

# **CALTECH Full Stack Java Developer Capstone Project**

## **SANDEEP KUMAR JAKKARAJU**

*A Food Delivery Website -- Foody !!*

Project Objective:

Create a dynamic and responsive online food delivery web application for ordering food items of different cuisines from a restaurant.

Contains an Admin and an User portal.

When you login with Admin user you will be redirected to Admin Portal.

When you login with Non-Admin user you will be redirected to the End User Portal.

Admin Portal Features:

1. Signup/Sign In
2. Add and Remove Cuisines
3. Add and Remove Food Items For a Cuisine
4. Edit Food Items
5. Change Password
6. Logout

User Portal Features:

1. Signup/Sign In
2. Search food items based on keywords
3. Apply Filters on Search Results.
4. Add selected items to cart and customise order.
5. Pay for the order and receive an order summary.
6. Change Password and Logout.

=====

GIT REPOSITORY: <https://github.com/sandycaltechpgp/capstone-project>

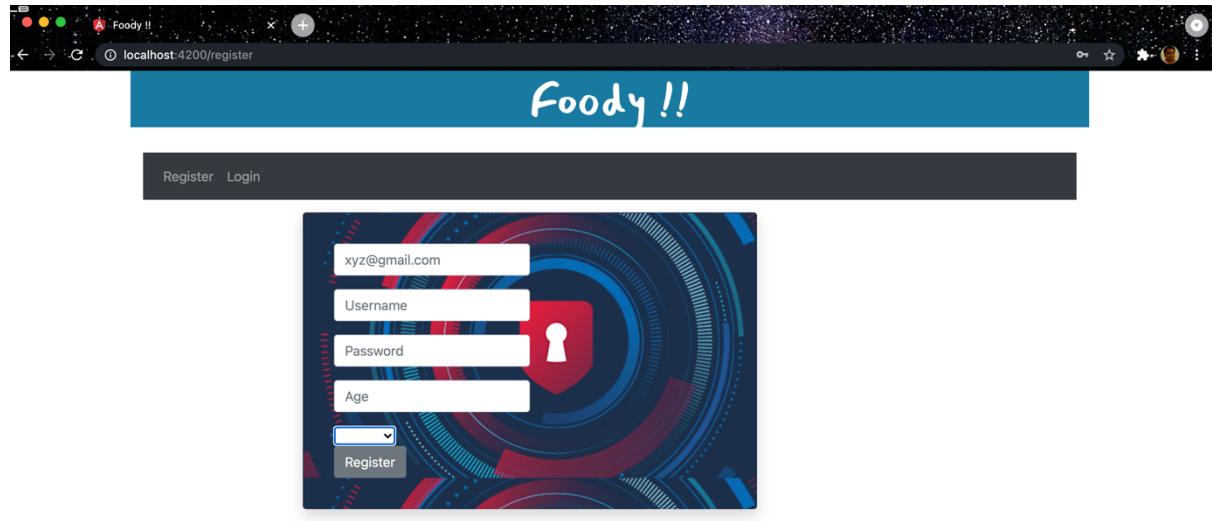
=====

TECH STACK:

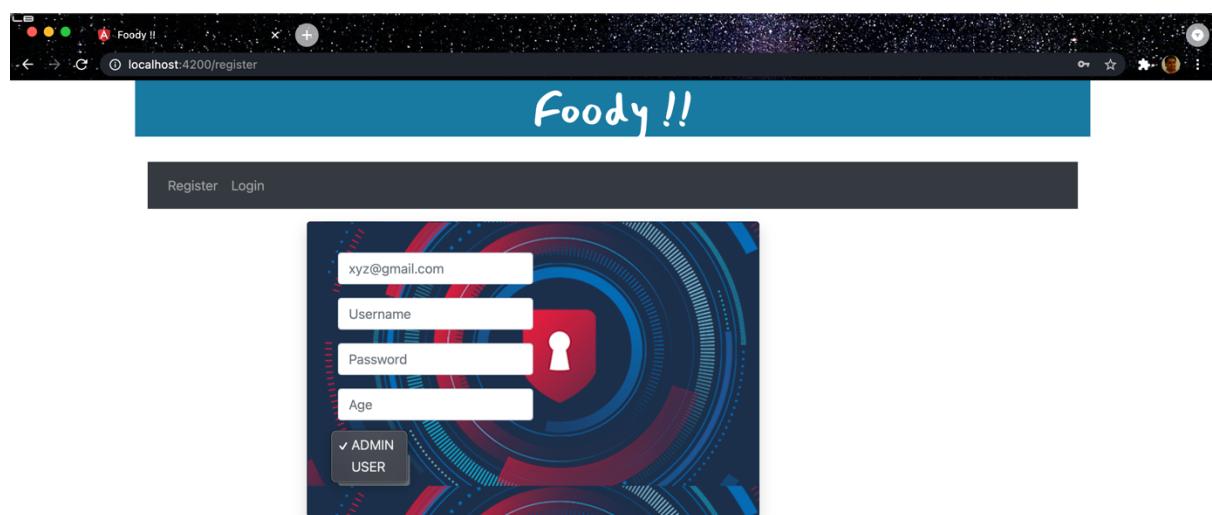
Java 8, Spring boot, JPA, MySQL, Docker, AWS, Angular 6, HTML, JS, Jenkins, Maven, GIT, GITHUB

