

Backend

MavenWrapperDownloader.java

```
import java.net.*;
import java.io.*;
import java.nio.channels.*;
import java.util.Properties;

public class MavenWrapperDownloader {

    private static final String WRAPPER_VERSION =
"0.5.6";
    private static final String DEFAULT_DOWNLOAD_URL =
"https://repo.maven.apache.org/maven2/io/takari/maven-
wrapper/"
        + WRAPPER_VERSION + "/maven-wrapper-" +
WRAPPER_VERSION + ".jar";
    private static final String
MAVEN_WRAPPER_PROPERTIES_PATH =
        ".mvn(wrapper/maven-wrapper.properties";

    private static final String MAVEN_WRAPPER_JAR_PATH =
        ".mvn(wrapper/maven-wrapper.jar";
    private static final String
PROPERTY_NAME_WRAPPER_URL = "wrapperUrl";

    public static void main(String args[]) {
        System.out.println("- Downloader started");
        File baseDirectory = new File(args[0]);
    }
}
```

```
        System.out.println("- Using base directory: " +  
baseDirectory.getAbsolutePath());  
        File mavenWrapperPropertyFile = new  
File(baseDirectory, MAVEN_WRAPPER_PROPERTIES_PATH);  
        String url = DEFAULT_DOWNLOAD_URL;  
        if(mavenWrapperPropertyFile.exists()) {  
            FileInputStream  
mavenWrapperPropertyFileInputStream = null;  
            try {  
                mavenWrapperPropertyFileInputStream =  
new FileInputStream(mavenWrapperPropertyFile);  
                Properties mavenWrapperProperties = new  
Properties();  
  
mavenWrapperProperties.load(mavenWrapperPropertyFileInpu  
tStream);  
                url =  
mavenWrapperProperties.getProperty(PROPERTY_NAME_WRAPPER  
_URL, url);  
            } catch (IOException e) {  
                System.out.println("- ERROR loading '" +  
MAVEN_WRAPPER_PROPERTIES_PATH + "'");  
            } finally {  
                try {  
  
if(mavenWrapperPropertyFileInputStream != null) {  
  
mavenWrapperPropertyFileInputStream.close();  
                }  
            } catch (IOException e) {  
                }  
        }
```

```
        }

        System.out.println("- Downloading from: " +
url);

        File outputFile = new
File(baseDirectory.getAbsolutePath(),
MAVEN_WRAPPER_JAR_PATH);

        if(!outputFile.getParentFile().exists()) {
            if(!outputFile.getParentFile().mkdirs()) {
                System.out.println(
                    "- ERROR creating output
directory '" +
outputFile.getParentFile().getAbsolutePath() + "'");

            }
        }

        System.out.println("- Downloading to: " +
outputFile.getAbsolutePath());

        try {
            downloadFileFromURL(url, outputFile);
            System.out.println("Done");
            System.exit(0);
        } catch (Throwable e) {
            System.out.println("- Error downloading");
            e.printStackTrace();
            System.exit(1);
        }
    }

    private static void downloadFileFromURL(String
 urlString, File destination) throws Exception {
        if (System.getenv("MVNW_USERNAME") != null &&
System.getenv("MVNW_PASSWORD") != null) {
```

```
        String username =
System.getenv("MVNW_USERNAME");
        char[] password =
System.getenv("MVNW_PASSWORD").toCharArray();
        Authenticator.setDefault(new Authenticator())
{
    @Override
    protected PasswordAuthentication
getPasswordAuthentication() {
    return new
PasswordAuthentication(username, password);
}
});
}
URL website = new URL(urlString);
ReadableByteChannel rbc;
rbc = Channels.newChannel(website.openStream());
FileOutputStream fos = new
FileOutputStream(destination);
fos.getChannel().transferFrom(rbc, 0,
Long.MAX_VALUE);
fos.close();
rbc.close();
}
}
```

maven-wrapper.properties

```
distributionUrl=https://repo.maven.apache.org/maven2/org
/apache/maven/apache-maven/3.8.2/apache-maven-3.8.2-
bin.zip
```

```
wrapperUrl=https://repo.maven.apache.org/maven2/io/takari/maven-wrapper/0.5.6/maven-wrapper-0.5.6.jar
```

Foodbox.sql

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
CREATE TABLE `address` (
  `id` int(11) NOT NULL,
  `street` varchar(55) NOT NULL,
  `city` varchar(55) NOT NULL,
  `zipCode` varchar(55) NOT NULL,
  `user_id` int(55) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
CREATE TABLE `cuisine` (
  `id` bigint(11) NOT NULL,
  `cuisine_name` varchar(411) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `cuisine` (`id`, `cuisine_name`) VALUES
(1, 'Arab'),
(2, 'Chinese'),
(3, 'Persian'),
(4, 'Indian'),
(5, 'Italian');
CREATE TABLE `product` (
  `id` bigint(11) NOT NULL,
  `name` varchar(111) NOT NULL,
```

