

Full Stack Java Developer

Phase-6: Capstone Project

- **Project:** Foodbox
- **Developer:** Ankit Yadav

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.

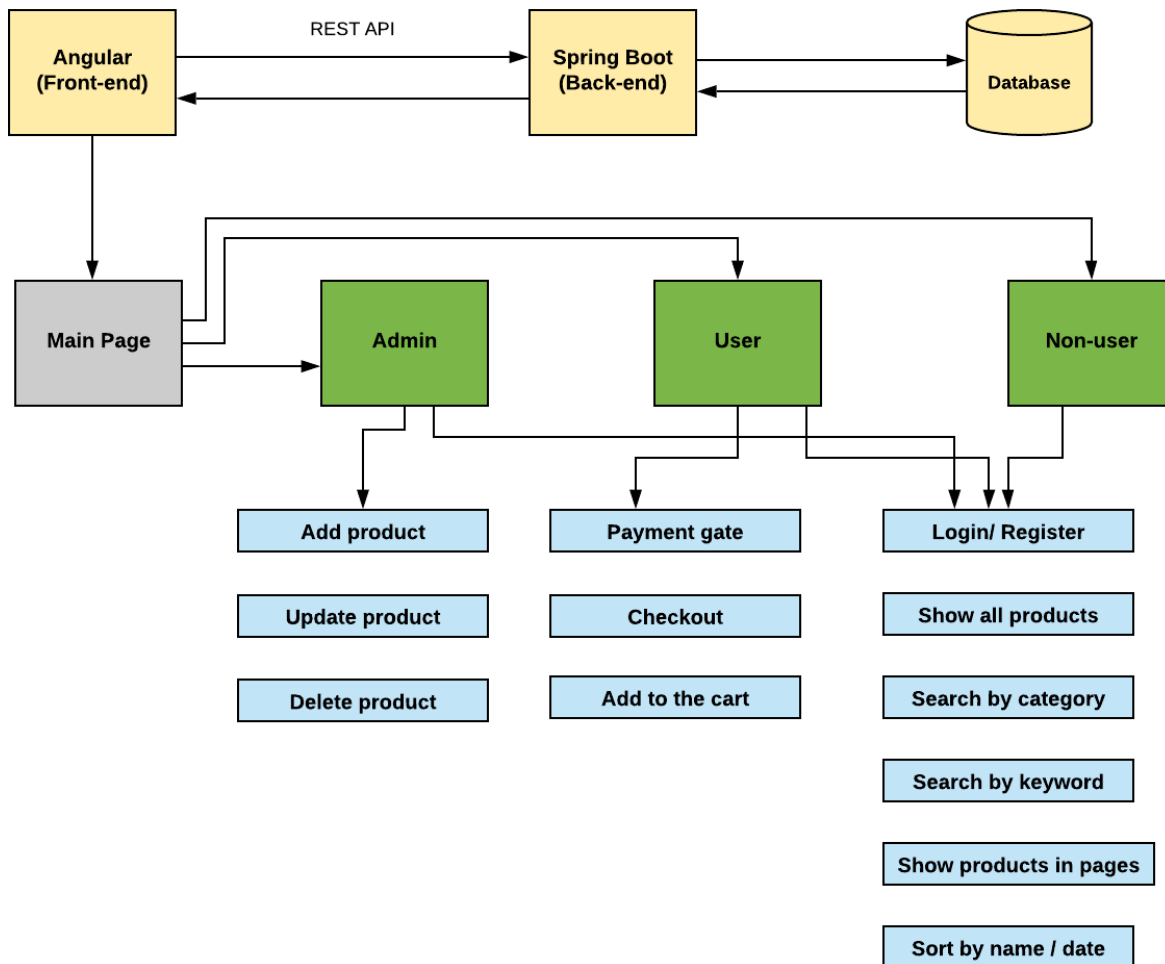
Sprints	Tasks	Days
Sprint 1	<ul style="list-style-type: none"> ○ Create database and tables. ○ Add some rows and metadata to the tables ○ Initialize a Spring Boot project for the Back-End side. ○ Create REST APIs with spring Data JPA Repositories. ○ Create desired DAO methods for the Back-End side 	1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