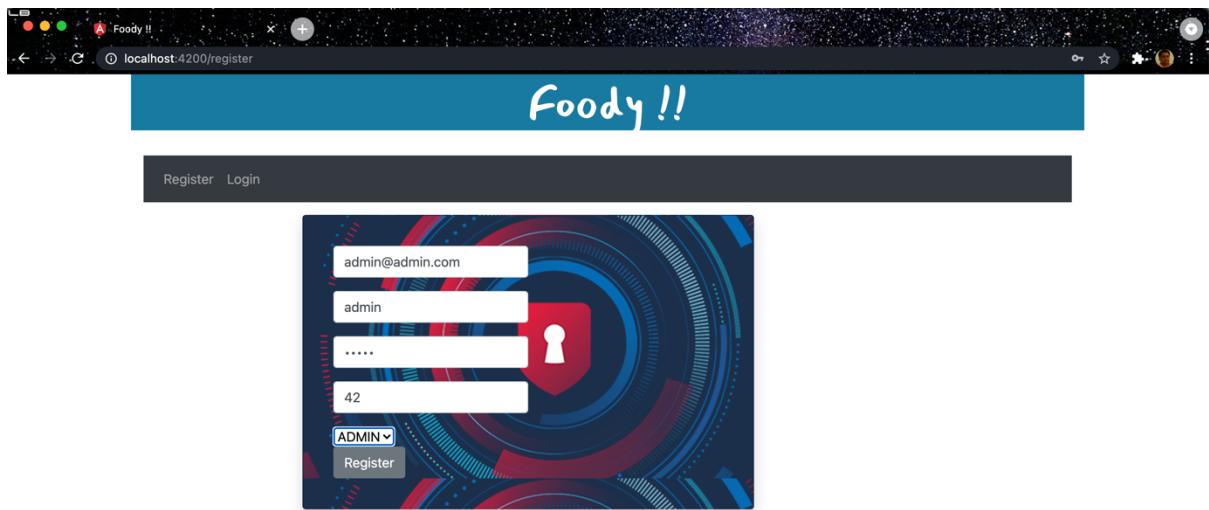
=====

REGISTRATION PAGE: ( BOTH USER AND ADMIN CAN REGISTER HERE )

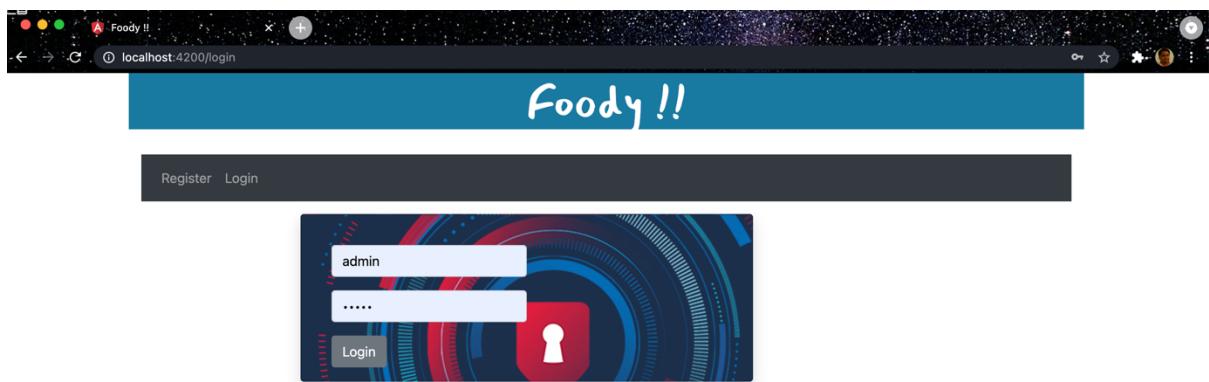


CHOOSE – ADMIN/USER





LOGIN : ADMIN



LANDING PAGE FOR ADMIN USER:

The screenshot shows a web browser window titled "Foody !!". The URL in the address bar is "localhost:4200/admin". The main content area has a blue header bar with the text "Foody !!". Below the header is a dark navigation bar with links: Home, Orders, ChangePassword, Add Cusine, and Log out. The main content area is titled "Cusines" and features a "Click to Add" button. There are four cards, each representing a cuisine: "dosa" (with an image of a dosa), "chickencurry" (with an image of chicken curry), "chickentikka" (with an image of chicken tikka), and "fish fry" (with an image of fish fry). Each card has "Edit" and "Remove" buttons.