```
`description` varchar(1000) DEFAULT NULL,
`price` decimal(55,0) DEFAULT NULL,
`image_url` varchar(500) DEFAULT NULL,
`date_created` datetime(6) DEFAULT NULL,
`last_updated` datetime(6) DEFAULT NULL,
`cuisine_id` bigint(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `product` (`id`, `name`, `description`,
`price`, `image_url`, `date_created`, `last_updated`,
`cuisine_id`) VALUES
(1, 'Hummus', 'There's a lot of controversy as to where hummus is really from. Is it Israeli, Greek or Arabic? Although the answer is not clear, there's no doubt that this is a staple dish in Arabic cuisine, making hummus a must-try for anyone looking to get better acquainted with the food from this region. This simple dish is made of chickpeas, tahini, garlic and lemon - and is simply delicious. It's best to order hummus to share and eat it with pita bread, it makes for an amazing appetizer!', '20', 'https://img.theculturetrip.com/1440x/smart/wp-content/uploads/2017/07/hummus-1058001_1920.jpg', '2021-09-15 23:17:00.000000', '2021-09-13 23:17:03.000000',
1),
(2, 'Fattoush', 'This delicious bread salad is traditional to the Levantine cuisine, common in Arabic countries such as Lebanon and Syria. Fattoush is made with mixed greens and small pieces of fried Arabic bread, giving some crunchiness. It often also includes pomegranate. This is again known as a great salad to share with others, making the dining experience a way of bringing people together and bonding over the mouth-
```

watering cuisine.', '23',
'https://img.theculturetrip.com/1440x807/smart/wp-content/uploads/2017/07/starter-1872862_1920-e1501253889682.jpg', '2021-07-15 23:17:00.000000',
'2021-09-06 08:52:35.000000', 1),
(4, 'Manakeesh', 'The pizza of the Arabic world,
manakeesh is a round bread sprinkled with either cheese,
ground meat or herbs (zaatar). It\'s ideal for breakfast
or lunch. Varieties come from both fancy Levantine
restaurants or street vendors.', '33',
'https://dynaimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2Fcdn.cnn.com%2Fcnnnext%2Fdam%2Fassets%2F170209174500-manakeesh.jpg', '2021-07-15
23:17:00.000000', '2021-07-13 23:17:03.000000', 1),
(5, 'Moutabal', 'Just when you\'re ready to declare
hummus the best dip on the planet, you find moutabal.
Similar to baba ghanoush, the dip offers a similar
consistency with an eggplant kick. Spiced up with chili,
it delivers a zing.', '12',
'https://dynaimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2Fcdn.cnn.com%2Fcnnnext%2Fdam%2Fassets%2F170209172813-baba-ganoush.jpg', '2021-
07-15 23:17:00.000000', '2021-07-15 23:17:00.000000',
1),
(6, 'Umm ali', 'Egyptian bread pudding, or umm ali, is a
hearty pastry cooked in milk and cream. Versions are
made with croissant pieces, raisins, pistachios, vanilla
and condensed milk.', '12',
'https://dynaimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2Fcdn.cnn.com%2Fcnnnext%2Fdam%2Fassets%2F170209172813-baba-ganoush.jpg'

Fdam%2Fassets%2F170209174450-umm-ali.jpg', '2021-01-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 1),
(7, 'Shanklish', 'This cow or sheep milk cheese is usually fashioned into golf ball-sized bites and rolled in zaatar herbs or chili flakes (the latter version favored in Syria). Also often enjoyed with diced tomato, onion and olive oil.', '31',
https://dynainimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2F%2Fcdn.cnn.com%2Fcnnnext%2Fdamp%2Fassets%2F170209174456-shanklish.jpg', '2020-05-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 1),
(8, 'Shish tawook', 'Served with pure garlic paste, this simple skewered chicken dish is hugely popular in Lebanon, Syria, Egypt, Iraq and the Gulf region. Dishes are served with fries and pita bread.', '44',
https://dynainimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2F%2Fcdn.cnn.com%2Fcnnnext%2Fdamp%2Fassets%2F170209174452-shish-tawook.jpg', '2021-03-14 23:24:19.000000', '2021-05-19 23:24:19.000000', 1),
(9, 'Dolma', 'Dolma goes horribly wrong when the stuffed vine leaves become flaccid and slimy after being left out in the sun for too long. They need to be fresh with succulent lamb or juicy vegetables.', '31',
https://dynainimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2F%2Fcdn.cnn.com%2Fcnnnext%2Fdamp%2Fassets%2F170209172805-dolma.jpg', '2021-08-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 1),
(10, 'Kofta', 'Common in Iran and Pakistan, these balls of minced lamb or beef have a spicy, onion kick.\r\nYou can fry, grill, barbecue or bake the patties, but

they\re best served [with](#) a distinctive spicy sauce. In the Arab region, you\ll find them in cylinder shapes, often on a stick.', '41',
'https://dynaimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2F%2Fcdn.cnn.com%2Fcnnnext%2Fdam%2Fassets%2F170209172746-kofta.jpg', '2021-11-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 1),
(11, 'Knafeh', 'This delicious cheesecake uses Nabusi cheese, which is common to Palestine, Syria and Lebanon. The blush coloring comes from orange blossom water or rose water.', '31',
'https://dynaimage.cdn.cnn.com/cnn/q_auto,w_602,c_fill,g_auto,h_339,ar_16:9/http%3A%2F%2Fcdn.cnn.com%2Fcnnnext%2Fdam%2Fassets%2F170209172748-knafeh.jpg', '2021-07-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 1),
(12, 'Murgh Makhani', 'Perhaps one of the most familiar Indian dish to the American diner, butter chicken first appeared in Delhi in the 1940s. It has a mild flavor that won\nt blow [out](#) your tastebuds, but you can increase the white pepper [or](#) curry powder for a more assertive taste.', '54',
'[https://www.thespruceeats.com/thmb/HkR_HR9fqAR3HaC9cYNwwE9diBo=/974x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/butter-chicken-479366-step-12-3536fe3a146d4f31abafaf571790e544.jpg](https://www.thespruceeats.com/thmb/HkR_HR9fqAR3HaC9cYNwwE9diBo=/974x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/butter-chicken-479366-step-12-3536fe3a146d4f31abafaf571790e544.jpg)', '2017-08-09 23:17:00.000000', '2021-08-12 23:17:00.000000', 4),
(13, 'Chole', 'In Northern India, classic [and](#) easy chole chickpea curry often appears [as](#) a favorite menu item [and](#) [as](#) a result, has become a worldwide sensation. It can also serve a crowd, especially [if](#) you serve it hot along [with](#) fried Indian leavened bread like poori [or](#) bhatura.

Once you have the chickpeas, onions, and tomatoes, along with garlic and ginger pastes, some common Indian spices will bring it all together.', '42',