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

GitHub Repository:-

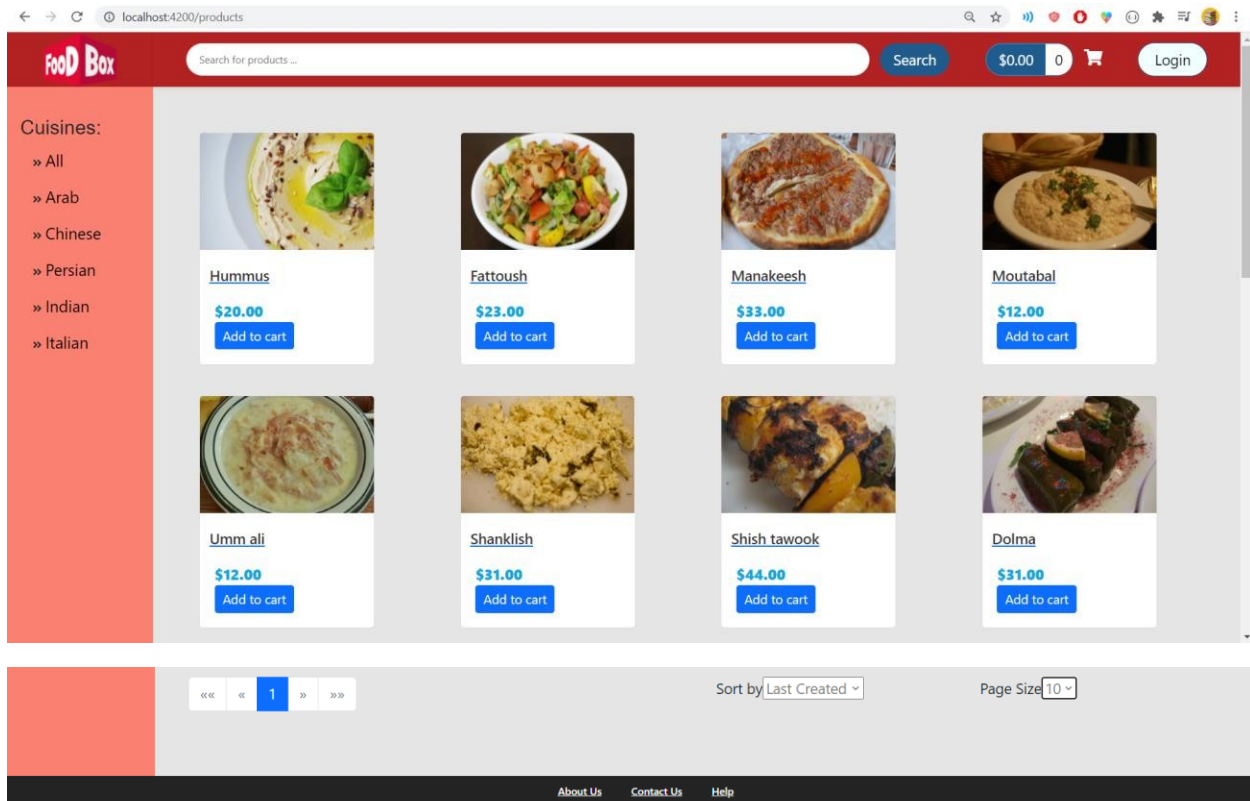
<https://github.com/Dexter2022/Foodbox>

How to run the program locally

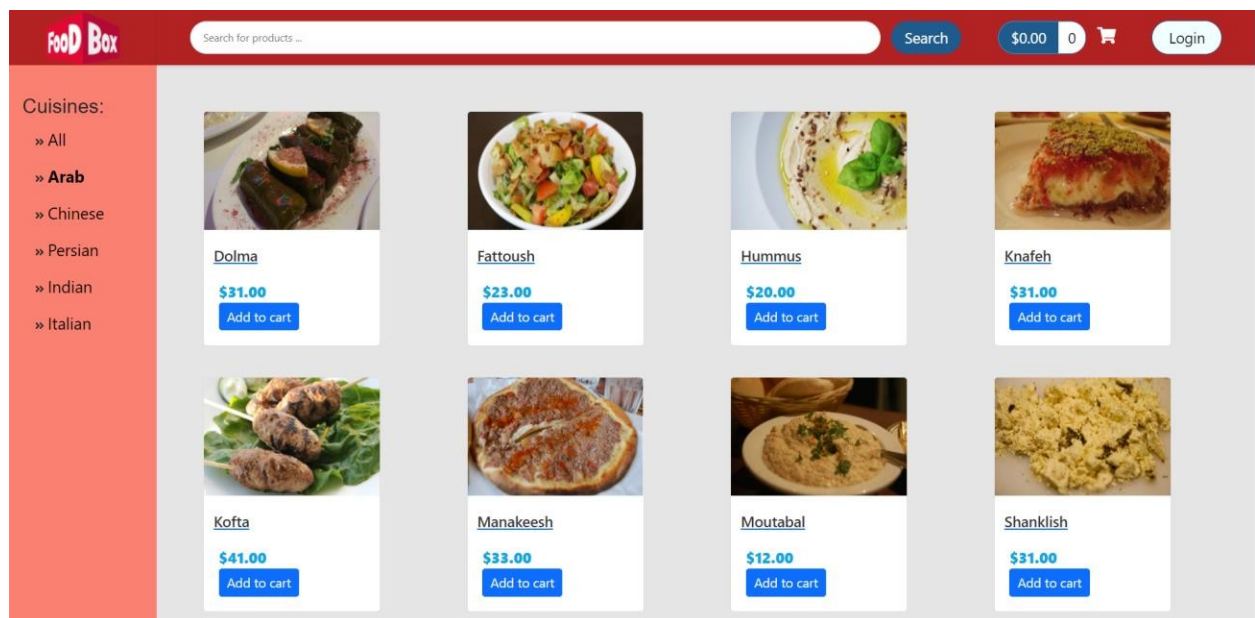
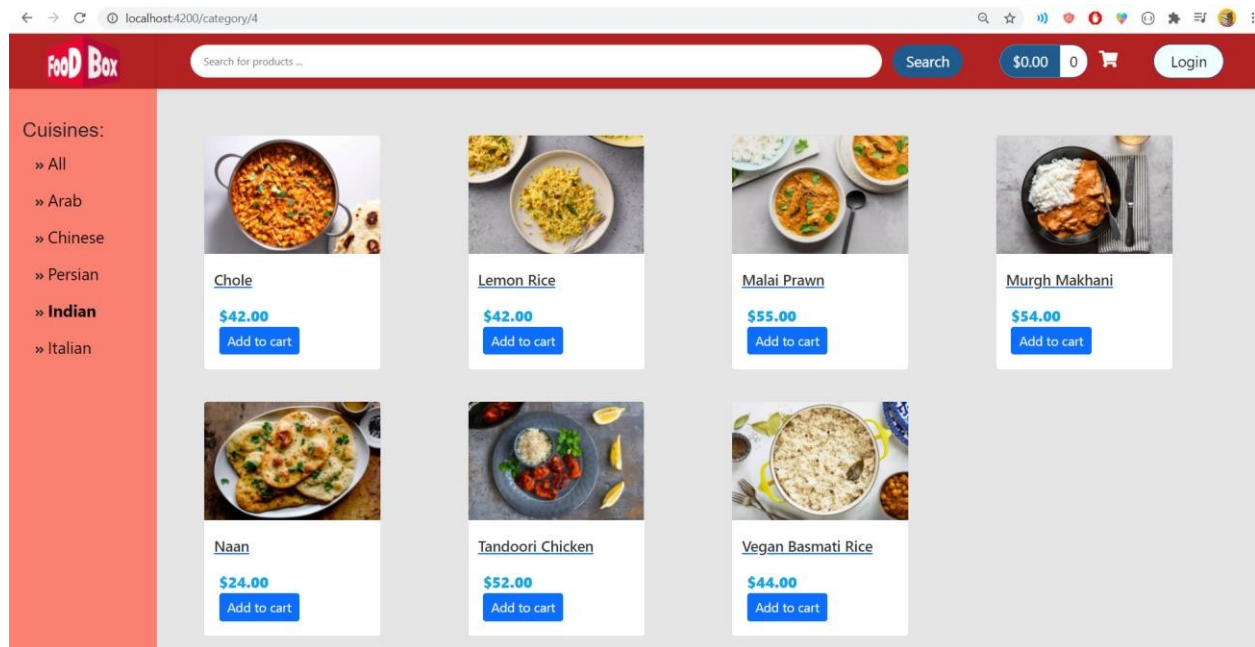
- clone project
- Import the "`\Back-End\foodbox\database\foodbox.sql`" file to your database administration tool.
- Go to "`\Back-End\foodbox\src\main\resources\application.properties`" file, open it.
- Edit some values of the database' properties to be suit to your database administration tool.
- Run the back-end project as a maven project:
 - *cd to your project "Back-end\foodbox"*
 - *mvn compile*
 - *mvn exec:java -Dexec.mainClass=com.simplilearn.foodbox*
- Open another command line for the front-end part.
- cd to your project "Front-end-end\foodbox"
- install the following:
 - *npm install --save-dev*
 - *npm install @angular/localize --save*
 - *npm install bootstrap --save*
 - *npm install font-awesome --save*
- Run using *ng serve --open*
- It would be displayed in <http://localhost:4200/>

Screenshots

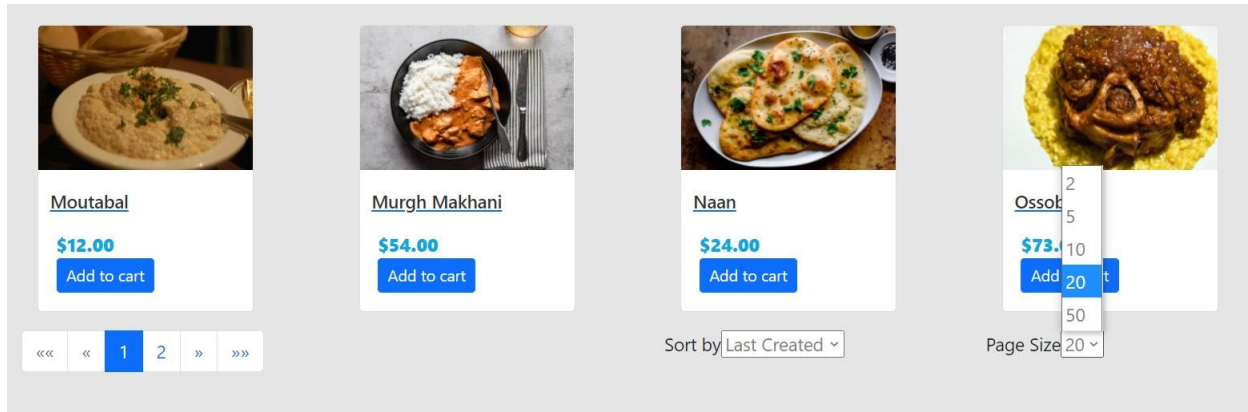
- General look of the store:



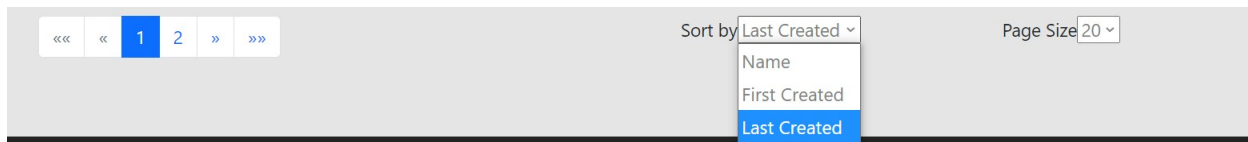
- Filter the product according to the cuisine



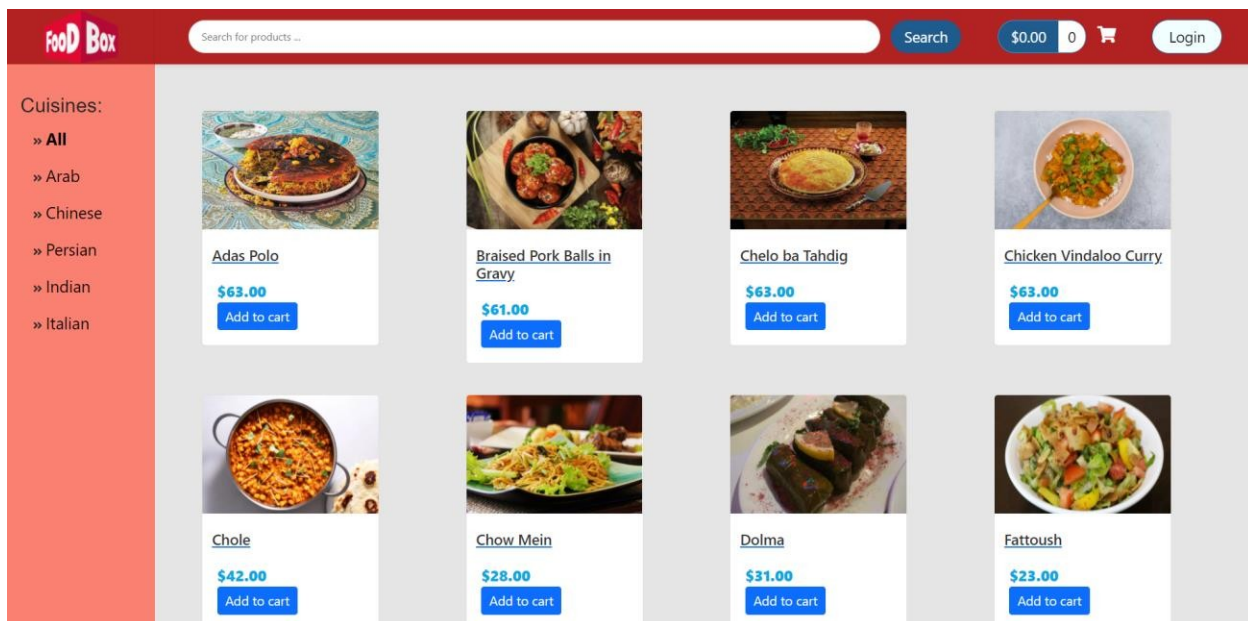
- Filtering by page and sorting are available