SIDE MENU TO FILTER BY CUISINE:

The screenshot shows the same web browser window as the previous one, but with a vertical sidebar on the left. The sidebar is titled "Cusines" and lists several categories: "Mughlai", "North India", "South India", and "Chinese". The main content area is identical to the first screenshot, showing the "Cusines" title, "Click to Add" button, and four cuisine cards: "dosa", "chickencurry", "chickentikka", and "fish fry", each with "Edit" and "Remove" buttons.

## FILTER BY CUISINE:

The screenshot shows a web application titled "Foody !!". The main header has a blue bar with the title. Below it is a navigation bar with links: Home, Orders, ChangePassword, Add Cusine, and Log out. On the left, there's a sidebar titled "Cusines" with a list of categories: Mughlai, North India, South India, and Chinese. The main content area is titled "Cuisines" and contains a button "Click to Add". Below this are two cards: one for "naan" showing an image of flatbread with dipping sauces, and another for "veg thali" showing an image of a traditional Indian meal on a platter. Each card has "Edit" and "Remove" buttons.

## ADD FOOD ITEMS - ADMIN

The screenshot shows a "Cusines" section with a "Click to hide" button. Below it is a form for adding a product. The form includes fields for Product Name, description, quantity, price, and a dropdown menu set to "Mughlai". There is also a file input field labeled "Choose file" with the message "No file chosen" and an "Add Product" button. To the right of the form is an "Image preview" area featuring a camera icon and the text "NO IMAGE". At the bottom, there are four small preview cards for "dosa", "chickencurry", "chickentikka", and "fish fry".

## DEMO OF ADD FOOD ITEM WITH CUISINE:

The screenshot shows a form for adding a food item. The title is 'Cuisines'. The form fields include:

- Name: cheese naan
- Description: cheese naan
- Price: 1000
- Calories: 100
- Cuisine: North India
- Image preview: An image of cheese naan.
- Image file: cheese-naan.jpeg
- Action button: Add Product

## COME BACK TO LANDING PAGE OF ADMIN:

The screenshot shows the main admin dashboard with the title 'Foody !!'. The left sidebar lists cuisines: Mughlai, North India, South India, and Chinese. The main content area shows a grid of food items:

Food Item	Image	Action Buttons
naan		Edit Remove
veg thali		Edit Remove
cheese naan		Edit Remove

## CHANGE PASSWORD - ADMIN

The screenshot shows a web browser window titled "Foody !!". The URL in the address bar is "localhost:4200/admin/cp". The page has a dark header with the title "Foody !!". Below the header is a navigation bar with links: Home, Orders, ChangePassword, Add Cusine, and Log out. The main content area contains a form for changing a password. The form fields are: "Old Password" (input type="password"), "New Password" (input type="password"), "Confirm Password" (input type="password"), and a "Change Password" button.

---

## ADD CUISINE:

The screenshot shows a web browser window titled "Foody !!". The URL in the address bar is "localhost:4200/admin/addcat". The page has a dark header with the title "Foody !!". Below the header is a navigation bar with links: Home, Orders, ChangePassword, Add Cusine, and Log out. The main content area contains a form for adding a cuisine. The form field is: "Cusine" (input type="text") containing the value "Leboneese". Below the input field is a "Add Cusine" button.

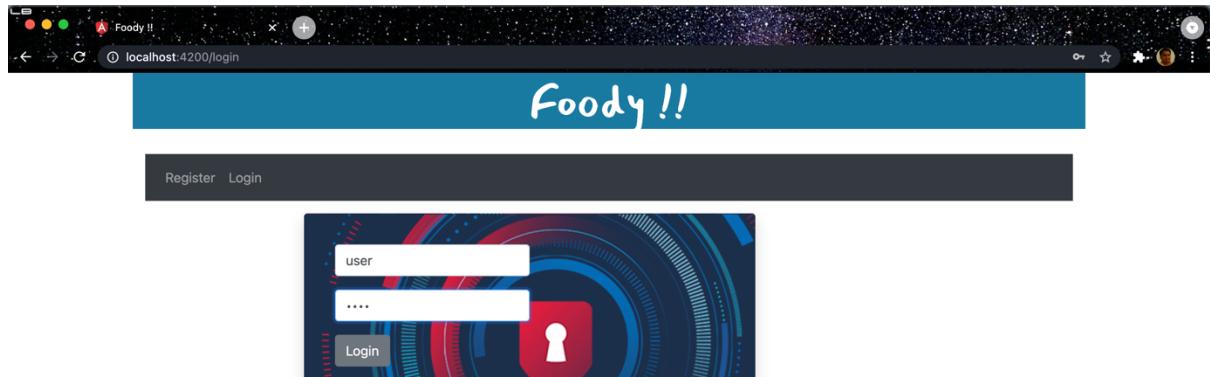
SEE THE NEW CUISINE IN LEFT MENU:

The screenshot shows the 'Cuisines' section of the Foody !! application. On the left, a sidebar lists various cuisines: Mughlai, North India, South India, Chinese, and Leboneese. The main content area is titled 'Cusines' and contains a button 'Click to Add'. Below this are four cards, each representing a dish: 'dosa' (with an image of a dosa), 'chickencurry' (with an image of chicken curry), 'chickentikka' (with an image of chicken tikka), and 'fish fry' (with an image of fish fry). Each card has 'Edit' and 'Remove' buttons at the bottom.

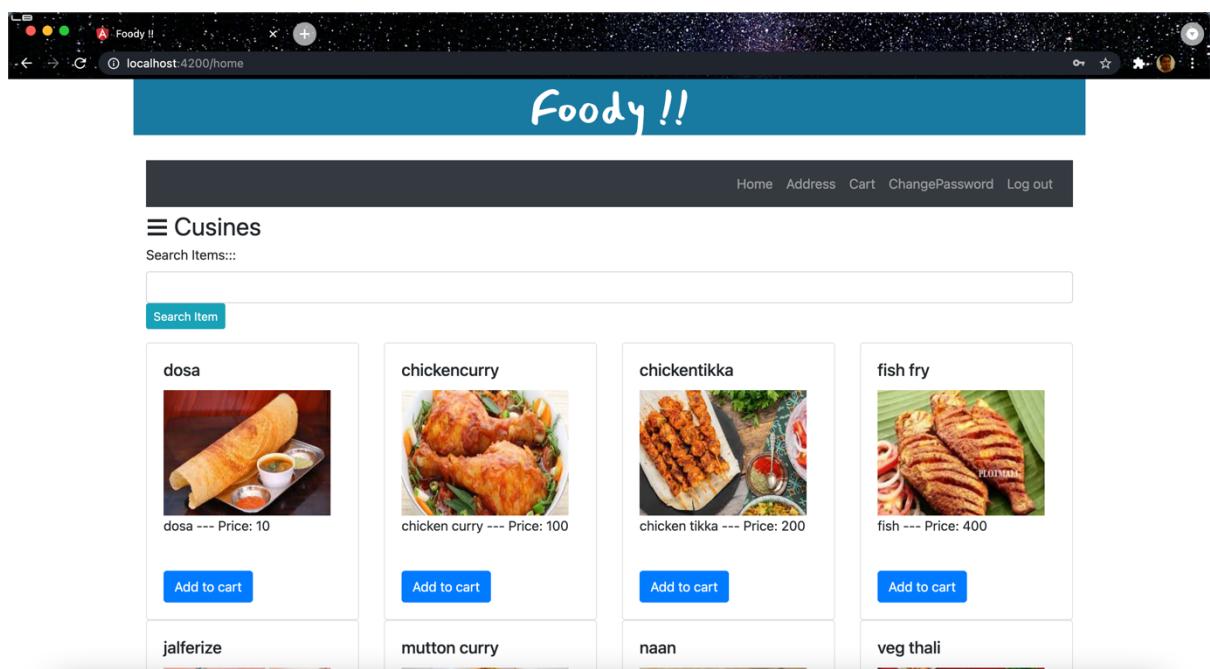
REGISTER END USER:

The screenshot shows the registration page for end users. The top navigation bar includes 'Register' and 'Login' links. The main form is set against a dark blue background featuring a circular, futuristic design. It contains four input fields: email ('user@user.com'), password ('user'), confirm password ('....'), and age ('53'). A dropdown menu shows 'USER' selected. To the right of the inputs is a red shield icon containing a keyhole. At the bottom of the form is a 'Register' button.

LOGIN END USER:



LANDING PAGE FOR END USER:



LEFT MENU FOR END USER:

The screenshot shows a web browser window for 'Foody !!' at 'localhost:4200/home'. A vertical sidebar on the left lists cuisines: 'Mughlai', 'North India', 'South India', 'Chinese', and 'Leboneese'. The main content area displays a grid of food items under the heading 'Cusines'. Each item has a thumbnail, name, price, and an 'Add to cart' button.