'[https://www.thespruceeats.com/thmb/XfZ-PtUQPfh9QdurxuNpaxndDQ=/1300x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/chole-chickpea-curry-1957946-hero-01-cddf957cc78d4032ba29ec2adbc05a97.jpg](https://www.thespruceeats.com/thmb/XfZ-PtUQPfh9QdurxuNpaxndDQ=/1300x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/chole-chickpea-curry-1957946-hero-01-cddf957cc78d4032ba29ec2adbc05a97.jpg)', '2018-08-18 23:17:00.000000', '2021-08-12 23:17:00.000000', 4),
(14, 'Tandoori Chicken', 'Although traditionally cooked in a clay oven, you can prepare the yogurt-marinated charred chicken in a regular oven (or on the grill). You do need to plan ahead, as the chicken should sit in the refrigerator for at least 6 hours or overnight to let the flavors develop. If you prefer, you can sear the marinated chicken cubes first on the stovetop to achieve that signature tandoori char.', '52',
'[https://www.thespruceeats.com/thmb/SxIRu8AgdaHqXYKo10Gc9D97D1g=/1643x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/ChickenTandooriSpruce-f238bf7413984e088268565c5b660b11.jpg](https://www.thespruceeats.com/thmb/SxIRu8AgdaHqXYKo10Gc9D97D1g=/1643x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/ChickenTandooriSpruce-f238bf7413984e088268565c5b660b11.jpg)', '2019-05-07 23:24:19.000000', '2021-08-12 23:17:00.000000', 4),
(15, 'Chicken Vindaloo Curry', 'This is a mild and sweetly-spiced recipe for chicken vindaloo curry. Contrary to current belief, curries do not have to be hot and fiery. In fact, they never started out that way in India. Creating the curry paste is the most important part of this dish, so don\'t skimp on the ingredients.', '63',
'[https://www.thespruceeats.com/thmb/F8XL5PEpY4vE38-MEe8LDuclmtw=/900x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/ChickenVindalooCurry-1957946-hero-01-f8xl5pepy4ve38-me8lduclmtw=900x0](https://www.thespruceeats.com/thmb/F8XL5PEpY4vE38-MEe8LDuclmtw=/900x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/ChickenVindalooCurry-1957946-hero-01-f8xl5pepy4ve38-me8lduclmtw=900x0)

0):strip_icc():format(webp)/authentic-chicken-vindaloo-recipe-468712017-step-hero_1-57c6cc32841d4726b5f2eedd298b206d.jpg', '2021-08-12 23:17:00.000000', '2021-05-19 23:24:19.000000', 5), (16, 'Lemon Rice', 'South Indian lemon rice makes a great weekday meal because it comes together quickly. It often appears solo or alongside raita, yogurt, chutney, or kosambari (a type of salad).', '42', '[https://www.thespruceeats.com/thmb/MmwoX0JRUMSC3gv-aSXVvyX-PHE=1729x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/south-indian-lemon-rice-1957723-hero-02-18047e2504ad42fa96817e047c075ac3.jpg](https://www.thespruceeats.com/thmb/MmwoX0JRUMSC3gv-aSXVvyX-PHE=1729x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/south-indian-lemon-rice-1957723-hero-02-18047e2504ad42fa96817e047c075ac3.jpg)', '2021-05-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 4), (17, 'Naan', 'Naan, a puffy flatbread, goes with just about everything, including many popular dishes like tandoori chicken and all kinds of kebabs. While naan traditionally bakes in a tandoor or earthen oven, they work just as well in your own oven. The ingredient list includes yeast, flour, sugar, and water, as well as yogurt and ghee.', '24', '[https://www.thespruceeats.com/thmb/6P7g2QY-3QHWs6a58LrJ2tbbw2w=/1391x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/naan-leavened-indian-flatbread-1957348-final-08-116a2e523f6e4ee693b1a9655784d9b9.jpg](https://www.thespruceeats.com/thmb/6P7g2QY-3QHWs6a58LrJ2tbbw2w=/1391x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/naan-leavened-indian-flatbread-1957348-final-08-116a2e523f6e4ee693b1a9655784d9b9.jpg)', '2021-05-02 23:24:19.000000', '2021-05-19 23:24:19.000000', 4), (18, 'Malai Prawn', 'This Bengali speciality often appears alongside jeera rice. Cook the shrimp in creamy coconut milk with whole spices for a deeply flavorful dish. Using head-on shrimp will add even more shellfish

character, but you can peel and devein them first if you prefer.', '55',
'[https://www.thespruceeats.com/thmb/lyFzDP7IniEpb1eW1E21rzpcvYc=/1237x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/creamy-prawn-curry-1957454-hero-01-9e15ed0ddd334cf4a03533f0b72919b9.jpg](https://www.thespruceeats.com/thmb/lyFzDP7IniEpb1eW1E21rzpcvYc=/1237x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/creamy-prawn-curry-1957454-hero-01-9e15ed0ddd334cf4a03533f0b72919b9.jpg)', '2021-05-01 23:24:19.000000', '2021-05-19 23:24:19.000000', 4),
(19, 'Vegan Basmati Rice', 'Fragrant, tender, and the perfect accompaniment to just about every main, all Indian feasts need basmati rice. Try this restaurant-style vegan recipe that results in a wonderfully spiced grain.', '44',
'[https://www.thespruceeats.com/thmb/RLpF45cRVBDZAUzr83bTUg6L914=/1633x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/easy-vegan-indian-basmati-rice-recipe-3378479-hero-01-35f4f27d496c41fea81589a7dce17eb1.jpg](https://www.thespruceeats.com/thmb/RLpF45cRVBDZAUzr83bTUg6L914=/1633x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/easy-vegan-indian-basmati-rice-recipe-3378479-hero-01-35f4f27d496c41fea81589a7dce17eb1.jpg)', '2021-05-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 4),
(20, 'Tamarind-Stuffed Fish', 'A luxurious whole fish preparation flavored with tangy tamarind and fragrant barberries, perfect for Nowruz, the Persian new year. Find barberries in well-stocked Middle Eastern groceries. Get the recipe for Persian Tamarind-Stuffed Fish » Matt Taylor-Gross', '88',
'<https://www.saveur.com/uploads/2019/02/08/VWK5QXQXNQDRTGHQGR3PR32C6M.jpg>', '2021-08-12 23:17:00.000000', '2021-08-12 23:17:00.000000', 3),
(21, 'Chelo ba Tahdig', 'Iranians usually serve chelo ba tahdig with stew, kebabs, or meat dishes. Herbs and alliums, feta, or walnuts often share the table. Get the recipe for Steamed Saffron Rice with Tahdig (Chelo ba

Tahdig) » Zachary Zavislak', '63',
'<https://www.saveur.com/uploads/2019/03/11/WSVNQHIRDDTVZHCGAMFEB5ZCM-1024x768.jpg>', '2017-07-02
23:17:00.000000', '2021-07-15 23:17:00.000000', 3),
(22, 'Kuku Sabzi', 'A classic Persian herb-loaded egg
dish with the fragrant lift of rose petals. Get the
recipe for Persian Kuku Sabzi » Matt Taylor-Gross',
'53',
'<https://www.saveur.com/uploads/2019/01/16/R2FL55HMXTZNKX7U75S4YEYS4U-1024x769.jpg>', '2021-05-19
23:24:19.000000', '2021-05-19 23:24:19.000000', 3),
(23, 'Tahcheen-e Morgh', 'Bake this tahcheen, a saffron
rice with chicken, in a clear glass baking dish so you
can check on the crisping of the rice's bottom layer.
Get the recipe for Baked Saffron Yogurt Rice with
Chicken (Tahcheen-e Morgh) » Zachary Zavislak', '36',