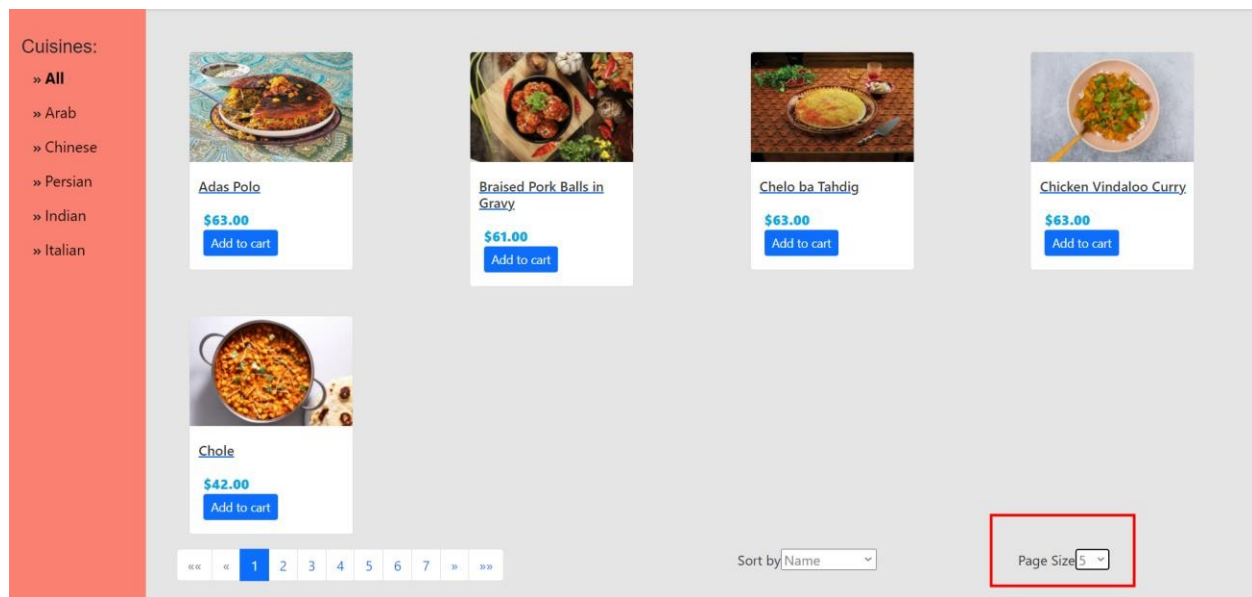
- Sorting can be in one of these three options



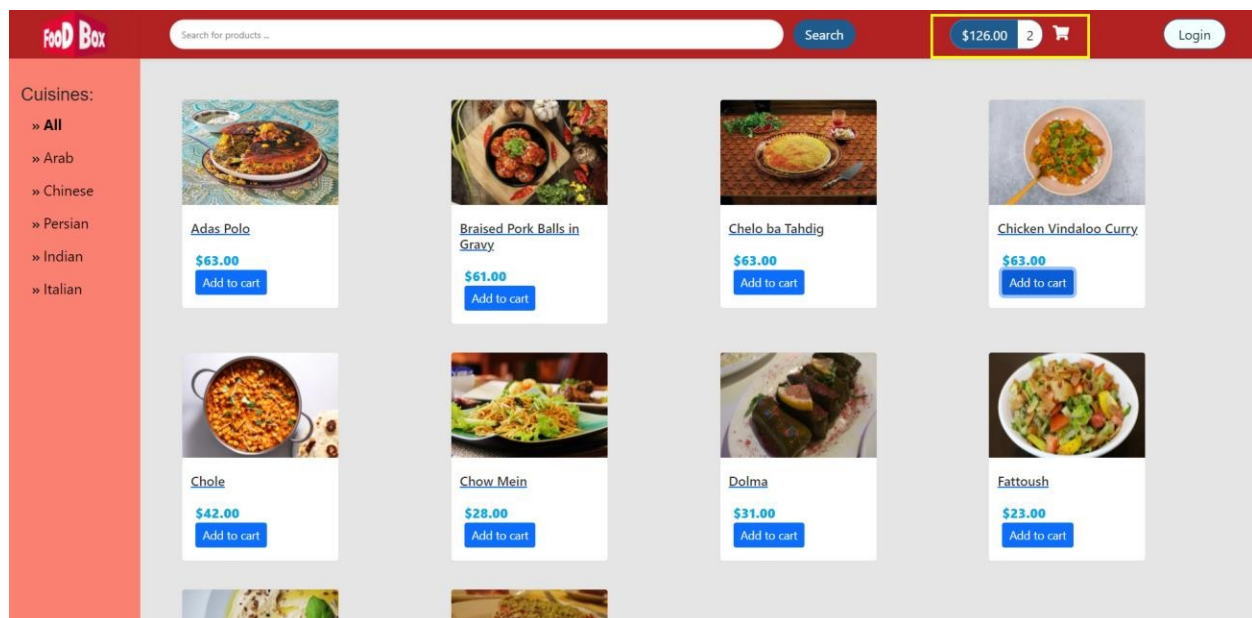
- Sorting by name



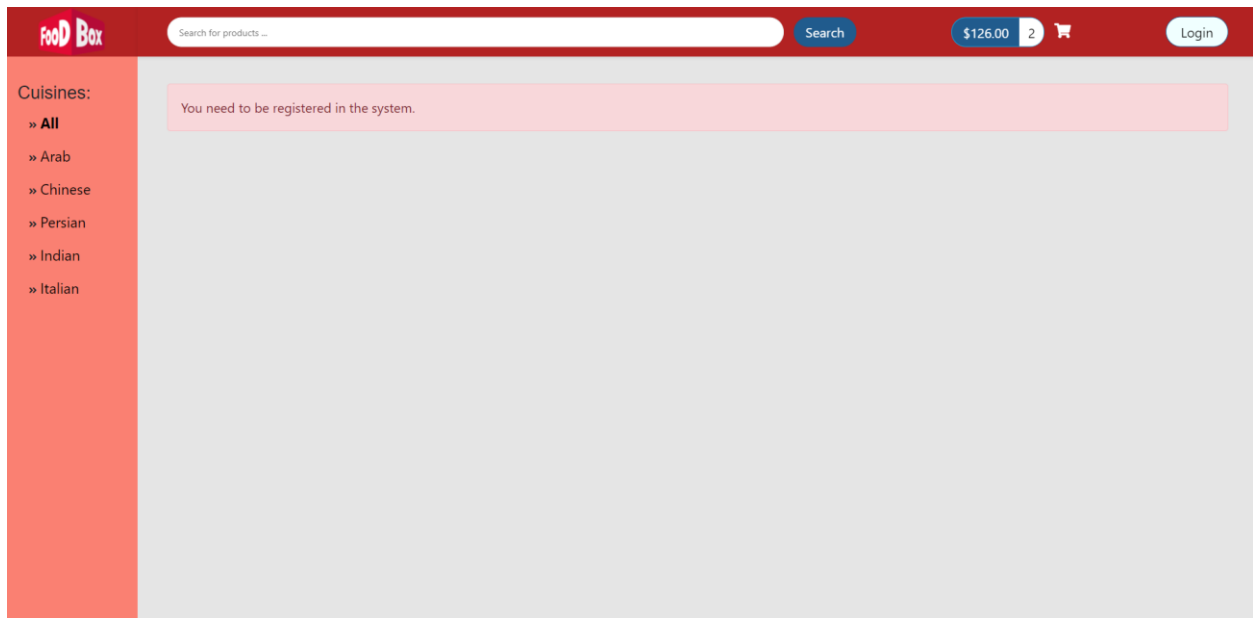
- Choosing the number of each displayed products in each page