Item	Description	Price	Action
dosa	A dosa served with chutney and sambar.	Price: 10	Add to cart
chickencurry	A plate of chicken curry with rice.	Price: 100	Add to cart
chickentikka	Chicken tikka skewers with naan bread.	Price: 200	Add to cart
fish fry	Grilled fish fillets with a side dish.	Price: 400	Add to cart
jalferize	A dish of jalebi or similar sweet.		
mutton curry	A plate of mutton curry.		
naan	A piece of naan bread.		
veg thali	A traditional Indian meal platter.		

SEARCH WITH KEYWORDS END USER:

The screenshot shows the same 'Foody !!' application interface. A search bar in the main content area contains the keyword 'dosa'. Below the search bar, a single item is displayed: 'dosa' with a price of 'Price: 10' and an 'Add to cart' button.

## SEARCH BY CUISINE LEFT MENU END USER

The screenshot shows a web browser window for the 'Foody !!' application at [localhost:4200/home](http://localhost:4200/home). The page has a dark theme with a blue header bar containing the logo 'Foody !!'. A left sidebar titled 'Cusines' lists various cuisines: Mughlai, North India, South India, Chinese, and Leboneese. The main content area is titled 'Cusines' and contains a search bar with placeholder text 'Search Items:::' and a 'Search Item' button. Below the search bar are four cards, each representing a dish:

- chickencurry**: chicken curry --- Price: 100. Includes an 'Add to cart' button.
- chickentikka**: chicken tikka --- Price: 200. Includes an 'Add to cart' button.
- fish fry**: fish --- Price: 400. Includes an 'Add to cart' button.
- mutton curry**: mutton --- Price: 455.9800109863281. Includes an 'Add to cart' button.

## ADD ADDRESS END USER

The screenshot shows a web browser window for the 'Foody !!' application at [localhost:4200/home/address](http://localhost:4200/home/address). The page has a dark theme with a dark header bar containing the navigation links 'Home', 'Address', 'Cart', 'ChangePassword', and 'Log out'. The main content area contains a form for adding an address:

Address:

City:

State:

Country:

Zipcode:

Phone Number:

## ADD ITEM TO CART:

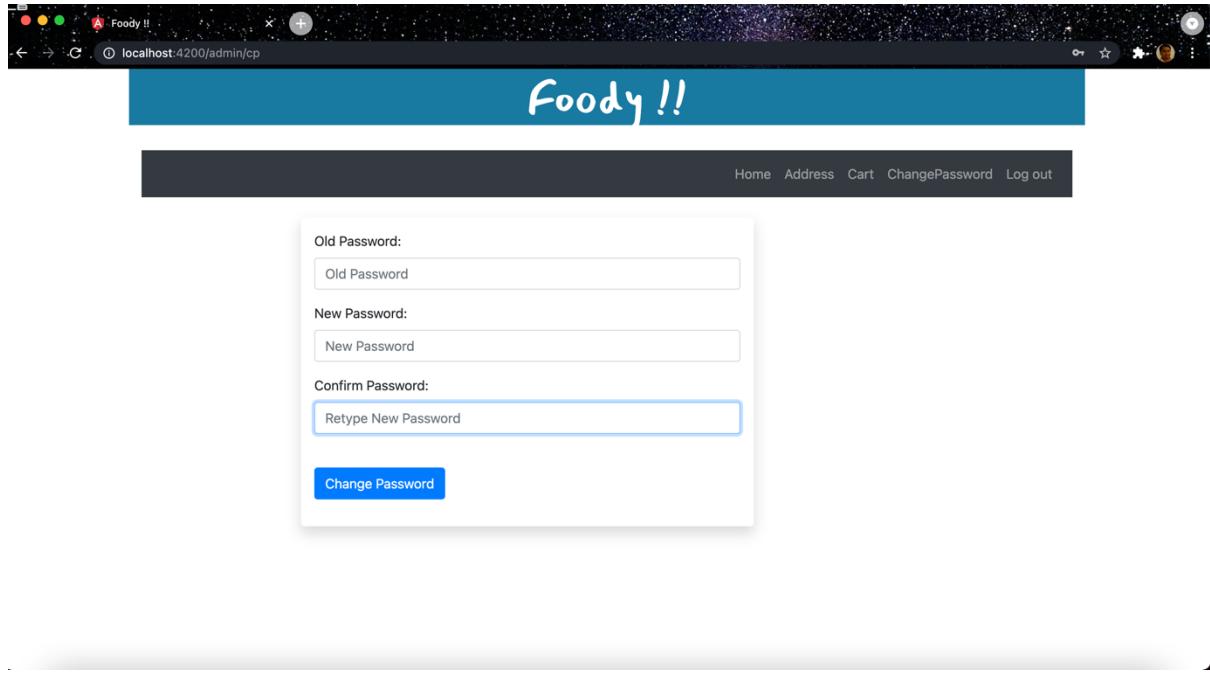
The screenshot shows a web browser window with the title 'Foody !!' in the tab bar. The address bar displays 'localhost:4200/home/cart'. The main content area has a blue header bar with the text 'Foody !!' in white. Below this is a dark navigation bar with links: Home, Address, Cart, ChangePassword, and Log out. A blue button labeled '< Continue Shopping' is visible. A message 'To change Quantity, Edit Quantity and Click on Update button' is displayed. A table lists a single item: 'dosa' with a price of '\$10.00'. The quantity is set to '1' in a text input field, with an 'Update' button next to it. To the right of the table, the subtotal is '\$10.00' and there is a 'Delete' button. At the bottom left, it says 'Total Sum : \$10.00'. On the far right, there is a blue button labeled 'Checkout >'.