'<https://www.saveur.com/uploads/2019/03/11/SI4KF6ISWT2J7XCCEZRWOH7DFY-1024x768.jpg>', '2021-08-12
23:17:00.000000', '2021-08-12 23:17:00.000000', 3),
(24, 'Adas Polo', 'A thin layer of lavash bread can be
cooked beneath the rice to serve as a tahdig. Press it
flat against the pot for even cooking.', '63',
'<https://www.saveur.com/uploads/2019/03/11/L2N233EB3VWQE43J7IJHYBS3Z4-1024x768.jpg>', '2021-08-12
23:17:00.000000', '2021-08-12 23:17:00.000000', 3),
(25, 'Sichuan Pork', 'Sichuan Pork, actually, poached
spicy slices of pork) is a famous Sichuan cuisine
dish.\r\n\r\nRather than cooking pork by stir-frying or
deep-frying, which consumes lots of oil and can very
easily overcook the pork, it is boiled in water with a
coating made from egg-white and starch to preserve its

freshness and tenderness.\r\n\r\nThe meaty broth is typical Sichuan cuisine, featuring a peppery and spicy taste. When eating, you'll find each piece of meat contains abundant juices with a fresh and fragrant spicy aroma.', '68',
'https://images.chinahighlights.com/allpicture/2019/11/7b3afb974d7946bbbe397f7c_894x670.webp', '2021-07-15 23:17:00.000000', '2021-07-13 23:17:03.000000', 2),
(26, 'Braised Pork Balls in Gravy', 'Braised pork balls in gravy are also known as "Chinese meatballs" (many of our clients call them this). If you like meatballs and spaghetti, you'll probably appreciate Chinese meatballs as well. The super-smooth and flavorful gravy is almost like a "drug sauce" keeping your taste buds wanting more and more. How many calories does it have? Trust me, you wouldn't care at all!\r\n\r\nTip: We suggest that you only order this dish in a decent restaurant where the meat quality is guaranteed and the prescribed lean: fat pork ratio of 7:3 is achieved in each meat ball.', '61',
'https://images.chinahighlights.com/allpicture/2019/11/a47d9323b7ae409ba96e628d_894x596.webp', '2022-07-29 23:17:00.000000', '2021-07-15 23:17:00.000000', 2),
(27, 'Shrimp with Vermicelli and Garlic', 'Shrimp with vermicelli and is a dish not only favored by foreigners but also the Chinese younger generation in recent years as well.\r\n\r\nThis steamed dish brings out the aroma of everything it has. The heavy garlic with a seafood sauce flavor blends with the freshness of juicy shrimp. The vermicelli at the bottom absorbs every single drip of tasty juice running down. Some Chinese yellow wine to capture the aroma and highlight the flavor is the secret

to this delicacy.', '14',
'https://images.chinahighlights.com/allpicture/2019/11/fd729ddd3fdf4fd9beefc79e_894x594.webp', '2021-05-19 23:24:19.000000', '2021-05-19 23:24:19.000000', 2),
(28, 'Chow Mein', '\"Chow mein\" (炒面 chǎomiàn) is the Cantonese pronunciation of the Chinese characters above, which means stir-fried noodles. Generally speaking, this stir-fried dish consists of noodles, meat (usually chicken, beef, shrimp, or pork), onions and celery.\r\n\r\nFor making chow mein, the noodles need to be cooked in boiling water for a while. After they become cool, then move to the step of stir-frying. S', '28',
'https://images.chinahighlights.com/allpicture/2018/03/10725167158a4b8589bf3140_894x596.webp', '2021-08-15 23:17:00.000000', '2021-05-19 23:24:19.000000', 2),
(29, 'Sweet and Sour Pork', 'Sweet and sour pork has a bright orange-red color, and a delicious sweet and sour taste.\r\n\r\nAt the very beginning there was only sweet and sour pork, but to meet demands, there have been some developments on this dish. Now, the pork can be substituted by other ingredients like chicken, beef or pork ribs. See How to Cook Sweet and Sour Pork.', '68',
'https://images.chinahighlights.com/allpicture/2019/11/a4ad4a7fe0cb401cb0be6383_894x598.webp', '2021-05-25 23:24:19.000000', '2021-05-19 23:24:19.000000', 2),
(31, 'Pizza', 'Though a slab of flat bread served with oil and spices was around long before the unification of Italy, there's perhaps no dish that is as common or as representative of the country as the humble pizza. Easy, cheap, and filling, pizza has long been a common snack

or meal, especially in Naples where tomato sauce was first added. When the Italian Queen Margherita came through the bustling city on a tour of her kingdom in 1889 she asked to try this dish that she saw so many of her subjects eating. A local entrepreneur served her the now legendary combination of tomato sauce, mozzarella, and basil, creating (or more likely, branding) the Margherita pizza. Whether by coincidence or design, the Margherita also displays the colors of the Italian flag.', '25', 'https://www.walksofitaly.com/blog/wp-content/uploads/2011/05/DSC_0195-768x510.jpg', '2014-07-05 23:17:00.000000', '2021-07-15 23:17:00.000000', 5), (32, 'Lasagna', 'Lasagna is a wide, flat pasta noodle, usually baked in layers in the oven. Like most Italian dishes, its origins are hotly contested, but we can at least say that's its stronghold is in the region of Emilia-Romagna, where it transformed from a poor man's food to a rich meal filled with the ragù, or meat sauce.', '86', 'https://www.walksofitaly.com/blog/wp-content/uploads/2016/07/11735294004_b68f0ab3ef_k-768x512.jpg', '2017-05-10 23:24:19.000000', '2021-05-19 23:24:19.000000', 5), (33, 'Ossobuco', 'The world-famous ossobuco alla milanese is a bone-in veal shank, cooked low and slow until meltingly tender in a broth of meat stock, white wine, and veggies. Traditionally, it's accompanied by a gremolata (lemon zest, garlic, and parsley) but that's optional. Although the Milanese like to claim this meaty masterpiece there are as many versions of it as there are nonnas in Lombardy, which is known for hearty, often rustic dishes that are good at coating the ribs and

staving off the winter chill. Despite the popularity of ossobuco (which literally means ‘hollow bone’), it’s not always common to see it on restaurant menus because it needs about three hours of cooking time. If you do get a chance to eat it in a restaurant or home, or even to cook it yourself, you should jump at the opportunity. It’s usually accompanied by polenta or the next item on our list.', '73', '<https://www.walksofitaly.com/blog/wp-content/uploads/2014/05/Osso-Buco-768x515.jpg>', '2021-08-24 23:17:00.000000', '2021-05-19 23:24:19.000000', 5),
(34, 'Polpette', 'Polpette can be made of meat, fish, potatoes, rice or vegetables and can be prepared stewed, baked or fried.', '63',
'<http://yourguidetoitaly.com/slowitaly/wp-content/uploads/2018/03/polpette-c.jpg>', '2021-07-15 23:17:00.000000', '2021-07-12 23:17:03.000000', 5);