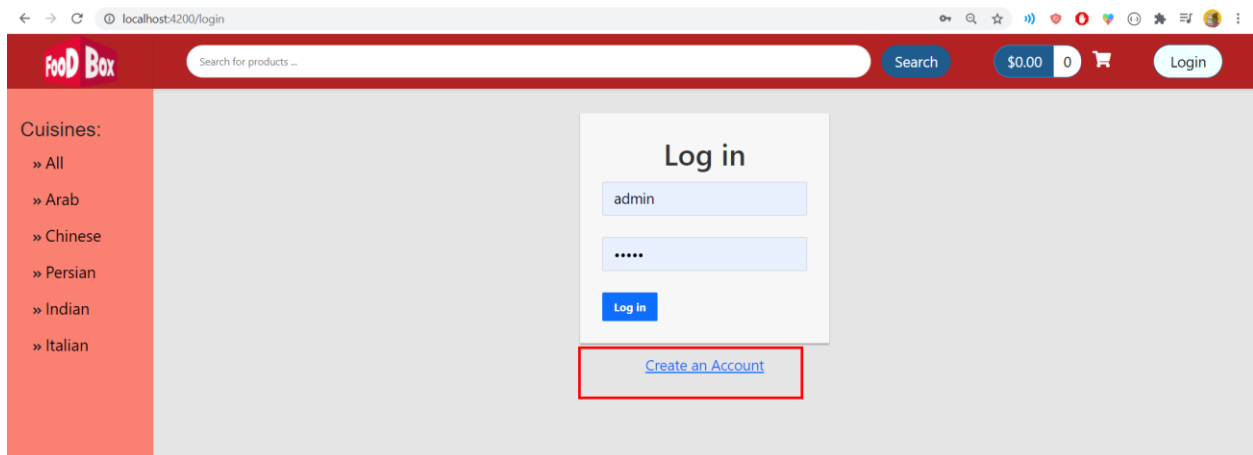
- You can add products to the cart



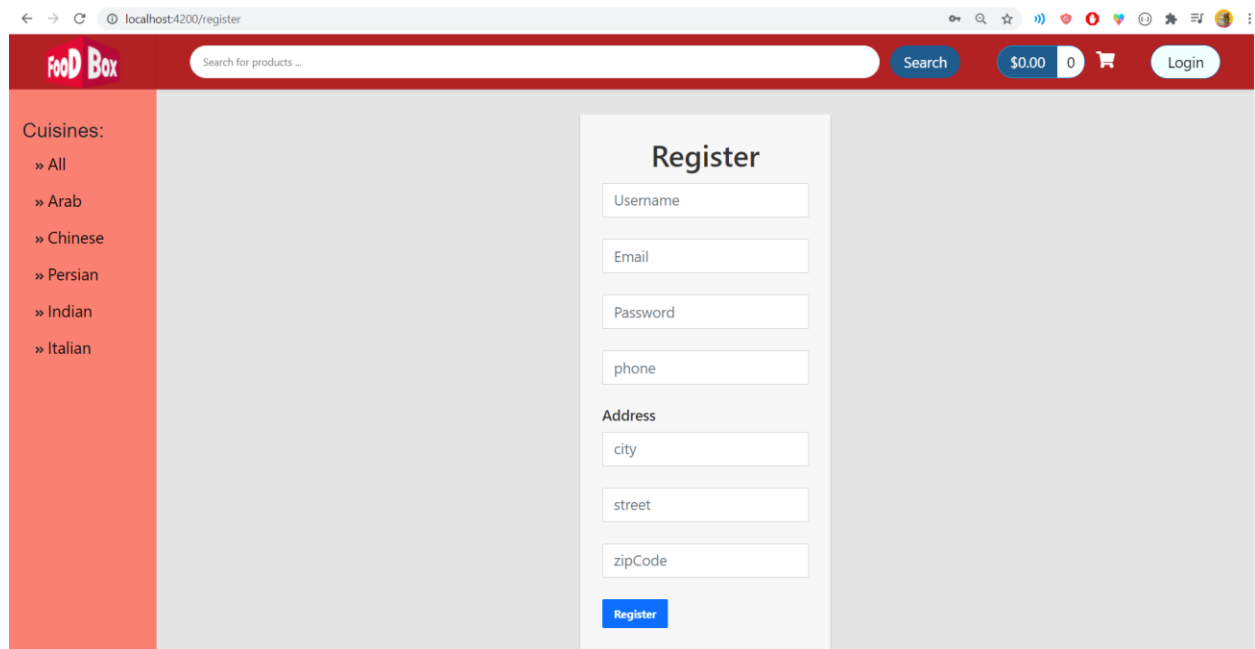
- However, you need to be registered to buy



- Login and then create account



- Register form



A screenshot of a web browser showing the 'Food Box' register form. The browser address bar shows 'localhost:4200/register'. The page has a red header with the 'Food Box' logo, a search bar, and a shopping cart icon showing '\$0.00' and '0' items. A left sidebar lists cuisines: All, Arab, Chinese, Persian, Indian, and Italian. The main content area is titled 'Register' and contains several input fields: Username, Email, Password, phone, and an 'Address' section with fields for city, street, and zipCode. A blue 'Register' button is at the bottom of the form.

Food Box

Search for products ...

Search

\$0.00 0

Login

Cuisines:

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

Register

Username

Email

Password

phone

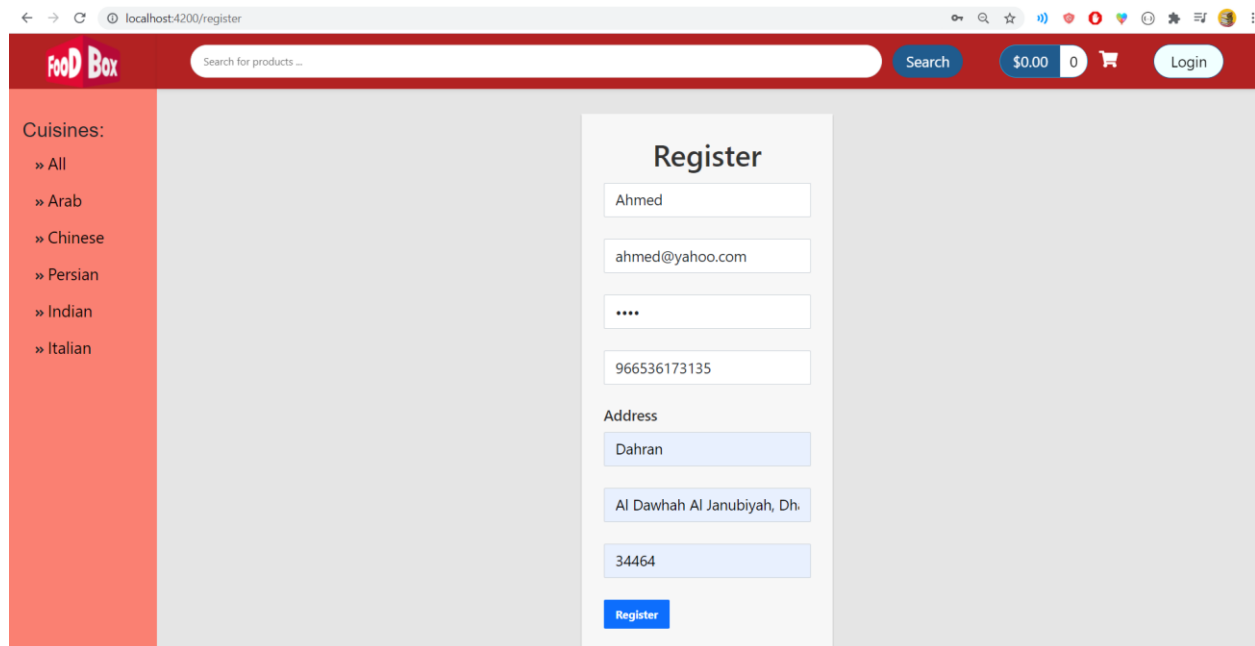
Address

city

street

zipCode

Register



A second screenshot of the 'Food Box' register form, identical to the first but with sample data entered into the fields. The browser address bar shows 'localhost:4200/register'. The page layout is the same, with the red header, search bar, shopping cart, and cuisine sidebar. The 'Register' form fields are now populated: Username is 'Ahmed', Email is 'ahmed@yahoo.com', Password is masked with '****', phone is '966536173135', and the address fields contain 'Dahran', 'Al Dawhah Al Janubiyah, Dh.', and '34464'. The blue 'Register' button remains at the bottom.

Food Box

Search for products ...

Search

\$0.00 0

Login

Cuisines:

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

Register

Ahmed

ahmed@yahoo.com

966536173135

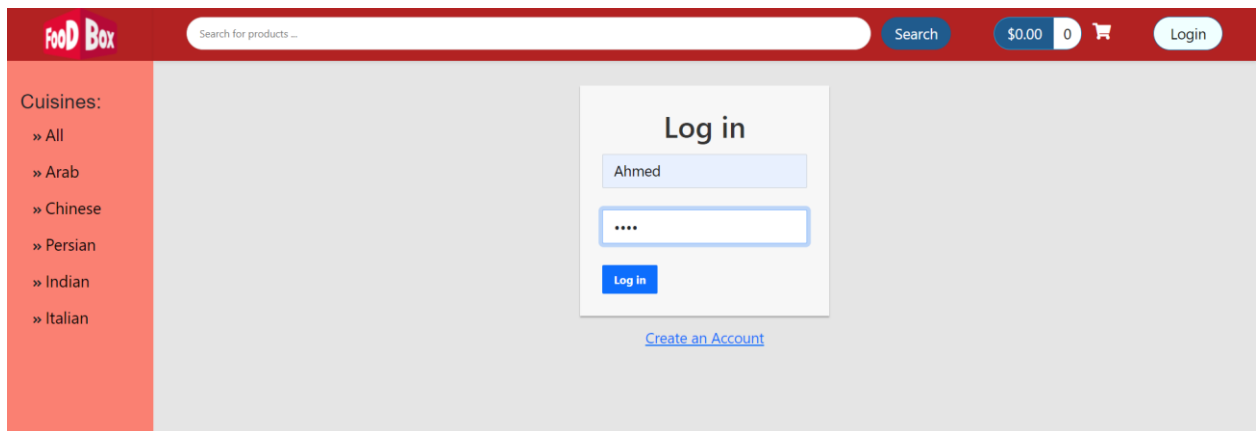
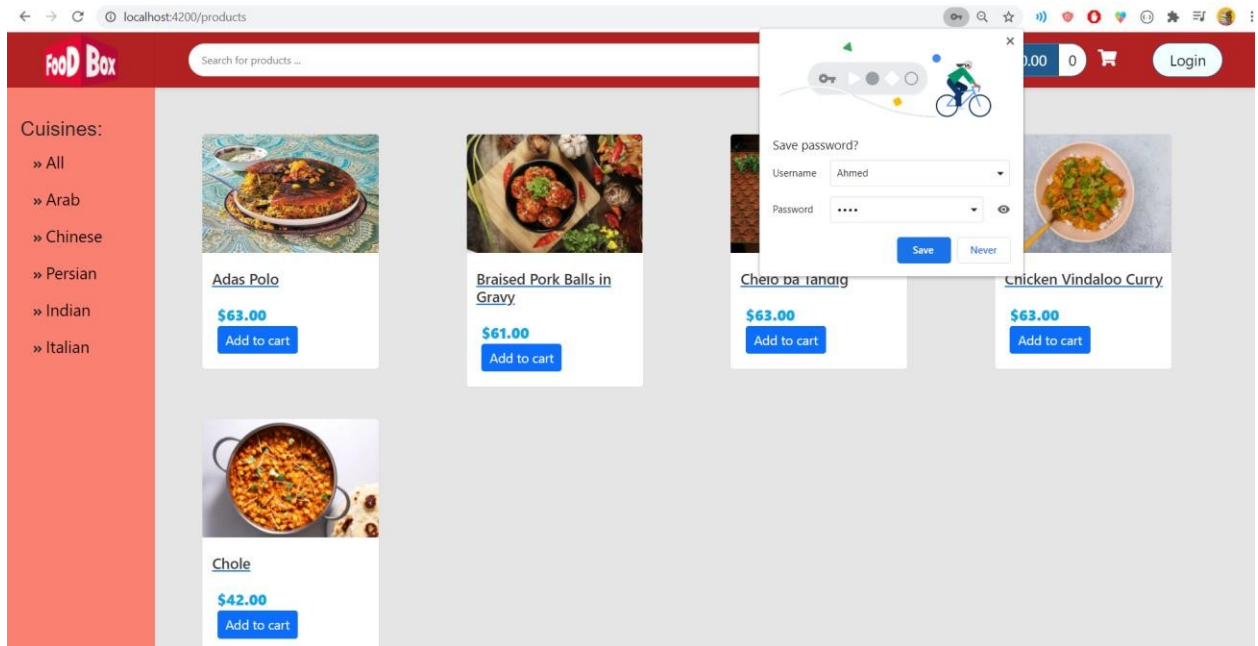
Address

Dahran

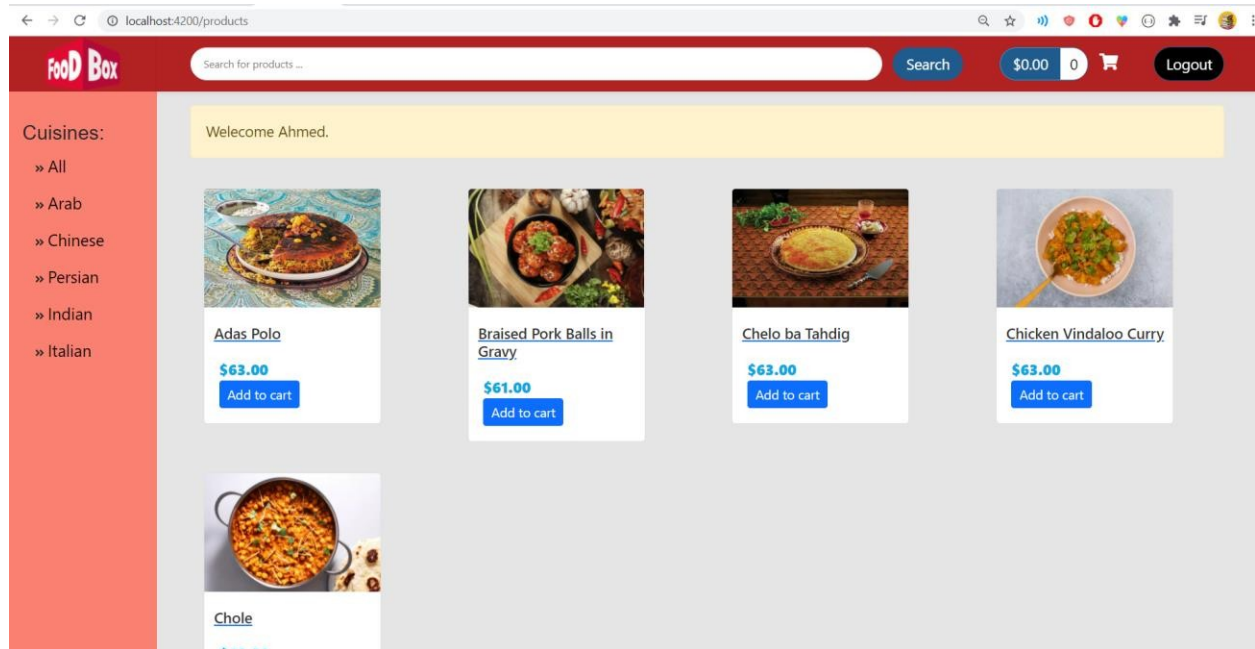
Al Dawhah Al Janubiyah, Dh.