Product	Price	Quantity	Subtotal
dosa	\$10.00	<input type="text" value="1"/>	\$10.00

Total Sum : \$10.00

Checkout >

CHANGE PASSWORD END USER:



=====

USING DOCKER TO RUN THE APP ON LOCALHOST:

*##### NOTE: NEEDS DOCKER SERVER RUNNING ON THE HOST #####*

BUILD THE DOCKER IMAGE WITH BELOW COMMAND:

```
$ docker build . -t foody:latest
```

RUN THE DOCKER IMAGE ON LOCALHOST

```
$ docker run -p80:80 -p8087:8087 foody:latest
```

ACCESS THE APP AT BELOW URL IN BROWSER:

URL: <http://localhost/>

=====

RUN THE APP ON LOCALHOST WITH BASH COMMANDS: (MAC/LINUX)

```
~/capstone-project$ cd backend-app
```

```
~/capstone-project/backend-app$ mvn install -DskipTests
```

THIS WILL RUN THE BACKEND SPRING BOOT SERVICES

```
~/capstone-project/backend-app $ mvn spring-boot run
```

ACCESS THE APP ON :

<http://localhost:8087>

---

```
~/capstone-project$ cd front-end-app
```

```
~/capstone-project/front-end-app$ npm install
```

THIS WILL RUN THE FRONTEND WEBSITE

```
~/capstone-project/backend-app $ ng serve
```

ACCESS THE APP ON:

<http://localhost:4200>

---

## CI/CD PIPELINE AUTOMATION – JENKINS AND AWS

### AWS EC2 CONSOLE

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Capacity Reservations, Images (AMIs), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays a table titled 'Instances (3) Info' with columns: Name, Instance ID, Instance ... (Status: Stopped), Instance type (t2.micro), Status check (blank), Alarm status (No alarms), and Public IPv4 DNS. The instance IDs listed are i-034be52a7b4cd0f8a, i-00e5b917bdbef960, and i-0bdd07e71bed62602.

Name	Instance ID	Instance ... Status: Stopped	Instance type	Status check	Alarm status	Public IPv4 DNS
-	i-034be52a7b4cd0f8a	Stopped	t2.micro	-	No alarms	-
-	i-00e5b917bdbef960	Stopped	t2.micro	-	No alarms	-
-	i-0bdd07e71bed62602	Stopped	t2.micro	-	No alarms	-

