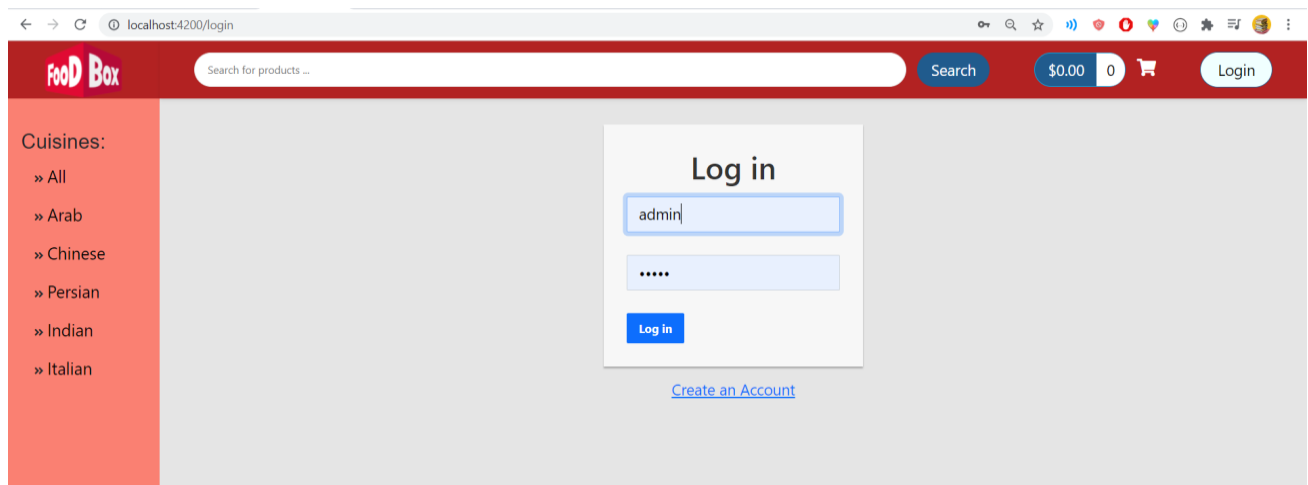
- Can add/remove products during review



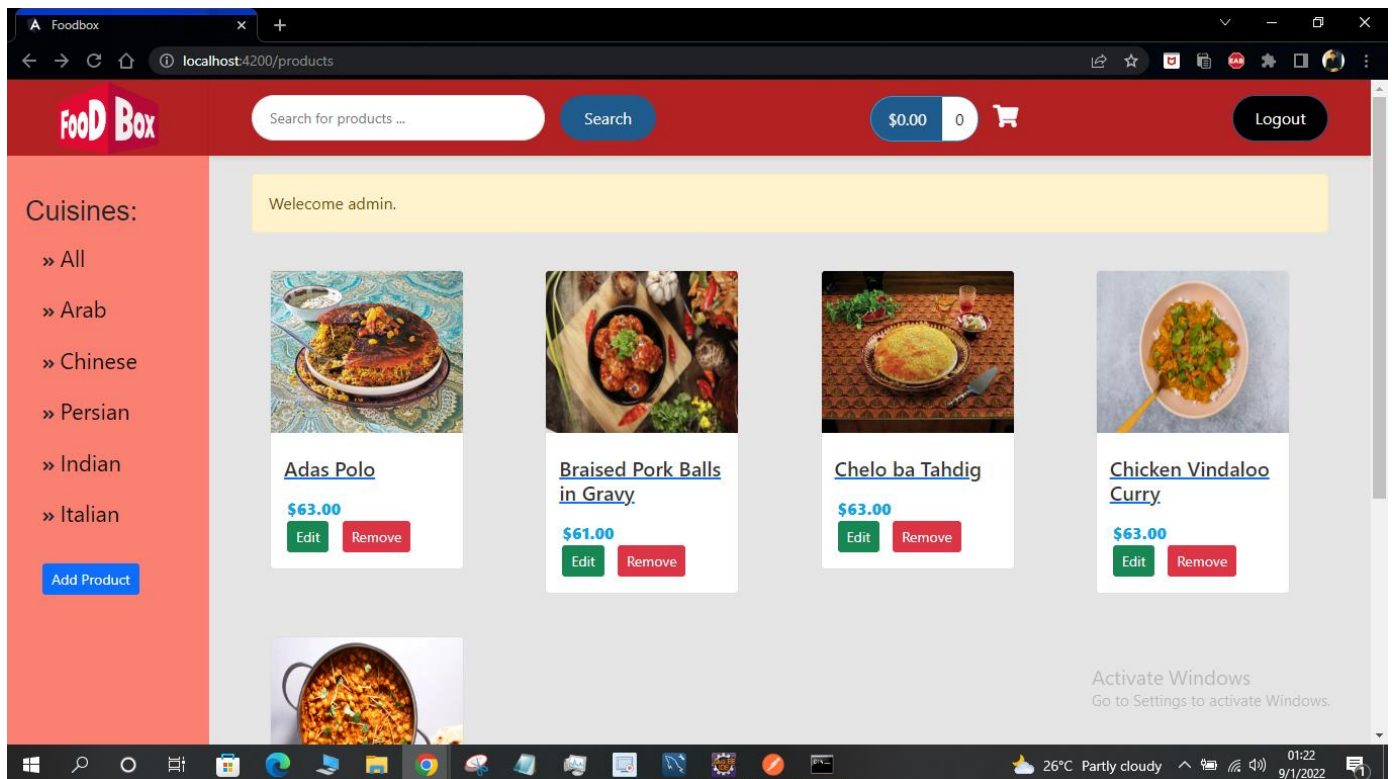
- Check-out and confirm



- Now, let's log in as admin
- Admin/admin



- Admin can:
  - Create product
  - Edit product
  - Delete product



Thank You