34464

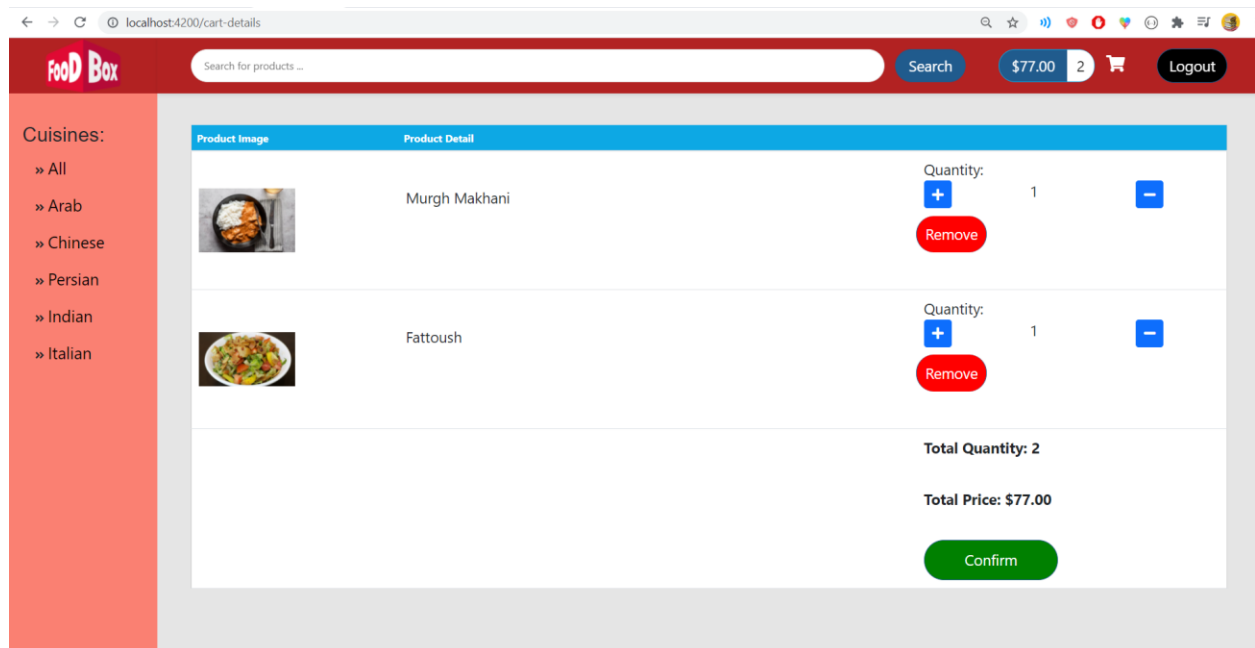
Register



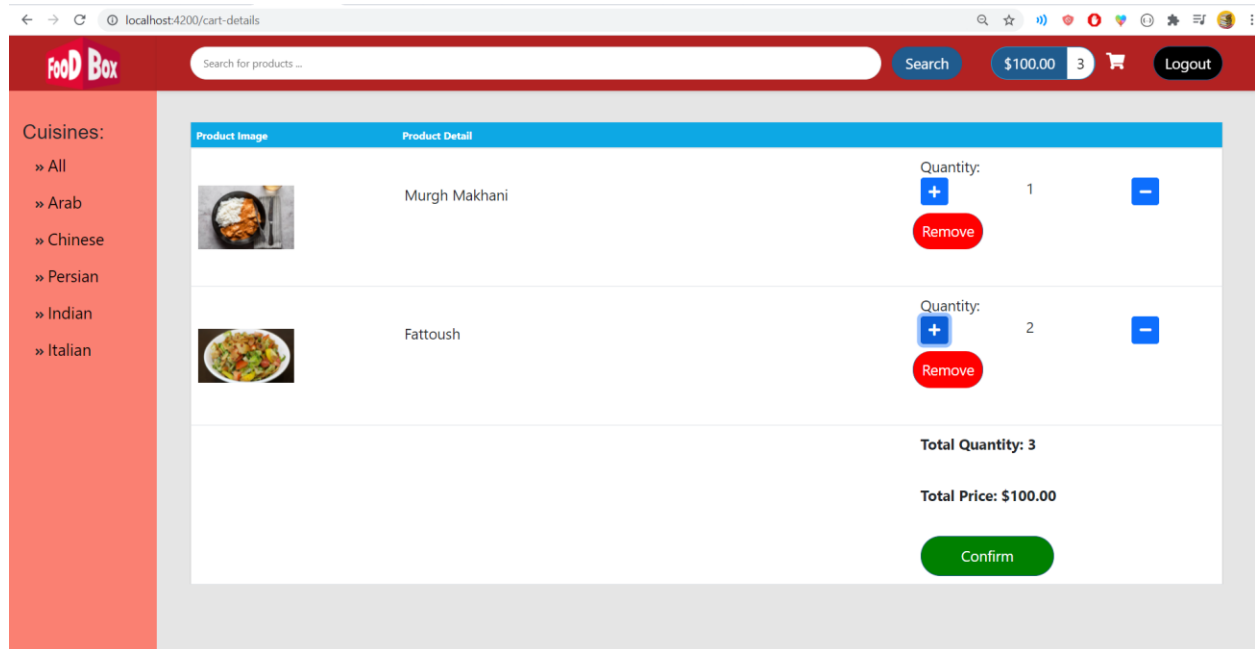
- User logged in



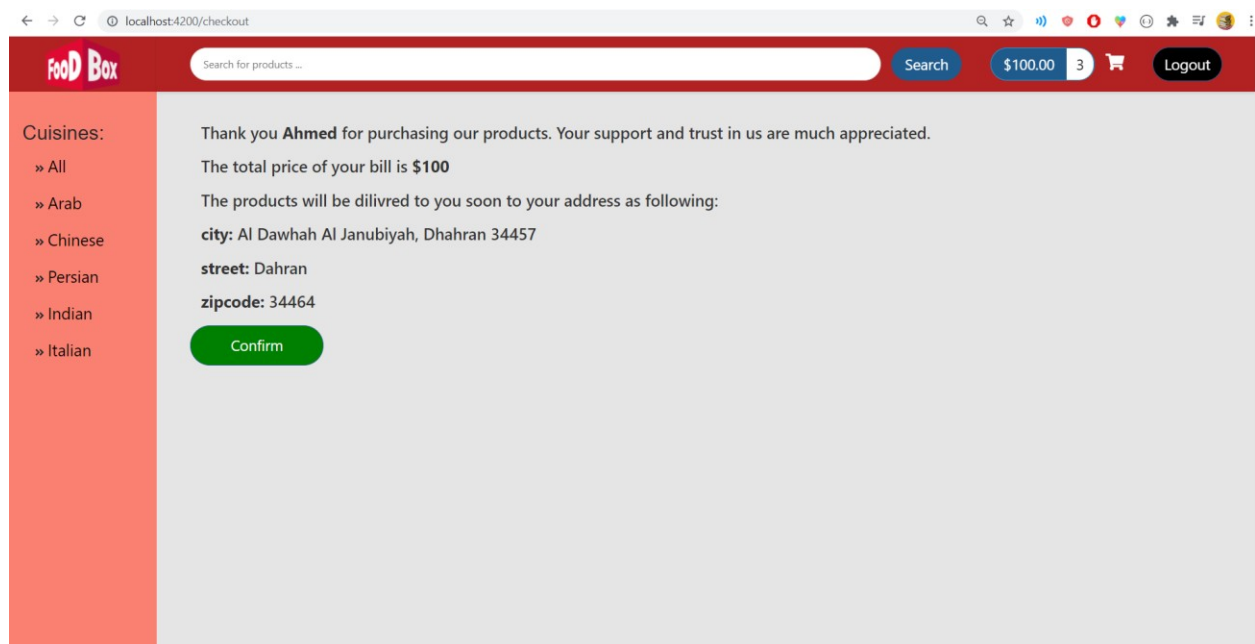
- User can now buy (Payment gateway)