```
CREATE TABLE `users` (
  `id` bigint(11) NOT NULL,
  `type` int(11) NOT NULL DEFAULT 0,
  `username` varchar(111) NOT NULL,
  `email` varchar(111) NOT NULL,
  `password` varchar(111) NOT NULL,
  `phone` int(22) DEFAULT NULL,
  `street` varchar(50) DEFAULT NULL,
  `city` varchar(50) DEFAULT NULL,
  `zipcode` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `users` (`id`, `type`, `username`, `email`,
`password`, `phone`, `street`, `city`, `zipcode`) VALUES
```

```
(1, 1, 'admin', 'admin@admin.com', 'admin', 5555555,
'he', 'h', 'h'),
(2, 0, 'admin2', 'admin2@admin.com', 'admin2', 21, 'he',
'h', 'h'),
(4, 0, '2', 'mm@f.com', '1314', 1414, 'Al Dawhah Al
Janubiyah, Dhahran 34457', 'Dahran', '34466'),
(5, 0, '1', '1', '1', 1, '1', '1', '1');

ALTER TABLE `address`
    ADD PRIMARY KEY (`id`);

ALTER TABLE `cuisine`
    ADD PRIMARY KEY (`id`);

ALTER TABLE `product`
    ADD PRIMARY KEY (`id`),
    ADD KEY `cuisine_id` (`cuisine_id`);

ALTER TABLE `users`
    ADD PRIMARY KEY (`id`);

ALTER TABLE `address`
    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `cuisine`
    MODIFY `id` bigint(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;

ALTER TABLE `product`
    MODIFY `id` bigint(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=58;

ALTER TABLE `users`
```

```
MODIFY `id` bigint(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=6;  
  
USE `phpmyadmin`;  
INSERT INTO `pma_table_uiprefs`(`username`, `db_name`,  
`table_name`, `prefs`, `last_update`) VALUES  
('root', 'foodbox', 'product',  
'{"sorted_col": "\`product\`.\`date_created\` ASC"}',  
'2021-09-17 09:43:27');  
COMMIT;
```

MyDataRestConfig.java

```
package com.simplilearn.foodbox.config;  
import com.simplilearn.foodbox.entity.User;  
import com.simplilearn.foodbox.entity.Cuisine;  
import com.simplilearn.foodbox.entity.Product;  
import  
org.springframework.beans.factory.annotation.Autowired;  
import  
org.springframework.context.annotation.Configuration;  
import  
org.springframework.data.rest.core.config.RepositoryRest  
Configuration;  
import  
org.springframework.data.rest.webmvc.config.RepositoryRe  
stConfigurer;  
import org.springframework.http.HttpMethod;
```

```
import
org.springframework.web.servlet.config.annotation.CorsRe
gistry;

import javax.persistence.EntityManager;
import javax.persistence.metamodel.EntityType;
import java.util.ArrayList;
import java.util.List;
import java.util.Set;

@Configuration
public class MyDataRestConfig implements
RepositoryRestConfigurer {

    private EntityManager entityManager;

    @Autowired
    public MyDataRestConfig(EntityManager theEntityManager) {
        entityManager = theEntityManager;
    }

    @Override
    public void
configureRepositoryRestConfiguration(RepositoryRestConfi
guration config, CorsRegistry cors) {

        HttpMethod[] theUnsupportedActions =
{HttpMethod.PUT, HttpMethod.POST, HttpMethod.DELETE};
        config.getExposureConfiguration()
            .forDomainType(Product.class)
```

```
        .withItemExposure((metadata, httpMethods)
-> httpMethods.disable(theUnsupportedActions))
        .withCollectionExposure((metadata,
httpMethods) ->
httpMethods.disable(theUnsupportedActions));

        config.getExposureConfiguration()
            .forDomainType(Cuisine.class)
            .withItemExposure((metadata, httpMethods)
-> httpMethods.disable(theUnsupportedActions))
            .withCollectionExposure((metadata,
httpMethods) ->
httpMethods.disable(theUnsupportedActions));
        config.getExposureConfiguration()
            .forDomainType(User.class)
            .withItemExposure((metadata, httpMethods)
-> httpMethods.disable(theUnsupportedActions))
            .withCollectionExposure((metadata,
httpMethods) ->
httpMethods.disable(theUnsupportedActions));

        // call an internal helper method
        exposeIds(config);
    }
    private void exposeIds(RepositoryRestConfiguration
config) {
        Set<EntityType<?>> entities =
entityManager.getMetamodel().getEntities();
        List<Class> entityClasses = new ArrayList<>();
        for (EntityType tempEntityType : entities) {
```

```
entityClasses.add(tempEntityType.getJavaType());
    }
    Class[] domainTypes = entityClasses.toArray(new
Class[0]);
    config.exposeIdsFor(domainTypes);
}
}
```

productController.java

```
package com.simplilearn.foodbox.controller;

import com.simplilearn.foodbox.dao.ProductRepository;
import com.simplilearn.foodbox.entity.Product;
import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.data.rest.webmvc.ResourceNotFoundException;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@CrossOrigin(origins = "http://localhost:4200")
@RestController
@RequestMapping("/api/v1/")
```