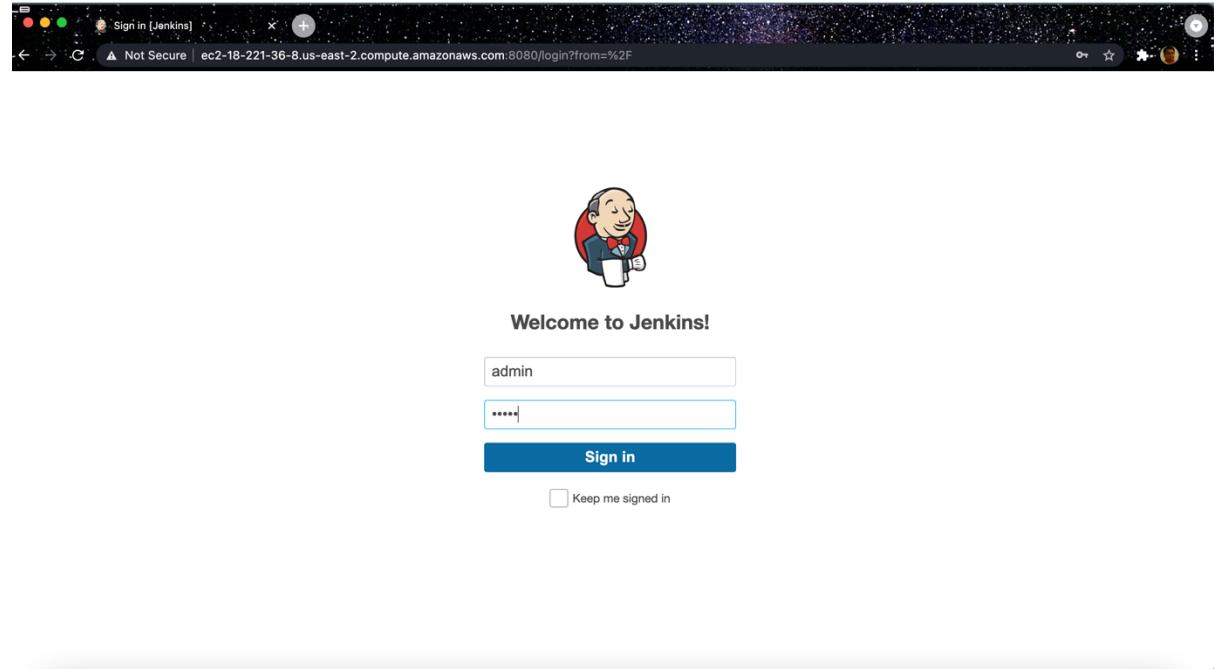
## INSTALL AND START JENKINS ON EC2

```
>Last login: Wed May 19 08:17:08 on console
[sandeepjakkaraju@sandeeps-MBP ~] % ssh -i "awstest.pem" ec2-user@ec2-18-221-36-8.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-221-36-8.us-east-2.compute.amazonaws.com (18.221.36.8)' can't be established.
ECDSA key fingerprint is SHA256:EjImwRp37ppexAf0JNyz4oPfs+HRV6CZFR9y+d70/I.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-221-36-8.us-east-2.compute.amazonaws.com,18.221.36.8' (ECDSA) to the list of known hosts.
Last login: Mon May 17 20:00:39 2021 from 49.206.53.12

[|_(-_-_)_
 _\|_|_|_|

https://aws.amazon.com/amazon-linux-2/
[bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-42-167 ~]$ 
[ec2-user@ip-172-31-42-167 ~]$ 
[ec2-user@ip-172-31-42-167 ~]$ 
[ec2-user@ip-172-31-42-167 ~]$ 
[ec2-user@ip-172-31-42-167 ~]$ 
[ec2-user@ip-172-31-42-167 ~]$ sudo service jenkins start
Starting jenkins (via systemctl): [ OK ]
[ec2-user@ip-172-31-42-167 ~]$ 
```

## LOGIN TO JENKINS:



## Configure the JENKINS FREE STYLE PROJECT

The screenshot shows the Jenkins dashboard with the CapstoneProject job listed. The job has the following details:

S	W	Name ↓	Last Success	Last Failure	Last Duration
X	Cloud icon	CapstoneProject	1 day 16 hr - #22	1 day 16 hr - #25	5 min 38 sec

Legend: S M L  
Atom feed for all | Atom feed for failures | Atom feed for just latest builds

Left sidebar navigation includes: New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View.

## JENKINS PROJECT CONFIGURATION: using docker

The screenshot shows the configuration page for the CapstoneProject job. The "Build Environment" tab is selected. The "Build" section contains the following configuration:

- Execute shell
- Command:

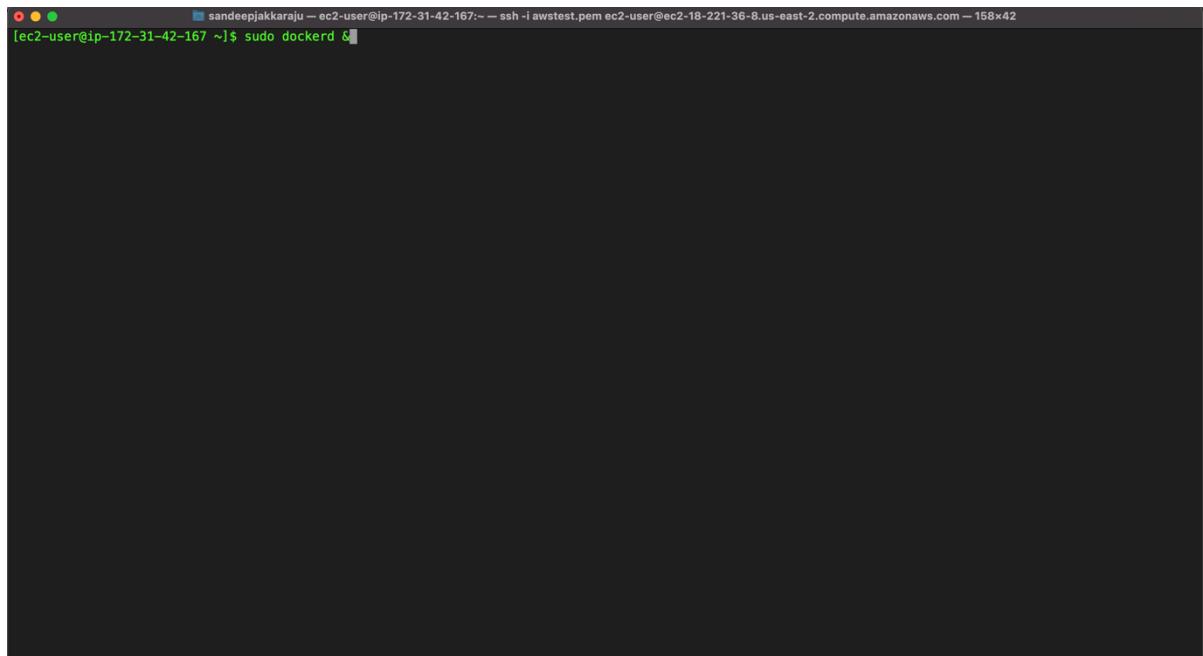
```
rm -rf /var/lib/jenkins/workspace/CapstoneProject/capstone-project
git clone https://github.com/sandycaltechpp/capstone-project.git
cd /var/lib/jenkins/workspace/CapstoneProject/capstone-project
sudo docker build -t myimage:latest < DockerfileJenkins
sudo docker run -p80:80 -p8087:8087 myimage:latest
```
- See the list of available environment variables
- Add build step ▾