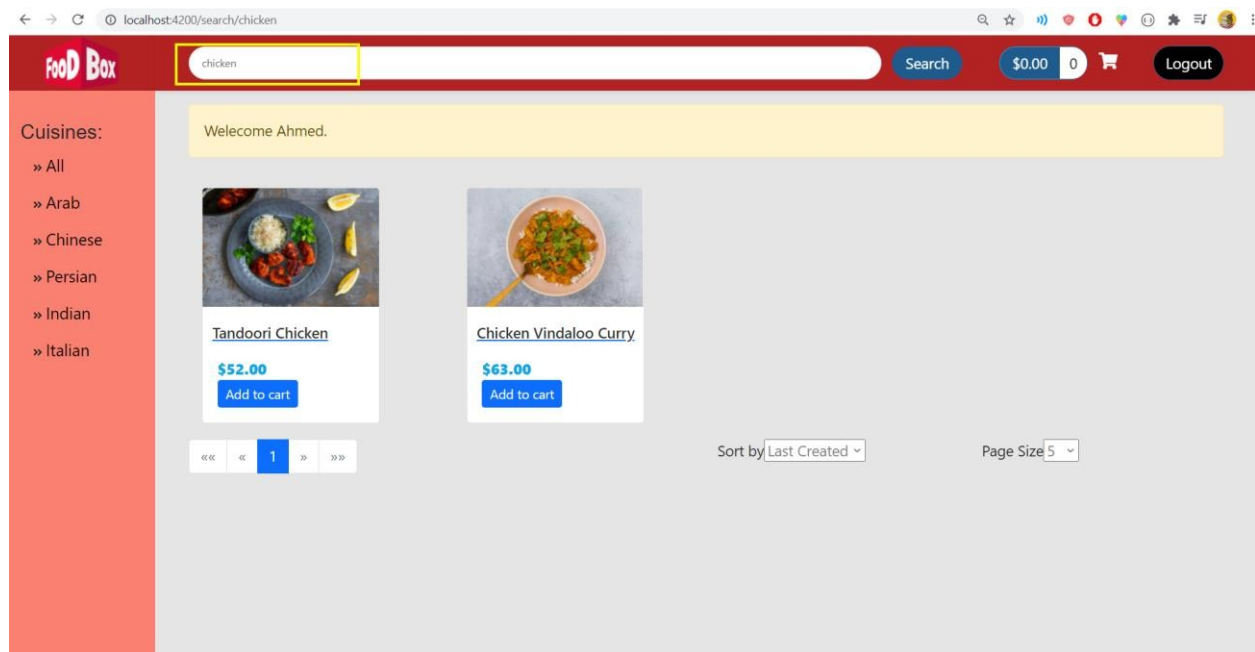
- Can add/remove products during review



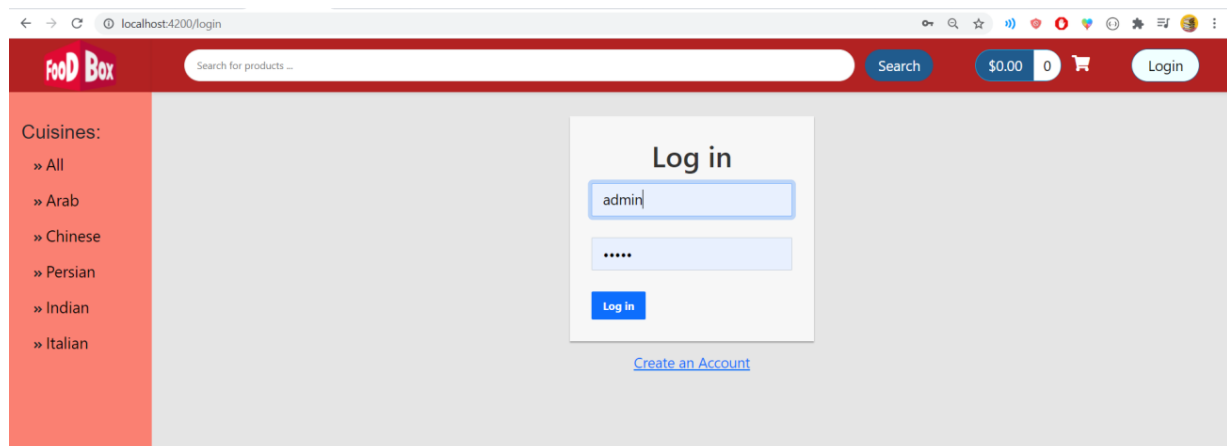
- Check-out and confirm



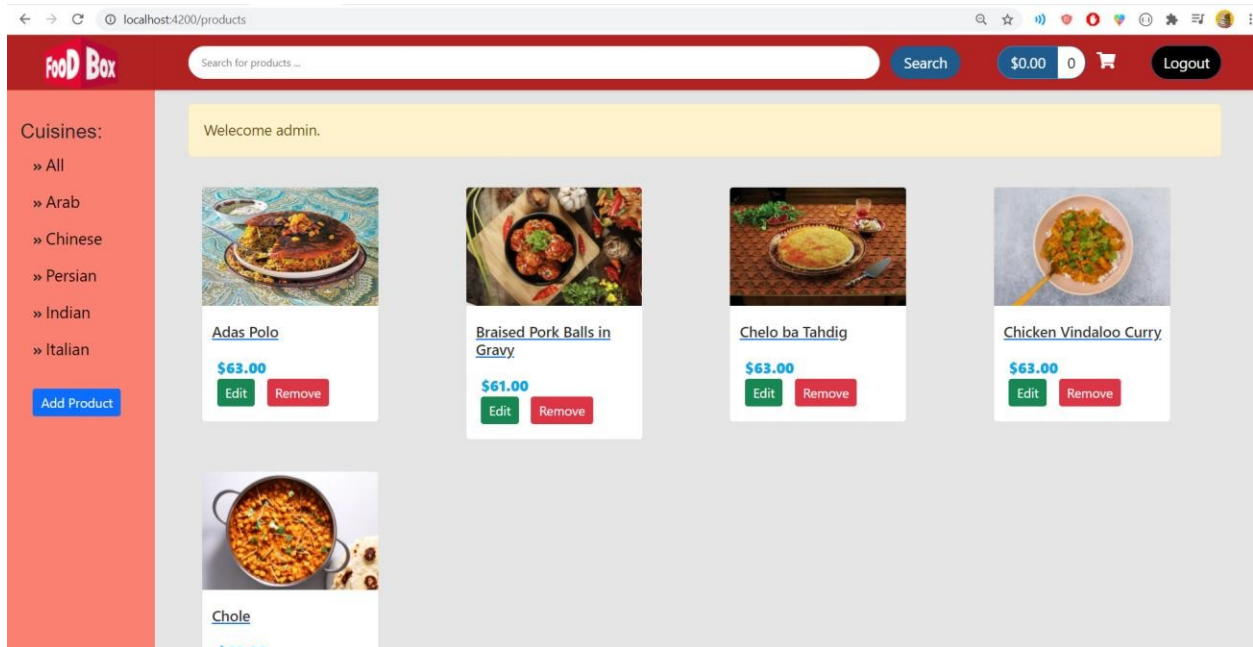
- Search products by Name



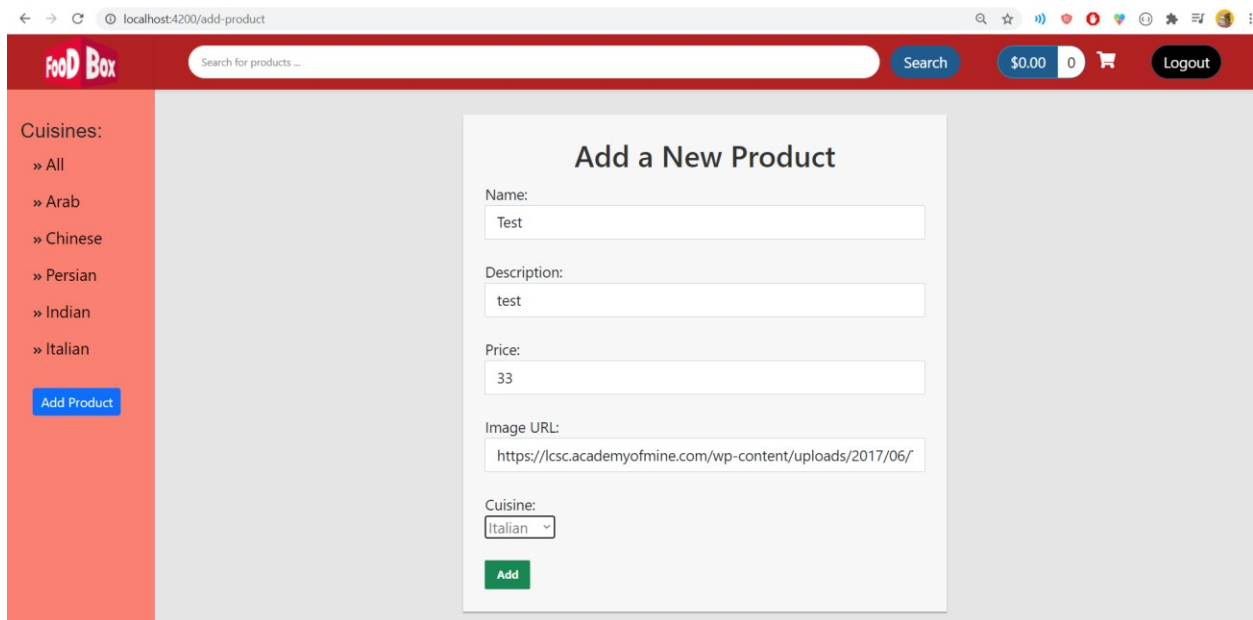
- Now, let's log in as admin
- Admin/admin



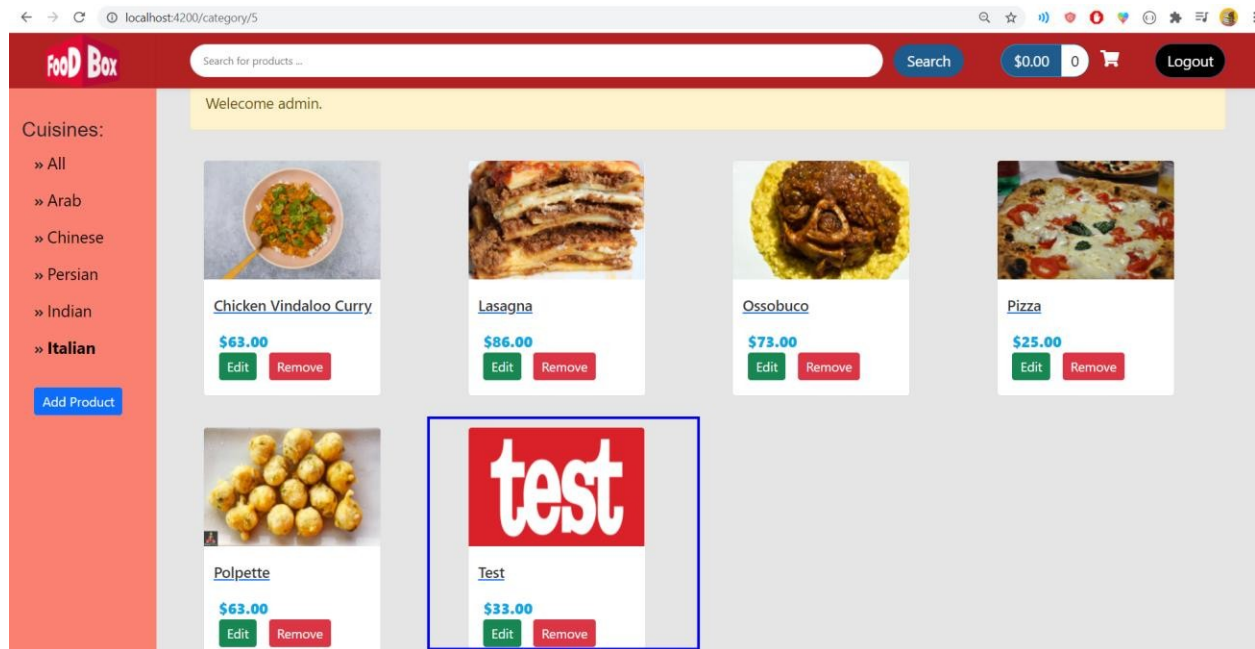
- Admin can:
 - Create product
 - Edit product
 - Delete product