```
public class ProductController {  
  
    @Autowired  
    private ProductRepository productRepository;  
  
    // get all products  
    @GetMapping("/products")  
    public List<Product> getAllproducts(){  
        return productRepository.findAll();  
    }  
  
    // create product rest api  
    @PostMapping("/products")  
    public Product createProduct(@RequestBody Product  
product) {  
        return productRepository.save(product);  
    }  
    @GetMapping("/products/{id}")  
    public ResponseEntity<Product>  
getproductById(@PathVariable Long id) {  
        Product product = productRepository.findById(id)  
            .orElseThrow(() -> new  
ResourceNotFoundException("product not exist with id :"  
+ id));  
        return ResponseEntity.ok(product);  
    }  
    @PutMapping("/products/{id}")  
    public ResponseEntity<Product>  
updateProduct(@PathVariable Long id, @RequestBody  
Product productDetails){  
        Product product = productRepository.findById(id)  
            .orElseThrow(() -> new  
ResourceNotFoundException("product not exist with id :"  
+ id));  
        product.setName(productDetails.getName());  
        product.setPrice(productDetails.getPrice());  
        product.setQuantity(productDetails.getQuantity());  
        productRepository.save(product);  
        return ResponseEntity.ok(product);  
    }  
}
```

```
        .orElseThrow(() -> new
ResourceNotFoundException("product not exist with id :"
+ id));

        product.setName(productDetails.getName());

product.setDescription(productDetails.getDescription());
        product.setPrice(productDetails.getPrice());

product.setImageUrl(productDetails.getImageUrl());


        Product updatedProduct =
productRepository.save(product);
        return ResponseEntity.ok(updatedProduct);
    }

// delete product rest api
@DeleteMapping("/products/{id}")
public ResponseEntity<Map<String, Boolean>>
deleteProduct(@PathVariable Long id){
    Product product = productRepository.findById(id)
        .orElseThrow(() -> new
ResourceNotFoundException("product not exist with id :"
+ id));

    productRepository.delete(product);
    Map<String, Boolean> response = new HashMap<>();
    response.put("deleted", Boolean.TRUE);
    return ResponseEntity.ok(response);
}
```

UserController.java

```
package com.simplilearn.foodbox.controller;

import com.simplilearn.foodbox.dao.UserRepository;
import com.simplilearn.foodbox.entity.User;
import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.data.rest.webmvc.ResourceNotFoundException;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import java.util.List;
@CrossOrigin(origins = "http://localhost:4200")
@RestController
@RequestMapping("/api/v1/")
public class UserController {

    @Autowired
    private UserRepository userRepository;
    @GetMapping("/users")
    public List<User> getAllUsers(){
        return userRepository.findAll();
    }
    @PostMapping("/users")
    public User createUser(@RequestBody User user) {
        return userRepository.save(user);
    }
    @GetMapping("/users/{id}")
```

```
public ResponseEntity<User>
getUserById(@PathVariable Long id) {
    User user = userRepository.findById(id)
        .orElseThrow(() -> new
ResourceNotFoundException("user not exist with id :" +
id));
    return ResponseEntity.ok(user);
}
```

CuisineRepository.java

```
package com.simplilearn.foodbox.dao;

import com.simplilearn.foodbox.entity.Cuisine;
import
org.springframework.data.jpa.repository.JpaRepository;
import
org.springframework.data.rest.core.annotation.Repository
RestResource;
import
org.springframework.web.bind.annotation.CrossOrigin;

@CrossOrigin("http://localhost:4200")
public interface CuisineRepository extends
JpaRepository<Cuisine, Long> {
}
```

ProductRepository.java

```
package com.simplilearn.foodbox.dao;
```

```
import com.simplilearn.foodbox.entity.Product;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import
org.springframework.data.jpa.repository.JpaRepository;
import
org.springframework.web.bind.annotation.CrossOrigin;
import
org.springframework.web.bind.annotation.RequestParam;

@CrossOrigin("http://localhost:4200")
public interface ProductRepository extends
JpaRepository<Product, Long> {
    Page<Product> findAllByOrderByNameAsc(Pageable
pageable);

    Page<Product>
findAllByOrderByDateCreatedDesc(Pageable pageable);

    Page<Product> findByCuisine(@RequestParam("cuisine")
int cuisine, Pageable pageable);

    Page<Product>
findByCuisineOrderByNameAsc(@RequestParam("cuisine") int
cuisine, Pageable pageable);

    Page<Product>
findByCuisineOrderByDateCreatedDesc(@RequestParam("cuisi
ne") int cuisine, Pageable pageable);

    // for searching
```

```
    Page<Product>
    findByNameContaining(@RequestParam("name") String name,
    Pageable pageable);

    Page<Product>
    findByNameContainingOrderByAsc(@RequestParam("name")
    String name, Pageable pageable);

    Page<Product>
    findByNameContainingOrderByDateCreatedDesc(@RequestParam
    ("name") String name, Pageable pageable);

}
```

UserRepository.java

```
package com.simplilearn.foodbox.dao;

import com.simplilearn.foodbox.entity.User;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import
org.springframework.data.jpa.repository.JpaRepository;
import
org.springframework.web.bind.annotation.CrossOrigin;
import
org.springframework.web.bind.annotation.RequestParam;
```

```
@CrossOrigin("http://localhost:4200")
public interface UserRepository extends
JpaRepository<User, Long> {

    Page<User> findByUsername(@RequestParam("username")
String username, Pageable pageable);
```

Address.java

```
package com.simplilearn.foodbox.entity;
import lombok.Data;

import javax.persistence.*;

@Entity
@Table(name="address")
@Data
public class Address {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;
```

Cuisine.java

```
package com.simplilearn.foodbox.entity;
```

```
import lombok.Getter;
import lombok.Setter;

import javax.persistence.*;

@Entity
@Table(name="cuisine")
// @Data -- known bug
@Getter
@Setter
public class Cuisine {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;

    @Column(name = "cuisine_name")
    private String cuisine_name;

    public String getCuisine_name() {
        return cuisine_name;
    }

    public void setCuisine_name(String cuisine_name) {
        this.cuisine_name = cuisine_name;
    }

    public Long getId() {
        return id;
    }
}
```

```
    public void setId(Long id) {  
        this.id = id;  
    }  
}
```

Product.java