The "Post-build Actions" section is currently empty.

Buttons at the bottom: Save and Apply.

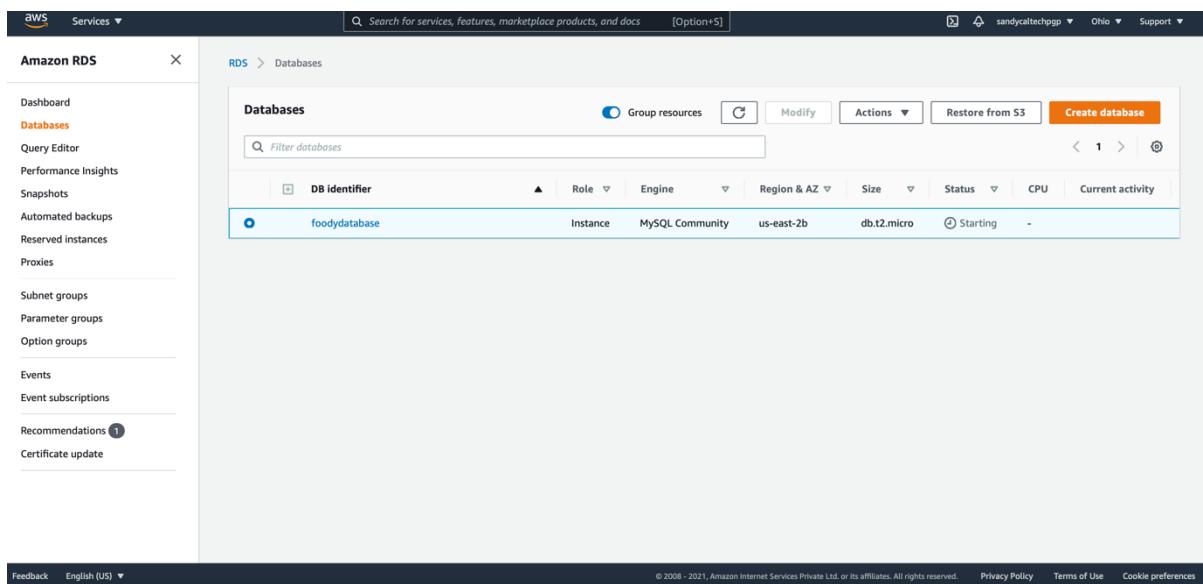
## START DOCKER SERVER ON EC2 INSTANCE

NOTE:: PLEASE HAVE MORE DISK SPACE IN THE INSTANCE FOR DOCKER TO RUN.



```
sandeepjakkaru ~ ec2-user@ip-172-31-42-167:~ ssh -i awstest.pem ec2-user@ec2-18-221-36-8.us-east-2.compute.amazonaws.com - 158x42
[ec2-user@ip-172-31-42-167 ~]$ sudo dockerd &
```

## START RDS – MYSQL



The screenshot shows the Amazon RDS console under the 'Databases' section. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, Recommendations (1), and Certificate update. The main area displays a table for databases. The table has columns for DB identifier, Role, Engine, Region & AZ, Size, Status, CPU, and Current activity. One row is visible for 'foodydatabase', which is listed as an Instance of MySQL Community in the us-east-2b region, db.t2.micro size, and currently 'Starting'. There are buttons for Group resources, Modify, Actions, Restore from S3, and Create database at the top of the table.

----THE END---