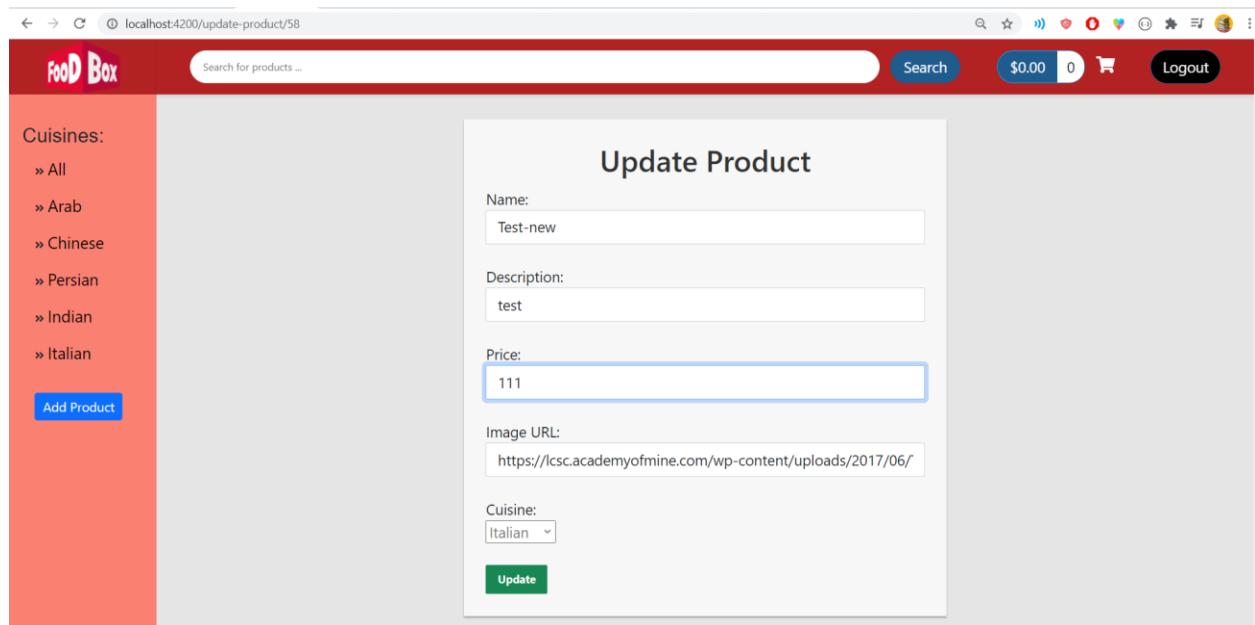
- Create a new product

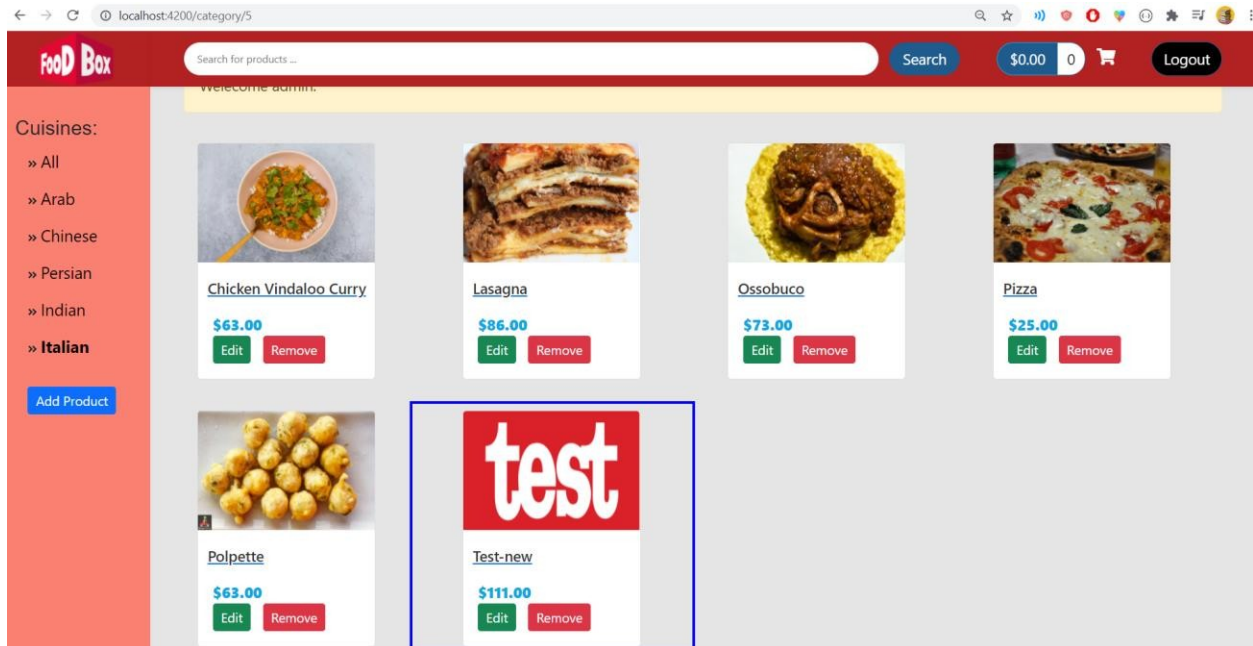


- The new product was added

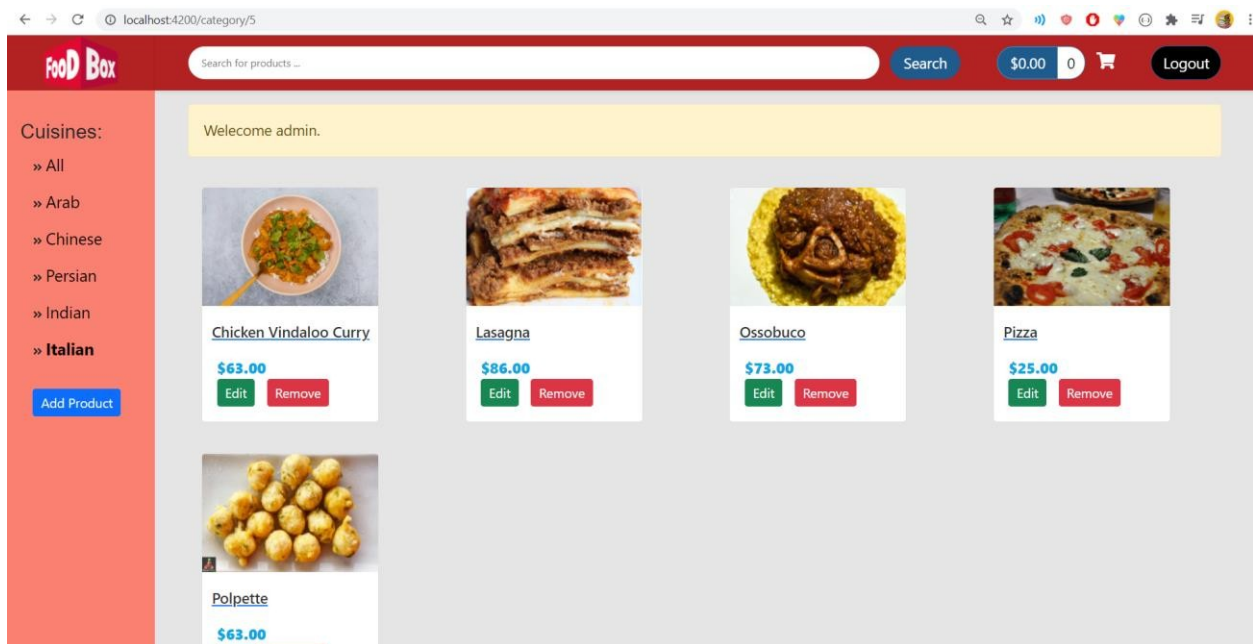


- Let's update it





- Click Remove to remove it



- Deployment

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with options like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity', and 'Reservations'. The main panel is titled 'Instances (3)' and contains a table of EC2 instances. Below the table, there is a large red text overlay that reads 'AWS EC2 instances'.

	Name	Instance ID	Instance ...	Instanc...	Status check	Alarm st...	Availabilit...	Public IPv4
<input type="checkbox"/>	-	i-06ad77f007f99...	⊖ Stopped	t3.micro	-	No ala...+	me-south-1a	-
<input type="checkbox"/>	Master	i-01cf01eb3c5e7...	⊕ Running	t3.medium	2/2 checks pi	No ala...+	me-south-1a	ec2-15-185
<input type="checkbox"/>	Slave	i-0f1faf8730c3a8...	⊕ Running	t3.micro	2/2 checks pi	No ala...+	me-south-1a	ec2-15-185

The screenshot shows the Jenkins web interface for a project named 'jenkins pipeline project'. The main area displays the 'Stage View' of the pipeline. The pipeline consists of several stages: 'Cloning Git', 'Install dependencies', 'Build', 'Package Build', 'Artifacts Creation', and 'Stash changes'. The 'Build' stage is highlighted in red, indicating a failure. The 'Stage View' table shows the duration of each stage for the last few builds.


	Cloning Git	Install dependencies	Build	Package Build	Artifacts Creation	Stash changes
Average stage times: (Average full run time: ~1min 12s)	1s	13s	38s	373ms	882ms	259ms
#27 Aug 27 12:19 No Changes	1s	5s	1min 2s	373ms	882ms	259ms
#26 Aug 27 12:12 No Changes	1s	4s	36s			
#25 Aug 27 9:02 AM			failed			
#24 Aug 27 9:01 AM			39s			

← → ↻ <http://ec2-15-185-146-219.me-south-1.compute.amazonaws.com> 🔊 🔒 🏠 📄 📁 📧 ⋮

Food Box \$0.00 0


Cuisines:

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




Adas Polo

\$63.00




Braised Pork Balls in Gravy

\$61.00




Chelo ba Tahdig

\$63.00



Chicken Vindaloo Curry

\$63.00



Chole

\$42.00