```
package com.simplilearn.foodbox.entity;  
import lombok.Data;  
import org.hibernate.annotations.CreationTimestamp;  
import org.hibernate.annotations.UpdateTimestamp;  
  
import javax.persistence.*;  
import java.math.BigDecimal;  
import java.util.Date;  
  
@Entity  
@Table(name="product")  
@Data  
public class Product {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private Long id;  
  
    @Column(name = "cuisine_id")  
    private int cuisine;  
  
    @Column(name = "name")
```

```
private String name;

@Column(name = "description")
private String description;

@Column(name = "price")
private BigDecimal price;

@Column(name = "image_url")
private String imageUrl;

@Column(name = "date_created")
@CreationTimestamp
private Date dateCreated;

@Column(name = "last_updated")
@UpdateTimestamp
private Date lastUpdated;

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getDescription() {
    return description;
}
```

```
public void setDescription(String description) {
    this.description = description;
}

public BigDecimal getPrice() {
    return price;
}

public void setPrice(BigDecimal price) {
    this.price = price;
}

public String getImageUrl() {
    return imageUrl;
}

public void setImageUrl(String imageUrl) {
    this.imageUrl = imageUrl;
}

public Date getDateCreated() {
    return dateCreated;
}

public void setDateCreated(Date dateCreated) {
    this.dateCreated = dateCreated;
}

public Date getLastUpdated() {
    return lastUpdated;
}
```

```
}

public void setLastUpdated(Date lastUpdated) {
    this.lastUpdated = lastUpdated;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}
}
```

User.java

```
package com.simplilearn.foodbox.entity;
import lombok.Data;

import javax.persistence.*;

@Entity
@Table(name="users")
@Data
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
```

```
private Long id;

@Column(name = "username")
private String username;

@Column(name = "email")
private String email;

@Column(name = "password")
private String password;

@Column(name = "phone")
private int phone;

@Column(name = "street")
private String street;

@Column(name = "city")
private String city;

@Column(name = "zipcode")
private String zipcode;

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}
```

```
public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public int getPhone() {
    return phone;
}

public void setPhone(int phone) {
    this.phone = phone;
}

public String getStreet() {
    return street;
}

public void setStreet(String street) {
    this.street = street;
}
```

```
public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
}

public String getZipcode() {
    return zipcode;
}

public void setZipcode(String zipcode) {
    this.zipcode = zipcode;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}
```

Application-foodbox.java

```
package com.simplilearn.foodbox;
```

```
import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class FoodboxApplication {

    public static void main(String[] args) {
        SpringApplication.run(FoodboxApplication.class,
args);
    }

}
```

FoodboxApplicationTests.java

```
package com.simplilearn.foodbox;
import org.junit.jupiter.api.Test;
import
org.springframework.boot.test.context.SpringBootTest;
@SpringBootTest
class FoodboxApplicationTests {
    @Test
    void contextLoads() {
    }
}
```

Front-end

Add-product.component.html

```
<br><br>
<div class="login-form" style="width:44%" >
  <form (ngSubmit) = "handleAddProduct()">
    <h2 class="text-center">Add a New Product</h2>

    <div class="form-group">
      Name:<input [(ngModel)] = "product.name"
name = "name" type="text" class="form-control"
required="required">
    </div>
    <br>
    <div class="form-group">
      Description:
      <input [(ngModel)] = "product.description"
name = "description" type="text" class="form-control"
required="required" >
    </div>
    <br>
    <div class="form-group">
      Price:
      <input [(ngModel)] = "product.price" name =
"price" type="text" class="form-control"
required="required">
    </div>
    <br>
    <div class="form-group">
      Image URL:
```

```

        <input [(ngModel)] = "product.imageUrl" name
= "imageUrl" type="text" class="form-control"
required="required">
    </div>
    <br>
    <div class="form-group">
        Cuisine:
        <br>
        <select [(ngModel)]="product.cuisine" name =
"selectedCuisine">
            <option *ngFor="let cuisine of
productCuisines"
[ngValue]="cuisine.id">{{cuisine.cuisine_name}}</option>
        </select>
    </div>
    <br>
    <div class="form-group">
        <button type="submit" class="btn btn-success
btn-block">Add</button>
    </div>
</form>
</div>

```

Cart-details.component.html

```

<div class="main-content">
    <div class="section-content section-content-p30">
        <div class="container-fluid">

            <div *ngIf="username == null" class="alert
alert-danger col-md-12" role="alert">

```



```
<div class="row no-gutters">
    <div class="col">
        <button
            (click)="incrementQuantity(tempCartItem)" class="btn btn-primary btn-sm">
            <i
                class="fas fa-plus"></i>
        </button>
    </div>

    <div class="col ml-4 mr-2">
        {{ tempCartItem.quantity }}
    </div>

    <div class="col">
        <button
            (click)="decrementQuantity(tempCartItem)" class="btn btn-primary btn-sm">
            <i
                class="fas fa-minus"></i>
        </button>
    </div>

<div class="col-8"></div>
</div>
```

```
        </div>

        <button
(click)="remove(tempCartItem)" class="au-btn-submit"
style="width:22px; background: red; color: white"
>Remove</button>

                <p class="mt-2"></p>
            </td>
        </tr>

        <tr>
            <td colspan="2"></td>
            <td style="font-weight: bold">
                <p>Total Quantity: {{ totalQuantity }}</p>
                <p>Total Price: {{ totalPrice | currency: 'USD' }}</p>
                <a (click)="getTotalPrice()" routerLink="/checkout"><button class="au-btn-submit"
style="width:181px; height: 55px; background: green;
color: white" >Confirm</button></a>
            <br>
        </td>
    </tr>
</table>
<div *ngIf="cartItems.length == 0"
class="alert alert-warning col-md-12" role="alert">
    Your shopping cart is empty.
</div>
```

```
        </div>
    </div>
</div>
```

Checkout.html

```
<div class="main-content">
    <div class="section-content section-content-p30">
        <div class="container-fluid">

            <div style="margin-left: 15px;">
                <h6 style="font-size:22px;">
                    <p>Thank you <b>{{username}}</b> for
                purchasing our products. Your support and trust in us
                are much appreciated. </p>
                    <p>The total price of your bill is
                <b>${{totalPrice}}</b> </p>
                    <p>The products will be delivered to you soon
                to your address as following:</p>
                    <p><b>city:</b> {{this.users[0]?.city}}</p>
                    <p><b>street:</b>
                {{this.users[0]?.street}}</p>
                    <p><b>zipcode:</b>
                {{this.users[0]?.zipcode}}</p>
                </h6>
            </div>
            <a (click)="backHome()"><button class="au-btn-submit" style="width:181px; height: 55px; background: green; color: white" >Confirm</button></a>
        </div>
```

```
    </div>
</div>
```

Login.component.html

```
<br>
<br>
<div class="login-form">
  <form (ngSubmit)="onSubmit()" >
    <h2 class="text-center">Log in</h2>
    <div>
      <div class="form-group">
        <input [(ngModel)] = "user.username" name =
"username" type="text" class="form-control"
placeholder="Username" required="required">
      </div>
      <br>
      <div class="form-group">
        <input [(ngModel)] = "user.password" name =
"password" type="password" class="form-control"
placeholder="Password" required="required">
      </div>
      <br>
      <div class="form-group">
        <button type="submit" class="btn btn-primary
btn-block">Log in</button>
      </div>
    </div>
  </form>
  <p class="text-center"><a
routerLink="/register">Create an Account</a></p>
```

```
</div>
```

Product-category-menu.component.html

```
<div class="menu-sidebar-content js-scrollbar1">
  <nav class="navbar-sidebar">
    <ul class="list-unstyled navbar-list">

      <li style="margin-left:-17px;font-size:27px;
font-family: Helvetica, sans-serif;"> Cuisines:</li>

      <li style="line-height: 2.3;">
        <i class="fa fa-angle-double-right"
style="font-size:14px"></i>
        <a style="text-decoration:none"
routerLink="/category" [ngClass]="{{ 'active-
link':clicked }}" (click)="Clicked()">
          All
        </a>
      </li>
      <li style="line-height: 2.3;" *ngFor="let
tempProductCategory of productCategories">
        <i class="fa fa-angle-double-right"
style="font-size:14px"></i>
        <a style="text-decoration:none"
routerLink="/category/{{ tempProductCategory.id }}"
routerLinkActive="active-link" (click)="UnClicked()">
          {{ tempProductCategory.cuisine_name
}}
        </a>
      </li>
    </ul>
  </nav>
</div>
```

```
<br>

</li>
<br>

<li>
    <button *ngIf="username == 'admin'" routerLink="/add-product" class="btn btn-primary btn-sm">Add Product</button>
</li>
</ul>
</nav>
</div>
```

Register.component.html

```
<br><br>
<div class="login-form" >
    <form (ngSubmit) = "saveUser()">
        <h2 class="text-center">Register</h2>
        <div class="form-group">
            <input [(ngModel)] = "user.username" name = "username" type="text" class="form-control" placeholder="Username" required="required">
        </div>
        <br>
        <div class="form-group">
            <input [(ngModel)] = "user.email" name = "email" type="text" class="form-control" placeholder="Email" required="required">
        </div>
    </form>
</div>
```

```
</div>
<br>
<div class="form-group">
    <input [(ngModel)] = "user.password" name =
"password" type="password" class="form-control"
placeholder="Password" required="required">
</div>
<br>
<div class="form-group">
    <input [(ngModel)] = "user.phone" name =
"phone" type="text" class="form-control"
placeholder="phone" required="required">
</div>
<br>

<h6>Address</h6>
<div class="form-group">
    <input [(ngModel)] = "user.city" name =
"city" type="text" class="form-control"
placeholder="city" required="required">
</div>
<br>
<div class="form-group">
    <input [(ngModel)] = "user.street" name =
"street" type="text" class="form-control"
placeholder="street" required="required">
</div>
<br>
<div class="form-group">
```

```
        <input [(ngModel)] = "user.zipcode" name =
"zipcode" type="text" class="form-control"
placeholder="zipCode" required="required">
    </div>
    <br>

    <div class="form-group">
        <button type="submit" class="btn btn-primary
btn-block">Register</button>
    </div>

</form>
<p class="text-center"><a routerLink="/login">Back
to Login</a></p>
</div>
```

Search-component.html

```
div class="form-header">

    <input #myInput type="text"
        placeholder="Search for products ..."
        class="au-input au-input-xl"
        (keyup.enter)="doSearch(myInput.value)" />

    <button (click)="doSearch(myInput.value)" class="au-
btn-submit">
        Search
    </button>
```

```
</div>
```

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Foodbox</title>
  <base href="/">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="icon" type="image/x-icon"
href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

Karma.conf.js

```
module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-
angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
```

```
        require('karma-jasmine-html-reporter'),
        require('karma-coverage'),
        require('@angular-devkit/build-
angular/plugins/karma')
    ],
    client: {
        jasmine: {

        },
        clearContext: false      },
    jasmineHtmlReporter: {
        suppressAll: true
    },
    coverageReporter: {
        dir: require('path').join(__dirname,
'./coverage/foodbox'),
        subdir: '..',
        reporters: [
            { type: 'html' },
            { type: 'text-summary' }
        ]
    },
    reporters: ['progress', 'kjhtml'],
    port: 9876,
    colors: true,
    LogLevel: config.LOG_INFO,
    autoWatch: true,
    browsers: ['Chrome'],
    singleRun: false,
    restartOnFileChange: true
});
```

```
};
```

tsconfig.app.json

```
{  
  "extends": "./tsconfig.json",  
  "compilerOptions": {  
    "outDir": "./out-tsc/app",  
    "types": []  
  },  
  "files": [  
    "src/main.ts",  
    "src/polyfills.ts"  
  ],  
  "include": [  
    "src/**/*.d.ts"  
  ]  
}
```

tsconfig.spec.json

```
{  
  "extends": "./tsconfig.json",  
  "compilerOptions": {  
    "outDir": "./out-tsc/spec",  
    "types": [  
      "jasmine"  
    ]  
  },  
  "files": [  
    "src/test.ts"  
  ]  
}
```

```
"src/test.ts",
"src/polyfills.ts"
],
"include": [
  "src/**/*.spec.ts",
  "src/**/*.d.ts"
]
}
```

X